FACULTAD DE ESTUDIOS ESTADÍSTICOS

MÁSTER EN MINERÍA DE DATOS E INTELIGENCIA DE NEGOCIOS

Curso 2019/2020

Trabajo de Fin de Máster

TITULO: Minería de datos aplicada a los precios de las subastas agrícolas de la provincia de Almería

***Alumno*: Lorena Acién Pérez**

***Tutor*: Javier Castro Cantalejo**

Septiembre de 2019

UNIVERSIDAD COMPLUTENSE
MADRID

Resumen

La provincia de Almería es una de las principales exportadoras de hortalizas en Europa, utilizando agricultura intensiva bajo plástico. El desarrollo de la actividad agrícola ha sido el impulsor de la economía almeriense.

En este trabajo, realizaremos un análisis de los datos sobre los precios de las subastas públicas donde se comercializan hortalizas, concretamente las variedades de pimiento para las diferentes empresas de la provincia de Almería. Posteriormente, analizaremos modelos de predicción para, por un lado, predecir el precio de la subasta, y por otro, ver si a un agricultor de la zona le interesa vender sus hortalizas la semana próxima o el día actual. Todo esto lo realizaremos con técnicas estadísticas adquiridas durante el máster de Minería de Datos e Inteligencia de Negocio.

Palabras clave: Poniente Almeriense, modelos de predicción, subasta agrícola, minería de datos.

Abstract

The province of Almeria is one of the main exporters of vegetables in Europe, using intensive agriculture under plastic. The development of agricultural activity has been the driving force of Almeria's economy.

In this Project, we will make a data analysis of the prices in the public agricultural auctions where vegetables are marketed. In this work we will specifically focus on the pepper varieties for the different companies in the province of Almeria. Later, we will analyze prediction models for predicting prices in the public agricultural auctions. Secondly, we will investigate whether the vegetable producer is interested in selling their vegetables the following week or on the current day. This will be conducted using statistical techniques learned during the Data Mining and Business Intelligence Master.

Keywords: Poniente Almeriense, prediction models, farming sale, data mining.

Índice

1. Introducción 7
2. Objetivos 8
3. Metodología 9
4. Muestreo de la base de datos 17
5. Exploración de las variables 18
   I. Variables Cualitativas 18
   II. Variables Cuantitativas 20
6. Modificación de la base de datos 24
   I. Creación de variables 24
   II. Tratamiento de valores ausentes y atípicos 29
   III. Selección de variables 30
7. Modelización para variable objetivo continua 30
   I. Regresión Lineal 31
      i. Evaluación del mejor modelo 33
   II. Redes Neuronales 35
      i. Estudio de los parámetros para la función de activación 35
      ii. Estudio de los parámetros para el método de optimización 37
      iii. Estudio del número de nodos ocultos 38
      iv. Estudio de la posibilidad de aplicar Early Stopping 39
      v. Ejecución de los modelos 41
      vi. Evaluación del mejor modelo 42
8. Evaluación para variable objetivo continua 42
9. Modelización para variable objetivo dicotómica 43
   I. Regresión Logística 44
      i. Evaluación del mejor modelo 45
   II. Redes Neuronales 46
      i. Estudio del número de nodos y el punto de corte 47
      ii. Estudio de los parámetros para la función de activación 48
      iii. Estudio de los parámetros para el método de optimización 48
      iv. Ejecución de los modelos 51
      v. Evaluación del mejor modelo 52
   III. Bagging 53
      i. Evaluación del mejor modelo 56
   IV. Random Forest 56

i. Evaluación del mejor modelo 57
V. Gradient Boosting 57
i. Evaluación del mejor modelo 58
VI. Ensamblado 59
i. Evaluación del mejor modelo 61
10. Evaluación para variable objetivo dicotómica 61
11. Conclusiones 62
12. Bibliografía 64
13. ANEXO 65
I. SOFTWARE PYTHON: Muestreo de los datos – Web Scraping 65
II. SOFTWARE R: Modificación de la base de datos 68
III. SOFTWARE SAS: Modelización de la base de datos 81

1. Introducción

El elemento impulsor de la economía almeriense es la agricultura. Comenzó en los años 60 cuando se partió de una agricultura de secano, dedicada al olivo y a los cereales. Posteriormente, se comenzó a exportar uva de mesa y con ella, llegaron los primeros invernaderos que cambiarían el futuro de la agricultura almeriense, gracias al elevado nivel de producción tanto para la exportación internacional como para el consumo nacional.

Almería destaca por el cultivo bajo plástico que se lleva a cabo en invernaderos, es la zona de Europa donde se consiguen mayores rendimientos de cultivo de hortalizas por metro cuadrado. Se cultivan todo tipo de hortícolas como tomates, berenjenas, calabacines, judías, además del pimiento que es el producto más cultivado. Esto ha requerido los servicios de empresas de todo tipo relacionadas con el sector y auxiliares, lo que ha derivado en un sistema empresarial muy fuerte dedicado al sector hortofrutícola desde la construcción, semilleros y comercialización, pasando por servicios e investigación.

En la zona conocida como el Poniente Almeriense hay variedad de empresas especializadas en el sector agrícola, que se encargan de realizar subastas para comercializar las hortalizas por España e internacionalmente, las denominamos alhóndigas, pues su nombre proviene del castellano antiguo, alfóndiga, pues era un establecimiento donde se vendía y almacenaba grano, para abastecer a los labradores de la zona.

Las alhóndigas que se encargan de organizar las subastas públicas funcionan como intermediarias entre el agricultor y la empresa que se encarga de vender al consumidor las hortalizas, dichas empresas nombran a comerciales que asisten a las subastas donde pujan el precio que estén dispuestas a pagar por el producto que requieran.
Se realiza una subasta a la baja, donde se empieza a cotizar por encima del primer precio de la subasta del día anterior, sino hubiera, la anterior disponible, se realiza una subasta independiente por cada hortaliza. Los comerciales realizan sus pujas, una vez que termina, los comerciales desde el que ha pujado más hasta el que menos eligen viendo los distintos productos de los agricultores cuál llevarse y qué cantidad.

Los agricultores llevan sus hortalizas a la alhóndiga, tras ser pesadas, se exponen para que los comerciales elijan el producto que deseen comprar. La empresa encargada de realizar esta subasta se lleva un porcentaje de la venta. Al agricultor trabajar con la alhóndiga le da una estabilidad, pues si los productos no son vendidos pueden ser almacenados en la empresa sin coste alguno hasta la subasta del día siguiente, además de ser abastecidos con recursos necesarios como cajas, supervisión por parte de ingenieros agrónomos, etc…

De forma diaria en cada corrida se realizan las subastas excepto domingo y festivos, y posteriormente se pública el listado de precios con todos los distintos cortes de los productos subastados, llamamos corte a cada puja de cada comercial.

Nos hemos centrado en la provincia de Almería porque la autora del trabajo es original de dicha provincia, por lo que se conoce más en profundidad el sector de la zona para tomar decisiones y poder aclarar dudas surgidas durante el trabajo.

Los datos que usaremos son recogidos de la página web *Agroprecios.com.* Este sitio web ofrece los precios de las subastas agrícolas de Almería, Granada y Murcia. Disponibles los datos de las subastas desde el año 2000.

Nuestro trabajo se centrará en las alhóndigas de la provincia de Almería, un total de 19 empresas, en el estudio de 6 hortalizas, todas variedades de pimiento, que son las que más se comercializan en esta provincia, (Pimiento Largo Rojo, Pimiento Largo Verde, Pimiento Corto Amarillo, Pimiento Corto Verde, Pimiento Italiano Verde, Pimiento Corto Rojo).

Los datos recogidos incluyen el precio por corte de cada producto para cada empresa, la información obtenida es desde 2015 en la mayoría de los productos hasta marzo de 2019. Por tanto, tenemos un total de 68587 observaciones.

2. Objetivos

En primer lugar, y antes de empezar con la elaboración del trabajo nos fijaremos unos objetivos fundamentales y la metodología necesaria para su obtención, que la explicaremos en el siguiente capítulo.

En el proyecto nos hemos centrado en dos objetivos principales, por un lado, predecir el precio del primer corte de la subasta y, por otro lado, predecir si el precio del primer corte de la subasta será mayor dentro de una semana u hoy.
Nos hemos fijado dos objetivos principales porque consideramos interesante hacer un análisis desde el punto de vista de la empresa y del agricultor, y así estudiar la predicción para una variable dependiente continua y binaria.

Para las empresas de la zona les sería de gran utilidad saber el precio a futuro de sus productos, pues así pondrán hacer estimaciones de gastos y beneficios. Para los agricultores sería muy interesante estudiar si el precio de una determinada hortaliza es superior dentro de 7 días u hoy para ellos sacarle mayores beneficios a su cultivo.

Al proponer estos objetivos fundamentales, nos surgen objetivos secundarios a cumplir para obtenerlos con éxito. Los más importantes serían:

- Conocer a fondo la base de datos que tenemos, para analizar su estructura y posibles singularidades que presente.
- Detectar valores atípicos que presente la muestra y analizar los valores ausentes.
- Hacer una selección de las variables, así como estudiar la creación de variables para nuestra base de datos.
- Encontrar los parámetros idóneos para los modelos de predicción con las distintas técnicas utilizadas.
- Hacer una comparación entre los distintos modelos con diferentes técnicas, para así encontrar el mejor modelo predictivo.

- Realizar un análisis de los resultados obtenidos con cada técnica para así entender el comportamiento de nuestro modelo.

3. Metodología

Las técnicas estadísticas multivariantes que se utilizarán para cumplir los objetivos descritos serán:

- Por un lado, para predecir el precio del primer corte de la subasta utilizaremos técnicas de Regresión Lineal y Redes Neuronales.
- Para la predicción de saber si el precio del primer corte de la subasta de la semana que viene será mayor que el precio actual utilizaremos modelos con técnicas basadas en Regresión Logística, Redes Neuronales, Bagging, Random Forest, Gradient Boosting y Ensamble de Modelos.

La metodología que llevaremos a cabo para alcanzar los objetivos planteados es SEMMA (muestrear, explorar, modificar, modelizar y evaluar). Nosotros lo seguiremos en este orden, pero no es estrictamente necesario.

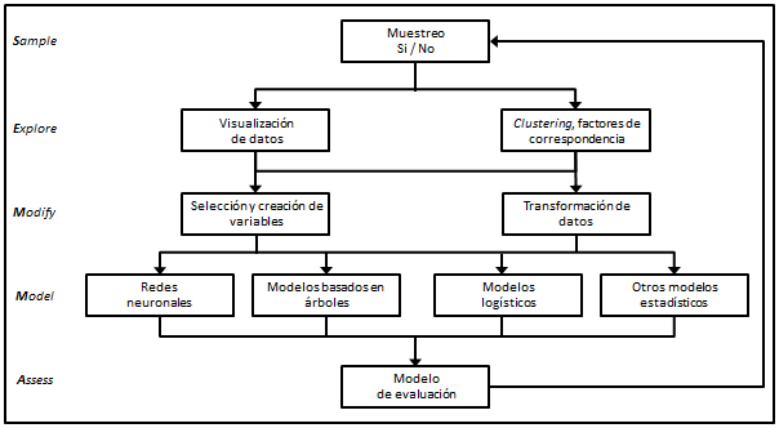


Ilustración 1. Diagrama resumen de la metodología SEMMA.

- Sample, muestrear, en el estudio se ha llevado a cabo técnicas de Web Scraping para conseguir la base de datos que utilizaremos en el trabajo.

  Web Scraping es una técnica utilizada mediante programas de software para extraer información de páginas web, en nuestro caso hemos utilizado Python para llevarlo a cabo.

- Explore, explorar, es necesario hacer una exploración de los datos para detectar anomalías o tendencias, para ello realizaremos un análisis descriptivo de las variables de la muestra. El software utilizado ha sido SAS Guide.

- Modify, modificar, es aconsejable modificar los datos creando, seleccionando y transformando las variables para encontrar el mejor modelo.

  Nuestra base de datos ha sido modificada durante un largo proceso para crear nuevas variables que nos puedan ser de ayuda. El software utilizado ha sido R.

- Model, modelización, en este paso deberemos estudiar modelos que nos permitan predecir la variable objetivo. Como tenemos dos objetivos principales, cada uno tendrá variables objetivo independientes a estudiar.

  Para predecir el primer precio de la subasta explicaremos a continuación cada una de las técnicas estadísticas a utilizar por nuestros modelos:

  - Regresión Lineal, es un modelo matemático que tiene por objetivo predecir una variable y, variable dependiente u objetivo, a partir de un conjunto de variables independientes x_i. El modelo puede ser expresado como:

    $$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon,$$

    donde ϵ es el error cometido, o lo que es lo mismo, la parte de la variable objetivo Y no explicada a partir de las variables independientes y β_0 , $\beta_1, \dots , \beta_n$ miden la influencia que tienen las variables independientes sobre la variable objetivo.

    Para poder llevar a cabo la predicción será necesario conocer el valor de los parámetros, para ello buscaremos el valor que minimice el error cometido por el modelo, que viene dado por $y - \hat{y}$, siendo $\hat{y}$ la variable objetivo predicha.

    En nuestra muestra tenemos variables cualitativas, estas necesitaran un trato diferente porque su inclusión en el modelo no es tan directa como las variables cuantitativas pues habrá que cambiar los valores de categorías a valores numéricos, por lo que construimos variables dummys como categorías menos una tenga la variable, valdrá 1 si la observación correspondiente toma ese valor y, 0, en otro caso.

    Para evaluar los modelos de regresión lineal lo haremos con la medida de ajuste del error cuadrático medio,

    $$ASE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$$

    Siendo $y_i, \hat{y}_i$ los datos originales y predichos respectivamente, $\bar{y}_i$ predicciones corregidas, n el número de observaciones y p número de parámetros.

    Nos centraremos en el error cuadrático medio como medida de ajuste para comparar modelos, pues para nuestro problema es más importante que las errores mayores pesen más que los errores menos significativos.

  - Redes Neuronales, es un modelo computacional inspirado en el comportamiento observado de las redes neuronales biológicas. Consiste en un conjunto de nodos de entrada, las variables independientes del modelo y_i, conectada a los nodos de la capa oculta, son variables artificiales que no existen como tal en nuestros datos, la combinación de estas nos devuelve la predicción de nuestra variable dependiente.

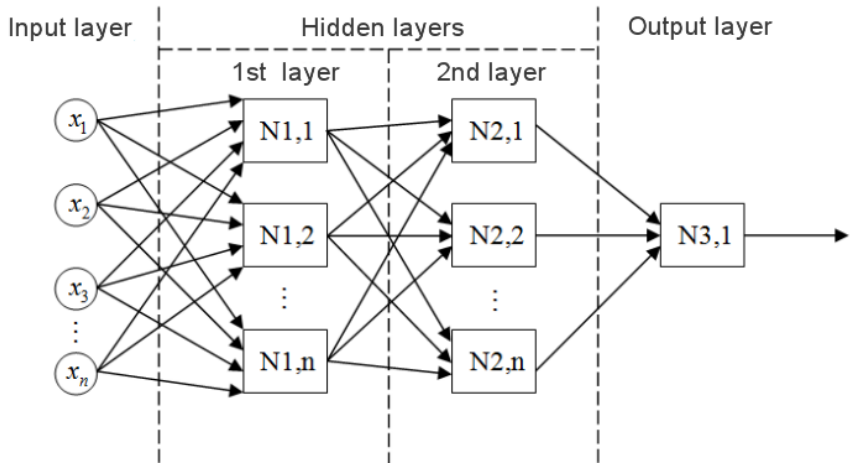


Ilustración 2. Ejemplo de Red Neuronal.

La capa input, donde se encuentran los nodos de entrada, se conectan a los nodos de la capa oculta mediante la función de combinación, donde aparecen los pesos w_{ij} que hacen el papel de parámetros a estimar.

La función de combinación más habitual es la lineal. Tras aplicar la función de combinación, aplicamos a cada nodo oculto la función de activación, representada por f. Finalmente aplicamos combinación y después activación de la capa oculta a la capa output.

Al parámetro constante b_i se denomina bias o sesgo.

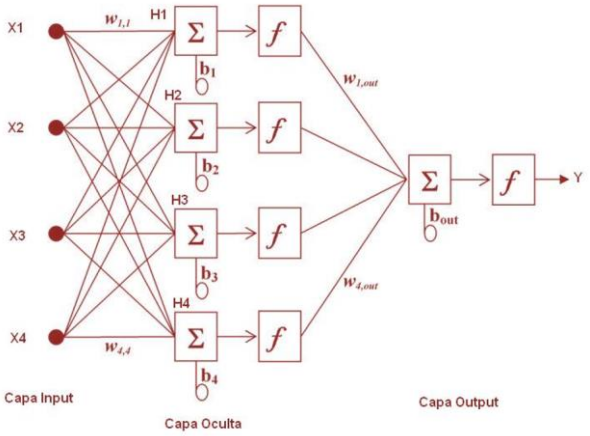


Ilustración 3. Ejemplo de Red Neuronal.

El objetivo computacional de las redes neuronales es la estimación de parámetros w_{ij} y b_i, para ello entrenaremos el modelo de manera iterativa variando entre las diferentes técnicas de optimización numérica, variando los valores de los parámetros hasta encontrar el menor error posible en los datos de validación.

La técnica de redes neuronales está basada en Teoremas de aproximación universal, dichos teoremas enuncian que cualquier función continua puede aproximarse al nivel requerido con una red neuronal con al menos una capa oculta y un número de nodos a determinar. Es decir, existe una

relación entre las variables de entrada y la variable de salida y esta relación no es lineal, desconocida, pudiendo aproximarla con una función construida con la red neuronal. (Portela 2019)

La red neuronal funciona mejor que los modelos de predicción habituales si las relaciones reales entre las variable independiente son no lineales o complejas, aunque las redes neuronales tienden a aparecer sobreajuste, por lo que será necesario muchos datos de entrenamiento y validación para tener como resultado un modelo robusto.

Para la predicción de la variable objetivo binaria nos planteábamos si el precio de la subasta de la semana próxima es superior al día actual utilizaremos las siguientes técnicas estadísticas:

- Regresión Logística, recordemos que los modelos de regresión tienen por objetivo predecir una variable y, variable dependiente u objetivo, a partir de un conjunto de variables independientes x_i. El modelo de regresión logística sigue la relación:

$$p_1 = P\left(Y = 1 \middle| x_{1,} x_2, \dots, x_n\right) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)}}$$

lo que implica que $\log\left(\frac{p_1}{1-p_1}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$ este término recibe el nombre de logit y es el logaritmo del odds ratio.

El concepto de odds ratio viene a definir el cociente entre los odds, cociente entre la probabilidad de que ocurra el suceso y la probabilidad de que no ocurra, de un suceso bajo una determinada condición y el odds de ese mismo suceso bajo otra condición, lo que permitirá evaluar el efecto de dichas condiciones sobre las probabilidades del suceso.

$$OR = \frac{odds(evento|x = 1)}{odds(evento|x = 0)} = \frac{e^{\beta_0 + \beta_1}}{e^{\beta_0}} = e^{\beta_1}$$

Al igual que en la regresión lineal, es necesario crear variables dummys para las variables independientes categóricas.

Los parámetros que definen la regresión logística son desconocidos, lo más habitual es estimarlos por el método de máxima verosimilitud, la idea de este método es estimar los parámetros como aquellos valores que maximicen la probabilidad del conjunto de datos. La función de máxima verosimilitud viene dada por:

$$L(\beta) = \prod_{\underset{y_i}{i}=1} p_{i1} \prod_{\underset{y_j}{j}=0} (1 - p_{1j}),$$

donde p_{i1} y p_{1j} representan las probabilidades del proceso que vienen dadas por:

$$p_{1i} = P\left(Y = 1 \middle| x_{1i,} x_{2i}, \dots, x_{ni}\right) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_n x_{ni})}}$$

Una vez estimados los valores de los parámetros, evaluaremos la significatividad de los mismos, para ello podemos utilizar de manera conjunta el contraste de razón de verosimilitudes, y el contraste de Wald.

- Aplicaremos a los modelos también técnicas de redes neuronales explicadas anteriormente.
- Árboles de decisión, son modelos de predicción de variables de clase, es una herramienta sencilla que representan una segmentación de los datos de forma jerárquica y secuencial.

  Explicaremos en qué consisten los árboles de decisión, pues hemos utilizado tres técnicas que utilizan árboles.

  El árbol parte de un nodo raíz, dividiendo una de las variables independientes escogidas en partes, nodos hijos, dichas partes deben de ser los más heterogéneas posibles entre ellos y homogéneos con la variable dependiente, se elige un punto de corte que minimice el error para establecer las normas de inclusión en un nodo u otro, así de manera recursiva va creciendo el árbol hasta llegar a un nodo terminal.

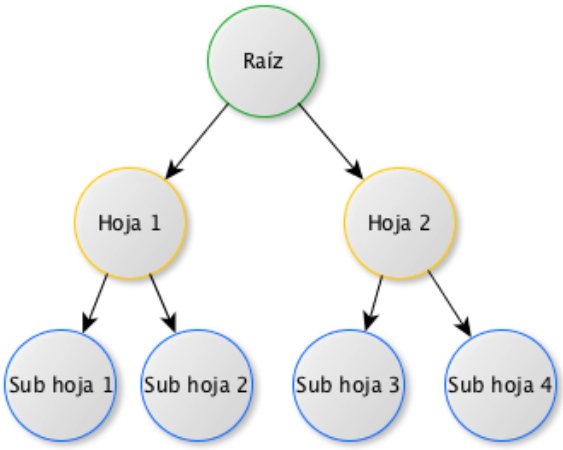


Ilustración 4. Estructura de árbol.

Para establecer un punto de corte y la variable a utilizar por el nodo utilizaremos una serie de métodos de selección:

- Test de la χ^2, consiste en utilizar el p-valor asociado al estadístico χ^2.
- Índice de Gini, evalúa cómo de homogéneo es un nodo.
- Entropía, evalúa cómo de homogéneo es un nodo.

Si el árbol es muy grande, o tiene muchos nodos será necesario simplificarlo, para ello haremos una poda hasta tener el tamaño necesario.

Los árboles tienen grandes ventajas como que tratan de manera automática los valores ausentes, es fácil interpretarlo, en cambio, tiene poca capacidad predictiva y gran varianza.

- Bagging, Bootstrap Averaging, es un algoritmo de combinación de árboles predictores creando un 'bosque', la técnica bagging consiste en:

  Dados los datos de tamaño N,

  1) Repetir m veces i) y ii):

     (i) Seleccionar N observaciones con reemplazamiento de los datos originales.

     (ii) Aplicar un árbol y obtener predicciones para todas las observaciones originales N.

  2) Promediar las m predicciones obtenidas en el apartado 1).

  Bagging funciona bien cuando los modelos tienen muchas variables con relación débil pero estable con la variable dependiente, cuando existen relaciones no lineales, o cuando hay interacciones ocultas o muchas variables categóricas.

  Para mejorar la sensibilidad en vez de procesar solamente una muestra inicial, se generan T submuestras de la muestra inicial de igual tamaño, tomadas aleatoriamente y con reemplazamiento. A estas nuevas muestras se las conoce como muestras bootstrap.

  Los principales parámetros para controlar son:

  - El tamaño de las muestras n y si se va a utilizar Bootstrap.
  - El número de iteraciones m a promediar.
  - Características de los árboles:
    - La profundidad del árbol.
    - El número de divisiones máxima en cada nodo.
    - El p-valor para las divisiones en cada nodo.
    - El número de observaciones mínimo en una rama-nodo.

- Random Forest, es un algoritmo de combinación de árboles predictores creando un 'bosque'. Es una modificación de bagging, que consiste en incorporar aleatoriedad en las variables utilizadas para segmentar cada nodo del árbol. La técnica RF consiste en:

  Dados los datos de tamaño N,

  1) Repetir m veces i), ii), iii):

     (i) Seleccionar N observaciones con reemplazamiento de los datos originales.

     (ii) Aplicar un árbol de la siguiente manera: En cada nodo, seleccionar p variables de las k originales y de las p elegidas, escoger la mejor variable para la partición del nodo.

     (iii)Obtener predicciones para todas las observaciones originales N.

2) Promediar las m predicciones obtenidas en el apartado 1).

El algoritmo Random Forest hace su propia selección de variables evitando decidirse rígidamente por un set de variables como en el caso de bagging.

Los principales parámetros para controlar son:

- El tamaño de las muestras n y si se va a utilizar Bootstrap.
- El número de iteraciones m a promediar.
- El número de variables p a muestrear en cada nodo.
- Características de los árboles:
  - La profundidad del árbol.
  - El número de divisiones máxima en cada nodo.
  - El p-valor para las divisiones en cada nodo.
  - El número de observaciones mínimo en una rama-nodo.

- Gradient Boosting, es una técnica basada en árboles que a medida que repite la construcción de árboles va actualizando las predicciones intentando minimizar los residuos en la dirección de decrecimiento.

  Conforme va creando árboles, va ajustando las predicciones cada vez más a los datos, aprendiendo unos árboles de otros.

  La técnica Gradient Boosting utiliza la función logit como función base, y la Deviance como función de error. El objetivo es ir retocando la función logit para que en cada paso del algoritmo se actualicen las probabilidades predichas y los residuos (Portela 2019):

  Se define $L\big(y_i, f(x_i)\big) = \log(1 + e^{-2y_i f(x_i)})$ donde $y_i = 1, 0$.

  La función $f(x_i)$ se define como $f(x_i) = \frac{1}{2}\log(\frac{\hat{p}_i^{(m)}}{1-\hat{p}_i^{(m)}})$, con $p_i = P(y_i = 1)$.

  Pasos que seguir en el algoritmo:

  1. $\hat{p}_i^{(0)} = \%$ de 1 en los datos,
  2. Calcular el residuo actual $r_i^{(m)} = y_i - \hat{p}_i^{(m)}$ (este residuo es el gradiente, dada la función de error Deviance).
  3. Ajustar mediante un árbol de regresión los residuos $r_i^{(m)}$, variable dependiente, X vector de variables predictoras → $\hat{r}_i^{(m)}$.
  4. Actualizar f_i mediante $f_i^{(m+1)} = f_i^{(m)} + v * \hat{r}_i^{(m)} = \frac{1}{2}\log\left(\frac{\hat{p}_i^{(m)}}{1-\hat{p}_i^{(m)}}\right) + v * \hat{r}_i^{(m)}$. Donde v es un parámetro de regularización encargado de definir la importancia de cada predicción.

5. Actualizar la probabilidad predicha mediante $\hat{p}_i^{(m+1)} = \frac{1}{(1+e^{-2f_i^{(m+1)}})}$.
6. Volver al paso (2).

Los principales parámetros para controlar son:

- La constante de regularización v (shrink). Normalmente entre (0.001 y 0.3). Cuanto más alta, más rápido converge, pero demasiado alta es poco fiable.
- El número de iteraciones m.
- Características de los árboles:
  - ❖ La profundidad del árbol.
  - ❖ El número de divisiones máxima en cada nodo.
  - ❖ El p-valor para las divisiones en cada nodo.
  - ❖ El número de observaciones mínimo en una rama-nodo.

Las principales ventajas de los modelos hechos con técnicas Gradient Boosting son que es invariante frente a transformaciones monótonas, es fácil de implantar, pues tiene pocos parámetros a monitorizar, gran fiabilidad predictiva, en cambio, si los datos son relativamente sencillos la técnica no tiene nada nuevo que aportar y pueden ser preferibles modelos sencillos.

- Ensamblado de modelos, consisten en la construcción de predicciones a partir de la combinación de varios modelos.

  Para la construcción de las predicciones utilizaremos la combinación de los modelos explicados en este apartado.

  Las técnicas básicas de combinado de modelos son:

  - Bagging, random forest es un caso particular.
  - Boosting, gradient boosting es un caso particular.
  - Stacking, existen tres opciones básicas de combinar las predicciones:
    - ❖ Promedio, se calcula el promedio de las predicciones. Si se trata de clasificación, se obtiene el promedio de las probabilidades. Se puede utilizar también promedio ponderado.
    - ❖ Voto, se predice el resultado con mayoría entre las predicciones.
    - ❖ Combinación a partir de otro algoritmo.

  En nuestro caso utilizaremos la opción del promedio.

  Las ventajas del ensamblado podrían ser:

  - Modelos bastantes robustos, pues unos se corrigen a otros.

- Se reduce la varianza del error en general.

Como desventajas:

- Aumenta la complejidad.
- Puede llevar al sobreajuste.
- Los resultados no son interpretables.

- Assess, evaluación, una vez calculados los modelos con las técnicas definidas hay que comprobar la calidad de las predicciones y comparar los modelos obtenidos.

  Si la variable objetivo es continua, las comparaciones las haremos fijándonos en el ASE, error cuadrático medio, el número de variables, entre otras cosas.

  Si la variable objetivo es categórica, las comparaciones las haremos fijándonos en la tasa de error, entre otras cosas.

4. Muestreo de la base de datos

La base de datos ha sido recogida con los datos de la página web *Agroprecios.com.* Esta web se encarga de actualizar diariamente los precios de las subastas agrícolas de Almería, Granada y Murcia.

La muestra utilizada ha sido adquirida por el método de Web Scraping, una técnica que permite extraer información de sitios web. El método se ha realizado mediante un programa en Python, técnica adquirida en el máster.

Nosotros nos centraremos en las empresas de la provincia de Almería, un total de 19 empresas, y en el estudio de 6 hortalizas, variedades de pimiento.
Los datos recogidos incluyen el precio por corte de cada producto para cada empresa, la información obtenida es desde 2015 en la mayoría de los productos hasta marzo de 2019. Por tanto, tenemos un total de 68587 observaciones.

Los precios de nuestros productos se encuentran la web de *agroprecios*. Se necesita una cuenta para poder tener los datos anteriores a 10 días, y se acceden a los datos por días en una tabla. No hay ninguna opción de descargar los datos como tal, por lo que hemos creado un script para poder extraer dichos datos en CSV.

La web cuenta con algunos scripts en javascript que la hacen funcionar. Además, cuenta con un login para autenticarse en el inicio. Por esto hemos decidido utilizar la librería Selenium con el driver de Chrome, para emular un navegador y poder interaccionar con la web.

El objetivo es obtener los datos de las hortalizas en formato CSV de los últimos 4 años. Para hacer esto, tendremos que crear un script que visite los datos de interés y extraiga la información necesitada.
Para ello creamos una clase con las variables que extraeremos para la base de datos, en este caso fecha, producto, empresa, media y un array con todos los precios de los cortes, X1, X2, …, X20.
Además, necesitaremos una función que reciba este objeto, y la escriba en un fichero con el formato CSV.

Con la librería de Selenium podremos interactuar con los elementos visuales, rellenando campos, haciendo clic en ciertos elementos o cualquier cosa que haríamos nosotros con el navegador.
El primer paso es iniciar sesión, ir a la pestaña *precios de los productos,* seleccionar el día, la hortaliza y las alhóndigas de donde queremos la información. Estos nos dará una tabla HTML con los precios de las diferentes cortes, esta es la tabla que contiene la información que queremos descargar, con el método *.text* extraemos la información.
Queda iterar el proceso hasta los 4 años que queremos.

Una vez recogido los datos de la web a través de técnicas de Web Scraping, se necesitó un largo proceso de depuración y corrección de errores, donde se unificó todos los datos recogidos en una única base de datos, se corrigieron erratas como los nombres del producto recogidos de diferente formato, los formatos indicados en los datos recogidos como fechas. Además, el programa que realiza el scraping cada vez que la velocidad de internet tenía una pequeña bajada saltaba al día anterior, por lo que hemos tenido que manualmente completar los datos de los días que no se tenían información.

5. Exploración de las variables

Realizaremos un análisis descriptivo para conocer las principales características de nuestro conjunto de datos. Es importante antes de realizar la depuración de nuestros datos y los modelos de predicción conocer la calidad de nuestras variables para hacer un estudio más exhaustivo que nos permitirá conocer posibles errores como valores fuera de rango o la existencia de valores ausentes y además nos proporcionará una idea de la estructura de los datos.

Nuestro conjunto de datos inicial consta de 68587 observaciones con 24 variables. A continuación, daremos una breve explicación de las variables que forman nuestra base de datos, haremos una distinción entre variables cualitativas y variables cuantitativas.

I. Variables Cualitativas

Las variables cualitativas son variables cuyos valores son un conjunto de categorías que toman un número limitado de posibles valores. Estas variables permiten agrupar los datos según sus características.

Las variables cualitativas se clasifican según:

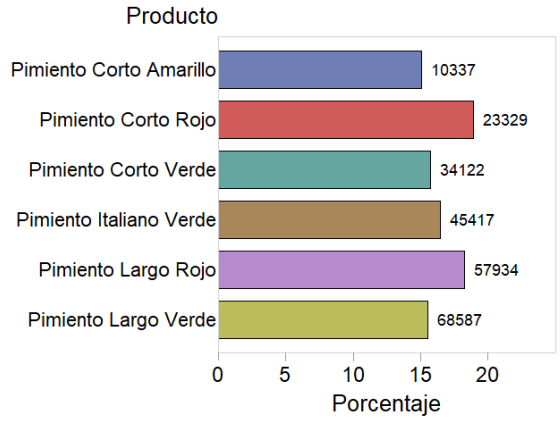
- Dicotómicas: Sólo toman dos posibles valores.
- Politómicas: Cuando hay más de dos categorías. Esta se puede subdividir entre:
    - Nominal, no se puede definir un orden natural entre las categorías.
    - Ordinal, Es posible establecer relaciones entre las categorías.

Las variables cualitativas de la base de datos son:

- Producto, nos describe como su nombre indica el nombre de producto. Es una variable cualitativa nominal.

Producto	Frecuencia	Porcentaje	Frecuencia acumulada	Porcentaje acumulado
Pimiento Corto Amarillo	10337	15.07	10337	15.07
Pimiento Corto Rojo	12992	18.94	23329	34.01
Pimiento Corto Verde	10793	15.74	34122	49.75
Pimiento Italiano Verde	11295	16.47	45417	66.22
Pimiento Largo Rojo	12517	18.25	57934	84.47
Pimiento Largo Verde	10653	15.53	68587	100.00

Tabla 1. Descriptivo de la variable Producto.



Vemos como la hortaliza más comercializada en la subasta es la variedad de color rojo, tanto largo como corto.

Tabla 2. Frecuencias de la variable Producto.

- Empresa, nos dice la empresa en la que se hizo la subasta. Es una variable cualitativa nominal.

Empresa	Frecuencia	Porcentaje	Frecuencia acumulada	Porcentaje acumulado
AGRO SAN ISIDRO	464	0.68	464	0.68
AGROEJ. BERJA	3307	4.82	3771	5.50
AGROEJ. DALIAS	1738	2.53	5509	8.03
AGROEJ. EJIDO	6458	9.42	11967	17.45
AGROPON. - NIJAR	391	0.57	12358	18.02
AGROPONIENTE	7062	10.30	19420	28.31
AGROPONIENTE 2	4185	6.10	23605	34.42
AGRUP. - EL VISO	1574	2.29	25179	36.71
AGRUP. - NIJAR	35	0.05	25214	36.76
AGRUPAADRA	5802	8.46	31016	45.22
AGRUPAEJIDO	7090	10.34	38106	55.56
AGRUPALMERIA	1	0.00	38107	55.56
CEHORPA	6803	9.92	44910	65.48
COSTA ALMERIA	5701	8.31	50611	73.79
FEMAGO	3783	5.52	54394	79.31
LA COSTA	3161	4.61	57555	83.92
LA UNION	7233	10.55	64788	94.46
LA UNION ADRA	2174	3.17	66962	97.63
UNION 4 VIENTOS	1625	2.37	68587	100.00

Tabla 3. Descriptivo de la variable Empresa.

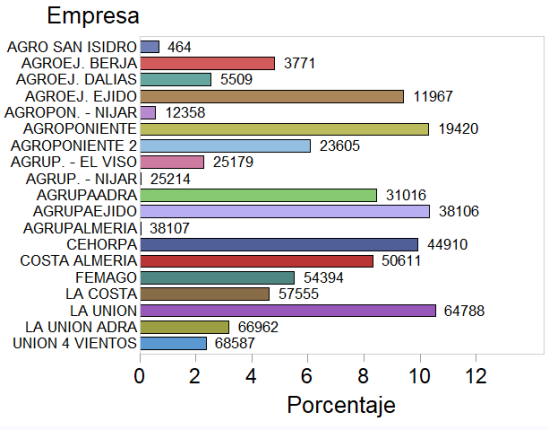


Tabla 4. Frecuencias de la variable Empresa.

Según el gráfico de frecuencias la empresa con mayor cuota de mercado es La Unión, pues es una de las mayores empresa de la zona. Con menor cuota de mercado, tenemos a las empresas Agrupalmería y Agrup. - Nijar pues estas empresas no están especializadas en la subasta de pimiento.

II. Variables Cuantitativas

Las variables cuantitativas son aquellas variables que toman valores numéricos. Estas variables permiten dar un resultado con un valor numérico exacto.

Las variables cuantitativas se clasifican según:

- Discreta: toman un número finito de valores. No puede tomar valores entre dos cifras ya especificadas.
- Continua: toman un número ilimitados de valores.

Las variables cuantitativas de nuestros datos iniciales son:

- Fecha, nos describe la fecha en la que se realizó la subasta. Esta en formato AAAA-MM-DD. Es una variable de tipo Fecha.

Variable de análisis : Fecha				
Media	Dev std	Mediana	Mínimo	Máximo
20834.48	447.2239849	20816.00	20090.00	21630.00

Tabla 5. Descriptivo de la variable Fecha.

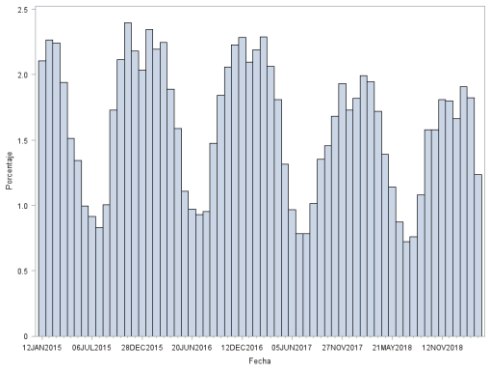


Ilustración 5. Histograma de la variable Fecha.

El análisis descriptivo para las variables de tipo Fecha no nos aportan información, en cambio, con el histograma podemos ver cómo estamos ante una serie temporal estacionaria. Los valores típicos se concentran de Octubre a Mayo.

- X1, precio al que se ha vendido el primer corte, es decir, el comprador que más ha pujado en la subasta. Expresado en céntimos. Variable cuantitativa continua. Esta variable será nuestra variable objetivo.

Variable de análisis : X1				
Media	Dev std	Mediana	Mínimo	Máximo
99.3277589	39.1227097	92.0000000	4.0000000	364.0000000

Tabla 6. Descriptivo de la variable X1.

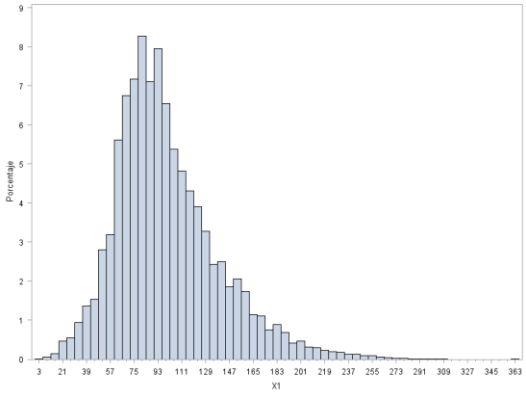


Tabla 7. Histograma de la variable X1.

La variable X1 vemos que su media es de aproximadamente un euro.

Por el histograma podemos ver qué esta variable se comporta como una normal, teniendo más peso la zona derecha que la izquierda.

- X2, precio al que se ha vendido el segundo corte, es decir, el segundo comprador que más ha pujado en la subasta. Expresado en céntimos. Variable cuantitativa continua.

Variable de análisis : X2				
Media	Dev std	Mediana	Mínimo	Máximo
96.9792812	38.1719688	90.0000000	3.0000000	350.0000000

Tabla 8. Descriptivo de la variable X2.

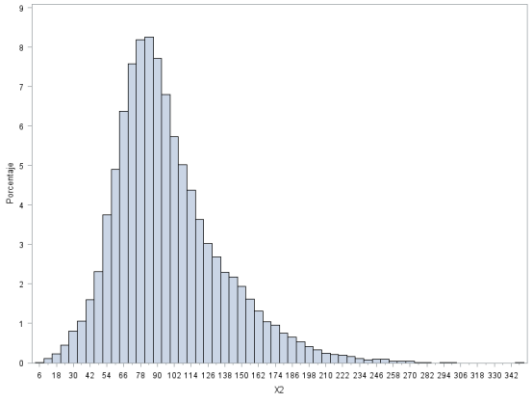


Ilustración 6. Histograma de la variable X2.

Se comporta de manera similar a la variable anterior.

La variable X2 presenta una media, desviación típica, mediana, mínimo y máximo menor que la anterior variable. Es lógico, pues sería el segundo precio por una subasta a la baja.

- $X_j \ \forall j, 3 \leq j \leq 20$, precio al que se ha vendido el j corte. Variable cuantitativa continua.

Variable de análisis : X3				
Media	Dev std	Mediana	Mínimo	Máximo
94.7012945	37.4948200	88.0000000	6.0000000	341.0000000

Tabla 9. Descriptivo de la variable X3.

Variable de análisis : X5				
Media	Dev std	Mediana	Mínimo	Máximo
92.9720196	36.9942018	86.0000000	5.0000000	280.0000000

Tabla 10. Descriptivo de la variable X5.

Variable de análisis : X7				
Media	Dev std	Mediana	Mínimo	Máximo
93.8850312	37.5759772	87.0000000	6.0000000	274.0000000

Tabla 11. Descriptivo de la variable X7.

Variable de análisis : X11				
Media	Dev std	Mediana	Mínimo	Máximo
92.9306845	39.7669248	88.0000000	11.0000000	246.0000000

Tabla 12. Descriptivo de la variable X11.

Variable de análisis : X13				
Media	Dev std	Mediana	Mínimo	Máximo
94.1353945	42.3909714	88.0000000	14.0000000	233.0000000

Tabla 13. Descriptivo de la variable X13.

Variable de análisis : X15				
Media	Dev std	Mediana	Mínimo	Máximo
94.3212435	44.9331728	85.0000000	19.0000000	224.0000000

Tabla 14. Descriptivo de la variable X15.

Variable de análisis : X18				
Media	Dev std	Mediana	Mínimo	Máximo
110.9090909	47.4140370	121.0000000	28.0000000	174.0000000

Tabla 15. Descriptivo de la variable X18.

Variable de análisis : X19				
Media	Dev std	Mediana	Mínimo	Máximo
126.0000000	51.5072810	149.0000000	67.0000000	162.0000000

Tabla 16. Descriptivo de la variable X19.

Variable de análisis : X20				
Media	Dev std	Mediana	Mínimo	Máximo
66.0000000	.	66.0000000	66.0000000	66.0000000

Tabla 17. Descriptivo de la variable X20.

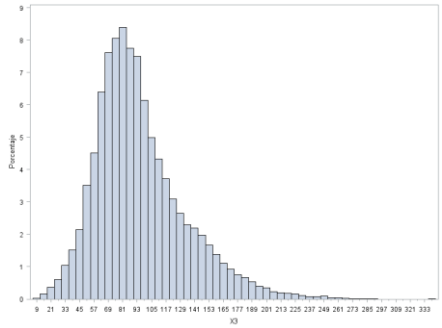


Ilustración 7. Histograma de la variable X3.

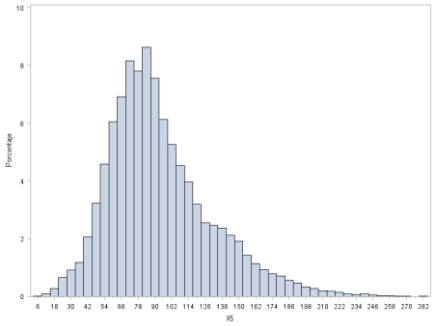


Ilustración 8. Histograma de la variable X5.

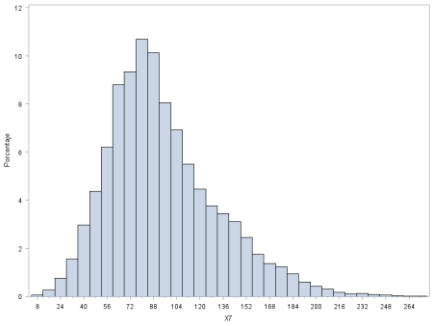


Ilustración 9. Histograma de la variable X7.

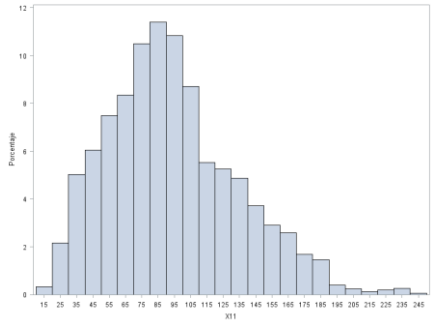


Ilustración 10. Histograma de la variable X11.

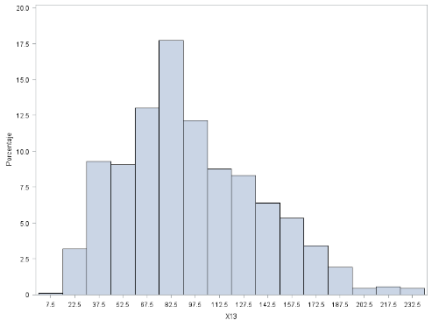


Ilustración 11. Histograma de la variable X13.

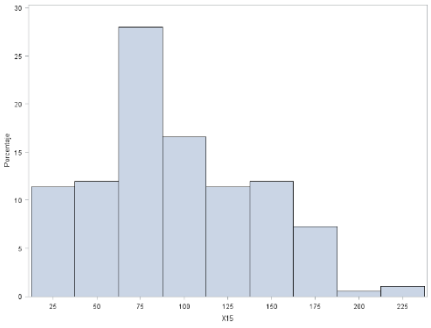


Ilustración 12. Histograma de la variable X15.

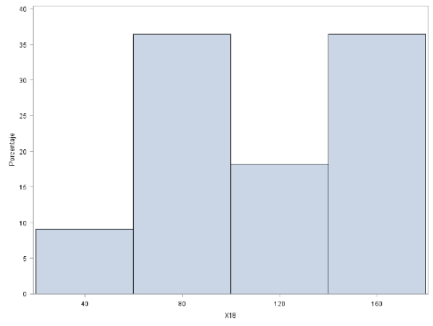


Ilustración 13. Histograma de la variable X18.

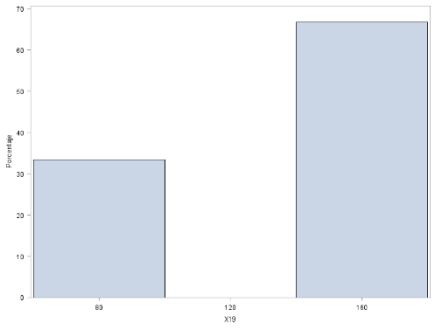


Ilustración 14. Histograma de la variable X19.

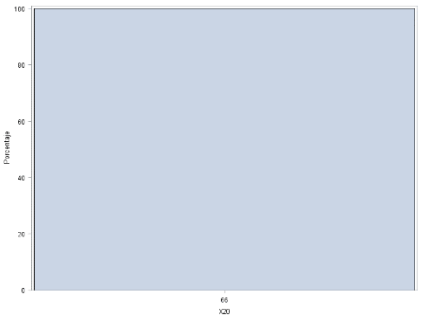


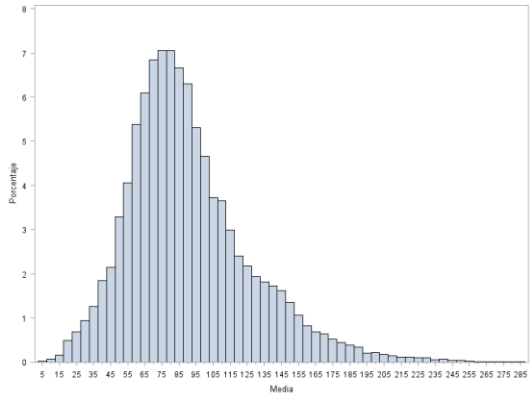
Ilustración 15. Histograma de la variable X20.

Las variables X3, …, X20 vemos como simulan una normal y a medida que se va a acercando a X20 está distribución va desapareciendo por la falta de valores no ausentes.

- Media, esta variables nos indica la media a la que se ha vendido el producto de la empresa, es decir, la media aritmética de X1, X2, … , X20.

Variable de análisis : Media				
Media	**Dev std**	**Mediana**	**Mínimo**	**Máximo**
90.5717439	36.1990877	84.5000000	4.0000000	285.0000000

Tabla 18. Descriptivo de la variable Media.



La variable Media se comporta como una normal, teniendo más peso la zona derecha que la izquierda.

Ilustración 16. Histograma de la variable Media.

6. Modificación de la base de datos

En este capítulo estudiaremos que variables podemos agregar a nuestra base de datos que nos sean de ayuda para crear los modelos de predicción, y analizaremos posibles transformaciones para nuestras variables, como el estudio de los valores faltantes y atípicos.

I. Creación de variables

Además de las variables explicadas, que han sido recogidas inicialmente en la base de datos, hemos creado dos variables categóricas y 26 variables continuas, que nos ayudará para crear una base de datos de mayor calidad para posteriormente ser el propio modelo de predicción el que decida cuales son las variables que más información le aportan.

Las variables cualitativas creadas han sido:

- Fecha_semana, esta variable nos dice que día de semana se realizó la subasta. En calendario americano, es decir, siendo 1-Domingo, 2-Lunes, 3-Martes, 4-Miércoles, 5-Jueves, 6-Viernes, 7-Sábado. Es una variable cualitativa ordinal.

Fecha_semana	Frecuencia	Porcentaje	Frecuencia acumulada	Porcentaje acumulado
2	10997	16.03	10997	16.03
3	11687	17.04	22684	33.07
4	11798	17.20	34482	50.27
5	11546	16.83	46028	67.11
6	11426	16.66	57454	83.77
7	11133	16.23	68587	100.00

Tabla 19. Descriptivo de la variable Fecha_semana.

Vemos que esta variable está muy balanceada en todas sus categorías.

- Precio_semana, esta variables es dicotómica, pues nos dice si el primer precio (X1) para la empresa y producto es mayor que el primer precio de la semana que viene.

Esta variable será nuestra variable objetivo para el segundo objetivo que nos hemos planteado.

Calcularemos variables cuantitativas que nos den información acerca del primer precio de la subasta del día, semana y año anterior, así como el mínimo, máximo y su media.

A la hora de plantearnos estas variables pensamos que el primer precio del día, semana y año anterior sería respectivamente una, seis y 299 subastas anteriores respectivamente, (calculamos que un año aproximadamente tendría 299 días laborables).
Una vez definidas dichas variables nos dimos cuentas de errores que habíamos pasado por alto, ya que, diariamente no siempre se realiza la subasta para todos los productos, así que definimos una segunda opción buscando explícitamente el valor buscado para una fecha concreta.

Con esto hemos creados dos opciones buscando la misma idea, las variables cuantitativas creadas han sido:

- Subasta_dia_anterior_op1, esta variable nos dice el precio del primer corte (X1) del día anterior, sino existiera el anterior disponible para la misma empresa y mismo producto. Variable cuantitativa continua.

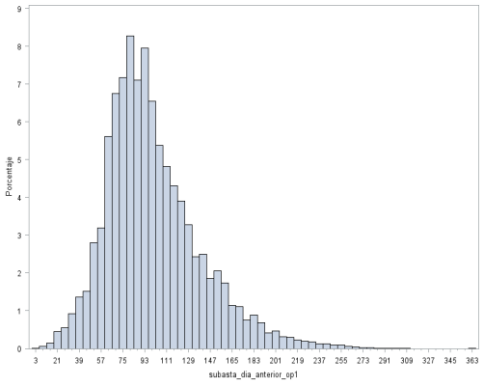


Ilustración 17. Histograma de la variable subasta_dia_anterior_op1.

La variable Subasta_dia_anterior_op1 se comporta como una normal, teniendo más peso la zona derecha que la izquierda.

Tiene una media aproximadamente igual a 1 es muy similar a X1.

- Media_dia_anterior_op1, nos indica la media aritmética del precio del primer corte de la subasta del día anterior para todas las empresas y el mismo producto. Variable cuantitativa continua.

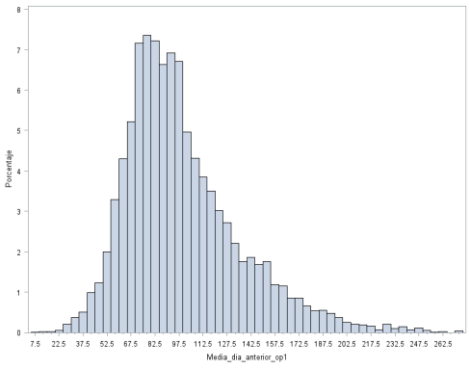


Ilustración 18. Histograma de la variable Media_dia_anterior_op1.

La variable Media_dia_anterior_op1 se comporta como una normal, teniendo más peso la zona central y derecha.
No hay mucha diferencia entre esta variable y la anterior pues la variable Subasta_dia_anterior_op1.

- Min_dia_anterior_op1, nos indica el valor mínimo del precio del primer corte de la subasta del día anterior para todas las empresas y el mismo producto. Variable cuantitativa continua.

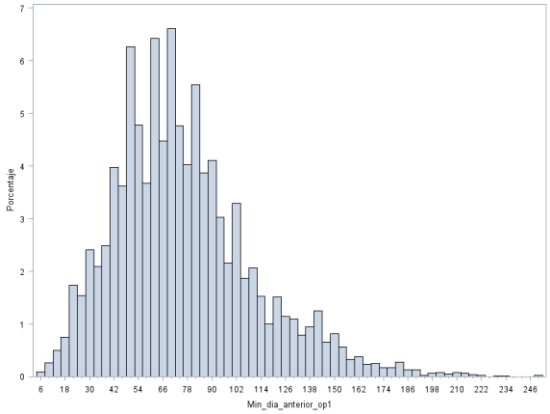


Ilustración 19. Histograma de la variable Min_dia_anterior_op1.

La variable Min_dia_anterior_op1 no podemos afirmar que posee una normal, pues los valores entre el 54 – 90 no se comportan como esta.
Es curioso que el mínimo se encuentre 251 céntimos, pues es un precio muy alto para el pimiento.

- Max_dia_anterior_op1, nos indica el valor máximo del precio del primer corte de la subasta del día anterior para todas las empresas y el mismo producto. Variable cuantitativa continua.

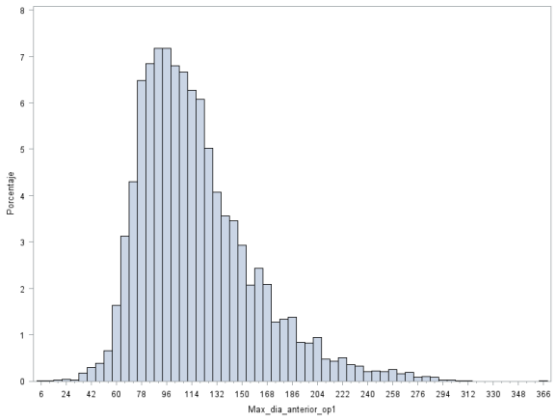


Ilustración 20. Histograma de la variable Max_dia_anterior_op1.

Esta variable sigue una distribución normal con más peso en la zona central.
La variable Max_dia_anterior tiene una media igual a 116.82.

- Subasta_semana_anterior_op1, esta variable nos dice el precio del primer corte de seis subastas anteriores para la misma empresa y mismo producto. Sabemos que los domingos no existe subasta por lo que la semana consta de 6 días. Variable cuantitativa continua.

- Media_semana_anterior_op1, nos indica la media aritmética del precio del primer corte de seis subastas anteriores para todas las empresas y el mismo producto. Variable cuantitativa continua.

- Min_semana_anterior_op1, nos indica el valor mínimo del precio del primer corte de seis subastas anteriores para todas las empresas y el mismo producto. Variable cuantitativa continua.

- Max_semana_anterior_op1, nos indica el valor máximo del precio del primer corte de seis subastas anteriores para todas las empresas y el mismo producto. Variable cuantitativa continua.

- Subasta_año_anterior_op1, esta variable nos dice el precio del primer corte de doscientos noventa y nueve subastas anteriores para la misma empresa y mismo producto. Suponiendo que no hay subastas los domingos y que al año hay 14 festivos y 52 domingos, un año consta de 299 días. Variable cuantitativa continua.

- Media_año_anterior_op1, nos indica la media aritmética del precio del primer corte de doscientos noventa y nueve subastas anteriores para todas las empresas y el mismo producto. Variable cuantitativa continua.

- Min_año_anterior_op1, nos indica el valor mínimo del precio del primer corte de doscientos noventa y nueve subastas anteriores para todas las empresas y el mismo producto. Variable cuantitativa continua.

- Max_año_anterior_op1, nos indica el valor máximo del precio del primer corte de doscientos noventa y nueve subastas anteriores para todas las empresas y el mismo producto. Variable cuantitativa continua.

- Subasta_dia_anterior_op2, esta variable nos dice el precio del primer corte (X1) del día anterior, sino existiera el anterior disponible para la misma empresa y mismo producto. Variable cuantitativa continua.

- Media_dia_anterior_op2, nos indica la media aritmética el precio del primer corte de la subasta del día anterior para todas las empresas y el mismo producto. Variable cuantitativa continua.

- Min_dia_anterior_op2, nos indica el valor mínimo del precio del primer corte de la subasta del día anterior para todas las empresas y el mismo producto. Variable cuantitativa continua.

- Max_dia_anterior_op2, nos indica el valor máximo del precio del primer corte de la subasta del día anterior para todas las empresas y el mismo producto. Variable cuantitativa continua.

- Subasta_semana_anterior_op2, esta variable nos dice el precio del primer corte de la subasta de la semana anterior, sino existiera, la anterior disponible para la misma empresa y mismo producto. Variable cuantitativa continua.

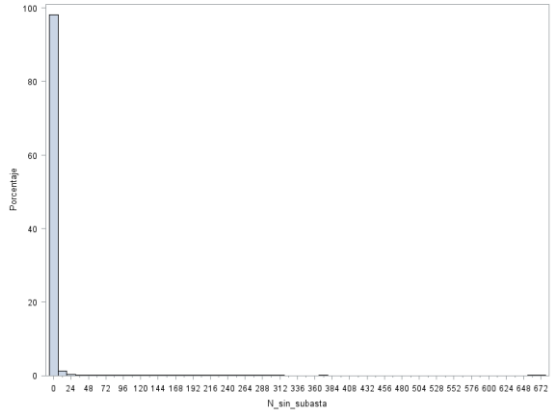
- Media_semana_anterior_op2, nos indica la media aritmética del precio del primer corte de la subasta de la semana anterior para todas las empresas y el mismo producto. Variable cuantitativa continua.

- Min_semana_anterior_op2, nos indica el valor mínimo del precio del primer corte de la subasta de la semana anterior para todas las empresas y el mismo producto. Variable cuantitativa continua.

- Max_semana_anterior_op2, nos indica el valor máximo del precio del primer corte de la subasta de la semana anterior para todas las empresas y el mismo producto. Variable cuantitativa continua.

- Subasta_año_anterior_op2, esta variable nos dice el precio del primer corte de la subasta del año anterior, sino existiera, la anterior disponible para la misma empresa y mismo producto. Variable cuantitativa continua.

- Media_año_anterior_op2, nos indica la media aritmética del precio del primer corte de la subasta del año anterior para todas las empresas y el mismo producto. Variable cuantitativa continua.
- Min_año_anterior_op2, nos indica el valor mínimo del precio del primer corte de la subasta del año anterior para todas las empresas y el mismo producto. Variable cuantitativa continua.
- Max_año_anterior_op2, nos indica el valor máximo del precio del primer corte de la subasta del año anterior para todas las empresas y el mismo producto. Variable cuantitativa continua.
- N_sin_subasta, indica los días sin subasta anterior a ella para la fecha, producto y empresa, es decir, los lunes como mínimo N_sin_subasta es igual a 1, ya que, los domingos no hay subasta. Variable cuantitativa discreta.



Con esta variable vemos como la moda es cero, lo cual quiere decir que lo más frecuente es que se realicen subastas diariamente.

Vemos como hay muchos datos atípicos llegando a alcanzar 673 días sin subasta.

Ilustración 21. Histograma de la variable N_sin_subasta.

- Mes, indica el mes del año de la el cual se realizó la subasta.

Como hemos explicado, hemos hecho dos tipos de opciones para hallar el precio del primer corte de subastas anteriores. La opción 1 es la más sencilla de realizar a la hora de programar, ya que, para cada producto lo hemos ordenado por empresa y posteriormente por fecha, y hemos calculado los datos que queríamos a través de los índices. Para hallar el precio del primer corte de la subasta i, será el precio del primer corte de la subasta $i-1$, si ambas tienen la misma empresa. De la misma forma lo hemos hallado para la semana y año anterior. En esta opción no se tienen en cuenta debilidades como, por ejemplo, si una empresa ha realizado subasta de una hortaliza todos los días de la semana y a la semana siguiente sólo se ha producido subasta un miércoles, el valor de la variable Subasta_semana_anterior_op1 debería de corresponder al miércoles de la semana anterior y con esta opción lo asocia al lunes, ya que sería 6 posiciones menos. Este error va arrastrando a la hora de calcular la variable, por lo que cada vez tiene menos veracidad. Por eso, decidimos realizar una segunda opción que describiera lo que en un principio buscábamos de estas variables, que tienen estas debilidades.

Hemos dejado las dos opciones en la base de datos para que sean los propios modelos los que decidan si son de utilidad.

La creación de las variables explicadas y la corrección de errores como cambios de formato (con las variables formato fecha y la variable Fecha_semana de continua a categórica), corrección de erratas a la hora de venir definida la variable Producto, entre otras cosas, se ha realizado a través de Studio R y los gráficos del estudio estadístico con SAS Guide. En el anexo se adjunta el código utilizado.

II. Tratamiento de valores ausentes y atípicos

Una vez que ya tenemos pensado las posibles variables de entrada en nuestra base de datos, tenemos que analizar los valores ausentes.

Consideraremos que no serán observaciones de entrada en nuestro modelo aquellas filas que contengan más de un 30% de variables sin informar, es decir, más de 14 variables, puede ser que es un porcentaje no muy restrictivo, pero en nuestra base de datos la mayoría de las observaciones vienen sin informar las variables X9, X10, …, X20.

Hemos creado una variable que hemos llamado numMissing, esta variable nos dice cuál es el número de variables ausentes por observación.

Haciendo un análisis descriptivo simple de la base de datos con la condición descrita, nos queda:

Estadísticos descriptivos de la variable de intervalo

Variable	Etiqueta	Ausente	N	Mínimo	Máximo	Media	Desviación estándar	Asimetría	Curtosis
Fecha		0	15329	20091.0	21630.00	20924.77	420.449	-0.10706	-0.96223
Max_a_o_anterior_op1	Max_año_anterior_op1	357	14972	13.0	309.00	123.46	44.658	0.99862	1.19172
Max_a_o_anterior_op2	Max_año_anterior_op2	408	14921	15.0	309.00	125.08	45.321	0.99698	1.08749
Max_dia_anterior_op1		0	15329	5.0	364.00	119.89	42.074	0.98137	1.13306
Max_dia_anterior_op2		0	15329	13.0	364.00	119.93	42.111	0.98100	1.11487
Max_semana_anterior_op1		0	15329	11.0	364.00	121.15	42.987	1.06450	1.57699
Max_semana_anterior_op2		4	15325	5.0	289.00	121.12	42.863	1.00144	1.19127
Media		0	15329	17.3	275.30	99.88	36.787	0.86836	0.99277
Media_a_o_anterior_op1	Media_año_anterior_op1	357	14972	13.0	273.29	105.61	38.880	0.88608	0.93221
Media_a_o_anterior_op2	Media_año_anterior_op2	408	14921	15.0	273.29	106.70	39.617	0.89148	0.85762
Media_dia_anterior_op1		0	15329	5.0	254.00	103.30	37.337	0.84772	0.88739
Media_dia_anterior_op2		0	15329	13.0	254.00	103.13	37.376	0.84879	0.88207
Media_semana_anterior_op1		0	15329	11.0	254.00	104.41	37.665	0.87061	0.97572
Media_semana_anterior_op2		4	15325	5.0	254.00	104.22	37.832	0.86270	0.94055
Min_a_o_anterior_op1	Min_año_anterior_op1	357	14972	6.0	233.00	80.18	35.520	0.76088	0.73085
Min_a_o_anterior_op2	Min_año_anterior_op2	408	14921	4.0	251.00	80.65	36.977	0.75207	0.68584
Min_dia_anterior_op1		0	15329	5.0	220.00	79.51	35.155	0.65254	0.57996
Min_dia_anterior_op2		0	15329	4.0	220.00	79.03	35.233	0.65527	0.57131
Min_semana_anterior_op1		0	15329	4.0	220.00	80.67	35.060	0.63943	0.59338
Min_semana_anterior_op2		4	15325	4.0	220.00	80.04	35.224	0.63191	0.57033
N_sin_subasta		0	15329	0.0	3.00	0.17	0.395	2.07678	3.57372
X1		0	15329	25.0	305.00	115.43	39.271	0.92470	1.10056
X10		10458	4871	13.0	248.00	93.04	38.614	0.62845	0.25430
X11		12364	2965	11.0	246.00	92.41	39.276	0.60123	0.18125
X12		13546	1783	16.0	241.00	92.38	40.293	0.57068	0.07063
X13		14479	850	14.0	233.00	93.63	41.534	0.56234	0.02824
X14		14929	400	19.0	229.00	92.80	42.012	0.64137	0.10764
X15		15145	184	19.0	224.00	93.91	44.446	0.51582	-0.31506
X16		15264	65	25.0	221.00	93.35	45.963	0.65587	-0.26554
X17		15307	22	29.0	176.00	96.77	44.243	0.22229	-1.07628
X18		15319	10	28.0	174.00	105.60	46.405	-0.12289	-1.02783
X19		15327	2	67.0	149.00	108.00	57.983	.	.
X2		0	15329	22.0	296.00	109.67	38.478	0.93475	1.13464
X20		15328	1	66.0	66.00	66.00	.	.	.
X3		0	15329	18.0	286.00	105.12	37.977	0.92717	1.12233
X4		0	15329	15.0	277.00	101.29	37.649	0.90689	1.07938
X5		0	15329	12.0	272.00	97.85	37.400	0.87557	1.00467
X6		0	15329	8.0	269.00	94.60	37.284	0.83966	0.94261
X7		2941	12388	6.0	265.00	94.87	37.734	0.78296	0.70092
X8		5690	9639	11.0	253.00	94.75	38.058	0.71886	0.48886
X9		7964	7365	11.0	250.00	93.93	38.272	0.66628	0.35407
a_o_anterior	año_anterior	1908	13421	20090.0	21265.00	20657.07	351.862	0.07271	-1.16112
dia_anterior		0	15329	20090.0	21629.00	20923.60	420.448	-0.10707	-0.96224
numMissing		0	15329	1.0	14.00	11.92	1.950	-0.97222	0.63016
semana_anterior		11	15318	20090.0	21623.00	20918.34	420.008	-0.10614	-0.96263
subasta_a_o_anterior_op1	subasta_año_anterior_op1	2421	12908	5.0	297.00	110.58	39.900	0.94539	1.15331
subasta_a_o_anterior_op2	subasta_año_anterior_op2	1908	13421	16.0	302.00	115.62	41.551	0.95974	0.93733
subasta_dia_anterior_op1		0	15329	19.0	302.00	115.51	39.237	0.92149	1.13978
subasta_dia_anterior_op2		0	15329	19.0	302.00	115.51	39.237	0.92149	1.13978
subasta_semana_anterior_op1		18	15311	23.0	364.00	114.65	39.492	0.94971	1.23990
subasta_semana_anterior_op2		11	15318	24.0	294.00	114.81	39.436	0.94082	1.16273

Estadísticos de sumarización de la variable de clase

Variable	Etiqueta	Tipo	Número de niveles	Ausente
Empresa		C	12	0
Fecha_semana		C	6	0
Producto		C	6	0
precio_semana		N	2	98

Ilustración 22. Estudio descriptivo de las variables.

Esta parte la hemos realizado con SAS Miner, veremos cuantas observaciones ausentes hay por variable. Consideraremos que una variable que contenga un porcentaje superior al 30% de valores ausentes no será candidata para nuestra base de datos.

Por lo que eliminaremos las variables, X7, X8, X9, X10, X11, X12, X13, X14, X15, X16, X17, X18, X19, X20 y subasta_a_o_anterior_op1.

Las variables X7, …, X20 normalmente vienen sin informar porque el producto ha tenido menos de 7 comerciales dispuesto a pujar por él, es una situación común.

Ahora tendremos que pensar la mejor opción para realizar la imputación de los datos, pues en nuestro modelo no dejaremos valores ausentes, lo realizaremos con R Studio. Para cada variable lo haremos de manera distinta, según el siguiente criterio:

- Precio_semana, para esta variable los datos ausentes serán observaciones que elimaneros, pues es una variable objetivo.
- Max_año_anterior_op1, Max_año_anterior_op2, Max_semana_anterior_op2, Media_año_anterior_op1, Media_año_anterior_op2, Media_semana_anterior_op2, Min_año_anterior_op1, Min_año_anterior_op2, Min_semana_anterior_op2, año_anterior, semana_anterior, subasta_año_anterior_op2, subasta_semana_anterior_op1, subasta_semana_anterior_op2, con el valor del día anterior disponible para el producto y empresa, sino estuviera disponible eliminamos dicha observación.

Una vez tratados los datos ausentes nos quedan un total de 12936 observaciones.

Por otro lado, consideramos no tratar los valores atípicos de nuestra base de datos, pues sabemos que los datos están bien recogidos y estos valores nos pondrán ser de gran ayuda a la hora de modelizar, ya que, entrenaremos a nuestro modelo para posibles escenarios que no sean muy comunes.

III. Selección de variables

La selección de variables la realizaremos con los modelos de regresión en el siguiente capítulo, aunque haremos una preselección, ya que, en nuestros modelos no incluiremos variables de tipo fecha, pues consideramos que no son variables predictoras, ya que, en sí mismas no aportan información. Hemos añadido como hemos explicado antes la variable Mes, en ella nos dirá el mes de la subasta, nos parece interesante esta variable, ya que, se comporta de manera similar en el mismo mes a lo largo de los años.

Hemos considerado no realizar interacciones, ya que, nuestra muestra tiene muchas variables e íbamos a tener problemas de memoria computacional.

7. Modelización para variable objetivo continua

Como hemos explicado al inicio del trabajo, tenemos dos objetivos claros en el proyecto, por un lado, predecir el precio del primer precio de la subasta (X1), para ello, haremos modelos de regresión lineal y a partir de éstos cogeremos los mejores modelos para hacer la selección de variables para las redes neuronales.

Hemos considerado no realizar modelos de series temporales pues creemos que son modelos con peor poder predicción que la regresión o la red neuronal, y hemos preferido focalizar los recursos en éstos.

Las funciones utilizadas en este paso son proporcionadas por el profesor Javier Portela, en la asignatura de Técnicas de Machine Learning, importada en el máster.

No todas las variables definidas de nuestra base de datos serán variables input para nuestros modelos, pues hay variables que nos aportan información a futuro y no podrán ser utilizadas, se tratan de las variables Media, X1, …, X20.

I. Regresión Lineal

Una vez que nuestros datos están depurados aplicaremos los modelos de regresión lineal, con una partición 80% para entrenamiento y 20% para prueba, lo hacemos para semillas diferentes para tener unos datos más fiables y cambiando los datos de entrenamiento para cada semilla. Esta regresión nos servirá para la selección de variables para la red neuronal.

Aplicaremos los modelos de regresión lineal con los siguientes criterios y selección de variables:

i. Criterio de selección de variables AIC, selecciona el modelo con el menor Criterio de Información de Akaike, con selección de variables hacia atrás o backward, (empieza con el modelo con todas las variables y va eliminando las variables hasta encontrar el modelo con todas las variables significativas), hacia delante o forward, (empieza con el modelo sin ninguna variable y va añadiendo variables hasta no dejar ninguna variable fuera del modelo que nos aporte información), o paso a paso o Stepwise, (es similar al anterior pero tiene la característica de que puede eliminar variables que ya han entrado en el modelo).

ii. Criterio de selección de variables BIC, selecciona el modelo con el menor Criterio Bayesiano de Schwarz, con selección de variables hacia atrás o backward, hacia delante o forward o, paso a paso o Stepwise.

iii. Criterio de selección de variables SBC, selecciona el modelo con el menor Criterio de Schwarz, con selección de variables hacia atrás o backward, hacia delante o forward o, paso a paso o Stepwise.

La función utilizada nos devuelve una tabla de frecuencias de los modelos que aparecen seleccionados por la regresión en los diferentes archivos train para cada semilla. Estos modelos serán los candidatos para probar con la validación cruzada.

Para la selección de variables AIC con Stepwise y AIC con Forward, nos ha quedado que los dos modelos que más se han repetido con las diferentes semillas son:

Efecto	Count	Percent	Modelo
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, Producto, Fecha_semana, mes.	11	982.143	1
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, Producto, Empresa, Fecha_semana, mes.	8	714.286	2

Tabla 20. Efectos de la Regresión Lineal con selección de variables AIC, Stepwise y Forward.

Para la selección de variables BIC, Stepwise y BIC con Forward, los tres modelos que más se han repetido para las distintas semillas:

Efecto	Count	Percent	Modelo
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, Producto, Fecha_semana, mes.	11	982.143	3
subasta_dia_anterior_op1, subasta_semana_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, Producto, Fecha_semana, mes.	8	714.286	4
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, Producto, Empresa, Fecha_semana, mes.	8	714.286	5

Tabla 21. Efectos de la Regresión Lineal con selección de variables BIC, Stepwise y Forward.

Para la selección de variables SBC, Backward, Stepwise y Forward, nos han quedado:

Efecto	Count	Percent	Modelo
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, Producto, Fecha_semana, mes.	52	464.286	6
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, Producto, Fecha_semana.	31	276.786	7
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, Fecha_semana.	28	250.000	8

Tabla 22. Efectos de la Regresión Lineal con selección de variables SBC, Backward, Stepwise y Forward.

Para la selección de variables AIC y BIC, Backward, nos han quedado:

Efecto	Count	Percent	Modelo
subasta_dia_anterior_op1, subasta_semana_anterior_op2, subasta_año_anterior_op2, N_sin_subasta, Media_año_anterior_op1, Max_año_anterior_op1, Producto, Empresa, Fecha_semana, mes.	3	267.857	9

Tabla 23. Efectos de la Regresión Lineal con selección de variables AIC y BIC, Backward.

Los modelos 1, 3 y 6 son iguales, 2 y 5 también son iguales.

Una vez que tenemos los modelos de regresión lineal aplicaremos validación cruzada a los modelos de regresión lineal:

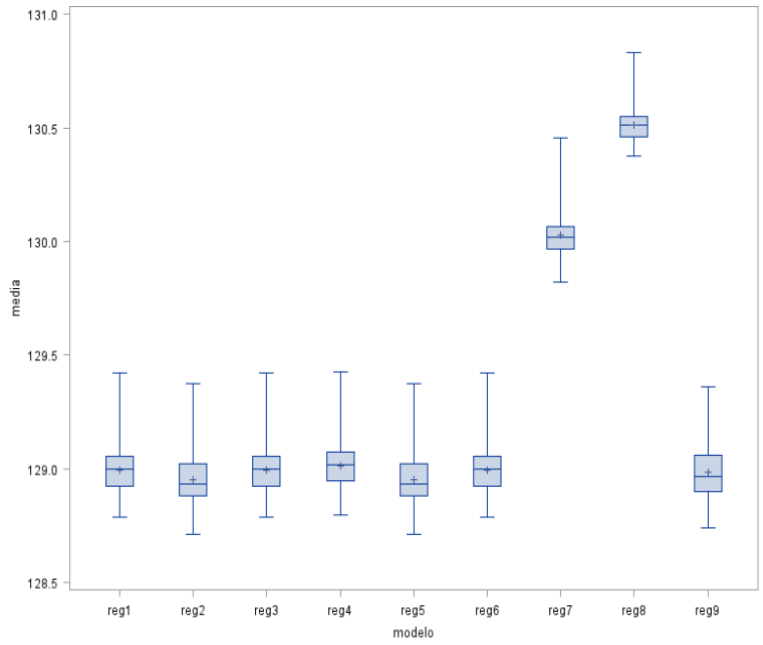


Ilustración 23. Diagrama de cajas y bigotes de los modelos de Regresión Lineal.

Vemos como todas las redes creadas están muy ajustadas por la media del error cuadrático medio. A pesar de ello, las regresiones que menos ecm tienen serían los modelos 2 y 5, que son iguales.

Utilizaremos el modelo finalista como selección de variables para encontrar a la mejor red neuronal en el siguiente capítulo.

i. Evaluación del mejor modelo

Tras haber probado con las diferentes combinaciones de parámetros y para las diferentes semillas hemos calculado más de 1900 modelos, llegando a la conclusión que el mejor modelo contiene 3 variables continuas, subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta y 4 variables categóricas Producto, empresa, fecha_semana y mes.

El modelo hallado con la técnica de regresión lineal presenta una media de error entre de 128.960.

La ecuación matemática que define el modelo quedaría como:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon$$

Los parámetros β_i y ϵ han sido estimados durante el proceso del capítulo. Tomando los valores de los parámetros obtenidos nos quedaría la función:

$$\begin{aligned} precio = {} & 7.946731 + 0.892460 * \text{sub\_dia\_op1} + 0.052998 * \text{sub\_sem\_op2} + 3.918776 \\ & * \text{N\_sin \_subasta} + 1.949185 * \text{Producto Pimiento Corto Amarillo} \\ & + 1.558361 * \text{Producto Pimiento Corto Rojo} - 0.244733 \\ & * \text{Producto Pimiento Corto Verde} + 0.546628 \\ & * \text{Producto Pimiento Italiano Verde} + 2.590865 \\ & * \text{Producto Pimiento Largo Rojo} - 10.197511 * \text{Empresa AGROEJ. BERJA} \\ & - 2.025338 * \text{Empresa AGROEJ. EJIDO} - 2.084684 \\ & * \text{Empresa AGROPONIENTE} + 0.321915 * \text{Empresa AGROPONIENTE 2} \\ & - 3.644764 * \text{Empresa AGRUPAADRA} - 2.914307 \\ & * \text{Empresa AGRUPAEJIDO} - 3.285341 * \text{Empresa CEHORPA} - 2.842747 \\ & * \text{Empresa COSTA ALMERIA} - 1.955258 * \text{Empresa FEMAGO} + 2.281324 \\ & * \text{Empresa LA COSTA} - 1.991968 * \text{Empresa LA UNION} - 4.860778 \\ & * \text{Fecha \_ semana 2} + 2.174478 * \text{Fecha \_ semana 3} - 4.273696 \\ & * \text{Fecha \_ semana 4} - 2.796055 * \text{Fecha \_ semana 5} + 0.674780 \\ & * \text{Fecha \_ semana 6} + 1.760374 * \text{mes 1} + 1.162465 * \text{mes 2} + 0.3171060 \\ & * \text{mes 3} - 2.075838 * \text{mes 4} - 0.153434 * \text{mes 5} + 1.054149 * \text{mes 6} \\ & + 0.4929940 * \text{mes 7} + 1.693385 * \text{mes 8} - 0.065955 * \text{mes 9} - 1.776583 \\ & * \text{mes 10} + 0.223829 * \text{mes 11} \end{aligned}$$

Podemos ver como el mes de noviembre tiene un 22,38% menos de posibilidades de que el precio aumente que en el mes de diciembre. Diciembre es la categoría que coge como referencia, como ya explicamos creamos tantas dummys como categorías menos una tiene la variable.

Para la variables continuas, podemos ver que si aumentamos en una unidad la variable que nos cuenta el primer precio de la subasta del día anterior nuestra probabilidad de que el precio aumente se incrementa en un 89,24%.

Conforme hay más diferencia de días entre subastas vemos como el precio crece significativamente, también observamos que variable que nos dice el precio de la subasta del día anterior aporta más al modelo que la de la semana anterior. El precio suele bajar si la subasta se realiza en lunes o miércoles, manteniendo constantes el resto de las variables.

La empresa La Costa es la que suele vender a más alto precio. Además, durante el mes de abril, mayo, septiembre y octubre el precio del pimiento decrece, pues son los meses que más kilogramos de producto nos encontramos en el mercado.

II. Redes Neuronales

En este capítulo calcularemos modelos de redes neuronales para predecir la variable objetivo X1.

Empezaremos usando el modelo con las variables halladas en regresión lineal, es decir, el modelo 2.

De las primeras cosas a decidir cuándo realizamos modelos de redes neuronales será el número de nodos en la capa oculta a elegir para nuestro modelo. Para ello, a través de la bondad de ajuste de la red, ASE, variaremos el número de nodos de la red desde 1 hasta 10, ya que la red debería de ser en forma de embudo, ya que sino corremos el peligro que tener sobreajuste en nuestro modelo.

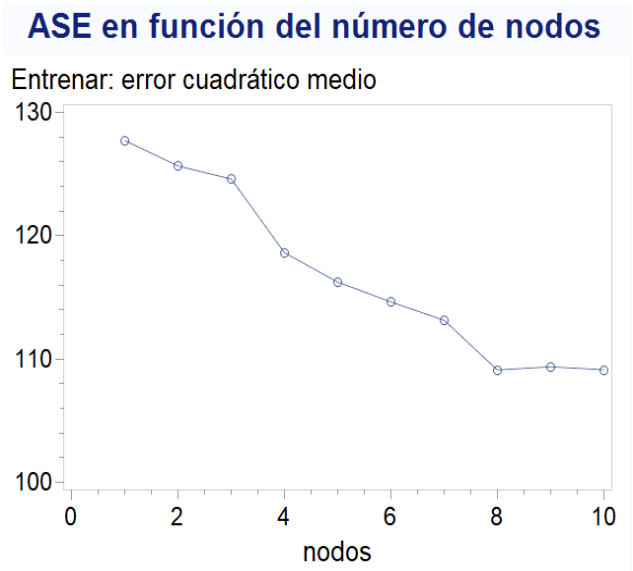


Ilustración 24. ASE en función del número de nodos.

Con este gráfico vemos como error baja conforme aumentamos el número de nodos, esto se debe a que las redes con mayor número de nodos se adaptan mejor a los datos, aunque producirían sobreajuste.

Vemos interesante calcular las redes con 4, 6 y 8 nodos, pues es cuando menor error teniendo en cuenta el número de nodos. A pesar de ello, luego calcularemos a través de validación cruzada cual es el número ideal de nodos para nuestra red.

i. Estudio de los parámetros para la función de activación

Decidiremos a través de validación cruzada la mejor función de activación para nuestra red, según el error cuadrático medio. Lo veremos a través de un diagrama de cajas y bigotes, donde en el eje Y nos muestra la media del ecm entre los resultados obtenidos en la validación cruzada y en el eje de abscisas las distintas funciones de activación a probar. La función de activación nos define una salida del nodo dado un conjunto de valores de entradas, se buscan funciones que su derivada sean simples. Probaremos entre las 7 funciones clásicas:

- Tangente hiperbólica, transforma los valores introducidos en un rango entre -1 y 1. Viene dada por $th(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.
- Logarítmica, sigmoidal, transforma los valores entre 0 y 1. Su función es $f(x) = \frac{1}{1+e^{-x}}$.
- Gaussiana, transforma los valores entre 0 y 1. Su función es $f(x) = e^{-x^2}$.
- Softmax, función exponencial normalizada, transforma la salida en forma de probabilidades, por lo que está acotada entre 0 y 1. Su función es $f(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$.
- Lineal, identidad, permite ver el valor de entrada neto, su función queda $f(x) = x$.
- Seno, transforma los valores entre -1 y 1. Su función $f(x) = sen(x)$.
- Arco tangente, transforma los valores de entrada entre $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Su función nos queda $f(x) = arc\ tg(x)$.

Hemos utilizado para ellas el método de optimización Levmar, aunque posteriormente estudiaremos si es el adecuado. Hemos probamos con varios método para dar unos resultados "fiables".

Para la red de 4 nodos, observamos que las funciones tanh, log, arc tg, lineal, sin, sof están muy ajustadas según el ecm, por lo que todas ellas serán probadas en los modelos finales, aunque para la búsqueda de los siguientes parámetros nos decantamos por utilizar la función tangente hiperbólica, pues es la que mejor resultados queda respecto al sesgo-varianza.

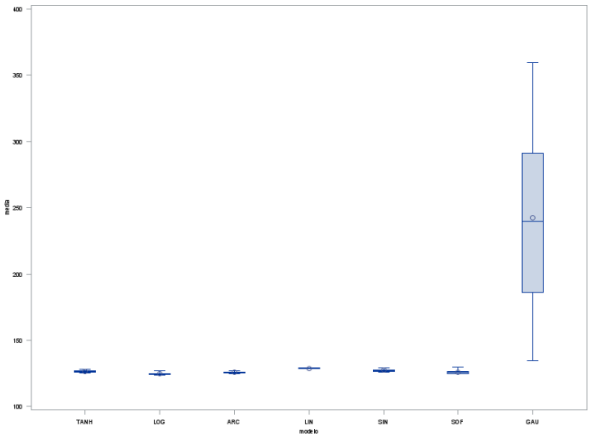


Ilustración 25. Diagrama de cajas y bigotes para decidir la función de activación.

Para la red de 6 nodos nos queda como posibles funciones de activación a utilizar todas las probadas excepto la gaussiana, y seno que nos da peores resultados. Para la red de 6 nodos utilizaremos la función arco tangente, por tener menor ecm.

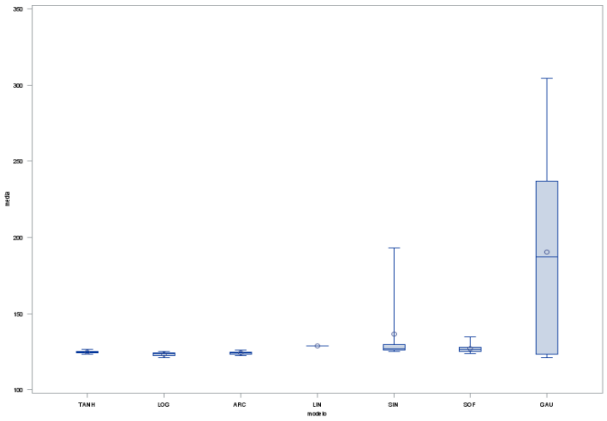


Ilustración 26. Diagrama de cajas y bigotes para decidir la función de activación.

Para la red de 8 nodos en un principio decidimos utilizar la función de activación logarítmica por tener menor media de ecm.

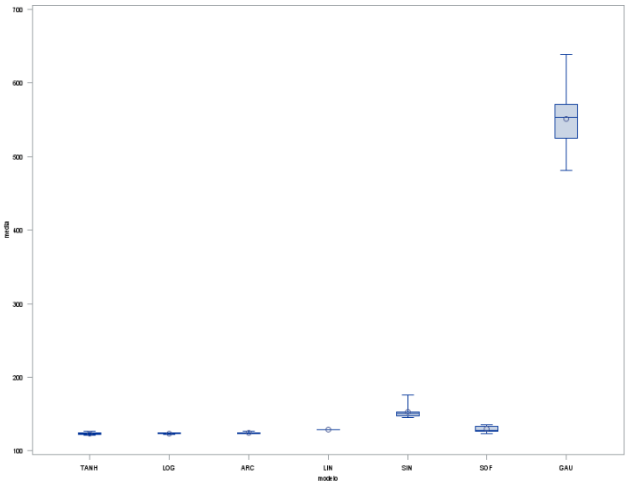


Ilustración 27. Diagrama de cajas y bigotes para decidir la función de activación.

ii. Estudio de los parámetros para el método de optimización

Una vez que hemos decidido la función de activación, haremos el mismo procedimiento para calcular el mejor método de optimización para nuestra red para 4, 6 y 8 capas ocultas. Al igual que antes y en adelante en este capítulo, los resultados los graficamos con un diagrama de cajas y bigotes, donde tenemos en el eje de ordenadas la media de los ecm calculados en la validación cruzada y en el eje de abscisas los diferentes modelos a comparar.

Probaremos con los métodos Trureg, Levmar, Quanew.

Para la red de 4 nodos, con la función de activación tangente hiperbólica, nos queda como mejor resultado el método de optimización Levmar, vemos como la media de los errores están más agrupados y menos variabilidad que en las dos opciones restantes.

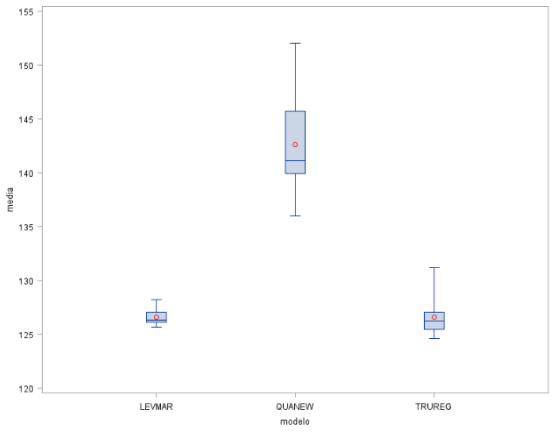


Ilustración 28. Diagrama de cajas y bigotes para decidir el método de optimización.

Para la red de 6 nodos, con la función de activación arco tangente, nos quedamos con el método de optimización Levmar (Levenberg Marquardt), por tener menor error y menor variabilidad.

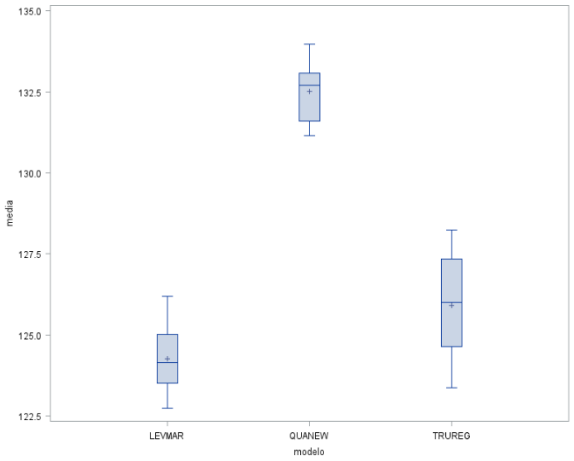


Ilustración 29. Diagrama de cajas y bigotes para decidir el método de optimización.

La red de 8 nodos calculada con la función de activación logarítmica nos queda que el mejor método para minimizar su error sería el algoritmo Levmar.

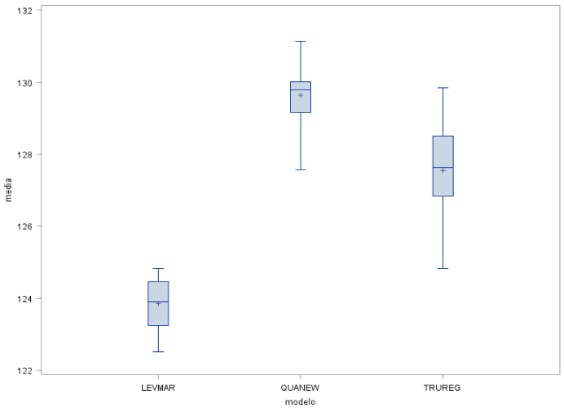


Ilustración 30. Diagrama de cajas y bigotes para decidir el método de optimización.

iii. Estudio del número de nodos ocultos

Una vez decididos la función de activación, los métodos de optimización, tenemos que analizar cuál es el número de nodos que funcionan mejor para nuestra red. Al inicio del capítulo hicimos una breve estimación, pero queremos asegurarnos a través de prueba y error, con validación cruzada, elegir el número de nodos en la capa oculta que mejor se ajusta a nuestros datos, con 5 grupos y 10 semillas diferentes, se plantea el estudio entre 2 y 9 nodos, ya que, recordamos que tenemos 7 variables input. Visualizaremos los datos mediante un diagrama de cajas y bigotes, describiendo los modelos y la media de su ecm.

Para el modelo con función de activación tangente hiperbólica y método de optimización Levmar, el número de nodos ideal sería 2, a pesar de tener un ecm mayor que las demás opciones tiene menor variabilidad.

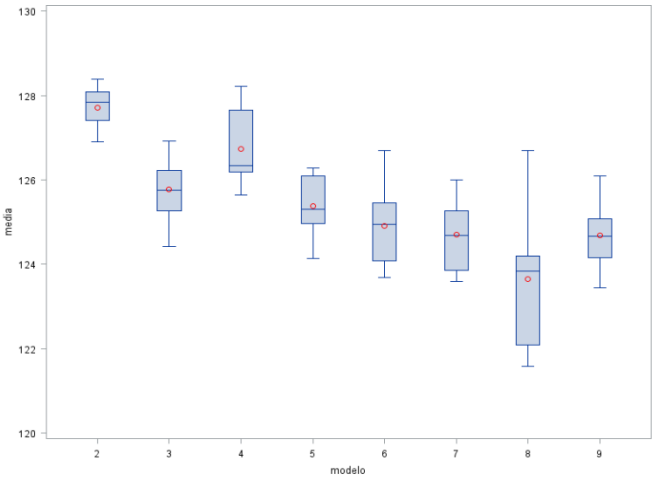


Ilustración 31. Diagrama de cajas y bigotes para decidir el número de nodos.

La red con función de activación arco tangente y método de optimización Levmar, el número de nodos ideal sería 3, a pesar de no ser de las opciones que menor error presenta, vemos como hay poca diferencia con respecto a las demás que tienen mayor variabilidad.

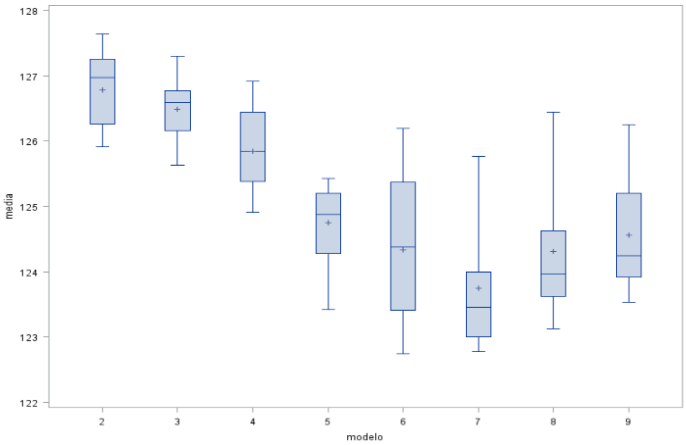


Ilustración 32. Diagrama de cajas y bigotes para decidir el número de nodos.

Para el modelo con función de activación logarítmica y método de optimización Levmar, el número de nodos ideal sería 2, por la misma casuística que los casos anteriores.

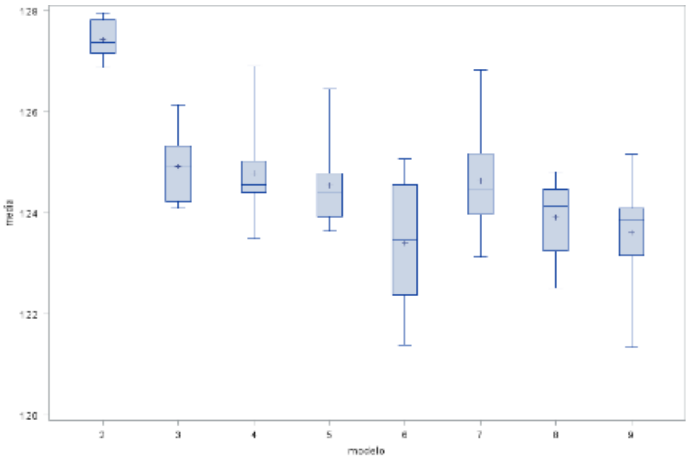


Ilustración 33. Diagrama de cajas y bigotes para decidir el número de nodos.

iv. Estudio de la posibilidad de aplicar Early Stopping

Estudiaremos la posibilidad de aplicar Early Stopping, para evitar el sobreajuste en nuestros modelos y ver cuántas iteraciones se puede ejecutar nuestras redes antes de exceder el ajuste. Para ello hemos calculado el error en cada iteración para los datos de entrenamiento y validación para cada red con los parámetros hasta ahora decididos.

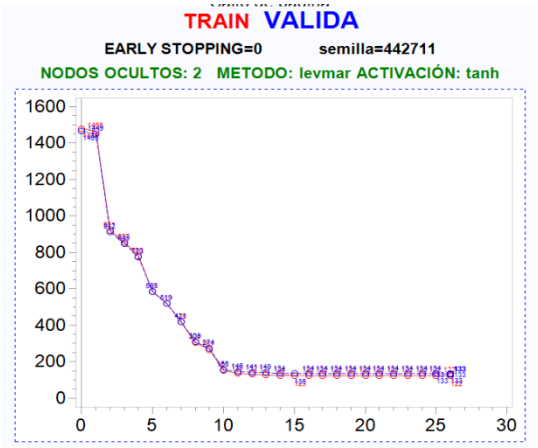


Ilustración 34. Error en función de las iteraciones para entrenamiento y validación.

Para esta red con dos nodos ocultos, función de activación tangente hiperbólica y algoritmo de optimización Levmar, aplicaremos Early Stopping en la iteración 5 y 10, es cuando el error se estabiliza.

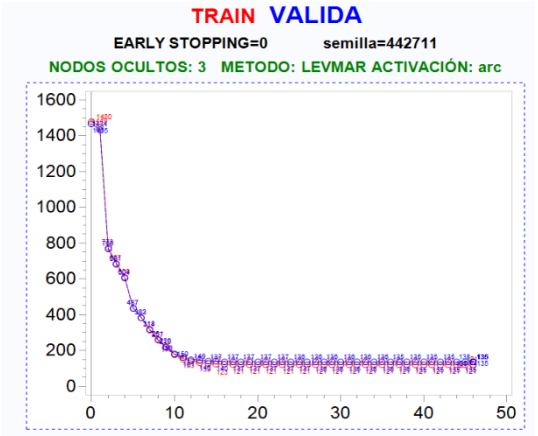


Ilustración 35. Error en función de las iteraciones para entrenamiento y validación.

La red con tres nodos ocultos, función de activación arco tangente y algoritmo de optimización Levmar, aplicaremos Early Stopping en la iteración 10, es cuando el error se estabiliza.

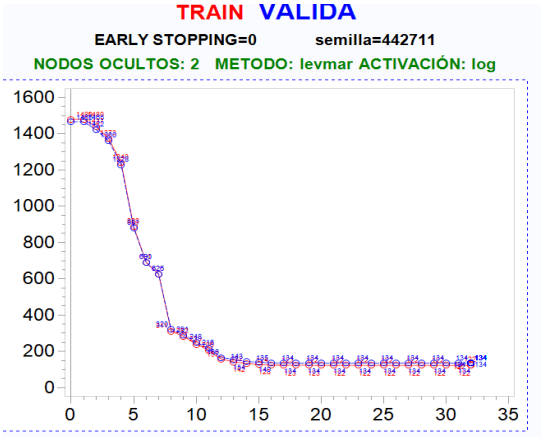


Ilustración 36. Error en función de las iteraciones para entrenamiento y validación.

La red con dos nodos ocultos, función de activación logarítmica y algoritmo de optimización Levmar, aplicaremos Early Stopping en la iteración 7 y 12, estudiaremos en cual es ellas funciona mejor el modelo.

Vemos como no hay mucha diferencia entre el error con los datos de entrenamiento y validación por lo que podemos afirmar que no tenemos sobreajuste, es algo a destacar.

v. Ejecución de los modelos

Por último, realizaremos validación cruzada con 55 semillas diferentes en las redes con los parámetros decididos, con 5 grupos y haremos la comparación visual entre los diferentes modelos según un diagrama de cajas y bigotes.

Hemos comparado las 8 redes:

- Dos nodos ocultos, función de activación Arco tangente y algoritmo de optimización Levmar. Hemos calculado sin y con aplicarle Early Stopping en la iteración 5 y 10. Modelos 1, 2 y 3, respectivamente.
- Tres nodos ocultos, función de activación Logarítmica y algoritmo de optimización Levmar. Hemos calculado sin y con aplicarle Early Stopping. Modelos 4 y 5, respectivamente.
- Dos nodos ocultos, función de activación Arco tangente y algoritmo de optimización Quanew. Hemos calculado sin y con aplicarle Early Stopping en la iteración 7 y 12. Modelos 6, 7 y 8, respectivamente.

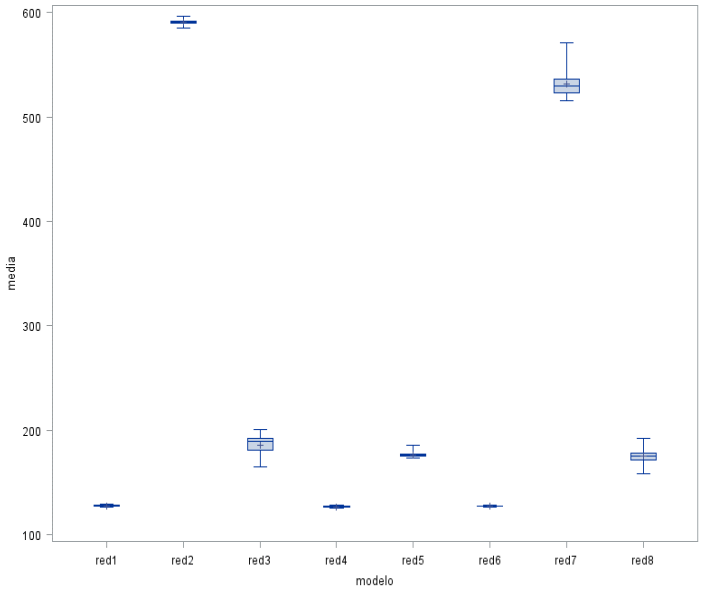


Ilustración 37. Diagrama de cajas y bigotes para comparar los modelos de redes neuronales.

Se ven a simple vista los mejores modelos son el 1º, 4º y 6º. Estudiándolos por separados tomamos la decisión de elegir el modelo final la red con dos nodos ocultos, función de activación Arco tangente y algoritmo de optimización Levmar sin aplicar Early Stopping (Modelo 6).

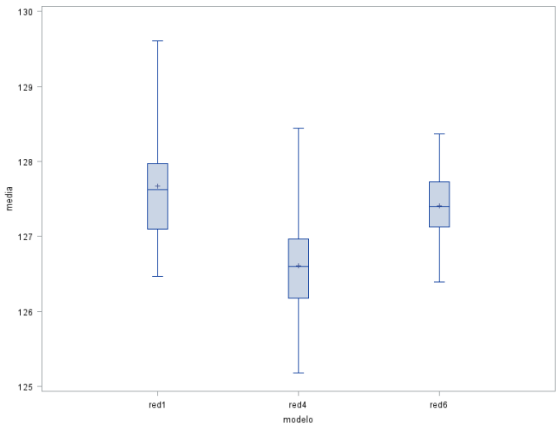


Ilustración 38. Diagrama de cajas y bigotes para comparar los modelos de redes neuronales.

vi. Evaluación del mejor modelo

Una vez calculado los parámetros para encontrar el mejor modelo de redes neuronales hemos calculado un total de más de 1100 modelos.

Hemos encontrado el mejor modelo creado por las variables subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, producto, empresa, fecha_semana y mes tiene una media de error de 127.411.

Estudiamos la importancia de las variables dentro del modelo y vemos como la variable más importante el precio de la subasta del día anterior.

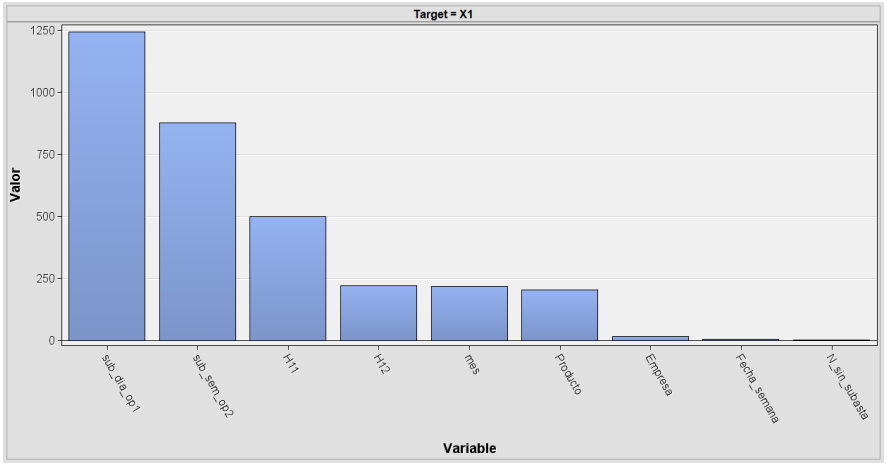


Ilustración 39. Importancia de la variable del modelo con técnicas de redes neuronales.

8. Evaluación para variable objetivo continua

Una vez realizada todas las pruebas que hemos considerado necesarias para encontrar los mejores modelos de regresión y redes neuronales, realizaremos una comparativa entre los dos modelos finales que nos han quedado.

Todos los modelos aplicaremos validación cruzada, con 55 semillas y 5 grupos.

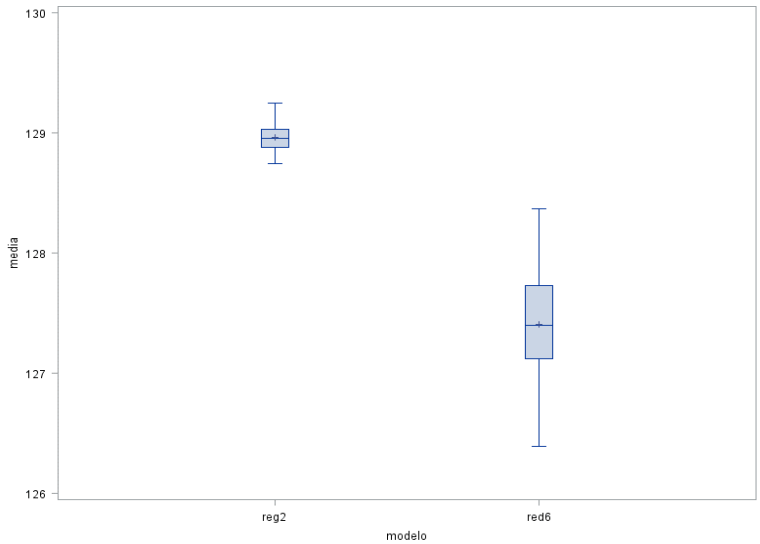


Ilustración 40. Diagrama de cajas y bigotes para comparar los modelos finales de redes neuronales y regresión lineal.

Entre los modelos probados, el modelo que mejor resultados nos da en relación sesgo-varianza es el modelo 2 de regresión lineal, a pesar de tener un error mayor.

A la hora de comenzar el trabajo pensábamos que las redes neuronales ganarían en cuanto a error considerablemente sobre cualquier modelo de regresión, a pesar de lo que pensábamos vemos como la regresión tiene un error mayor mínimo, por lo el modelo final elegido sería este, pues la regresión son modelos mucho más sencillos computacionalmente y más fáciles de explicar.

Estudiaremos la importancia de la variable dentro del modelo hallado con técnicas de regresión lineal y nos queda resultados similares a los encontrados en la red.

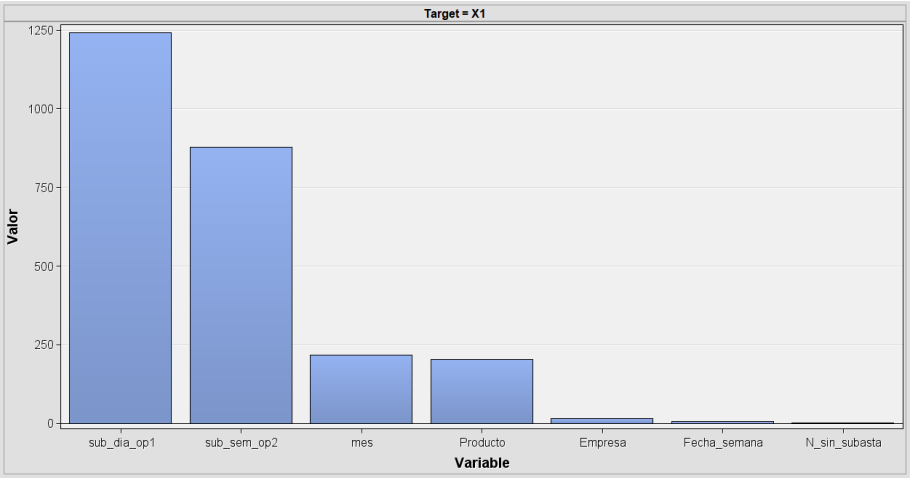


Ilustración 41. Importancia de la variable para el modelo de regresión lineal.

9. Modelización para variable objetivo dicotómica

En los próximos capítulos modelizaremos nuestra base de datos para encontrar el mejor modelo de predicción para una variable binaria.

Recordamos que nuestro objetivo es saber si el precio de una determinada clase de pimiento en una cierta empresa será más caro la semana que viene u hoy, dentro de 7 días. Para ello calcularemos modelos de Regresión Logística, Redes Neuronales, Bagging, Random Forest, Gradient Boosting, SVM y Emsablado.

Los conocimiento de estas técnicas han sido adquiridas en la asignatura de Técnicas de Machine Learning. Aprovecharemos los recursos proporcionados por Javier Portela, profesor de dicha asignatura.

Al igual que en el capítulo anterior, nuestra base de datos no podrá utilizar todas las variables que vienen recogidas como variables input, pues aportan información a futuro.

Utilizaremos en este capítulo como medida de ajuste la tasa de error, que viene dado por las malas predicciones entre las totales, es decir, los falsos positivos más los falsos negativos entre el total.

Este capítulo será realizado en SAS Guide.

I. Regresión Logística

Realizaremos modelos de regresión logística sobre nuestros datos depurados, con una partición 80% para entrenar el modelo y 20% para hacer testing.

En primer lugar, realizaremos regresión logística repetidas veces con diferentes datos de entrenamiento con selección de variables Forward, Stepwise y Backward. Calculamos una tabla de frecuencias para ver cuáles son los modelos seleccionados que más se repiten entre las diferentes semillas y éstos serán los candidatos para probar con validación cruzada.

Para el método de selección de variables Stepwise y Forward nos quedan los mismos resultados, estos son los dos modelos seleccionados que más se han repetido entre las diferentes semillas:

Efecto	Count	Percent	Modelo
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, Producto, Empresa, Fecha_semana, mes.	77	68.75000	1
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, max_año_anterior_op1, Producto, Empresa, Fecha_semana, mes.	7	6.2500	2
subasta_semana_anterior_op2, N_sin_subasta, Producto, Empresa, Fecha_semana, mes.	7	6.2500	3

Tabla 24. Efectos de la Regresión Logística con selección de variables Stepwise y Forward.

Realizando la regresión logística con la selección de variables Backward, nos quedan:

Efecto	Count	Percent	Modelo
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, media_semana_anterior_op1, media_semana_anterior_op2, Producto, Empresa, Fecha_semana, mes.	34	30.3571	4
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, media_semana_anterior_op1, media_dia_anterior_op2, min_dia_anterior_op2, media_semana_anterior_op2, Producto, Empresa, Fecha_semana, mes.	8	7.1429	5

Tabla 25. Efectos de la Regresión Logística con selección de variables Backward.

Con los modelos descritos aplicaremos validación cruzada, lo realizaremos para 55 semillas distintas. Analizaremos los resultados gráficamente a través de un diagrama de cajas y bigotes, donde compararemos los modelos de regresión logística por la media de la tasa de fallos. En el diagrama tenemos, en el eje de ordenadas la media de la tasa de error de la validación cruzada y eje de abscisas los distintos modelos a comparar.

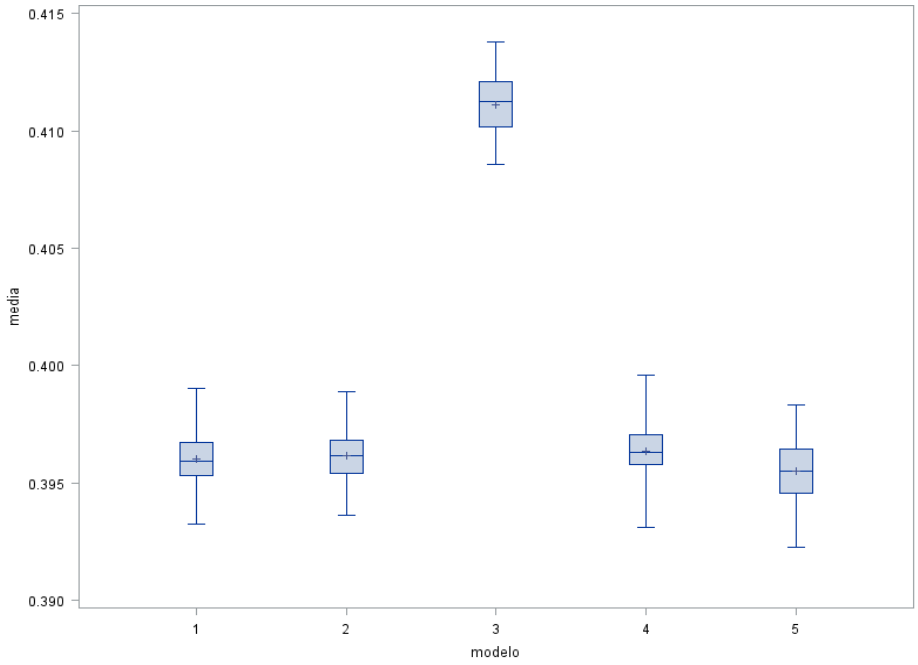


Ilustración 43. Diagrama de cajas y bigotes para comparar los modelos de regresión logística.

Los modelos que consideramos que tienen mejores resultados son los modelos 2 y 5, pues a pesar de tener mucha variabilidad, vemos como las medias de las tasas de error son simétricas comparándolos con los demás modelos, y la tasa de fallos en media no es significativa para seleccionar uno de ellos, a excepción del modelo 3. Estos modelos serán los que nos servirán para la selección de variables para aplicar técnicas de Redes Neuronales, Bagging y SVM.

Como modelo finalista con técnicas de regresión logística sería el modelo 2 por tener menor variabilidad que en las demás opciones.

i. Evaluación del mejor modelo

Una vez calculado un total de más 1000 modelos con técnicas de regresión logística, combinando distintos criterios de selección de variables y distintas semillas, llegamos a que el modelo finalista está compuesto por las variables subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, max_año_anterior_op1, Producto, Empresa, Fecha_semana, mes, teniendo un error igual a 0.39.

Analysis of Maximum Likelihood Estimates

Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq	Estimador estandarizado	Exp(Est)
Intercept		1	-1.4481	0.1621	79.84	<.0001		0.235
Empresa	AGROEJ. BERJA	1	0.1480	0.5747	0.07	0.7967		1.160
Empresa	AGROEJ. EJIDO	1	-0.2521	0.2367	1.13	0.2869		0.777
Empresa	AGROPONIENTE	1	-0.4300	0.1308	10.81	0.0010		0.651
Empresa	AGROPONIENTE 2	1	0.4938	0.4547	1.18	0.2775		1.638
Empresa	AGRUPAADRA	1	-0.1069	0.1427	0.56	0.4536		0.899
Empresa	AGRUPAEJIDO	1	-0.3191	0.1295	6.07	0.0137		0.727
Empresa	CEHORPA	1	-0.1255	0.1303	0.93	0.3357		0.882
Empresa	COSTA ALMERIA	1	-0.3659	0.1321	7.67	0.0056		0.694
Empresa	FEMAGO	1	-0.3623	0.1486	5.95	0.0147		0.696
Empresa	LA COSTA	1	0.1047	0.4266	0.06	0.8061		1.110
Empresa	LA UNION	1	-0.3768	0.1430	6.94	0.0084		0.686
Fecha_semana	2	1	-0.3659	0.0983	13.87	0.0002		0.694
Fecha_semana	3	1	0.1273	0.0478	7.10	0.0077		1.136
Fecha_semana	4	1	0.0555	0.0480	1.33	0.2481		1.057
Fecha_semana	5	1	0.1171	0.0485	5.83	0.0158		1.124
Fecha_semana	6	1	0.0394	0.0486	0.66	0.4172		1.040
N_sin_subasta		1	0.5651	0.1048	29.05	<.0001	0.1237	1.760
Producto	Pimiento Corto Amarillo	1	-0.1387	0.0676	4.20	0.0403		0.871
Producto	Pimiento Corto Rojo	1	0.0310	0.0539	0.33	0.5656		1.031
Producto	Pimiento Corto Verde	1	0.3351	0.0784	18.26	<.0001		1.398
Producto	Pimiento Italiano Verde	1	0.1155	0.0481	5.75	0.0165		1.122
Producto	Pimiento Largo Rojo	1	-0.4340	0.0453	91.63	<.0001		0.648
max_ano_op1		1	0.000716	0.000546	1.72	0.1899	0.0173	1.001
mes	1	1	-0.4394	0.0550	63.93	<.0001		0.644
mes	2	1	-0.5967	0.0566	111.35	<.0001		0.551
mes	3	1	0.0902	0.0594	2.30	0.1291		1.094
mes	4	1	0.4882	0.0680	51.50	<.0001		1.629
mes	5	1	0.2753	0.0787	12.23	0.0005		1.317
mes	6	1	-0.1632	0.0932	3.07	0.0799		0.849
mes	7	1	0.0868	0.1091	0.63	0.4263		1.091
mes	8	1	-0.1947	0.1085	3.22	0.0727		0.823
mes	9	1	0.1654	0.0916	3.26	0.0710		1.180
mes	10	1	0.4838	0.0808	35.84	<.0001		1.622
mes	11	1	-0.0693	0.0698	0.98	0.3210		0.933
sub_dia_op1		1	0.0172	0.000988	304.90	<.0001	0.3654	1.017
sub_sem_op2		1	-0.00262	0.000946	7.68	0.0056	-0.0560	0.997

Ilustración 44. Análisis del estimado de máxima verosimilitud de la Regresión Logística.

En la columna *Estimate* podemos ver los parámetros β_i de cada variable independiente.

Además, en la columna *Exp(Est)* nos da los resultados del Odds Ratio, por lo que podemos decir que cuando el producto subastado es vendido en el mes de abril o marzo hay aproximadamente 1.6 veces más posibilidades de que el precio de la semana que viene sea superior a esta, comparado con vender el producto en el mes de diciembre, suponiendo constantes todas las demás variables.

Si aumentamos en una unidad la variable precio de la subasta de la semana pasada empeoran las posibilidades de que el precio de la semana que viene sea mayor que el día actual.

II. Redes Neuronales

En este capítulo calcularemos modelos con la técnica de Redes Neuronales para nuestra base de datos. Estudiaremos el número de nodos en la capa oculta, el punto de corte, la función de activación, método de optimización, todo ello a través de validación cruzada, las decisiones las tomaremos en base a la tasa de fallos, y teniendo cuidado en no cometer sobreajuste.

Las variables utilizadas han sido seleccionadas previamente, a través de técnicas de regresión logística. Utilizaremos dos sets de variables, los modelos 2 y 5, explicados en el capítulo anterior.

i. Estudio del número de nodos y el punto de corte

Para elegir el número de nodos y el punto de corte de la decisión, lo haremos por validación cruzada para una única partición, variando entre 2 y 10 nodos con paso 1, decidimos estudiar el número de nodos comprendido en este rango porque pensamos que lo ideal es que la red tenga forma de embudo a ser posible, como estudiamos dos sets de variables de entrada, uno de ellos con 8 y 11 variables.

Para el modelo 2, compuesto por las variables subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, max_año_anterior_op1, Producto, Empresa, Fecha_semana, mes, creemos que la mejor decisión sería escoger una red con 4 nodos con corte 55, ya que, presenta un sesgo inferior al 40%, similar a los demás modelos con un número mayor de nodos. Aunque como vemos en el diagrama de cajas y bigotes las redes con el mismo corte presentan una tasa de error similar independientemente del número de nodos en la capa oculta.

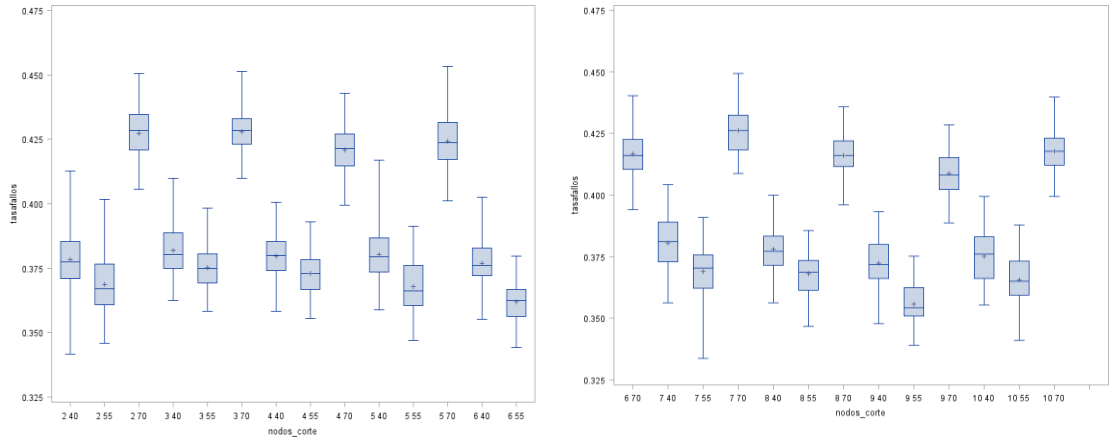


Ilustración 45. Diagrama de cajas y bigotes para decidir el corte y número de nodos de la red neuronal.

Para el modelo 5, compuesto por las variables subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, media_semana_anterior_op1, media_dia_anterior_op2, min_dia_anterior_op2, media_semana_anterior_op2, Producto, Empresa, Fecha_semana, mes, el número de nodos elegido es 6 con un corte de 55.

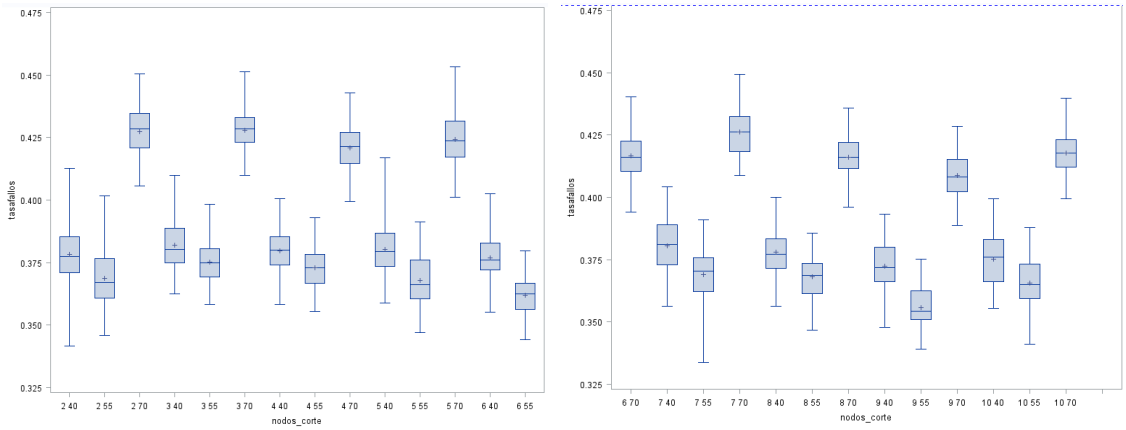


Ilustración 46. Diagrama de cajas y bigotes para decidir el corte y número de nodos de la red neuronal.

ii. Estudio de los parámetros para la función de activación

Buscamos la función de activación que mejor se ajusta a nuestra red para el modelo 2 con 4 nodos y para el modelo 5 con 6 nodos en la capa oculta, al igual que hasta ahora, lo haremos con validación cruzada para 55 semillas diferentes. Compararemos a través de la media de la tasa de error las funciones tangente hiperbólica, arco seno, lineal, seno, logarítmica, gaussiana y softmax. Lo graficaremos a través de un diagrama de cajas y bigotes donde por un lado tenemos los modelos con las diferentes funciones y por otro, la media de la tasa de fallos como resultado de la validación cruzada realizada.

Para el modelo 2, vemos en el gráfico como las funciones de activación tangente hiperbólica, lineal y seno tienen una tasa de error media bastante menor al de las demás funciones, por lo que probaremos la combinación de éstas para hallar el método de optimización que más se ajusta a nuestra red de 4 nodos.

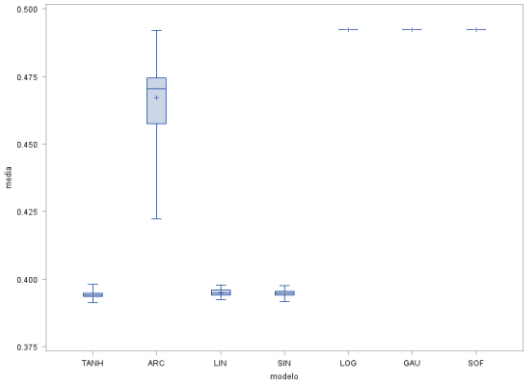


Ilustración 47. Diagrama de cajas y bigotes para decidir la función de activación.

Vemos como los resultados son similares para el modelo 5, con 6 nodos en la capa oculta. Al igual que en el caso anterior, probaremos la combinación de las 3 funciones de activación con mejores resultados que nos han quedado. En ambos casos, para hallar estos resultados hemos utilizado el método de optimización back propagation.

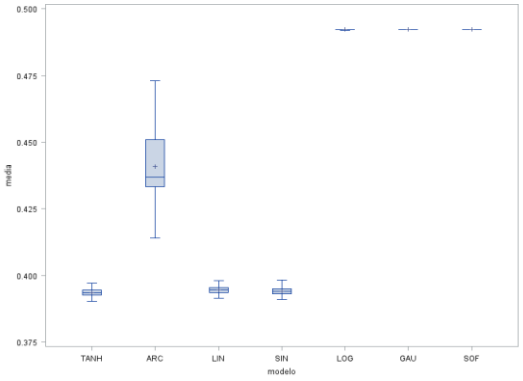


Ilustración 48. Diagrama de cajas y bigotes para decidir la función de activación.

iii. Estudio de los parámetros para el método de optimización

Analizaremos el método de optimización, para ello compararemos a través de validación cruzada la tasa de error media, de los métodos Back Propagation, Levmar y Quanew, explicados en el capítulo de Redes Neuronales con variable objetivo continua.

Para ambos sets de variables de entrada estudiaremos su método de optimización para las funciones de activación tangente hiperbólica, lineal y seno.

Modelo 2 con 4 nodos en la capa oculta

Función de activación lineal, elegimos como mejor método Levmar, pues los datos (las medias de las tasas de errores) son simétricos, a pesar de tener mucha variabilidad los otros métodos tienen los mismos problemas.
Vemos como todos ellos están muy ajustados por la tasa de error media.

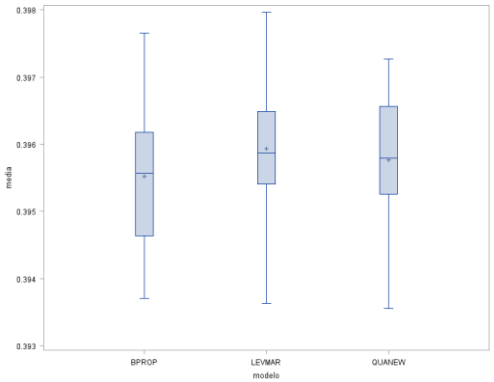


Ilustración 49. Diagrama de cajas y bigotes para decidir el método de optimización.

Función de activación seno, en esta gráfica vemos como hay más diferencia de error entre los diferentes métodos.

El método de optimización elegido es el quanew, pues tiene un error menor con menos variabilidad. Además, sus datos (las medias de las tasas de errores) son simétricos.

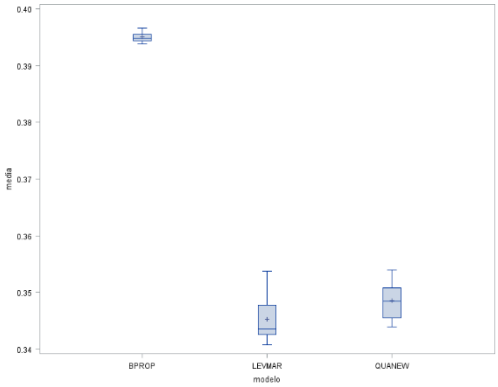


Ilustración 50. Diagrama de cajas y bigotes para decidir el método de optimización.

Función de activación tangente hiperbólica, a simple vista vemos como el método idóneo para nuestra red sería el método levmar.

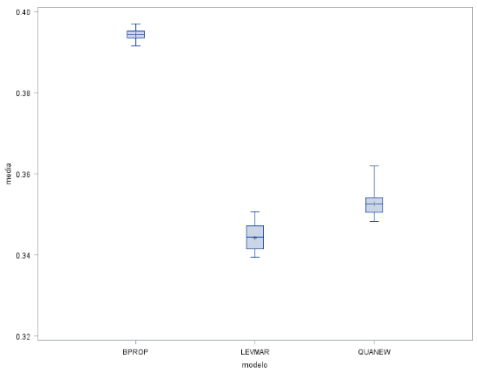


Ilustración 51. Diagrama de cajas y bigotes para decidir el método de optimización.

Modelo 5 con 6 nodos en la capa oculta

Función de activación lineal, con método de optimización levmar.

Los tres métodos poseen un error similar, pero levmar tiene menos variabilidad, con unos datos (las medias de las tasas de errores) más simétricos.

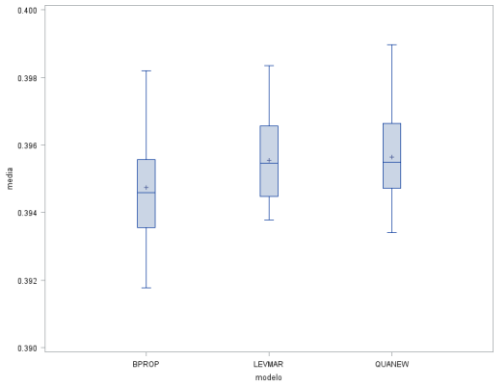


Ilustración 52. Diagrama de cajas y bigotes para decidir el método de optimización.

Función de activación seno, elegimos como mejor método levmar, menor error medio, con menor variabilidad.

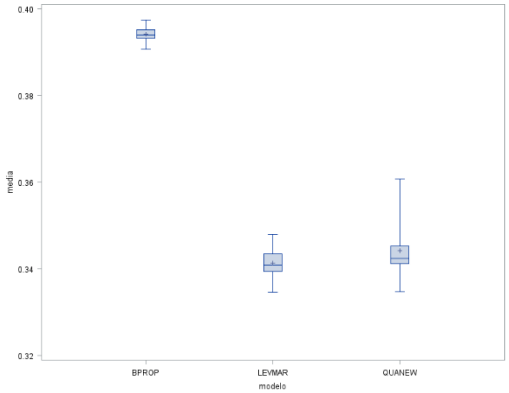


Ilustración 53. Diagrama de cajas y bigotes para decidir el método de optimización.

Función de activación tangente hiperbólica, elegimos como mejor método de optimización levmar. Al igual que en el caso anterior, menor error y variabilidad que en los demás métodos.

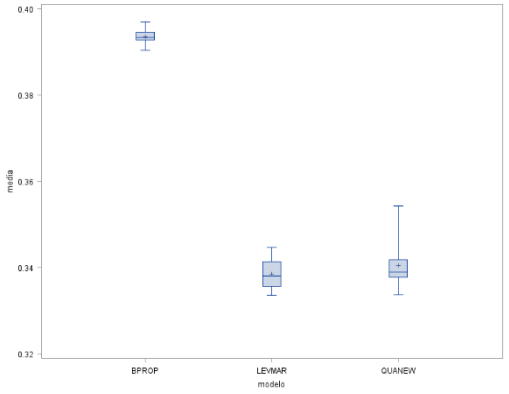


Ilustración 54. Diagrama de cajas y bigotes para decidir el método de optimización.

iv. Ejecución de los modelos

Una vez decidido el número de nodos, la función de activación y el método de optimización nos han quedado un total de 6 redes neuronales a probar por validación cruzada para elegir el mejor modelo. Nos han quedado las redes:

Set de variables	Nº nodos	Función de activación	Método de optimización	Modelo
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, max_año_anterior_op1, Producto, Empresa, Fecha_semana, mes.	4	Seno	Quanew	Red1
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, max_año_anterior_op1, Producto, Empresa, Fecha_semana, mes.	4	Lineal	Levmar	Red2
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, max_año_anterior_op1, Producto, Empresa, Fecha_semana, mes.	4	Tanh	Levmar	Red3
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, media_semana_anterior_op1, media_dia_anterior_op2, min_dia_anterior_op2, media_semana_anterior_op2, Producto, Empresa, Fecha_semana, mes.	6	Tanh	Levmar	Red4
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, media_semana_anterior_op1, media_dia_anterior_op2, min_dia_anterior_op2, media_semana_anterior_op2, Producto, Empresa, Fecha_semana, mes.	6	Lineal	Levmar	Red5

subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, media_semana_anterior_op1, media_dia_anterior_op2, min_dia_anterior_op2, media_semana_anterior_op2, Producto, Empresa, Fecha_semana, mes.	6	Seno	Levmar	Red6

Tabla 27. Resumen de los parámetros de las redes neuronales a comparar.

No hemos aplicado early stopping por problemas con el espacio computacional.

Una vez hecha las redes mencionadas con validación cruzada, las graficamos a través de un diagrama de cajas y bigotes como llevamos haciendo hasta ahora para poder compararlas con la media de la tasa de error.

Viendo el diagrama podemos descartar en una primera instancia los modelos de redes 2 y 5, por tener un mayor error, y la red 4 por tener demasiada variabilidad con respecto a las demás.
Analizando los modelos de redes restantes 1, 3 y 6, nos queda que la red 6 la descartamos por tener un error similar a las demás, pero tiene más variables de entrada.
Finalmente nos quedamos con el modelo 3, tiene un error menor y menor variabilidad que el modelo 1.

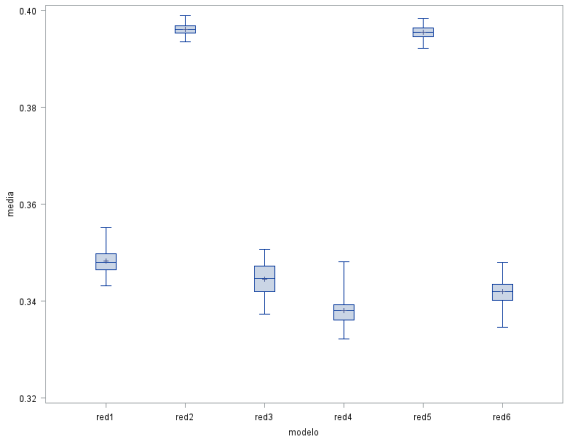


Ilustración 55. Diagrama de cajas y bigotes para comparar los modelos de redes neuronales.

v. Evaluación del mejor modelo

Después de un proceso exhaustivo de la búsqueda de la mejor red neuronal para nuestro conjunto de datos, hemos probado un proceso de más de 5000 modelos con diferentes semillas y combinando diferentes parámetros de las redes.

El modelo elegido contiene a las variables subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, max_año_anterior_op1, Producto, Empresa, Fecha_semana, mes, y un total de 4 nodos. Con la función de activación tanh y el método de optimización Levmar. Tiene un error cuadrático medio igual a 0.34.

Estudiaremos la importancia de las variables en el modelo para ver cuáles son las variables que más valor aportan a nuestro modelo de Redes Neuronales. Lo haremos a través de SAS Miner con los parámetros que hemos decidido para una única semilla.

Nos quedan que la variable más importantes son el precio de la subasta del día anterior, mes, y el precio de la subasta de la semana anterior. Parece lógico pensar que son las más importantes, pues el precio de la subasta del día y la semana anterior serán precios

similares al precio del día actual, y el mes, pues normalmente siempre hay precios similares en los meses del año independientemente del año que estemos.
Nos parece interesante resaltar que los nodos de la red son de gran importancia para el modelo.

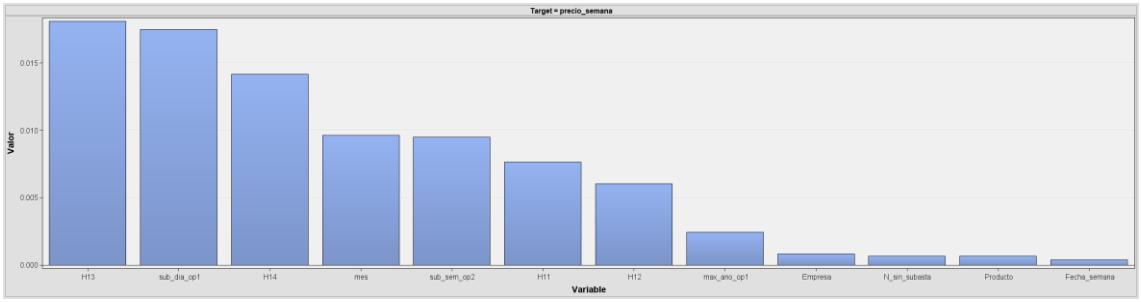


Ilustración 56. Importancia de la variable del modelo de redes neuronales.

El punto de corte donde el modelo clasificará que el precio de la semana que viene es superior al precio actual es aquellos que tengan una probabilidad superior al punto de corte, en este caso, 0.55, este parámetro no es utilizado para el entrenamiento.

III. Bagging

En este apartado realizaremos modelos bagging, se trata de un modelo de predicción basado en árboles. Los árboles de decisión generalmente no tienen el mismo nivel de precisión predictiva que los modelos vistos hasta ahora, sin embargo, al agregar muchos árboles de decisión, el rendimiento predictivo puede mejorarse.

La idea de Bagging es ajustar muchos árboles de decisión en paralelo formando un 'bosque', donde cada uno de ellos participen aportando su predicción. Cada árbol es distinto al anterior porque se entrena con diferentes muestras haciendo remuestreo a partir de la muestra original, por lo que cada modelo se ajusta de manera independiente a los demás.

La construcción del mejor modelo Bagging se llevará a cabo a través de un proceso de prueba y error, combinando los diferentes parámetros propios del modelo. Dichos modelos se realizarán con validación cruzada y se compararán a través de la media de la tasa de fallos.
Utilizaremos los dos sets de variables resultado de los mejores modelos de Regresión Logística, recordamos que eran los modelos 2 y 5, pues nos servirá como selección de variables de nuestros datos originales como en el caso de los modelos de Redes Neuronales. Además, probaremos con todas las variables de entrada para ver si hay diferencia significativa en el error haciendo la selección de variables previa.

Los parámetros fijos que dejaremos será el número máximo de árboles a crear, lo hemos dejado en 100 árboles pues creemos que es más que suficiente y así no ocupar exceso de memoria computacional, además definimos un punto de corte igual a 0.5, si el modelo tiene una probabilidad superior al 50% el precio de la subasta de la semana que viene será superior al del día actual. Combinaremos los parámetros, del set de variables de entrada, como hemos comentado, tamaño mínimo de la hoja, p-valor para decidir el nivel de restricción en la regla de división, porcentaje de la población que se muestrea en la construcción de cada árbol.

Los modelos creados son los siguientes:

Set de variables	Tamaño mínimo de la hoja	P-valor	% de población que se muestrea para construir el árbol	Modelo
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, max_año_anterior_op1, Producto, Empresa, Fecha_semana, mes.	5	0.1	80%	Bg1
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, max_año_anterior_op1, Producto, Empresa, Fecha_semana, mes.	5	0.2	70%	Bg2
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, max_año_anterior_op1, Producto, Empresa, Fecha_semana, mes.	6	0.15	60%	Bg3
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, media_semana_anterior_op1, media_dia_anterior_op2, min_dia_anterior_op2, media_semana_anterior_op2, Producto, Empresa, Fecha_semana, mes.	5	0.1	80%	Bg4
subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, media_semana_anterior_op1, media_dia_anterior_op2, min_dia_anterior_op2, media_semana_anterior_op2, Producto, Empresa, Fecha_semana, mes.	5	0.2	70%	Bg5

subasta_dia_anterior_op1, subasta_semana_anterior_op2, N_sin_subasta, media_semana_anterior_op1, media_dia_anterior_op2, min_dia_anterior_op2, media_semana_anterior_op2, Producto, Empresa, Fecha_semana, mes.	6	0.15	60%	Bg6
Todas las variables del conjunto de datos después de haber sido tratadas, un total de 28 variables.	5	0.1	80%	Bg7
Todas las variables del conjunto de datos después de haber sido tratadas, un total de 28 variables.	5	0.2	70%	Bg8
Todas las variables del conjunto de datos después de haber sido tratadas, un total de 28 variables.	6	0.15	60%	Bg9

Tabla 28. Resumen de los parámetros de los modelos Bagging a comparar.

Los modelos de bagging los compararemos a través de un diagrama de cajas y bigotes, a través de una validación cruzada para garantizar que son independientes entre los datos de entrenamiento y validación.

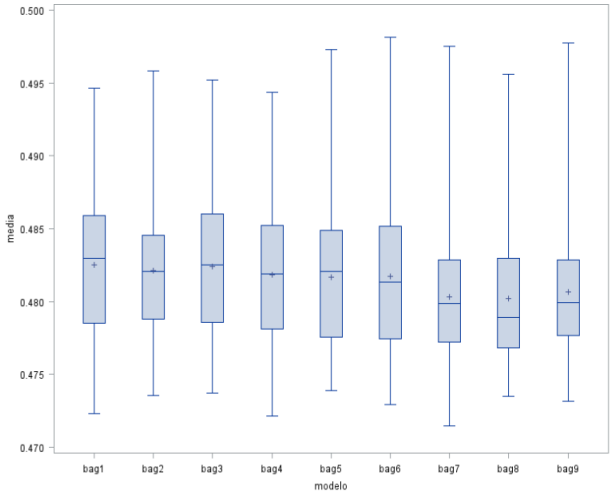


Ilustración 57. Diagrama de cajas y bigotes para comparar los modelos de Bagging.

Todos los modelos calculados están muy iguales en cuanto la media de la tasa de fallos, pero cabe destacar el modelo bag3 posee menor variabilidad y unos datos (las medias de las tasas de errores) con mayor simetría con respecto a los demás, y menor número de variables.
Este modelo será comparado con los demás modelos resultantes de las otras técnicas.

i. Evaluación del mejor modelo

Hemos analizado más de 450 modelos de Bagging con diferentes combinaciones de parámetros y semillas. Como conclusión, hemos elegido el modelo Bag3 como ganador.

Analizaremos la importancia de la variable dentro del modelo, con SAS Miner, nos queda la gráfica,

Importancia de la variable

Nombre de la variable	Número de reglas de división
sub_dia_op1	3716
sub_sem_op2	3575
Producto	3379
max_ano_op1	2710
mes	2295
Fecha_semana	1265
Empresa	1201
N_sin_subasta	1170

Ilustración 58. Importancia de las variables del modelo ganador por la técnica Bagging.

Vemos como las variables más importante son precio de la subasta del día y semana anteriores, vemos como son los resultados similares a los de los modelos de Redes Neuronales.

IV. Random Forest

La técnica de Random Forest es una técnica basada en árboles, es una modificación de Bagging, incorpora la aleatoriedad de las variables utilizadas para segmentar cada nodo del árbol. Los árboles van a estar no correlaciones al igual que en la técnica Bagging, la diferencia está que en RF antes de hacer los árboles el modelo elige m variables predictoras influyentes, haciendo su propia selección de variables.

Realizaremos un proceso de prueba y error, como hasta ahora hemos venido haciendo, combinando los diferentes parámetros del modelo.

Utilizaremos el set de variables de entrada todas las variables tratadas de la base de datos, pues como hemos mencionado el propio modelo realizará la selección de variables conveniente.

Los parámetros que dejaremos fijos será el número máximo de divisiones del nodo que será 2, pues queremos construir únicamente árboles binarios, construiremos un máximo de 100 árboles por problemas con la memoria computacional, elegiremos una profundidad igual a 10, número de hojas finales, el punto de corte igual a 0.5 y tomaremos un 75% de la población que se muestrea para construir el árbol. Realizaremos una combinación con los restantes parámetros.

Los modelos creados son los siguientes:

Nº de variables a sortear en cada nodo	Tamaño mínimo de la hoja	P-valor	Modelo
6	5	0.1	RF1
6	6	0.2	RF2
8	5	0.1	RF3
8	6	0.2	RF4
10	5	0.1	RF5
10	6	0.2	RF6
12	5	0.1	RF7
12	6	0.2	RF8

Tabla 29. Resumen de los parámetros de los modelos Random Forest a comparar.

Los modelos descritos los hemos realizado con validación cruzada, y nos ha quedado,

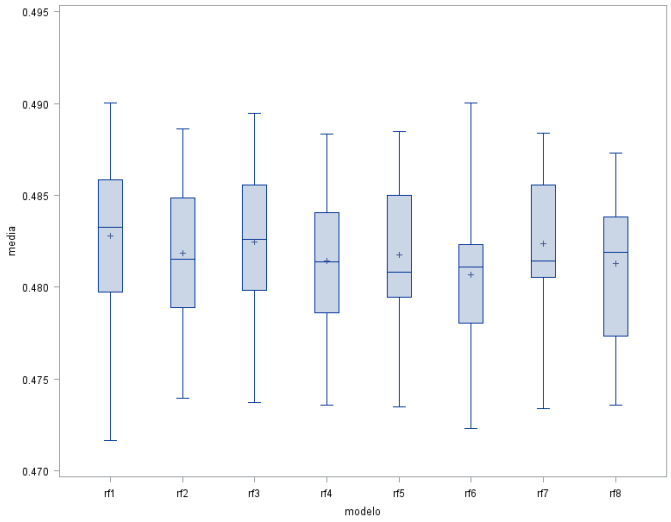


Ilustración 59. Diagrama de cajas y bigotes para comparar los modelos de Random Forest.

Todos los modelos creados a través de técnicas de Random Forest tienen una tasa de error media inferior al 49%, son modelos muy similares.
Si tenemos que elegir un modelo finalista creemos que el mejor sería el modelo 4, porque a pesar de que el modelo 5 y 8 tengan un error menor, el modelo 5 tiene mucha variabilidad y el modelo 8 los datos (las medias de las tasas de errores) no están agrupados.

i. Evaluación del mejor modelo

Una vez hecho los 440 modelos con técnicas Random Forest, creando combinaciones entre sus parámetros y probando con diferentes semillas llegamos a la conclusión que el mejor modelo es el formado por 8 variables con un p-valor igual a 0.2, lo que no es muy restrictivo y el tamaño mínimo de su hoja es igual 6.

Estudiaremos la importancia de sus variables y nos queda que,

Importancia de la variable

Nombre de la variable	Número de reglas de división
mes	1402
sub_dia_op2	1313
sub_ano_op2	1172
Producto	990
sub_sem_op2	927
avg_ano_op1	826
min_ano_op1	771
avg_ano_op2	760
max_ano_op1	756
sub_sem_op1	730
min_ano_op2	729
max_sem_op1	723
max_sem_op2	710
max_ano_op2	708
min_dia_op1	705
sub_dia_op1	673
min_sem_op1	632
min_dia_op2	629
avg_sem_op2	622
avg_sem_op1	603
avg_dia_op1	599
min_sem_op2	583
max_dia_op2	575
max_dia_op1	531
N_sin_subasta	507
avg_dia_op2	482
Empresa	464
Fecha_semana	291

Ilustración 60. Importancia de las variables del modelo ganador por la técnica Random Forest.

Vemos como los resultados no son similares al modelo ganador de la técnica Bagging.
Resulta curioso ver como Fecha_semana es la variable que menos se utiliza en el modelo y hasta ahora la hemos utilizado dentro de la selección de variable hecha por técnicas de Regresión Logística.
Vemos como las variables más utilizadas son mes, precio de la subasta del dia y año anterior.

V. Gradient Boosting

La técnica Gradient Boosting al igual que Random Forest y Bagging, explicadas en capítulos anteriores, son métodos basados en árboles.
El algoritmo Gradient Boosting consiste en construir un bosque modificando las predicciones iniciales intentando ir minimizando los residuos en la dirección de decrecimiento, ajustando así las predicciones más a los datos.

Como variables de entrada al modelo serán utilizadas todas las variables tratadas en la muestra de nuestros datos, ya que, el modelo hace su propia selección de variables.

Los modelos creados son los siguientes:

Tamaño mínimo de la hoja final	Máxima profundidad	Parámetro de regularización	Modelo
5	4	0.01	Gb1
4	5	0.01	Gb2
3	6	0.01	Gb3
5	4	0.02	Gb4
4	5	0.02	Gb5
3	6	0.02	Gb6
5	4	0.01	Gb7
4	5	0.01	Gb8
3	6	0.01	Gb9
7	7	0.03	Gb10

Tabla 30. Resumen de los parámetros de los modelos Gradient Boosting a comparar.

Sobre los modelos con los parámetros planteados realizamos validación cruzada y comparándolos a través de un diagrama de cajas y bigotes nos queda:

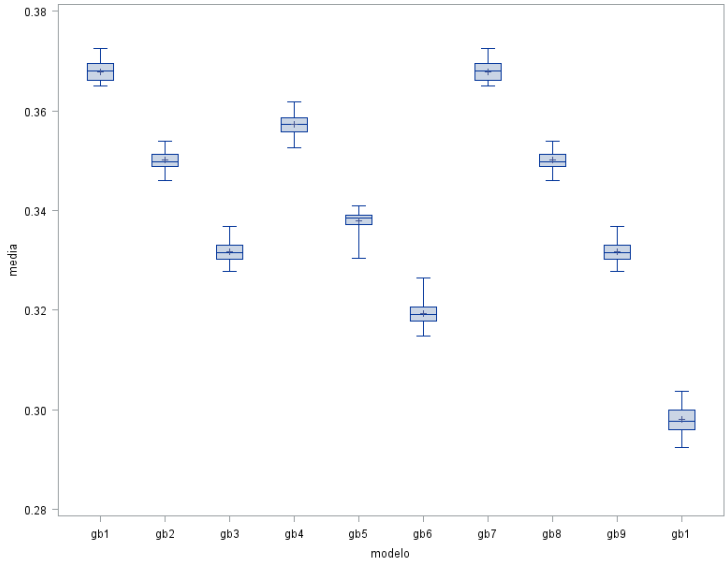


Ilustración 61. Diagrama de cajas y bigotes para comparar los modelos de Gradient Boosting.

A simple vista podemos afirmar que el mejor modelo es el modelo 10, pues es el modelo que menor media de tasa de fallos tiene.

i. Evaluación del mejor modelo

El modelo finalista realizado a través de técnicas de Gradient Boosting es el modelo con el parámetro de regularización igual a 0.3, el más alto entre los probados. Tiene un tamaño mínimo de hoja igual a 7.

Las variables que más aportan al modelo son:

Ilustración 62. Importancia de las variables del modelo ganador por la técnica Gradient Boosting.

Observamos como hay diferencias en comparación con la importancia de la variable de los modelos calculados hasta ahora.
La variable más importante del modelo sería subasta dia anterior, mes y Producto.

VI. Ensamblado

El ensamblado de modelos combina varias técnicas de machine learning, nosotros utilizaremos las técnicas vistas hasta ahora, Regresión Logística, Redes Neuronales, Bagging, Random Forest y Gradient Boosting.

Para esta técnica utilizaremos para la combinación de modelos el método Stacking, combinando las predicciones a partir del promedio, es decir, a partir de los mejores modelos calculados con las técnicas vistas hasta ahora, la variable respuesta será un promedio de ellas. Calcularemos combinaciones entre ellas para buscar los mejores resultados.

La combinación de modelos a probar por validación cruzada con 5 grupos y 55 semillas diferentes será:

Técnica	Modelo
Logística	LOG
Red	RED
Random Forest	RF
Gradient Boosting	GB
Logística + Red	LOG-RED
Logística + Random Forest	LOG-RF
Logística + Gradient Boosting	LOG-GB
Red + Random Forest	RED-RF
Red + Gradient Boosting	RED-GB
Random Forest + Gradient Boosting	RF-GB
Logística + Red + Random Forest	L-R-RF

Logística + Red + Gradient Boosting	L-R-BG
Logística + Random Forest + Gradient Boosting	L-RF-GB
Red + Random Forest + Gradient Boosting	R-RF-GB
Logística + Red + Random Forest + Gradient Boosting	LRRFGB
0.2*Logística + 0.1*Red + 0.5*Random Forest + 0.2*Gradient Boosting	(LRRFGB)P

Tabla 31. Resumen de los modelos calculados con técnicas de modelos de ensamblado.

Lo graficamos a través de un diagrama de cajas y bigotes para poder compararlos visualmente, por lado tenemos la media de la tasa de error y en el eje de abscisas los modelos en el orden definido. Nos queda:

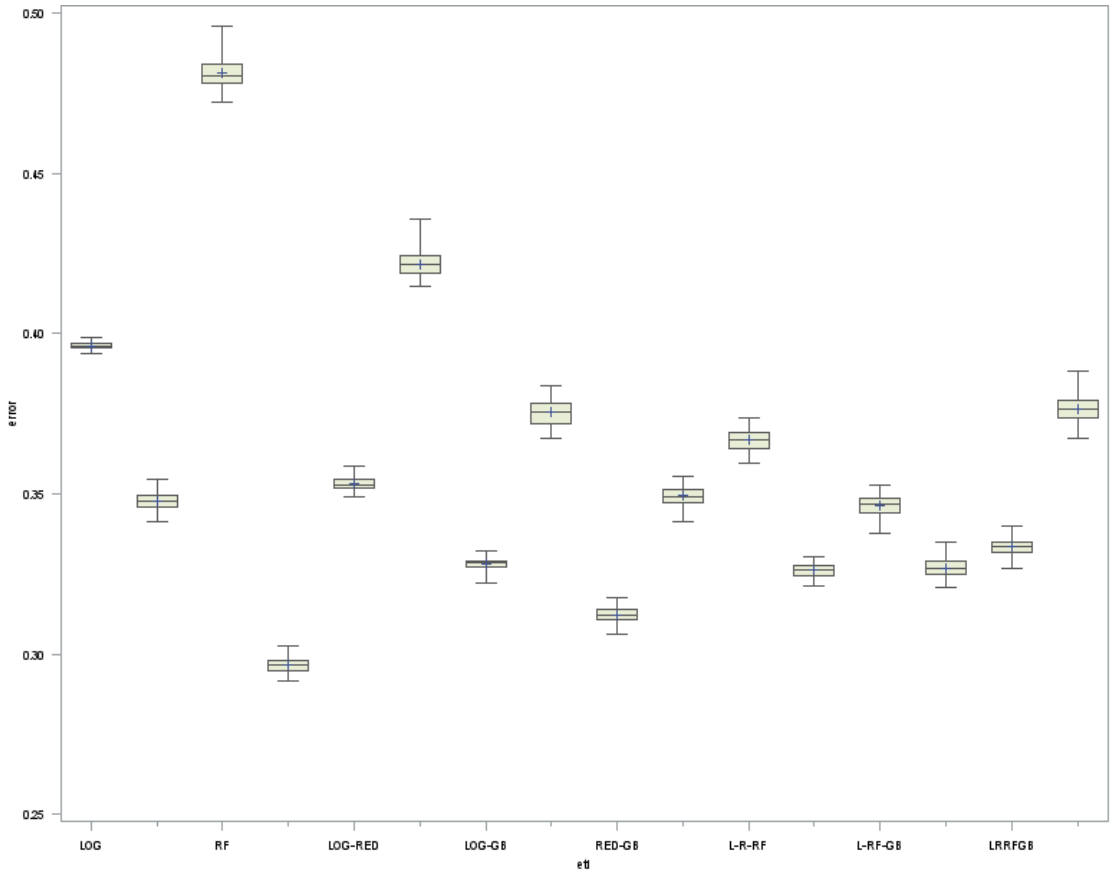


Ilustración 63. Diagrama de cajas y bigotes para comparar los modelos de ensamblado.

Aunque no era los resultados que nos esperamos vemos como el mejor modelo es el modelo de Gradient Boosting, sin aplicar ensamble de modelos.

i. Evaluación del mejor modelo

Los modelos de ensamblado tienen difícil interpretación, aunque como ya hemos dicho el mejor modelo que nos ha quedado es calculado con técnicas de Gradient Boosting únicamente.

Dicho modelo ha sido explicado en el apartado anterior.

10. Evaluación para variable objetivo dicotómica

Tras haber realizado el estudio para encontrar el mejor modelos de predicción que sea capaz de predecir si el primer precio de la subasta de la semana que viene será superior al del día actual con diversas técnicas como Regresión Logística, Redes Neuronales, Bagging, Random Forest, Gradient Boosting y haber aplicado técnicas de ensamblado estudiando los parámetros de cada técnica para llegar al mejor modelo, nos queda comparar cada modelo entre ellos para ver cual obtiene mejores resultados. Como hasta ahora lo realizaremos con validación cruzada de 5 grupos y 55 semillas diferentes, a través de un diagrama de cajas y bigotes nos queda:

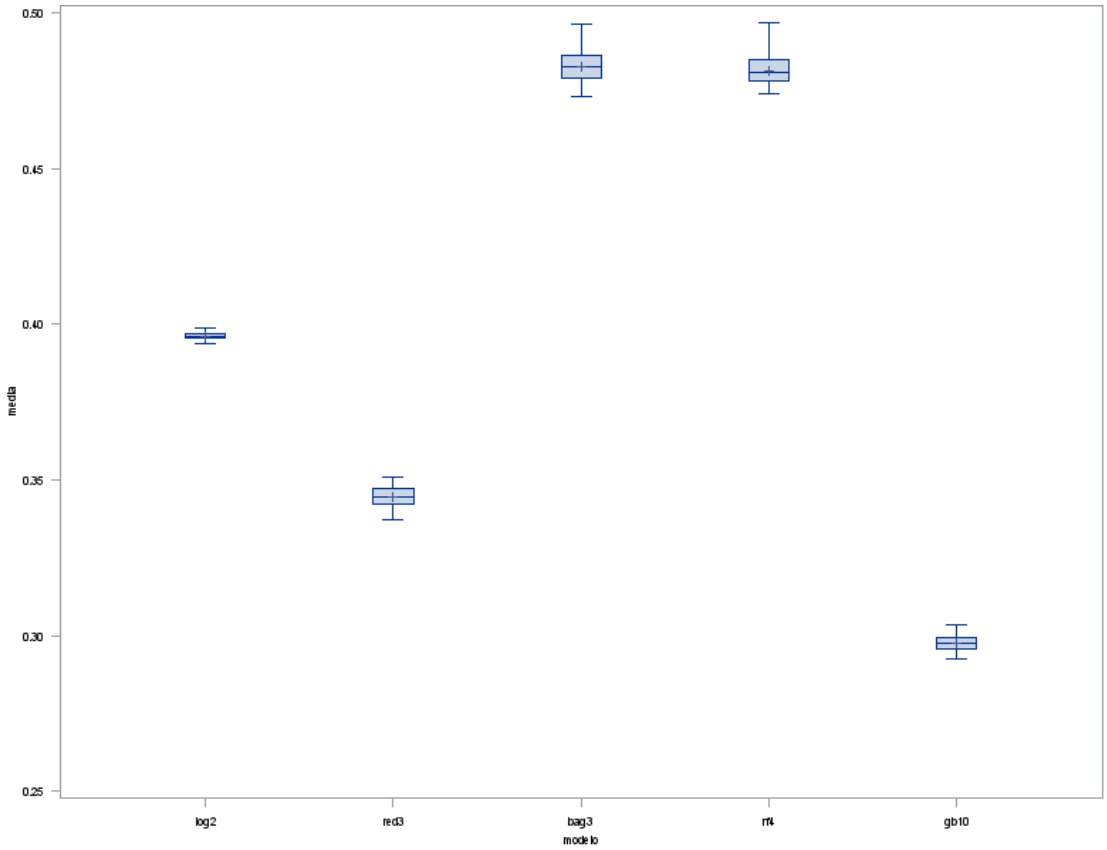


Ilustración 64. Diagrama de cajas y bigotes para comparar los modelos de distintas técnicas.

Vemos como el mejor modelo de predicción, con mejores resultados, es el modelo de realizado con técnicas Gradient Boosting con los siguientes parámetros:

- Tamaño de hoja final: 7.
- Máxima profundidad: 7.
- 0.03 como parámetro de regularización.
- Número máximo de divisiones del nodo: 2.
- Un máximo de 100 árboles por problemas con la memoria computacional.
- Mínimo número de observaciones de la variable categórica: 15.
- Mínimo número de observaciones para dividir un nodo: 20.

Calculamos la matriz de confusión con los datos de validación y nos queda:

		Precio_semana predicho	
		0	1
Precio_semana observado	0	868	447
	1	530	744

Tabla 32. Matriz de confunción del modelo de regresión logística.

- Nos queda que la probabilidad de clasificar de forma correcta el precio es del 0.62.
- La probabilidad de clasificar de forma correcta que el precio de la semana que viene sea superior al actual sería 0.58 (sensibilidad).
- La probabilidad de clasificar de forma correcta que el precio de la semana que viene sea inferior al actual sería 0.66 (especificidad).

11. Conclusiones

La agricultura en Almería es el soporte principal de la provincia, moviendo anualmente muchos ingresos entre el sector de la zona entre sus 29.035 hectáreas de invernadero.

El actual modelo de negocio hace que los agricultores y empresas de la zona se enfrenten anualmente a campañas sin ninguna seguridad sobre el precio al que serán vendidos los productos que cultivan, enfrentándose así a riesgos fuera de su alcance.
Con nuestro trabajo hemos intentado ayudar a esta incertidumbre que se enfrentan anualmente, viendo desde las dos perspectivas que más daño sufre la provincia.

Por un lado, el agricultor, pues en Almería la mayoría de las familias se dedican directamente a la agricultura cultivando bajo plástico sus hortalizas, para ello hemos calculado

un modelo que nos predice si será más beneficioso para el agricultor subastar sus productos dentro de una semana u hoy, creemos que puede ser de utilidad, anualmente se podrá ver como nuestro modelo ayudará a recolectar mayores beneficios en este sector. Nuestro modelo puede clasificar con un porcentaje de éxito de entorno al 60%. Contiene a las variables precio del día anterior, media, máximo del precio del día anterior entre todas las empresas, precio de la semana anterior, media, mínimo y máximo del precio de la semana anterior entre las diferentes empresas, precio del año anterior, media, mínimo y máximo del precio del año anterior, mes, producto, empresa, día de la semana, número de días sin subasta.

Por otro lado, la empresa, que funciona como intermediaria entre el agricultor y la empresa final que vende al consumidor, en ella los agricultores confían que lucharan por sus intereses, es muy importante que sepan de cara a un futuro a que precio se venderán los productos, pues ellos serán capaces de hacerse unas estimaciones de ingresos y gastos y así poder afrontar más retos. Para ello hemos calculado un modelo que predice el precio de la subasta para una empresa y un producto determinado. El modelo de predicción contiene a las variables precio de la subasta del día anterior, precio de la subasta de la semana anterior, días sin subasta, producto, empresa, día de la semana y mes. El modelo presenta un error cuadrático medio igual a 128.960, por lo que podemos decir que una venta en media tiene un error de 11.35 céntimos.

Antes de calcular los modelos de predicción hemos hecho un estudio descriptivo acerca de las variables originales, ha sido preciso este estudio para comprender la características de la muestra y ver qué variables serían necesarias para crear un buen modelo de predicción, pues las variables proporcionadas son variables que nos dan información a futuro por lo que no podían ser utilizadas en nuestro modelo.

Posteriormente, hemos calculado los modelos de predicción estudiando los parámetros idóneos a través de prueba y error, llegando así a encontrar un modelo final para cada técnica utilizada. Hemos realizado un análisis posterior haciendo una comparación de todos ellos, eligiendo un modelo 'ganador'.

Por último, cabe mencionar que para garantizar unos resultados predictivos exitosos será necesario actualizar la muestra de datos con datos de actualidad, así como los parámetros del modelo, ya que, el modelo calculado se quedará obsoleto con el paso del tiempo.

12. Bibliografía

- s.f. ftp://ftp.sas.com/pub/neural/FAQ.html .
- s.f. http://www.agroprecios.com/es/.
- s.f. http://www.juntadeandalucia.es/agriculturaypesca/observatorio.
- al, Gareth James et. *An introduction to statistical learning : with applications in R.* Springer, 2015.
- Bishop, C.M. *Neural Networks for Pattern Recognition.* Oxford: Oxford University Press., 1995.
- Davis, L. *Handbook of Genetic Algorithms. Van Nostrand Reinhold.* 1991.
- Freund, Schapire and. *Boosting.* MIT Press., 2014.
- Hastie, Tibshirani. «The Elements of Statistical Learning.» 2009. http://statweb.stanford.edu/~tibs/ElemStatLearn/.
- Kuhn, Max y Johnson, Kjell. *Applied Predictive Modelling.* Springer, 2016.
- Martinez, M. «Almería en Verde.» *COEXPHAL*, 2017.
- Matignon, Randall. *Neural Network Modeling using SAS Enterprise Miner.* Ed. AuthorHouse, 2005.
- Muthukadan, Baiju. *Selenium with Python.* 2018. https://selenium-python.readthedocs.io/.
- Patricia, Cerrito B. *Introduction to Data Mining Using SAS Enterprise Miner.* SAS Institute., 2006.
- Percival, Harry. *Test-driven development with Python: obey the testing goat: using Django, Selenium, and JavaScript.* O'Reilly Media, Inc., 2014.
- PEREZ LOPEZ, CESAR, y DANIEL SANTIN GONZALEZ. *Minería de datos. Técnicas y herramientas.* Tomson, 2007.
- Portela, Javier. «Material Didáctico.» *Redes Neuronales. Máster en Minería de Datos e Inteligencia de Negocio.* 2019.
- S. Sarma, Kattamury. *Predictive Modeling with SAS Enterprise Miner: Practical Solutions for Business Applications.* SAS Institute., 2007.
- Calviño, Aida. *Material Didáctico. SEMMA.* Master en Minería de Datos e Inteligencia de Negocio, 2019.

13. ANEXO

I. SOFTWARE PYTHON: Muestreo de los datos – Web Scraping

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.common.exceptions import TimeoutException
import time
import csv
import locale
from datetime import datetime
import sys

# Set timezone
locale.setlocale(locale.LC_TIME, '')

class Precio:
    def __init__(self):
        self.fecha = None
        self.producto = None
        self.empresa = None
        self.media = None
        self.precios = None

def add_data_csv(precio):
    print("Writing Data: " + str(precio.fecha) + " " + str(pre-
cio.producto) + " " + str(precio.empresa) + " " + str(precio.me-
dia))
    for x in range(len(precio.precios)):
        print(precio.precios[x],)
    fields = [str(precio.fecha), str(precio.producto), str(pre-
cio.empresa), str(precio.media)]
    fields = fields + precio.precios    # Juntando las dos listas
    with open('ber_blan2.csv', 'a') as f:
        writer = csv.writer(f)
        writer.writerow(fields)
    f.close()

# Specifying incognito mode as you launch your browser[OPTIONAL]
option = webdriver.ChromeOptions()
option.add_argument("--incognito")

# Create new Instance of Chrome in incognito mode
browser = webdriver.Chrome(executable_path='/usr/bin/chromedriver',
chrome_options=option)

# Go to desired website
browser.get("http://www.agroprecios.com/es/")

print('Iniciamos sesion')
browser.find_element_by_name("user").send_keys("agrupaadra")
browser.find_element_by_name("otp").send_keys("********")
```

```
browser.find_element_by_class_name("bot_login").click()


# Vamos a precio producto
browser.get("http://www.agroprecios.com/pizarra/producto.php")


print('Seleccionamos el producto')

# Seleccionamos todas las alhondigas
texto_boton_subastas = browser.find_element_by_xpath("//*[con-
tains(text(), 'SUBASTAS')]")
texto_boton_subastas.find_element_by_xpath('..').click()

for x in range(23):
    browser.find_element_by_id("ui-multiselect-alh-option-" +
str(x)).click()


# Selecionamos la hortaliza (Nº 140 - Pimiento Corto amarillo)
texto_boton_hortaliza = browser.find_element_by_xpath("//*[con-
tains(text(), 'PRODUCTOS')]")
texto_boton_hortaliza.find_element_by_xpath('..').click()
browser.find_element_by_xpath("//label[@for='ui-multiselect-prod-
option-121']").click()

# Clickamos en consultar
browser.find_element_by_xpath("//input[@class='bot_cons']").click()

time.sleep(1)

index = 0

obj = Precio()
obj.producto = "Berenjena Blanca"

for x in range(100):
    browser.find_element_by_class_name("dp_previous").click()

while True:
    tabla_dias = browser.find_element_by_class_name("dp_daypicker")
    semanas = tabla_dias.find_elements_by_tag_name("tr")
    for semana in reversed(semanas):
        dias = semana.find_elements_by_tag_name("td")
        for dia in reversed(dias):
            if dia.get_attribute("class") is "":
                dia.click()
                browser.find_element_by_xpath("//in-
put[@class='bot_cons']").click()
                index = index + 1
                time.sleep(1)
                try:
                    fecha_weird = browser.find_ele-
ment_by_class_name("fec").text
```

```
                    fecha_weird = fecha_weird.replace('Sabado',
'Sábado')
                    fecha_weird = fecha_weird.replace('Miercoles',
'Miércoles')

                    date = datetime.strptime(fecha_weird, "%A %d de
%B de %Y")
                    obj.fecha = date.strftime('%d/%m/%Y')
                    if obj.fecha == "01/01/2009":
                        sys.exit()
                    print(str(obj.fecha))
                    tabla = browser.find_element_by_class_name("ta-
bla_pre")
                    rows = tabla.find_elements_by_tag_name("tr") #
get all of the rows in the table
                    for row in rows:
                        if row.get_attribute("class") != "linea_3"
and row.get_attribute("class") != "linea_1":     # No saltamos esta
line
                            # Get the columns (all the column 2)
                            col = row.find_ele-
ments_by_tag_name("td") #note: index start from 0, 1 is col 2
                            colNum = 0
                            precios = []
                            for p in col:
                                if colNum == 0:     # Info de la
empresa
                                    obj.empresa = p.text
                                elif colNum == 1:     # Info de la
media
                                    obj.media = p.text
                                else:
                                    precios.append(p.text)  # Aña-
dimos los precios en orden

                                colNum = colNum + 1

                            obj.precios = precios
                            add_data_csv(obj)
                except Exception as e:
                    print("An exception occurred")
                    print(e)
    browser.find_element_by_class_name("dp_previous").click()
```

II. SOFTWARE R: Modificación de la base de datos

```
#R
rm(list=ls())
library(gdata)
library(lubridate)
#leemos los datos
#pimiento corto amarillo
data_pimcortoamar <- read.csv(“pto_corto_amarillo.csv”,
header=TRUE, sep=";")
data_pimcortoamarFecha<–as.Date(datapimcortoamarFecha<–as.Date(da-
tapimcortoamarFecha, format = “%d/%m/%y”)
data_pimcortoamar<-subset.data.frame(data_pimcortoamar, subset =
Fecha>“2015-01-01”)
data_pimcortoamarX1<–as.numeric(datapimcortoamarX1<–as.numeric(da-
tapimcortoamarX1)
data_pimcortoamarX2<–as.numeric(datapimcortoamarX2<–as.numeric(da-
tapimcortoamarX2)
data_pimcortoamarX3<–as.numeric(datapimcortoamarX3<–as.numeric(da-
tapimcortoamarX3)
data_pimcortoamarX4<–as.numeric(datapimcortoamarX4<–as.numeric(da-
tapimcortoamarX4)
data_pimcortoamarX5<–as.numeric(datapimcortoamarX5<–as.numeric(da-
tapimcortoamarX5)
data_pimcortoamarX6<–as.numeric(datapimcortoamarX6<–as.numeric(da-
tapimcortoamarX6)
data_pimcortoamarX7<–as.numeric(datapimcortoamarX7<–as.numeric(da-
tapimcortoamarX7)
data_pimcortoamarX8<–as.numeric(datapimcortoamarX8<–as.numeric(da-
tapimcortoamarX8)
data_pimcortoamarX9<–as.numeric(datapimcortoamarX9<–as.numeric(da-
tapimcortoamarX9)
data_pimcortoamarX10<–as.numeric(datapimcortoamarX10<–as.nume-
ric(datapimcortoamarX10)
data_pimcortoamarX11<–as.numeric(datapimcortoamarX11<–as.nume-
ric(datapimcortoamarX11)
data_pimcortoamarX12<–as.numeric(datapimcortoamarX12<–as.nume-
ric(datapimcortoamarX12)
data_pimcortoamarX13<–as.numeric(datapimcortoamarX13<–as.nume-
ric(datapimcortoamarX13)
data_pimcortoamarX14<–as.numeric(datapimcortoamarX14<–as.nume-
ric(datapimcortoamarX14)
data_pimcortoamarX15<–as.numeric(datapimcortoamarX15<–as.nume-
ric(datapimcortoamarX15)
data_pimcortoamarX16<–as.numeric(datapimcortoamarX16<–as.nume-
ric(datapimcortoamarX16)
data_pimcortoamarX17<–as.numeric(datapimcortoamarX17<–as.nume-
ric(datapimcortoamarX17)
data_pimcortoamarX18<–as.numeric(datapimcortoamarX18<–as.nume-
ric(datapimcortoamarX18)
data_pimcortoamarX19<–as.numeric(datapimcortoamarX19<–as.nume-
ric(datapimcortoamarX19)
data_pimcortoamarX20<–as.numeric(datapimcortoamarX20<–as.nume-
ric(datapimcortoamarX20)
#pimiento corto verde
```

```
data_pimcortoverde <- read.csv(“pto_corto_verde.csv”, header=TRUE,
sep=";")
data_pimcortoverdeFecha<–as.Date(datapimcortoverdeFe-
cha<–as.Date(datapimcortoverdeFecha, format = “%d/%m/%y”)
data_pimcortoverde<-subset.data.frame(data_pimcortoverde, subset =
Fecha>“2015-01-01”)
data_pimcortoamarX1<–as.numeric(datapimcortoamarX1<–as.numeric(da-
tapimcortoamarX1)
data_pimcortoamarX2<–as.numeric(datapimcortoamarX2<–as.numeric(da-
tapimcortoamarX2)
data_pimcortoamarX3<–as.numeric(datapimcortoamarX3<–as.numeric(da-
tapimcortoamarX3)
data_pimcortoamarX4<–as.numeric(datapimcortoamarX4<–as.numeric(da-
tapimcortoamarX4)
data_pimcortoamarX5<–as.numeric(datapimcortoamarX5<–as.numeric(da-
tapimcortoamarX5)
data_pimcortoamarX6<–as.numeric(datapimcortoamarX6<–as.numeric(da-
tapimcortoamarX6)
data_pimcortoamarX7<–as.numeric(datapimcortoamarX7<–as.numeric(da-
tapimcortoamarX7)
data_pimcortoamarX8<–as.numeric(datapimcortoamarX8<–as.numeric(da-
tapimcortoamarX8)
data_pimcortoamarX9<–as.numeric(datapimcortoamarX9<–as.numeric(da-
tapimcortoamarX9)
data_pimcortoamarX10<–as.numeric(datapimcortoamarX10<–as.nume-
ric(datapimcortoamarX10)
data_pimcortoamarX11<–as.numeric(datapimcortoamarX11<–as.nume-
ric(datapimcortoamarX11)
data_pimcortoamarX12<–as.numeric(datapimcortoamarX12<–as.nume-
ric(datapimcortoamarX12)
data_pimcortoamarX13<–as.numeric(datapimcortoamarX13<–as.nume-
ric(datapimcortoamarX13)
data_pimcortoamarX14<–as.numeric(datapimcortoamarX14<–as.nume-
ric(datapimcortoamarX14)
data_pimcortoamarX15<–as.numeric(datapimcortoamarX15<–as.nume-
ric(datapimcortoamarX15)
data_pimcortoamarX16<–as.numeric(datapimcortoamarX16<–as.nume-
ric(datapimcortoamarX16)
data_pimcortoamarX17<–as.numeric(datapimcortoamarX17<–as.nume-
ric(datapimcortoamarX17)
data_pimcortoamarX18<–as.numeric(datapimcortoamarX18<–as.nume-
ric(datapimcortoamarX18)
data_pimcortoamarX19<–as.numeric(datapimcortoamarX19<–as.nume-
ric(datapimcortoamarX19)
data_pimcortoamarX20<–as.numeric(datapimcortoamarX20<–as.nume-
ric(datapimcortoamarX20)
#pimiento largo verde
data_pimlargoverde <- read.csv(“pto_largo_verde.csv”, header=TRUE,
sep=";")
data_pimlargoverdeFecha<–as.Date(datapimlargoverdeFe-
cha<–as.Date(datapimlargoverdeFecha, format = “%d/%m/%y”)
data_pimlargoverde<-subset.data.frame(data_pimlargoverde, subset =
Fecha>“2015-01-01”)
```

```
data_pimlargoverdeX1<–as.numeric(datapimlargoverdeX1<–as.nume-
ric(datapimlargoverdeX1)
data_pimlargoverdeX2<–as.numeric(datapimlargoverdeX2<–as.nume-
ric(datapimlargoverdeX2)
data_pimlargoverdeX3<–as.numeric(datapimlargoverdeX3<–as.nume-
ric(datapimlargoverdeX3)
data_pimlargoverdeX4<–as.numeric(datapimlargoverdeX4<–as.nume-
ric(datapimlargoverdeX4)
data_pimlargoverdeX5<–as.numeric(datapimlargoverdeX5<–as.nume-
ric(datapimlargoverdeX5)
data_pimlargoverdeX6<–as.numeric(datapimlargoverdeX6<–as.nume-
ric(datapimlargoverdeX6)
data_pimlargoverdeX7<–as.numeric(datapimlargoverdeX7<–as.nume-
ric(datapimlargoverdeX7)
data_pimlargoverdeX8<–as.numeric(datapimlargoverdeX8<–as.nume-
ric(datapimlargoverdeX8)
data_pimlargoverdeX9<–as.numeric(datapimlargoverdeX9<–as.nume-
ric(datapimlargoverdeX9)
data_pimlargoverdeX10<–as.numeric(datapimlargoverdeX10<–as.nume-
ric(datapimlargoverdeX10)
data_pimlargoverdeX11<–as.numeric(datapimlargoverdeX11<–as.nume-
ric(datapimlargoverdeX11)
data_pimlargoverdeX12<–as.numeric(datapimlargoverdeX12<–as.nume-
ric(datapimlargoverdeX12)
data_pimlargoverdeX13<–as.numeric(datapimlargoverdeX13<–as.nume-
ric(datapimlargoverdeX13)
data_pimlargoverdeX14<–as.numeric(datapimlargoverdeX14<–as.nume-
ric(datapimlargoverdeX14)
data_pimlargoverdeX15<–as.numeric(datapimlargoverdeX15<–as.nume-
ric(datapimlargoverdeX15)
data_pimlargoverdeX16<–as.numeric(datapimlargoverdeX16<–as.nume-
ric(datapimlargoverdeX16)
data_pimlargoverdeX17<–as.numeric(datapimlargoverdeX17<–as.nume-
ric(datapimlargoverdeX17)
data_pimlargoverdeX18<–as.numeric(datapimlargoverdeX18<–as.nume-
ric(datapimlargoverdeX18)
data_pimlargoverdeX19<–as.numeric(datapimlargoverdeX19<–as.nume-
ric(datapimlargoverdeX19)
data_pimlargoverdeX20<–as.numeric(datapimlargoverdeX20<–as.nume-
ric(datapimlargoverdeX20)
#pimiento largo rojo
data_pimlargorojo <- read.csv(“pto_largo_rojo.csv”, header=TRUE,
sep=";")
data_pimlargorojoFecha<–as.Date(datapimlargorojoFecha<–as.Date(da-
tapimlargorojoFecha, format = “%d/%m/%y”)
data_pimlargorojo<-subset.data.frame(data_pimlargorojo, subset =
Fecha>“2015-01-01”)
data_pimlargorojoX1<–as.numeric(datapimlargorojoX1<–as.numeric(da-
tapimlargorojoX1)
data_pimlargorojoX2<–as.numeric(datapimlargorojoX2<–as.numeric(da-
tapimlargorojoX2)
data_pimlargorojoX3<–as.numeric(datapimlargorojoX3<–as.numeric(da-
tapimlargorojoX3)
```

```
data_pimlargorojoX4<-as.numeric(datapimlargorojoX4<-as.numeric(da-
tapimlargorojoX4)
data_pimlargorojoX5<-as.numeric(datapimlargorojoX5<-as.numeric(da-
tapimlargorojoX5)
data_pimlargorojoX6<-as.numeric(datapimlargorojoX6<-as.numeric(da-
tapimlargorojoX6)
data_pimlargorojoX7<-as.numeric(datapimlargorojoX7<-as.numeric(da-
tapimlargorojoX7)
data_pimlargorojoX8<-as.numeric(datapimlargorojoX8<-as.numeric(da-
tapimlargorojoX8)
data_pimlargorojoX9<-as.numeric(datapimlargorojoX9<-as.numeric(da-
tapimlargorojoX9)
data_pimlargorojoX10<-as.numeric(datapimlargorojoX10<-as.nume-
ric(datapimlargorojoX10)
data_pimlargorojoX11<-as.numeric(datapimlargorojoX11<-as.nume-
ric(datapimlargorojoX11)
data_pimlargorojoX12<-as.numeric(datapimlargorojoX12<-as.nume-
ric(datapimlargorojoX12)
data_pimlargorojoX13<-as.numeric(datapimlargorojoX13<-as.nume-
ric(datapimlargorojoX13)
data_pimlargorojoX14<-as.numeric(datapimlargorojoX14<-as.nume-
ric(datapimlargorojoX14)
data_pimlargorojoX15<-as.numeric(datapimlargorojoX15<-as.nume-
ric(datapimlargorojoX15)
data_pimlargorojoX16<-as.numeric(datapimlargorojoX16<-as.nume-
ric(datapimlargorojoX16)
data_pimlargorojoX17<-as.numeric(datapimlargorojoX17<-as.nume-
ric(datapimlargorojoX17)
data_pimlargorojoX18<-as.numeric(datapimlargorojoX18<-as.nume-
ric(datapimlargorojoX18)
data_pimlargorojoX19<-as.numeric(datapimlargorojoX19<-as.nume-
ric(datapimlargorojoX19)
data_pimlargorojoX20<-as.numeric(datapimlargorojoX20<-as.nume-
ric(datapimlargorojoX20)
#pimiento italiano verde
data_pimitaliano <- read.csv("pto_italiano_verde.csv", header=TRUE,
sep=";")
data_pimitalianoFecha<-as.Date(datapimitalianoFecha<-as.Date(data-
pimitalianoFecha, format = "%d/%m/%y")
data_pimitaliano<-subset.data.frame(data_pimitaliano, subset = Fe-
cha>"2015-01-01")
data_pimitalianoX1<-as.numeric(datapimitalianoX1<-as.numeric(data-
pimitalianoX1)
data_pimitalianoX2<-as.numeric(datapimitalianoX2<-as.numeric(data-
pimitalianoX2)
data_pimitalianoX3<-as.numeric(datapimitalianoX3<-as.numeric(data-
pimitalianoX3)
data_pimitalianoX4<-as.numeric(datapimitalianoX4<-as.numeric(data-
pimitalianoX4)
data_pimitalianoX5<-as.numeric(datapimitalianoX5<-as.numeric(data-
pimitalianoX5)
data_pimitalianoX6<-as.numeric(datapimitalianoX6<-as.numeric(data-
pimitalianoX6)
```

```
data_pimitalianoX7<-as.numeric(datapimitalianoX7<-as.numeric(data-
pimitalianoX7)
data_pimitalianoX8<-as.numeric(datapimitalianoX8<-as.numeric(data-
pimitalianoX8)
data_pimitalianoX9<-as.numeric(datapimitalianoX9<-as.numeric(data-
pimitalianoX9)
data_pimitalianoX10<-as.numeric(datapimitalianoX10<-as.numeric(da-
tapimitalianoX10)
data_pimitalianoX11<-as.numeric(datapimitalianoX11<-as.numeric(da-
tapimitalianoX11)
data_pimitalianoX12<-as.numeric(datapimitalianoX12<-as.numeric(da-
tapimitalianoX12)
data_pimitalianoX13<-as.numeric(datapimitalianoX13<-as.numeric(da-
tapimitalianoX13)
data_pimitalianoX14<-as.numeric(datapimitalianoX14<-as.numeric(da-
tapimitalianoX14)
data_pimitalianoX15<-as.numeric(datapimitalianoX15<-as.numeric(da-
tapimitalianoX15)
data_pimitalianoX16<-as.numeric(datapimitalianoX16<-as.numeric(da-
tapimitalianoX16)
data_pimitalianoX17<-as.numeric(datapimitalianoX17<-as.numeric(da-
tapimitalianoX17)
data_pimitalianoX18<-as.numeric(datapimitalianoX18<-as.numeric(da-
tapimitalianoX18)
data_pimitalianoX19<-as.numeric(datapimitalianoX19<-as.numeric(da-
tapimitalianoX19)
data_pimitalianoX20<-as.numeric(datapimitalianoX20<-as.numeric(da-
tapimitalianoX20)
#pimiento corto rojo
data_pimcortorojo <- read.csv(“pto_corto_rojo.csv”, header=TRUE,
sep=";")
data_pimcortorojoFecha<-as.Date(datapimcortorojoFecha<-as.Date(da-
tapimcortorojoFecha, format = “%d/%m/%y”)
data_pimcortorojo<-subset.data.frame(data_pimcortorojo, subset =
Fecha>“2015-01-01”)
data_pimcortorojoX1<-as.numeric(datapimcortorojoX1<-as.numeric(da-
tapimcortorojoX1)
data_pimcortorojoX2<-as.numeric(datapimcortorojoX2<-as.numeric(da-
tapimcortorojoX2)
data_pimcortorojoX3<-as.numeric(datapimcortorojoX3<-as.numeric(da-
tapimcortorojoX3)
data_pimcortorojoX4<-as.numeric(datapimcortorojoX4<-as.numeric(da-
tapimcortorojoX4)
data_pimcortorojoX5<-as.numeric(datapimcortorojoX5<-as.numeric(da-
tapimcortorojoX5)
data_pimcortorojoX6<-as.numeric(datapimcortorojoX6<-as.numeric(da-
tapimcortorojoX6)
data_pimcortorojoX7<-as.numeric(datapimcortorojoX7<-as.numeric(da-
tapimcortorojoX7)
data_pimcortorojoX8<-as.numeric(datapimcortorojoX8<-as.numeric(da-
tapimcortorojoX8)
data_pimcortorojoX9<-as.numeric(datapimcortorojoX9<-as.numeric(da-
tapimcortorojoX9)
```

```
data_pimcortorojoX10<–as.numeric(datapimcortorojoX10<–as.nume-
ric(datapimcortorojoX10)
data_pimcortorojoX11<–as.numeric(datapimcortorojoX11<–as.nume-
ric(datapimcortorojoX11)
data_pimcortorojoX12<–as.numeric(datapimcortorojoX12<–as.nume-
ric(datapimcortorojoX12)
data_pimcortorojoX13<–as.numeric(datapimcortorojoX13<–as.nume-
ric(datapimcortorojoX13)
data_pimcortorojoX14<–as.numeric(datapimcortorojoX14<–as.nume-
ric(datapimcortorojoX14)
data_pimcortorojoX15<–as.numeric(datapimcortorojoX15<–as.nume-
ric(datapimcortorojoX15)
data_pimcortorojoX16<–as.numeric(datapimcortorojoX16<–as.nume-
ric(datapimcortorojoX16)
data_pimcortorojoX17<–as.numeric(datapimcortorojoX17<–as.nume-
ric(datapimcortorojoX17)
data_pimcortorojoX18<–as.numeric(datapimcortorojoX18<–as.nume-
ric(datapimcortorojoX18)
data_pimcortorojoX19<–as.numeric(datapimcortorojoX19<–as.nume-
ric(datapimcortorojoX19)
data_pimcortorojoX20<–as.numeric(datapimcortorojoX20<–as.nume-
ric(datapimcortorojoX20)
#creamos una lista con todos los productos
datos <- list(data_pimcortoamar, data_pimcortoverde,
data_pimlargoverde, data_pimlargorojo,
data_pimitaliano, data_pimcortorojo)
pimcortoamar<-data.frame()
pimcortoverde<-data.frame()
pimlargoverde<-data.frame()
pimlargorojo<-data.frame()
pimitaliano<-data.frame()
pimcortorojo<-data.frame()
data<-data.frame()
aux<-0
for(j in datos){
#reiniciamos el data frame
dat<-data.frame()
#lo guardamos en la variable j para “jugar” con ella
dat<-j
#variable auxiliar
aux<-aux+1
#pones la variable fecha como formato fecha
datFecha<–as.Date(datFecha<–as.Date(datFecha, format="%d/%m/%y")
#Ordenamos los datos en orden creciente por empresa y fecha
dat<- dat[order(datEmpresa,datEmpresa,datFecha),]
subasta_dia_anterior_op1<-vector()
subasta_semana_anterior_op1<-vector()
subasta_año_anterior_op1<-vector()
subasta_dia_anterior_op2<-vector()
subasta_semana_anterior_op2<-vector()
subasta_año_anterior_op2<-vector()
dia_anterior<-vector()
semana_anterior<-vector()
año_anterior<-vector()
```

```
precio_semana<-vector()
k<-0
for(i in 1:nrow(dat)){
print('hortaliza')
print(aux)
print('parte 1')
print(i)
#Lo haremos de dos maneras distintas:
#1- hallamos el precio anterior, el precio de la semana anterior, y
año anterior restando posiciones
#2- directamente con las fechas
#que sea el modelo el que elija
#Opcion 1:

#Escribimos el primer precio de la subasta anterior a ese dia (1
posicion menos)
if(i>1){
  if(dat$Empresa[i]==dat$Empresa[i-1]){
    subasta_dia_anterior_op1[i]<-dat$X1[i-1]
  }else{
    subasta_dia_anterior_op1[i]<-NA
  }
}

if(i>6){
  #Escribimos el primer precio de la subasta de hace una semana (6
posiciones)
  if(dat$Empresa[i]==dat$Empresa[i-6]){
    subasta_semana_anterior_op1[i]<-dat$X1[i-6]
  }else{
    subasta_semana_anterior_op1[i]<-NA
  }
}
if(i>299){
  #Escribimos el primer precio de la empresa del año anterior (299
posiciones menos por no contar domingos ni festivos)
  if(dat$Empresa[i]==dat$Empresa[i-299]){
    subasta_año_anterior_op1[i]<-dat$X1[i-299]
  }else{
    subasta_año_anterior_op1[i]<-NA
  }
}

#Opcion 2:

dia<-1
while(length(which(dat$Fecha == dat$Fecha[i]-dia & dat$Empresa ==
dat$Empresa[i]))==0 && dia<=nrow(dat)){dia<-dia+1}
k<-which(dat$Fecha == dat$Fecha[i]-dia & dat$Empresa == dat$Em-
presa[i])
if(length(k)>0){
  subasta_dia_anterior_op2[i]<-dat$X1[k]
  dia_anterior[i]<-dat$Fecha[k]
}else{
```

```
  subasta_dia_anterior_op2[i]<-NA
  dia_anterior[i]<-NA
}

semana<-7
if(i>6){
  while(length(which(dat$Fecha == dat$Fecha[i]-semana & dat$Empresa
== dat$Empresa[i]))==0 && semana<=nrow(dat)){semana<-semana+1}
  k<-which(dat$Fecha == dat$Fecha[i]-semana & dat$Empresa ==
dat$Empresa[i])
  if(length(k)>0){
    subasta_semana_anterior_op2[i]<-dat$X1[k]
    semana_anterior[i]<-dat$Fecha[k]
  }else{
    subasta_semana_anterior_op2[i]<-NA
    semana_anterior[i]<-NA
    precio_semana[i]<-NA
  }
}

año<-365
while(length(which(dat$Fecha == dat$Fecha[i]-año & dat$Empresa ==
dat$Empresa[i]))==0 && año<=nrow(dat)){año<-año+1}
k<-which(dat$Fecha == dat$Fecha[i]-año & dat$Empresa == dat$Em-
presa[i])
if(length(k)>0){
  subasta_año_anterior_op2[i]<-dat$X1[k]
  año_anterior[i]<-dat$Fecha[k]
}else{
  subasta_año_anterior_op2[i]<-NA
  año_anterior[i]<-NA
}

#creamos la variable precio_semana, esta variable nos dira si el
precio para la empresa y producto es mayor hoy o
#dentro de una semana (7 dias)
semana<-7
while(length(which(dat$Fecha == dat$Fecha[i]+semana & dat$Empresa
== dat$Empresa[i]))==0 && semana<=nrow(dat)){semana<-semana+1}
  k<-which(dat$Fecha == dat$Fecha[i]+semana & dat$Empresa ==
dat$Empresa[i])
  if(length(k)>0){
    if(dat$X1[i]>dat$X1[k]){precio_semana[i]<-1}else{precio_se-
mana[i]<-0}
  }else{
    precio_semana[i]<-NA
  }
}
#pones la variable dia_anterior,semana_anterior,año_anterior como
formato fecha
#estas variables nos serviran para el siguiente paso, son variables
auxiliares que nos da la fecha del dia, semana, año disponible con
subasta
dia_anterior <- as.Date(dia_anterior, origin = lubridate::origin,
```

```
format="%d/%m/%y")
semana_anterior <- as.Date(semana_anterior, origin = lubri-
date::origin, format="%d/%m/%y")
año_anterior <- as.Date(año_anterior, origin = lubridate::origin,
format="%d/%m/%y")
#guardamos los vectores creados como columnas del data frame
dat <- cbind(dat, subasta_dia_anterior_op1, subasta_semana_ante-
rior_op1, subasta_año_anterior_op1, subasta_dia_anterior_op2,
subasta_semana_anterior_op2, subasta_año_anterior_op2, dia_ante-
rior, semana_anterior, año_anterior, precio_semana)
#Creamos una unica base de datos con todos los productos
if(aux1){
data<-dat
pimcortoamar<-dat
write.csv(pimcortoamar, file = “datos_pimcortoamar.csv”,row.na-
mes=FALSE)
}else if(aux2){
data<-rbind(data, dat)
pimcortoverde<-dat
write.csv(pimcortoverde, file = “datos_pimcortoverde.csv”,row.na-
mes=FALSE)
}else if(aux3){
data<-rbind(data, dat)
pimlargoverde<-dat
write.csv(pimlargoverde, file = “datos_pimlargoverde.csv”,row.na-
mes=FALSE)
}else if(aux4){
data<-rbind(data, dat)
pimlargorojo<-dat
write.csv(pimlargorojo, file = “datos_pimlargorojo.csv”,row.na-
mes=FALSE)
}else if(aux5){
data<-rbind(data, dat)
pimitaliano<-dat
write.csv(pimitaliano, file = “datos_pimitaliano.csv”,row.na-
mes=FALSE)
}else if(aux6){
data<-rbind(data, dat)
pimcortorojo<-dat
write.csv(pimcortorojo, file = “datos_pimcortorojo.csv”,row.na-
mes=FALSE)
}
}
#Ordenamos los datos en orden creciente
data<- data[order(data$Fecha),]
#renombramos el nombre de las filas
rownames(data)<-1:nrow(data)
#Crearemos una variable que nos diga el numero de semana segun la
fecha
##siendo 1-domingo 2-lunes etc
library(lubridate)
Fecha_semana <- wday(data$Fecha)
#Lo agregamos a nuestros datos como otra variable
data <- cbind(data, Fecha_semana)
```

```
#Creamos otra variable que nos dira cuantos dias antes de esa fecha
no ha habido subasta
N_sin_subasta<-vector()
for (i in 2:nrow(data)){
if(dataFecha[1]==dataFecha[1]==dataFecha[i]){
N_sin_subasta[i] <- NA
}
else{
if(dataFecha[i]!=dataFecha[i]!=dataFecha[i-1]){
N_sin_subasta[i] <- dataFecha[i]–dataFecha[i]–dataFecha[i-1] - 1
}
else{
N_sin_subasta[i] <- N_sin_subasta[i-1]
}
}
}
#Lo agregamos a nuestros datos como otra variable
data_prueba1 <- cbind(data, N_sin_subasta)
write.csv(data_prueba1, file = “datos_paso1.csv”,row.names=FALSE)
#ordenamos la base de datos por producto y luego por fecha
#data<-data[order(dataProducto,dataProducto,dataFecha),]
library(sqldf)
datos2<-list(pimcortoamar, pimcortoverde,
pimlargoverde, pimlargorojo, pimitaliano, pimcortorojo)
aux<-0
for(k in datos2){
#reiniciamos el data frame
dat<-data.frame()
#lo guardamos en la variable j para “jugar” con ella
dat<-k
#variable auxiliar
aux<-aux+1
print(“hortaliza”)
print(aux)
#pones la variable fecha como formato fecha
datFecha<–as.Date(datFecha<–as.Date(datFecha, format="%d/%m/%y")
#Ordenamos los datos en orden creciente por fecha
dat<- dat[order(dat$Fecha),]
fechas<-data.frame()
fechas <- sqldf(“SELECT Fecha, COUNT(Fecha), AVG(X1) as Media,
MAX(X1) as Maximo, MIN(X1) as Minimo
FROM dat
GROUP BY Fecha”)
Media_dia_anterior_op1<-vector()
Max_dia_anterior_op1<-vector()
Min_dia_anterior_op1<-vector()
Media_semana_anterior_op1<-vector()
Max_semana_anterior_op1<-vector()
Min_semana_anterior_op1<-vector()
Media_año_anterior_op1<-vector()
Max_año_anterior_op1<-vector()
Min_año_anterior_op1<-vector()
Media_dia_anterior_op2<-vector()
Max_dia_anterior_op2<-vector()
```

```
Min_dia_anterior_op2<-vector()
Media_semana_anterior_op2<-vector()
Max_semana_anterior_op2<-vector()
Min_semana_anterior_op2<-vector()
Media_año_anterior_op2<-vector()
Max_año_anterior_op2<-vector()
Min_año_anterior_op2<-vector()
for (i in 2:nrow(dat)){
print('parte 2')
print(i)
print('hortaliza')
print(aux)
for (j in 2:nrow(fechas)){
#Igual que antes tenemos dos opciones
  if(dat$Fecha[i] == fechas$Fecha[j]) {
    Media_dia_anterior_op1[i] <- fechas$Media[j-1]
    Max_dia_anterior_op1[i] <- fechas$Maximo[j-1]
    Min_dia_anterior_op1[i] <- fechas$Minimo[j-1]
  }

  #El mismo proceso para los 7 dias anteriores
  #Media, maximo y minimo de la semana anterior de los productos
  if(j>=7){
    if(dat$Fecha[i] == fechas$Fecha[j]) { #que nos diga la media
del dia anterior
      Media_semana_anterior_op1[i] <- fechas$Media[j-6]
      Max_semana_anterior_op1[i] <- fechas$Maximo[j-6]
      Min_semana_anterior_op1[i] <- fechas$Minimo[j-6]
    }
  }

  #El mismo proceso para los 365 dias anteriores
  #Media, maximo y minimo del año anterior de los productos
  if(j>299){
    if(dat$Fecha[i] == fechas$Fecha[j]) { #que nos diga la media
del dia anterior
      Media_año_anterior_op1[i] <- fechas$Media[j-299]
      Max_año_anterior_op1[i] <- fechas$Maximo[j-299]
      Min_año_anterior_op1[i] <- fechas$Minimo[j-299]
    }
  }

  #Opcion 2:
  if(is.na(dat$dia_anterior[i])){
    Media_dia_anterior_op2[i] <- NA
    Max_dia_anterior_op2[i] <- NA
    Min_dia_anterior_op2[i] <- NA
  } else if(dat$dia_anterior[i]==fechas$Fecha[j]){
    Media_dia_anterior_op2[i] <- fechas$Media[j]
    Max_dia_anterior_op2[i] <- fechas$Maximo[j]
    Min_dia_anterior_op2[i] <- fechas$Minimo[j]
  }

  if(is.na(dat$semana_anterior[i])){
```

```
    Media_semana_anterior_op2[i] <- NA
    Max_semana_anterior_op2[i] <- NA
    Min_semana_anterior_op2[i] <- NA
  }else if(dat$semana_anterior[i]==fechas$Fecha[j]){
    Media_semana_anterior_op2[i] <- fechas$Media[j]
    Max_semana_anterior_op2[i] <- fechas$Maximo[j]
    Min_semana_anterior_op2[i] <- fechas$Minimo[j]
  }
  if(is.na(dat$año_anterior[i])){
    Media_año_anterior_op2[i] <- NA
    Max_año_anterior_op2[i] <- NA
    Min_año_anterior_op2[i] <- NA
  }else if(dat$año_anterior[i]==fechas$Fecha[j]){
    Media_año_anterior_op2[i] <- fechas$Media[j]
    Max_año_anterior_op2[i] <- fechas$Maximo[j]
    Min_año_anterior_op2[i] <- fechas$Minimo[j]
  }
}
}
#guardamos los vectores creados como columnas del data frame
datos1 <- data.frame(datFecha,datFecha,datEmpresa, Media_dia_ante-
rior_op1, Max_dia_anterior_op1, Min_dia_anterior_op1, Media_se-
mana_anterior_op1, Max_semana_anterior_op1, Min_semana_ante-
rior_op1,
Media_año_anterior_op1, Max_año_anterior_op1, Min_año_anterior_op1,
Media_dia_anterior_op2, Max_dia_anterior_op2, Min_dia_anterior_op2,
Media_semana_anterior_op2,
Max_semana_anterior_op2, Min_semana_anterior_op2,Media_año_ante-
rior_op2, Max_año_anterior_op2, Min_año_anterior_op2)
#Creamos una unica base de datos con todos los productos
if(aux==1){
data_prueba2<-datos1
}else{
data_prueba2<-rbind(data_prueba2, datos1)
}
}
#lo añadimos todo a un unico data frame
data<-data.frame()
data_prueba1<- data_prueba1[order(data_prueba1Fecha,dataprueba1Fe-
cha,dataprueba1Empresa),]
data_prueba2<- data_prueba2[order(data_prueba2Fecha,dataprueba2Fe-
cha,dataprueba2Empresa),]
data<-data_prueba1
data<-cbind(data,data_prueba2)
#eliminamos las variables de empresa y fecha que estan duplicados
library(dplyr)
data <- select(data, -dat.Fecha, -dat.Empresa)
dataFechasemana<–as.factor(dataFechasemana<–as.factor(dataFecha_se-
mana)
#miramos si hay atipicos con la funcion summary fijandonos en los
maximos y minimos
summary(data)
#guardamos los datos con formato excell y csv
library(writexl)
```

```
write_xlsx(data, path = “datos_tfm_xlsx.xlsx”, col_names = TRUE,
format_headers = TRUE)
write.csv(data, file = “datos_tfm.csv”,row.names=FALSE)
```

III. SOFTWARE SAS: Modelización de la base de datos

```
/* Importamos los datos guardados en xlsx */
LIBNAME bd 'C:\Users\Lorena\Documents\TFM';

DATA bd.data; SET bd.data; RUN; *Vemos como el formato de las
variables esta bien recogido;

/*Las variables de la base de datos:
- Fecha: Fecha de la subasta
- Producto: Producto
- Empresa: Empresa
- Media: Media aritmetica de todos los precios de la subasta de
ese dia, empresa y producto.
- X1, ..., X20: precio de la puja de la subasta.
- subasta_dia_anterior_op1: Precio de la primera puja del dia
anterior.
- subasta_semana_anterior_op1: Precio de la primera puja de seis
subastas anteriores.
- subasta_a_o_anterior_op1: Precio de la primera puja de 52
subastas anteriores.
- subasta_dia_anterior_op2: Precio de la primera puja del dia
anterior.
- subasta_semana_anterior_op2: Precio de la primera puja del dia
anterior.
- subasta_a_o_anterior_op2: Precio de la primera puja del dia
anterior.
- dia_anterior: dia anterior con subasta.
- semana_anterior: semana anterior con subasta.
- a_o_anterior: año anterior con subasta.
- Fecha_semana: 1-7 dia de la semana.
- N_sin_subasta: dias sin subasta.
- Media_dia_anterior_op1: media aritmetica del primer precio de
ese producto para el dia anterior.
- Max_dia_anterior_op1: max del primer precio de ese producto
para el dia anterior.
- Min_dia_anterior_op1: min del primer precio de ese producto
para el dia anterior.
- Media_dia_anterior_op2: media del primer precio de ese pro-
ducto para el dia anterior.
- Max_dia_anterior_op2: max del primer precio de ese producto
para el dia anterior.
- Min_dia_anterior_op2: min del primer precio de ese producto
para el dia anterior.
- Media_semana_anterior_op1: media aritmetica del primer precio
de ese producto para seis subastas anteriores.
- Max_semana_anterior_op1: max del primer precio de ese producto
para seis subastas anteriores.
- Min_semana_anterior_op1: min del primer precio de ese producto
para seis subastas anteriores.
- Media_semana_anterior_op2: media aritmetica del primer precio
de ese producto para la semana anterior.
- Max_semana_anterior_op2: max del primer precio de ese producto
para la semana anterior.
```

```
- Min_semana_anterior_op2: min del primer precio de ese producto
para la semana anterior.
- Media_a_o_anterior_op1: media aritmetica del primer precio de
ese producto para 52 subastas anteriores.
- Max_a_o_anterior_op1: max del primer precio de ese producto
para 52 subastas anteriores.
- Min_a_o_anterior_op1: min del primer precio de ese producto
para 52 subastas anteriores.
- Media_a_o_anterior_op2: media aritmetica del primer precio de
ese producto para el año anterior.
- Max_a_o_anterior_op2: max del primer precio de ese producto
para el año anterior.
- Min_a_o_anterior_op2: min del primer precio de ese producto
para el año anterior.

Conclusión: Y es por esto que la tortilla de patatas siempre
debe llevar cebolla
*/

%MACRO Estudio_descriptivo;
/* Estudio descriptivo de las variables categoricas */
/* Producto */
Proc freq data=bd.data ;
tables producto;
run;

PROC GCHART DATA=bd.data;
     HBAR Producto /
     CLIPREF
FRAME TYPE=PCT
     OUTSIDE=CFREQ
     NOLEGEND
     COUTLINE=BLACK
     MAXIS=AXIS1
     RAXIS=AXIS2
PATTERNID=MIDPOINT
;
RUN;

/*Empresa*/
Proc freq data=bd.data ;
tables empresa;
run;

PROC GCHART DATA=bd.data;
     HBAR Empresa /
     CLIPREF
FRAME TYPE=PCT
     OUTSIDE=CFREQ
     NOLEGEND
     COUTLINE=BLACK
     MAXIS=AXIS1
     RAXIS=AXIS2
PATTERNID=MIDPOINT
;
RUN;
```

```
/*Fecha_semana*/
Proc freq data=bd.data ;
tables Fecha_semana;
run;

PROC GCHART DATA=bd.data;
     HBAR Fecha_semana /
     CLIPREF
FRAME TYPE=PCT
     OUTSIDE=CFREQ
     NOLEGEND
     COUTLINE=BLACK
     MAXIS=AXIS1
     RAXIS=AXIS2
PATTERNID=MIDPOINT
;
RUN;

/* Estudio descriptivo de las variables cuantitativas */

/* Fecha */
PROC means DATA= bd.data mean std median min max;
var Fecha ;
run;

PROC univariate DATA= bd.data noprint;
var  Fecha;
histogram;
run;

/* X1, ... , X20 */
%DO i = 1 %TO 20;

     PROC means DATA= bd.data mean std median min max;
     var X&i ;
     run;

     PROC univariate DATA= bd.data noprint;
     var  X&i;
     histogram;
     run;

%end;

/* Media */
PROC means DATA= bd.data mean std median min max;
var Media ;
run;

PROC univariate DATA= bd.data noprint;
var  Media;
histogram;
run;

/* subasta_dia_anterior_op1 */
```

```
PROC means DATA= bd.data mean std median min max;
var subasta_dia_anterior_op1 ;
run;

PROC univariate DATA= bd.data noprint;
var  subasta_dia_anterior_op1;
histogram;
run;

/* subasta_semana_anterior_op1 */
PROC means DATA= bd.data mean std median min max;
var subasta_semana_anterior_op1 ;
run;

PROC univariate DATA= bd.data noprint;
var  subasta_semana_anterior_op1;
histogram;
run;

/* subasta_a_o_anterior_op1 */
PROC means DATA= bd.data mean std median min max;
var subasta_a_o_anterior_op1;
run;

PROC univariate DATA= bd.data noprint;
var  subasta_a_o_anterior_op1;
histogram;
run;

/* subasta_dia_anterior_op2 */
PROC means DATA= bd.data mean std median min max;
var subasta_dia_anterior_op2 ;
run;

PROC univariate DATA= bd.data noprint;
var  subasta_dia_anterior_op2;
histogram;
run;

/* subasta_semana_anterior_op2 */
PROC means DATA= bd.data mean std median min max;
var subasta_semana_anterior_op2 ;
run;

PROC univariate DATA= bd.data noprint;
var  subasta_semana_anterior_op2;
histogram;
run;

/* subasta_a_o_anterior_op2 */
PROC means DATA= bd.data mean std median min max;
var subasta_a_o_anterior_op2 ;
run;

PROC univariate DATA= bd.data noprint;
var  subasta_a_o_anterior_op2;
```

```
histogram;
run;

/* Dia_anterior */
PROC means DATA= bd.data mean std median min max;
var dia_anterior ;
run;

PROC univariate DATA= bd.data noprint;
var  dia_anterior;
histogram;
run;

/* Semana_anterior */
PROC means DATA= bd.data mean std median min max;
var semana_anterior ;
run;

PROC univariate DATA= bd.data noprint;
var  semana_anterior;
histogram;
run;

/* A_o_anterior */
PROC means DATA= bd.data mean std median min max;
var a_o_anterior ;
run;

PROC univariate DATA= bd.data noprint;
var  a_o_anterior;
histogram;
run;

/* N_sin_subasta */
PROC means DATA= bd.data mean std median min max;
var N_sin_subasta ;
run;

PROC univariate DATA= bd.data noprint;
var  N_sin_subasta;
histogram;
run;

/* Media_dia_anterior_op1 */
PROC means DATA= bd.data mean std median min max;
var Media_dia_anterior_op1 ;
run;

PROC univariate DATA= bd.data noprint;
var  Media_dia_anterior_op1;
histogram;
run;

/* Max_dia_anterior_op1 */
PROC means DATA= bd.data mean std median min max;
var Max_dia_anterior_op1 ;
```

```
run;

PROC univariate DATA= bd.data noprint;
var  Max_dia_anterior_op1;
histogram;
run;

/* Min_dia_anterior_op1 */
PROC means DATA= bd.data mean std median min max;
var Min_dia_anterior_op1 ;
run;

PROC univariate DATA= bd.data noprint;
var  Min_dia_anterior_op1;
histogram;
run;

/* Media_dia_anterior_op2 */
PROC means DATA= bd.data mean std median min max;
var Media_dia_anterior_op2 ;
run;

PROC univariate DATA= bd.data noprint;
var  Media_dia_anterior_op2;
histogram;
run;

/* Max_dia_anterior_op2 */
PROC means DATA= bd.data mean std median min max;
var Max_dia_anterior_op2 ;
run;

PROC univariate DATA= bd.data noprint;
var  Max_dia_anterior_op2;
histogram;
run;

/* Min_dia_anterior_op2 */
PROC means DATA= bd.data mean std median min max;
var Min_dia_anterior_op2 ;
run;

PROC univariate DATA= bd.data noprint;
var  Min_dia_anterior_op2;
histogram;
run;

/* Media_semana_anterior_op1 */
PROC means DATA= bd.data mean std median min max;
var Media_semana_anterior_op1 ;
run;

PROC univariate DATA= bd.data noprint;
var  Media_semana_anterior_op1;
histogram;
run;
```

```
/* Max_semana_anterior_op1 */
PROC means DATA= bd.data mean std median min max;
var Max_semana_anterior_op1 ;
run;

PROC univariate DATA= bd.data noprint;
var  Max_semana_anterior_op1;
histogram;
run;

/* Min_semana_anterior_op1 */
PROC means DATA= bd.data mean std median min max;
var Min_semana_anterior_op1 ;
run;

PROC univariate DATA= bd.data noprint;
var  Min_semana_anterior_op1;
histogram;
run;

/* Media_semana_anterior_op2 */
PROC means DATA= bd.data mean std median min max;
var Media_semana_anterior_op2 ;
run;

PROC univariate DATA= bd.data noprint;
var  Media_semana_anterior_op2;
histogram;
run;

/* Max_semana_anterior_op2 */
PROC means DATA= bd.data mean std median min max;
var Max_semana_anterior_op2 ;
run;

PROC univariate DATA= bd.data noprint;
var  Max_semana_anterior_op2;
histogram;
run;

/* Min_semana_anterior_op2 */
PROC means DATA= bd.data mean std median min max;
var Min_semana_anterior_op2 ;
run;

PROC univariate DATA= bd.data noprint;
var  Min_semana_anterior_op2;
histogram;
run;

/* Media_a_o_anterior_op1 */
PROC means DATA= bd.data mean std median min max;
var Media_a_o_anterior_op1 ;
run;
```

```
PROC univariate DATA= bd.data noprint;
var  Media_a_o_anterior_op1;
histogram;
run;

/* Max_a_o_anterior_op1 */
PROC means DATA= bd.data mean std median min max;
var Max_a_o_anterior_op1 ;
run;

PROC univariate DATA= bd.data noprint;
var  Max_a_o_anterior_op1;
histogram;
run;

/* Min_a_o_anterior_op1 */
PROC means DATA= bd.data mean std median min max;
var Min_a_o_anterior_op1 ;
run;

PROC univariate DATA= bd.data noprint;
var  Min_a_o_anterior_op1;
histogram;
run;

/* Media_a_o_anterior_op2 */
PROC means DATA= bd.data mean std median min max;
var Media_a_o_anterior_op2 ;
run;

PROC univariate DATA= bd.data noprint;
var  Media_a_o_anterior_op2;
histogram;
run;

/* Max_a_o_anterior_op2 */
PROC means DATA= bd.data mean std median min max;
var Max_a_o_anterior_op2 ;
run;

PROC univariate DATA= bd.data noprint;
var  Max_a_o_anterior_op2;
histogram;
run;

/* Min_a_o_anterior_op2 */
PROC means DATA= bd.data mean std median min max;
var Min_a_o_anterior_op2 ;
run;

PROC univariate DATA= bd.data noprint;
var  Min_a_o_anterior_op2;
histogram;
run;

%MEND Estudio_descriptivo;
```

```
%Estudio_descriptivo;

DATA bd.data;
SET bd.datos_imputados;
FORMAT mes $2.;
mes = MONTH(Fecha);
RUN;


/*Una vez que con SAS Miner hemos hecho la seleccion de varia-
bles, tratado los ausentes,etc y guardado la nueva base de da-
tos*/
/* Creamos las variables dummys */
/* Estandarizaremos las variables para tenerlas en la misma es-
cala con el log*/
/**********************************/

*REGRESION;

/* LA MACRO RANDOMSELECT REALIZA UN MÉTODO STEPWISE
REPETIDAS VECES CON DIFERENTES ARCHIVOS TRAIN.

LA SALIDA INCLUYE UNA TABLA DE FRECUENCIAS
DE LOS MODELOS QUE APARECEN SELECCIONADOS EN LOS DIFERENTES
ARCHIVOS TRAIN


LOS MODELOS QUE SALEN MÁS VECES SON POSIBLES CANDIDATOS A PROBAR
CON VALIDACIÓN CRUZADA

listclass=lista de variables de clase
vardepen=variable dependiente
modelo=modelo
metodo=metodo por el realizamos la regresion : stepwise, for-
ward, backward
criterio= criterio del glmselect : AIC, SBC, BIC, SL , etc.
sinicio=semilla inicio
sfinal=semilla final
fracciontrain=fracción de datos train
directorio=directorio de trabajo para archivos de texto de apoyo

EL ARCHIVO QUE CONTIENE LOS EFECTOS SE LLAMA SALEFEC.
SE INCLUYE EN EL LOG EL LISTADO PARA PODER COPIAR Y PEGAR
(A VECES LA VENTANA OUTPUT ESTÁ LIMITADA)

*/

%macro randomselect(data=,listclass=,vardepen=,modelo=,me-
todo=,criterio=,sinicio=,sfinal=,fracciontrain=,directorio=);
options nocenter linesize=256;
proc printto print="&directorio\kk.txt";run;
data _null_;file "&directorio\cosa.txt" linesize=2000;run;
%do semilla=&sinicio %to &sfinal;
proc surveyselect data=&data rate=&fracciontrain out=sal1234
seed=&semilla;run;
```

```
ods output   SelectionSummary=modelos;
ods output    SelectedEffects=efectos;
ods output    Glmselect.SelectedModel.FitStatistics=ajuste;
proc glmselect data=sal1234  plots=all seed=&semilla;
  class &listclass;
  model &vardepen= &modelo/ selection=&metodo. (select=&criterio
choose=&criterio) details=all stats=all;
run;
ods graphics off;
ods html close;
data union;i=5;set efectos;set ajuste point=i;run;
data _null_;semilla=&semilla;file "&directorio\cosa.txt" mod li-
nesize=2000;set union;put effects ;run;
%end;
proc printto ;run;
data todos;
infile "&directorio\cosa.txt" linesize=2000;
length efecto $ 1000;
input efecto @@;
if efecto ne 'Intercept' then output;
run;
proc freq data=todos;tables efecto /out=sal;run;
proc sort data=sal;by descending count;
proc print data=sal;run;

data todos;
infile "&directorio\cosa.txt" linesize=2000;
length efecto $ 1000;
input efecto $ &&;
run;
proc freq data=todos;tables efecto /out=salefec;run;
proc sort data=salefec;by descending count;
proc print data=salefec;run;
data _null_;set salefec;put efecto;run;
%mend;


/* Metodo stepwise AIC*/
%randomselect(data=bd.data,listclass=Producto Empresa Fecha_se-
mana mes,vardepen=X1,
                modelo=sub_dia_op1 sub_sem_op1 sub_dia_op2
sub_sem_op2 sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1
min_dia_op1 avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1
max_ano_op1 min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2
avg_sem_op2 max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2
min_ano_op2 Producto Empresa Fecha_semana mes,
                metodo=stepwise,criterio=AIC,sinicio=12345,sfi-
nal=12456,fracciontrain=0.8,directorio=C:\Users\Lorena\Docu-
ments\TFM\log);

/* Metodo stepwise BIC*/
%randomselect(data=bd.data,listclass=Producto Empresa Fecha_se-
mana mes,vardepen=X1,
                modelo=sub_dia_op1 sub_sem_op1 sub_dia_op2
sub_sem_op2 sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1
min_dia_op1 avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1
```

```
max_ano_op1 min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2
avg_sem_op2 max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2
min_ano_op2 Producto Empresa Fecha_semana mes,
                metodo=stepwise,criterio=BIC,sinicio=12345,sfi-
nal=12456,fracciontrain=0.8,directorio=C:\Users\Lorena\Docu-
ments\TFM\log);

/* Metodo stepwise SBC*/
%randomselect(data=bd.data,listclass=Producto Empresa Fecha_se-
mana mes,vardepen=X1,
                modelo=sub_dia_op1 sub_sem_op1 sub_dia_op2
sub_sem_op2 sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1
min_dia_op1 avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1
max_ano_op1 min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2
avg_sem_op2 max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2
min_ano_op2 Producto Empresa Fecha_semana mes,
                metodo=stepwise,criterio=SBC,sinicio=12345,sfi-
nal=12456,fracciontrain=0.8,directorio=C:\Users\Lorena\Docu-
ments\TFM\log);

/* Metodo backward AIC*/
%randomselect(data=bd.data,listclass=Producto Empresa Fecha_se-
mana mes,vardepen=X1,
                modelo=sub_dia_op1 sub_sem_op1 sub_dia_op2
sub_sem_op2 sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1
min_dia_op1 avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1
max_ano_op1 min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2
avg_sem_op2 max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2
min_ano_op2 Producto Empresa Fecha_semana mes,
                metodo=backward,criterio=AIC,sinicio=12345,sfi-
nal=12456,fracciontrain=0.8,directorio=C:\Users\Lorena\Docu-
ments\TFM\log);

/* Metodo backward BIC*/
%randomselect(data=bd.data,listclass=Producto Empresa Fecha_se-
mana mes,vardepen=X1,
                modelo=sub_dia_op1 sub_sem_op1 sub_dia_op2
sub_sem_op2 sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1
min_dia_op1 avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1
max_ano_op1 min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2
avg_sem_op2 max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2
min_ano_op2 Producto Empresa Fecha_semana mes,
                metodo=backward,criterio=BIC,sinicio=12345,sfi-
nal=12456,fracciontrain=0.8,directorio=C:\Users\Lorena\Docu-
ments\TFM\log);

/* Metodo backward SBC*/
%randomselect(data=bd.data,listclass=Producto Empresa Fecha_se-
mana mes,vardepen=X1,
                modelo=sub_dia_op1 sub_sem_op1 sub_dia_op2
sub_sem_op2 sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1
min_dia_op1 avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1
max_ano_op1 min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2
avg_sem_op2 max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2
min_ano_op2 Producto Empresa Fecha_semana mes,
```

```
                metodo=backward,criterio=SBC,sinicio=12345,sfi-
nal=12456,fracciontrain=0.8,directorio=C:\Users\Lorena\Docu-
ments\TFM\log);

/* Metodo forward AIC*/
%randomselect(data=bd.data,listclass=Producto Empresa Fecha_se-
mana mes,vardepen=X1,
                modelo=sub_dia_op1 sub_sem_op1 sub_dia_op2
sub_sem_op2 sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1
min_dia_op1 avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1
max_ano_op1 min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2
avg_sem_op2 max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2
min_ano_op2 Producto Empresa Fecha_semana mes,
                metodo=forward,criterio=AIC,sinicio=12345,sfi-
nal=12456,fracciontrain=0.8,directorio=C:\Users\Lorena\Docu-
ments\TFM\log);

/* Metodo forward BIC*/
%randomselect(data=bd.data,listclass=Producto Empresa Fecha_se-
mana mes,vardepen=X1,
                modelo=sub_dia_op1 sub_sem_op1 sub_dia_op2
sub_sem_op2 sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1
min_dia_op1 avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1
max_ano_op1 min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2
avg_sem_op2 max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2
min_ano_op2 Producto Empresa Fecha_semana mes,
                metodo=forward,criterio=BIC,sinicio=12345,sfi-
nal=12456,fracciontrain=0.8,directorio=C:\Users\Lorena\Docu-
ments\TFM\log);

/* Metodo forward SBC*/
%randomselect(data=bd.data,listclass=Producto Empresa Fecha_se-
mana mes,vardepen=X1,
                modelo=sub_dia_op1 sub_sem_op1 sub_dia_op2
sub_sem_op2 sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1
min_dia_op1 avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1
max_ano_op1 min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2
avg_sem_op2 max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2
min_ano_op2 Producto Empresa Fecha_semana mes,
                metodo=forward,criterio=SBC,sinicio=12345,sfi-
nal=12456,fracciontrain=0.8,directorio=C:\Users\Lorena\Docu-
ments\TFM\log);

/*************************************************************
****
/* MACRO VALIDACIÓN CRUZADA PARA REGRESIÓN NORMAL

%macro cruzada(archivo=,vardepen=,conti=,categor=,ngrupos=,si-
nicio=,sfinal=);

archivo=archivo de datos
vardepen=nombre de la variable dependiente
conti=variables independientes continuas
categor=variables independientes categóricas
ngrupos=número de grupos a dividir por validación cruzada
inicio=semilla de inicio
```

```
sfinal=semilla final

*******
SALIDA
*******
La macro obtiene la suma y media de errores al cuadrado por CV
para cada semilla.
Esta información está contenida en el archivo final.
Variables:
media=media de errores de validación cruzada por cada ejecución-
semilla
suma=suma de errores de validación cruzada por cada ejecución-
semilla

NOTAS

1) Se puede poner antes de ejecuciones largas la sentencia
     options nonotes;
para no llenar la ventana log y que no nos pida borrarla.
Para volver a ver comentarios-errores en log:
     options notes;

2) Para comparar modelos, añadir la variable modelo despues de
ejecutada la macro con
cada uno de los modelos:
%cruzada...
data final1;set final;modelo=1;run;
...
%cruzada...
data final5;set final;modelo=5;run;

y finalmente union y un boxplot:

data union;set final1 final2...final5;
proc boxplot data=final;plot media*modelo;run;
**************************************************************
*********/

%macro cruzada(archivo=,vardepen=,conti=,categor=,ngrupos=,si-
nicio=,sfinal=);
data final;run;
/* Bucle semillas */
%do semilla=&sinicio %to &sfinal;
     data dos;set &archivo;u=ranuni(&semilla);
     proc sort data=dos;by u;run;
     data dos;
     retain grupo 1;
     set dos nobs=nume;
     if _n_>grupo*nume/&ngrupos then grupo=grupo+1;
     run;
     data fantasma;run;
     %do exclu=1 %to &ngrupos;
          data tres;set dos;if grupo ne &exclu then var-
dep=&vardepen;
          proc glm data=tres noprint;/*<<<<<******SE PUEDE QUI-
TAR EL NOPRINT */
```

```
          %if &categor ne %then %do;class &categor;model var-
dep=&conti &categor;%end;
          %else %do;model vardep=&conti;%end;
          output out=sal p=predi;run;
          data sal;set sal;resi2=(&vardepen-predi)**2;if
grupo=&exclu then output;run;
          data fantasma;set fantasma sal;run;
     %end;
     proc means data=fantasma sum noprint;var resi2;
     output out=sumaresi sum=suma mean=media;
     run;
     data sumaresi;set sumaresi;semilla=&semilla;
     data final (keep=suma media semilla);set final sumaresi;if
suma=. then delete;run;
%end;
proc print data=final;run;
%mend;


/* Metodo stepwise y forward AIC */
%cruzada(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op2 N_sin_subasta,categor=Producto Fecha_semana
mes,ngrupos=5,sinicio=12345,sfinal=12456);
data regresion1;set final;modelo='reg1';run;
%cruzada(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op2 N_sin_subasta,categor=Producto Empresa Fecha_semana
mes,ngrupos=5,sinicio=12345,sfinal=12456);
data regresion2;set final;modelo='reg2';run;
/* Metodo stepwise y forward BIC */
%cruzada(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op2 N_sin_subasta,categor=Producto Fecha_semana
mes,ngrupos=5,sinicio=12345,sfinal=12456);
data regresion3;set final;modelo='reg3';run;
%cruzada(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op1 sub_sem_op2 N_sin_subasta,categor=Producto Fecha_se-
mana mes,ngrupos=5,sinicio=12345,sfinal=12456);
data regresion4;set final;modelo='reg4';run;
%cruzada(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op2 N_sin_subasta,categor=Producto Empresa Fecha_semana
mes,ngrupos=5,sinicio=12345,sfinal=12456);
data regresion5;set final;modelo='reg5';run;
/* Metodo stepwise y forward y backward SBC */
%cruzada(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op2 N_sin_subasta,categor=Producto Fecha_semana
mes,ngrupos=5,sinicio=12345,sfinal=12456);
data regresion6;set final;modelo='reg6';run;
%cruzada(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op2 N_sin_subasta,categor=Producto Fecha_semana,ngru-
pos=5,sinicio=12345,sfinal=12456);
data regresion7;set final;modelo='reg7';run;
%cruzada(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op2 N_sin_subasta,categor=Fecha_semana,ngrupos=5,si-
nicio=12345,sfinal=12456);
data regresion8;set final;modelo='reg8';run;
/* Metodo backward AIC y BIC */
```

```
%cruzada(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op2 sub_ano_op2 N_sin_subasta avg_ano_op1
max_ano_op1,categor=Producto Empresa Fecha_semana mes,ngru-
pos=5,sinicio=12345,sfinal=12456);
data regresion9;set final;modelo='reg9';run;

/* Una vez que tenemos hechas las regresiones lineales con las
variables halladas en la seleccion de
variables haremos un diagrama de cajas para compararlas y que-
darnos con el mejor modelo de regresion lineal*/
data union;set regresion1 regresion2 regresion3 regresion4 re-
gresion5 regresion6 regresion7 regresion8 regresion9;
proc boxplot data=union;plot media*modelo;run;

/*************************************************************/
/* Ahora procederemos a las redes neuronales cogiendo como se-
leccion de variables las salidas que nos ha quedado con menor
ecm*/

proc dmdb data = bd.data dmdbcat = cataprueba;
     target X1;
     var X1 sub_dia_op1 sub_sem_op2 N_sin_subasta; *metemos la v
objetivo;
     class Producto Empresa Fecha_semana mes;
run;

/* Con la macro repito lo que nos devuelve es el numero de nodos
que deberemos de usar para la red a traves de validacion cru-
zada*/
%macro repito;
data union;run;
%do nodos=1 %to 10 %by 1;
proc neural data=bd.data dmdbcat=cataprueba ;
input sub_dia_op1 sub_sem_op2 N_sin_subasta; *sin la v.obetivo;
input Producto Empresa Fecha_semana mes /level=nominal;
target  X1;
hidden &nodos;
prelim 2 preiter=3;
train tech=levmar;
score data=bd.data out=salpredi outfit=salfit;
run;
data salfit;set salfit;nodos=&nodos;if _n_=2 then output;
data union;set union salfit;run;
%end;
data union;set union;if _n_=1 then delete;run;
proc print data=union;run;
%mend;

%repito;
/*Nos sale un grafico, con este grafico elegimos un intervalo de
NODOS para el que hacer la red*/
/*El punto mas bajo y teniendo en cuenta las iteraciones sera el
idoneo*/
tittle 'ASE en función del número de nodos';
symbol i=join v=circle;
proc gplot data=union;plot _ASE_*nodos;run;
```

```
/*Por el diagrama de bigotes elegimos la red con 4, 6 y 8 nodos
para probar*/

/* Ahora veremos cual es la fun activacion y el metodo idoneo
para nuestra red, para ello calcularemos modelos de redes neuro-
nales
     con validacion cruzada para cada f. activacion y metodo y
nos quedaremos con la que menor ecm tenga*/

/* Definimos la funcion que hace la validacion cruzada para la
red */
/* MACRO VALIDACIÓN CRUZADA PARA REDES NEURONALES

%macro cruzadaneural(archivo=,vardepen=,conti=,categor=,ngru-
pos=,sinicio=,sfinal=,ocultos=,algo=,acti=,early=);

archivo=archivo de datos
vardepen=nombre de la variable dependiente
conti=variables independientes continuas
categor=variables independientes categóricas
ngrupos=número de grupos a dividir por validación cruzada
sinicio=semilla de inicio
sfinal=semilla final
nodos= número de nodos
early=iteraciones early stopping (dejar como early=, si no se
desea)
algo=algoritmo (poner bprop mom=0.2 learn=0.1 si es bprop)

La macro obtiene la suma y media de errores al cuadrado por CV
para cada semilla.
Esta información está contenida en el archivo llamado final

*/

%macro cruzadaneural(archivo=,vardepen=,conti=,categor=,ngru-
pos=,sinicio=,sfinal=,ocultos=,algo=,acti=,early=);
/*Si no se quiere información en output usar esto (cambiar el
archivo de destino):
proc printto print='&directorio\basura.txt';
*/
proc printto print="C:\Users\Lorena\Desktop\TFM\basura.txt";
data final;run;
%do semilla=&sinicio %to &sfinal;
     data dos;set &archivo;u=ranuni(&semilla);
     proc sort data=dos;by u;run;
     data dos (drop=nume);
      retain grupo 1;
      set dos nobs=nume;
      if _n_>grupo*nume/&ngrupos then grupo=grupo+1;
     run;
     data fantasma;run;
     %do exclu=1 %to &ngrupos;
          data trestr tresval;
          set dos;if grupo ne &exclu then output trestr;else
output tresval;
          PROC DMDB DATA=trestr dmdbcat=catatres;
```

```
           target &vardepen;
           var &vardepen &conti;
           %if &categor ne %then %do;class &categor;%end;
           run;
           proc neural data=trestr dmdbcat=catatres random=789 ;
           input &conti;
           %if &categor ne %then %do;input &categor /level=nomi-
nal;%end;
           target &vardepen;
           hidden &ocultos /act=&acti;/*<<<<<******PARA DATOS
LINEALES ACT=LIN (funciónde activación lineal)
           NORMALMENTE PARA DATOS NO LINEALES MEJOR ACT=TANH */
           /* A PARTIR DE AQUI SON ESPECIFICACIONES DE LA RED,
SE PUEDEN CAMBIAR O AÑADIR COMO PARÁMETROS */

/* ESTO ES PARA EARLY STOPPING (maxiter=numero de iteraciones
limitado)*/

     %if &early ne %then %do;
           nloptions maxiter=&early;
           netoptions randist=normal ranscale=0.1 ran-
dom=15115;%end;
           /* %else %do;prelim 10;%end;*/
     %if &early ne %then %do;
           train maxiter=&early /* early stopping cambiar ma-
xiter=25 por ejemplo */ outest=mlpest technique=&algo;%end;
           %else %do;train maxiter=100 /* early stopping cambiar
maxiter=25 por ejemplo */ outest=mlpest technique=&algo/* bprop
mom=0.2 learn=0.1*/;%end;
           score data=tresval role=valid out=sal ;
           run;
           data sal;set sal;resi2=(p_&vardepen-&varde-
pen)**2;run;
           data fantasma;set fantasma sal;run;
     %end;
     proc means data=fantasma sum noprint;var resi2;
     output out=sumaresi sum=suma mean=media;
     run;
     data sumaresi;set sumaresi;semilla=&semilla;
     data final (keep=suma media semilla);set final sumaresi;if
suma=. then delete;run;
%end;
proc printto;run;
proc print data=final;run;
%mend;

/*En primer lugar elegiremos la funcion de activacion, probare-
mos entre TANH LOG ARC LIN SIN SOF GAU, le aplicaremos la red
neuronal
con validacion cruzada y luego decidiremos cual es la mejor fun
act para nuestra red*/
/*Probaremos con la red de 4,6,8 nodos*/
%macro activalcruza(ocultos=);
%let lista='TANH LOG ARC LIN SIN SOF GAU';
%let nume=7;
%do  i=1 %to &nume;
```

```
data _null_;activa=scanq(&lista,&i);call symput('acti-
va',left(activa));run;
 %cruzadaneural(archivo=bd.data,vardepen=X1,
conti= sub_dia_op1 sub_sem_op2 N_sin_subasta,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12360,ocultos=&ocultos,acti=&ac-
tiva,algo=LEVMAR,early=);

 data final&i;set final;modelo="&activa";put modelo=;run;
%end;
data union&ocultos;set final1 final2 final3 final4 final5 final6
final7;run;
%mend;


/*red con 4 nodos*/
%activalcruza(ocultos=4);
proc print data=union4;run;
proc boxplot data=union4;plot media*modelo;run;
/*red con 6 nodos*/
%activalcruza(ocultos=6);
proc print data=union6;run;
proc boxplot data=union6;plot media*modelo;run;
/*red con 8 nodos*/
%activalcruza(ocultos=8);
proc print data=union8;run;
proc boxplot data=union8;plot media*modelo;run;
/*Por el diagrama de bigotes decidimos que la mejor fun act es
para 4 nodos: tanh
para 6 nodos: arc
para 8 nodos: log
*/
options nonotes;
/*Una vez que sabemos cual es la mejor fun de activacion habra
que elegir el mejor metodo, iteraremos probando los metodos de
BPROP LEVMAR QUANEW TRUREG a nuestra red con validacion cru-
zada*/
%macro algovalcruza(ocultos=,acti=);
%let lista='LEVMAR QUANEW TRUREG';
%let nume=3;
%do  i=1 %to &nume;
data _null_;meto=scanq(&lista,&i);call symput('me-
to',left(meto));run;
 %cruzadaneural(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op2 N_sin_subasta,categor=Producto Empresa Fecha_semana
mes,ngrupos=5,sinicio=12345,sfinal=12360,ocultos=&ocul-
tos,acti=&acti,algo=&meto,early=);
 data final&i;set final;modelo="&meto";put modelo=;run;
%end;
data union&ocultos;set final1 final2 final3;run;
%mend;

/*red con 4 nodos con la funcion de activacion tanh*/
%algovalcruza(ocultos=4, acti=tanh);
proc print data=union4;run;
proc boxplot data=union4;plot media*modelo;run;
```

```
/*red con 6 nodos con la funcion de activacion log */
%algovalcruza(ocultos=6, acti=arc);
proc print data=union6;run;
proc boxplot data=union6;plot media*modelo;run;
/*red con 8 nodos con la funcion de activacion arc*/
%algovalcruza(ocultos=8, acti=log);
proc print data=union8;run;
proc boxplot data=union8;plot media*modelo;run;
/*Por el diagrama de bigotes decidimos el mejor metodo es
para 4 nodos: levmar
para 6 nodos: levmar
para 8 nodos: levmar
*/


/* Buscamos el numero ideal de nodos para nuestra red entre 2 y
9 nodos, lo haremos por validacion cruzada*/

%macro nodosvalcruza(ini=,fin=,increme=,archivo=,varde-
pen=,conti=,categor=,acti=,algo=);
%do nod=&ini %to &fin %by &increme;
 %cruzadaneural(archivo=&archivo,vardepen=&varde-
pen,conti=&conti,categor=&categor,ngrupos=5,
      sinicio=12345,sfinal=12355,ocul-
tos=&nod,algo=&algo,acti=&acti,early=);

 data finaln&nod;set final;modelo=&nod;run;
%end;
data union&acti;set %do i=&ini %to &fin %by &increme; finaln&i
%end;;run;
%mend;

/*f.act = tanh, metodo =  levmar*/
%nodosvalcruza(ini=2,fin=9,increme=1,archivo=bd.data,varde-
pen=X1,conti=sub_dia_op1 sub_sem_op2 N_sin_subasta,
categor=Producto Empresa Fecha_semana
mes,acti=tanh,algo=levmar);
/*proc print data=uniontanh;run;*/
/*proc boxplot data=uniontanh;plot media*modelo;run;*/

/*f.act = arc, metodo =  levmar*/
%nodosvalcruza(ini=2,fin=9,increme=1,archivo=bd.data,varde-
pen=X1,conti=sub_dia_op1 sub_sem_op2 N_sin_subasta,
categor=Producto Empresa Fecha_semana mes,acti=arc,algo=levmar);
/*proc print data=unionarc;run;*/
/*proc boxplot data=unionarc;plot media*modelo;run;*/

/*f.act = log, metodo = levmar*/
%nodosvalcruza(ini=2,fin=9,increme=1,archivo=bd.data,varde-
pen=X1,conti=sub_dia_op1 sub_sem_op2 N_sin_subasta,
categor=Producto Empresa Fecha_semana mes,acti=log,algo=levmar);
/*proc print data=unionlog;run;*/
/*proc boxplot data=unionlog;plot media*modelo;run;*/

/* Tenemos que ver si para nuestras redes seria conveniente
aplicar early stopping*/
```

```
/* Nos queda por ver el numero ideal para early stopping para
eso lo haremos con la macro red neuronal*/
/************************************************************
*********************
 MACRO redneuronal(archivo=,listclass=,listconti=,vardep=,por-
cen=,semilla=,ocultos=,algo=,acti=);

archivo= archivo de datos
listclass= lista de variables de clase
listconti= lista de variables continuas
vardep=variable dependiente
porcen= porcentaje de training
semilla=semilla para hacer la partición
ocultos=número de nodos ocultos
************************************************************
*********************/

%macro redneuronal(archivo=,listclass=,listconti=,vardep=,por-
cen=,semilla=,ocultos=,algo=,acti=);

%if &listclass eq %then %do;

PROC DMDB DATA=&archivo dmdbcat=catauno;
target &vardep;
var &listconti &vardep;
run;
%end;
%else %do;
PROC DMDB DATA=&archivo dmdbcat=catauno;
target &vardep;
var &listconti &vardep;
class &listclass;
run;
%end;

data ooo;set &archivo;run;
data datos;set ooo nobs=nume;tr=int(&porcen*nume);call
symput('tr',left(tr));u=ranuni(&semilla);run;
proc sort data=datos;by u;run;
data datos valida;set datos;if _n_>tr then output valida;else
output datos;run;

proc neural data=datos dmdbcat=catauno validata=valida graph;
input &listconti / id=i;
input &listclass / level=nominal;
target &vardep / id=o;
hidden &ocultos / id=h act=&acti;
nloptions maxiter=10000;
netoptions randist=normal ranscale=0.1 random=15115;
train maxiter=10000 outest=mlpest estiter=1 technique=&algo;
score data=datos out=mlpout outfit=mlpfit;
score data=valida out=mlpout2 outfit=mlpfit2 role=valid;
run;

data mlpest2 ;
k=3;
```

```
retain iterepocas 0;
set mlpest;
eval=_VOBJERR_;
x3=lag3(eval);
x6=lag6(eval);
if _n_>6 and eval>x3 and eval>x6 then iterepocas=_n_;
run;

data;
set mlpest2 nobs=nume;
if iterepocas ne 0 then do;control=1;
call symput('earlystop',left(iterepocas));
stop;
end;
if _n_=nume and control ne 1 then do;
ka=0;
call symput('earlystop',left(ka));
end;
run;

data fin;j=&earlystop;set mlpest point=j;output;stop;run;

data mlpest;set mlpest nobs=nume; if _n_=&earlystop then do;
cosa1=put(_OBJERR_,20.6) ;
cosa2=put(_VOBJERR_,20.6) ;
end;
else do;cosa1=' ';cosa2=' ';end;
run;

title1
h=2 box=1 j=c c=red 'TRAIN' c=blue '  VALIDA'
h=1.5 j=c c=black "EARLY STOPPING=&earlystop " "semilla=&semi-
lla"
h=1 j=c c=green "NODOS OCULTOS: &ocultos  " " METODO: &algo "
"ACTIVACIÓN: &acti";
;

symbol1 c=red v=circle i=join pointlabel=("#cosa1" h=1 c=red po-
sition=bottom  j=c);
symbol2 c=blue v=circle i=join pointlabel=("#cosa2" h=1 c=blue
position=top j=c);

axis1 label=none;
proc gplot data=mlpest;plot _OBJERR_ *_iter_=1 _VOB-
JERR_*_iter_=2
/overlay href=&earlystop vaxis=axis1 haxis=axis1 ;run;

proc print data=fin;
var  _iter_ _OBJERR_  _AVERR_  _VNOBJ_   _VOBJ_  _VOBJERR_  _VA-
VERR_
;run;

%mend;

/*f.act = tanh, metodo = levmar */
```

```
%redneuronal(archivo=bd.data,listclass=Producto Empresa Fe-
cha_semana mes,listconti=sub_dia_op1 sub_sem_op2 N_sin_subasta,
vardep=X1,porcen=0.80,semilla=442711,ocul-
tos=2,algo=LEVMAR,acti=tanh);

/*f.act = arc, metodo = levmar */
%redneuronal(archivo=bd.data,listclass=Producto Empresa Fe-
cha_semana mes,listconti=sub_dia_op1 sub_sem_op2 N_sin_subasta,
vardep=X1,porcen=0.80,semilla=442711,ocul-
tos=3,algo=LEVMAR,acti=arc);

/*f.act = log, metodo = levmar */
%redneuronal(archivo=bd.data,listclass=Producto Empresa Fe-
cha_semana mes,listconti=sub_dia_op1 sub_sem_op2 N_sin_subasta,
vardep=X1,porcen=0.80,semilla=442711,ocul-
tos=2,algo=LEVMAR,acti=log);


/* Validacion cruzada para redes neuronales con earlystopping y
sin earlystopping para luego comparar*/
/*f.act = tanh, metodo = levmar, nodos = 2, sin early stopping*/
%cruzadaneural(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op2 N_sin_subasta,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,ocul-
tos=2,algo=levmar,acti=tanh,early=);
data red1;set final;modelo='red1';run;
/*f.act = tanh, metodo = levmar, nodos = 2, early stopping=5*/
%cruzadaneural(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op2 N_sin_subasta,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,ocul-
tos=2,algo=levmar,acti=tanh,early=5);
data red2;set final;modelo='red2';run;
/*f.act = tanh, metodo = levmar, nodos = 2, early stopping=10*/
%cruzadaneural(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op2 N_sin_subasta,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,ocul-
tos=2,algo=levmar,acti=tanh,early=10);
data red3;set final;modelo='red3';run;
/*f.act = arc, metodo = levmar, nodos = 3, sin earlystopping*/
%cruzadaneural(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op2 N_sin_subasta,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,ocultos=3,algo=levmar,acti=arc,early=);
data red4;set final;modelo='red4';run;
/*f.act = arc, metodo = levmar, nodos = 3, earlystopping=10*/
%cruzadaneural(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op2 N_sin_subasta,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,ocul-
tos=3,algo=levmar,acti=arc,early=10);
data red5;set final;modelo='red5';run;
/*f.act = log, metodo = quanew, nodos = 3, sin earlystopping */
```

```
%cruzadaneural(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op2 N_sin_subasta,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,ocultos=2,algo=levmar,acti=log,early=);
data red6;set final;modelo='red6';run;
/*f.act = log, metodo = quanew, nodos = 3, earlystopping = 7*/
%cruzadaneural(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op2 N_sin_subasta,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,ocul-
tos=2,algo=levmar,acti=log,early=7);
data red7;set final;modelo='red7';run;
/*f.act = log, metodo = quanew, nodos = 3, earlystopping = 12*/
%cruzadaneural(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op2 N_sin_subasta,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,ocul-
tos=2,algo=levmar,acti=log,early=12);
data red8;set final;modelo='red8';run;

/*Comparamos las redes*/
data union;set red1 red2 red3 red4 red5 red6 red7 red8;
proc boxplot data=union;plot media*modelo;run;


data union;set red1 red4 red6;
proc boxplot data=union;plot media*modelo;run;


/*Comparamos los resultados de la red con la regresion lineal*/
data union;set red2 rg2 rg11;
proc boxplot data=union;plot media*modelo;run;

/************************ VARIABLE OBJETIVO BINARIA
************************/

/****** REGRESION LOGISTICA ******/

/* LA MACRO RANDOMSELECTlog REALIZA REGRRESION REPETIDAS VECES
CON DIFERENTES ARCHIVOS TRAIN.

LA SALIDA INCLUYE UNA TABLA DE FRECUENCIAS
DE LOS MODELOS QUE APARECEN SELECCIONADOS EN LOS DIFERENTES
ARCHIVOS TRAIN

LOS MODELOS QUE SALEN MÁS VECES SON POSIBLES CANDIDATOS A PROBAR
CON VALIDACIÓN CRUZADA

listclass=lista de variables de clase ATENCIÓN, EN ESTA LISTA
SOLO PONER VARIABLES
                QUE SE VAYAN A USAR (BIEN COMO EFECTOS PRINCIPA-
LES O INTERACCIONES)
vardepen=variable dependiente
modelo=modelo
sinicio=semilla inicio
sfinal=semilla final
```

```
fracciontrain=fracción de datos train
directorio=directorio para archivos basura

EL ARCHIVO QUE CONTIENE LOS EFECTOS SE LLAMA SALEFEC.
SE INCLUYE EN EL LOG EL LISTADO PARA PODER COPIAR Y PEGAR
(A VECES LA VENTANA OUTPUT ESTÁ LIMITADA)

*/

%macro randomselectlog(data=,listclass=,vardepen=,modelo=,me-
todo=,sinicio=,sfinal=,fracciontrain=,directorio=);
options nocenter linesize=256;
proc printto print="&directorio\kk.txt";run;
data;file "&directorio\cosa2.txt" ;run;
%do semilla=&sinicio %to &sfinal;
proc surveyselect data=&data rate=&fracciontrain out=sal1234
seed=&semilla;run;

%if &listclass ne %then %do;
ods output type3=parametros;
proc logistic data=sal1234;
     class &listclass;
    model &vardepen= &modelo/ selection=&metodo. ;
run;
data parametros;length effect $20. modelo $ 20000;retain modelo
" ";set parametros end=fin;effect=cat(' ',effect);
if _n_ ne 1 then modelo=catt(modelo,' ',effect);if fin then
do;variable=modelo;output;end;
run;
%end;
%else %do;
ods output  Logistic.ParameterEstimates=parametros;
proc logistic data=sal1234;
       model &vardepen= &modelo/ selection=&metodo. ;
run;
%end;
ods graphics off;
ods html close;
data;file "&directorio\cosa2.txt" mod;set parametros;
%if &listclass ne %then %do; put variable @@;%end;
%else %do; if _n_ ne 1 then put variable @@;%end;
run;
%end;
proc printto ;run;
data todos;
infile "&directorio\cosa2.txt";
length efecto $ 400;
input efecto @@;
if efecto ne 'Intercept' then output;
run;
proc freq data=todos;tables efecto /out=sal;run;
proc sort data=sal;by descending count;
proc print data=sal;run;

data todos;
infile "&directorio\cosa2.txt";
```

```
length efecto $ 200;
input efecto $ &&;
run;
proc freq data=todos;tables efecto /out=sal;run;
proc sort data=sal;by descending count;
proc print data=sal;run;
data;set sal;put efecto;run;
%mend;


/* Metodo stepwise*/
%randomselectlog(data=bd.data,listclass=Producto Empresa Fe-
cha_semana mes,vardepen=precio_semana,
                modelo= sub_dia_op1 sub_sem_op1 sub_dia_op2
sub_sem_op2 sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1
min_dia_op1 avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1
max_ano_op1 min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2
avg_sem_op2 max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2
min_ano_op2 Producto Empresa Fecha_semana mes,
                metodo=stepwise,sinicio=12345,sfinal=12400,frac-
ciontrain=0.8,directorio=C:\Users\Lorena\Documents\TFM\log);
/*Candidatos:
sub_dia_op1 sub_sem_op2 N_sin_subasta Producto Empresa Fecha_se-
mana mes
sub_dia_op1 sub_sem_op2 max_ano_op1 N_sin_subasta Producto Em-
presa Fecha_semana mes
sub_sem_op2 N_sin_subasta Producto Empresa Fecha_semana mes
*/

/* Metodo backward*/
%randomselectlog(data=bd.data,listclass=Producto Empresa Fe-
cha_semana mes,vardepen=precio_semana,
                modelo= sub_dia_op1 sub_sem_op1 sub_dia_op2
sub_sem_op2 sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1
min_dia_op1 avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1
max_ano_op1 min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2
avg_sem_op2 max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2
min_ano_op2 Producto Empresa Fecha_semana mes,
                metodo=backward,sinicio=12345,sfinal=12400,frac-
ciontrain=0.8,directorio=C:\Users\Lorena\Documents\TFM\log);
/*Candidatos:
sub_dia_op1 sub_sem_op2 N_sin_subasta avg_sem_op2 Producto Em-
presa Fecha_semana mes
sub_dia_op1 sub_sem_op2 N_sin_subasta avg_sem_op1 avg_dia_op2
min_dia_op2 min_sem_op2 Producto Empresa Fecha_semana mes
*/

/* Metodo forward*/
%randomselectlog(data=bd.data,listclass=Producto Empresa Fe-
cha_semana mes,vardepen=precio_semana,
                modelo= sub_dia_op1 sub_sem_op1 sub_dia_op2
sub_sem_op2 sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1
min_dia_op1 avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1
max_ano_op1 min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2
avg_sem_op2 max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2
min_ano_op2 Producto Empresa Fecha_semana mes,
```

```
                metodo=forward,sinicio=12345,sfinal=12400,frac-
ciontrain=0.8,directorio=C:\Users\Lorena\Documents\TFM\log);
/*Candidatos:
sub_dia_op1 sub_sem_op2 N_sin_subasta Producto Empresa Fecha_se-
mana mes
sub_dia_op1 sub_sem_op2 max_ano_op1 N_sin_subasta Producto Em-
presa Fecha_semana mes
sub_sem_op2 N_sin_subasta Producto Empresa Fecha_semana mes
*/

/*Una vez que tenemos nuestra seleccion de variables aplicaremos
validacion cruzada a la regresion logistica para ver cual es la
mejor regresion para la prediccion de nuestro modelo*/

/* VALIDACIÓN CRUZADA LOGÍSTICA PARA VARIABLES DEPENDIENTES BI-
NARIAS

*****************************************************************
*****************
                                                PARÁMETROS
*****************************************************************
*****************

BÁSICOS

archivo=        archivo de datos
vardepen=       variable dependiente binaria
categor=        lista de variables independientes categóricas
conti=               lista de variables independientes conti-
nuas Y TODAS LAS INTERACCIONES
ngrupos=        número de grupos validación cruzada
sinicio=        semilla inicial para repetición
sfinal=              semilla final para repetición
objetivo=       tasafallos,sensi,especif,porcenVN,porcenFN,por-
cenVP,porcenFP,precision,tasaciertos


El archivo final se llama final. La variable media es la media
del obejtivo en todas las pruebas de validación cruzada
(habitualmente tasa de fallos).

*/

%macro cruzadalogistica(archivo=,vardepen=,conti=,categor=,ngru-
pos=,sinicio=,sfinal=,objetivo=tasafallos);
title ' ';
data final;run;
/* Bucle semillas */
%do semilla=&sinicio %to &sfinal;
     data dos;set &archivo;u=ranuni(&semilla);
     proc sort data=dos;by u;run;
     data dos (drop=nume);
     retain grupo 1;
     set dos nobs=nume;
     if _n_>grupo*nume/&ngrupos then grupo=grupo+1;
     run;
```

```
      data fantasma;run;
      %do exclu=1 %to &ngrupos;
            data tres;set dos;if grupo ne &exclu then var-
dep=&vardepen;
            proc logistic data=tres noprint;/*<<<<<******SE PUEDE
QUITAR EL NOPRINT */
            %if (&categor ne) %then %do;class &categor;model var-
dep=&conti &categor ;%end;
            %else %do;model vardep=&conti;%end;
            output out=sal p=predi;run;
            data sal2;set sal;pro=1-predi;if pro>0.5 then
pre11=1; else pre11=0;
            if grupo=&exclu then output;run;
            proc freq data=sal2;tables pre11*&varde-
pen/out=sal3;run;
            data estadisticos (drop=count percent pre11 &varde-
pen);
            retain vp vn fp fn suma 0;
            set sal3 nobs=nume;
            suma=suma+count;
            if pre11=0 and &vardepen=0 then vn=count;
            if pre11=0 and &vardepen=1 then fn=count;
            if pre11=1 and &vardepen=0 then fp=count;
            if pre11=1 and &vardepen=1 then vp=count;
            if _n_=nume then do;
            porcenVN=vn/suma;
            porcenFN=FN/suma;
            porcenVP=VP/suma;
            porcenFP=FP/suma;
            sensi=vp/(vp+fn);
            especif=vn/(vn+fp);
            tasafallos=1-(vp+vn)/suma;
            tasaciertos=1-tasafallos;
            precision=vp/(vp+fp);
            F_M=2*Sensi*Precision/(Sensi+Precision);
            output;
            end;
            run;

            data fantasma;set fantasma estadisticos;run;
      %end;
      proc means data=fantasma sum noprint;var &objetivo;
      output out=sumaresi sum=suma mean=media;
      run;
      data sumaresi;set sumaresi;semilla=&semilla;
      data final (keep=suma media semilla);set final sumaresi;if
suma=. then delete;run;
%end;
proc print data=final;run;
%mend;

/*Stepwise y Forward*/
%cruzadalogistica(archivo=bd.data,vardepen=precio_semana,cate-
gor=Producto Empresa Fecha_semana mes,
conti=sub_dia_op1 sub_sem_op2 N_sin_subasta,ngrupos=5,si-
nicio=12345,sfinal=12400,objetivo=tasafallos);
```

```
data logistica1;set final;modelo=1;run;

%cruzadalogistica(archivo=bd.data,vardepen=precio_semana,cate-
gor=Producto Empresa Fecha_semana mes,
conti=sub_dia_op1 sub_sem_op2 max_ano_op1 N_sin_subasta,ngru-
pos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data logistica2;set final;modelo=2;run;

%cruzadalogistica(archivo=bd.data,vardepen=precio_semana,cate-
gor=Producto Empresa Fecha_semana mes,
conti=sub_sem_op2 N_sin_subasta,ngrupos=5,sinicio=12345,sfi-
nal=12400,objetivo=tasafallos);
data logistica3;set final;modelo=3;run;

/* Backward */
%cruzadalogistica(archivo=bd.data,vardepen=precio_semana,cate-
gor=Producto Empresa Fecha_semana mes,
conti=sub_dia_op1 sub_sem_op2 N_sin_subasta avg_sem_op2,ngru-
pos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data logistica4;set final;modelo=4;run;

%cruzadalogistica(archivo=bd.data,vardepen=precio_semana,cate-
gor=Producto Empresa Fecha_semana mes,
conti=sub_dia_op1 sub_sem_op2 N_sin_subasta avg_sem_op1
avg_dia_op2 min_dia_op2 min_sem_op2,ngrupos=5,sinicio=12345,sfi-
nal=12400,objetivo=tasafallos);
data logistica5;set final;modelo=5;run;

data union;set logistica1 logistica2 logistica3 logistica4 lo-
gistica5; run;
proc boxplot data=union;plot media*modelo;run;

/* REDES NEURONALES */

/* NUMERO DE NODOS Y PUNTO DE CORTE CON VALIDACION CRUZADA*/
/* ***************MACRO neuralbinariaba-
sica************************

El objetivo de esta macro es obtener un resultado básico de la
red
con una sola partición training test.

NOTA: LA VARIABLE DEPENDIENTE DEBE ESTAR CODIFICADA COMO 0 y 1,
SIENDO 1 LA CATEGORÍA DE INTERÉS

SE FIJAN LOS NODOS, PUNTO DE CORTE, SEMILLA, PORCENTAJE
OBVIAMENTE SE PUEDEN CAMBIAR LAS OPCIONES INTERNAS DEL PROC NEU-
RAL:
PRELIM, TRAIN, ETC.

El archivo que contiene la performance se llama estadisticos.

*/

%macro neuralbinariabasica(archivo=,listconti=,listclass=,var-
dep=,nodos=,corte=,semilla=,porcen=,algo=);
```

```
title '';
data archivobase;set &archivo nobs=nume;ene=int(&porcen*nume);
call symput('ene',left(ene));
run;

proc sort data=archivobase;by &vardep;run;

proc surveyselect data=archivobase out=muestra outall N=&ene
seed=&semilla;
/*si se quiere estratificacion en el muestreo quitar los comen-
tarios en strata*/
/* strata &vardep /alloc=proportional;*/run;
data train valida;set muestra;if selected=1 then output
train;else output valida;run;

PROC DMDB DATA=train dmdbcat=cataprueba;
target &vardep;
var &listconti;
class &listclass &vardep;
run;

%if &listclass ne %then %do;
proc neural data=train dmdbcat=cataprueba;
input  &listconti;
input &listclass /level=nominal;
target  &vardep /level=nominal;
hidden &nodos;
prelim 5;
train tech=&algo;
score data=valida out=salpredi outfit=salfit ;
run;
%end;

%else %do;
proc neural data=train dmdbcat=cataprueba;
input  &listconti;
target  &vardep /level=nominal;
hidden &nodos;
prelim 5;
train tech=&algo;
score data=valida out=salpredi outfit=salfit ;
run;
%end;

data salpredi;set salpredi;if p_&vardep.1>&corte/100 then
predi1=1;else predi1=0;run;
proc freq data=salpredi;tables predi1*&vardep/out=sal1;run;

/* Cálculo de estadísticos */

data estadisticos (drop=count percent predi1 &vardep);
retain vp vn fp fn suma 0;
set sal1 nobs=nume;
suma=suma+count;
if predi1=0 and &vardep=0 then vn=count;
if predi1=0 and &vardep=1 then fn=count;
```

```
if predi1=1 and &vardep=0 then fp=count;
if predi1=1 and &vardep=1 then vp=count;
if _n_=nume then do;
if vn=. then vn=0;if fn=. then fn=0;if vp=. then vp=0;if fp=.
then fp=0;
porcenVN=vn/suma;
porcenFN=FN/suma;
porcenVP=VP/suma;
porcenFP=FP/suma;
sensi=vp/(vp+fn);
especif=vn/(vn+fp);
tasafallos=1-(vp+vn)/suma;
tasaciertos=1-tasafallos;
precision=vp/(vp+fp);
if vp=0 then precision=0;
if vp=0 then sensi=0;
if vn=0 then especif=0;
F_M=2*Sensi*Precision/(Sensi+Precision);
output;
end;
run;
proc print data=estadisticos;run;

%mend;

/* MACRO VARIARBIS

UTILIZO LA MISMA MACRO PERO VARIANDO LA SEMILLA Y EL PUNTO DE
CORTE, Y
CONSERVANDO LA INFORMACIÓN PARA UN BOXPLOT */

%macro variarbis(vardep=,seminicio=,semifin=,inicionodos=,fi-
nalnodos=,increnodos=,archivo=,listconti=,listclass=,algo=);
title '';
data union;run;

%do semilla=&seminicio %to &semifin;
%do nodos=&inicionodos %to &finalnodos %by &increnodos;

%neuralbinariabasica(archivo=&archivo,listconti=&listconti,list-
class=&listclass,
vardep=&vardep,nodos=&nodos,corte=50,semilla=&semilla,por-
cen=0.80,algo=&algo);
%do corti=40 %to 70 %by 15;
data salpredi;set salpredi;if p_&vardep.1>&corti/100 then
predi1=1;else predi1=0;run;
proc freq data=salpredi;tables predi1*&vardep/out=sal1;run;

/* Cálculo de estadísticos */

data estadisticos (drop=count percent predi1 &vardep);
retain vp vn fp fn suma 0;
set sal1 nobs=nume;
suma=suma+count;
if predi1=0 and &vardep=0 then vn=count;
if predi1=0 and &vardep=1 then fn=count;
```

```
if predi1=1 and &vardep=0 then fp=count;
if predi1=1 and &vardep=1 then vp=count;
if _n_=nume then do;
if vn=. then vn=0;if fn=. then fn=0;if vp=. then vp=0;if fp=.
then fp=0;
porcenVN=vn/suma;
porcenFN=FN/suma;
porcenVP=VP/suma;
porcenFP=FP/suma;
sensi=vp/(vp+fn);
especif=vn/(vn+fp);
tasafallos=1-(vp+vn)/suma;
tasaciertos=1-tasafallos;
precision=vp/(vp+fp);
if vp=0 then precision=0;
if vp=0 then sensi=0;
if vn=0 then especif=0;
F_M=2*Sensi*Precision/(Sensi+Precision);
output;
end;
run;
   data estadisticos;set estadisticos;nodos=&nodos;semilla=&se-
milla;corte=&corti;run;
   data union;set union estadisticos;run;

%end;

%end;
%end;

data union;set union;if _n_=1 then delete;
proc sort data=union;by nodos corte;
proc print data=union;run;
data unionfin;retain nivel 0;set union;
by nodos corte;
if first.corte=1 then nivel=nivel+1;
output;
run;
proc print data=unionfin;var nodos corte nivel;run;
proc boxplot data=unionfin;
plot (porcenVN porcenFN porcenVP porcenFP especif tasafallos F_M
tasaciertos sensi precision )*nivel;run;

%mend;

/*modelo 2*/
%variarbis(archivo=bd.data,listclass=Producto Empresa Fecha_se-
mana mes,
listconti=sub_dia_op1 sub_sem_op2 max_ano_op1 N_sin_subasta,var-
dep=precio_semana,seminicio=12345,semifin=12400,iniciono-
dos=2,finalnodos=10,
increnodos=1,algo=BPROP);

data unionfinal;
set unionfin;
nodos_corte=catx(' ', nodos, corte);
```

```
run;

proc boxplot data=unionfinal;
plot (tasafallos tasaciertos)*nodos_corte;run;

/*modelo 5*/
%variarbis(archivo=bd.data,listclass=Producto Empresa Fecha_se-
mana mes,
listconti=sub_dia_op1 sub_sem_op2 N_sin_subasta avg_sem_op1
avg_dia_op2 min_dia_op2 min_sem_op2,vardep=precio_semana,semi-
nicio=12345,semifin=12400,inicionodos=1,finalnodos=15,
increnodos=1,algo=BPROP);

data unionfinal;
set unionfin;
nodos_corte=catx(' ', nodos, corte);
run;

proc boxplot data=unionfinal;
plot (tasafallos tasaciertos)*nodos_corte;run;

/* REDES NEURONALES CON VALIDACION CRUZADA*/
%macro cruzadabinarianeural(archivo=,vardepen=,conti=,cate-
gor=,ngrupos=,sinicio=,sfinal=,nodos=,algo=,obje-
tivo=,early=300,acti=tanh,directorio=C:\Users\Lorena\Desk-
top\TFM);
title ' ';
data final;run;
proc printto print="&directorio\basura.txt";

/* Bucle semillas */
%do semilla=&sinicio %to &sfinal;
     data dos;set &archivo;u=ranuni(&semilla);
     proc sort data=dos;by u;run;
     data dos (drop=nume);
     retain grupo 1;
     set dos nobs=nume;
     if _n_>grupo*nume/&ngrupos then grupo=grupo+1;
     run;
     data fantasma;run;
     %do exclu=1 %to &ngrupos;

     data trestr tresval;
          set dos;if grupo ne &exclu then output trestr;else
output tresval;
          PROC DMDB DATA=trestr dmdbcat=catatres;
          target &vardepen;
          var &conti;
          class &vardepen;
          %if &categor ne %then %do;class &categor &varde-
pen;%end;
          run;
          proc neural data=trestr dmdbcat=catatres random=789 ;
          input &conti;
          %if &categor ne %then %do;input &categor /level=nomi-
nal;%end;
```

```
           target &vardepen /level=nominal;
           hidden &nodos /act=&acti;/*<<<<<******PARA DATOS LI-
NEALES ACT=LIN (funciónde activación lineal)
           NORMALMENTE PARA DATOS NO LINEALES MEJOR ACT=TANH */
           /* A PARTIR DE AQUI SON ESPECIFICACIONES DE LA RED,
SE PUEDEN CAMBIAR O AÑADIR COMO PARÁMETROS */

           /*nloptions maxiter=500*/;
           netoptions randist=normal ranscale=0.15 random=15459;
           /* Si se desea hacer early stopping se pone prelim 0
y se marca como comentario
           la línea prelim 15...*/
           /*prelim 0 */
           prelim 15 preiter=10 pretech=&algo;
           train maxiter=&early outest=mlpest technique=&algo;
           score data=tresval role=valid out=sal ;
           run;
           data sal2;set sal;pro=1-%str(p_&vardepen)0;if pro>0.5
then pre11=1; else pre11=0;run;
                       proc freq data=sal2;tables pre11*&varde-
pen/out=sal3;run;

     data estadisticos (drop=count percent pre11 &vardepen);
           retain vp vn fp fn suma 0;
           set sal3 nobs=nume;
           suma=suma+count;
           if pre11=0 and &vardepen=0 then vn=count;
           if pre11=0 and &vardepen=1 then fn=count;
           if pre11=1 and &vardepen=0 then fp=count;
           if pre11=1 and &vardepen=1 then vp=count;
           if _n_=nume then do;
           porcenVN=vn/suma;
           porcenFN=FN/suma;
           porcenVP=VP/suma;
           porcenFP=FP/suma;
           sensi=vp/(vp+fn);
           especif=vn/(vn+fp);
           tasafallos=1-(vp+vn)/suma;
           tasaciertos=1-tasafallos;
           precision=vp/(vp+fp);
           F_M=2*Sensi*Precision/(Sensi+Precision);
           output;
           end;
           run;
     data fantasma;set fantasma estadisticos;run;
     %end;
     proc means data=fantasma sum noprint;var &objetivo;
     output out=sumaresi sum=suma mean=media;
     run;
     data sumaresi;set sumaresi;semilla=&semilla;
     data final (keep=suma media semilla);set final sumaresi;if
suma=. then delete;run;
%end;
proc printto ;
proc print data=final;run;
%mend;
```

```
/*FUNCION DE ACTIVACION CON VALIDACION CRUZADA*/

%macro activalcruza(conti=,categor=,nodos=);
%let lista='TANH ARC LIN SIN LOG GAU SOF';
%let nume=7;
%do  i=1 %to &nume;
data _null_;activa=scanq(&lista,&i);call symput('acti-
va',left(activa));run;
     %cruzadabinarianeural(archivo=bd.data,vardepen=precio_se-
mana,conti=&conti,categor=&categor,
     ngrupos=5,sinicio=12345,sfinal=12400,nodos=&no-
dos,algo=bprop mom=0.8 learn=0.2,acti=&activa,objetivo=tasafa-
llos);
data final&i;set final;modelo="&activa";put modelo=;run;
%end;
data union&nodos;set final1 final2 final3 final4 final5 final6
final7;run;
%mend;

/*modelo 2*/
/*nodos 4*/
%activalcruza(conti=sub_dia_op1 sub_sem_op2 max_ano_op1
N_sin_subasta,categor=Producto Empresa Fecha_semana mes,no-
dos=4);
proc print data=union4;run;
proc boxplot data=union4;plot media*modelo;run;

/*modelo 5*/
/*nodos 6*/
%activalcruza(conti=sub_dia_op1 sub_sem_op2 N_sin_subasta
avg_sem_op1 avg_dia_op2 min_dia_op2 min_sem_op2,categor=Producto
Empresa Fecha_semana mes,nodos=6);
proc print data=union6;run;
proc boxplot data=union6;plot media*modelo;run;

/*ALGORIMO DE OPTIMIZACION CON VALIDACION CRUZADA*/
/*Una vez que sabemos cual es la mejor fun de activacion habra
que elegir el mejor metodo, iteraremos probando los metodos de
BPROP LEVMAR QUANEW TRUREG a nuestra red con validacion cru-
zada*/
%macro algovalcruza(conti=,categor=,nodos=,acti=);
%let lista='BPROP LEVMAR QUANEW';
%let nume=3;
%do  i=1 %to &nume;
data _null_;meto=scanq(&lista,&i);call symput('me-
to',left(meto));run;
     %cruzadabinarianeural(archivo=bd.data,vardepen=precio_se-
mana,conti=&conti,categor=&categor,
     ngrupos=5,sinicio=12345,sfinal=12400,nodos=&no-
dos,algo=&meto,acti=&acti,objetivo=tasafallos);
 data final&i;set final;modelo="&meto";put modelo=;run;
%end;
data union&nodos;set final1 final2 final3;run;
%mend;
```

```
/*modelo 2*/
/*nodos 4*/
/* fun acti sin*/
%algovalcruza(conti=sub_dia_op1 sub_sem_op2 max_ano_op1
N_sin_subasta,categor=Producto Empresa Fecha_semana mes,no-
dos=4,acti=sin);
data modelo21;set modelo21;run;
proc print data=modelo21;run;
proc boxplot data=modelo21;plot media*modelo;run;

/*modelo 2*/
/*nodos 4*/
/* fun acti lin*/
%algovalcruza(conti=sub_dia_op1 sub_sem_op2 max_ano_op1
N_sin_subasta,categor=Producto Empresa Fecha_semana mes,no-
dos=4,acti=lin);
data modelo22;set union4;run;
proc print data=modelo22;run;
proc boxplot data=modelo22;plot media*modelo;run;

/*modelo 2*/
/*nodos 4*/
/* fun acti tanh*/
%algovalcruza(conti=sub_dia_op1 sub_sem_op2 max_ano_op1
N_sin_subasta,categor=Producto Empresa Fecha_semana mes,no-
dos=4,acti=tanh);
data modelo23;set union4;run;
proc print data=modelo23;run;
proc boxplot data=modelo23;plot media*modelo;run;

/*modelo 5*/
/*nodos 6*/
/* fun acti tanh*/
%algovalcruza(conti=sub_dia_op1 sub_sem_op2 N_sin_subasta
avg_sem_op1 avg_dia_op2 min_dia_op2 min_sem_op2,categor=Producto
Empresa Fecha_semana mes,nodos=6,acti=tanh);
data modelo51;set union6;run;
proc print data=modelo51;run;
proc boxplot data=modelo51;plot media*modelo;run;

/*modelo 5*/
/*nodos 6*/
/* fun acti sin*/
%algovalcruza(conti=sub_dia_op1 sub_sem_op2 N_sin_subasta
avg_sem_op1 avg_dia_op2 min_dia_op2 min_sem_op2,categor=Producto
Empresa Fecha_semana mes,nodos=6,acti=sin);
data modelo52;set union6;run;
proc print data=modelo52;run;
proc boxplot data=modelo52;plot media*modelo;run;

/*modelo 5*/
/*nodos 6*/
/* fun acti lin*/
%algovalcruza(conti=sub_dia_op1 sub_sem_op2 N_sin_subasta
avg_sem_op1 avg_dia_op2 min_dia_op2 min_sem_op2,categor=Producto
Empresa Fecha_semana mes,nodos=6,acti=lin);
```

```
data modelo53;set union6;run;
proc print data=modelo53;run;
proc boxplot data=modelo53;plot media*modelo;run;

/*MODELOS FINALES REDES CON VALIDACION CRUZADA*/

/*modelo 2*/
/*nodos 4*/
/*fun acti sin*/
/*algo quanew*/
%cruzadabinarianeural(archivo=bd.data,vardepen=precio_se-
mana,conti=sub_dia_op1 sub_sem_op2 max_ano_op1 N_sin_subasta,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,nodos=4,algo=QUANEW,acti=SIN,obje-
tivo=tasafallos);
data red1;set final;modelo='red1';run;

/*modelo 2*/
/*nodos 4*/
/*fun acti lin*/
/*algo levmar*/
%cruzadabinarianeural(archivo=bd.data,vardepen=precio_se-
mana,conti=sub_dia_op1 sub_sem_op2 max_ano_op1 N_sin_subasta,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,nodos=4,algo=levmar,acti=lIN,obje-
tivo=tasafallos);
data red2;set final;modelo='red2';run;

/*modelo 2*/
/*nodos 4*/
/*fun acti tanh*/
/*algo levmar*/
%cruzadabinarianeural(archivo=bd.data,vardepen=precio_se-
mana,conti=sub_dia_op1 sub_sem_op2 max_ano_op1 N_sin_subasta,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,nodos=4,algo=levmar,acti=tanh,obje-
tivo=tasafallos);
data red3;set final;modelo='red3';run;

/*modelo 5*/
/*nodos 6*/
/*fun acti tanh*/
/*algo levmar*/
%cruzadabinarianeural(archivo=bd.data,vardepen=precio_se-
mana,conti=sub_dia_op1 sub_sem_op2 N_sin_subasta avg_sem_op1
avg_dia_op2 min_dia_op2 min_sem_op2,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,nodos=6,algo=levmar,acti=tanh,obje-
tivo=tasafallos);
data red4;set final;modelo='red4';run;

/*modelo 5*/
/*nodos 6*/
/*fun acti lin*/
/*algo levmar*/
```

```
%cruzadabinarianeural(archivo=bd.data,vardepen=precio_se-
mana,conti=sub_dia_op1 sub_sem_op2 N_sin_subasta avg_sem_op1
avg_dia_op2 min_dia_op2 min_sem_op2,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,nodos=6,algo=levmar,acti=lin,obje-
tivo=tasafallos);
data red5;set final;modelo='red5';run;

/*modelo 5*/
/*nodos 6*/
/*fun acti sin*/
/*algo levmar*/
%cruzadabinarianeural(archivo=bd.data,vardepen=precio_se-
mana,conti=sub_dia_op1 sub_sem_op2 N_sin_subasta avg_sem_op1
avg_dia_op2 min_dia_op2 min_sem_op2,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,nodos=6,algo=levmar,acti=sin,obje-
tivo=tasafallos);
data red6;set final;modelo='red6';run;

/*data union; set red1 red2 red3 red4 red5 red6;run;*/
/*proc boxplot data=union;plot media*modelo;run;*/


/************************** BAGGING
************************************/

/* LA MACRO CRUZADARANDOMFORESTBIN REALIZA VALIDACIÓN CRUZADA
REPETIDA PARA VARIABLE
DEPENDIENTE BINARIA

SI SE DESEA HACER BAGGING SIMPLEMENTE SE PONE EL NÚMERO TOTAL DE
VARIABLES EN
EL PARAMETRO variables=

PARÁMETROS:

porcenbag=porcentaje de observaciones en cada iteración
variables=número de variables a sortear en cada nodo
tamhoja=tamaño mínimo de hoja final
maxtrees=iteraciones
maxbranch=divisiones máximas en un nodo
maxdepth=máxima profundidad
pvalor=p-valor para las divisiones de nodos

*/


%macro cruzadarandomforestbin(archivo=,vardep=,conti=,categor=,
maxtrees=100,variables=3,porcenbag=0.80,maxbranch=2,tam-
hoja=5,maxdepth=10,pvalor=0.1,
ngrupos=4,sinicio=12345,sfinal=12355,objetivo=tasafallos);

data final;run;
/* Bucle semillas */
%do semilla=&sinicio %to &sfinal;
```

```
data dos;set &archivo;u=ranuni(&semilla);
      proc sort data=dos;by u;run;
      data dos ;
      retain grupo 1;
      set dos nobs=nume;
      if _n_>grupo*nume/&ngrupos then grupo=grupo+1;
      run;


data fantasma;run;

%do exclu=1 %to &ngrupos;
data tres;set dos;if grupo ne &exclu then vardep=&vardep;


ods listing close;
proc hpforest data=tres
maxtrees=&maxtrees
vars_to_try=&variables
trainfraction=&porcenbag
leafsize=&tamhoja
maxdepth=&maxdepth
alpha=&pvalor
exhaustive=5000
missing=useinsearch ;
target vardep/level=nominal;
input &conti/level=interval;
%if (&categor ne) %then %do;
      input &categor/level=nominal;
      %end;
score out=salo;
run;
ods listing ;

data salo;merge salo tres;
if p_vardep1>0.5 then pre11=1;else pre11=0;
if grupo=&exclu;
run;

      proc freq data=salo;tables pre11*&vardep/out=sal3;run;
            data estadisticos (drop=count percent pre11 &vardep);
            retain vp vn fp fn suma 0;
            set sal3 nobs=nume;
            suma=suma+count;
            if pre11=0 and &vardep=0 then vn=count;
            if pre11=0 and &vardep=1 then fn=count;
            if pre11=1 and &vardep=0 then fp=count;
            if pre11=1 and &vardep=1 then vp=count;
            if _n_=nume then do;
            porcenVN=vn/suma;
            porcenFN=FN/suma;
            porcenVP=VP/suma;
            porcenFP=FP/suma;
            sensi=vp/(vp+fn);
            especif=vn/(vn+fp);
```

```
            tasafallos=1-(vp+vn)/suma;
            tasaciertos=1-tasafallos;
            precision=vp/(vp+fp);
            F_M=2*Sensi*Precision/(Sensi+Precision);
            output;
            end;
            run;

data fantasma;set fantasma estadisticos;run;

%end;/* fin grupos */
      proc means data=fantasma sum noprint;var &objetivo;
      output out=sumaresi sum=suma mean=media;
      run;
      data sumaresi;set sumaresi;semilla=&semilla;
      data final (keep=suma media semilla);set final sumaresi;if
suma=. then delete;run;
%end;/* fin semillas validación cruzada repetida*/


proc print data=final;run;


%mend;

/*modelo 2*/
%cruzadarandomforestbin(archivo=bd.data,vardep=precio_se-
mana,conti=sub_dia_op1 sub_sem_op2 max_ano_op1 N_sin_subasta,
categor=Producto Empresa Fecha_semana mes,
maxtrees=100,variables=8,porcenbag=0.80,maxbranch=2,tam-
hoja=5,maxdepth=10,pvalor=0.1,
ngrupos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data bag1; set final; modelo='bag1'; run;

/*modelo 2*/
%cruzadarandomforestbin(archivo=bd.data,vardep=precio_se-
mana,conti=sub_dia_op1 sub_sem_op2 max_ano_op1 N_sin_subasta,
categor=Producto Empresa Fecha_semana mes,
maxtrees=100,variables=8,porcenbag=0.70,maxbranch=2,tam-
hoja=5,maxdepth=10,pvalor=0.2,
ngrupos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data bag2; set final; modelo='bag2'; run;

/*modelo 2*/
%cruzadarandomforestbin(archivo=bd.data,vardep=precio_se-
mana,conti=sub_dia_op1 sub_sem_op2 max_ano_op1 N_sin_subasta,
categor=Producto Empresa Fecha_semana mes,
maxtrees=100,variables=8,porcenbag=0.60,maxbranch=2,tam-
hoja=6,maxdepth=10,pvalor=0.15,
ngrupos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data bag3; set final; modelo='bag3'; run;

/*modelo 5*/
%cruzadarandomforestbin(archivo=bd.data,vardep=precio_se-
mana,conti=sub_dia_op1 sub_sem_op2 N_sin_subasta avg_sem_op1
avg_dia_op2 min_dia_op2 min_sem_op2,
```

```
categor=Producto Empresa Fecha_semana mes,
maxtrees=100,variables=11,porcenbag=0.80,maxbranch=2,tam-
hoja=5,maxdepth=10,pvalor=0.1,
ngrupos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data bag4; set final; modelo='bag4'; run;

/*modelo 5*/
%cruzadarandomforestbin(archivo=bd.data,vardep=precio_se-
mana,conti=sub_dia_op1 sub_sem_op2 N_sin_subasta avg_sem_op1
avg_dia_op2 min_dia_op2 min_sem_op2,
categor=Producto Empresa Fecha_semana mes,
maxtrees=100,variables=11,porcenbag=0.70,maxbranch=2,tam-
hoja=5,maxdepth=10,pvalor=0.2,
ngrupos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data bag5; set final; modelo='bag5'; run;

/*modelo 5*/
%cruzadarandomforestbin(archivo=bd.data,vardep=precio_se-
mana,conti=sub_dia_op1 sub_sem_op2 N_sin_subasta avg_sem_op1
avg_dia_op2 min_dia_op2 min_sem_op2,
categor=Producto Empresa Fecha_semana mes,
maxtrees=100,variables=11,porcenbag=0.60,maxbranch=2,tam-
hoja=6,maxdepth=10,pvalor=0.15,
ngrupos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data bag6; set final; modelo='bag6'; run;

/*sin seleccion de variables*/
%cruzadarandomforestbin(archivo=bd.data,vardep=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,
maxtrees=100,variables=28,porcenbag=0.80,maxbranch=2,tam-
hoja=5,maxdepth=10,pvalor=0.1,
ngrupos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data bag7; set final; modelo='bag7'; run;

/*sin seleccion de variables*/
%cruzadarandomforestbin(archivo=bd.data,vardep=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,
maxtrees=100,variables=28,porcenbag=0.70,maxbranch=2,tam-
hoja=5,maxdepth=10,pvalor=0.2,
ngrupos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data bag8; set final; modelo='bag8'; run;

/*sin seleccion de variables*/
%cruzadarandomforestbin(archivo=bd.data,vardep=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
```

```
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,
maxtrees=100,variables=28,porcenbag=0.60,maxbranch=2,tam-
hoja=6,maxdepth=10,pvalor=0.15,
ngrupos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data bag9; set final; modelo='bag9'; run;


/*data union;*/
/*set bag1 bag2 bag3 bag4 bag5 bag6 bag7 bag8 bag9;*/
/*run;*/
/*proc boxplot data=union;*/
/*plot media*modelo;run;*/

/************************** RANDOM FOREST
***********************************/
/* Le especificamos las variables que queremos de salida */

/*6 variables*/
%cruzadarandomforestbin(archivo=bd.data,vardep=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,
maxtrees=100,variables=6,porcenbag=0.75,maxbranch=2,tam-
hoja=5,maxdepth=10,pvalor=0.1,
ngrupos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data rf1; set final; modelo='rf1'; run;

%cruzadarandomforestbin(archivo=bd.data,vardep=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,
maxtrees=100,variables=6,porcenbag=0.75,maxbranch=2,tam-
hoja=6,maxdepth=10,pvalor=0.2,
ngrupos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data rf2; set final; modelo='rf2'; run;

/*8 variables*/
%cruzadarandomforestbin(archivo=bd.data,vardep=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,
maxtrees=100,variables=8,porcenbag=0.75,maxbranch=2,tam-
hoja=5,maxdepth=10,pvalor=0.1,
ngrupos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
```

```
data rf3; set final; modelo='rf3'; run;

%cruzadarandomforestbin(archivo=bd.data,vardep=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,
maxtrees=100,variables=8,porcenbag=0.75,maxbranch=2,tam-
hoja=6,maxdepth=10,pvalor=0.2,
ngrupos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data rf4; set final; modelo='rf4'; run;

/*10 variables*/
%cruzadarandomforestbin(archivo=bd.data,vardep=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,
maxtrees=100,variables=10,porcenbag=0.75,maxbranch=2,tam-
hoja=5,maxdepth=10,pvalor=0.1,
ngrupos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data rf5; set final; modelo='rf5'; run;

%cruzadarandomforestbin(archivo=bd.data,vardep=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,
maxtrees=100,variables=10,porcenbag=0.75,maxbranch=2,tam-
hoja=6,maxdepth=10,pvalor=0.2,
ngrupos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data rf6; set final; modelo='rf6'; run;

/*12 variables*/
%cruzadarandomforestbin(archivo=bd.data,vardep=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,
maxtrees=100,variables=12,porcenbag=0.75,maxbranch=2,tam-
hoja=5,maxdepth=10,pvalor=0.1,
ngrupos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data rf7; set final; modelo='rf7'; run;

%cruzadarandomforestbin(archivo=bd.data,vardep=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
```

```
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,
maxtrees=100,variables=12,porcenbag=0.75,maxbranch=2,tam-
hoja=6,maxdepth=10,pvalor=0.2,
ngrupos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data rf8; set final; modelo='rf8'; run;

/*data union;*/
/*set rf1 rf2 rf3 rf4 rf5 rf6 rf7 rf8;*/
/*run;*/
/*proc boxplot data=union;*/
/*plot media*modelo;run;*/

/************ GRADIENT BOOSTING ***********/
/* LA MACRO CRUZADATREEBOOSTBIN REALIZA VALIDACIÓN CRUZADA REPE-
TIDA PARA VARIABLE
DEPENDIENTE BINARIA

PARÁMETROS:

leafsize=tamaño mínimo de hoja final
iteraciones
shrink=constante v de regularización
maxbranch=divisiones máximas en un nodo
maxdepth=máxima profundidad
mincatsize=minimo número de observaciones var. categórica
minobs=mínimo número de observaciones para dividir un nodo

criterion=ProbF,

*/


%macro cruzadatreeboostbin(archivo=,vardepen=,conti=,cate-
gor=,ngrupos=,sinicio=,sfinal=,leafsize=5,
iteraciones=100,shrink=0.01,maxbranch=2,maxdepth=4,min-
catsize=15,minobs=20,objetivo=tasafallos);
data final;run;
/* Bucle semillas */
%do semilla=&sinicio %to &sfinal;
     data dos;set &archivo;u=ranuni(&semilla);
     proc sort data=dos;by u;run;
     data dos ;
     retain grupo 1;
     set dos nobs=nume;
     if _n_>grupo*nume/&ngrupos then grupo=grupo+1;
     run;
     data fantasma;run;
     %do exclu=1 %to &ngrupos;
          data tres;set dos;if grupo ne &exclu then var-
dep=&vardepen;

     proc treeboost data=tres
     exhaustive=1000 intervaldecimals=max
```

```
      leafsize=&leafsize iterations=&iteraciones
maxbranch=&maxbranch
      maxdepth=&maxdepth mincatsize=&mincatsize missing=usein-
search shrinkage=&shrink
      splitsize=&minobs;
      %if (&categor ne) %then %do;
      input &categor/level=nominal;
      %end;
      input &conti/level=interval;
      target vardep /level=binary;
      save fit=iteraciones importance=impor model=modelo ru-
les=reglas;
      subseries largest;
      score out=sal;

      data sal2;set sal;pro=1-p_vardep0;if pro>0.5 then pre11=1;
else pre11=0;
            if grupo=&exclu then output;run;
            proc freq data=sal2;tables pre11*&varde-
pen/out=sal3;run;
            data estadisticos (drop=count percent pre11 &varde-
pen);
            retain vp vn fp fn suma 0;
            set sal3 nobs=nume;
            suma=suma+count;
            if pre11=0 and &vardepen=0 then vn=count;
            if pre11=0 and &vardepen=1 then fn=count;
            if pre11=1 and &vardepen=0 then fp=count;
            if pre11=1 and &vardepen=1 then vp=count;
            if _n_=nume then do;
            porcenVN=vn/suma;
            porcenFN=FN/suma;
            porcenVP=VP/suma;
            porcenFP=FP/suma;
            sensi=vp/(vp+fn);
            especif=vn/(vn+fp);
            tasafallos=1-(vp+vn)/suma;
            tasaciertos=1-tasafallos;
            precision=vp/(vp+fp);
            F_M=2*Sensi*Precision/(Sensi+Precision);
            output;
            end;
            run;

            data fantasma;set fantasma estadisticos;run;
      %end;
      proc means data=fantasma sum noprint;var &objetivo;
      output out=sumaresi sum=suma mean=media;
      run;
      data sumaresi;set sumaresi;semilla=&semilla;
      data final (keep=suma media semilla);set final sumaresi;if
suma=. then delete;run;
%end;
proc print data=final;run;
%mend;
```

```
%cruzadatreeboostbin(archivo=bd.data,vardepen=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,
ngrupos=5,sinicio=12345,sfinal=12400,leafsize=5,
iteraciones=100,shrink=0.01,maxbranch=2,maxdepth=4,min-
catsize=15,minobs=20,objetivo=tasafallos);
data gb1;set final;modelo='gb1'; run;

%cruzadatreeboostbin(archivo=bd.data,vardepen=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,leafsize=4,
iteraciones=100,shrink=0.01,maxbranch=2,maxdepth=5,min-
catsize=15,minobs=20,objetivo=tasafallos);
data gb2;set final;modelo='gb2'; run;

%cruzadatreeboostbin(archivo=bd.data,vardepen=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,leafsize=3,
iteraciones=100,shrink=0.01,maxbranch=2,maxdepth=6,min-
catsize=15,minobs=20,objetivo=tasafallos);
data gb3;set final;modelo='gb3'; run;

%cruzadatreeboostbin(archivo=bd.data,vardepen=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,leafsize=5,
iteraciones=100,shrink=0.02,maxbranch=2,maxdepth=4,min-
catsize=15,minobs=20,objetivo=tasafallos);
data gb4;set final;modelo='gb4'; run;

%cruzadatreeboostbin(archivo=bd.data,vardepen=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,leafsize=4,
```

```
iteraciones=100,shrink=0.02,maxbranch=2,maxdepth=5,min-
catsize=15,minobs=20,objetivo=tasafallos);
data gb5;set final;modelo='gb5'; run;

%cruzadatreeboostbin(archivo=bd.data,vardepen=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,leafsize=3,
iteraciones=100,shrink=0.02,maxbranch=2,maxdepth=6,min-
catsize=15,minobs=20,objetivo=tasafallos);
data gb6;set final;modelo='gb6'; run;

%cruzadatreeboostbin(archivo=bd.data,vardepen=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,leafsize=5,
iteraciones=100,shrink=0.01,maxbranch=2,maxdepth=4,min-
catsize=15,minobs=20,objetivo=tasafallos);
data gb7;set final;modelo='gb7'; run;

%cruzadatreeboostbin(archivo=bd.data,vardepen=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,leafsize=4,
iteraciones=100,shrink=0.01,maxbranch=2,maxdepth=5,min-
catsize=15,minobs=20,objetivo=tasafallos);
data gb8;set final;modelo='gb8'; run;

%cruzadatreeboostbin(archivo=bd.data,vardepen=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,leafsize=3,
iteraciones=100,shrink=0.01,maxbranch=2,maxdepth=6,min-
catsize=15,minobs=20,objetivo=tasafallos);
data gb9;set final;modelo='gb9'; run;

%cruzadatreeboostbin(archivo=bd.data,vardepen=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
```

```
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,leafsize=7,
iteraciones=100,shrink=0.03,maxbranch=2,maxdepth=7,min-
catsize=15,minobs=20,objetivo=tasafallos);
data gb10;set final;modelo='gb10'; run;

/*data union;*/
/*set gb1 gb2 gb3 gb4 gb5 gb6 gb7 gb8 gb9 gb10;*/
/*run;*/
/*proc boxplot data=union;*/
/*plot media*modelo;run;*/

/************************ ENSAMBLADO
*****************************/
/*
MACRO CRUZADASTACK PARA BINARIA

HACE VALIDACIÓN CRUZADA CON LOS SIGUIENTES MÉTODOS:

RED NEURONAL (parámetro nodos de la macro; cualquier otra espe-
cificación
                      de la red como algoritmo, iteraciones, ac-
tivación, etc. se cambia dentro del código)
LOGÍSTICA

RANDOM FOREST

GRADIENT BOOSTING (párámetros itera y v)

1) LA MACRO SE PUEDE CAMBIAR A CONVENIENCIA INTERNAMENTE, SOBRE
TODO LOS PARÁMETROS DE LA RED NEURONAL, boosting, etc.

2) SI NO HAY VARIABLES DE CLASE EN EL ARCHIVO:

     A) QUITAR TODOS LOS APARTADOS CLASS O PONER * AL PRINCIPIO
PARA QUE APAREZCA COMO COMENTARIO
     B) BORRAR &listclass DE TODA LA MACRO

*/


%macro cruzadastack
(archivo=,vardepen=,listclass=,listconti=,ngrupos=,semi-
nicio=,semifinal=,
nodos=,algo=,rediter=,/*red*/
maxtrees=,vars_to_try=,trainfraction=,leafsize=,maxdepth=,/*ran-
dom forest */
bleafsize=,iterations=,bmaxbranch=,bmaxdepth=,shrinkage=/* g
boosting*/);
data final;run;
proc printto print='C:\Users\Lorena\Documents\TFM\ensam-
blado\ca.txt' log='C:\Users\Lorena\Documents\TFM\ensam-
blado\loga.txt';
```

```
%do semilla=&seminicio %to &semifinal;/*<<<<<******AQUI SE PUE-
DEN CAMBIAR LAS SEMILLAS */
data dos;set &archivo;u=ranuni(&semilla);
proc sort data=dos;by u;run;

data dos (drop=nume);
retain grupo 1;
set dos nobs=nume;
if _n_>grupo*nume/&ngrupos then grupo=grupo+1;
run;

data fantasma;run;
data unionsalfin;run;
data unifin;run;

%do exclu=1 %to &ngrupos;

data tres;set dos;if grupo ne &exclu then vardep=&varde-
pen*1;run;


/*************************************************************/
/* LOGISTICA */
proc logistic data=tres noprint;/*<<<<<******SE PUEDE QUITAR EL
NOPRINT */
class Producto Empresa Fecha_semana mes;
model vardep=Producto Empresa Fecha_semana mes sub_dia_op1
sub_sem_op2 max_ano_op1 N_sin_subasta;
score out=saco;
;run;
/*************************************************************/

data sal1 (drop=p_1);set saco;predi1=p_1;run;



/*************************************************************/
/*RED */
PROC DMDB DATA=tres dmdbcat=catatres;
target vardep ;
var sub_dia_op1 sub_sem_op2 max_ano_op1 N_sin_subasta;
class vardep Producto Empresa Fecha_semana mes;
;run;

proc neural data=tres dmdbcat=catatres ;
input sub_dia_op1 sub_sem_op2 max_ano_op1 N_sin_subasta/ id=i;
input Producto Empresa Fecha_semana mes /level=nominal;
target vardep/ id=o level=nominal;
hidden &nodos/ id=h act=tanh;
netoptions randist=normal ranscale=0.15 random=15459;
prelim 15 preiter=10 ;
train maxiter=&rediter technique=&algo;
score data=tres out=salred;
run;
```

```
data sal2 (keep=predi2 grupo vardep);set salred;predi2=p_var-
dep1;run;


/*************************************************************/
/*RANDOM FOREST*/
/*************************************************************/

proc hpforest data=tres
maxtrees=&maxtrees vars_to_try=&vars_to_try trainfrac-
tion=&trainfraction leafsize=&leafsize maxdepth=&maxdepth
exhaustive=5000
missing=useinsearch ;
target vardep/level=nominal;
input &listconti/level=interval;
input &listclass/level=nominal;
score out=salo;
run;

data sal3 (keep=&vardepen predi3 grupo vardep);set
salo;predi3=p_vardep1;run;



/*************************************************************/
/*GRADIENT BOOSTING */
/*************************************************************/

     proc treeboost data=tres
     exhaustive=1000 intervaldecimals=max
     leafsize=&bleafsize iterations=&iterations
maxbranch=&bmaxbranch
     maxdepth=&bmaxdepth mincatsize=10 missing=useinsearch
shrinkage=&shrinkage
     splitsize=10;
     input &listclass/level=nominal;
     input &listconti/level=interval;
     target vardep /level=binary;
     subseries largest;
     score out=salboost;
run;
data sal4 (keep=predi4 grupo vardep);set salboost;predi4=p_var-
dep1;run;

data unionsal ;merge sal1 sal2 sal3 sal4;
run;

data salfin (keep=&vardepen vardep predi1 predi2 predi3 predi4
grupo);set unionsal;if grupo=&exclu then output;run;

data salfin(keep=&vardepen vardep predi1-predi16);set salfin;
predi5=(predi1+predi2)/2; /* LOG-RED */
predi6=(predi1+predi3)/2;/* LOG-RFOR */
predi7=(predi1+predi4)/2;/* LOG-BOOST*/
predi8=(predi2+predi3)/2;/* RED-RFOR */
predi9=(predi2+predi4)/2;/* RED-BOOST */
```

```
predi10=(predi3+predi4)/2;/* RFOR-BOOST */
predi11=(predi1+predi2+predi3)/3;/* LOG-RED-RFOR */
predi12=(predi1+predi2+predi4)/3;/* LOG-RED-BOOST*/
predi13=(predi1+predi3+predi4)/3;/* LOG-RFOR-BOOST*/
predi14=(predi2+predi3+predi4)/3;/* RED-RFOR-BOOST*/
predi15=(predi1+predi2+predi3+predi4)/4;/* LOG-RED-RFOR-BOOST*/
predi16=(predi1*0.2+predi2*0.1+predi3*0.5+predi4*0.2);/* LOG-
RED-RFOR-BOOST ponderado*/
run;

data unionsalfin;set unionsalfin salfin;run;

data salbis;
array predi{16};
array pre{16};
set salfin;
do i=1 to 16;
if predi{i}>0.5 then pre{i}=1;
if predi{i}<=0.5 then pre{i}=0;
end;
run;
data salbos;run;
%do j=1 %to 16;
proc freq data=salbis noprint;tables pre&j*&vardepen /out=sal-
confu;run;
data confu&j (keep=tasa&j);retain buenos 0 malos 0;set salconfu
nobs=nume;
if &vardepen=pre&j then buenos=buenos+count;
if &vardepen ne pre&j then malos=malos+count;
if _n_=nume then do;tasa&j=malos/(malos+buenos);output;end;
run;
data salbos;merge salbos confu&j;run;
;
%end;

data fantasma;set fantasma salbos;run;

%end;
/* FIN GRUPOS */
proc means data=fantasma noprint;var tasa1-tasa16;
output out=mediaresi mean=ase1-ase16;
run;
data mediaresi;set mediaresi;semilla=&semilla;run;
data final (keep=ase1-ase16 semilla);set final mediaresi;if
ASE1=. then delete;run;

data unifin;set unifin unionsalfin;run;
%end;
proc printto; run;
proc print data=final;run;
%mend;

options mprint=0;
%cruzadastack
(archivo=bd.data,vardepen=precio_semana,listclass=Producto Em-
presa Fecha_semana mes,
```

```
listconti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
ngrupos=5,seminicio=12345,semifinal=12400,
nodos=4,algo=levmar,rediter=100,/*red*/
maxtrees=100,vars_to_try=8,trainfraction=0.75,leaf-
size=6,maxdepth=10,/*random forest */
bleafsize=7,iterations=100,bmaxbranch=2,bmaxdepth=7,shrin-
kage=0.03 /* g boosting*/);

data cajas;
array ase{16};
set final;
do i=1 to 16;
modelo=i;
error=ase{i};
output;
end;
run;

/* EN ESTAS OPCIONES SE CAMBIA LA LETRA Y LA ALTURA DEL TEXTO EN
LOS EJES CON HTEXT.*/
options font="Courier New" bold 8;
run;goptions htext=8pt;

proc sort data=cajas;by modelo;
data eti;length eti $ 13;
input modelo eti $;
cards;
1 LOG
2 RED
3 RF
4 GB
5 LOG-RED
6 LOG-RF
7 LOG-GB
8 RED-RF
9 RED-GB
10 RF-GB
11 L-R-RF
12 L-R-BG
13 L-RF-GB
14 R-RF-GB
15 LRRFGB
16 (LRRFGB)P
;
data cajas2;merge cajas eti;by modelo;
title1
h=2 box=1 j=c c=red j=c ;

options font="Courier New" bold 10;
run;goptions htext=6pt;

ods graphics off;
```

```
proc boxplot data=cajas2;plot error*ETI /
cboxes         = dagr
cboxfill       = ywh
vaxis=0.20 to 0.35 by 0.01 ;run;

/*EVALUACION V CONTINUA*/

%randomselect(data=bd.data,listclass=Producto Empresa Fecha_se-
mana mes,vardepen=X1,
                modelo=sub_dia_op1 sub_sem_op1 sub_dia_op2
sub_sem_op2 sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1
min_dia_op1 avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1
max_ano_op1 min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2
avg_sem_op2 max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2
min_ano_op2 Producto Empresa Fecha_semana mes,
                metodo=stepwise,criterio=AIC,sinicio=12345,sfi-
nal=12456,fracciontrain=0.8,directorio=C:\Users\Lorena\Docu-
ments\TFM\log);
%cruzada(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op2 N_sin_subasta,categor=Producto Empresa Fecha_semana
mes,ngrupos=5,sinicio=12345,sfinal=12400);
data regresion2;set final;modelo='reg2';run;
%cruzada(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op1 sub_dia_op2 sub_sem_op2 sub_ano_op2 N_sin_subasta
avg_dia_op1 max_dia_op1 min_dia_op1 avg_sem_op1 max_sem_op1
min_sem_op1 avg_ano_op1 max_ano_op1 min_ano_op1 avg_dia_op2
max_dia_op2 min_dia_op2 avg_sem_op2 max_sem_op2 min_sem_op2
avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400);
data regresionb;set final;modelo='regbasico';run;


%cruzadaneural(archivo=bd.data,vardepen=X1,conti=sub_dia_op1
sub_sem_op2 N_sin_subasta,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,ocultos=2,algo=levmar,acti=log,early=);
data red6;set final;modelo='red6';run;

/*Comparamos las modelos v continua*/
data union;set regresion2 regresionb;
proc boxplot data=union;plot media*modelo;run;

/*EVALUACION V BINARIA*/

%cruzadalogistica(archivo=bd.data,vardepen=precio_semana,cate-
gor=Producto Empresa Fecha_semana mes,
conti=sub_dia_op1 sub_sem_op2 max_ano_op1 N_sin_subasta,ngru-
pos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data logistica2;set final;modelo='log2';run;
%cruzadabinarianeural(archivo=bd.data,vardepen=precio_se-
mana,conti=sub_dia_op1 sub_sem_op2 max_ano_op1 N_sin_subasta,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,nodos=4,algo=levmar,acti=tanh,obje-
tivo=tasafallos);
```

```
data red3;set final;modelo='red3';run;
%cruzadarandomforestbin(archivo=bd.data,vardep=precio_se-
mana,conti=sub_dia_op1 sub_sem_op2 max_ano_op1 N_sin_subasta,
categor=Producto Empresa Fecha_semana mes,
maxtrees=100,variables=8,porcenbag=0.60,maxbranch=2,tam-
hoja=6,maxdepth=10,pvalor=0.15,
ngrupos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data bag3; set final; modelo='bag3'; run;
%cruzadarandomforestbin(archivo=bd.data,vardep=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,
maxtrees=100,variables=8,porcenbag=0.75,maxbranch=2,tam-
hoja=6,maxdepth=10,pvalor=0.2,
ngrupos=5,sinicio=12345,sfinal=12400,objetivo=tasafallos);
data rf4; set final; modelo='rf4'; run;
%cruzadatreeboostbin(archivo=bd.data,vardepen=precio_se-
mana,conti=sub_dia_op1 sub_sem_op1 sub_dia_op2 sub_sem_op2
sub_ano_op2 N_sin_subasta avg_dia_op1 max_dia_op1 min_dia_op1
avg_sem_op1 max_sem_op1 min_sem_op1 avg_ano_op1 max_ano_op1
min_ano_op1 avg_dia_op2 max_dia_op2 min_dia_op2 avg_sem_op2
max_sem_op2 min_sem_op2 avg_ano_op2 max_ano_op2 min_ano_op2,
categor=Producto Empresa Fecha_semana mes,ngrupos=5,si-
nicio=12345,sfinal=12400,leafsize=7,
iteraciones=100,shrink=0.03,maxbranch=2,maxdepth=7,min-
catsize=15,minobs=20,objetivo=tasafallos);
data gb10;set final;modelo='gb10'; run;

/*Comparamos las modelos v binaria*/
data union;set logistica2 red3 bag3 rf4 gb10;
proc boxplot data=union;plot media*modelo;run;
```