Algoritmos de Clasificación para el apoyo en el diagnóstico de celiaquía

Classification Algorithms to support the diagnosis of celiac disease

Cristian Emanuel Anei
Jennifer Marmolejos Urbáez
Jennifer Zapata Arciénega

Grado en Ingeniería Informática
Universidad Complutense de Madrid

Trabajo Fin de Grado

15 de septiembre de 2022

Tutores

Mercedes García Merayo

Resumen en castellano

La mayoría de las personas que padecen la enfermedad de celiaquía están sin diagnosticar, ya sea porque los síntomas se confunden con otras enfermedades o porque el resultado de las pruebas tiene un retraso excesivo. Este proyecto tiene como objetivo minimizar este problema, facilitando el diagnóstico e identificando la enfermedad en pacientes que no siguen una clínica convencional. Para esto se unirá la medicina con la inteligencia artificial.

De esta manera se realizará un análisis detallado de los datos de los pacientes obtenidos a través de los médicos, desarrollando un algoritmo de *Machine Learning* para la identificación y clasificación de la información, centrando en los algoritmos de clasificación.

Palabras clave

Algoritmo de clasificación, *Machine Learning*, Celiaquiá

Abstract

Nowadays, most people with celiac disease are not diagnosed. That is either because symptoms are confused with those of other diseases or just because the results of the tests are excessively delayed. The aim of this project is to try to minimize this problem by facilitating diagnosis and identifying the disease in patients who do not follow a common clinic. Medicine will be combined with artificial intelligence for this purpose.

That is how a detailed analysis of the patient data obtained through doctors will be carried out: by developing a machine learning algorithm in this way that will be useful for the identification and classification of information (mostly focused on classification algorithms).

Keywords

Classification Algorithm, Machine Learning, Celiac Disease

Índice general

Capítulo 1

Introducción

En este capítulo presentamos brevemente la motivación detrás del desarrollo de este Trabajo de Fin de Grado, el objetivo que queremos alcanzar y la estructura del resto de la memoria.

1.1. Motivación

En la actualidad, se ha detectado que el 1 % de la población mundial padece la enfermedad celiaquía según la OMS. Las personas con esta enfermedad no absorben el gluten, y suelen presentar daños en el intestino delgado junto a otros síntomas que hacen que la vida de la persona sea más complicada. Se estima que un porcentaje muy elevado de pacientes (75 %) están sin diagnosticar, debido a que los síntomas que produce suelen aparecer en otras enfermedades.

La confirmación del diagnóstico suele retrasarse por diversos motivos, como la imposibilidad de asistencia médica para analizar los datos del paciente, o la lentitud de obtención de los resultados. Por ello, la introducción de *Inteligencia Artificial* puede ayudar, con un estudio previo de la información, a un diagnóstico de la enfermedad. Además, no sólo ayuda en la resolución final, sino que también, puede servir como seguimiento de un paciente sin síntomas, o una posible evolución de la enfermedad en pacientes que no la tienen pero

presentan los síntomas.

Como resultado de todo ello y partiendo como base de un trabajo realizado con anterioridad sobre algoritmos de *clustering*, en este proyecto nos centraremos en facilitar el diagnóstico de la enfermedad mucho más rápido, utilizando algoritmos de *Machine Learning* de clasificación, con el fin de aprovechar de forma más eficiente los datos que tenemos, comprobando cada componente y su posible utilidad en el algoritmo, así como también seleccionaremos el que mejor resultados nos pueda dar.

1.2. Objetivos

El objetivo de este proyecto es poder complementar los desarrollos de modelos de *Machine Learning* que se han realizado con anterioridad sobre este estudio clínico. En pocas palabras, se trata de buscar un diagnóstico de la enfermedad en pacientes que no siguen una clínica convencional.
De partida contamos con una base de datos que se debe analizar para conseguir el máximo provecho de ella. Esta base de datos contiene información de dos tipos de pacientes: los que padecen la enfermedad celiaca y los que no. Con esta información se proyectan los siguientes desafíos:

- Análisis de la base de datos: en este paso se profundizará en los diferentes atributos, comprobando su utilidad en otros estudios médicos y en proyectos anteriores. Es lo que se conoce cómo ingeniería de atributos.
- Aumentar la utilidad de los atributos que se han definido como útiles para el proyecto.
- Utilizar *Machine Learning* para agilizar la búsqueda de los algoritmos que mejor resultado puedan dar.
- Selección y definición de los algoritmos de *Machine Learning* a utilizar.
- Desarrollo, implementación y puesta en producción de los algoritmos.

- Selección de métricas de puntuación de los diferentes algoritmos de *Machine Learning*.
- Análisis de resultados y exposición de posibles mejoras para siguientes lineas de trabajo.

1.3. Estructura de la memoria

El resto de la memoria en la que se presenta este Trabajo de Fin de Grado está estructurada en los siguientes capítulos:

- *Formateo de los datos.*
- *Fundamentos de los algoritmos utilizados.*
- *Métodos de evaluación.*
- *Resultados.*
- *Contribuciones individuales al proyecto.*
- *Conclusiones y trabajos futuros.*

Capítulo 1

Introduction

In this chapter, we will briefly present the aim behind the development of this Final Degree Project, the objective we want to achieve and the structure of the rest of the memory

1.1. Aim

Nowadays, it is known that 1 % of the total world population suffers from celiac disease, according to the World Health Organization (WHO). People suffering from this disease do not absorb gluten, and usually have the small intestine damaged, along with other symptoms that make life difficult for them. It is estimated that a large percentage of patients (almost 75 %) are not diagnosed, because the symptoms produced by the celiac disease can be confused with other diseases.

Confirmation of the diagnosis is often delayed for various reasons, such as absence of medic assistance for patient data analysis, or even the time that it takes to obtain the results. Thus, implementation of Artificial Intelligence may help to obtain a diagnosis of the Disease, with a prior study of the information. Furthermore, it not only helps in the final resolution, but it also may be useful as an asymptomatic patient tracking, or a possible evolution of the disease in patients who do not have it but present the symptoms.

As a result, and having work carried out with prior knowledge about clustering algo-

rithms as a reference, this project will be focused on making the diagnosis of the disease much faster, using Machine Learning classification algorithms in order to make efficient use of the available data, checking each component and its possible usefulness in the algorithm, as well as choosing the one that best suits our data.

1.2. Objectives

The objective of this project is to be able to complement the developments of Machine Learning that have been done previously about this clinical study. In brief, it is about finding a diagnosis of the disease in patients who do not follow a conventional clinic.

Firstly, we have a database that must be analyzed to make the most of the database. This database contains information on two types of patients: those who have celiac disease and those who don't. With the given information, the next challenges are presented:

- Database analysis: in this step, the different attributes will be researched, proving its usefulness in other medical studies and in previous projects. This is known as is known as feature engineering.
- Increase the usefulness of the attributes that have been defined as useful for the project.
- Take advantage of Machin Learnig to improve the search for the algorithms that best perform result they can give.
- Selection and definition of the Machine Learning algorithms to be used.
- Development, implementation and start-up of algorithms.
- Scores selection of different Machine Learning algorithms.
- Analysis of results and presentation of possible improvements for the following lines of work.

1.3. Work Structure

The rest of this paper is divided in the following chapters:

- *Data formatting*
- *Used technologies*
- *Theorical basis of the clustering algorithms*
- *Theoretical basis of evaluation methods*
- *Results*
- *Individual contributions to the project*
- *Conclusions and future work:*

Capítulo 2

Formateo de los datos

En este segundo capítulo se presenta la preparación de los datos con la finalidad de construir unos datos organizados en una estructura establecida y conocida llamada *ABT o Tabla de base de análisis*. Esta fase contiene todas las actividades para convertir y transformar los recursos obtenido en dicha tabla o dataset. Además, incluiremos tras la transformación y manipulación de los datos un reporte de la calidad de los datos recogidos.

Para familiarizarnos con los datos es muy común hacer un análisis exploratorio con gráficos y estadísticas descriptivas. Además este análisis nos ayuda a comprender y garantizar que la información que tenemos es suficiente para realizar nuestro modelo.

2.1. Estudio del dataset

En esta faceta inicial, los primeros pasos que seguidos fueron: conocer la terminología médica de la enfermedad tratada, analizar, realizar un perfilado del dataset del que disponíamos y comprobar que esta información era apropiada para los tipos de modelos algorítmicos de Machine Learning que se va a aplicar en el proyecto.

Si estuviésemos en el caso de que el Machine Learning fuese lo suficientemente avanzado, la información que tenemos podría ser útil, pero actualmente no es así, y estamos en el punto

de que la cantidad de información es escasa, ya que partimos de un archivo de poco peso. Esto a primera vista podría no ser un problema, según un estudio realizado por Michele Banko y Eric Brill, investigadores de Microsoft, en el que concluyeron según los resultados que "es mejor reconsiderar gastar más recursos económicos en adquirir una datos útiles que en los propios algoritmos que se van a utilizar" [3].

Por ello, gran parte del proyecto se centrará en reorganizar los datos que vamos a analizar para que sean de utilidad. Este proceso inicial se conoce como *Feature Engineering*, el cual utilizaremos en los apartados siguientes.

2.1.1. Estudio de *profiling*

En este apartado se va a examinar y visualizar la información del dataset que nos han facilitado. Utilizaremos una herramienta de *profiling* que nos permitirá realizar un análisis estadístico de la calidad de los datos, eso nos dará la capacidad de comprender de qué tipo de datos disponemos. Nuestra idea es determinar qué datos no son representativos o esenciales, ya sea por pacientes con mucha información incompleta o con datos irrelevantes, que nos pueden perjudicar a la hora de aplicar nuestro algoritmo; también evitamos que los datos tengan *bias*, es decir, que alguna de las características con las que trabajamos tengan prioridad sobre otras. Para ello normalizaremos la información.

Para realizar este análisis, utilizamos la biblioteca de pandas *Profiling* [1], un módulo Python de código abierto. Esto nos ayudó en la manipulación de la información con un análisis exhaustivo del dataset, facilitando la comprensión de este, creando un informe interactivo en un documento HTML.

Para empezar creamos un informe a partir de la base de datos completa, que contiene la información sin manipular extraída directamente de los datos proporcionados por los médicos. El informe generado se divide en cuatro secciones. La primera sección, que se muestra en la Figura 2.1, contiene la información general de la base de datos. Esta consiste en un resumen de estadísticas que incluye el número de variables, el porcentaje de celdas

[1] https://pandas-profiling.ydata.ai/docs/master/index.html

vacías o filas duplicadas y el tamaño total de los datos. Uno de los apartados más importantes de esta sección son los *warnings*, ya que indican las variables que tienen una gran pérdida de datos o correlación con otras variables y alertas sobre cardinalidad.

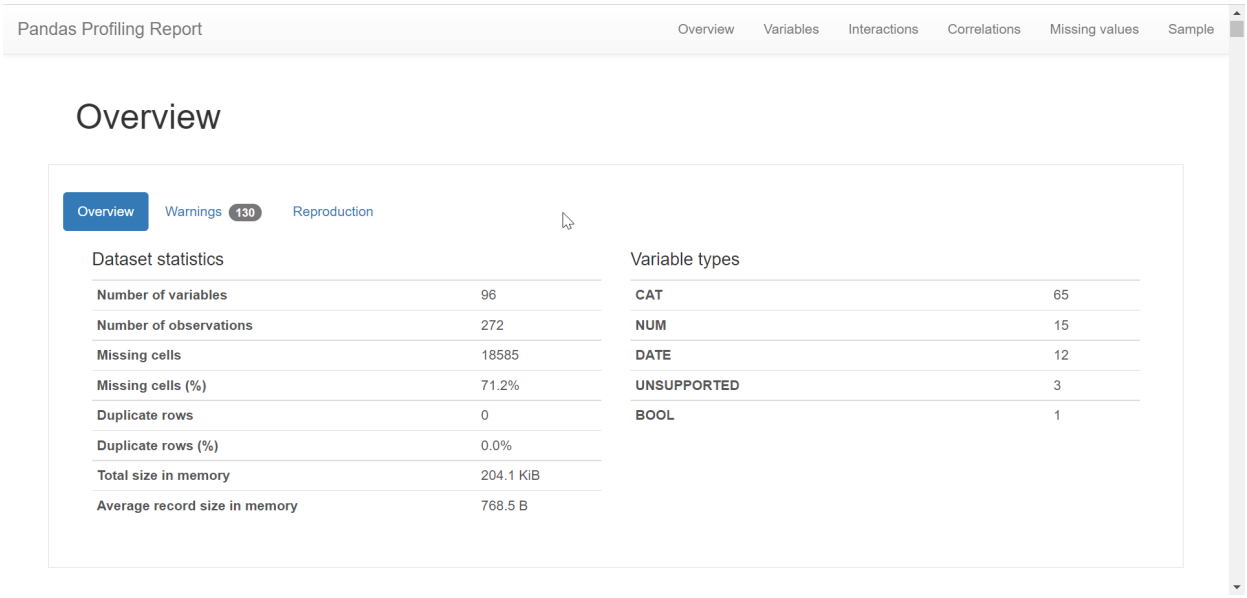


Figura 2.1: Estadísticas generales

La siguiente sección nos muestra un análisis detallado de todas las variables que contiene la base de datos, dependiendo del tipo de variable la información que se muestre será diferente. Esta sección es la que más hemos utilizado para comprender los datos.

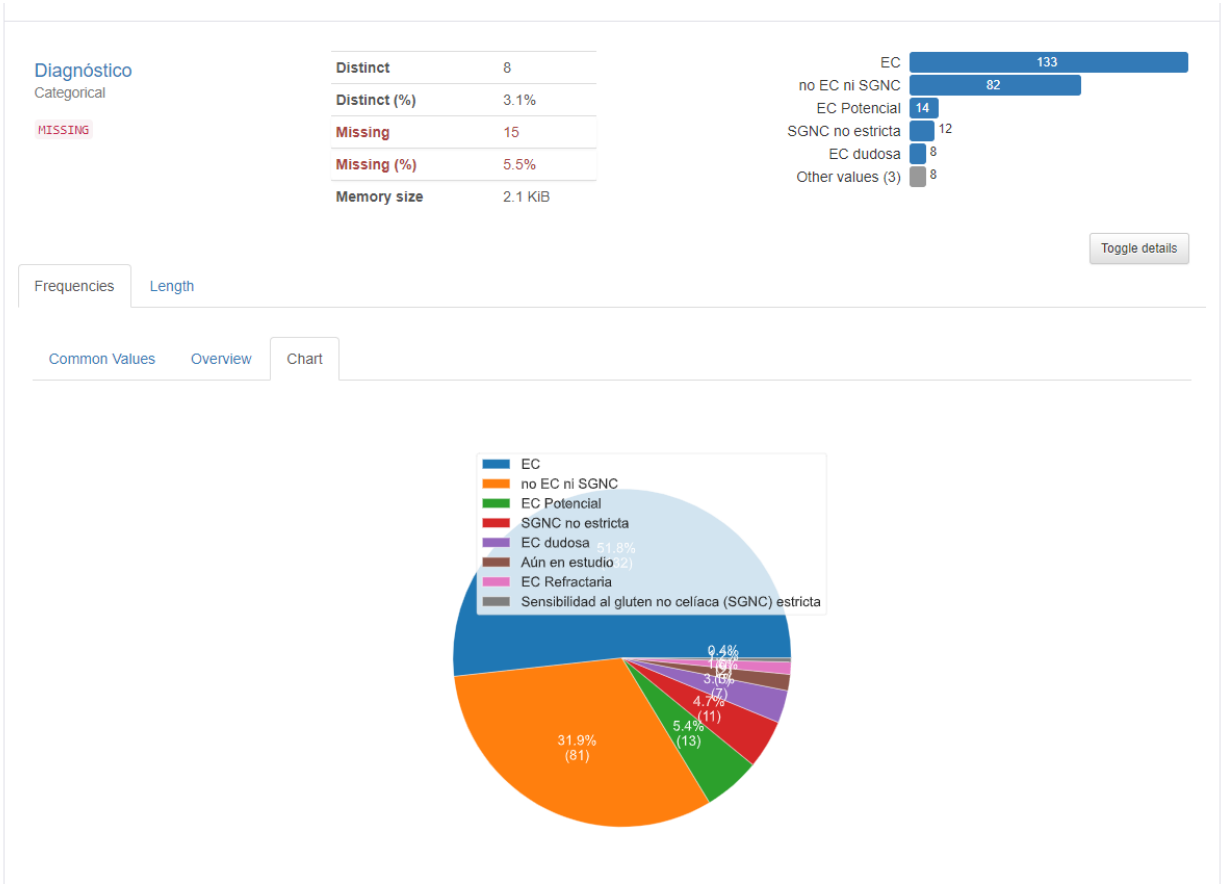


Figura 2.2: Variable tipo *string*

En la Figura 2.2 muestra las variables tipo *string* donde se pueden ver los valores únicos, el porcentaje de diferencias, el porcentaje de pérdida de datos y el tamaño de los datos. Además, realiza una representación histográfica de los valores frecuentes o una representación en gráfico circular.

En cuanto a las variables de tipo numéricas, estas no presentan la misma información que las anteriores, ya que se añade el cálculo de los valores mínimos y máximos, además de la media aritmética, los valores negativos y la cantidad de ceros que contiene. Esta información viene dada por histogramas como se muestra en la Figura 2.3.

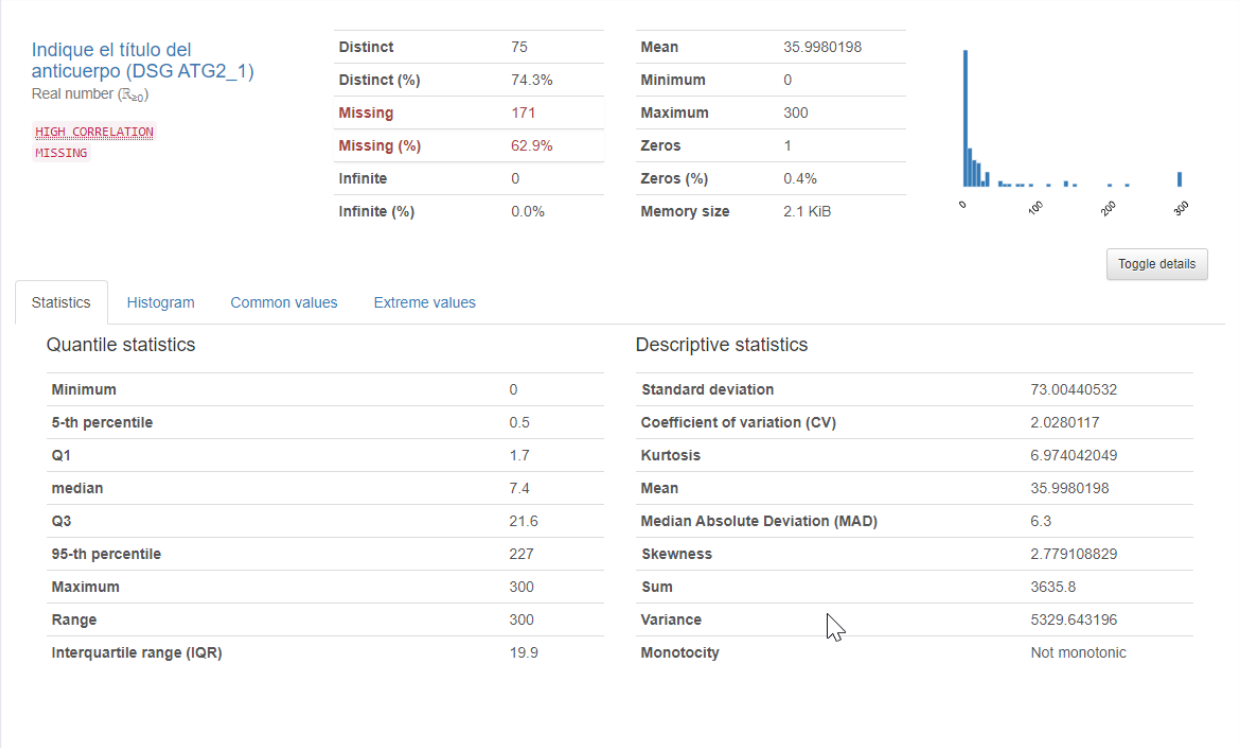


Figura 2.3: Variable tipo numérica

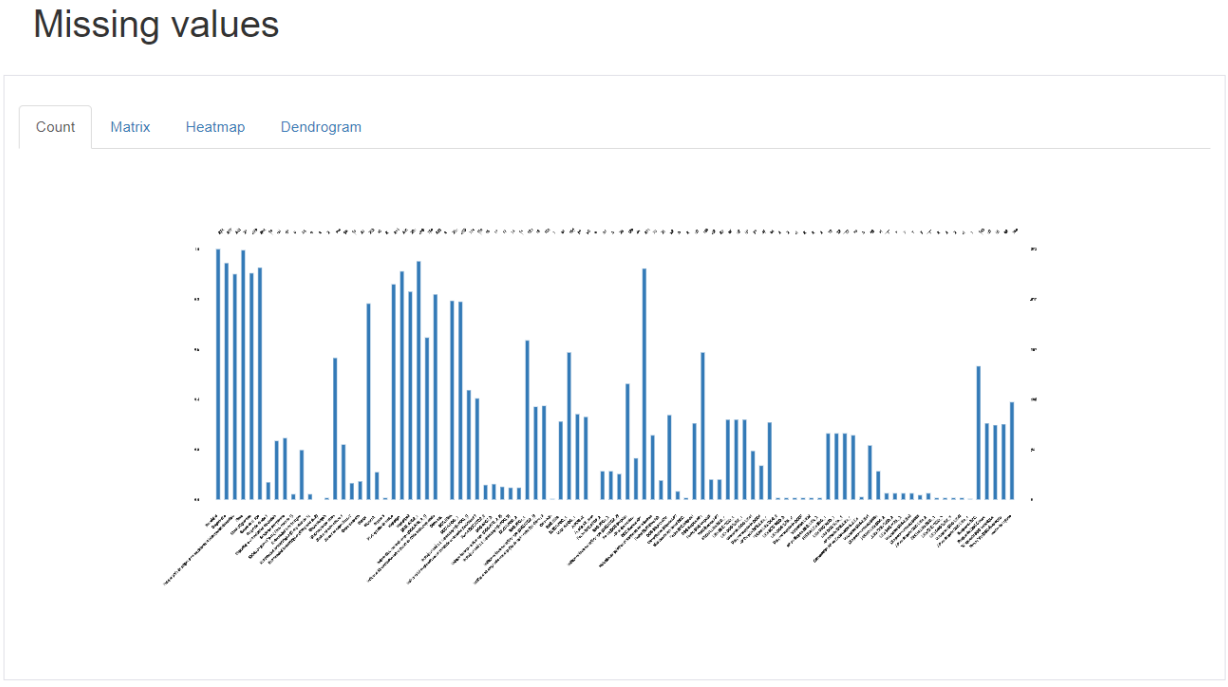


Figura 2.4: Información perdida

La tercera sección del informe consiste en las correlaciones de los datos, es decir, la relación lineal que hay entre las variables. En el análisis del informe inicial nos hemos dado cuenta de la gran cantidad de valores atípicos, repetidos, nulos o vacíos que contenía el dataset. Estos errores provocan que existan muchas correlaciones falsas entre las variables, dándonos información inexacta que no es de utilidad, por lo cual este apartado del reporte es irrelevante.

Por último, en la Figura 2.4, se representa la pérdida de información del dataset completo en una gráfica.

2.1.2. Estudio de la base de datos

Tras el análisis del *profiling*, encontramos columnas con mucha pérdida de información y que por lo tanto no son relevantes para el proyecto. De manera que, partiendo de la base de datos completa, y tras un estudio detallado de las columnas del dataset, hemos decidido quedarnos con las columnas que consideraremos a continuación:

- *Diagnóstico(Figura 2.5):* Es la columna etiqueta, es decir, aquella que usaremos en el algoritmo para clasificar si padece o no celiaquía según las pruebas, o si tiene síntomas pero no sufre esta enfermedad.

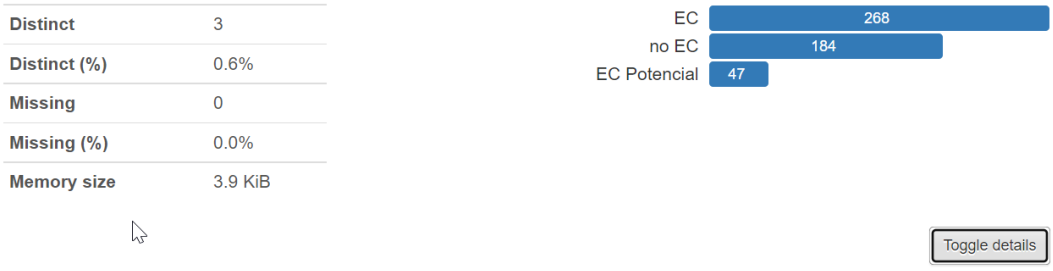


Figura 2.5: Análisis del diagnóstico

- *País de origen o en su defecto la información disponible(Figura 2.6):* como su nombre indica contiene la información del país de origen del paciente.

Figura 2.6: Análisis del país de origen

- *Sexo(Figura 2.7):* Es importante conocer el género del paciente ya que la enfermedad se desarrolla de forma diferente en hombres y mujeres, por ejemplo, las mujeres en estado de gestación son propensas a padecer la enfermedad.

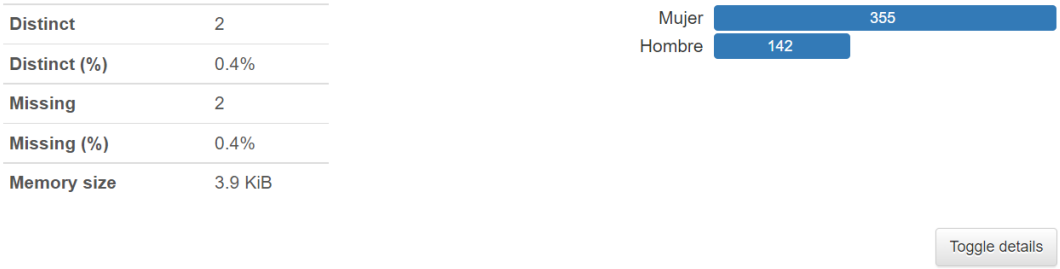


Figura 2.7: Análisis del sexo

- *Edad del paciente(Figura 2.8):* La enfermedad puede aparecer a cualquier edad, pero se suele diagnosticar con más frecuencia entre los 20 y 40 años, por ello esta variable es significativa ya que nuestro grupo de estudio tiene una media de edad de 39 años.

Figura 2.8: Análisis de la edad

- *Grupo de riesgo(Figura 2.9):* Personas que debido a distintos factores tienen más riesgo al contraer la enfermedad. La celiaquía tiene carácter autoinmune y por lo tanto las personas con otras enfermedades de este tipo son grupo de riesgo. Otro de los principales grupos de riesgo son por predisposición genética, sobre todo si se trata de familiares de primer grado.

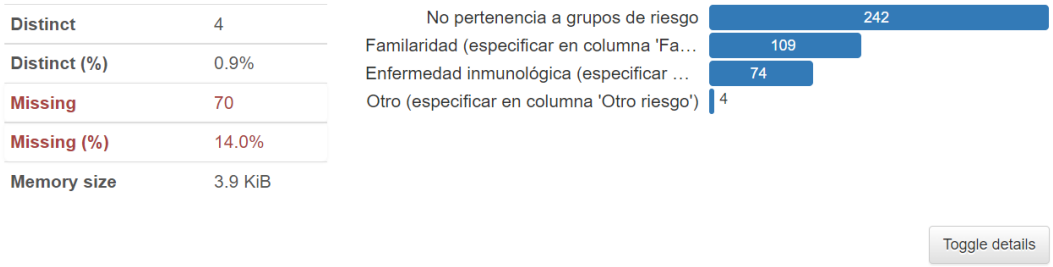


Figura 2.9: Análisis del grupo de riesgo

- *Otro/s riesgo/s(Figura 2.10):* Esta variable contiene otros riesgos que no son tan comunes, pero que son necesarios para nuestro estudio, como el Síndrome de Turner, Lupus eritematoso sistémico (LES) , o Hipogammaglobulinemia.

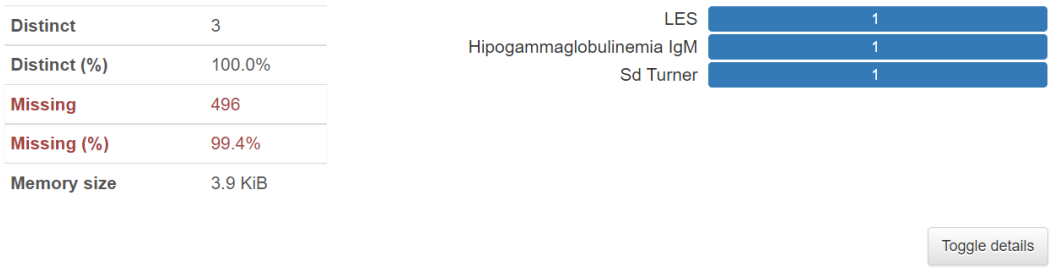


Figura 2.10: Análisis de otros riesgos

- *Grado de parentesco(Figura 2.11):* Grado de parentesco del familiar que padece la enfermedad celiaca si es que lo tiene. Es interesante conocer si hay familiares que padecen la enfermedad ya que, sin ser una enfermedad hereditaria, hay mayor riesgo a padecerla si hay familiares afectados. A través de estudios se ha concluido que si el 7,06 % de los progenitores padece esta enfermedad, habrá más probabilidad de que se herede [1].

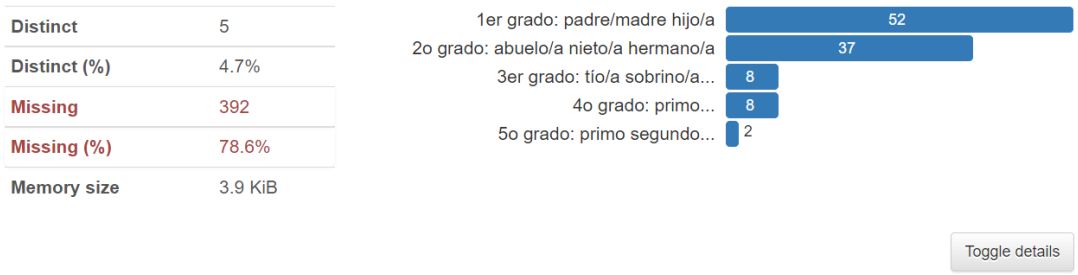


Figura 2.11: Análisis del parentesco

- *Enfermedad inmunológica(Figura 2.12):* Este campo incluye las enfermedades que sufre el paciente, al ser enfermedades sistémicas afecta a todos los órganos del cuerpo, provocando que los síntomas puedan ser muy variados. En concreto, las enfermedades digestivas autoinmunes como la enfermedad de Hashimoto, que afecta al intestino, puede ser una afección altamente relacionada y que debemos tener en cuenta a la hora de diagnosticar dicha enfermedad.

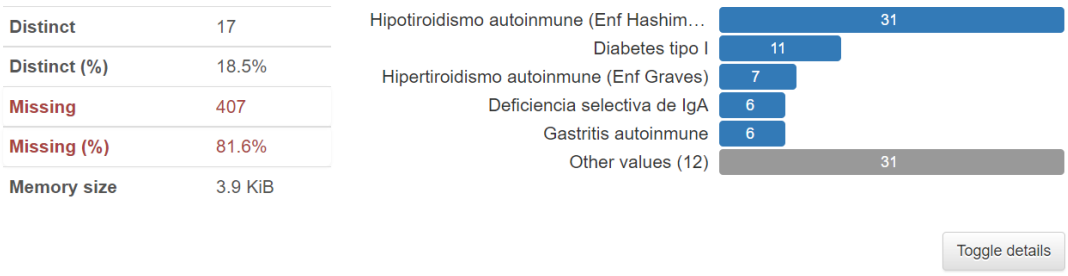


Figura 2.12: Análisis de la enfermedad inmunológica

- *Síntomas específicos, otros síntomas(Figura 2.13):* Esta columna contiene los síntomas que presenta el paciente. Puede llevar en muchos casos a que se identifique correctamente una Enfermedad Celíaca Clásica pero existen algunos casos en los que no está relacionado directamente y se podría diagnosticar de forma errónea.

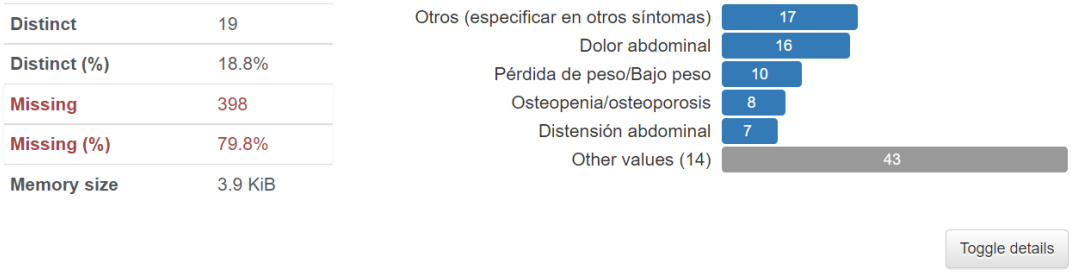


Figura 2.13: Análisis de los síntomas

- *Signos(Figura 2.14):* Los signos, a diferencia de los síntomas, son manifestaciones objetivas, medibles, fruto de un estudio médico. En ello incluimos déficit de hierro o vitamínico.

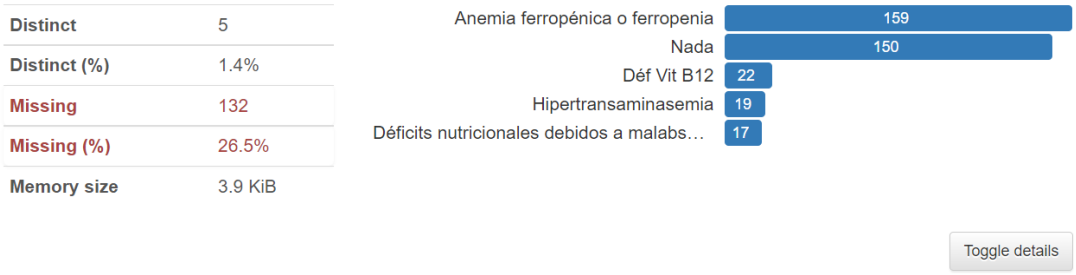


Figura 2.14: Análisis de los signos

- *HLA: grupos de riesgo, Haplotipo1, Haplotipo2(Figura 2.15):* La enfermedad celiaca está asociada con los genes localizados en la región HLA (Human Leukocyte Antigen). HLA-DQ2 y HLA-DQ8 son dos haplotipos (conjunto de genes que se heredan juntos) que están asociados a una mayor predisposición de desarrollar esta enfermedad. En esta columna aparecen los valores del HLA que se calculan a partir de los Haplotipos de los progenitores. En caso de que los progenitores no tengan riesgo, se asignará el valor "Sin riesgo". Estos son los valores que puede tomar este campo, ordenados de mayor a menor riesgo:

  1. DQ2.5, doble dosis: Dos haplotipos con valor DQ2.5, o un haplotipo con valor DQ2.2 y otro con valor DQ2.5. Que portando este alelo, junto con otros nos puede indicar en un 82 % que puede padecerla [4]
  2. DQ2.5 una dosis y DQ8 doble dosis: HLA DQ2.5 una dosis es tener un haplotipo con valor DQ2.5, o un haplotipo con valor DQ7.5 y otro con valor DQ2.2. Para tener HLA DQ8 es preciso tener ambos Haplotipos con valor DQ8. junto con otros nos puede indicar en un 31 % que puede padecerla [4]
  3. DQ8 una dosis: Uno de los haplotipos con valor DQ8.
  4. DQ2.2. DQ7.5: Para que el valor sea HLA DQ2.2 es preciso tener al menos un haplotipo DQ2.2 (y en caso de que solo sea uno, que el otro haplotipo sea sin riesgo). Para tener HLA DQ7.5, es preciso que al menos un hapltipo sea DQ7.5, junto con otros nos puede indicar en un 43 % que puede padecerla. [4]

5. Sin riesgo.

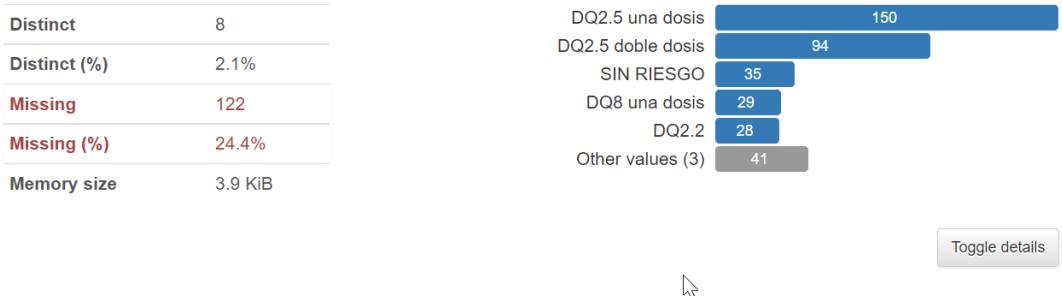


Figura 2.15: Análisis de HLA

- *DCG_ATG2_1, DCG_ATG2_2.(Figura 2.16):* Estas columnas indican los resultados de las pruebas del ATG (anticuerpo anitransglutaminasa) cuando el paciente está siguiendo una dieta con gluten (DCG). Ambas columnas se refieren a la misma prueba tomada en distintos momentos. El valor de estas columnas se obtiene a partir de los valores numéricos de las variables *Indicar título del anticuerpo (DCG ATG_2_1)*, que lo clasifican en tres valores: "Positivo", si el resultado numérico es mayor a 20 o "Negativo", si el resultado numérico es menor que 20. Por otra parte, sólo serán válidos los resultados de las pruebas realizadas con el kit "Aeskulisa tTg-A de Grifols", ya que este es un kit fiable.

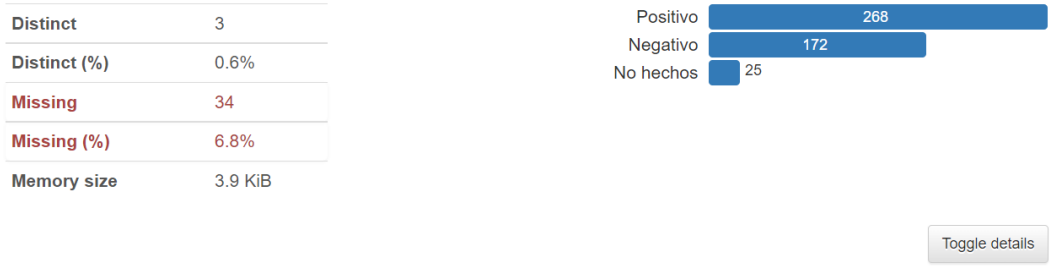
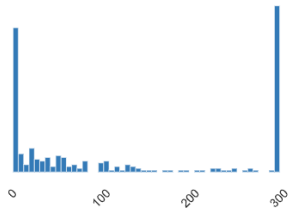


Figura 2.16: Análisis de las pruebas de ATG

- *Indicar título del anticuerpo (DCG ATG_2_1), Indicar título del anticuerpo (DCG ATG2_2)(Figura 2.17):* Se trata del resultado numérico de la prueba que mide el nivel de ATG. Puede tener un valor comprendido entre 0 y 300.

Distinct	158
Distinct (%)	52.7%
Missing	199
Missing (%)	39.9%
Infinite	0
Infinite (%)	0.0%

Mean	128.3403333
Minimum	0.1
Maximum	300
Zeros	0
Zeros (%)	0.0%
Memory size	3.9 KiB



Toggle details

Figura 2.17: Análisis del título de anticuerpo DCG ATG2

- *Indicar el kit empleado con el punto de corte entre paréntesis(Figura 2.18):* Esta columna no nos sirve para realizar el algoritmo, pero sí para comprobar que kit se ha usado para llevar a cabo el test de anticuerpos ATG y así poder desechar los títulos de anticuerpos que sea han obtenido con pruebas poco fiables.

Distinct	2
Distinct (%)	0.5%
Missing	118
Missing (%)	23.6%
Memory size	3.9 KiB

Toggle details

Figura 2.18: Análisis del kit empleado

- *DCG EMA(Figura 2.19):* Se puede ver los resultados de la prueba que mide anticuerpos Anti-endomisio (EMA) que han sido realizadas mientras el paciente seguía una DCG. Normalmente esta prueba se realiza solamente cuando el análisis de ATG2 resulta positiva con niveles relativamente bajos, ya que se trata de una prueba significativamente cara. Los resultados que obtiene suelen ser muy acertados dado que si un paciente da positivo, este es celíaco con total certeza.

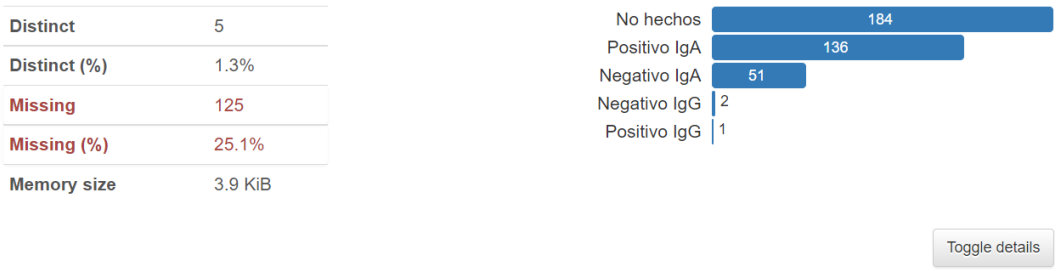


Figura 2.19: Análisis de los resultados de la prueba de DCG EMA

- *DCG A-PDG_1(Figura 2.20):* mide el nivel del anticuerpo PDG cuando el paciente sigue una Dieta Con Gluten. PDG es menos preciso para detectar celiaquía, tanto es así que incluso se pueden dar falsos positivos, que es cuando los pacientes dan positivo en este anticuerpo pero sin embargo no tienen celiaquía. Sin embargo, vamos incluir esta prueba ya que se realiza en la mayoría de los hospitales, y queremos comprobar la relación con la enfermedad.

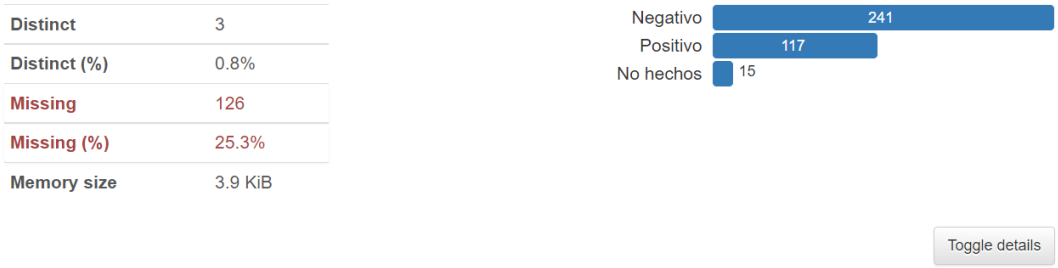
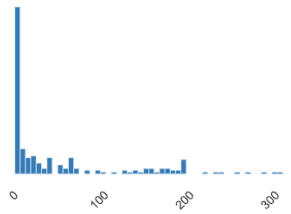


Figura 2.20: Análisis del nivel de DCG A-PDG_1

- *Indique el título del anticuerpo (A-PDG_1)(Figura 2.21):* Es similar a columnas anteriores, como *ATG*, con la diferencia de que consideraremos positivas las pruebas con un valor superior a 25.

Distinct	138	Mean	51.09330144
Distinct (%)	66.0%	Minimum	0.5
Missing	290	Maximum	314
Missing (%)	58.1%	Zeros	0
Infinite	0	Zeros (%)	0.0%
Infinite (%)	0.0%	Memory size	3.9 KiB



Toggle details

Figura 2.21: Análisis del título del anticuerpo

- *Indicar el kit empleado con el punto de corte entre paréntesis(Figura 2.22):* Solo serán válidas las pruebas realizadas con el kit Euroimmun.

Distinct	2
Distinct (%)	1.0%
Missing	298
Missing (%)	59.7%
Memory size	3.9 KiB

Toggle details

Figura 2.22: Análisis del kit empleado

- *DSG ATG2_1, Indique el título del anticuerpo (DSG ATG2_1), Indicar el kit empleado con el punto de corte entre paréntesis (Figura 2.23):* Se mide la presencia del anticuerpo ATG, pero a diferencia de las columnas anteriores, las pruebas se realizaron cuando el paciente seguía una dieta sin gluten(DSG).

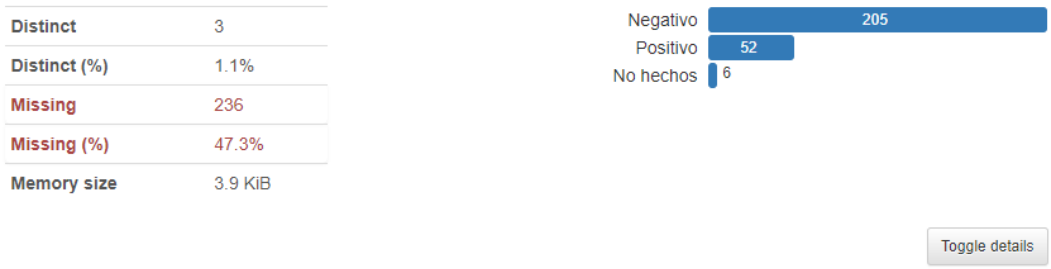


Figura 2.23: Análisis del DSG ATG2_1

- *Indicar título del anticuerpo (DSG ATG2_2)(Figura 2.24):* Es la misma prueba que la columna anterior pero realizada en momentos distintos. Si la prueba se ha repetido, seleccionaremos el valor más bajo.

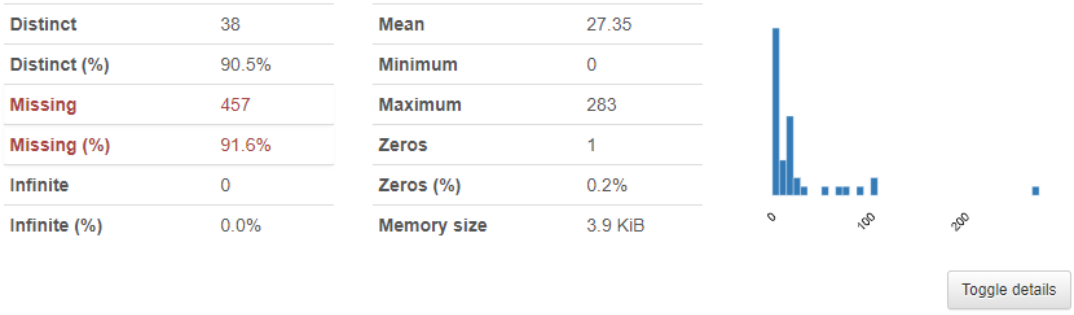


Figura 2.24: Análisis del título del anticuerpo

- *DSG EMA(Figura 2.25):* Se mide los anticuerpos Anti-endomisio, que han sido resultados de la prueba hecha cuando el paciente seguía una Dieta Sin Gluten. Como ya mencionábamos antes acerca de esta prueba, al ser muy cara y muy selectiva a la hora de hacerla, esta columna tiene muchos valores no hechos.

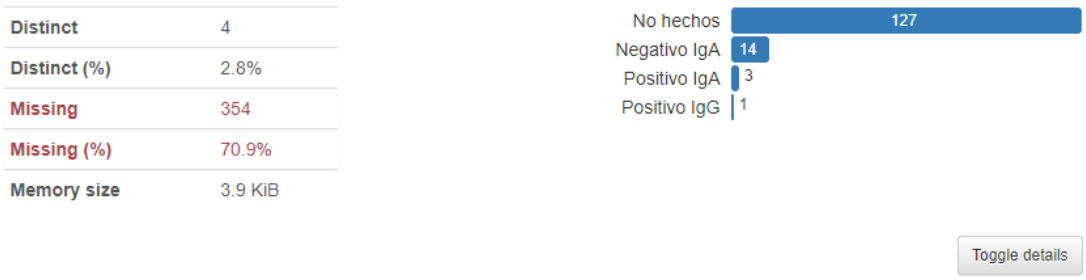


Figura 2.25: Análisis del DSG EMA

- *DSG A-PDG_1, Valor A-PDG_1(Figura 2.26):* Indican la presencia del anticuerpo PDG cuando el paciente esta siguiendo una DSG. En caso de que la prueba se haya repetido, se elegirá el valor más bajo.

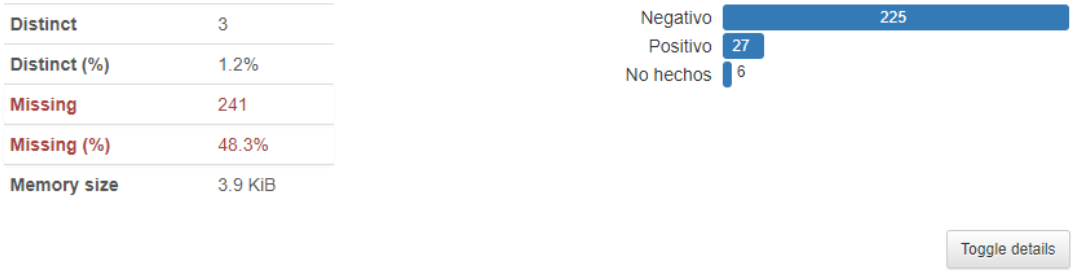


Figura 2.26: Análisis del DSG A-PDG_1

- *DSG Biopsia AP1, DSG Biopsia AP2, AP Biopsia DSG LIEs_1(Figura 2.27):* Muestran los resultados de las biopsias cuando el paciente estaba siguiendo una DSG.

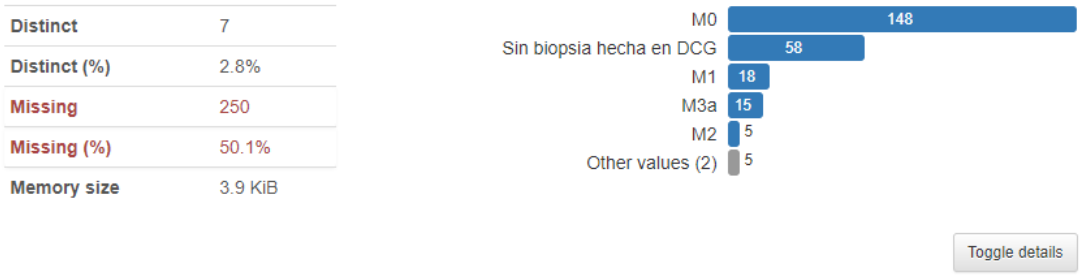


Figura 2.27: Análisis del DSG Biopsia AP1

- *DCG Biopsia-AP1, DCG Biopsia-AP2, AP Biopsia DCG LIEs_1(Figura 2.28):* Muestra el resultado de las biopsias realizadas cuando el paciente sigue una DCG. Refleja el daño intestinal del paciente que cuanto mayor sea el número, mayor es el daño provocado.

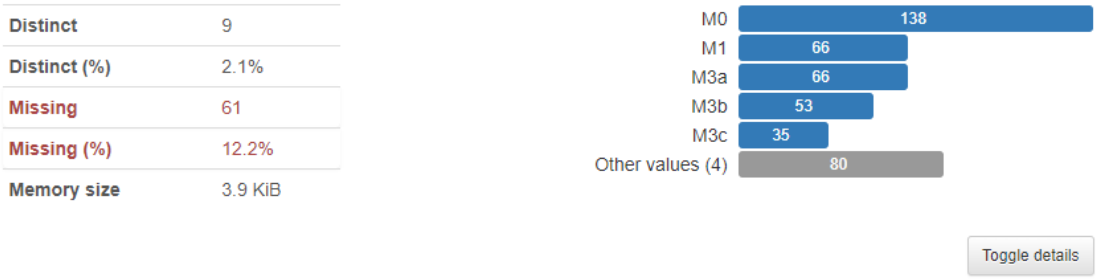


Figura 2.28: Análisis del DCG Biopsia AP1

- *Helicobacter pylori en el momento de la biopsia(Figura 2.29):* Se determina si el paciente tenia la bacteria Helicobacter pylori en su intestino en el momento de realizarse la biopsia. Simplemente contendrá valores de SI o NO.

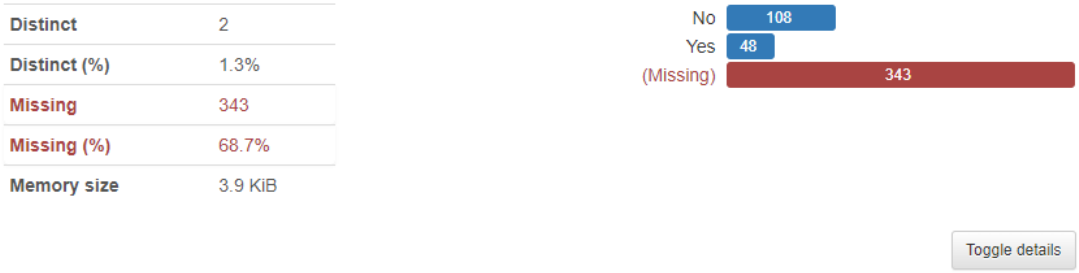


Figura 2.29: Análisis del Helicobacter pylori

- *LIEs DCG %GD_1, LIEs DCG %iNK_1(Figura 2.30):* LIEs se refiere a la prueba de linfograma de linfocitos intraepiteliales. Es una prueba que resulta útil a la hora de detectar la enfermedad celiaca en pacientes con un diagnóstico anómalo o asintomático. En esta columna se muestran los resultados de la prueba cuando el paciente sigue una DCG. Si %GD es mayor o igual que 10 y el parámetro %iNK es menor que 10, entonces

el paciente se califica como “Compatible con la enfermedad celiaca”. Si solo se cumple la primera propiedad mencionada pero la segunda no, entonces el paciente se califica como “Compatible con EC en DSG”.

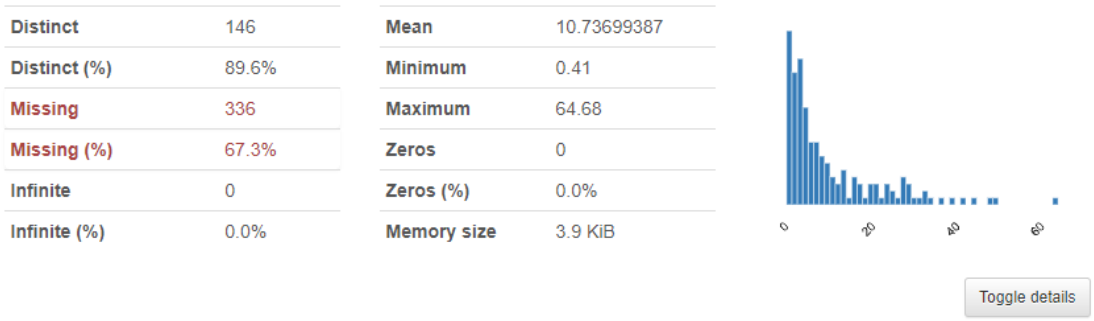


Figura 2.30: Análisis del LIEs DCG %GD_1

- *LIEs DSG %GD_1, LIEs DSG %iNK_1(Figura 2.31):* Se trata de la misma prueba que la columna anterior cuando el paciente sigue una DSG, así que su tratamiento se hará de forma similar.

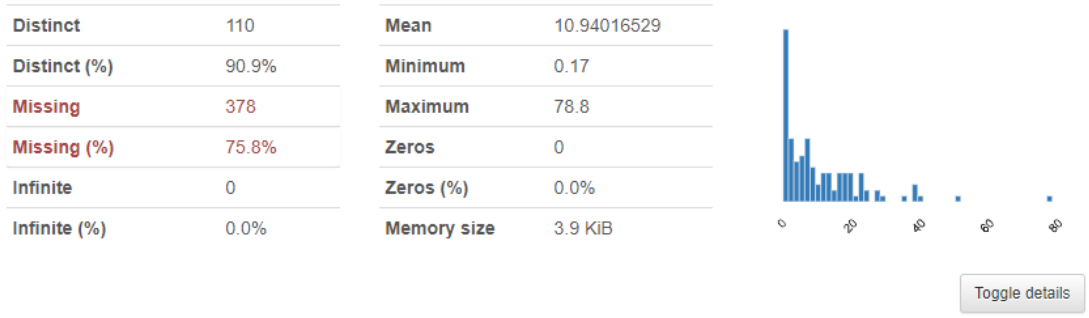


Figura 2.31: Análisis del LIEs DSG %iNK_1

2.2. Manipulación de datos

Para poder empezar a trabajar con los algoritmos de clasificación nuestra base de datos debe contener características suficientemente relevantes para llegar a los resultados correctos,

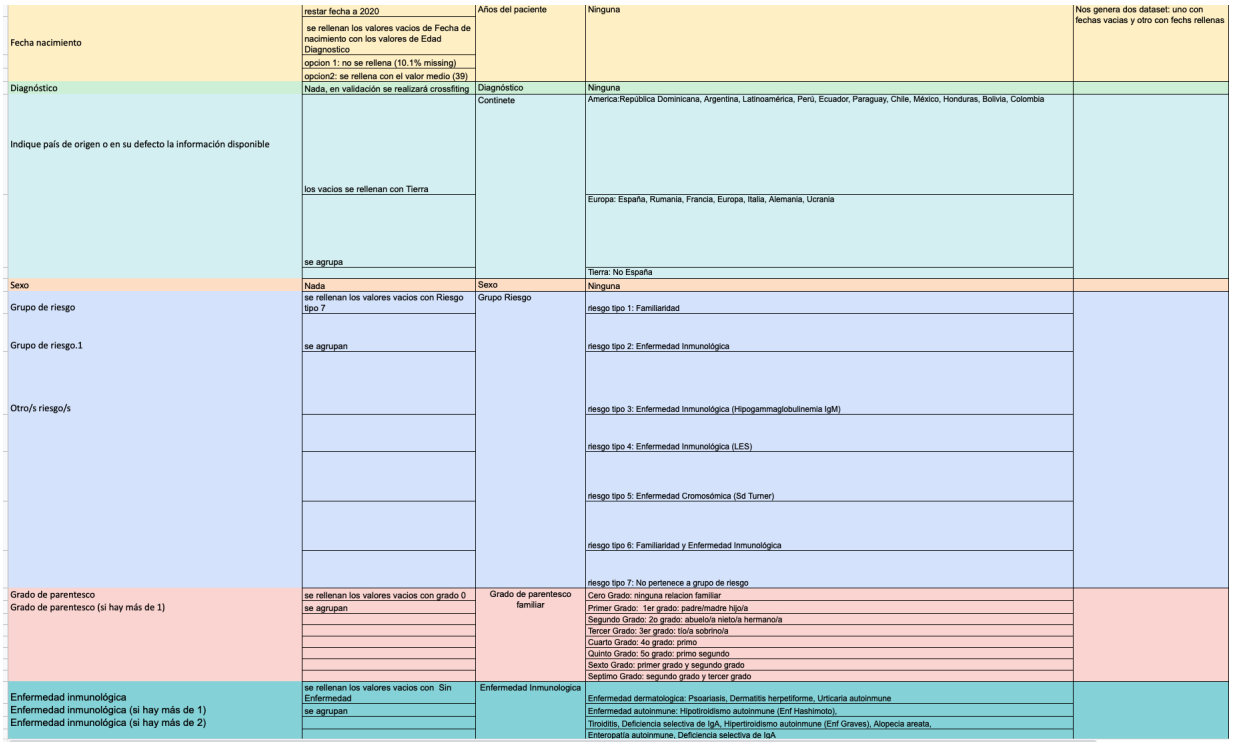
con el fin de llegar a estos resultados seguiremos con el proceso de *Feature Engineering*, que hace referencia a las técnicas de ingeniería utilizadas para seleccionar y transformar variables más relevantes y comprensibles a partir de nuestro dataset sin pulir.

El proceso de *Feature Engineering* cuenta con cuatro pasos a seguir, creación, transformación, extracción y selección de las variables, que son favorables para conseguir que nuestro algoritmo funcione de forma acertada.

2.2.1. Selección y transformación de *features*

Este apartado consiste en reconocer las variables más significativas para nuestro algoritmo, este proceso es subjetivo, en cuanto al objetivo del proyecto, y se realiza observando el reporte de calidad de datos. Si existen columnas/categorías con un porcentaje de información perdida o si la cardinalidad de las categorías es uno en vez de dos o tres, por elemento dentro de las categorías.

Para ello, se crea la ABT *(Analytical Base Table)*, que nos indica que columnas se van a utilizar y contiene una descripción de su utilidad e información adicional, como se muestra en la figura 2.32.



Fecha nacimiento	restar fecha a 2020	Años del paciente	Ninguna	Nos genera dos dataset: uno con fechas vacias y otro con fechs rellenas
	se rellenan los valores vacios de Fecha de nacimiento con los valores de Edad Diagnostico			
	opcion 1: no se rellena (10.1% missing)			
	opcion2: se rellena con el valor medio (39)			
Diagnóstico	Nada, en validación se realizará crossfiting	Diagnóstico	Ninguna	
Indique país de origen o en su defecto la información disponible		Continete	America:República Dominicana, Argentina, Latinoamérica, Perú, Ecuador, Paraguay, Chile, México, Honduras, Bolivia, Colombia	
	los vacios se rellenan con Tierra			
			Europa: España, Rumania, Francia, Europa, Italia, Alemania, Ucrania	
	se agrupa			
			Tierra: No España	
Sexo	Nada	Sexo	Ninguna	
Grupo de riesgo	se rellenan los valores vacios con Riesgo tipo 7	Grupo Riesgo	riesgo tipo 1: Familiaridad	
Grupo de riesgo.1	se agrupan		riesgo tipo 2: Enfermedad Inmunológica	
Otro/s riesgo/s			riesgo tipo 3: Enfermedad Inmunológica (Hipogammaglobulinemia IgM)	
			riesgo tipo 4: Enfermedad Inmunológica (LES)	
			riesgo tipo 5: Enfermedad Cromosómica (Sd Turner)	
			riesgo tipo 6: Familiaridad y Enfermedad Inmunológica	
			riesgo tipo 7: No pertenece a grupo de riesgo	
Grado de parentesco	se rellenan los valores vacios con grado 0	Grado de parentesco familiar	Cero Grado: ninguna relacion familiar	
Grado de parentesco (si hay más de 1)	se agrupan		Primer Grado: 1er grado: padre/madre hijo/a	
			Segundo Grado: 2o grado: abuelo/a nieto/a hermano/a	
			Tercer Grado: 3er grado: tio/a sobrino/a	
			Cuarto Grado: 4o grado: primo	
			Quinto Grado: 5o grado: primo segundo	
			Sexto Grado: primer grado y segundo grado	
			Septimo Grado: segundo grado y tercer grado	
Enfermedad inmunológica	se rellenan los valores vacios con Sin Enfermedad	Enfermedad Inmunologica	Enfermedad dermatologica: Psoariasis, Dermatitis herpetiforme, Urticaria autoinmune	
Enfermedad inmunológica (si hay más de 1)	se agrupan		Enfermedad autoinmune: Hipotiroidismo autoinmune (Enf Hashimoto),	
Enfermedad inmunológica (si hay más de 2)			Tiroiditis, Deficiencia selectiva de IgA, Hipertiroidismo autoinmune (Enf Graves), Alopecia areata,	
			Enteropatia autoinmune, Deficiencia selectiva de IgA	

Figura 2.32: Selección de *features* relevantes: ABT

Además, se han revisado las variables numéricas, y si estas contienen un valor perdido la transformamos a su valor nulo para que el algoritmo funcione correctamente. De esta forma, añadimos también un valor más para que no haya datos perdidos en las siguientes variables.

- *DCG_ATG2_1,DSG ATG2_1, DCG EMA, DSG EMA, DSG A-PDG_1 ,DCG A-PDG_1 (Figura 2.33):* Todas estas variables se transforman de la misma forma, los valores que toman son "Positivo" o "Negativo", y "No Hecho" en el caso de que no se ha haya realizado la prueba o sea un valor perdido.

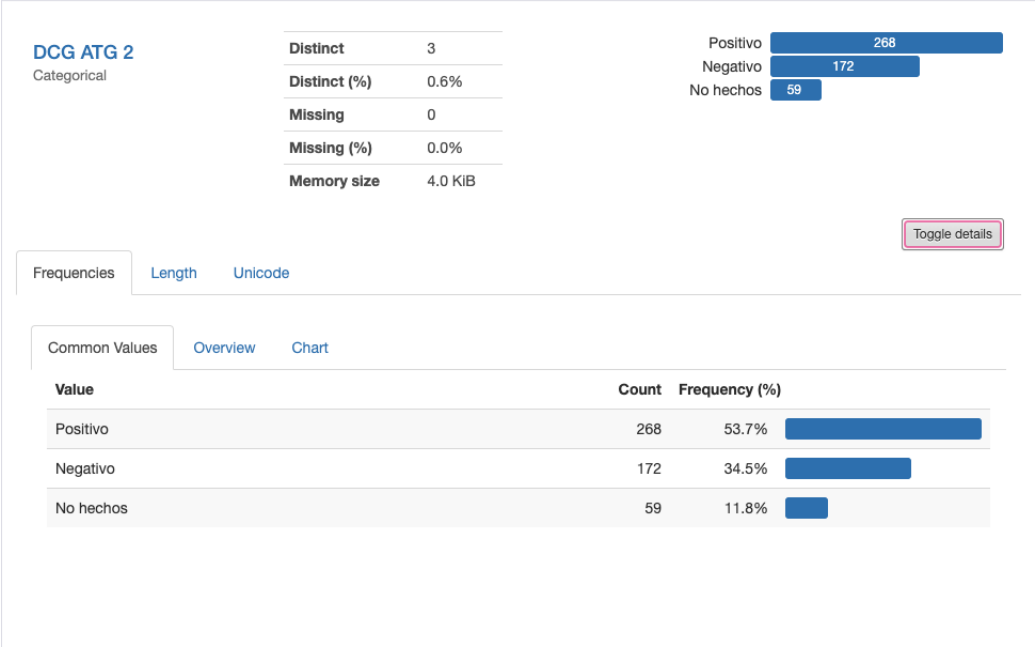


Figura 2.33: Transformación de DCG_ATG2_1

- *Helicobacter pylori en el momento de la biopsia (Figura 2.34):* Esta característica solo admite los valores "Si" o "No". En este caso la variable nula por la que se sustituye los valores perdidos es "No".

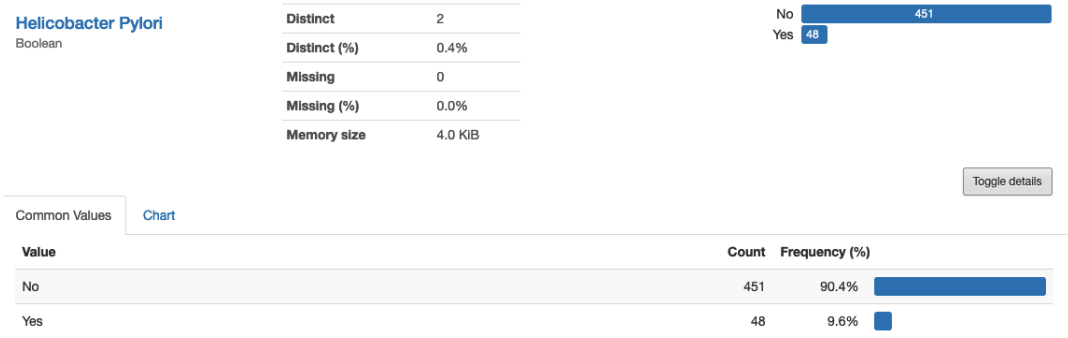


Figura 2.34: Transformación de Helicobacter pylori

- *DCG Biopsia-AP1,DSG Biopsia-AP1, DCG Biopsia-AP2,DsG Biopsia-AP2 (Figura 2.35):* Las variables relacionadas con las biopsias realizadas admiten los siguientes

valores, Marsh 1, Marsh 2, Marsh 3a, Marsh 3b y Marsh 3c. Si el valor esta vacío se añadirá “Sin biopsia hecha en DCG”.

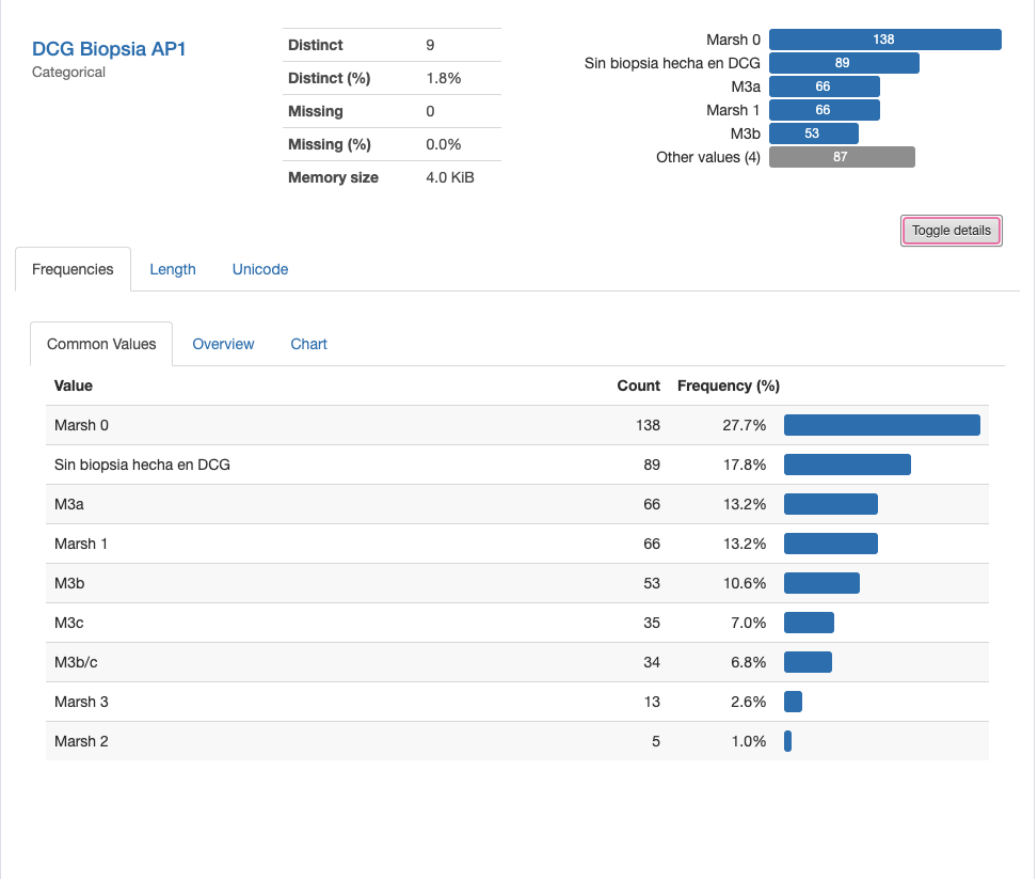


Figura 2.35: Transformación de DCG Biopsia-AP1

- *AP Biopsia DCG LIEs_1, AP Biopsia DSG LIEs_1 (Figura 2.36):* Estas variables pueden obtener los siguientes valores Marsh 0, Marsh 0-Hp negativo, Marsh 0-Hp positivo, M1-Hp negativo, M3b, Marsh 1, M3a, M3c, M1-Hp positivo, Marsh 2 y M3b-Hp negativo. En el caso de que haya valores vacíos se sustituirán por Marsh 0.

  En las dos columnas anteriores, los valores que representan son:

  - Marsh 0: indica que no se ha producido daño en el intestino delgado.
  - Marsh 1: el intestino delgado tiene las ILS dañadas en un 25 %.
  - Marsh 2: las vellosidades son normales pero contienen criptas hiperplásicas.

- Marsh 3: en este caso existe una atrofia en las vellosidades intestinales, criptas hiperplásicas y aumento de ILS.

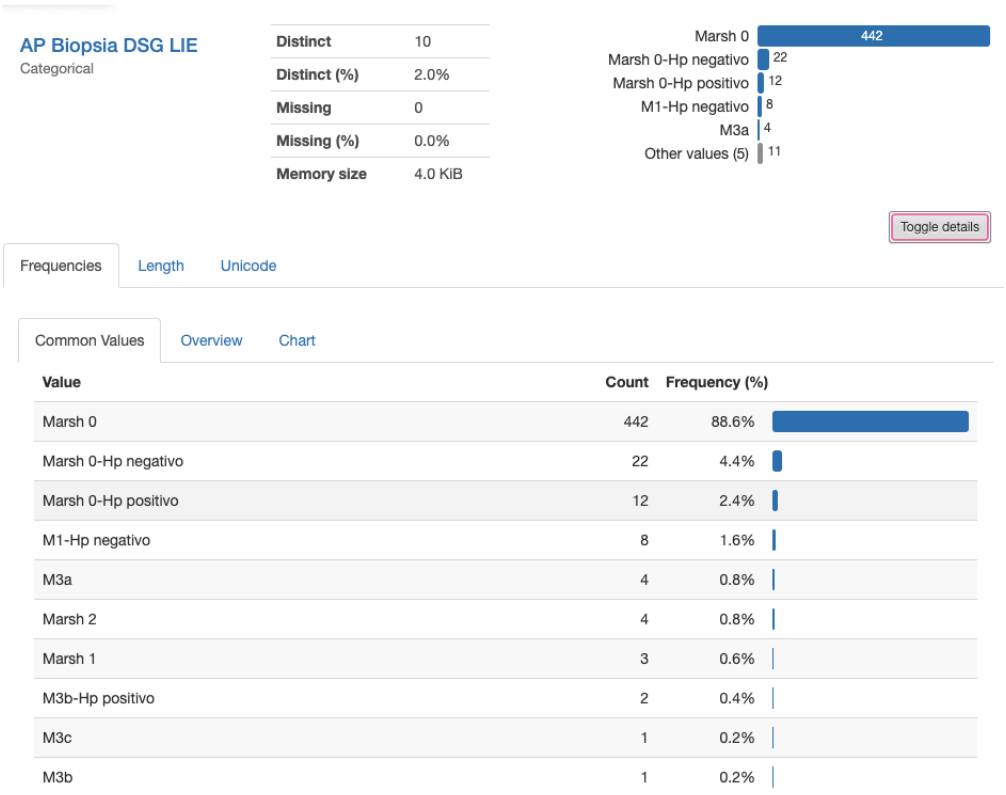


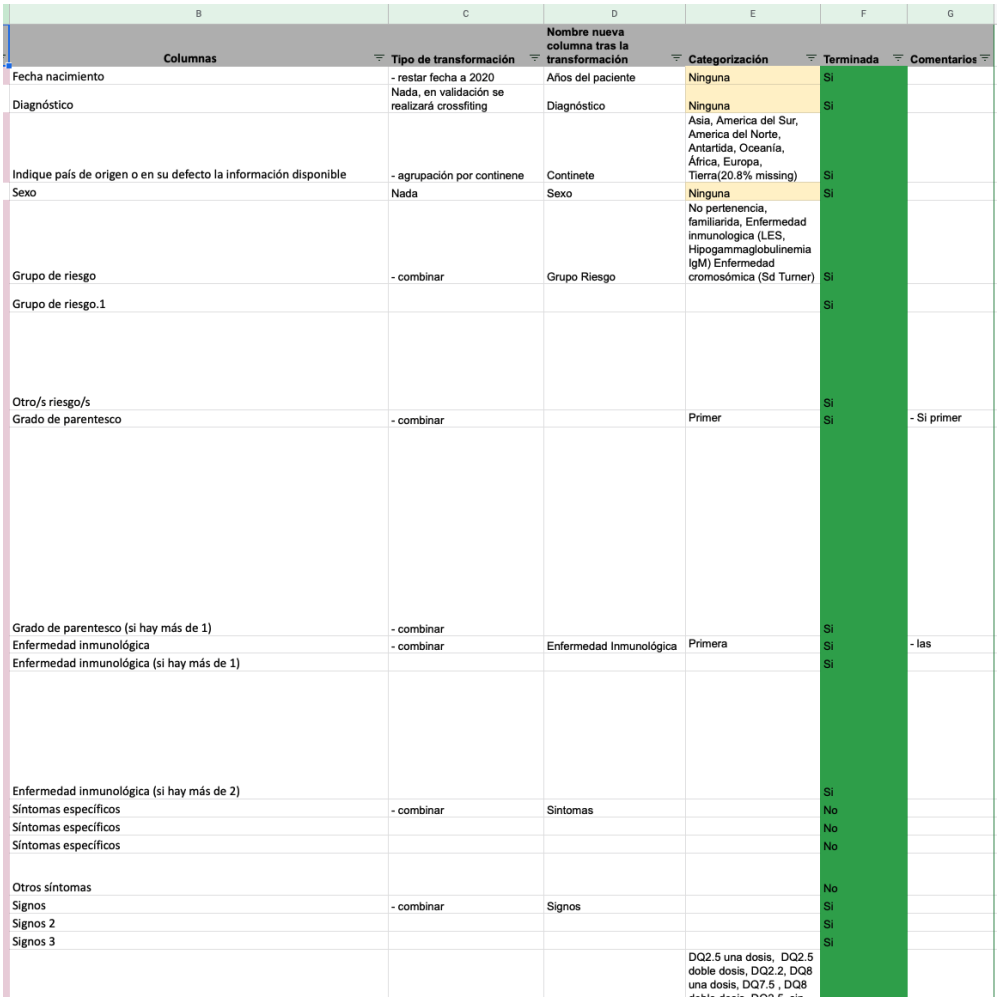
Figura 2.36: Transformación de AP Biopsia DSG LIEs

2.2.2. Extracción y creación de *features*

En este proceso se combinan y transforman las columnas, para crear una categoría final para el modelo, utilizando una nueva hoja de Excel como se presenta en la figura 2.37, se valora si las variables deberían unirse para formar una nueva y así extraerlas para reducir el volumen de pérdida de datos.

En esta etapa, además, se ha realizado una exploración de los datos, para ello se ha generado un reporte de calidad por cada categoría, en este reporte se incluye información como, en las categorías continuas como el valor mínimo, los percentiles, el rango de valor que toman y las variables descriptivas como la desviación estándar o la varianza; las categorías

categóricas contienen, por su parte, los valores que contiene, es decir, los elementos que forman la categoría, la moda y un histograma.



Columnas	Tipo de transformación	Nombre nueva columna tras la transformación	Categorización	Terminada	Comentarios
Fecha nacimiento	- restar fecha a 2020	Años del paciente	Ninguna	Si	
Diagnóstico	Nada, en validación se realizará crossfiting	Diagnóstico	Ninguna	Si	
Indique país de origen o en su defecto la información disponible	- agrupación por continene	Continete	Asia, America del Sur, America del Norte, Antartida, Oceanía, África, Europa, Tierra(20.8% missing)	Si	
Sexo	Nada	Sexo	Ninguna	Si	
Grupo de riesgo	- combinar	Grupo Riesgo	No pertenencia, familiarida, Enfermedad inmunologica (LES, Hipogammaglobulinemia IgM) Enfermedad cromosómica (Sd Turner)	Si	
Grupo de riesgo.1				Si	
Otro/s riesgo/s				Si	
Grado de parentesco	- combinar		Primer	Si	- Si primer
Grado de parentesco (si hay más de 1)	- combinar			Si	
Enfermedad inmunológica	- combinar	Enfermedad Inmunológica	Primera	Si	- las
Enfermedad inmunológica (si hay más de 1)				Si	
Enfermedad inmunológica (si hay más de 2)				Si	
Síntomas específicos	- combinar	Sintomas		No	
Síntomas específicos				No	
Síntomas específicos				No	
Otros síntomas				No	
Signos	- combinar	Signos		Si	
Signos 2				Si	
Signos 3				Si	
			DQ2.5 una dosis, DQ2.5 doble dosis, DQ2.2, DQ8 una dosis, DQ7.5 , DQ8		

Figura 2.37: Extracción y creación de *features*: ABT Extendida

- *Fecha nacimiento (Figura 2.38) :* Con esta columna se crea *Años del paciente*, si la fecha esta vacía se coloca el valor de edad diagnóstico, y para los valores que están vacíos se coloca la media o 0. Esto genera un dataset extra, valoraremos cual de ellos nos da mejor resultado.

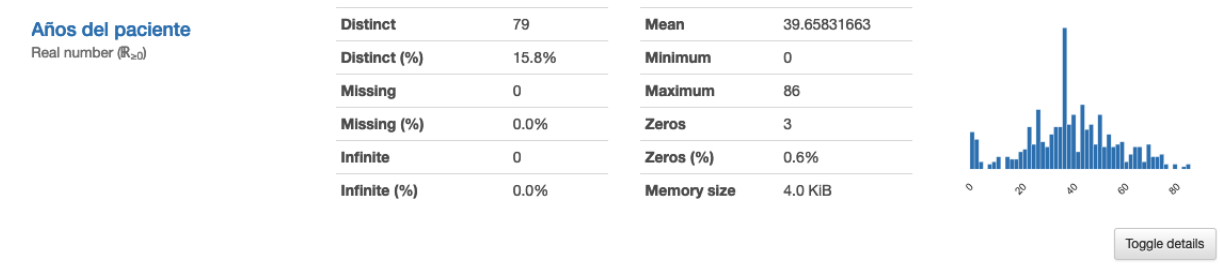


Figura 2.38: Extracción y creación de Fecha nacimiento

- *País de origen (Figura 2.39):* se crea una nueva característica llamada País y se añade el valor "tierra" para los valores perdidos.

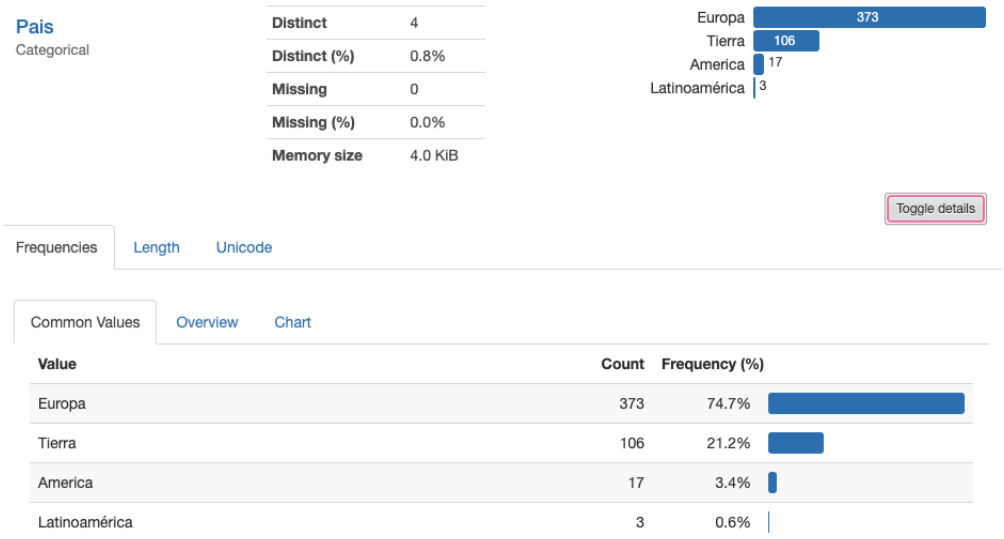


Figura 2.39: Extracción y creación de Pais

- *Grupo de riesgo, Grupo de riesgo.1, Otro/s riesgo/s (Figura 2.40):* Se unen estas tres características para formar una nueva. La forma de combinarlas va a ser agrupar distintos tipos de riesgo o combinaciones de estos grupos de riesgo de forma que se puedan presentar en una sola columna.

Figura 2.40: Extracción y creación de Grupo de riesgo

- *Grado de parentesco (Figura 2.41):* Se unen todas las características de grado de parentesco en una sola. Para los casos en los que existen varios grados de parentesco por un paciente, se crean grados nuevos que simbolizan la pertenencia a ambos grados. Los valores vacíos pasaran a ser Grado 0.

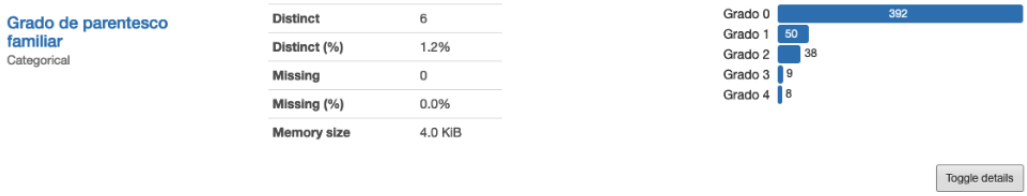


Figura 2.41: Extracción y creación de Grado de parentesco

- *Enfermedad Inmunológica (Figura 2.42):* Se unen todas las características de Enfermedad Inmunológica en una, y se rellenan los valores vacíos como "Sin enfermedad"

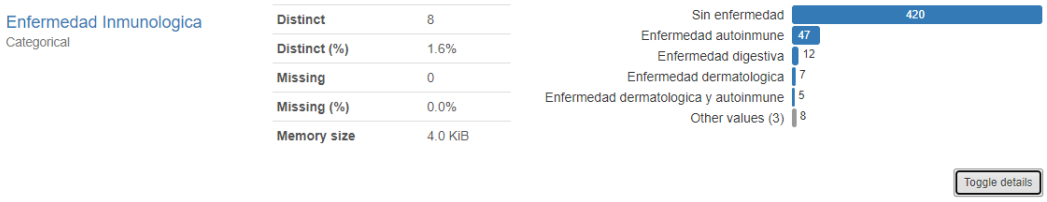


Figura 2.42: Extracción y creación de Enfermedad Inmunológica

- *Síntomas (Figuras 2.43, 2.44):* Se unen todas las características de síntomas en una sola. Al haber tantos síntomas distintos, los hemos agrupado por tipos de síntomas y por grupos. Los valores vacíos se clasifican como asintomáticos.

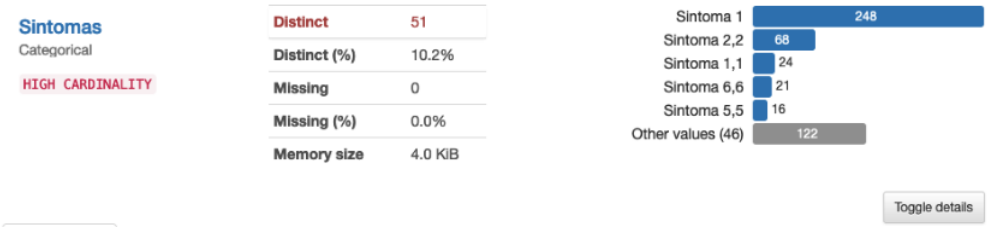


Figura 2.43: Extracción y creación de Síntomas por tipos

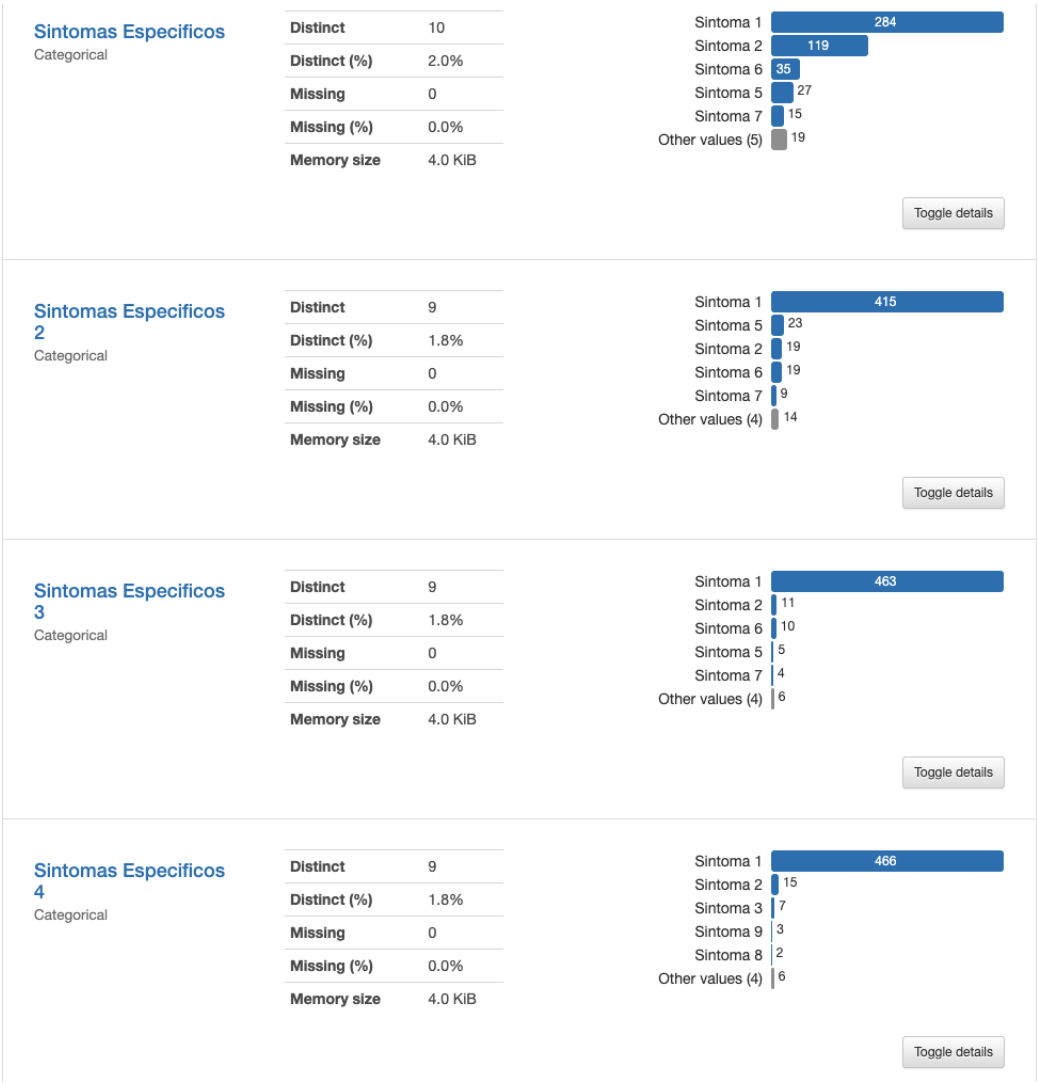


Figura 2.44: Extracción y creación de Síntomas por grupos

- *Signos (Figura 2.45):* Se unen todas las características de signos en una y se asigna el valor "Sin enfermedad" a los valores vacíos.

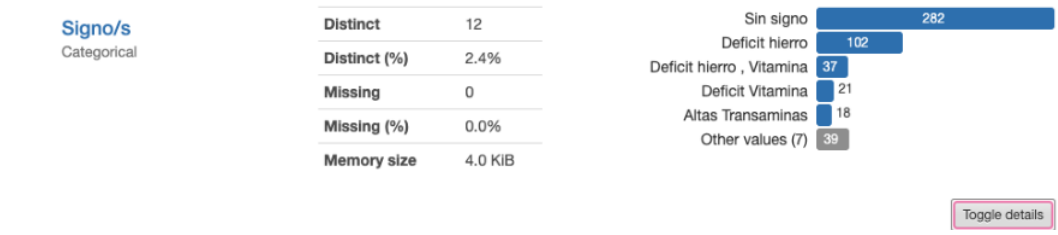


Figura 2.45: Extracción y creación de Signos:

- *HLA, Haplotipo 1, Haplotipo 2 (Figura 2.46):* Dependiendo de haplotipo de padre y de la madre creamos una columna de HLA con los riesgos resultantes.

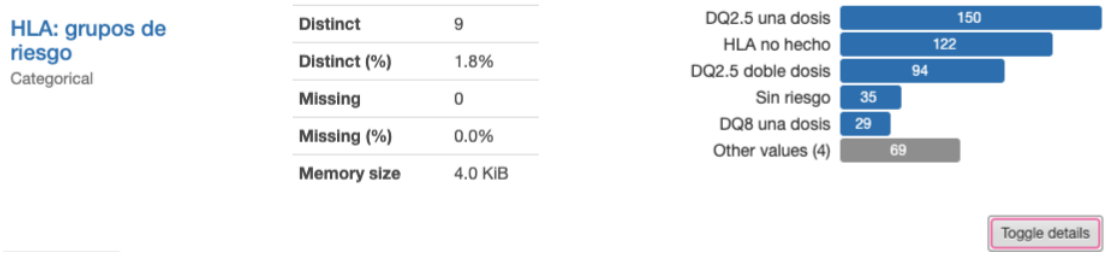


Figura 2.46: Extracción y creación de HLA

Tratamiento de los datos categóricos

Tras la creación de características más favorables, las variables que tienen un gran volumen de valores necesitan una categorización, es decir etiquetar los datos según la tendencia de estos.

Estas son las variables que hemos categorizado automáticamente con un programa en Python, utilizando la biblioteca de Pandas Profiling y creando un reporte final para comprobar las estadísticas de los datos.

- *Grupo de riesgo, Grupo de riesgo.1, Otro/s riesgo/s (Figura 2.47):*
  - riesgo tipo 1: Familiaridad
  - riesgo tipo 2: Enfermedad Inmunológica
  - riesgo tipo 3: Enfermedad Inmunológica (Hipogammaglobulinemia IgM)
  - riesgo tipo 4: Enfermedad Inmunológica (LES)

- riesgo tipo 5: Enfermedad Cromosómica (Sd Turner)
- riesgo tipo 6: Familiaridad y Enfermedad Inmunológica
- riesgo tipo 7: No pertenece a grupo de riesgo / Otro (especificar en columna 'Otro riesgo')

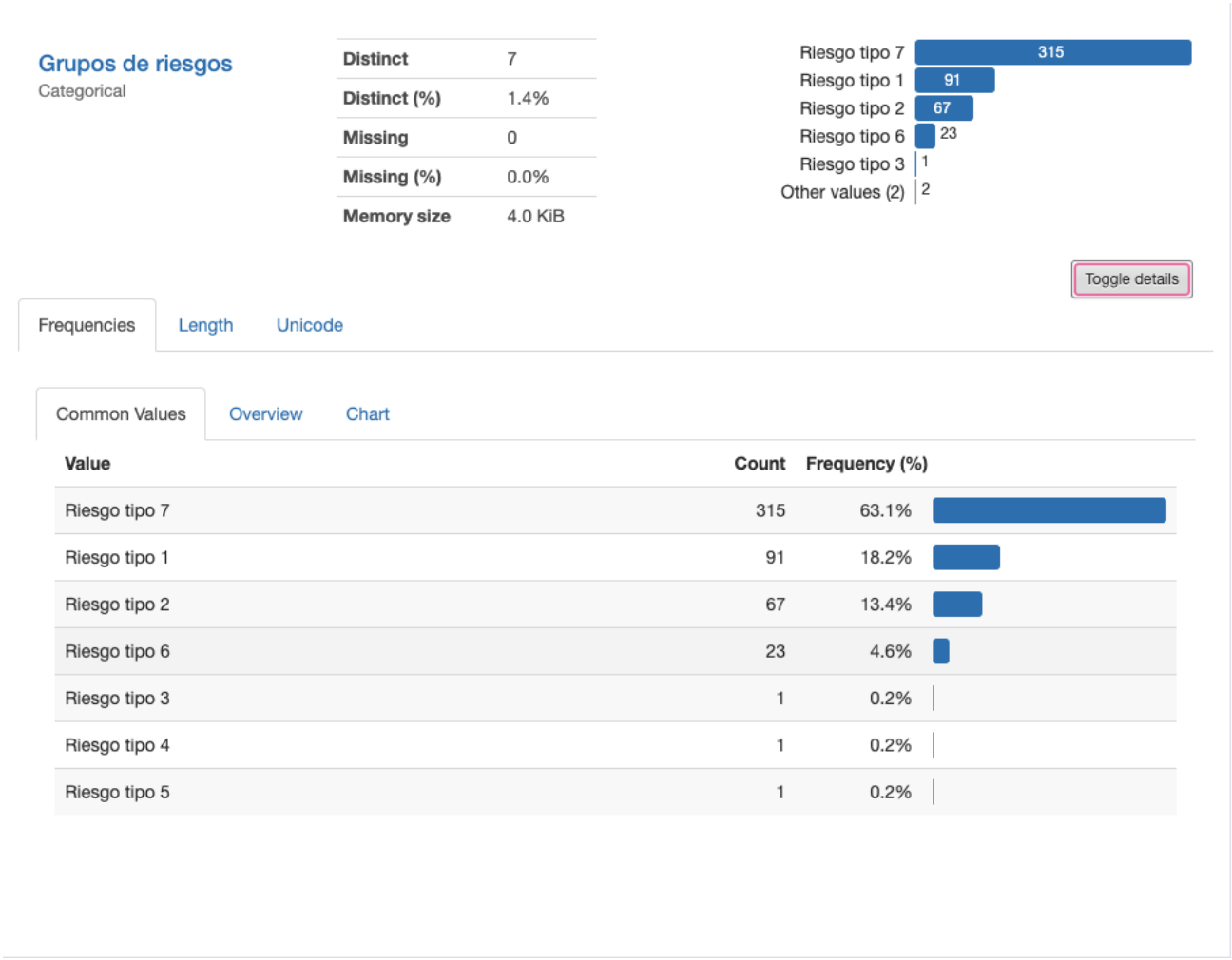


Figura 2.47: Extracción y creación de Grupo de riesgo formateado

- *Grado de parentesco (Figura 2.48):*
  - Cero Grado: ninguna relación familiar
  - Primer Grado: 1º grado: padre/madre hijo/a

- Segundo Grado: 2º grado: abuelo/a nieto/a hermano/a
- Tercer Grado: 3º grado: tío/a sobrino/a
- Cuarto Grado: 4º grado: primo
- Quinto Grado: 5º grado: primo segundo

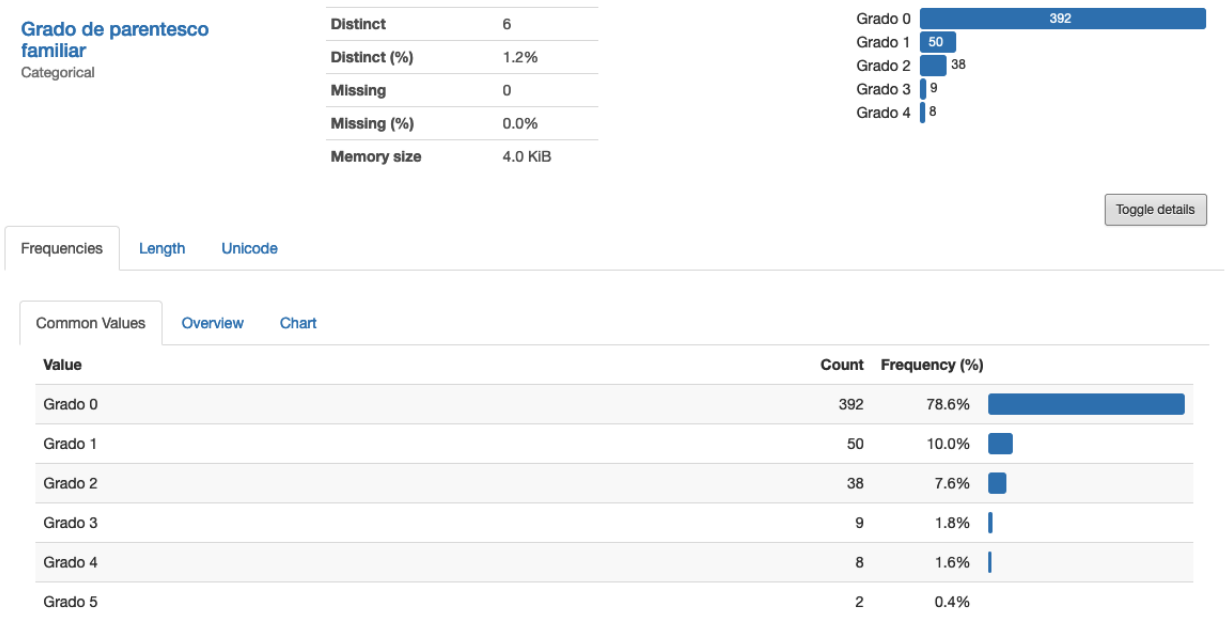


Figura 2.48: Extracción y creación de Grado de parentesco

- *Enfermedad Inmunológica (Figura 2.49)*
  - Enfermedad dermatológica: Psoriasis, Dermatitis herpetiforme, Urticaria autoinmune
  - Enfermedad autoinmune: Hipotiroidismo autoinmune (Enf Hashimoto), Tiroiditis, Deficiencia selectiva de IgA, Hipertiroidismo autoinmune (Enf Graves), Alopecia areata,Enteropatía autoinmune, Deficiencia selectiva de IgA, Cirrosis biliar primaria

- Enfermedad digestiva: Gastritis autoinmune, Enfermedad inflamatoria intestinal, Hepatitis autoinmune
- Enfermedad osea: Artritis reumatoide
- Sin enfermedad: Otra o dato perdido
- Enfermedad autoinmune y dermatológica: psoriasis y Hipotiroidismo, dermatitis y Hipotiroidismo, urticaria autoinmune y deficiencia selectiva de IgA
- Enfermedad autoinmune y digestivo: Diabetes tipo I y hipotiroidismo
- Enfermedad dermatológica y digestiva: psoriasis y enfermedad inflamatoria intestinal, psoriasis y Gastritis autoinmune, diabetes tipo I y dermatitis,

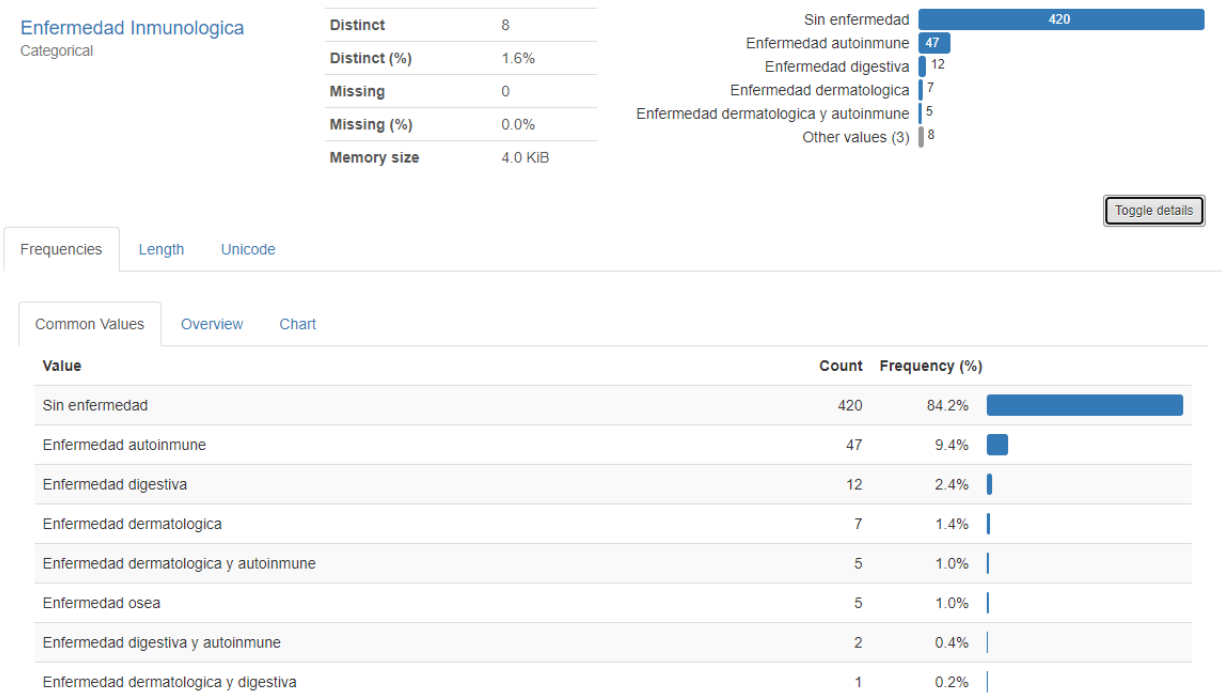


Figura 2.49: Extracción y creación de Enfermedad Inmunológica

- *Síntomas (Figura 2.50)*
  - Síntoma 1 - Asintomático: Nan/asintomático.

- Síntoma 2 - Síntomas digestivos: Trastorno gastrointestinal tipo SII, Dispepsia, Vómitos, Estreñimiento, Diarrea crónica, Saciedad precoz, Regurgitación ácida, Digestiones lentas, Diarrea/Estreñimiento, Malabsorción, Saciedad precoz, Regurgitación ácida, Rectorragia, Encopresis, Intolerancias alimentarias, Artralgias, Disfagia, Esofagitis Eosinofilica, Desnutrición, Flatulencias, Reflujo, Diarrea/Estreñimiento y Saciedad precoz.

- Síntoma 3 - Síntomas cerebrales: Migrañas-Migrañas crónicas, Fibromialgia, Cefalea, Disestesias, Prurito y Sensibilidad química múltiple.

- Síntoma 4 - Síntomas en extremidades: Cervicalgia.

- Síntoma 5 - Síntomas abdominales: Distensión/Dolor abdominal.

- Síntoma 6 - Síntomas físicos: Astenia/Fatiga crónica, Retraso crecimiento/baja estatura, Pérdida de peso/Bajo peso, Sobrepeso y Retraso crecimiento/baja estatura.

- Síntoma 7 - Síntomas oseo: Espondilóllistesis, Osteopenia/osteoporosis, Osteoporosis, artritis erosiva en manos y Artritis.

- Síntoma 8 - Síntomas psicológicos: Trastorno alucinatorio delirante,Depresión, ansiedad, Irritabilidad y Anorexia nerviosa.

- Síntoma 9 - Síntomas dermatológicos: Dermatitis herpetiforme, Angioedema, Psoriasis, Dermatitis atópica, yExantema.

- Síntoma 10 - Síntomas bucales: Aftas orales.

- Síntoma 11 - Síntomas reproductivos: Infertilidad/abortos recurrentes, Infertilidad y Hiperprolactinemia.

- Síntoma 12 respiratorio: TEP (Tromboembolismo pulmonar)

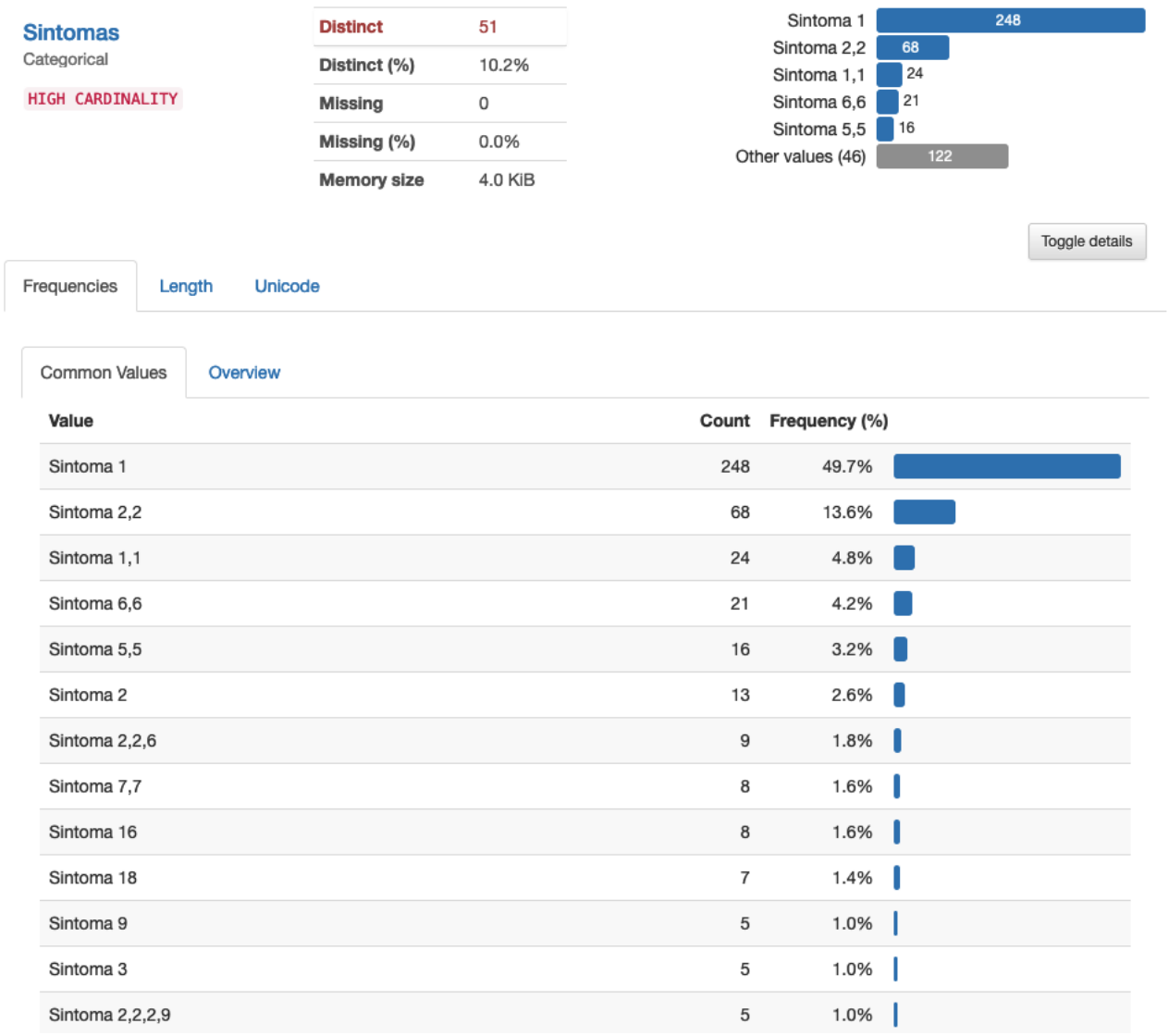


Figura 2.50: Extracción y creación de síntomas

- *Signos (Figura 2.51)*
  - Déficit Hierro: Anemia ferropénica o ferropenia.
  - Déficit Vitamina: Déficit Vitamina B12
  - Altas Transaminasas: Hipertransaminasemia
  - Déficit Nutricional: Déficit nutricional debidos a malabsorción
  - Deficit hierro y vitamina: Anemia ferropénica o ferropenia y Déf Vit B12

- Déficit nutricional y vitamina: Déficit nutricional debidos a malabsorción y Déf Vit B12
- Deficit hierro y nutricional: Anemia ferropénica o ferropenia y Déficits nutricionales debidos a malabsorción
- Déficit hierro, nutricional y vitamínico

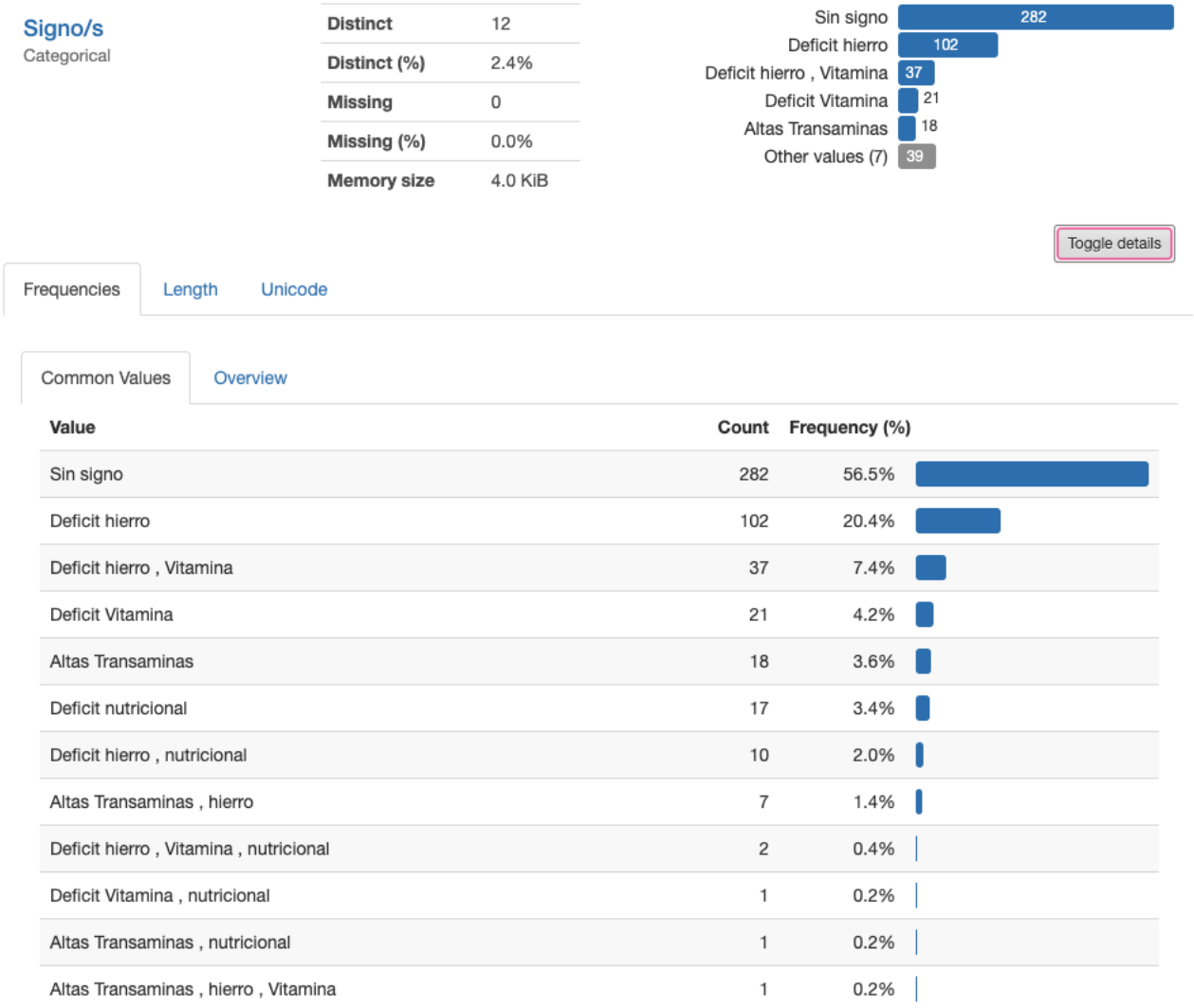


Figura 2.51: Extracción y creación de Signos

- *HLA (Figura 2.52):*

- HLA: grupos de riesgo
- Haplotipo1
- Haplotipo2

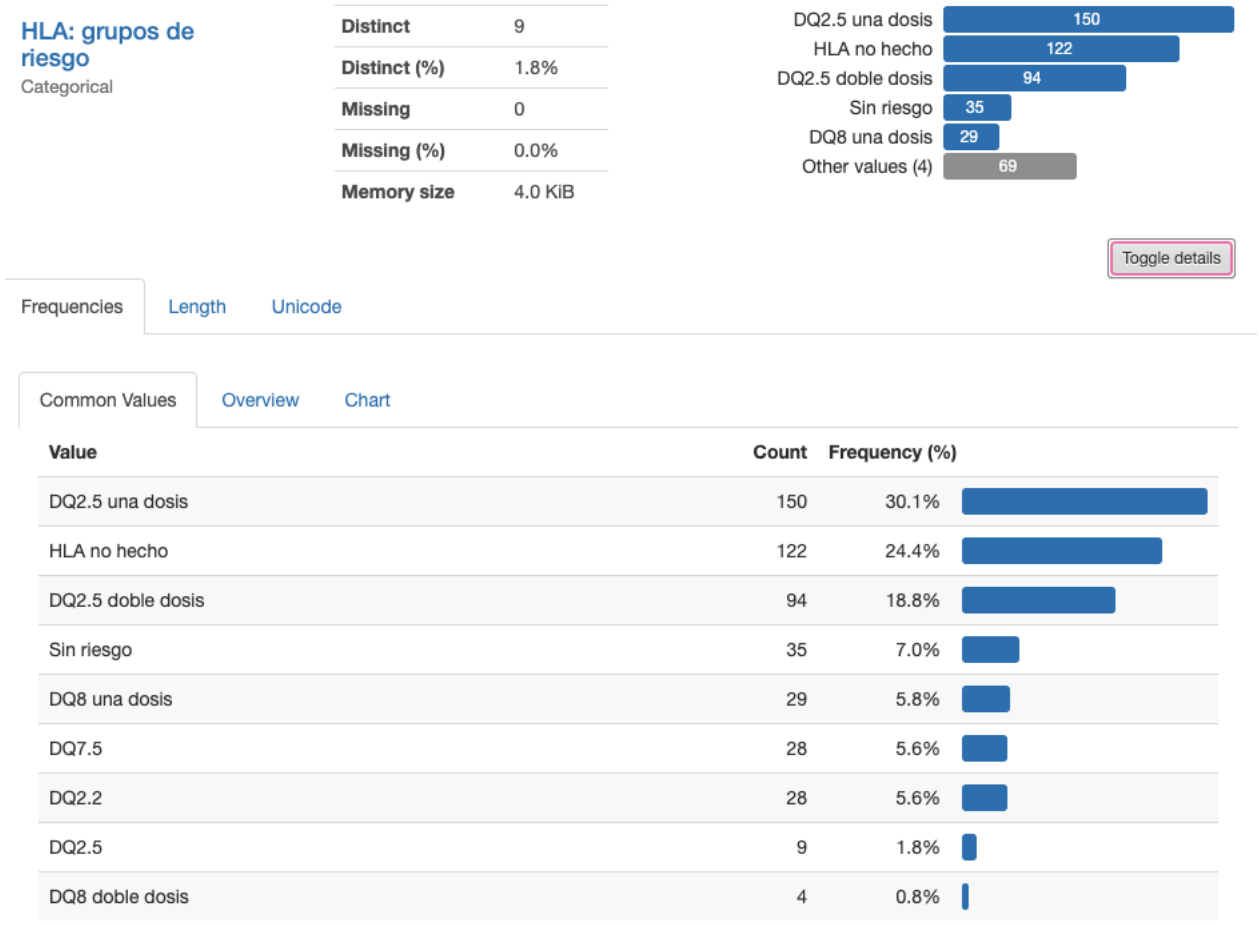


Figura 2.52: Extracción y creación de HLA

2.2.3. Codificación de las columnas categorizadas

Una vez transformados los datos, la información será decodificada para poder ser ejecutada por los algoritmos, muchos de ellos necesitan un pre-procesamiento y una codificación. En este caso según el tipo de columna se ha realizado un tipo de codificación:

- Count Vectorizer: convierte una categorización de texto a su representación simbólica del contador del texto. Se produce por ejemplo en las columnas de signos o enfermedad inmunológica.

- Label Encoder: categoriza numéricamente las etiquetas de texto que tenemos. Se produce en las columnas: síntomas, HLA, grado parentesco o grupo de riesgo

- One Hot Encoder: realiza una transformación binaria del texto, asociando 0 o 1 a las categorias. En este caso, se ha ejecutado por ejemplo en la columna sexo.

2.3. Datasets generados

Finalmente, tras la selección de los atributos del algoritmo y teniendo en cuenta que no vamos a eliminar pacientes del dataset, se han generado los siguientes dataset:

- *Dataset I (síntomas separados con fecha completa):* en este caso al producir mucha información el atributo síntoma, se ha decidido establecer que se mantienen las columnas originales pero con la categorización anterior y las fechas rellenas con la edad media.[2]

- *Dataset II (síntomas no separados con fecha completa)* la columna va a transformarse en una, generando alta combinaciones de las categorías de síntomas y las fechas estan completas.[3]

- *Dataset III (síntomas separados con fecha incompleta):* es el proceso de mantener la categorización y no rellenar las fechas. [4]

- *Dataset IV (síntomas no separados con fecha incompleta):* es el proceso de mantener la categorización y rellenar las fechas. [5]

[2] https://drive.google.com/file/d/19oGMhl7VTOQebSSmHyRsdp8gDifyItOQ/view?usp=sharing
[3] https://drive.google.com/file/d/1S4bQKBEuc-EJfSL84ITeoJ2YjRvT25ar/view?usp=sharing
[4] https://drive.google.com/file/d/1DBUw4M_V2YZ-JmmDM4ZCXiB-OJIvXhtT/view?usp=sharing
[5] https://drive.google.com/file/d/13j5YJNnEaPs6o5DyjzjsX3MiObnrtob1/view?usp=sharing

Con estos dataset lo que se desea conseguir es profundizar en la importancia de los valores nulos, vacíos o con información redundante.

Realizando un análisis previo podríamos llegar a concluir que en el caso de la separación de síntomas, se produciría un caso de *overfitting*, que ocurre cuando las predicciones del modelo seleccionado por el algoritmo es complejo, tal que, el modelo encaja directamente con los datos proporcionados, y se convierte en un modelo sensible a los cambios por ruido de los datos. En este caso, se produce porque la columna síntomas tiene una gran variedad de opciones y casos vacíos o huecos que puede que no tengan relación con la etiqueta seleccionada. El *overfitting* se relaciona con los parámetros de varianza elevada y *bias* bajo.

Por contraste, el *underfitting* es el problema que ocurre cuando la predicción del modelo seleccionado es demasiado simple para representar la relación entre las variables descriptivas del dataset. En nuestro caso, hemos evitado este tipo de problemas eliminando las columnas Indicar título del anticuerpo ATG que tienen una relación directa con los las columnas DSG/DCG ATG. Se relaciona con los parámetros de varianza baja y *bias* elevado.

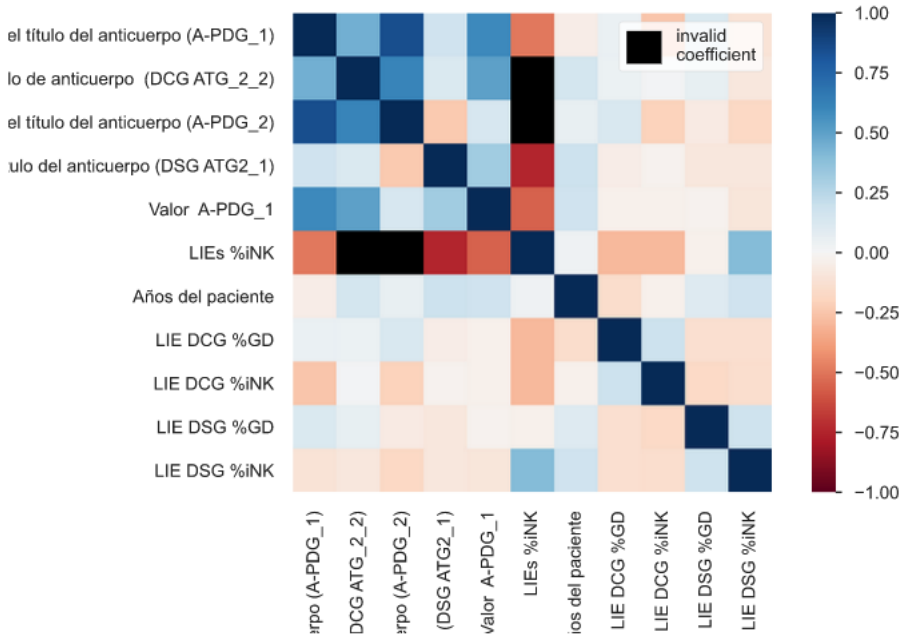


Figura 2.53: Relación entre las categorías que producen underfitting

Como se puede observar en la matriz de confusión de la correlación Pearson, Figura 2.53, los valores cercanos a 1 nos indican una mayor relación y los cercanos a -1, lo contrario. Para calcular esta medida se divide la covarianza de la categoría x entre el producto de la desviación estándar de la categoría del x e y.

Esto es un análisis previo, en capítulos posteriores comprobaremos la utilidad de dicha separación, y que soluciones se podría realizar para evitar problemas y mejorar el efecto de las medidas sobre los objetivos planteados.

Capítulo 3

Tecnologías

En este apartado vamos a hablar sobre las tecnologías que hemos utilizado para almacenar y visualizar la información, además de las herramientas en las que nos hemos apoyado para realizar el código y la optimización de selección de algoritmos

3.1. Archivos

El formato de los archivos que más hemos utilizado para visualizar la información y almacenarla son el *Excel o CSV*, sobre todo para observar las tablas de datos y clasificar la información, del mismo modo para el análisis de los reportes empleamos el formato *HTML*. Estos archivos los explicamos de forma más detallada en los siguientes apartados

3.1.1. CSV/ Excel

En el punto de inicial del proyecto, tan solo se contaba con un archivo en formato *CSV* que constituía la base de datos completa de los pacientes, la cual provenía directamente de las fuentes médicas.
El primer paso ha sido convertir a formato *Excel* dicha base de datos para facilitarnos la visualización inicial de filas y columnas, además de las ventajas que tiene el trabajar con este tipo de formatos, que incluye la posibilidad de transformar, transponer y editar columnas

ofilas.

3.1.2. HTML

Aun así, trabajar con los datos en formato tan crudo no facilita el análisis de estos, y no permite tener una visión general de los datos. Por ello, haciendo uso de librerías que se comenta en otros apartados (*Panda Profiling*), se ha generado un informe detallado sobre la base de datos. Dicho informe es en formato *HTML*. Se trata de una página interactiva con numerosas opciones y estadísticas. Esta contiene :

- Detalles de cada columna, como los valores distintos que tiene, el número de veces que aparece cada valor, número de valores nulos, o el rango de valores en caso de columnas con valores numéricos.

- Interacciones entre columnas: la página nos permite seleccionar dos columnas y esta nos mostrará un diagrama de Dispersión mostrando la relación que estas tienen entre ellas.

- Correlaciones: disponemos de distintas estadísticas que miden la correlación entre pares de datos. Tenemos cuatro medidas distintas:

    1. Coeficiente de Correlación de Pearson (r): Nos da un valor entre -1 y 1 por cada par de columnas, siendo -1 correlación negativa, 0 correlación inexistente, y +1 correlación linear positiva. Para calcular r de dos columnas X e Y, se divide la covarianza de X e Y entre el producto de sus desviaciones típicas.

    2. Coeficiente de correlación de Spearman (ρ): Es la correlación monótona entre dos columnas. El valor resultante es el mismo que en el anterior caso, un valor entre -1 y 1. Se calcula dividiendo la covarianza de los coeficientes de X e Y, entre el producto de sus desviaciones típicas.

    3. Coeficiente de correlación de rango de Kendall (τ): Mide la asociación ordinal entre dos columnas. Es similar al Coeficiente de correlación de Spearman, y su

valor es de -1 a 1, -1 siendo correlación negativa, 0 falta de correlación y 1 correlación positiva. Para calcular t de dos variables X e Y, se determina el número de pares de observaciones concordantes y pares de observaciones discordantes. Estas dos magnitudes se restan y se dividen entre el número total de pares.

4. Coeficiente de Correlación de Phik ϕ (κ): Se trata de una nueva forma de analizar el coeficiente de correlación. Se basa en una mejora del Coeficiente de Correlación de Pearson, diseñada para funcionar consistentemente con variables categóricas, ordinales y de intervalos. Es capaz de detectar dependencia no lineal. Además, usa al coeficiente de correlación de Pearson en caso de una distribución de entrada normal bivariada. Estas son características útiles cuando se estudia la matriz de correlación de variables con tipos mixtos.

- Datos perdidos: nos muestra información sobre los *Missing values* de cada columna. Estos datos los podemos encontrar en cuatro formatos: Gráfica simple, Matriz, Mapa de calor o Dendograma.

3.2. Lenguaje de programación

A lo largo del proyecto, se han tenido que programar para diversos fines: el reporte del dataset mencionado en el Capítulo 2, la modificación de las filas y columnas del dataset,y la implementación de algoritmos de clasificación, entre otros.
El lenguaje usado ha sido *Python*, es el principal lenguaje que se utiliza para implementar *Machine Learning*; por su versatilidad, su sentencia sencilla y fácil de leer, y la cantidad de macros que se han implementado para facilitar su uso en esta área. En este proyecto, se ha empleado las siguientes librerías y macros.

3.2.1. Anaconda

Anaconda es una distribución libre y abierta, su principal uso es en *Data Science* y el aprendizaje automático. En esencia, es un gestor de paquetes y entornos que cuenta con

más de 7500 paquetes de código abierto, siendo una suite muy completa para el estudio de datos en *Python*. A continuación se revisarán las bibliotecas útiles y necesarias para llevar a cabo las tareas de programación y visualización de los datos.

Pandas

Pandas es una biblioteca de software libre desarrollada por Simon Brugma, como extensión de *NumPy* (librería que da soporte a la creación de vectores, y matrices multidimensionales de gran tamaño, además de diversas funciones matemáticas para operar con estas). Su principal uso es la manipulación y el análisis de datos de alto rendimiento. La biblioteca fue creada originalmente para la gestión de datos financieros, y se puede usar como alternativa de las hojas de cálculo.

Dada esta explicación, es fácil entender la utilidad para nuestro trabajo. La tabla de datos que se maneja en este proyecto se puede exportar en Pandas como un *DataFrame*, este es una clase de las librerías de *Pandas-Profiling* que se refiere a una tabla bidimensional de tamaño variable, es decir, el reporte generado que explicamos en el capítulo anterior es un *DataFrame* originado de forma muy sencilla con esta biblioteca.

Matplotlib

Se trata de una biblioteca de código abierto para la generación de gráficos de excelente calidad, a partir de datos contenidos en listas o arrays, por lo tanto, tiene como base la biblioteca *Numpy*. Es multiplataforma, puede ejecutarse desde scripts o desde una consola de *Python*. Ha sido de gran utilidad para visualizar de forma gráfica ciertas estadísticas de los datos, lo que simplifica mucho el análisis.

Scikit-Learn

Esta librería ha sido de vital importancia para este proyecto, y es, por tanto, de las más importantes. Se trata de una biblioteca de código abierto que proporciona una amplia gama de algoritmos de aprendizaje supervisados y no supervisados en *Python*. Está construida

sobre *SciPy (Scientific Python)*, e incluye varios algoritmos de clasificación, regresión y análisis de grupos.

Esta librería ha sido útil, en la fase de preprocesamiento e implementación de algoritmos. Una vez terminadas las fases de exploración y limpieza de los datos, se precisaba de una herramienta para su transformación, para presentarlos de la forma óptima para los algoritmos, ya que se pueden comportar de forma inesperada si los datos no parecen más o menos tener una distribución normal. Esta librería ofrece herramientas para todo tipo de transformaciones (lógicas, lineales, algebraicas, no lineales). Estas son algunas de las funciones utilizadas: *MaxAbsScaler, StandarScaler y StandarScalerWrapper*; para la implementación de algoritmos, se han seleccionado las macros asociadas a esta librería: *LightGBM, XGBoostClassifier, RandomForest, ExtremeRandomTees, y Logistic Regression.*

3.3. Herramientas externas

Se ha incluido en el proyecto herramientas con licencia de estudiante, para facilitar la programación, como es *Pycharm*; y para seleccionar los algoritmos con mejores resultados en los análisis, como es *AutoML.*

3.3.1. AutoML

La plataforma que se ha utilizado ha sido *Azure AutoML*[1]. Se trata de un servicio basado en la nube que presenta diferentes opciones para algoritmos de *Machine Learning.* Este presenta diferentes algoritmos tanto de clasificación, regresión y de previsión. Permite probar con distintos hiperparámetros de un modelo dado, además ayuda con la elección de los algoritmos más adecuados y en como pre-procesar la información antes de trabajar con los algoritmos.

Existen dos formas de trabajar con *AutoML*:

- *Azure ML Studio*: no se trabaja con tanto código y se facilita una interfaz gráfica muy

[1] https://docs.microsoft.com/en-us/azure/machine-learning/concept-automated-ml

cómoda.

- *Azure ML Python SDK*: para usuarios experimentados tanto en código como en *Data Science*.Resulta más complicada de usar, pero de esta manera se pueden explotar todas las herramientas que *Azure ML* tiene disponibles.

No se ha optado por ninguna de las dos opciones en solitario,por el contrario se ha escogido una combinación de ambas, haciendo uso de *Azure ML Studio* debido a la comodidad de su interfaz, y recurriendo para la implementación a *Azure ML Python SDK*, con librerías como *azureml.core*. Esta es útil para administración de objetivos de proceso, la creación y administración de espacios de trabajo, además de experimentos, envío de las ejecuciones del modelo, y recibir los resultados y registros de las ejecuciones.

3.3.2. Pycharm

Pycharm es un *IDE* especialmente diseñado para el lenguaje *Python*. Hemos elegido este programa sobre otros tantos, que son parecidos debido a su soporte para el *Data Science* con Anaconda. Ofrece una consola de *Python* con la instalación de Anaconda y la previsualización de los *dataframe*, gráficos, etc. Además, proporciona diversas licencias para estudiantes, por lo que la hace una herramienta ideal para este proyecto.

Capítulo 4

Fundamentos de los algoritmos de clasificación

En este capítulo describiremos los fundamentos teóricos de los algoritmos y modelos implementados para resolver el problema descrito. Se ha realizado una investigación para averiguar la utilidad de cada algoritmo en el proyecto. A continuación, se verán los métodos de preprocesamiento como el escalado de datos y los algoritmos de aprendizaje que hemos elegido y nos han dado mejor resultado. Los segundos, principalmente, se agrupan en algoritmos de regresión logística multi-etiquetas y algoritmos de árboles de decisión.

4.1. Escalado de los datos

Esta etapa pertenece al grupo de preprocesamiento de los datos, es decir, modificarán los datos antes de ser introducidos en los modelos, por lo tanto, conforman el último paso de proyecto antes del modelado. Los algoritmos de *Machine Learning* son sensibles a estos escalados, por lo general son los que pertenecen a redes neuronales o de *clustering* son muy susceptibles a esto, que por el objetivo de este proyecto están descartados.

Existen diferentes formas de preprocesamiento, el escalado de datos que transforma la información dentro de un rango de valores o la normalización que hace que el rango datos

tengan el mismo valor medio y desviación típica. Se debe tener cuidado a la hora de realizar el escalado de los datos, ya que se pueden arruinar los datos de entrada si se realiza una mala normalización, o si se hace una elección descuidada del algoritmo de normalización.

Los beneficios que nos trae realizar este paso son:

- Aumentar disminución de la convergencia del descenso del gradiente.
- Los coeficientes de regresión de los modelos lineales están relacionados con el escalado de datos, por lo que será mejor el resultado si estos están dentro de un rango.
- Los valores extremos afectan negativamente, los de mayor valor prevalecerán sobre los de menor valor en el algoritmo.

4.1.1. MaxAbsScaler

Este método escala los valores de las columnas por el valor máximo absoluto, estos están dentro de un rango [-1,1]. En nuestro caso, como solamente trabajamos con valores positivos, el rango de nuestros datos sera de [0, 1]. Para este fin, se aplica la siguiente transformación a los datos:

$$z = \frac{x}{|x_{max}|}$$

4.1.2. StandardScaler

Este método estandariza los valores asumiendo que estos están normalizados, eliminando la media y escalando a una varianza unitaria. Se hace de la siguiente manera:

$$z = \frac{x - media}{varianza}$$

4.1.3. Sparce Normalizer

Este escalado de datos a diferencia de los anteriores, se produce por filas, no por columnas, aplicando la siguiente fórmula:

$$z = \frac{(x - min(x))}{(max(x) - min(x))} * 100$$

Este preprocesamiento se caracteriza por normalizar matrices en el que la mayoría de sus componentes está formado por ceros.

4.2. Algoritmos

Dentro del amplio campo del *Machine Learning*, encontramos diversos grupos de algoritmos. En este proyecto se trabaja con algoritmos de clasificación, los cuales pertenecen al grupo de aprendizaje automático supervisado. El aprendizaje supervisado trabaja con datos etiquetados, ya que se conoce el grupo al cual se clasifica los atributos. Una definición amplia de este tipo de algoritmos, sería la búsqueda de patrones en los atributos de entrada en la fase de entrenamiento, para poder categorizar y etiquetarlos en subgrupos.

4.2.1. Regresión Logística

La regresión logística es un algoritmo que estima la probabilidad de que una instancia pertenezca a un grupo binario. En nuestro caso, calcula que probabilidad hay de que un paciente pertenece a la etiqueta de diagnóstico, en caso de que fuesen dos etiquetas. La manera en que funciona este algoritmo, es realizando una suma ponderada de las características de entrada, y con el resultado aplica una función logística 4.1.

$$\hat{p} = h_\theta(x) = \sigma(x^T\theta) \tag{4.1}$$

Como se puede deducir observando la función, **n** es el número de características que tiene la función, y las **x** los valores que estas toman en nuestro modelo; por otro lado, θ es la función sigmoidea que se aplica a las variables características:

$$\theta(t) = \frac{1}{1 + \exp(-t)} \tag{4.2}$$

Esta aplicación nos permite normalizar estos valores, de tal manera, que nos devuelve un rango entre 0 y 1. Esta sigue una curva sigmoidea, con forma de S, como se puede apreciar en la imagen 4.1

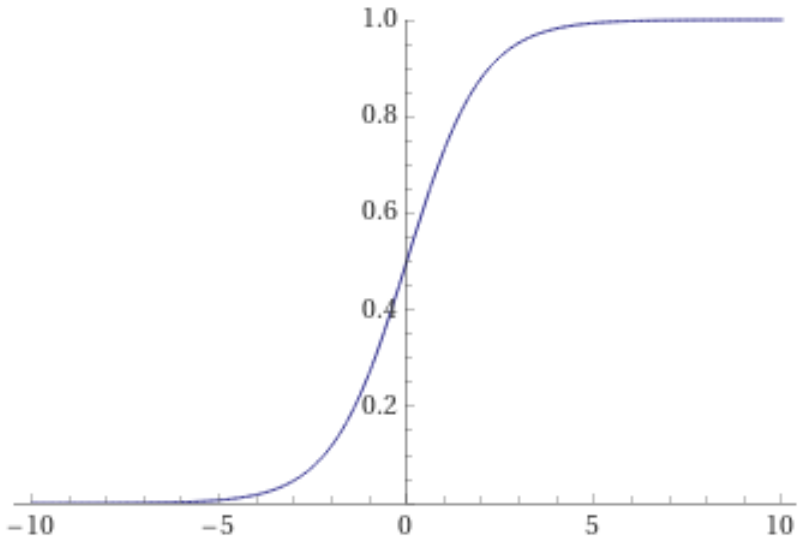


Figura 4.1: Función Sigmoidea

Una vez estimada la probabilidad de que una instancia o característica pertenezca a una de las clases, la estimación de **y** es simple 4.3:

$$\hat{y} = \begin{cases} 0 & \hat{p} < 0,5 \\ 1 & \hat{p} >= 0,5 \end{cases} \tag{4.3}$$

Si el valor es 0 para nuestra característica en la ecuación 4.2, al aplicar la función de probabilidad nos dará un valor negativo y, por lo tanto, no pertenece a la etiqueta seleccionada.

El objetivo de entrenar un algoritmo es encontrar un conjunto de parámetros de la función sigmoidea que estime la mayor probabilidad de etiquetas positivas para nuestras variables característica, que se llama función de coste. La utilizada es la siguiente:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log(\hat{p}^{(i)}) + (1-y^{(i)})\log(1-\hat{p}^{(i)}\right] \tag{4.4}$$

Lo que hace esta función es realizar una corrección por cada instancia, aplicando a cada probabilidad calculada el logaritmo de la media negativa. Cuanto menor sea este valor, mejor es el resultado obtenido. Aunque es una función complicada de minimizar, es convexa por lo que existe un mínimo global.

Como se explica al inicio de la definición de este algoritmo, este se utiliza para el cálculo de dos etiquetas, es decir, es binario. Si bien no es nuestro caso, tenemos varias clases y necesitamos aplicar dicho algoritmo con más de dos etiquetas diferentes, para ello existe el algoritmo **Softmax Regresión** o lo que viene a ser lo mismo, **Regresión Logística Múltiple**. Hay que hacer hincapié en que es Multiclase, es decir, permite varias etiquetas o clasificación, no diferentes salidas para una misma etiqueta.[2]

Así bien, el modelo primero calcula la puntuación para cada clase, k:

$$s_k(x) = x^T * \theta^{(k)} \tag{4.5}$$

Después del primer paso, se encarga de calcular la probabilidad de que una variable o característica pertenezca a una clase k:

$$\hat{P}_k = \sigma(s(x))_k = \frac{\exp s_k(x)}{\sum_{j=1}^{K} \exp s_j(x)} \tag{4.6}$$

En la ecuación 4.6:

- K es el número de clases
- s(x) es el vector que contiene las puntuaciones de las instancias
- $\sigma(s(x))_k$ es la estimación de que la probabilidad de la instancia x pertenezca a la clase k, dando la puntuación por cada variable o instancia.

Como se puede entender, esta función, al hacer por cada conjunto de instancia una probabilidad para todas las clases que existe, se debe elegir la que mayor probabilidad tiene:

$$\hat{y} = \arg\max_x \sigma(s(x))_k = \arg\max_x s_k(x) = \arg\max_x (\theta^{(k))^T} * x \tag{4.7}$$

Una vez calculada la probabilidad, tenemos que minimizar la función de coste, en nuestro caso se llama entropía cruzada, que nos indica como de bien estimadas están las probabilidades calculadas con las diferentes clases.

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\left(y_k^{(i)}\log(\hat{p}^{(i)})\right) \tag{4.8}$$

Si sustituimos k = 2, nos encontramos con la ecuación 4.8, por lo que tienen las mismas características. Una vez explicada la versión teórica del algoritmo, en la parte práctica se traduce a la selección de los valores hiper parámetros:

- **multi_class**: lo primero que hay que entender es, que por defecto, la regresión logística no puede ser usada para tareas de clasificación que tienen más de dos etiquetas de clase, también llamadas clasificación multi-clase. Por ello, Scikit nos ofrece herramientas para modificar el problema y hacer que acepte tareas multi-clase. Hemos implementado las siguientes:

  - **One vs Rest**: consiste en utilizar un solo clasificador por cada clase. Requiere que estas sean columnas binarias.
  - **Multinominal**:es el modelo de regresión para que sea compatible con la perdición de clases con múltiples etiquetas.

- **C**: se trata de la inversa de la fuerza de regularización, dada por la siguiente fórmula (donde C es un número de tipo float positivo):

$$C = \frac{1}{\lambda} \tag{4.9}$$

  Este valor nos proporciona el número de variables a elegir, que se van a sobre-ajustar al modelo, sería como los grados de libertad que tiene el modelo sobre las variable. Los valores que se han utilizado son: 0,5, 10000 y 4700, estos se han ido eligiendo aleatoriamente.

- **Class_Weight**: determina si se le da más peso a ciertas columnas sobre otras. Se han utilizado los valores null y balanced, esta última sigue la siguiente fórmula:

$$\frac{nvariables}{nclases * np.bincount(y)} \tag{4.10}$$

- **Penalty**: especifica la norma de la penalización. Los valores opcionales del parámetro de penalización son "l1ƨ "l2", estas son sus definiciones:

  - **L1 - Lasso Regression**: nos ayuda a mejorar la *Prediccion Accurate* y disminuir el número de coeficientes, añadiendo una penalización absoluta a lambda. Esto provoca que el modelo se entrene con menos variables, las cercanas a 0 se eliminan.

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{i=1}^{p} |\beta_j| \tag{4.11}$$

  - **L2 - Ridge Regression**: nos permite analizar los datos para encontrar si existe alguna relación muy estrecha entre los coeficientes, para evitar tener una varianza muy alta, añadiendo una penalización cuadrática a lambda.

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{i=1}^{p} \beta_j^2 \tag{4.12}$$

  El valor que se ha elegido es la regularización de L2, porque es el que combina con el hiperparámetro multinominal.

- **Solver**: determina la forma de optimización para la función de pérdida de Regresión logística. Hay cuatro algoritmos para elegir, que son *liblinear, lbfgs, newton-cg y sag*. Se ha elegido el método *lbfgs*, lo que nos permite realizar una minimización de la función de optimización, con menos costes de memoria.

4.2.2. Arboles de decisión

Como indica su nombre, este tipo de algoritmos toman la decisión de clasificación como un modelo tipo árbol. Es importante destacar, que se puede utilizar, tanto en algoritmos de regresión como en clasificación debido a la facilidad de la toma de decisiones. La decisión de la toma de qué rama del algoritmo está dando mejor resultado se decide a través de una lógica en la implementación, por lo que, esto es lo que diferencia a los árboles entre otros detalles como el coste de memoria.

Se puede decir que este tipo de algoritmos no necesitan mucho preprocesamiento, debido a que este suele ir mejorando por iteración, ya que, aprende de los resultados de los subconjuntos del dataset anteriores. Pero es importante añadir que no es recomendado usar con columnas que añadan ruido porque puede provocar *overfitting* rápidamente.

Random Forest Classifier

Este clasificador se encuentra en el grupo de algoritmos de árboles de decisión por conjunto, esto quiere decir, que la decisión final se toma de la selección del resultado medio de cada predictor. El recurso que se utiliza es diferente para cada árbol utilizado, estos están alimentados con una porción totalmente diferente del dataset, reutilizando estas partes para cada predictor.

Se utilizan, tanto para algoritmos de clasificación como de regresión. En este caso, como se puede deducir, los predictores tienen un alto nivel de *bias* por haber sido entrenado con partes del dataset, tras la agregación se reduce el *bias* y la *varianza*. Lo que se produce con la agregación de los resultados, es que suelen tener un bias parecido pero una varianza menor con respecto a si hubiese sido entrenado con el dataset completo.

- **n_estimators**: el número de árboles de decisión creados para la predicción. La media utilizada en este proyecto es de 10.

- **max_features**: la función o fórmula usada para que cada nodo seleccione un conjunto

de atributos. Se ha seleccionado log2 en todos los casos.

$$\log max\_features = \sqrt{n\_features}$$

- **min_sample_leaf**: determina el número mínimo de hojas requeridas para cada nodo. Se han seleccionado 0,01 a 0,03. Se realiza un truncado del valor hacia arriba.
- **min_sample_split**: el mínimo número de conjuntos para cada hoja del nodo. Se ha seleccionado 0,1 y 0,01.
- **criterion**: la función mide la calidad de cada partición. Es la que nos permite seleccionar la mejor partición de los atributos. La fórmula es:

$$\sum_{k} p_i \left(1 - p_i\right)$$

Extreme Random Trees

Los árboles extremadamente aleatorios añaden a los árboles anteriores una nueva característica, esta vez, el corte a dividir cada nodo es aleatorio. No se escoge una división que nos dé el mejor resultado, sino que se realiza de manera aleatoria. . También, habría que añadir que a la hora de escoger el subconjunto para entrenar, no lo reemplaza, por lo que se utiliza el pasting. Se utilizan los mismos parámetros que en *Random Forest*.

Gradient Boosting

Este tipo de algoritmos entran en la categoría también de entrenamiento con árboles de decisión por conjuntos, esta vez los árboles que se van creando aprenden de los .$^{\text{er}}$rores de sus árboles predecesores".Para la implementación de este algoritmo, se han utilizado dos librerías: *LightGBM y XGBoost*.

El uso de estas dos librerías nos proporcionan diferentes ventajas:

- **LightGBM**: utiliza la técnica de GOOS, en la que las ramas que crecen son las que

están generando mejores resultados. Esto es debido a que si crece mucho el conjunto de datos empleado, y estos datos tienen un mayor gradiente, crea un subconjunto con el que se realiza nuevamente las pruebas.

- **XGBoost**: se controla su crecimiento por el parámetro de aprendizaje, por lo que los mismos niveles del árbol van a tener las mismas características hasta que se llegue al máximo o hasta que se consiga el mejor resultado.

Utiliza los mismos hiperparámetros usados con anterioridad, añadiendo estos extras:

- **eta**: es el coeficiente de aprendizaje. Suele acomodarse en cada paso. Se ha empleado 0.5.
- **alpha**: término de regularización L1. El valor usado ha sido 0,4.
- **lambda**: término de regularización L2. El valor utilizado ha sido 0,7.
- **bosting type**: nos indica el tipo de árbol de decisión empleado. En nuestro caso es: *gbdt*, que es *Gradient Boosting Decision Tree*.

Capítulo 5

Fundamentos de los métodos de evaluación

En la fase de evaluación, el objetivo es determinar como de buenos son los algoritmos utilizados. Al tratarse de un proyecto relacionado con el sector sanitario, se deben evaluar los modelos, para asegurarnos de que son fiables, y que consiguen la eficacia clínica deseada.

5.1. Accuracy

Sirve para calcular el porcentaje de aciertos. Es la métrica más intuitiva y directa para medir un modelo. Matemáticamente hablando, es el porcentaje de predicciones correctas. Se expresa mediante un número del 0 (ninguna predicción correcta) al 1 (todas las predicciones son correctas), y tiene la siguiente fórmula:

$$accuracy = \frac{\#predicciones\_correctas}{\#predicciones\_totales}$$

O su fórmula binaria:

$$accuracy = \frac{Verdaderos\_positivos + Verdaderos\_negativos}{Verdaderos\_positivos + Verdaderos\_negativos + Falso\_positivo + Falso\_negativo}$$

Esta medida es útil por lo general, en caso de que tengamos un conjunto de datos irregular o no balanceado, no nos lo mostrará.

5.1.1. Precisión

La precisión es la observación de los valores positivos predichos en la medida de *Accuracy*. La fórmula es la siguiente:

$$precision = \frac{Verdaderos\_positivos}{Verdaderos\_positivos + Falsos\_positivos}$$

Vemos que esta medida ignora completamente los casos en los que el resultado es negativo. Por esta razón, suele complementarse con la medida *Recall*.

5.1.2. Recall

Esta métrica se refiere a la exhaustividad o sensibilidad del modelo. Esto se refiere a la fracción de todos los resultados positivos reales, que coinciden con el valor que deberían tener:

$$recall = \frac{Verdaderos\_positivos}{Verdaderos\_positivos + Falsos\_negativos}$$

Estas últimas medidas son las que usaremos para determinar la matriz de confusión, que se ve en los resultados del Capítulo 6

5.1.3. F1

F1 es la combinación de la medida de precisión y *recall*. Esta combinación se hace a través de una media armónica. Esta nueva aplicación pondera con mayor peso a las medidas con valores más pequeños, por lo que sólo se obtendrá un valor alto cuando precisión y *recall* sean ambas altas. El dominio de esta medida es 0 a 1, su fórmula es:

$$F1 = \frac{2}{\frac{1}{precision} + \frac{1}{Recall}}$$

Es especialmente útil cuando se trabaja con bases de datos no balanceadas. Parecido a las funciones anteriores, existen dos tipos:

- *F1 micro*: calcula la media armónica globalmente, contando verdaderos y falsos positivos y verdaderos y falsos negativos.
- *F1 macro*: utiliza los datos individuales tanto de verdaderos y falsos positivos como de verdaderos y falsos negativos. Hace la media entre precisión y exhaustividad sin considerar la proporcionalidad de las clases en los datos.

5.1.4. Specifty

Esta medida nos indica los valores negativos que se han predicho como negativos. Es una fórmula que no la calculamos directamente, pero es útil para las siguientes medidas:

$$specifty = \frac{Verdaderos\_negativos}{Verdaderos\_negativos + Falsos\_positivos}$$

5.2. ROC Curve

Es un gráfico que muestra el resultado del rendimiento de los modelos en todos los umbrales. Para ello, realiza una comparación de las medidas de *Recall* y la evaluación de los falsos positivos. Esto nos permite comprender la capacidad del modelo de distinguir entre las clases etiquetas.

La fórmula de falsos positivos es la siguiente:

$$FPR = \frac{Falsos\_positivos}{Falsos\_positivos + Verdaderos\_negativos}$$

O también, si ya se tiene calculado Specifity:

$$FPR = 1 - Specifity$$

5.3. AUC-Area under the ROC Curve

Es el área bidimensional debajo de la *ROC curve*. AUC proporciona una medida agregada de rendimiento teniendo en cuenta distintos umbrales, y resulta que la probabilidad del modelo clasifica un ejemplo positivo aleatorio más alto que un ejemplo negativo aleatorio.

Podemos destacar el parámetro *average*. Determina el tipo de promedio que se aplica a los datos. Si no se pasa ningún valor al argumento, se devuelven resultados por cada clase. Admite tres posibles valores:

- *micro*: calcula métricas globales considerando cada etiqueta de la matriz por separado.
- *macro*: calcula métricas para cada etiqueta, y encuentra su media no ponderada.
- *weighted*: calcula métricas para cada etiqueta, y encuentra su media, ponderada por el número de instancias verdaderas en cada etiqueta.

5.3.1. Average Precision Score

Calcula la precisión del promedio (AP) del resultado de la predicción. AP resume la curva de precisión y exhaustividad, en forma de media ponderada de la precisión alcanzada en cada umbral. Se trata de una serie, en la que en cada iteración se retro-alimenta del resultado de la iteración anterior, como se puede deducir de su fórmula:

$$AP = \sum_{n} (R_n - R_{n-1}) P_n$$

Los valores R_n *y* P_n se refieren a la precisión y a la exhaustividad alcanzadas en el umbral enésimo.

La función que calcula el promedio acepta tres valores en el parámetro *average*, que determina el tipo de promedio aplicado a los datos:

- *Micro*: es parecido al valor *micro* del apartado anterior, calcula métricas globales considerando cada elemento de la matriz indicadora de etiquetas por separado.
- *Macro*: calcula la media no ponderada, utilizando métricas para cada etiqueta.
- *Weighted*: calcula métricas para cada etiqueta, y encuentra su media ponderada.

Capítulo 6

Resultados

En este capítulo se muestran los resultados obtenidos de aplicar los algoritmos del Capítulo 4, y medir los resultados con las medidas del Capítulo 5. En nuestro caso, al apoyarnos de la herramienta de *AutoML*, nos ha permitido disminuir el error de cálculo y únicamente se ha aplicado aquellos algoritmos que se aconsejan en la herramienta.

Tras la selección de *AutoML*, para conocer los mejores resultados se han combinado los diferentes tipos de preprocesamiento, en nuestro caso: *MaxAbsScaler, StandarScalerWrapper y SparseNormalizer*; con los siguientes algoritmos: *RandomForest, LightGBM, XGBoostClassifier, ExtremeRandomTrees y Logistic Regression*. Estas combinaciones en algunos dataset generarán unos resultados de test muy bajos, y se han omitido en este caso, para no generar confusión sobre el resultado obtenido. Dividiremos los resultados por dataset, para conocer el valor de las medidas indicadas, y finalmente, se mostrará el resultado con las muestras de test de la mejor combinación de cada dataset. Además, para la ejecución de dicha herramienta se ha permitido el uso de *deep learning*, para conocer qué columnas han influido más en el resultado de dicha clasificación.

6.1. Resultados de las medidas en los diferentes Dataset

En este punto se explicarán las variables utilizadas, además de mostrar las medidas obtenidas por los algoritmos seleccionados en cada uno de los Dataset.

6.1.1. Dataset I

En este apartado tratamos el dataset con síntomas separados en diferentes columnas y con fecha rellenas con el valor medio. Para empezar vamos a indicar que variables ha considerado más relevantes a la hora de clasificar (Figura 6.1):

- DCG Biopsia AP1
- DCG ATG 2
- LIE DCG %GD
- DCG A-PDG

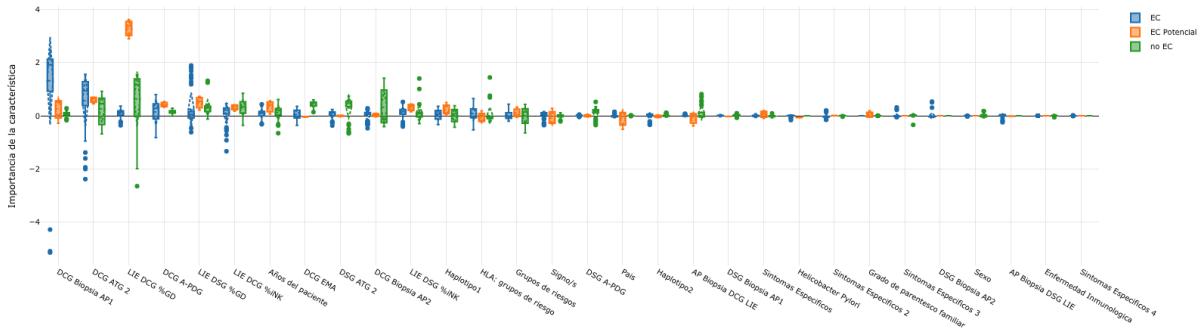


Figura 6.1: Selección de Features Relevantes: Dataset I

Como podemos observar en la figura, las columnas indicadas con anterioridad son las que mejor porcentaje de acierto tienen sobre si padece la enfermedad. Cuanto más alejado del 0, más han influido a la hora de pronosticar la enfermedad. Para estos algoritmos que nos han dado los mejores resultados, se les ha realizado el siguiente preprocesamiento:

- Random Forest con Standard Scaler Wrapper

- LightGBM con Max Abs Scaler
- XGboostClassifier con Max Abs Scaler
- Extreme Random Forest con Max Abs Scaler
- Logistic Regressioin con Standard Scaler Wrapper

Pasaremos a mostrar los medidas que se han obtenido en los algoritmos en la tabla 6.1.

Porcentaje por Algoritmo					
Algoritmo	Accuracy	AUC media	Media Precision Score	Balanced Accuracy	F1
XGboostClassifier	92,0 %	98 %	98 %	81 %	92 %
LightGBM	92,50 %	99,0 %	98,0 %	80,0 %	92,0 %
Random Forest	87,0 %	97,5 %	95,0 %	65,0 %	87,0 %
ExtremeRandomTrees	85 %	96,0 %	96,0 %	94,0 %	85,0 %
Logistic Regression	85,0 %	94,0 %	87,0 %	70,0 %	85,0 %

Cuadro 6.1: Resultado del Dataset I

6.1.2. Dataset II

En este apartado tratamos el dataset con síntomas unidos en una columna y con fecha rellenas con el valor medio. Para empezar vamos a indicar que variables ha considerado más relevantes a la hora de clasificar:

- DCG Biopsia AP1
- DCG ATG 2
- DSG ATG 2
- LIE DCG %GD

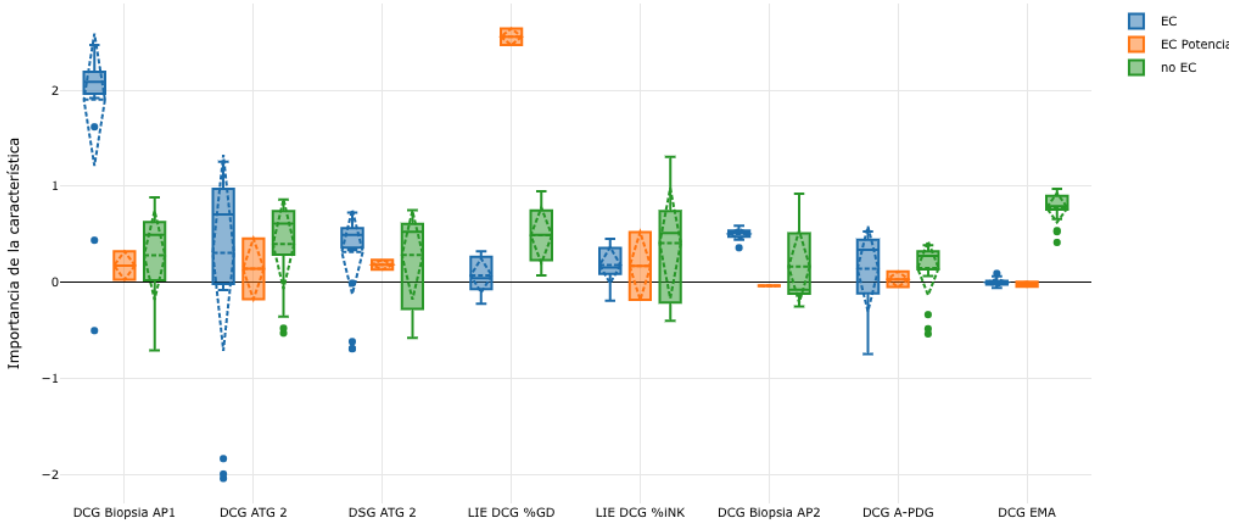


Figura 6.2: Seleccion de Features Relevantes: Dataset II

Como podemos observar en la Figura 6.2, las columnas indicadas con anterioridad son las que mejor tienen porcentaje de acierto sobre sí padece la enfermedad. Cuanto más alejado del 0, más han influido a la hora de pronosticar la enfermedad.

Pasaremos a mostrar los medidas que se han obtenido en los algoritmos en la tabla 6.2.

Porcentaje por Algoritmo					
Algoritmo	Accuracy	AUC media	Media Precision Score	Balanced Accuracy	F1
Random Forest	86,6 %	95,1 %	89,3 %	69,3 %	86 %
LightGBM	88,8 %	94,0 %	90,0 %	80,0 %	88,0 %
XGboostClassifier	87,0 %	94,0 %	89,0 %	73,0 %	87,0 %
ExtremeRandomTrees	83,3 %	94,0 %	87,0 %	61,0 %	83,0 %
Logistic Regression	78,0 %	68,0 %	85,0 %	63,0 %	78,0 %

Cuadro 6.2: Resultado del Dataset II

Para estos algoritmos que nos han dado los mejores resultados, se les ha realizado el siguiente preprocesamiento:

- XGboostClassifier con Max Abs Scaler
- LightGBM con Max Abs Scaler
- Random Forest con Standard Scaler Wrapper
- Extreme Random Forest con Max Abs Scaler
- Logistic Regressioin con Max Abs Scaler

6.1.3. Dataset III

En esta sección estamos tratando el dataset con síntomas separados en diferentes columnas y con fecha incompleta. Para empezar vamos a indicar que variables ha considerado más relevantes a la hora de clasificar:

- DCG Biopsia AP1
- LIE DCG %INK
- DCG ATG 2
- LIE DCG %GD

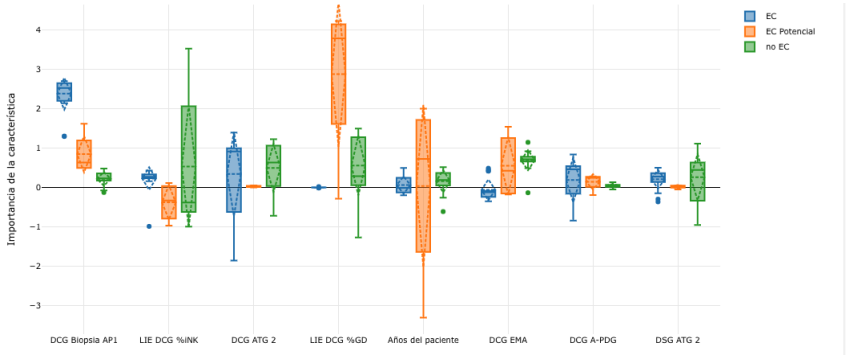


Figura 6.3: Seleccion de Features Relevantes: Dataset III

Como podemos observar en la Figura 6.3, las columnas indicadas con anterioridad son las que mejor tienen porcentaje de acierto sobre sí padece la enfermedad. Cuanto más alejado

del 0, más han influido a la hora de pronosticar la enfermedad. Pasaremos a mostrar los medidas que se han obtenido en los algoritmos en la tabla 6.3.

Porcentaje por Algoritmo					
Algoritmo	Accuracy	AUC media	Media Precision Score	Balanced Accuracy	F1
XGboostClassifier	95 %	99 %	98 %	83,3 %	95,0 %
LightGBM	95,0 %	98,0 %	96 %	83 %	95 %
RandomForest	80 %	96 %	94,3 %	58 %	80 %
ExtremeRandomTrees	90 %	97 %	95 %	79 %	90 %
Logistic Regression	82 %	95 %	89 %	60,1 %	82 %

Cuadro 6.3: Resultado del Dataset III

Para estos algoritmos que nos han dado los mejores resultados, se les ha realizado el siguiente preprocesamiento:

- XGboostClassifier con Standard Scaler Wrapper
- LightGBM con Max Abs Scaler
- Random Forest con Max Abs Scaler
- Extreme Random Forest con Max Abs Scaler
- Logistic Regression con Max Abs Scaler

6.1.4. Dataset IV

En este apartado tratamos el dataset con síntomas en una sola columna y con fecha incompleta. Para empezar vamos a indicar que variables ha considerado más relevantes a la hora de clasificar:

- DCG Biopsia AP1
- DCG ATG 2
- LIE DCG %GD
- DCG A-PDG

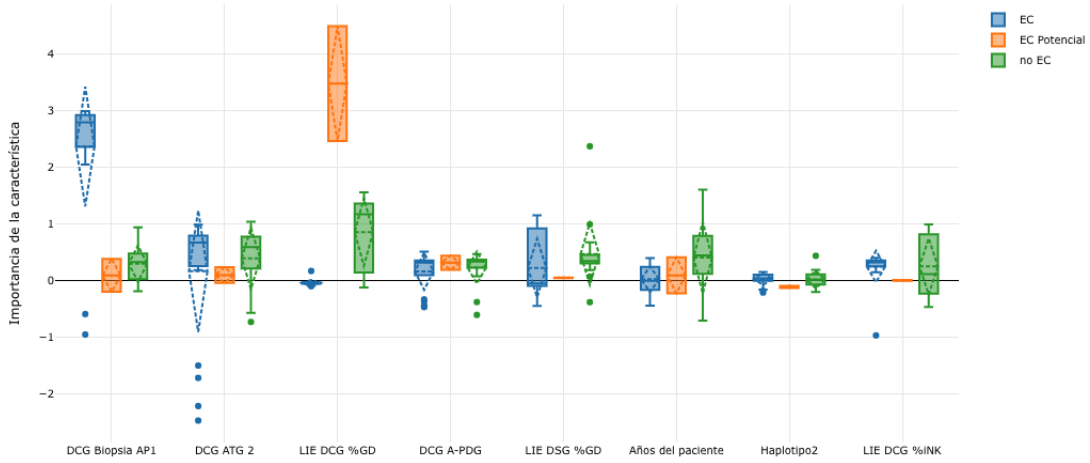


Figura 6.4: Seleccion de Features Relevantes: Dataset IV

Como podemos observar en la Figura 6.4, las columnas indicadas con anterioridad son las que mejor porcentaje de acierto tienen sobre sí padece la enfermedad. Cuanto más alejado del 0, más han influido a la hora de pronosticar la enfermedad.

Pasaremos a mostrar los medidas que se han obtenido en los algoritmos en la tabla 6.4.

Porcentaje por Algoritmo					
Algoritmo	Accuracy	AUC media	Media Precision Score	Balanced Accuracy	F1
XGboostClassifier	85 %	94 %	89 %	68 %	69 %
LightGBM	84 %	95 %	90 %	64 %	84 %
Random Forest	85 %	94 %	89 %	68 %	69 %
ExtremeRandomTrees	81 %	94 %	88 %	59 %	81 %
Logistic Regression	80 %	94 %	87 %	59 %	56 %

Cuadro 6.4: Resultado del Dataset IV

Para estos algoritmos que nos han dado los mejores resultados, se les ha realizado el siguiente preprocesamiento:

- XGboostClassifier con Standard Scaler Wrapper
- LightGBM con Max Abs Scaler
- Random Forest con Standard Scaler Wrapper
- Extreme Random Forest con Max Abs Scaler
- Logistic Regressioin con Standard Scaler Wrapper

6.2. Comparación dataset tests

Existen muchas maneras de seleccionar la parte de test de un dataset, en nuestro caso nos hemos decantado por una división por porcentaje, el 70 % de los datos del dataset se destinan a validación y entrenamiento y el otro 30 % a la parte de prueba. Los resultados obtenidos se muestran en la tabla 6.5.

Porcentaje test por Dataset				
Datasets	Algoritmo	Pre procesimiento	AUC	Accuracy
Dataset III	Random Forest	Standard Scaler Wrapper	91,5 %	82 %
Dataset IV	XGBoost Classifier	Max Abs Scaler	95 %	88 %
Dataset II	XGBoost Classifier	Standard Scaler Wrapper	94,5 %	86 %
Dataset I	XGBoost Classifier	Standard Scaler Wrapper	94,9 %	87 %

Cuadro 6.5: Resultado del test

Para complementar esta información, se ha generado los siguientes gráficos correspondientes a las matrices de confusión. La Figura 6.5 es el resultado del Dataset I.

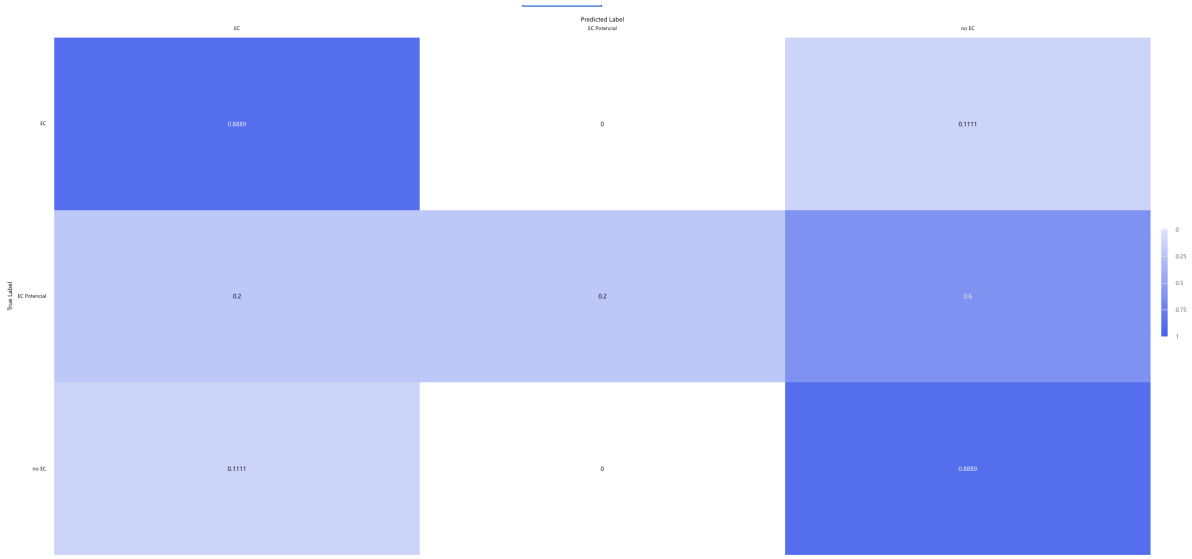


Figura 6.5: Matriz confusión Dataset I

Para el Dataset II, se ha generado la matriz de confusión mostrada en la Figura 6.6.

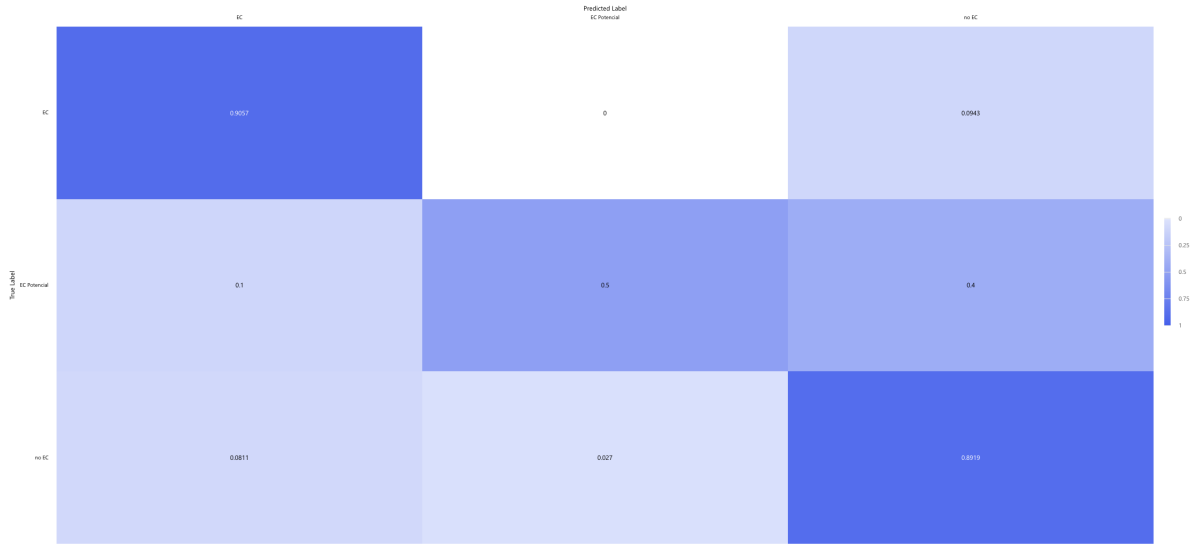


Figura 6.6: Matriz de confusión Dataset II

Para el Dataset III, se ha generado la matriz de confusión mostrada en la Figura 6.7.

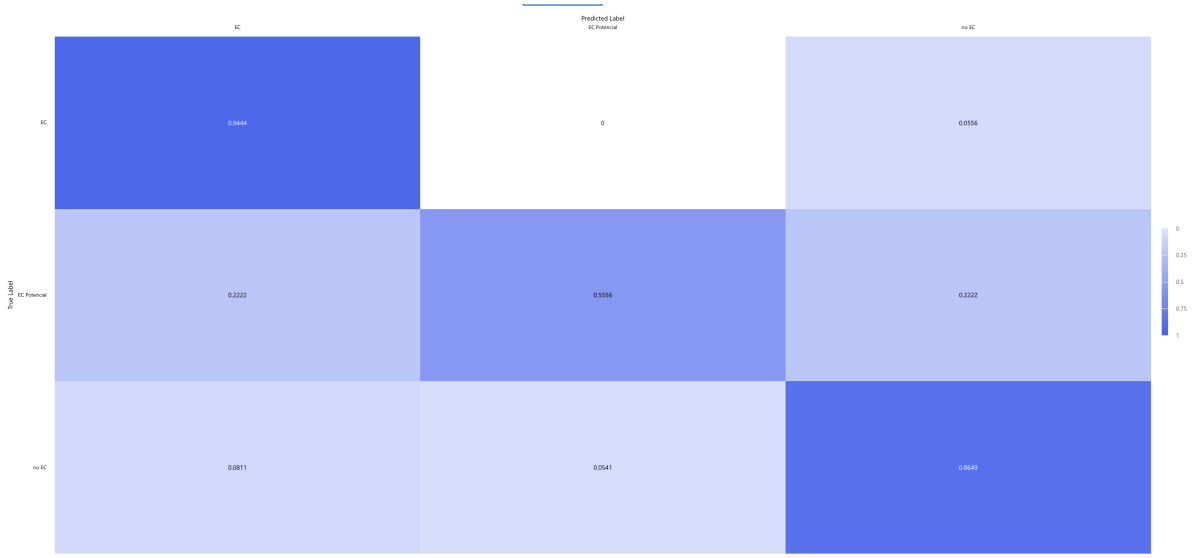


Figura 6.7: Matriz de confusión Dataset III

Para el Dataset IV, se ha generado la matriz de confusión mostrada en la Figura 6.8.

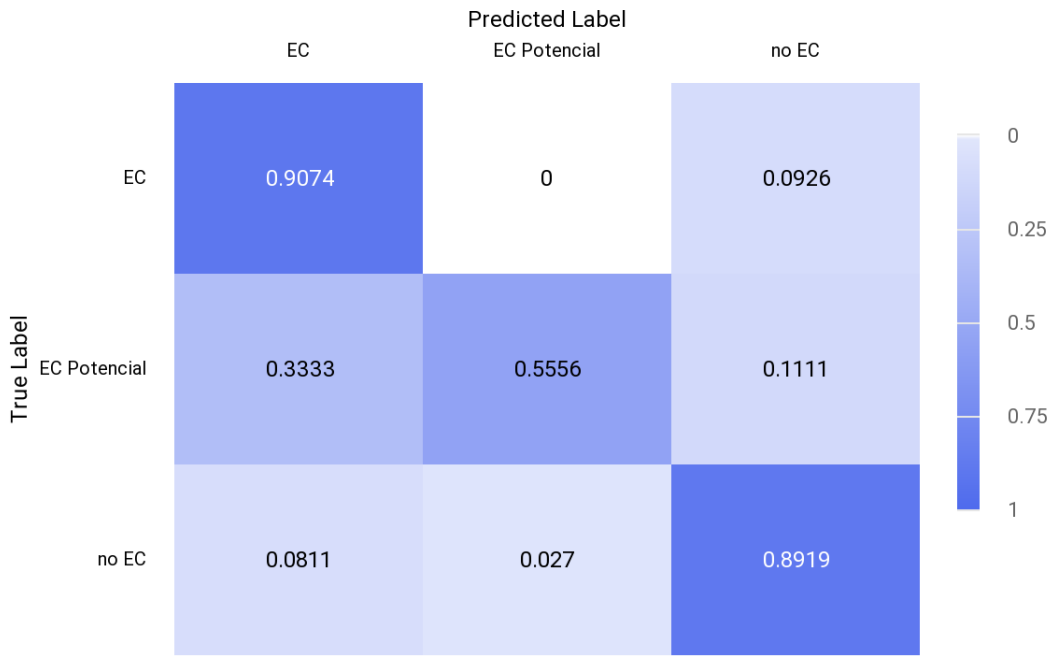


Figura 6.8: Matriz de confusión Dataset IV

Estos gráficos muestran la relación entre las etiquetas predichas correctamente y las etiquetas que no se han predicho correctamente y están generando confusión entre lo que debería predecir y lo que realmente ha predicho. Son los valores obtenidos de *recall y precision*.

Capítulo 7

Conclusión y trabajo futuro

En este último apartado realizaremos un resumen de los resultados obtenidos, una pequeña comparación con otros proyectos y las líneas futuras que debería seguir.

7.1. Conclusión

Este trabajo tiene como objetivo el desarrollo de un algoritmo de *Machine Learning*, para la identificación y clasificación, de los pacientes que padecen la enfermedad de celiaquía. Los resultados obtenidos han sido suficientes para poder corroborar la veracidad de un diagnóstico automático y fiable, realizaremos una explicación detallada de los pasos y puntos que han sido cruciales, para que dicha seguridad haya sido efectiva.

7.1.1. Tratamiento de la información

Como era de esperar, la base de datos no tenía suficiente información, por diversas razones, por lo que al inicio del proyecto, lo primero que se planteó, fue analizar exhaustivamente qué valores tenían los atributos o campos del Excel. Es un proceso lento e importante, porque categorizar la información nos permite reducir el número de pacientes con información vacía o nula, entre otras razones, esto se daba por representar un mismo dato en diversas columnas, lo que tiende el algoritmo a recibir ruido y se produzca un *overfitting* de las medidas;

además, no únicamente este tipo de tratamiento es importante, sino también la reducción de valores vacíos, a través, de un proceso de selección en el que extraíamos los valores mínimos o categorías cero. Otro dilema del que partimos, es que, la mayoría de algoritmos de *Machine Learning* no permiten valores de cadena de caracteres, esto obviamente se soluciona con un paso intermedio, entre la categorización y la aplicación/ejecución de los algoritmos; este paso es la codificación sobre las columnas transformadas para ver, que o cuáles, eran las codificaciones que mejor se adaptaban al tipo de datos y categorías correctamente. Es un apartado fundamental más importante que el de transformación, ya que, muchos de los resultados obtenidos antes de esta codificación eran bastante bajos, en comparación a los que hemos obtenido. Y gracias a este fuerzo previo de prueba-error, acertamos con los resultados finales.

Cabe destacar, que por lo general, este proceso previo suele infravalorarse, sobre todo, en proyectos que la información o cantidad de datos es elevada, sin embargo, es la solución que hay que practicar preliminarmente, debido a que la mayoría de algoritmos actuales aprenden de la información y, por lo tanto, cuanto más procesada esté mejor entenderá y lo aprenderá.

7.1.2. Selección de Algoritmos

En este proyecto nos hemos centrado en los algoritmos de clasificación. Los resultados obtenidos han sido por la sintonización de los hiperparámetros descritos en el Capítulo 4 y de probar los diferentes preprocesamientos con los distintos hiperparámetro. Los resultados obtenidos han sido los esperados, si nos fijamos en la matriz de confusión de los mejores algoritmos podemos concluir que, el valor mínimo medio para identificar correctamente que un paciente tiene la enfermedad es del 89 %, y para identificar correctamente que no tiene la enfermedad es del mismo valor. De lo que nos percatamos, es que los pacientes con la enfermedad potencial suele generar error de confusión, para diferenciarse de las etiquetas de EC y no EC, esto puede ser debido a cantidad de datos de algunas columnas. Los valores mínimos son en torno a 10 % en casos que si tiene la enfermedad, y un 60 % en los que no tienen la enfermedad. Esto sobre todo, ocurre en los dataset con los síntomas separados y

las fechas completas, es decir, el dataset II, aunque es uno de los que mejor rendimiento nos está proporcionando, puede generarnos problemas en caso de que el paciente no tenga la enfermedad y la diagnostique como EC potencial.
En cuanto, a los algoritmos y las combinaciones con pre-procesamiento las que contienen Standard Scaler Wrapper suelen ser bastante alto y el que mejor resultado da en test en general es el XGBoost Classifier.

7.2. Trabajo futuro

El desarrollo de este proyecto nos ha permitido conocer a fondo la información tratada y utilizada de la parametrización de los algoritmos. Por lo tanto, una vez conocido a fondo estas principales características, exponemos una lista de futuros desarrollos, para implantar nuevas metodologías a la hora de entrenar un algoritmo que detecte la enfermedad celiaca. Se ha dividido en función de la información recogida y de los algoritmos a utilizar:

7.2.1. Información recogida

- La columna de diagnóstico se podría mejorar en cuanto a las etiquetas que en la actualidad contienen, si se realiza una revisión con los médicos. La propuesta es la siguiente [4]:

    - EC asintomático o silente: aquellos que padecen la enfermedad, pero no tienen síntomas aparentes
    - EC clásica: son los que presentan los síntomas de problemas digestivos.
    - EC no clásica: no presenta los síntomas de un cuadro típico de celiaquía como malabsorción.
    - EC subclinica: con manifestaciones por debajo de un cuadro de celiaquía.
    - EC sintomática: con síntomas evidentes de problemas digestivos.

- EC potencial: pacientes que dan serología positiva pero con biopsia intestinal normal.
- No EC: no padece la enfermedad

Se sugiere dicha división porque se acerca más a un posible estado de un paciente real.

- Mejora del grado de relación familiar: dentro del propio dataset se podría añadir una columna que asocie a diversos pacientes si tienen una EC positiva o dudosa, esto permitiría que los síntomas, signos o riesgos en diversos pacientes se puedan correlacionar entre los familiares.

- Análisis continuo de los pacientes potenciales: un análisis del paciente durante un periodo de tiempo, que decida el médico, podría ayudarnos en la evolución de los síntomas, signos y enfermedades que pueda desarrollar el paciente con el tiempo.

- La forma de recoger la información: esto es importante porque la información ha necesitado de una transformación y una selección de variables profunda, debido a que esta era pobre en datos y en cantidad. Si la manera de obtener la información de los signos y los síntomas es más fácil, mejorará la calidad.

  Se propone hacer esta recogida a través de una aplicación de escritorio o tablet en el caso de los médicos, y para que los pacientes recojan su estado actual, es decir, síntomas diarios, se recomienda una aplicación de móvil o tablet. Los médicos podrían acceder a la información de los pacientes del estudio continuamente y rellenar los datos que reciba de los análisis.

- Se añadiría un campo en el dataset en el caso de que el paciente esté embarazado o no, ya que esto junto con otros síntomas son alicientes a la hora de desarrollar la enfermedad.

- Asociaría imágenes de las colonoscopias a este proyecto, aunque no sean parte del dataset, se podría utilizar nueva información para analizar si estas imágenes determinan algún tipo de anomalía.

7.2.2. Algoritmos

- Se recomienda el uso de algoritmos de redes neuronales (DNN), esto podría permitir distribuir los pesos de las neuronas en los valores más importantes, por ejemplo en el caso de la columna HLA los valores de DQ2.5 que tengan doble dosis pesarán más que aquellos que tienen DQ8 o DQ2.2 con una dosis, esto mejoraría el etiquetado de los datos [5]

- En el caso de que se utilice imágenes, usar *convolutional neural network*, para indicar si se ha producido una atrofia de las vellosidades.

Capítulo 7

Conclusions and Future work

In this last chapter we will summarize the results, we will compare it with other projects and see what future work is possible.

7.1. Conclusion

The objective of this work is the development of a Machine Learning algorithm, which will be useful for the identification and classification of patients suffering from celiac disease. The results obtained have been sufficient to corroborate the veracity of an automatic and reliable diagnosis. We will explain the steps and points that have been crucial for the reliability of the project.

7.1.1. Information processing

As expected, the database did not have enough information, for various reasons, so at the beginning of the project, the first thing that was proposed was to analyze what values the attributes or fields of the excel had. It is a slow and important process, because categorizing the information allows us to reduce the number of patients with empty or null information, among other reasons, this was due to representing the same data in different columns, which provokes noise in the algorithm and produces measure overfitting. The

reductiontion of empty values is a very important step too, through a selection process in which we extracted the minimum values or the zero categories. Another dilemma from which we start is that most algorithms do not allow string values, this is obviously solved with a intermediate step, between the categorization and the application/execution of the algorithms, and this was solved by applying various encodings on these columns, to see which ones best fit the type of data and categories correctly. This was even more important than transformation, since many of the results obtained before this coding were zero or they had a max. accuraccy of 10 %. Due to this previous work, we hit the final results.

This process is often underestimated, particularly in projects whith large amount of data. However, it is the solution that must be practiced preliminarily, because the current algorithms learn from the information and, therefore, the more processed it is, the better it will understand and learn it.

7.1.2. Algorithm Selection

In this project we have focused on classification algorithms. The results have been obtained by tuning the hyperparameters and testing the different preprocessing for each different hyperparameter. The results have been as expected. If we look at the confusion matrix of the best algorithms, we can conclude that the minimum value to identify if a patient has the disease is 89 %, and sa,e value to correctly identify that they do not have the disease. What we realize is that patients with potential disease usually generate trouble to differentiate between CD (Celiac Disease) and non-CD labels, this may be due to the importance of some columns in the classification algorithms, they are minimum values in around 10 % in cases that do have the disease, and 60 % in those that do not have the disease. This especially occurs in datasets that have separate symptoms or complete dates, that is, dataset II. Although it is one of the best performing datasets, it is giving us problems in case the patient does not have the disease and the diagnose given will be potential CD.

As for the algorithms and combinations with pre-processing, the ones that include Standard Scaler Wrapper use to be high percentage, and the one that gives the best results is

XGBoost Classifier.

7.2. Future Work

The development of this project has allowed us to know in depth the processed information and the usefulness of the parameterization of the algorithms. Therefore, once these main characteristics were done, we expose a list of future developments, to implement new methodologies when training an algorithm to detect celiac disease. It has been divided according to the information collected and the algorithms to be used.

7.2.1. Collected information

- The diagnosis column could be improved in terms of the labels that they contain currently, if a review is carried out with the doctors. The proposal is the following:

    - Asymptomatic or silent CD: those who have the disease but do not have apparent symptoms
    - Classic CD: they are those that present the symptoms of digestive problems.
    - Non-classical CD: does not present the symptoms of a typical picture of celiac disease such as malabsorption.
    - Subclinical CD: with manifestations below a picture of celiac disease.
    - Symptomatic CD: the symptoms are clear.
    - Potential CD: patients with positive serology but normal intestinal biopsy.
    - No EC: he does not have the disease.

  We believe this division fits better a true patient.

- Improvement of the degree of family relationship: within the dataset itself, a column that associates different patients, if they have a positive or doubtful CD could be

added. This would allow symptoms, signs, or risks in various patients to be correlated among relatives

- Continuous analysis of potential patients: an analysis of the patient over a period of time, decided by the doctor, could help us in the evolution of the symptoms, signs and diseases that the patient may develop over time.

- The way of collecting the information: this is important because the information has required a transformation and a deep selection of variables, due to the fact that it was poor in data and in quantity. If the way to obtain the information of the signs and symptoms is easier, it will improve its quality. It is proposed to do this collection through a desktop or tablet application in the case of doctors, and for patients to collect their current status, that is, daily symptoms. A mobile or tablet application is recommended. Physicians would be able to access study patient information continuously and fill in the data they receive from the analyses.

- A field would be added to the dataset in the event that the patient is pregnant or not, since this along with other symptoms are incentives when it comes to developing the illness.

- I would associate images of the colonoscopies to this project, although they are not part of the dataset, new information could be used to analyze if these images determine some type of anomaly.

7.2.2. Algorithms

The use of neural network algorithms (DNN) is recommended, this could allow distributing the weights of the neurons in the most important values, for example in the case of the HLA column, the values of DQ2.5 that have double dose will weigh more than those that have DQ8 or DQ2.2 with a dose. This would improve the labeling of the data.

If images are being used, convolutional neural network o CNN is recomended to indicate if atrophy has happened.

Capítulo 8

Contribuciones individuales al proyecto

8.1. Contribuciones de Jennifer Marmolejos Urbaez

Del proyecto me he encargado de dividir el proyecto en diversas etapas y, asignar las tareas correspondientes. En la etapa inicial, de análisis de la base de datos y de las variables que van a formar parte del proyecto, he analizado:

Helicobacter pylori en el momento de la biopsia, Olmesartán en el momento de la biopsia, Giardia en el momento de la biopsia, Sobrecrecimiento bacteriano en el momento de la biopsia, Fecha DCG Biopsia2, DCG Biopsia-AP2, Comentarios Biopsia-AP2, Más determinaciones DCG, Fecha DSG biopsia1, DSG Biopsia AP1, Fecha DSG biopsia2, DSG Biopsia AP2, Más determinaciones DSG, FECHA LIEs DCG_1, LIEs DCG %GD_1, LIEs DCG %iNK_1, Valoración DCG LIEs1, Día procesamiento DCG1, AP Biopsia DCG LIEs_1, FECHA LIEs DCG_2, LIEs DCG %GD_2, LIEs DCG %iNK_2, Día procesamiento DCG2, Valoración LIEs2, AP en Biopsia DCG LIEs_2, FECHA LIEs DSG_1, LIEs DSG %GD_1, LIEs DSG %iNK_1, AP Biopsia DSG LIEs_1, Comentarios AP en Endoscopia de LIEs, Valoración DSG LIEs1, Día procesamiento DSG1, FECHA LIEs DSG_2, LIEs DSG %GD_2, LIEs DSG %iNK_2, Valoración DSG LIEs2, Día procesamiento DSG2, AP en Biopsia DSG LIEs_2, FECHA LIEs DSG_3, LIEs DSG %GD_3, LIEs DSG %iNK_3, Valoración DSG LIEs3, AP en Biopsia DSG LIEs_3, Respuesta DSG, Respuesta DSG Clínica, Respues-

ta DSG Serológica, Respuesta DSG Histológica, Fecha provocación, Tiempo DSG, Edad en provocación, Comentarios, Clínica 6 días, Flatulencia, Distensión abdominal, Malestar abdominal

En la segunda etapa, transformación y creación de la lógica de la ETL, me he encargado del desarrollo en python del proyecto. Así, tras la selección y análisis, he aplicado la eliminación de aquellas que no eran necesarias, transformar y limpiar las que se han elegido para formar parte del algoritmo. En la tercera etapa, he seleccionado los algoritmos necesarios para aplicar en el proyecto basándome en el resultado obtenido con la herramienta de Azure. En esta etapa se han aplicado los algoritmos y se han generado los gráficos. Por último, he participado en la escritura del documento del TFG en los siguientes capítulos: 2, 4, 5, 6 y 7.

8.2. Contribuciones de Jennifer Zapata Arciénega

La primera parte del proyecto es el preprocesamiento de los datos y para ello, necesitaba informarme sobre el tema que íbamos a trabajar, es decir, entender la base de datos que nos había proporcionado nuestra tutora junto con el proyecto anterior de *Cluestering*.

Nos dividimos las columnas de la base de datos completa para realizar una leyenda donde explicamos cada variable si era útil o no, de las cuales he analizado las siguientes variables: Alteraciones hábito intestinal, Cansancio, Irritabilidad, Vómitos, Clínica día 6, Provocación larga, Duración provocación larga, Provocación 3 días, N CD8 total d0, N CD8 triple positiva d0, N CD8 total d6, N CD8 triple positiva d6, Cociente N CD8 día 6/día 0, % CD8 triple positiva d0, % CD8 triple positiva d6, N GD total d0, N GD triple positiva d0, N GD total d6, N GD triple positiva d6, % GD triple positiva d0, % GD triple positiva d6, CD62L negativos CD8 día 0, CD62L low CD8 día 0, Porcentaje CD62L neg en CD8_0, CD62L negativos GD día 0, CD62L low GD día 0, Porcentaje CD62L neg en GD_0, CD62L negativos CD8 día 6, CD62L low CD8 día 6, Porcentaje CD62L neg en CD8_6, CD62L negativos GD día 6, CD62L low GD día 6, Porcentaje CD62L neg en GD_6, Marcadores citometría, Comentarios, ELISPOT, ELISPOT SFC/106 QS17 día 0, ELISPOT SFC/106 QS17 día 6, ELISPOT SFC/106 SQ14 día 0, ELISPOT SFC/106 SQ14 día 6, ELISPOT SFC/106 QL10

día 0, ELISPOT SFC/106 QL10 día 6, ELISPOT comentarios, Solo sueros, FECHA LIEs, LIEs %GD, LIEs %iNK, Día procesamiento 1, Valoración LIEs, AP Biopsia LIEs, Dieta en determinación de LIEs, Comentarios, Natalia-pendiente, PACIENTE COMPLETO, EC REVISADA Y CONFIRMADA, Comentarios.

Al tener las variables necesarias seleccionadas continuamos con la transformación de estas, con las columnas repartidas cada uno se encargaba de decidir, según la información que nos aportaba la variable, si necesitaba una transformación o creación de una variable nueva.

Tras terminar la etapa de preprosesamiento de datos, junto con mis compañeros nos encargamos de recopilar información sobre los algoritmos que íbamos a utilizar

Finalmente, me he encargado escribir la motivación y abstract, además de corregir la memoria tanto errores gramaticales y ortográficos como imágenes y tablas.

8.3. Contribuciones de Cristian Emanuel Anei

La primera tarea de la que me he encargado, ha sido analizar una sección de la base de datos inicial, y decidir cuales de los datos de los que se me han asignado resultarían finalmente útiles para nuestro proyecto. De esta manera, he decidido que las siguientes columnas no serian tenidas en cuenta para el proyecto. Record Id, Grupo provocación, Edad diagnóstico, Especifique el familiar/es afectado/s, Otros kits, Fecha DCG ATG2_2, DCG ATG2_2, Indique título de anticuerpo (DCG ATG_2_2), Indique el título del anticuerpo (A-PDG_2), Otros Kits, A-PDG kit, A_pdg_kit_otros, Fecha DSG ATG2_2, DSG ATG2_2, Indique el título del anticuerpo (DSG ATG2_2), Fecha DCG Biopsia1, AP al diagnóstico.

A continuación, habría que hacer el análisis del resto de columnas que si serían relevantes para el trabajo. Este análisis consiste en una pequeña definición de las columnas, un análisis de los valores que puede tomar, y finalmente determinar la importancia real para el proyecto. Las siguientes columnas son las que han pasado la selección:

Fecha nacimiento, Diagnóstico, Indique país de origen o en su defecto la información disponible, Sexo, Grupo de riesgo, Grado de parentesco, Grado de parentesco (si hay más

de 1), Enfermedad inmunológica, Enfermedad inmunológica (si hay más de 1), Enfermedad inmunológica (si hay más de 2), Otro/s riesgo/s, Síntomas específicos, Síntomas específicos, Síntomas específicos, Otros síntomas, Signos, Signos 2, Signos 3, HLA: grupos de riesgo, Haplotipo1, Haplotipo2, DCG_ATG2_1, Indicar titulo del anticuerpo (DCG ATG_2_1), Ïndicar el kit empleado con el punto de corte entre paréntesis", DCG EMA, DCG A-PDG_1, Indique el título del anticuerpo (A-PDG_1), Ïndicar el kit empleado con el punto de , corte entre paréntesis", DCG A-PDG_2, DSG ATG2_1, Indique el título del anticuerpo (DSG ATG2_1), Indicar el kit empleado con el punto de corte entre paréntesis, DSG EMA, DSG A-PDG_1, Valor A-PDG_1, DCG Biopsia-AP1.

A continuación hacia falta realizar una definición mas detallada de cada columna, la cual serviría para incluirla en el Capitulo 2. Para esta tarea he realizado la explicación de las columnas mencionadas en el párrafo anterior, pero solo de las comprendidas entre la columna "Diagnosticoz "Signos 3".

Una vez determinados los datos y terminadas las tareas relacionadas a ellos, me he encargado de recopilar información acerca de las herramientas de trabajo que nos podían ser útiles, como se ve reflejado en el capitulo 3.

A continuación, me encargue de realizar un análisis de los algoritmos que ya han sido previamente seleccionados. Dicho análisis ha sido realizado por categorías de algoritmos y consiste en una descripción de cada algoritmo y algún tipo de fórmula matemática que lo defina. Para empezar, se analizaron los algoritmos de escalado de los datos, seguidos de los algoritmos de regresión y los arboles de decisión.

Además, he buscado información de los métodos de evaluación de dichos algoritmos. Dichos métodos han sido previamente seleccionados también, por lo que yo solo he buscado la información necesaria. De nuevo, como en el capitulo anterior, una breve descripción de cada método, y una formula matemática que lo respalde.

Hasta aquí mi contribución al proyecto, aunque cabe mencionar alguna contribución en la redacción final de la memoria, así como la traducción de algunos capítulos al inglés.

Bibliografía

[1] A. Pinzón-Rivadeneira E. Navalón-Ramon, . Juan-García. Prevalencia y características de la enfermedad celíaca en la fachada mediterránea peninsular. 42(8):514–522, 2016.

[2] Aurelien Geron. *Hands-On Machine Learning with Scikit-Learn, Keras Tensor Flow.* 1996.

[3] Eric Brill Michele Banko. Scaling to very very large corpora for natural language disambiguation. pages 26–33, 2001.

[4] Felipe Moscoso and Rodrigo Quera. Enfermedad celíaca: revisión. *Revista Médica Clínica Las Condes*, 26(5):613–627, 2015.

[5] Muhammad Khawar Sana, Zeshan M Hussain, Pir Ahmad Shah, and Muhammad Haisum Maqsood. Artificial intelligence in celiac disease. *Computers in Biology and Medicine*, 125:103996, 2020.