INFERENCIA DE DISTRIBUCIÓN GEOGRÁFICA DE JORNADAS LABORALES EN LA CIUDAD DE MADRID A PARTIR DE DATOS DE MOVILIDAD

INFERENCE OF GEOGRAPHICAL DISTRIBUTION OF WORKING DAYS IN THE CITY OF MADRID FROM MOBILITY DATA

TRABAJO FIN DE MÁSTER EN INTERNET DE LAS COSAS
CURSO 2019-2020

AUTOR
LUIS FERNANDO MARTÍN RUIZ

DIRECTOR
RAFAEL CABALLERO ROLDÁN

MÁSTER EN INTERNET DE LAS COSAS
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

INFERENCIA DE DISTRIBUCIÓN GEOGRÁFICA DE JORNADAS LABORALES EN LA CIUDAD DE MADRID A PARTIR DE DATOS DE MOVILIDAD

INFERENCE OF GEOGRAPHICAL DISTRIBUTION OF WORKING DAYS IN THE CITY OF MADRID FROM MOBILITY DATA

TRABAJO FIN DE MÁSTER EN INTERNET DE LAS COSAS
CURSO 2019-2020

AUTOR
LUIS FERNANDO MARTÍN RUIZ

DIRECTOR
RAFAEL CABALLERO ROLDÁN

CONVOCATORIA: JUNIO 2020
CALIFICACIÓN: 7,5

MÁSTER EN INTERNET DE LAS COSAS
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

AUTORIZACIÓN DE DIFUSIÓN

El abajo firmante, matriculada en el Máster en Internet de las Cosas de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: "INFERENCIA DE DISTRIBUCIÓN GEOGRÁFICA DE JORNADAS LABORALES EN LA CIUDAD DE MADRID A PARTIR DE DATOS DE MOVILIDAD", realizado durante el curso académico 2019-2020 bajo la dirección de Rafael Caballero Roldán en el Departamento de Sistemas Informáticos y Computación, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Luis Fernando Martín Ruíz

2 de julio de 2020

DEDICATORIA

Dedico este trabajo a mis padres y a mis
abuelos, Delia y Gelacio, de quienes
aprendí que el esfuerzo y constancia
siempre dan sus frutos.

AGRADECIMIENTOS

A mi compañero Konstantin, por su gran disposición y ayuda que me ha aportado a lo largo este trabajo. A VirtualDesk, por facilitarme los datos y las herramientas necesarias para poder realizar el proyecto. Y Rafael, sin su ayuda y apoyo, este trabajo no abría sido posible realizarlo con éxito.

RESUMEN

Actualmente, millones de madrileños utilizan el transporte público para ir al trabajo cada día; ya sea metro, Renfe, autobuses de EMT o autobuses interurbanos. Para hacerlo, todos ellos emplean la tarjeta de transporte sin contacto. Una misma tarjeta permite a los usuarios el acceso a todos los medios de transporte público. Esta tarjeta tiene distintos formatos, puede ser mensual (ilimitada durante 30 días) o disponer de un número predeterminado de viajes, bajo recarga, sin límite de tiempo (tarjeta Multi).

Partiendo de los datos generados por estas tarjetas recogidos por algunas estaciones de metro, este trabajo pretende buscar la relación entre la zona geográfica en que se haya la estación y otros parámetros. Por ejemplo, el tiempo transcurrido desde que una persona inicia un viaje por la mañana hasta su trabajo, y el inicio del viaje de vuelta hasta su casa; o si el usuario trabaja durante fin de semana o no. De la misma forma, esta investigación también pretende determinar si estos parámetros están vinculados a la renta per cápita media de la zona en la que el usuario vive.

A lo largo del trabajo de investigación se explicará el procedimiento seguido para obtener la información, a partir de una compleja base de datos Big Data. De igual forma, se presentarán los métodos de aprendizaje automático que han sido necesarios para poder comprobar la existencia de estas relaciones.

Palabras clave

Movilidad, transporte, metro de Madrid, CRTM, renta per cápita, jornada laboral

ABSTRACT

Nowadays, millions of citizens from Madrid use public transport to go to work every day; whether by metro, Renfe, EMT buses or intercity buses. To that end, they use the contactless transport card. The same card allows users access to all types of public transport. This card has different formats, it can be monthly (unlimited for 30 days) or have a predetermined number of trips, under recharge, without time limit (Multi).

Based on the data generated by these cards collected by some metro stations, this work lookup the relationship between the geographical area where is placed the station and other parameters. For example, the time elapsed from the moment a user starts a journey in the morning to work, and the start of his journey back home; or if the user works on weekend or not. In the same way, determining if these parameters are linked to the average per capita income of the area where the user lives.

Throughout this doc, the procedure followed to obtain the information will be explained, based on a complex Big Data will database. As well as, the machine learning methods that have been necessary to verify the existence of these relationships will be presented.

Keywords

Mobility, transport, Madrid metro, CRTM, per capita income, workday.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Capítulo 1 - Introducción

Desde el año 2017, la comunidad de Madrid ha dejado de emplear la tecnología magnética que ha pasado a ser totalmente sustituida por la tecnología sin contacto para gestionar todos los tipos de títulos que se utilizan en el transporte público. A fecha de del 31 de diciembre del 2017, el número de tarjetas de título personal eran de 2.602.223 [1].

La cantidad de información que se genera en torno a las validaciones de estás tarjetas, ya sea el metro, EMT, Renfe o autobuses interurbanos, es muy valiosa porque permite hacer multitud de análisis sobre estos datos. En este proyecto proponemos utilizar esta información para conocer detalles sobre los hábitos laborales en distintas zonas de Madrid capital y buscar relaciones, si las hay, con la renta per cápita de los distintas distritos o zonas de Madrid.

1.1 Motivación

A lo largo del desarrollo del **Máster en Interne de las Cosas (IoT)**, hemos aprendido las distintas piezas que existen dentro de un ecosistema IoT. Las etapas típicas para el procesamiento de datos en un entorno IoT, aunque pueden existir diferentes soluciones, son las siguientes:

1. **Captación de los datos**: en la asignatura de ***Arquitectura del nodo IoT***, descubrimos los distintos tipos de tecnologías y sensores que existen para poder recolectar los datos de distinta tipología: sensores de temperatura, humedad, etc.
2. **Comunicación entre los nodos:** en la asignatura de ***Redes, Protocolos e Interfaces I y II***, aprendimos cómo se comunican y qué protocolos usan los nodos entre sí de una solución IoT. Además, pudimos ver cómo se comunica un nodo con Internet (Cloud).
3. **Almacenamiento y tratamiento de datos:** los sensores pueden generar gigabytes o terabytes de información, con la asignatura de ***Tratamiento de***

datos masivos, aprendimos cómo y dónde almacenar los datos captados además del tratamiento que se deben realizar sobre los mismos.

4. **Modelos analíticos:** una vez están almacenados estos datos, con la asignatura de ***Inteligencia artificial aplicada a IoT***, aprendimos a cómo aplicar modelos analíticos para extraer información valiosa a partir de los datos recogidos por el sensor.
5. **Seguridad y legalidad:** en esta ultima asignatura aprendimos los aspectos legales y el cuidado de debemos tener sobre los datos que son recogidos y almacenados, además técnicas para desarrollar software seguro e intentar minimizar las posibles vulnerabilidades que puede tener el software desarrollado.

El motivo principal para realizar este trabajo es que en la empresa actual en la que trabajo, de nombre **VirtualDesk**, está desarrollando un proyecto llamado **Mobiam** [2] en conjunto **System** y la **Universidad Politécnica de Madrid**, que usa los datos de movilidad correspondientes al año 2019 de transporte público de la Comunidad de Madrid, a los cuáles tenía acceso y podía usarlos para realizar este trabajo de fin de máster. Teniendo en cuenta los puntos mencionados anteriormente, este trabajo se centra en **la parte de almacenamiento y tratamientos de los datos**, además de realizar algún **modelo analítico**. El Consorcio de Transportes es quién se ha encargado de gestionar y construir la infraestructura que permite captar los datos de las validaciones recogidas de las tarjetas de transporte sin contacto.

Otro motivo que me llevó a realizar este estudio es que la empresa disponía de un entorno BigData basado en Cloudera, el cual me permitió manejar la gran cantidad de datos de la que disponía para poder filtrarla y realizar el estudio sobre una cantidad mucho más reducida. Por otro lado, Rafael, mi director del trabajo, me propuso buscar la posible relación entre los hábitos laborales de los usuarios del transporte y la renta per cápita media del distrito en el que se encuentra la estación de partida.

1.2 Objetivos

Como hemos mencionado, la principal finalidad del trabajo es intentar buscar si existe relaciones entre las jornadas laborales de una persona o tiempo que pasa fuera de su hogar y la renta media per cápita de zona en la que reside. En concreto:

- ¿Qué día de la semana se trabaja más?
- ¿Cuántas horas está una persona fuera de su hogar de media?
- ¿Existe alguna relación entre vivir en una zona, la renta per cápita de la zona y el hecho de trabajar en fin de semana? ¿y en día festivo? ¿se trabajan más festivos en las zonas de renta más baja?
- ¿Cuál es la distancia media desde un punto inicio del viaje(casa) hasta el destino final (trabajo)?

1.3 Plan y estructura del trabajo

Para intentar responder a alas preguntas de la sección anterior, es necesario procesar y generar los datos en el formato que mejor se adecue al problema. Lo primero que hay que tener en cuenta es **que el volumen de datos es muy grande** ya que las tarjetas de transporte se pueden validar en distintos medios de transporte público: Metro, EMT, Renfe y autobuses Interurbanos. Como el objetivo final de trabajo es ver las relaciones entre los hábitos laborales y la renta per cápita de los distritos de Madrid, decidí centrarme en aquellas tarjetas de transporte que realizan alguna validación por la mañana en ciertas estaciones de Metro para posteriormente, exportar todas las validaciones realizadas en el período del mes de enero al mes junio del año 2019. Los pasos seguidos para realizar el trabajo final fueron:

- ***Definir estaciones de origen**:* determinamos las estaciones finales sobre las que centraríamos el estudio, eligiendo solo paradas de metro que estén en la mayoría de los distintos distritos de Madrid: *Puerta del su, Plaza Elíptica, Puente de Vallecas, Nueva Numancia, Atocha Renfe, Nuevos Ministerios, Plaza Castilla, Príncipe Pío, Moncloa, Ventas, Pueblo Nuevo, Canillejas, Chamartín,*

Cuatro Caminos, Avenida América, Sainz de Baranda, Alonso Martínez y Méndez Álvaro.

- ***Seleccionar aquellas tarjetas y viajes*** que pudieran corresponder a personas que acuden a su trabajo en transporte público de forma habitual**.** Para ello buscamos **tarjetas de transporte que validan** en la misma estación (de las estaciones definidas) **más de 90 veces** a lo largo de los meses de enero a junio **entre 6 y10 a.m**. Esto se hace pensado en las horas más comunes de entrada en los trabajos, ya que nos interesa la jornada laboral de estos usuarios. Es decir, asumimos que una persona que ha salido al menos 90 días (entre enero y junio) en la mañana dentro de la franja horaria de 6-10 a.m. está acudiendo a su trabajo. Por otro lado, también asumimos que una persona vuelve desde el trabajo por la tarde entre 15h y las 20 h, ya que es la hora de salía más común en los trabajos. Es decir, al final tendremos 90 viajes realizados por la mañana entre las 6-10 y viajes realizados en el mismo día por la tarde entre las 15h-20h. Somos conscientes de que esto no es necesariamente así; por un lado puede que alguna de estas personas no vaya al trabajo, y por otra parte "perdemos" usuarios que sí trabajan pero con un horario distinto, pero consideramos que es una aproximación que nos puede permitir inferir cómo son las jornadas laborales de los usuarios. Además, por esta misma razón, hay que tener en cuenta que no se va a procesar cualquier tipo de tarjeta, solo se analizarán aquellas que sean de tipo de **título personal o abono joven**.
- ***Exportar los datos****:* solo de aquellas tarjetas que cumplen las condiciones anteriores.
- ***Generar tablas finales****:* generar tablas con los datos necesarios para poder explicar los objetivos. Para ello se procesarán los datos exportados en Python, para crear las tablas que contendrán los datos necesarios que permitan contestar a los objetivos.
- ***Representación y conclusiones:*** por último, con las tablas finales creadas a partir de los datos de movilidad, se representaron los resultados finales mediante gráficas realizadas en Python y se respondieron a las cuestiones planteadas en este estudio.

1.4 Tecnologías

La cantidad de información que se genera entorno a las validaciones de estas tarjetas (en el metro, EMT, autobuses interurbanos) en un solo año es inmensa, decenas de gigabytes de información, la cuál no es posible tratar en un ordenador personal al estar limitado en recursos tales como la cantidad de memoria RAM. Por ello, en este tipo de problemáticas se utiliza plataformas **Big Data [3],** este término se define como la situación que se da en el momento que el conjunto de datos ha crecido de tal manera que es difícil de manejar y aún más difícil analizar para obtener un valor sobre ellos. Dentro de este marco de Big Data existe la distribución **Cloudera [4]** que es un sistema integrado basado en **Apache Hadoop [5],** que ofrece una serie de herramientas que facilitan el procesamiento y tratamiento de los datos. En este proyecto, se ha usado una herramienta almacenamiento llamada **Hive**, que permite ejecutar una serie de consultas en formato SQL para reducir la información y obtener los datos finales sobre los cuales se han realizado el análisis.

Como la cantidad de datos exportados de la plataforma Cloudera era considerablemente menor, para la exploración y análisis de los datos se utilizo **Python** [6] como lenguaje de programación y librerías como **Pandas**, **Numpy y Sklearn** para procesar y analizar los datos.

En el siguiente capítulo se explicará cómo se generan las tablas con los datos necesarios para realizar el estudio. En el capítulo 3 presentaremos los análisis de estos datos que nos permitirán contestar a los objetivos. Finalmente, el capítulo 4 presenta las conclusiones y trabajo futuro.

Capítulo 2 - Extracción y preprocesamiento de los datos

A lo largo de este capítulo, se explicará en profundidad cada uno de los procesos seguidos para obtener los datos finales permiten responder a las cuestiones planteadas en los objetivos del trabajo.

2.1 Origen y descripción del dataset

Los datos originales que se han usado para realizar el estudio provienen, del Consorcio Regional de Transportes de Madrid (CRTM).

Dada la dimensionalidad de los datos proporcionados por CRTM, estos estaban almacenados en **Cloudera**, pero ¿qué es Cloudera? Es una empresa que proporciona software basado en **Apache Hadoop**, que es un framework que permite procesar y almacenar datos de manera distribuida.

2.1.1 Cloudera: Ecosistema de Apache Hadoop

Inicialmente Hadoop se compone de dos partes: una destina al almacenamiento de los datos y otra dedicada al procesamiento de estos.

En cuanto a la primera parte, **HDFS** [7] (Hadoop Distributed File System) es el sistema de ficheros distribuidos altamente tolerante a fallos que usa Hadoop para almacenar los datos en un clúster de máquinas. Un clúster basado en HDFS, consta de dos tipos de máquinas. El **NameNode** que actúa como maestro y almacena la metainforamación necesaria para saber en qué máquina o máquinas están almacenados los datos que componen a un determinado fichero. Por otra parte, se encuentra el **DataNode** encargado de almacenar dichos datos. Como HDFS almacena la información de forma distribuida, por defecto, cada DataNode almacena 128MB de bloques de información de un archivo o fichero, es decir, si se almacena un fichero mayor a 128MB, el fichero se dividirá en trozos más pequeños y cada uno de trozos se

almacenará en un DataNode distinto, replicando la información(en caso de que se haya configurado) en otro nodo por si alguno de ellos deja de funcionar y no perder la disponibilidad del dato. En la figura 1 (de elaboración externa [8]) que se muestra a continuación se expone el proceso seguido para almacenar un fichero en los **DataNode**:

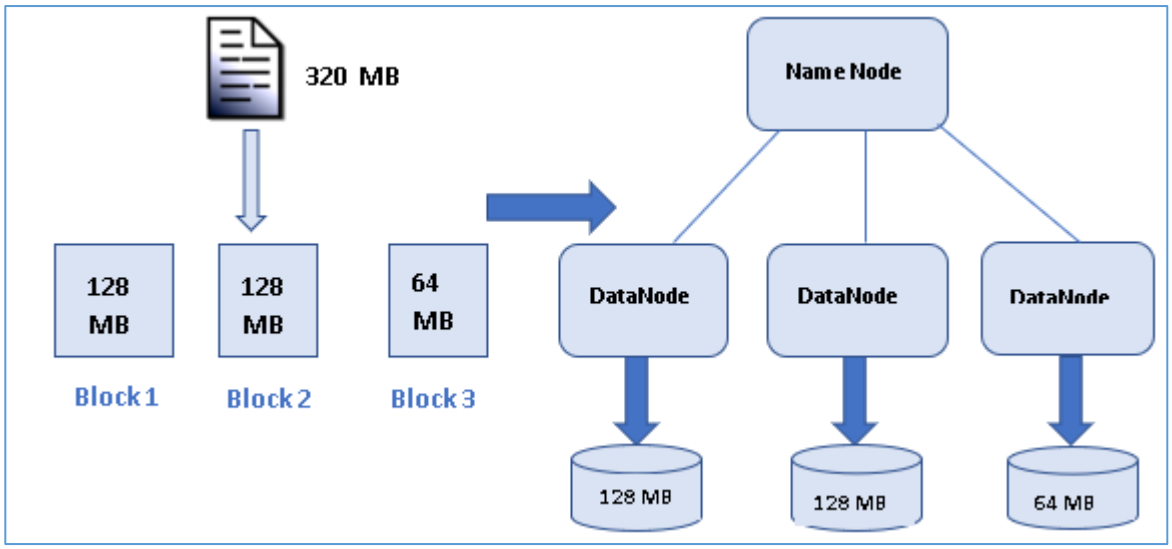

Figura 1 – Ejemplo de almacenamiento de un fichero en HDFS

Por otro lado, Hadoop usa el algoritmo **MapReduce** [9] como técnica de procesamiento para los datos. Al final del procesamiento el resultado es almacenado de nuevo en HDFS. El algoritmo MapReduce consta de dos fases: la primera fase de mapeo (**Map**), la cual trabaja con los datos sin procesar y produce valores intermedios que pasan a la fase de reducción (**Reduce**) para producir la salida final. El ejemplo más sencillo para entender este algoritmo es el conteo de palabras sobre un fichero: cada DataNode, ejecuta la fase Map en la que se cuenta el número de veces que aparece una palabra en cada trozo del fichero que tiene almacenado. Es decir, se generará una tupla clave-valor (palabra, nº de apariciones de la palabra) que pasará a la fase Reduce. En la fase de Reduce, se agruparán aquellas tuplas que tengan las mismas claves y se reducirá (en este caso se sumará) el número de veces que aparece cada palabra que será el resultado final del número de veces que aparece una determinada palabra en el global del fichero.

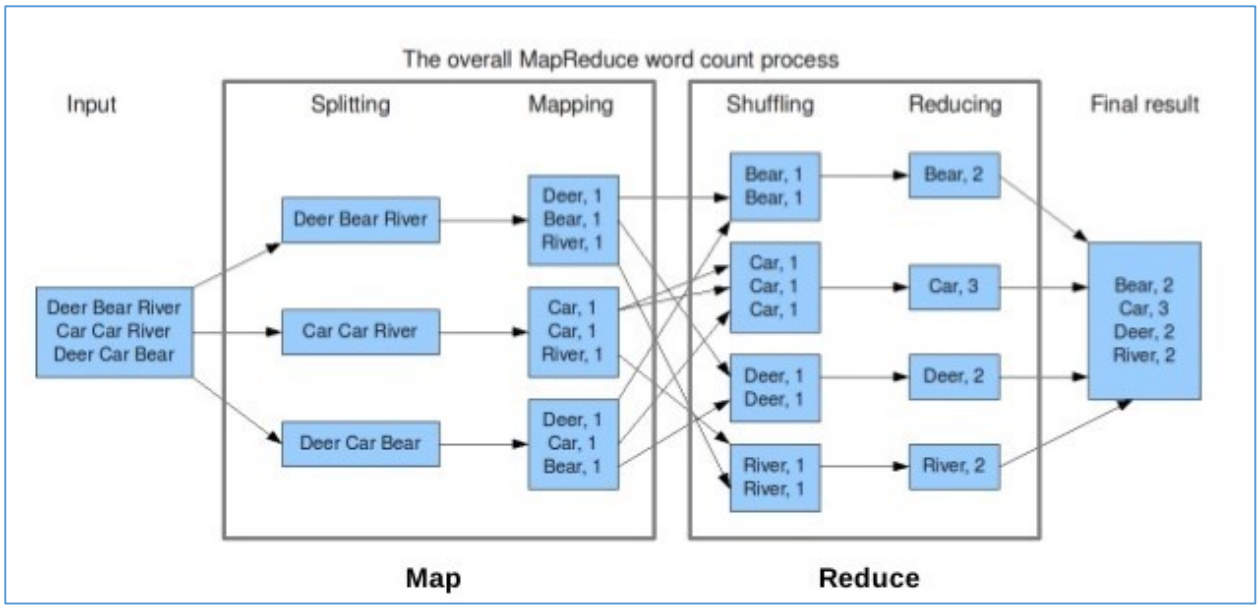

Figura 2 - Ejemplo de MapReduce (de elaboración externa [10])

Con el paso de los años, Apache Hadoop evolucionó y surgieron una serie de tecnologías, cada una de ellas cumpliendo una tarea específica, como puede ser la ingesta de datos en HDFS o el procesamiento de los datos usando por detrás el paradigma MapReduce. Una de las distribuciones más extendidas es **Cloudera**, la cual se fusionó con Hortonworks en el año 2018 y se ha convertido en líder del mercado, ya que permite realizar despliegues de *Data Lakes* de manera simple y ofrece una variedad de tecnologías o herramientas para poder procesar los datos. En torno al **Ecosistema Apache Hadoop [11]** existen una variedad proyectos open source. Algunos de los más conocidos son:

- **Ambari**: permite gestionar, monitorear y provisionar un clúster de Apache Hadoop.
- **HBase**: base de datos no relacional que funciona sobre HDFS.
- **Pig**: permite crear programas que por debajo usan MapReduce, ya sea en Java, Python o Javascript de forma simple y con pocas líneas de código.
- **Hive**: permite realizar consultas muy similares a SQL sobre datos almacenados en HDFS mediante lenguaje HQL (Hive Query Language). Por debajo también crea, crea flujos MapReduce.

- **Sqoop**: permite la ingesta de datos desde base de datos relacionales (Oracle, MySql) a HDFS o tablas de Hive.
- **Flume**: destinado a la ingesta de datos semiestructurados o no estructurados desde diferentes fuentes hacia HDFS.

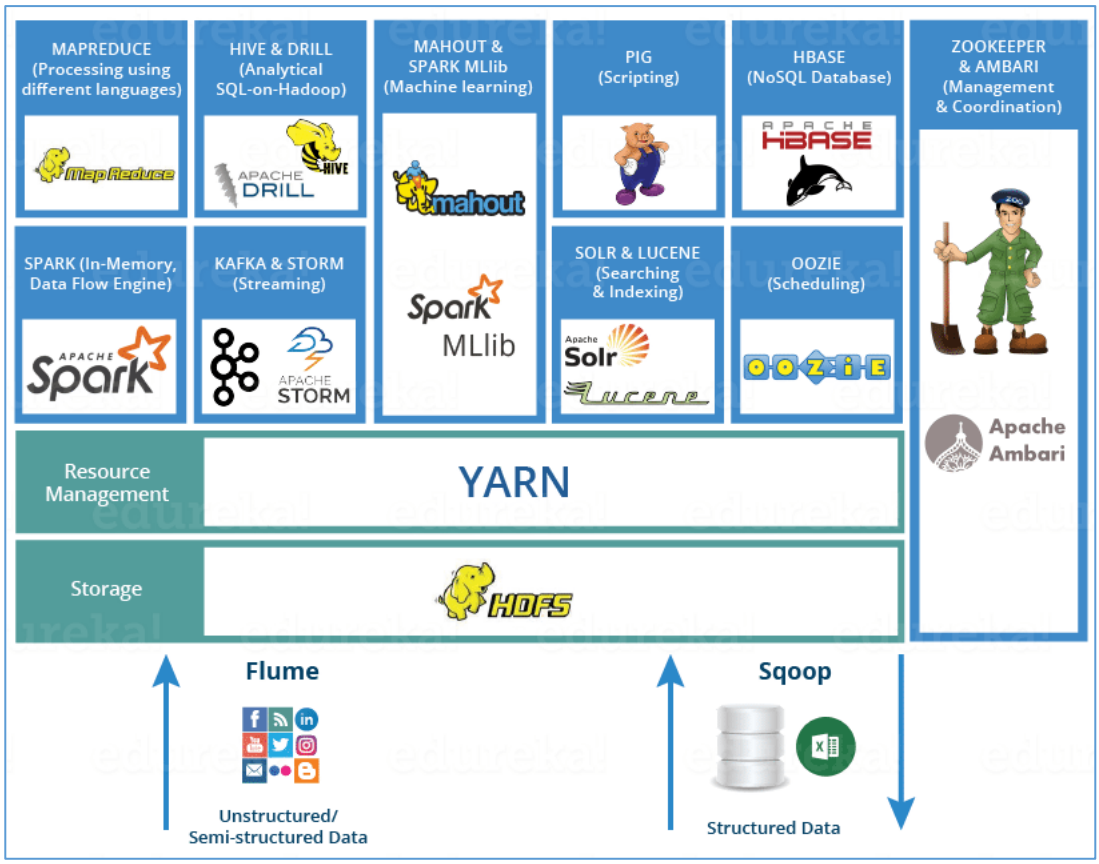

Figura 3 - Ecosistema Hadoop (de elaboración externa [12])

2.1.2 Datos del proyecto

Los datos de nuestro originalmente fueron originalmente engestados en HDFS mediante **Sqoop** y procesados mediante **Pig** para finalmente ser almacenarlos en tablas de **Hive**. En el entorno de Cloudera, existía una tabla en Hive con nombre ***tpub_etapas_all*** que contenía los viajes realizados por cada tarjeta durante el año 2019. Un viaje puede estar formado de varias etapas o partes, por ejemplo: un viaje se inicia en metro y luego se realiza un transbordo a un bus hasta llegar al destino. La tabla está constituida de los siguientes campos:

COLUMNA	DESCRIPCIÓN
ANIOSMES	año y mes en el que se realizó el viaje
IDSERIETARJETA	identificador de la tarjeta (anonimizada)
VIAJE	fecha y hora en la que se ha realizado el viaje
PARTE_VIAJE	parte del viaje: un viaje puede iniciarse en metro y continuar en bus
FXTRANSACCIONTERMINAL_LAG	fecha y hora de inicio de parte del viaje
ACTOR_DPP_LAG	empresa (metro, EMT, Renfe o Interurbanos) que gestiona la validación de inicio del viaje.
LINEA_RED_LAG	línea dentro de la empresa en la que se realizó el inicio de parte del viaje
PARADA_RED_LAG	parada de inicio de parte del viaje
ZONE_ETAPA_START_LAG	zona de transporte de inicio de parte del viaje
GEOCODIGO_ETAPA_START_LAG	código de la zona de la parada de inicio del viaje
MUNICIPIO_START_LAG	municipio de inicio donde se realizó el viaje
PARADA_RED_FINISH	parada de fin de parte del viaje
ZONE_ETAPA_FINISH	zona de transporte de fin de parte del viaje
GEOCODIGO_ETAPA_FINISH	código de la zona de la parada de fin del viaje
MUNICIPIO_FINISH	municipio de fin donde se realizó el viaje
DISTANCE	distancia acumulada recorrida hasta la parte actual del viaje
FXTRANSACCIONTERMINAL	fecha y hora de fin de parte del viaje
PASAJEROS_VIAJE_COMPLETO	número de viajeros
POINT_START_LAG_LAT	latitud de inicio de parte del viaje
POINT_START_LAG_LON	longitud de inicio de parte del viaje
POINT_FINISH_LAT	latitud de fin de parte del viaje
POINT_FINISH_LON	longitud de fin de parte del viaje

Tabla 1 - Campos de la tabla tpub_etapas_all

Por otro lado, existen otra serie de tablas que ayudaron a poder realizar filtros sobre los viajes realizados por las tarjetas transporte:

- *bit_ctipostitulo*: informa sobre los distintos tipos de títulos que puede tener una tarjeta de transporte. Hay 162 tipos distintos de títulos, pero solo nos centraremos en 4 de ellos:
  - 4097 – 30 DIAS ZONA A
  - 4107 – ANUAL ZONA A
  - 4181 – ABONO 30 DIAS JOVEN T. PLANA
  - 4182 – ABONO ANUAL JOVEN T. PLANA

COLUMNA	DESCRIPCIÓN
IDTIPOTITULO	Identificador del tipo de titulo
TXNOMBRETITULO	Nombre del título

Tabla 2 - bit_ctipostitulo

- *t_operadores*: contiene los distintos tipos de actores que usan la tarjeta de transporte. En este caso existen 43 valores distintos, por ejemplo, EMT o Renfe, pero en este caso nos interesa solo el operador con código *02* que es correspondiente a Metro.

COLUMNA	DESCRIPCIÓN
IDACTOR	Código del operador
TXNEMONICO	Nombre del tipo de operador
DEEMPRE	Nombre de la empresa

Tabla 3 - t_operadores

Por último, falta mencionar los datos correspondientes a la renta per cápita por cada distrito de Madrid. Para ello busqué los datos de los distritos de Madrid en el **Portal de Datos Abiertos** [13] proporcionado por el Ayuntamiento de Madrid y descargué el fichero en formato *zip* correspondiente llamado ***Distritos en formato geográfico***, ya que más adelante se presentarán los distritos en un mapa y es necesario tener las referencias geográficas. Al descargar la información, se descargó en formato **ETRS89**[14], que es un

sistema de referencia geodésico utilizado en Europa, pero para poder utilizarlo más adelante con las validaciones de metro, fue necesario transfórmalo a un formato GeoJson en formato **EPSG-4258** [15] (sistema de coordenadas geográfico). El fichero final contiene 21 filas (distritos) y las cinco siguientes columnas:

COLUMNA	DESCRIPCIÓN
NOMBRE	Nombre del distrito
CODDISTRIT	Código del distrito
SHAPE_AREA	Área del distrito
SHAPE_LEN	Longitud del distrito
GEOMETRY	Puntos geográficos del área

Tabla 4 – distritos.geojson

Aunque ya se tenía los datos correspondientes a las áreas geográficas de los distritos de Madrid, faltaban por descargar los datos correspondientes a la renta per cápita en cada distrito. Para poder obtenerlos, fue necesario consultar los datos en el **Instituto Nacional de Estadística (INE)** [16]. Se obtuvo un fichero CSV que contenía 21 filas (distritos) y las cuatro siguientes columnas:

COLUMNA	DESCRIPCIÓN
UNIDADES TERRITORIALES	Código de la unidad poblacional (distrito)
INDICADORES DE RENTA MEDIA	Indicador de renta media por persona
PERIODO	Año
TOTAL	Valor de la renta

Tabla 5 – renta.csv

Finalmente, se unificó ambos dataset correspondientes a la renta y los distritos en un solo fichero llamado ***distritos.geojson***. El fichero final que recoge los valores de la renta y las distribuciones geográficas por cada distrito contiene el siguiente formato:

COLUMNA	DESCRIPCIÓN
NOMBRE	Nombre del distrito
CODDISTRIT	Código del distrito
SHAPE_AREA	Área del distrito
SHAPE_LEN	Longitud del distrito
GEOMETRY	Puntos geográficos del área
RENTA	Valor de la renta

Tabla 6 – distritos_renta.geojson

2.2 Metodología para la selección de los datos

El primer paso para poder realizar el estudio fue realizar una selección de las tarjetas de transporte existentes en todo el dataset inicial. Como el objetivo final del estudio es poder ver las relaciones entre los distintos distritos de Madrid, es necesario explicar todo el proceso que se ha seguido para:

1. Exportar un dataset inicial en uno más pequeño para poder manipularlo en un ordenador personal.
2. Realizar una serie de transformaciones sobre los datos exportados.
3. Construir las tablas que contengan la información necesaria para poder responder a todas las cuestiones planteados como objetivos del estudio.

2.2.1 Selección de tarjetas de transportes

Inicialmente, disponemos de todos los viajes realizados en los distintos operadores existentes (EMT, Renfe, Metro, etc.) pero en este estudio, solo nos centraremos en tarjetas que inicien los viajes en ciertas estaciones de metro, dada la dimensionalidad del

dataset original. Para elegir las estaciones, la mejor decisión era intentar seleccionar dichas estaciones estuviesen distribuidas por la mayoría de los distritos de Madrid. Por esa razón, las estaciones seleccionadas finalmente, son las mostradas a continuación. Cabe señalar que una estación contiene un código o códigos (una estación como Nuevos Ministerios pueden tener varias entradas muy dispersas las cuales tiene códigos distintos):

- **Puerta del sur***:* códigos *205 y 209*
- **Plaza Elíptica***:* códigos *108 y 206.*
- **Puente de Vallecas***:* código *19.*
- **Nueva Numancia***:* código *20.*
- **Atocha Renfe***:* código *16.*
- **Nuevos Ministerios***:* códigos *120, 155 y 193.*
- **Plaza Castilla***:* códigos *164 y 190.*
- **Príncipe Pío***:* códigos *127, 198 y 238.*
- **Moncloa***:* códigos *53 y 125.*
- **Ventas***:* códigos *28 y 81.*
- **Pueblo Nuevo***:* códigos *78 y 137.*
- **Canillejas***:* código *74.*
- **Chamartín***:* códigos *189 y 261.*
- **Cuatro Caminos***:* códigos *6, 42 y 121.*
- **Avenida América***:* códigos *64, 118, 141 y 170.*
- **Sainz de Baranda***:* códigos *114 y 174.*
- **Alonso Martínez***:* códigos *57, 85 y 195.*
- **Méndez Álvaro***:* código *111.*

Una vez definidas las estaciones, queda pendiente definir en qué horario deben realizarse los viajes de ida al trabajo: solo se tendrán en cuenta aquellos viajes que se realizan por la mañana entre las 6 y las 10 a.m. en una misma estación. Pero, ¿cuántas veces una persona(tarjeta) debe realizar viajes en ese horario? Si se tiene en cuenta que un mes tiene 4 semanas, de las cuales podemos asumir que una persona puede trabajar 5 días en una semana. Por otra parte, como el análisis se realizó sobre 6 meses del año

2019, el número de total de días que puede trabajar una persona son: 4 (semanas) * 6 (meses) * 5 (días), es decir, 120 días. Además, hay que tener en cuenta que:

- Los días 1 y 7 de enero, 18 y 19 de abril, 1 y 2 de mayo son festivos.
- Una persona puede disfrutar de 22 o más días de vacaciones.
- Las personas pueden realizar teletrabajo uno o más días a la semana.
- En algún día de la semana las personas pueden ir al trabajo en transporte privado.

Teniendo en cuenta todo lo anterior, se tomó la decisión de definir que el número mínimo viajes que una persona (tarjeta de transporte) debe realizar en una misma estación entre las 6 y las 10 a.m. durante los seis meses debe ser mayor o igual a 90 veces.

Además de las estaciones, el horario y número de viajes que ha de realizar una tarjeta de transporte, hay que filtrar el tipo, el cuál debe ser solo el de **metro** (código **02**). Finalmente hay que transcribir todas las anteriores restricciones a una sentencia SQL que se ejecutará en Hive. Las **consultas Hive** suelen tardar varios minutos, ya que como se explicó en la sección del ecosistema de Apache Hadoop, las consultas que se realizan en Hive crean operaciones MapReduce y el resultado de cada una de esas operaciones se almacenan los resultados en HDFS, lo cual implica escribir en el disco de cada uno de los nodos y, en consecuencia, mayor tiempo de ejecución. Otro aspecto para tener en cuenta es la cantidad de datos que se están procesando. Por ello, inicialmente se realizaron las consultas sobre los datos del mes de enero a modo de prueba, y, una vez validados los resultados, se extendieron las consultas al periodo enero-junio del 2019 que es el periodo de estudio que se considera en este trabajo. El resultado de la consulta devuelve todas las tarjetas de transporte distintas que cumplen las anteriores condiciones y los identificadores de dichas tarjetas se guardan en una tabla llamada ***mobiam.tarjetaestudio***. A continuación, se enseña la consulta SQL final que se ejecutó para poder obtener las tarjetas que: inician un viaje en una misma estación (de las definidas) entre las 6 – 10h más de 90 veces y que además el tipo de título sea un abono normal o joven.

```
CREATE TABLE mobiam.TARJETAESTUDIO AS
SELECT DISTINCT t1.idserietarjeta
from
(select idserietarjeta
        from (
        select idserietarjeta, parada_red_lag, count(*) as numviajes
        from (select *, HOUR (viaje) as horaviaje from mobiam.tpub_etapas_all where aniomes in (201901, 201902, 201903, 201904, 201905, 201906)  and actor_dpp_lag = "02" and linea_red_lag= 0 and parada_red_lag in
        (1,6,16,19,20,28,42,53,57,64,74,78,81,85,108,111,114,118,120,120,121,125,127,137,141,155,155,164,170,174,189,190,193,193,195,198,205,206,209,238,261,263,266,273)) as tmp
        where horaviaje >= 6 and horaviaje < 10
        group by idserietarjeta, parada_red_lag
        ) as tmp2
        where
        numviajes >= 90
) as t1
INNER JOIN (
select idserietarjeta
from crtm.validaciones
where tipotitulo in (4107,4097,4181,4182)
) as t2ON t1.idserietarjeta = t2.idserietarjeta;
```

2.2.2 Exportación de los datos

El número total de tarjetas que cumplen los requisitos son 17743, pero los datos que necesitamos no son solo las tarjetas de transporte que cumplen esas condiciones, sino los datos que se encuentran en la tabla ***mobiam.tpub_etapas_all*** que contiene todos los viajes que realizados, tanto por la mañana como por la tarde, por dichas

tarjetas de transporte en el año 2019. Para poder exportar los datos, es necesario acceder a uno de los nodos del clúster y, vía línea de comandos llamar a la herramienta de **Hive** y especificar que el resultado de una consulta SQL debe guardarse en un fichero CSV. A continuación, se muestra el comando utilizado para realizar la llamada a la consulta que debe ejecutarse en Hive, en la cual simplemente se realiza un operación **inner join** con el campo **idserietarjeta** entre las tablas ***mobiam.tarjetaestudio*** y ***mobiam.tpub_etapas_all*** y adicionalmente filtrar que el año y mes en la que se realizó el viaje debe ser entre el mes de enero y junio incluido.

```
hive -e \
"set hive.resultset.use.unique.column.names=false; set hive.cli.print.header=true; select tpub_etapas_all.* from mobiam.tarjetaestudio as ts inner join mobiam.tpub_etapas_all on ts.idserietarjeta = tpub_etapas_all.idserietarjeta where aniomes in (201901, 201902, 201903, 201904, 201905, 201906) " | sed 's/[\t]/,/g'  > etapas.csv
```

El **dataset** exportado tiene un **tamaño** de aproximadamente **1.5Gb**, concretamente incluía **7.265.708 registros** o etapas exportadas. En los siguientes apartados exploraremos los datos en mayor profundidad para obtener las tablas que finalmente ayudarán a resolver los objetivos del estudio.

2.2.3 Análisis exploratorio en Python: viajes de mañana y tarde

Los datos exportados contienen las mismas columnas que la tabla ***tpub_etapas_all,*** pero antes de explicar el proceso seguido es necesario detallar el significado de una etapa. Un viaje puede contener varias etapas. Por ejemplo: el viaje

```
etapas_raw[etapas_raw['idserietarjeta'] == 'FFC910EDBBB920DFB844D61CD24E1AD0']
```

	idserietarjeta	viaje	parte_viaje	fxtransaccionterminal_lag	actor_dpp_lag	linea_red_lag	parada_red_lag
1026851	FFC910EDBBB920DFB844D61CD24E1AD0	2019-05-01 10:19:32	1	2019-05-01 10:19:32	02	0	64
1026852	FFC910EDBBB920DFB844D61CD24E1AD0	2019-05-01 10:19:32	2	2019-05-01 10:38:34	04	0	108

Figura 4 - Ejemplo de un viaje (elaboración propia)

de una persona puede iniciarse en una parada de metro y posteriormente puede realizar un transbordo para cambiarse a Renfe o a un autobús de la EMT para poder llegar a su destino. Veamos un ejemplo:

Como se puede observar en la imagen de la Figura 4, el viaje que se inicio el día 01-05-2019 a las 10h:19min:32seg. Dicho viaje consta de dos etapas:

1º. Etapa: se inicio en la parada de metro de Avenida de América (campo *parada_red_lag* con código 64) a las 10h:19min:32seg.

2º. Etapa: se inicio 10h:38min:34seg en una parada de Renfe (se sabe que es de Renfe porque el tipo de operador, *actor_dpp_lag* es 04 e indica que la para pertenece a Renfe, el código 02 indica que la parada pertenece a Metro). El inicio de esta etapa sería el final de la primera etapa.

Para realizar el análisis final, solo nos interesan conocer la parada en la que se inicia un viaje y la parada en la que se finaliza el viaje, con el fin de determinar la zona a la que va a trabajar. Como el dataset exportado contiene todos los viajes realizados por las tarjetas seleccionadas independientemente de la hora, hay que volver a aplicar los filtros realizados en la exportación realizada en las tablas de Hive. Antes de explicar los criterios que se han aplicado para poder verificar que tarjetas de transporte que cumplen los requisitos planteados inicialmente, hay que señalar que para procesar el dataset exportado de Hive de la tabla **tpub_*etapas_all,*** se utilizó Python, en concreto, se cargó el dataset en un Dataframe(tabla) en el cual, además, se añadieron seis nuevas columnas que más adelante ayudarán a obtener las tablas finales que se usarán para extraer los resultados del análisis. Las seis columnas nuevas creadas son:

- **hora_viaje**: hora en la que se realizó el viaje.
- **weekdaynum_viaje**: día de la semana en formato numérico en la que se realizó el viaje: 0 – Lunes, 1 – Martes, 2 – Miércoles, 3 – Jueves, 4 – Viernes, 5 – Sábado, 6 – Domingo.
- **weekday_viaje:** día de la semana en formato texto: L, M, X, J, V, S y D.
- **month_viaje**: mes en el que se realizó el viaje.
- **monthday_viaje**: día del mes en la que se realizó el viaje.

- **date_viaje**: día en el que se realizó el viaje sin tener en cuenta la hora, minutos ni segundos.

¿Qué tarjetas y viajes analizaremos? Aquellas que quedan después de aplicar los siguientes filtros:

1º. Hay que filtrar aquellas etapas en las que la hora de inicio del viaje, (el campo generado nuevo llamado **hora_viaje**) esté comprendido entre las 6 y las 10 horas la mañana.

2º. Agrupar las etapas por los campos **idserietarjeta** y **viaje** para realizar una agregación y obtener solo la primera y última etapa de un viaje. En caso de que un viaje solo contenga una etapa, la primera y última etapa será la misma. De esta forma, si un viaje contiene dos o más etapas o filas, obtendremos un único registro que contiene la etapa de inicio y de fin del viaje.

3º. También hay que filtrar que estos viajes solo se inicien en las paradas de metro seleccionadas inicialmente.

4º. Por otro lado, aplicados todos los filtros anteriores, hay que agrupar los viajes realizados por tarjeta y parada de inicio, es decir, los campos **idserietarjeta** y **parada_red_lag_first** y quedarnos con aquellas tarjetas que han iniciado más de 90 veces un viaje desde una misma estación de metro de las definidas inicialmente.

5º. Finalmente nos quedamos con los viajes realizados por la mañana (que contiene la etapa de inicio y fin del viaje) obtenidos en el *punto 3º* que solo son realizados por las tarjetas que quedan como resultado del *punto 4º*.

El número de tarjetas que cumplen las anteriores condiciones siguen siendo las mismas que se mencionaron en el punto anterior, lo cual indica que la exportación fue realizada con éxito**.** Hasta este punto tenemos todos los viajes realizados entre las 6 y las

10 de la mañana, pero nos falta unir los viajes realizados por la tarde ¿Por qué necesitamos los viajes de la tarde? Para poder responder a uno de los objetivos planteados: ¿Cuántas horas pasan de media desde que una persona sale de casa hasta que vuelve del trabajo?

¿Qué tarjetas y viajes realizados por la tarde analizaremos? Aquellos viajes que quedan como resultado de aplicar:

1º. Filtrar aquellas etapas en las que la hora de inicio del viaje, (el campo generado nuevo llamado **hora_viaje**) este comprendido entre las 15 y las 20 horas de la tarde.

2º. También hay que aplicar el filtro de que solo sean etapas de un viaje que se hayan realizado por alguna de las tarjetas que se han obtenido de los viajes realizados por la mañana.

3º. Agrupar las etapas por los campos **idserietarjeta** y **viaje** para realizar una agregación y obtener solo la primera y última etapa de un viaje. En caso de que un viaje solo contenga una etapa, la primera y última etapa será la misma. De esta forma, si un viaje contiene dos o más etapas o filas, obtendremos un único registro que contiene la etapa de inicio y de fin del viaje.

4º. Finalmente obtenemos los viajes realizados de la tarde (que contiene la etapa de inicio y de fin del viaje).

Del número de tarjetas que se obtuvieron de los viajes realizados por la mañana, hay 17738 tarjetas que también realizan algún viaje por la tarde entre las 15 y las 20. En resumen, hay 5 tarjetas que cumplían los requisitos de los viajes de la mañana, pero no realizan ningún viaje por la tarde, o al menos, no en el rango de horas que se han definido(15-20h).

Los datos de la mañana se han almacenado en un fichero CSV llamado result_morning_v2.csv y para los datos tarde se han almacenado en un fichero CSV llamado result_afternoon_v2.csv**.**

2.2.4 Unión de viajes realizados en la mañana y en la tarde

A continuación, se muestran los valores correspondientes al número de tarjetas distintas y número de viajes que existen para los datos de viajes en la mañana y en la tarde:

	MAÑANA	TARDE
NÚMERO DE TARJETAS	17.743	17.738
NÚMERO DE VIAJES	1.970.207	1.592.744

Tabla 7 - Datos de viajes mañana y tarde

Como se puede apreciar, hay muchos menos viajes realizados por la tarde, pero el objetivo es poder relacionar el viaje realizado por la mañana y por la tarde en un mismo día. Antes de realizar esta unión, hay que analizar el siguiente caso: para una tarjeta en un determinado día, por ejemplo, la tarjeta YYYYYY en el día 03-03-2019, ha podido haber realizado dos viajes distintos, uno que empieza a las 7 de la mañana y otro a las 9 o 10 de la mañana en el mismo día (por ejemplo, ha podido ir al médico antes de ir al trabajo). Además, la misma casuística puede darse por la tarde (por ejemplo, puede haber salido a las 17:00 y más tarde ha podido ir a dar un paseo a otra zona), el siguiente paso es ¿cómo podemos hacer resolver este problema? En el caso de la mañana, nos quedaremos con el último viaje realizado por la mañana y en el caso de la tarde, quedarnos con el primer viaje realizado.

Para poder realizar la tarea mencionada anteriormente, lo primero que se hizo fue cargar los datos obtenidos del apartado anterior, el fichero ***result_morning_v2.csv*** en un Dataframe y el fichero ***result_afternoon_v2.csv*** en otro Dataframe. Ambos Dataframe contienen las siguientes columnas:

- Los campos: **idserietarjeta** y **viaje** indican qué tarjeta ha realizado el viaje y la fecha exacta de inicio del viaje.
- Los siguientes campos: *aniomes, parte_viaje, fxtransaccionterminal_lag, actor_dpp_lag, linea_red_lag, parada_red_lag, zone_etapa_start_lag, geocodigo_etapa_start_lag, municipio_start_lag, parada_red_finish, zone_etapa_finish, geocodigo_etapa_finish, municipio_finish, distance, fxtransaccionterminal, pasajeros_viaje_completo, point_start_lag_lat, point_start_lag_lon, point_finish_lat, point_finish_lon, hora_viaje, weekdaynum_viaje, weekday_viaje, monthday_viaje, monthday_viaje y date_viaje*, contienen dos sufijos: "first" y "last" indicando que la columna contiene los datos correspondientes a la etapa inicial o la etapa final de un viaje. Por ejemplo, la columna aniomes hay dos columnas: "*aniomes_first*" haciendo referencia a la etapa de inicio del viaje y "*aniomes_last*" hace referencia a la etapa del fin del viaje.

Para ambos Dataframes, se añadió una columna nueva llamada **idserietarjeta_viaje** que juntaba la información de las columnas **idserietarjeta** y **date_viaje_last**, aunque hubiese sido indistinto haber seleccionado la columna *date_viaje_first* porque ambas columnas contienen la misma información, ya que las etapas inicial y final se realizan en la misma fecha(día-mes-año).

IDSERIETARJETA	DATE_VIAJE_LAST	IDSERIETARJETA_VIAJE
A3944A7A843CCACC6A8A76BB93783D40	01-01-2019	A3944A7A843CCACC6A8A76BB93783D40_01-01-2019
2EF67DAF0F757429D192E6E57BCA29CD	01-01-2019	2EF67DAF0F757429D192E6E57BCA29CD_01-01-2019

Tabla 8 - Ejemplo de unión de la columna tarjeta con la columna de fecha del viaje

Con esta nueva columna en ambos Dataframes, en la que se puede dar el caso de que existan filas duplicadas porque pueden existir varios viajes en un mismo día para una misma tarjeta, la librería de **Pandas** de **Python** ofrece una función para poder eliminar las filas duplicadas de un Dataframe en base a los valores duplicados existentes en una determinada columna o columnas y además, tiene una propiedad llamada ***keep*** que permite indicar el valor que se desea mantener tras el borrado de los duplicados (el primer o el último valor duplicado). Aplicando la función mencionada con anterioridad, tanto al Dataframe de los viajes de la mañana como los de la tarde, se obtienen dos tablas libres de posibles filas o viajes duplicados.

Inicialmente, la tabla o Dataframe de los viajes en la mañana contenía 1.970.207 filas o registros y tras aplicar los pasos correspondientes a la eliminación de viajes realizados por una tarjeta en un mismo día, manteniendo solo el último viaje realizado en ese día, el número de filas se redujo a 1.951.454, lo cual indica que había 18.753 viajes de algunas tarjetas que habían realizado más de un viaje en un mismo día. Por otro lado, la tabla de los viajes realizados en la tarde contenía 2.363.133 filas y después de aplicar los pasos correspondientes a la eliminación de dos o más viajes realizados por una tarjeta en un mismo día, manteniendo solo el primer viaje realizado en ese día (es el momento que sale del trabajo y regresa a casa), el número de filas se redujo a 1.738.491, es decir, había 624.642 viajes de algunas tarjetas que habían realizado más de un viaje en un mismo día.

Para poder obtener la tabla o el Dataframe que una los viajes realizados por la mañana con los viajes realizados por la tarde, objetivo que se busca en esta sección, la librería de Pandas ofrece una función llamada ***merge*** que permite unir dos Dataframe en base a un determinado campo (en este caso **idserietarjeta_viaje**), es decir, permite realizar una sentencia INNER JOIN similar a las que se pueden hacer en SQL como se aprecia en la figura 5 (elaboración externa [17])

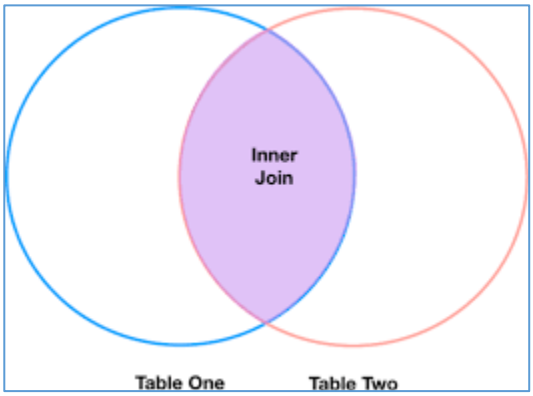

Figura 5 - Ejemplo de inner join en Pandas

Partiendo de la base de que ambas tablas contienen las mismas columnas, al realizar la unión la tabla final contendrá columnas con el mismo nombre, solo que unas de esas columnas contendrán los datos referentes al viaje realizado en la mañana y las demás a los datos referentes al viaje realizado en la tarde. La función *merge*, permite poner sufijos que se añadirán a las columnas correspondientes a cada tabla y en este caso se declararon los siguientes sufijos '*_moring*' y '*_afternoon*' para así poder identificar las columnas correspondientes al viaje de la mañana y de la tarde. Una vez realizado la unión de ambas tablas hay que volver a verificar que las tarjetas siguen del dataset cumplen el siguiente requisito: cada tarjeta inicia más de 90 viajes en la mañana en una misma estación.

Para terminar esta sección, cabe señalar que al resultado de la tabla final se le ha añadido una columna con nombre ***horario_jornada*** que contiene el valor de la resta entre la hora en la que se realizó el viaje de la tarde y el viaje de la mañana. Por último, se obtuvieron **7.878 tarjetas** distintas que cumplen todos los requisitos vistos hasta el momento y un total de **815.666 viajes** (mañana y tarde juntos). Sobre este Dataframe o tabla final, se van a realizar una serie de consultas para poder obtener datos que permitan contestar a los objetivos.

Capítulo 3 - Análisis de los datos

En esta sección se mostrará la estructura que contiene cada una de las tablas finales que surgieron del resultado de las consultas realizadas sobre los **815.666 viajes** obtenidos de la sección 2.2.4. Estas tablas contendrán la información necesaria que permitirá contestar a las preguntas de los objetivos planteados en este trabajo. Comencemos por ver el aspecto de las tablas finales.

3.1 Viajes y tarjetas

Para obtener el número de viajes que se han realizado por cada estación hay que agrupar los viajes por la parada de metro de origen del viaje en la mañana (columna con nombre *parada_red_lag_first_moring*) y contar el número de viajes que hay por cada estación. Esta consulta se apoya en utilizar dos funciones que ofrece Pandas: *groupby* y *agg*. Además, al resultado obtenido de utilizar las funciones *groupby* y *agg* para poder saber el número de viajes que hay en cada estación, también se añadieron las columnas para saber el distrito a la que pertenece cada estación de metro y la renta per cápita de cada distrito.

IDPARADA	PARADA	COD_DISTRITO	DISTRITO	RENTA	VALIDACIONES	PORCENTAJE
64	AVENIDA DE AMERICA	4	Salamanca	24683.0	112397	13.77978241
114	SAINZ DE BARANDA	3	Retiro	21598.0	109527	13.42792270
20	NUEVA NUMANCIA	13	Puente de Vallecas	9706.0	96681	11.85301337
6	CUATRO CAMINOS	6	Tetuán	15180.0	65848	8.07291220
19	PUENTE DE VALLECAS	13	Puente de Vallecas	9706.0	58698	7.19632791

1	PLAZA DE CASTILLA	5	Chamartín	26267.0	57525	7.05251905
28	VENTAS	4	Salamanca	24683.0	56613	6.94070857
53	MONCLOA	9	Moncloa - Aravaca	22792.0	45766	5.61087504
108	PLAZA ELIPTICA	12	Usera	9552.0	41326	5.06653459
127	PRINCIPE PIO	9	Moncloa - Aravaca	22792.0	36879	4.52133594
120	NUEVOS MINISTERIOS	7	Chamberí	22897.0	33733	4.13563885
74	CANILLEJAS	20	San Blas - Canillejas	13559.0	29252	3.58627183
57	ALONSO MARTINEZ	1	Centro	16711.0	21684	2.65844108
78	PUEBLO NUEVO	15	Ciudad Lineal	15111.0	16137	1.97838331
16	ATOCHA RENFE	2	Arganzuela	17738.0	12100	1.48345033
205	PUERTA DEL SUR	11	Carabanchel	10988.0	11517	1.41197500
111	MENDEZ ALVARO	2	Arganzuela	17738.0	6965	0.85390344
189	CHAMARTIN	5	Chamartín	26267.0	3018	0.37000439

Tabla 9 - Viajes por estación

La tabla contiene las siguientes columnas:

- IDPARADA: código de la estación de metro.
- PARADA: nombre de la estación de metro.
- COD_DISTRITO: código del distrito
- DISTRITO: Nombre del distrito de Madrid
- RENTA: renta media per cápita por persona
- VALIDACIONES: número de viajes que se han iniciado en la estación
- PORCENTAJE: porcentaje del número de viajes realizados por estación.

A continuación, se muestra en la figura 6 (de elaboración propia) una gráfica de la información recogida por la tabla 9

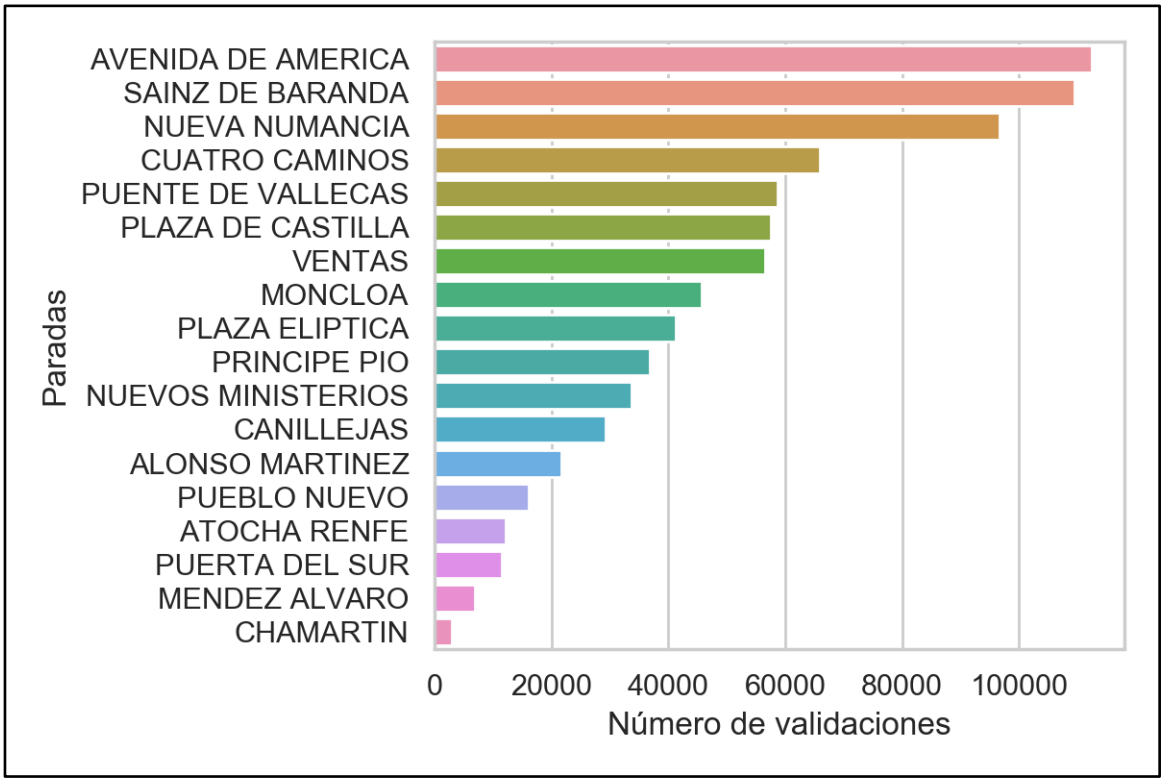

Figura 6 - Número de validaciones por estación

La estación de Avenida de América es la estación en la que se inician que más viajes seguida de la para de metro de Sainz de Baranda. Por la estación de Avenida de América pasan tres líneas de metro distintas: línea 6, línea 7 y línea 9, al estar tan bien comunicada es razonable que la mayoría de los viajes recogidos en el estudio se inicien en dicha estación.

Por otro lado, para conseguir el número de tarjetas, es decir el número de usuarios, que se han iniciado un viaje por cada estación, hay que agrupar los viajes por la parada de metro de origen del viaje en la mañana (columna con nombre *parada_red_lag_first_moring*) y contar el número de tarjetas distintas qué hay por cada estación. Para llevar a cabo esta consulta, también se utilizaron las funciones *groupby* y *agg*. Al resultado obtenido de utilizar las funciones *anteriores*, que determina el número

de tarjetas que hay en cada estación, también se añadieron las columnas que indican el distrito al que pertenece cada estación y la renta per cápita de cada distrito.

IDPARADA	PARADA	COD_DISTRITO	DISTRITO	RENTA	TARJETAS	PORCENTAJE
64	AVENIDA DE AMERICA	4	Salamanca	24683.0	3191	13.707045
114	SAINZ DE BARANDA	3	Retiro	21598.0	2393	10.279210
6	CUATRO CAMINOS	6	Tetuán	15180.0	2175	9.342784
20	NUEVA NUMANCIA	13	Puente de Vallecas	9706.0	2028	8.711340
1	PLAZA DE CASTILLA	5	Chamartín	26267.0	1792	7.697595
19	PUENTE DE VALLECAS	13	Puente de Vallecas	9706.0	1782	7.654639
28	VENTAS	4	Salamanca	24683.0	1365	5.863402
74	CANILLEJAS	20	San Blas - Canillejas	13559.0	1269	5.451031
53	MONCLOA	9	Moncloa - Aravaca	22792.0	1244	5.343643
120	NUEVOS MINISTERIOS	7	Chamberí	22897.0	1217	5.227663
127	PRINCIPE PIO	9	Moncloa - Aravaca	22792.0	1137	4.884021
108	PLAZA ELIPTICA	12	Usera	9552.0	913	3.921821
57	ALONSO MARTINEZ	1	Centro	16711.0	893	3.835911
78	PUEBLO NUEVO	15	Ciudad Lineal	15111.0	639	2.744845
16	ATOCHA RENFE	2	Arganzuela	17738.0	586	2.517182
205	PUERTA DEL SUR	11	Carabanchel	10988.0	291	1.250000
111	MENDEZ ALVARO	2	Arganzuela	17738.0	228	0.979381
189	CHAMARTIN	5	Chamartín	26267.0	137	0.588488

Tabla 10 - Número de tarjetas por estación

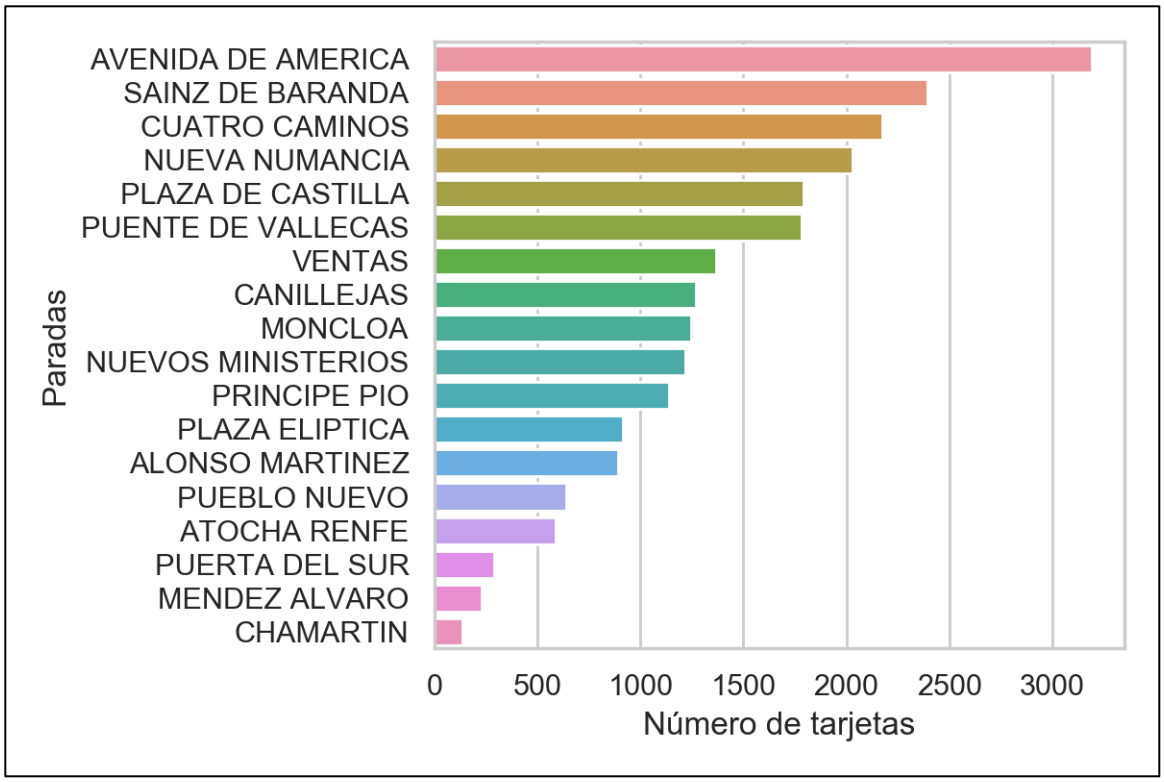

Figura 7 - Número de tarjetas por estación

Al igual que en la tabla del número de viajes realizados por estación, la Tabla 10 y la Figura 7 (de elaboración propia) se refleja que en la estación de Avenida de América es en la que más tarjetas inician algún viaje ratificando la alta correlación entre tabla 9 y la tabla 10.

3.2 Trabajo por día de la semana

Otro dato muy interesante para analizar es ver que día de la semana tiene más afluencia en cada parada. Para conseguir estos datos, es necesario agrupar los viajes por la estación de metro donde se haya realizado validación inicial en la mañana (columna con nombre *parada_red_lag_first_moring*) y por el día de la semana en el que se realizó el viaje: lunes, martes, miércoles, fueves, viernes, sábado y domingo (columna con nombre *weekday_viaje_first_moring*). De cada una de estas agregaciones, hay que contar el número de viajes que entran en cada agrupación y finalmente, también se

añadieron las columnas para saber el distrito a la que pertenece cada para de metro y la renta per cápita de cada distrito. La tabla obtenida contiene el formato mostrado en la tabla descrita a continuación.

COLUMNA	DESCRIPCIÓN
PARADA	Código de la estación de metro
PARADA_TEXT	Nombre de la estación de metro
DISTRITO_COD	Código del distrito
DISTRITO	Nombre del distrito
DISTRITO_RENTA_PERSONA	Renta per cápita media
L	Nº de viajes realizados el lunes
M	Nº de viajes realizados el martes
X	Nº de viajes del miércoles
J	Nº de viajes del jueves
V	Nº de viajes del viernes
S	Nº de viajes del sábado
D	Nº de viajes del domingo
L_per	% de viajes realizados el lunes sobre el total
M_per	% de viajes realizados el martes sobre el total
X_per	% de viajes realizados el miércoles sobre el total
J_per	% de viajes realizados el jueves sobre el total
V_per	% de viajes realizados el viernes sobre el total
S_per	% de viajes realizados el sábado sobre el total

D_per	% de viajes realizados el domingo sobre el total

Tabla 11 - Viajes realizados por día en cada estación

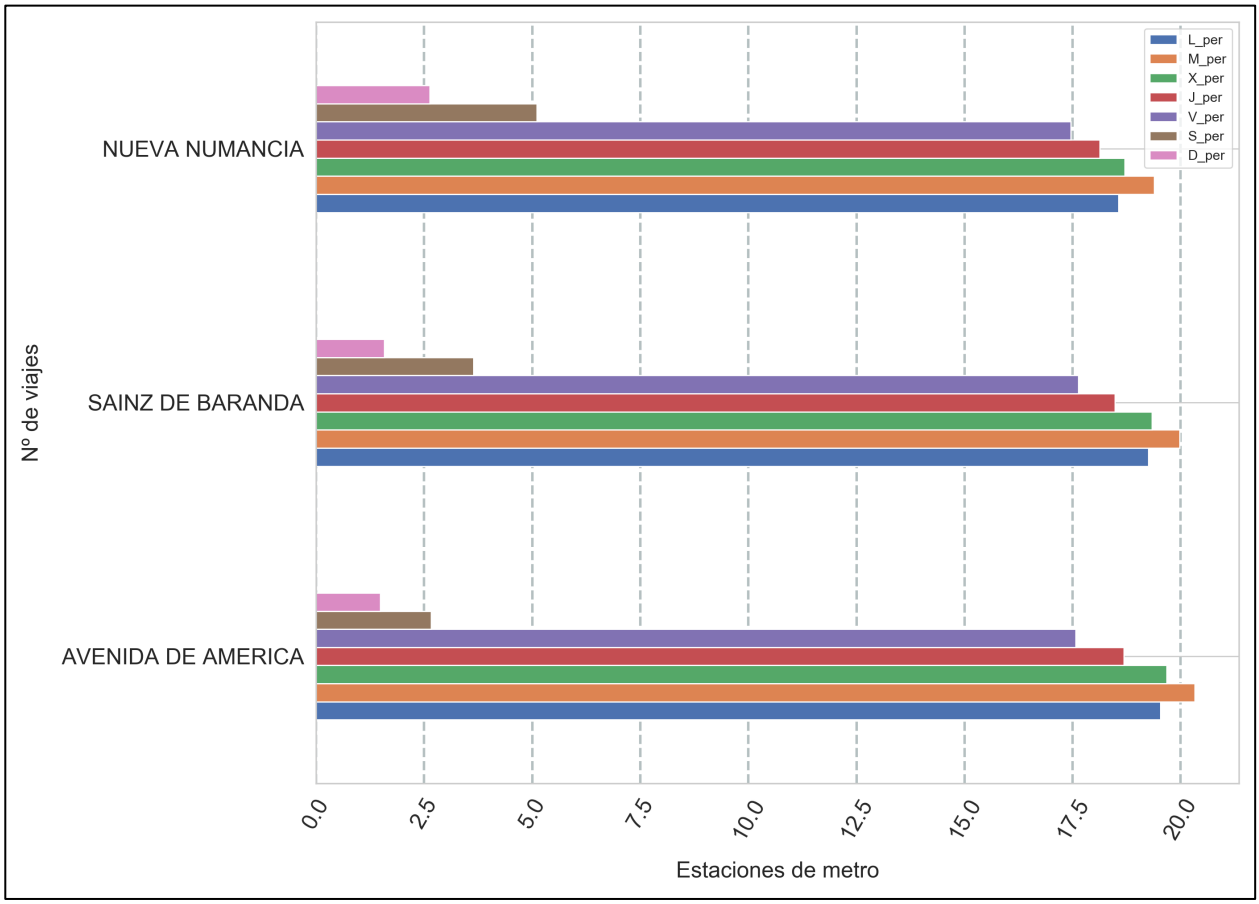

Figura 8 - Nº de viajes por día en cada estación

En la figura 8, se aprecia que el día en el que más viajes se realizan, o también se puede interpretar como el día en que más se trabaja, es el **martes** (color naranja). Por ejemplo, en la estación de Avenida de América, teniendo en cuenta el total de los viajes iniciados en dicha parada, más del 20% de los viajes se han realizado el martes. También se puede observar, que más del 95% de los viajes se realizan entre semana de lunes a viernes, aunque el viernes es el día en que el número de viajes baja considerablemente.

Esta tabla sirve para contestar a uno de los objetivos del trabajo: saber qué **día de la semana se trabaja más**, que en este caso **es el martes**.

3.3 Horas trabajas al día

Cuando una persona inicia un viaje por la mañana y regresa en el trayecto de la tarde, pasan cierta cantidad de horas, lo cual puede interpretarse como la jornada que tiene una persona. Para obtener este dato, al igual que se hizo en para obtener las tablas anteriores, se usaron agrupaciones y agregaciones para obtener la media de horas que pasan desde que salen de casa hasta que salen del trabajo en cada día de la semana, es decir, se agrupó todos los viajes realizados por el día de la semana y se calculó la media de horas(jornada) para cada día de la semana. Finalmente se obtuvo una tabla como la descrita a continuación:

DIA_SEMANA	JORNADA_MEDIA
D	8.76501286
S	8.79670599
V	8.90804396
J	9.53724426
X	9.54785076
M	9.55876565
L	9.56179619

Tabla 12 - Jornada media por día de la semana

Esta tabla permite contestar a otra de las preguntas que se plantearon inicialmente **¿Cuantas horas está una persona fuera de su hogar de media?** Calculando la media total entre todos los días, se obtuvo que la medía de horas trabajas estaba entorno a **9,2**, es decir, que de media una persona esta mas de 9 horas, pero para ser

más concretos, en realidad este caso se da de **lunes a jueves**, en el fin de semana la cantidad de horas esta comprendida entre las de 8 y las 9 horas.

La figura 9 es una representación gráfica de la tabla 12, donde se puede apreciar mejor los datos correspondientes a la tabla 12.

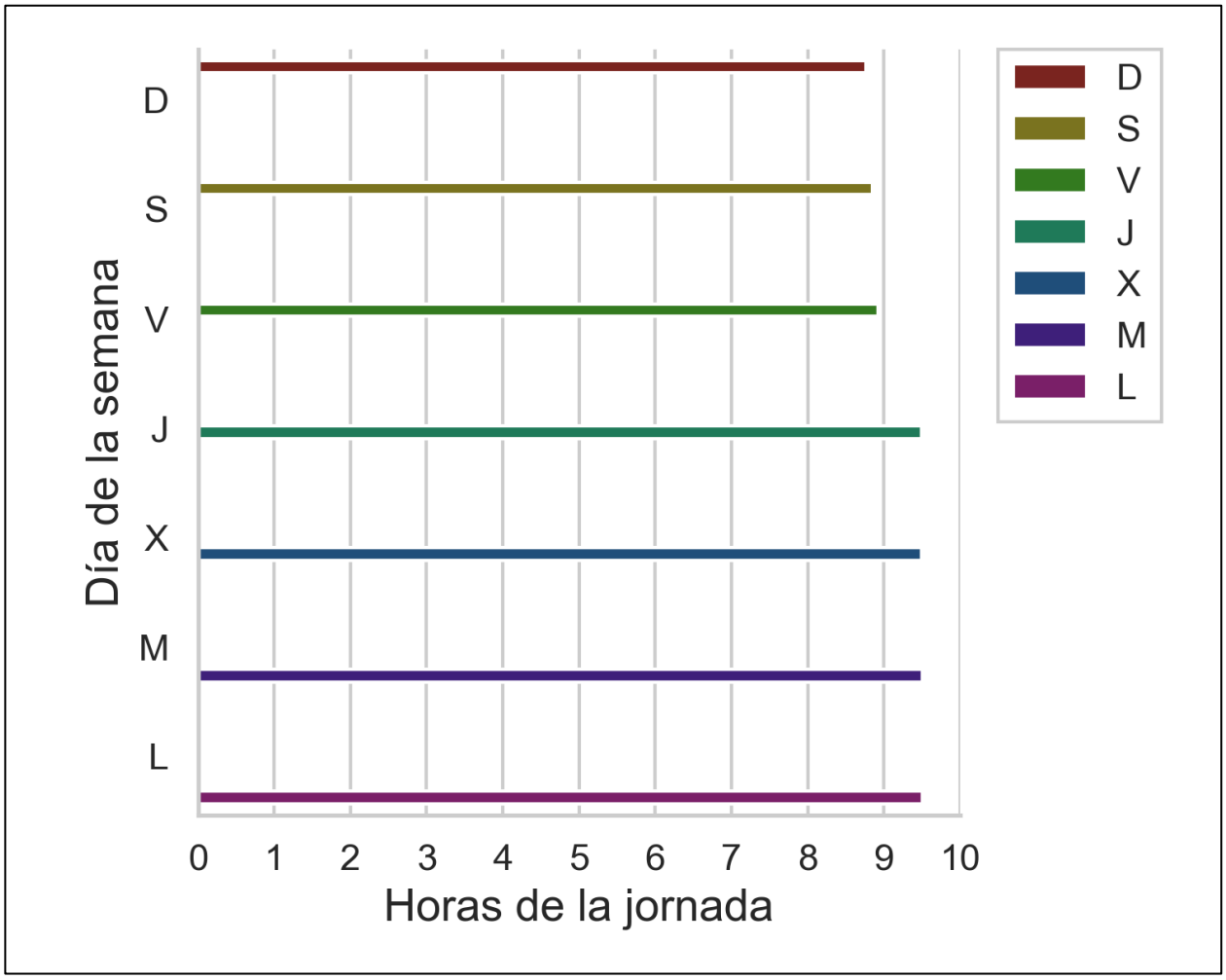

Figura 9 - Horario de la jornada media por día de la semana

Analizando la figura 9, se puede concluir y corroborar que no hay un día específico que se trabaje más horas, más bien hay un grupo de días en el que las personas pasan más horas en el trabajo que son de lunes a jueves, una media de más de 9 horas y 30 minutos. En el caso del fin de semana el número de horas baja en media hora y se queda entre las 8h 30min y las 9 horas.

Por último, hay que añadir que no se observaron diferencias significativas de las horas trabajas en cada día con la renta, pero sí se realizó la tabla de diferencias con el test estadístico **t-Student [18]** entre las horas trabajas por día de la semana en global. Para los días en los que hay diferencia de medias estadísticamente significativa se han marcado con el símbolo **<** ó **>,** y en el caso de que el test no indique que hay diferencia se ha marcado con el símbolo **=**. Los resultados finales se pueden apreciar en la tabla 13.

	L	M	X	J	V	S	D
L	**=**	**=**	**=**	**=**	>	>	>
M	**=**	**=**	**=**	**=**	>	>	>
X	**=**	**=**	**=**	**=**	>	>	>
J	**=**	**=**	**=**	**=**	>	>	>
V	<	<	<	<	**=**	>	>
S	<	<	<	<	**=**	**=**	>
D	<	<	<	<	**=**	**=**	**=**

Tabla 13 - Diferencias entre horas trabajadas por día de la semana

Vemos que el test señala los mismos resultados: de lunes a jueves el número de horas es similar, mientras que de media el viernes se trabaja menos horas, el sábado aun menos horas que el viernes y el domingo menos que el sábado.

3.4 Distancia media por estación y tarjeta

Por último, otro de los objetivos del estudio era poder saber la distancia media a la que desplaza el titular de una tarjeta hasta el trabajo desde una estación determinada. Por una parte, para conseguir la distancia media que se recorre desde una estación origen de las seleccionadas hasta el destino es necesario realizar una agrupación de los viajes por la estación de origen del trayecto y calcular la media de las distancias recorridas por todas las tarjetas que han iniciado el viaje en cada estación. En la tabla 14 se muestra los campos que contiene la consulta realizada:

COLUMNA	DESCRIPCIÓN
PARADA	Código de la estación de metro
PARADA_TEXT	Nombre de la estación de metro
DISTRITO_COD	Código del distrito
DISTRITO	Nombre del distrito
DISTRITO_RENTA_PERSONA	Renta per cápita media
DISTANCIA	Distancia media recorrida

Tabla 14 - Distancia media recorrida por origen de la estación

La figura mostrada a continuación permite contestar al objetivo, **¿cuál es la distancia media desde un punto inicio del viaje (casa) hasta el destino final (trabajo)?** Mediante la figura 10, se observa que la estación de inicio del viaje desde la cual los pasajeros recorren más distancia hasta llegar al destino es la estación de **Puerta del Sur**, unos 12 kilómetros. Haciendo la media entre todas las distancias recorridas en todas las estaciones, la media final es de **5 kilómetros.**

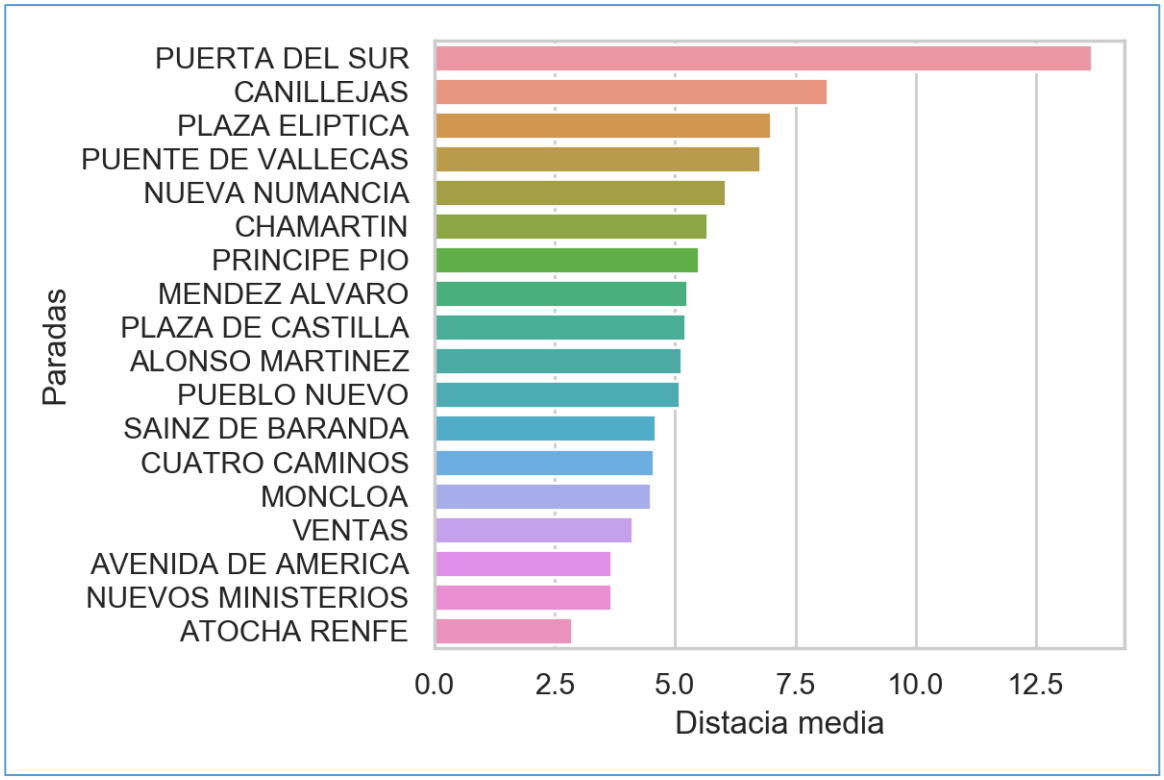

Figura 10 - Distancia media por estación

Además, se puede conseguir la distancia media recorrida por cada tarjeta de transporte desde cada estación de origen. La única diferencia con la consulta mencionada anteriormente para conseguir los datos de la tabla 14, es que es necesario agrupar por la estación de origen y tarjeta de transporte y realizar la media de las distancias recorridas por la tarjeta. Esta tabla finalmente contiene las siguientes columnas:

COLUMNA	DESCRIPCIÓN
PARADA	Código de la estación de metro
PARADA_TEXT	Nombre de la estación de metro
DISTRITO_COD	Código del distrito
DISTRITO	Nombre del distrito
DISTRITO_RENTA_PERSONA	Renta per cápita media
TARJETA	Identificador de la tarjeta
DISTANCIA	Distancia media recorrida

Tabla 15 - Distancia media recorrida por tarjeta y estación de origen

Con estas dos tablas podemos responder a uno de los objetivos que era el poder saber la distancia media recorrida por persona hasta su trabajo.

3.5 Relación entre trabajar fines de semana, festivos y renta per cápita por zona

Otra relación importante es saber si existe alguna relación entre trabajar en el fin de semana y la estación (zona) de inicio del viaje. Para lograr contestar a esta pregunta es necesario utilizar los datos de la tabla 11 que contiene los datos del número de viajes iniciados por cada estación en cada día de la semana en las estaciones que resultan de interés en este trabajo. Lo primero, fue juntar los datos correspondientes a los días entre semana (lunes-viernes) y los días del fin de semana (sábado-domingo). Los datos obtenidos fueron los siguientes:

IDPARADA	PARADA	COD_DISTRITO	DISTRITO	RENTA	ENTRESEMANA	FINDESEMANA
64	AVENIDA DE AMERICA	4	Salamanca	24683.0	0.965680	0.034320
114	SAINZ DE BARANDA	3	Retiro	21598.0	0.951448	0.048552
20	NUEVA NUMANCIA	13	Puente de Vallecas	9706.0	0.927939	0.072061
6	CUATRO CAMINOS	6	Tetuán	15180.0	0.958735	0.041265
28	VENTAS	4	Salamanca	24683.0	0.969982	0.030018
1	PLAZA DE CASTILLA	5	Chamartín	26267.0	0.964038	0.035962
19	PUENTE DE VALLECAS	13	Puente de Vallecas	9706.0	0.932510	0.067490
53	MONCLOA	9	Moncloa - Aravaca	22792.0	0.956244	0.043756
120	NUEVOS MINISTERIOS	7	Chamberí	22897.0	0.981064	0.018936
127	PRINCIPE PIO	9	Moncloa - Aravaca	22792.0	0.971463	0.028537
108	PLAZA ELIPTICA	12	Usera	9552.0	0.955437	0.044563
74	CANILLEJAS	20	San Blas - Canillejas	13559.0	0.974394	0.025606
57	ALONSO MARTINEZ	1	Centro	16711.0	0.948038	0.051962
78	PUEBLO NUEVO	15	Ciudad Lineal	15111.0	0.976610	0.023390
16	ATOCHA RENFE	2	Arganzuela	17738.0	0.947966	0.052034
205	PUERTA DEL SUR	11	Carabanchel	10988.0	0.930151	0.069849
111	MENDEZ ALVARO	2	Arganzuela	17738.0	0.971387	0.028613

189	CHAMARTIN	5	Chamartín	26267.0	0.978413	0.02158

Tabla 16 - Viajes realizados por estación entre semana y el fin de semana

Como el dato de interés a saber es la relación entre el hecho de trabajar en el fin de semana, interesan solo los valores de la **columna** del **fin de semana** (nombre de la variable ***finde_t***), pero hay un que inconveniente, y es que la columna toma valores muy pequeños (dado que se trabaja más entre semana) y es necesario estandarizar los datos para que tomen un rango entre los valores de 0 y 1, así se puede conseguir que aquellos valores más altos tomen valores muy cercanos a 1 y poder interpretarlos como que en la estación se trabaja más en los fines de semana y aquellos valores más pequeños tomen valores muy cercanos a 0 e interpretarlo como si en dicha estación se trabaja menos en los fines de semana. Para poder hacer esta operación, se utilizo una librería llamada ***preprocessing*** de ***Sklearn*** [19] que ofrece una función para llamada ***MinMaxScale*** que estandariza entre un rango dado (por defecto usa el rango entre 0 y 1). En la figura 11 se observa la distribución de la relación entre número de validaciones o viajes realizados entre la renta per cápita y el hecho de que se trabaje más en el fin de semana en cada una de las estaciones. La renta per cápita se representa mediante el tamaño y color de las circunferencias de la estación.

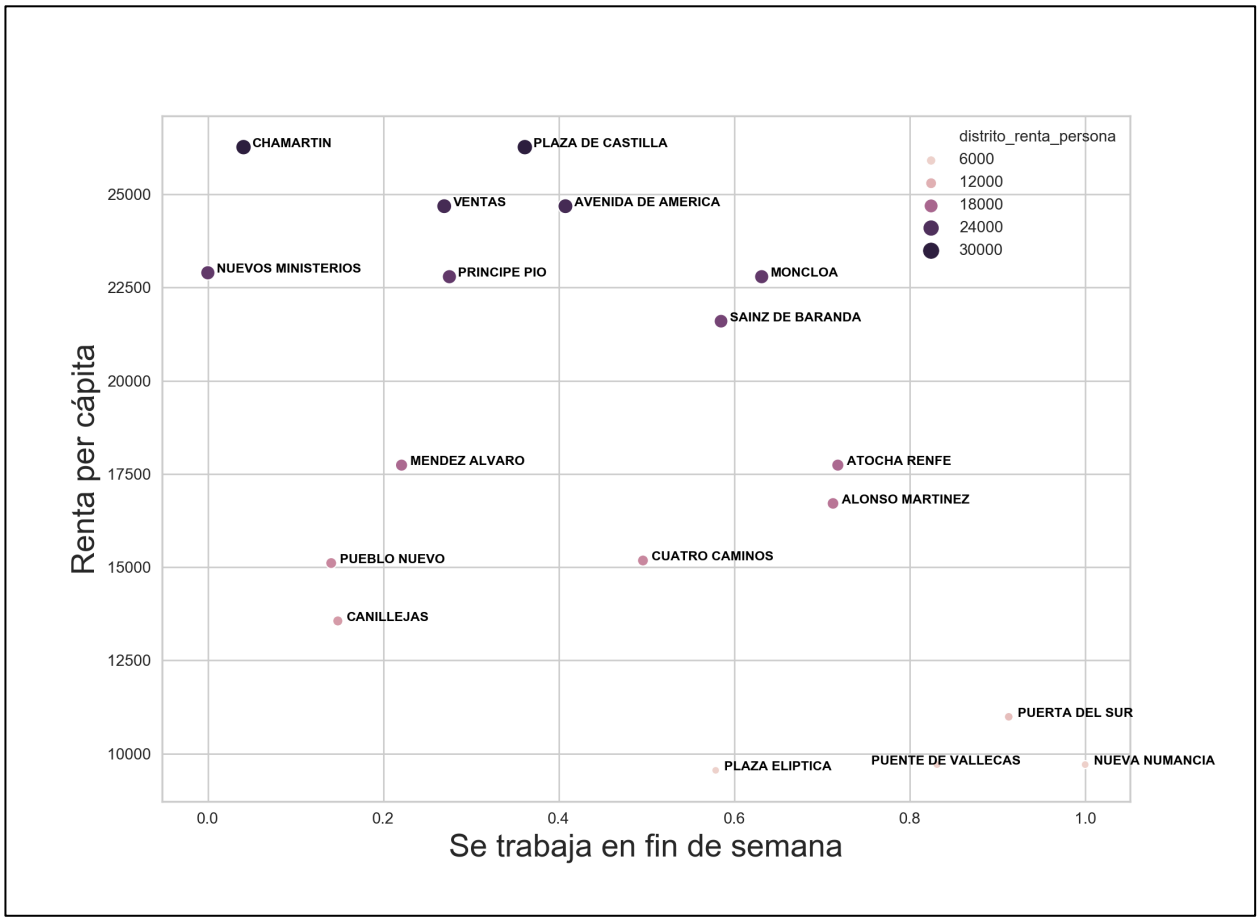

Figura 11 - Distribución del número de validaciones en cada estación en función del fin de semana

En las estaciones de *Puente de Vallecas*, *Nueva Numancia y Puerta del Sur* son en las que más se trabaja en el fin de semana, al contrario de las personas que viven cerca de *Nuevos Ministerios y Chamartín.* A priori si parece que hay una relación entre la renta per cápita de cada distrito y el hecho de trabajar el fin de semana, pero para estar seguros, es mejor ver la correlación entre ambas columnas.

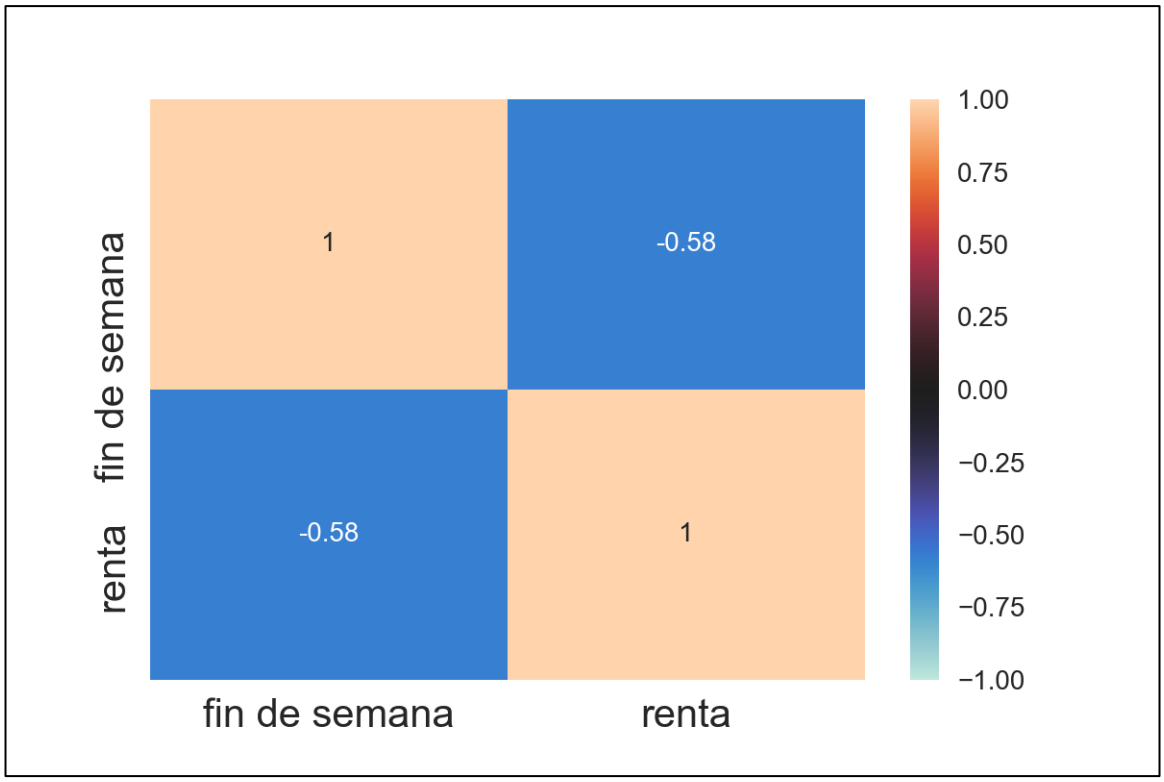

Figura 12 - Correlación renta y variable fin de semana

En la figura 11, la **correlación** que se observa es **negativa**, aunque no parece ser más significativa ya que toma el valor de ***0,58*** y para que sea significativa debe tomar valores muy cercanos a -1 o a 1. Aunque la correlación no es muy alta, se puede interpretar de la siguiente manera: cuanta más renta per cápita media tiene la zona en la que viven las personas hay menos tendencia a trabajar el fin de semana. La misma interpretación se puede hacer a la inversa, cuanta menos renta per cápita media tiene la zona en la que viven las personas hay más tendencia a trabajar el fin de semana.

3.5.1 Machine Learning: Regresión lineal

Con los datos anteriores de la renta per cápita y tras comprobar si en una estación se trabaja más o menos en el fin de semana, se intentó ver si existía algún modelo de regresión lineal [20] que permitiese predecir la renta per cápita en función de si el fin de semana se trabaja más o menos proporción. Es decir, se busca una recta que cumpla:

$$y = mx + n$$

Donde la variable *y* representa la variable a predecir, la renta per cápita por estación, y la variable *x* es la proporción de viajes que se realizan el fin de semana (si se trabaja más o menos en el fin de semana). Este tipo de modelos de aprendizaje automático se conocen como modelos supervisados, porque el modelo a entrenar aprende en base a los valores tomados por las variables de entrenamiento(características) y la variable a predecir (target). Para realizar el modelo de aprendizaje automático, se ha utilizado la librería de ***LinearRegression*** de ***Sklearn*** la cuál recibe dos parámetros: la característica que indica si se trabaja o no en el fin de semana (la variable x) y la variable a predecir es la renta per cápita (la variable *y*). Para determinar si el modelo es bueno o malo, se utilizo la medida estadística llamada **R-cuadrado o coeficiente de determinación** [21] permite determinar la calidad del modelo entrenado, tomando valores entre 0 y 1, en caso de que la renta per cápita dependiera únicamente de si se trabaja más horas en el fin de semana el valor de R-cuadrado sería muy cercano a uno, en el caso contrario, será muy cercano a cero.

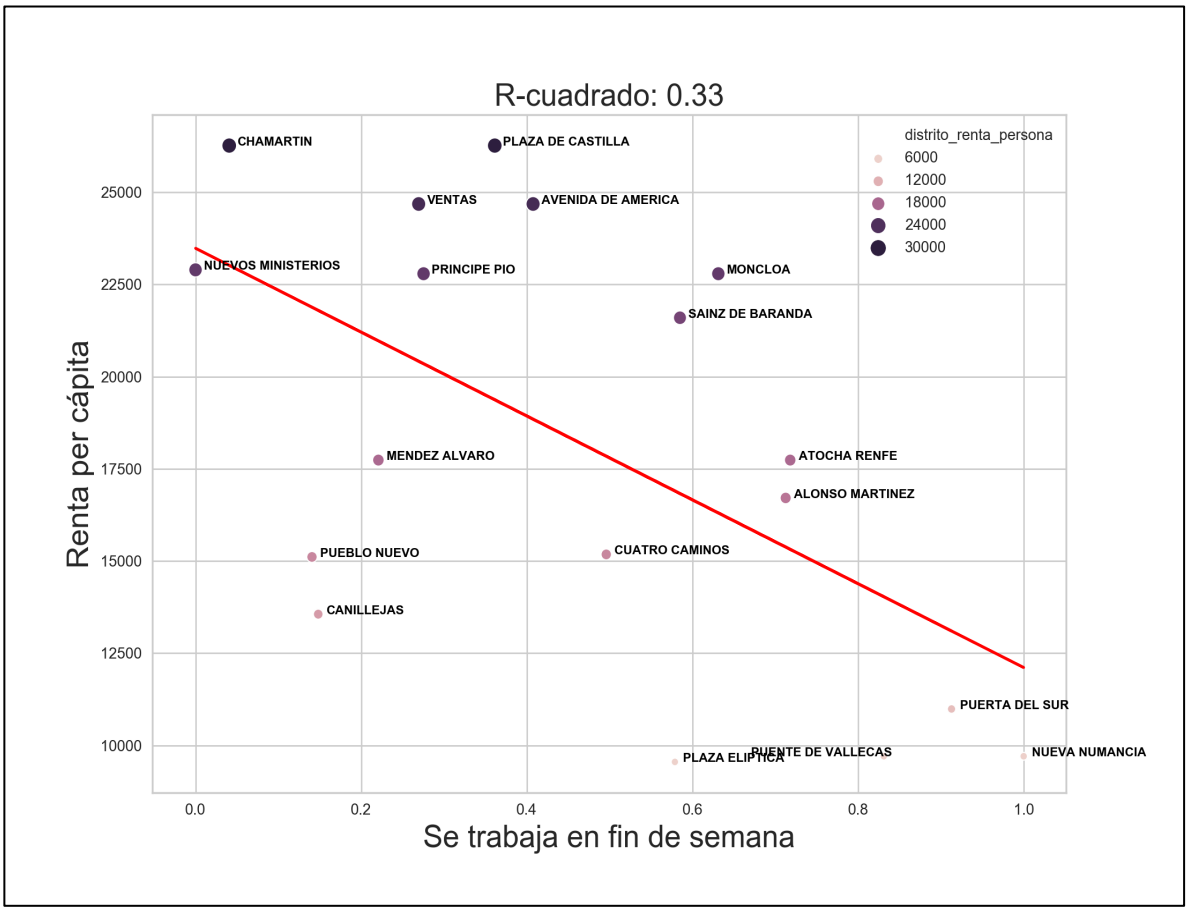

Figura 13 - Modelo de regresión lineal para la renta

En la figura 12 esta representada la ***línea de regresión*** que predice los valores de la renta y el valor obtenido para R-cuadrado que es un valor muy bajo y cercano al cero. Esto se puede interpretar de la siguiente manera: la renta no depende únicamente del valor horas que trabaja el fin de semana en una estación, por lo que **no se puede afirmar que esta recta o modelo de regresión lineal sea un modelo válido para predecir la renta per cápita media solo en base a si se trabaja en el fin de semana.**

3.5.2 Machine Learning: Clustering

Al no poder usar un modelo de regresión lineal, se buscó otras alternativas, como los **modelos no supervisados**, que a diferencia de los modelos supervisados (regresión lineal) solo usan las variables de entrenamiento o características (si se trabaja en el fin de semana) y no existe tiene una variable a predecir (target), más bien el algoritmo o modelo agrupará los datos de entrenamiento por similitud y devolverá las agrupaciones. En este caso, para realizar el **clustering** [22] se uso de la librería de **Kmeans** de Sklearn que recibe como parámetros los datos de entrenamiento, es decir, se trabaja en fin de semana (***t_finde***) y el número de agrupaciones (*k*) que puede usar para entrenar. Para saber qué número de agrupaciones es el idóneo, hay que probar distintos valores, en este caso se probó que la variable *k* tomará valores comprendidos entre el valor 2 y 12. Finalmente para evaluar que valor de agrupación es mejor, se utilizó una librería de Python llamada ***KElbowVisualizer***, la cual permite ver gráficamente el valor idóneo de *k* y utilizarlo para entrenar el modelo final de clustering de *Kmeans***.**

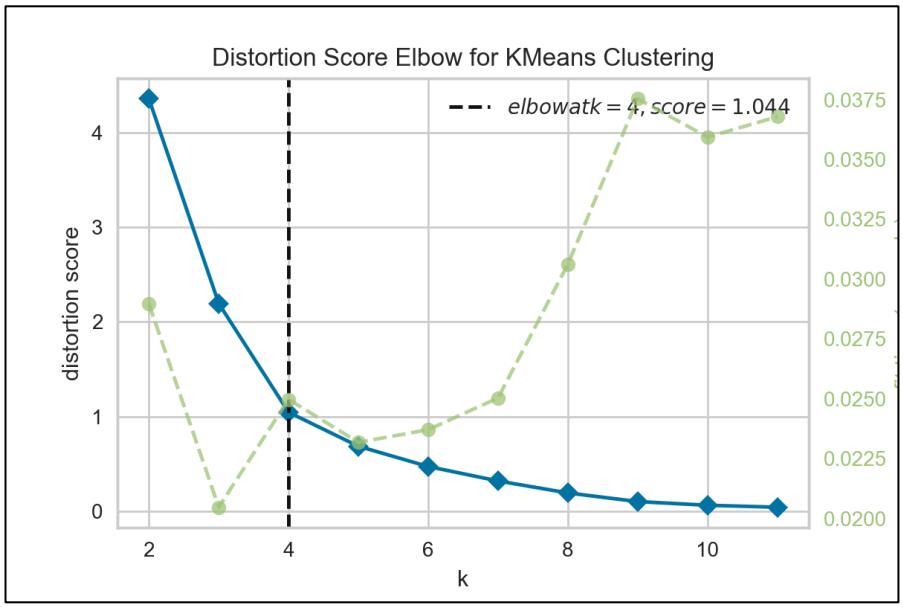

Figura 14 - Mejor valor de k para el clustering

En la figura 13, se puede observar que el mejor valor de *k* a elegir para poder agrupar los distintos valores tomados por la variable *t_finde (*si se trabaja más en el fin de semana*)* y así poder saber qué estaciones están más relacionadas en función de si se

trabaja más en el fin de semana es con el valor **k = 4**. Obsérvese que esta agrupación, que indica que se puede segmentar las estaciones según su actividad laboral en fin de semana, no tiene en cuenta la renta, que no está entre los datos de entrada. Por último, se entrenó el modelo de clustering de Kmeans indicando el valor para el número de agrupaciones a realizar debe ser es igual 4 y también los datos o características de entrenamiento sobre el cuál se debe realizar las agrupaciones (variable t_finde). Una vez aplicado el entrenamiento del modelo, las agrupaciones resultantes se muestran en la figura 14.

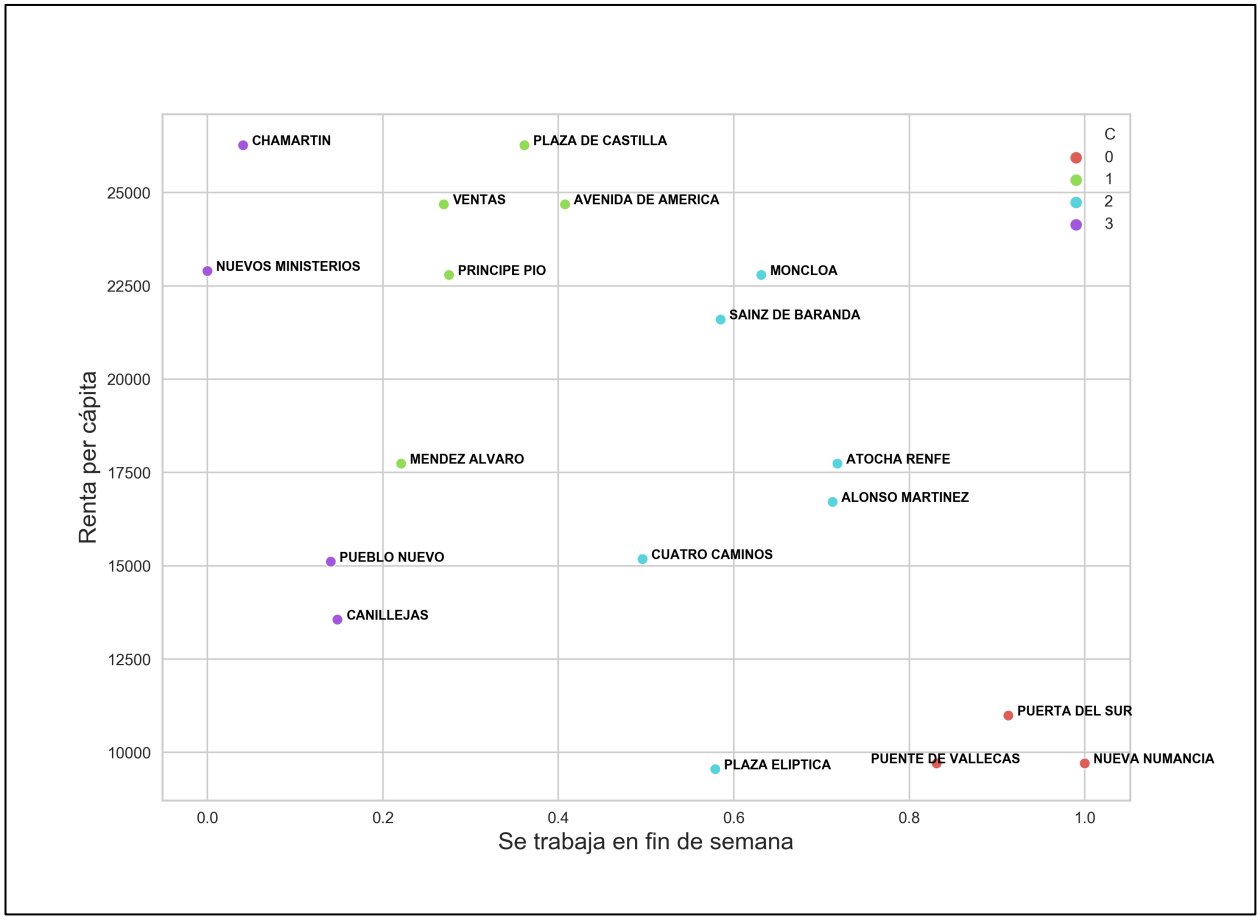

Figura 15 - Agrupaciones por estación y trabajo de fin de semana

Se puede observar una vez realizada la división o agrupación de las estaciones hay **una gran diferencia entre el grupo 0 (color rojo) y el 1 (color verde) o** entre **el grupo**

0 (color rojo) y el grupo 3 (color morado). Tanto el grupo 1 como el grupo 3 tienen de media rentas más altas y una media de número de horas trabajadas en fin de semana menor en comparación con el grupo 0 que tiene una renta mucha más baja y el número de horas trabajas en fin de semana es mucho mayor (casi rozan el 1).

3.5.3 Trabajo en festivos y renta per cápita

Resulta interesante plantearse si lo visto para fines de semana se extiende a otros días no laborables en la comunidad de Madrid. Esto lo hemos hecho en días festivos.

1. Clustering por rentas

En esta ocasión hemos segmentado las estaciones por las rentas de sus distritos. Nos salen 4 grupos:

Cluster 0	value
NUEVA NUMANCIA, PUENTE DE VALLECAS, PLAZA ELIPTICA, PUERTA DEL SUR	9988.0
Cluster 1	
CUATRO CAMINOS, CANILLEJAS, ALONSO MARTINEZ, PUEBLO NUEVO, ATOCHA RENFE, MENDEZ ALVARO	16006.16
Cluster 2	
SAINZ DE BARANDA, MONCLOA, PRINCIPE PIO, NUEVOS MINISTERIOS	22519.75
Cluster 3	
AVENIDA DE AMERICA, PLAZA DE CASTILLA, VENTAS, CHAMARTIN	25475.0

Tabla 17 - Clustering por renta

2. Añadir días festivos trabajados

Hemos seleccionado los días festivos en el calendario de la comunidad de Madrid y calculado, para cada estación qué proporción de viajes tenemos en cada caso, aplicando el escalador MinMax para que en cada festivo el máximo sea 1 y el mínimo 0. El resultado se muestra en la tabla:

1 ene.	7 ene.	18 ene.	19 abr.	1 mayo	2 mayo	15 mayo	parada	C
0.71	0.61	0.32	0.33	0.51	0.44	0.38	AVENIDA DE AMERICA	3
0.67	0.76	0.60	0.48	0.83	0.58	0.60	SAINZ DE BARANDA	2
0.85	0.87	0.79	0.76	0.96	1.00	0.93	NUEVA NUMANCIA	0
0.76	0.46	0.45	0.52	0.45	0.40	0.40	CUATRO CAMINOS	1
1.00	1.00	0.92	0.66	0.81	0.97	0.85	PUENTE DE VALLECAS	0
0.41	0.52	0.39	0.40	0.48	0.30	0.31	PLAZA DE CASTILLA	3
0.47	0.28	0.22	0.31	0.28	0.19	0.14	VENTAS	3
0.71	0.75	0.63	0.49	0.53	0.66	0.36	MONCLOA	2
0.78	0.63	0.60	0.48	0.80	0.67	0.45	PLAZA ELIPTICA	0
0.32	0.54	0.30	0.28	0.48	0.41	0.17	PRINCIPE PIO	2
0.09	0.13	0.10	0.20	0.30	0.15	0.11	NUEVOS MINISTERIOS	2
0.60	0.37	0.13	0.25	0.46	0.20	0.20	CANILLEJAS	1
0.27	0.85	0.56	0.44	0.87	0.82	0.63	ALONSO MARTINEZ	1
0.00	0.40	0.09	0.25	0.42	0.15	0.00	PUEBLO NUEVO	1
0.49	0.79	0.58	0.67	0.62	0.50	0.40	ATOCHA RENFE	1

0.51	0.68	1.00	1.00	1.00	0.62	1.00	PUERTA DEL SUR	0
0.42	0.00	0.21	0.10	0.29	0.00	0.03	MENDEZ ALVARO	1
0.00	0.44	0.00	0.00	0.00	0.18	0.70	CHAMARTIN	3

Tabla 18 - Días festivos trabajados

3. Interpretación de los resultados

Finalmente hemos considerado los datos de los festivos en cada clúster de renta y buscado diferencias estadísticas en las medias partir de la t-Student. El resultado es:

- El clúster 0, el de menor renta per cápita tiene una media de días festivos trabajados diferente, y en particular mayor, a todos los demás
- Entre el resto de los clústeres no hay diferencias significativas

3.5.4 Diferencias en hábitos laborales por renta per cápita

Para finalizar hemos repetido el mismo análisis, pero en lugar de considerar los días festivos hemos tratado todos los días. La segmentación es la misma, 4 grupos, de 0 a 3, con 0 el de menor renta y 3 el de mayor. Hemos contado en qué número de días se obtienen diferencias estadísticamente significativas en los días trabajados por clúster. La siguiente tabla resume el resultado:

-	C=0	C=1	C=2	C=3
C=0	0	59	64	101
C=1	59	0	2	1
C=2	64	2	0	2
C=3	101	1	2	0

Tabla 19 - Diferencias en hábitos laborales por renta

La tabla es simétrica y con ceros en la diagonal, por lo que hemos sombrado todos estos datos innecesarios. Se observa como para el clúster 0, de rentas más bajas, los días con media de viajes diferentes van subiendo según sube la renta, a mayor diferencia de renta más diferencias hasta llegar a 101 sobre un total de 182 días considerados. En cambio, las diferencias entre el resto de los clústeres son mínimas.

Capítulo 4 - Conclusiones y trabajo futuro

Obtener las tablas finales que contienen los datos necesarios para realizar el estudio que se pretendía en este trabajo de fin de máster, fue un trabajo más largo y complejo de lo previsto, por el hecho de tener que exportar una cantidad razonable de viajes y tarjetas desde la plataforma BigData, ya que en cada consulta que se hacía tardaba varios minutos y, además, era necesario comprobar que los datos cumplían los requisitos previstos antes de hacer la exportación final. Una vez realizada la primera tarea, el siguiente paso fue separar los viajes en la mañana y los viajes en la tarde, para finalmente juntarlos en un solo viaje (ida-vuelta) y poder obtener el tiempo que pasaba desde que salía de casa hasta que volvía del trabajo. A partir de estos datos, ya se pudieron hacer consultas con agrupaciones para obtener los datos que realmente permitieran responder a los objetivos descritos al inicio del documento:

- **¿Qué día de la semana se trabaja más?**

  En realidad, no solo hay un día en el que se trabaje más, sino que **de lunes a jueves** se trabajan más horas que en comparación con el fin de semana (viernes, sábado, y domingo).

- **¿Cuántas horas está una persona fuera de su hogar de media?**

  De medía se trabaja más de **9 horas.** No se encontró diferencias significativas en relación con la zona donde una persona reside y las horas trabajadas, por lo que no sé puede afirmar que los barrios con renta más alta trabajen menos horas al día que las zonas con renta más bajas.

- **¿Existe alguna relación entre vivir en una zona y el hecho de trabajar en fin de semana?**

  Las personas que viven en **zonas donde la renta per cápita es más baja**, tienen una mayor probabilidad de trabajar en los fines de semana, así como hacerlo en los días festivos.

- **¿Qué distancia se recorre para ir a trabajar?**

Hemos calculado una distancia media de 5 Km. Aunque en este caso no hemos buscado diferencia significativa de medias, se aprecia en la tabla que de nuevo los pasajeros de las estaciones de rentas más bajas son las que más kilómetros recorren.

El resumen final es que sí se observa influencia entre la renta y las costumbres laborales, y que esas diferencias son especialmente significativas entre las rentas más bajas y el resto.

Como a trabajo futuro proponemos trabajar en lo que son las limitaciones principales del trabajo:

1. La renta per cápita es por distritos, y puede variar considerablemente entre distintas zonas del mismo distrito. Estaría bien tener una renta per cápita de las calles aledañas a la estación de metro.
2. Una segunda limitación es la suposición de que las personas que toman el metro viven en la zona. En el caso de los intercambiadores como Avenida de América, Plaza de castilla, etc. Estas estaciones recogen muchos viajeros que acuden desde otras localidades de la región de Madrid. Creemos que esto no desvirtúa completamente el trabajo, ya que a Puerta del Sur de Madrid (distrito con renta baja), llegan viajeros de la zona sur de Madrid, que de media tiene una renta per cápita inferior a la de los municipios del norte, como las estaciones de Chamartín o Plaza de Castilla, que pertenecen a distritos con renta media más alta. Es decir, los intercambiadores se encuentran en distritos con una renta no demasiado diferente que la de los municipios cuyos viajes que llegan al intercambiador. En todo caso, esta es una apreciación subjetiva y esto se podría mejorar añadiendo información de los viajes interurbanos, de forma que sepamos qué viajeros han accedido por primera vez en Puerta del Sur, por ejemplo, y cuáles están haciendo un transbordo desde otra localidad.

3. Por último, el estudio se podría ampliar a más estaciones de metro y a otras localidades de la Comunidad de Madrid, dando una visión global de la relación entre hábitos y nivel de renta en la región.

También se podría exportar los datos con los mismos criterios, pero en el período del año del 1 de enero del 2020 al 30 de junio del 2020 y estudiar el efecto que ha tenido el COVID19 en cada una de las estaciones y ver si se ha reducido el número de validaciones y quizás intentar ver cuantas personas dejaron de ir al trabajo y se han quedado en casa teletrabajando

Chapter - Introduction

Since 2017, the Community of Madrid has allowed using magnetic technology, which has been completely replaced by contactless technology to manage all types of tickets used in public transport. As of 31 December 2017, the number of personal ticket cards was 2,602,223 [1].

The amount of information generated around the validations of these cards, whether it is the metro, EMT, Renfe or intercity buses, is very valuable because it allows for a multitude of analyses of this data. In this project, we propose to use this information to find out details about working habits in different areas of Madrid and to look for relationships (if any) with per capita income in the different districts or areas of Madrid.

Motivation

Throughout the development of the Master's Degree in the Interne of Things (IoT), we have learned the different pieces that exist within an IoT ecosystem. The typical stages for data processing in an IoT environment, although there may be different solutions, are the following:

i) **Data collection**: in the subject of Architecture of the IoT node, we discover the different types of technologies and sensors that exist to collect data of different types: temperature sensors, humidity, etc.

ii) **Communication between the nodes**: in the subject of **Networks, Protocols and Interfaces I and II**, we learned how they communicate and which protocols the nodes use with each other in an IoT solution. Besides, we could see how a node communicates with the Internet (Cloud).

iii) **Data storage and processing**: sensors can generate gigabytes or terabytes of information. With the subject of **Data Processing**, we learned how and where to store the captured data in addition to the processing that must be done on it.

iv) **Analytical models**: once these data are stored, with the subject of **Artificial Intelligence** applied to IoT, we learned how to apply analytical models to extract valuable information from the data collected by the sensor.

v) **Security and legality**: in this last subject we learned the legal aspects and the care we must have about the data that are collected and stored, as well as techniques to develop secure software and try to minimize the possible vulnerabilities that the developed software may have.

The reason for doing this work is that the current company I work for, called **VirtualDesk**, is developing a project called **Mobiam** [2] in conjunction with **System** and the **Universidad Politécnica de Madrid**, which uses the mobility data corresponding to the year 2019 for public transport in the Community of Madrid, to which I had access and could use them to do this end-of-master's work. Bearing in mind the points mentioned above, this work focuses on the **part of storage and treatment of the data**, in addition to carrying out some **analytical modeling**. The CRTM is in charge of managing and building the infrastructure that allows capturing the data from the validations collected from the contactless transport cards.

Another reason that led me to carry out this study is that the company had a BigData environment based on Cloudera, which allowed me to handle the large amount of data available to filter it and carry out the study on a much smaller amount. On the other hand, Rafael, my work director, proposed me to look for the possible relationship between the work habits of transport users and the average per capita income of the district where the departure station is located.

Objetives

As we have mentioned, the main purpose of work is to try to find out if there is a relationship between a person's working hours or time spent away from home and the average per capita income in the area where they live. Specifically:

- Which day of the week is worked the most?

- How many hours is a person away from home on average?
- Is there a relationship between living in an area, the income per capita in the area and working on weekends and on public holidays?
- What is the average distance from a starting point of the trip (home) to the final destination (work)?

Work plan and structure

To try to answer the questions in the previous section, it is necessary to process and generate the data in the format that best fits the problem. The first thing to bear in mind is that the volume of data is very large since the transport cards can be validated in different means of public transport: Metro, EMT, Renfe and interurban buses. As the final objective of the work is to see the relationship between work habits and per capita income in the districts of Madrid, I decided to focus on those transport cards that carry out some validation in the morning in certain Metro stations and then export all the validations carried out in the period from January to June 2019. The steps followed to carry out the final work were:

- **Defining origin stations**: we determined the final stations on which we would focus the study, choosing only metro stops that are in most of the different districts of Madrid: Puerta del Su, Plaza Elíptica, Puente de Vallecas, Nueva Numancia, Atocha Renfe, Nuevos Ministerios, Plaza Castilla, Príncipe Pío, Moncloa, Ventas, Pueblo Nuevo, Canillejas, Chamartín, Cuatro Caminos, Avenida América, Sainz de Baranda, Alonso Martínez and Méndez Álvaro.
- **Select cards and trips** that could correspond to people who go to work on public transport on a regular basis. To do this, we look for transport cards that are validated at the same station (of the defined stations) more than 90 times during the months of January to June between 6 and 10 a.m. This is done thinking about the most common hours of entry to work, since we are interested in the working day of these users. That

is, we assume that a person who has gone out at least 90 days (between January and June) in the morning within the 6-10 a.m. time frame is coming to work. On the other hand, we also assume that a person returns from work in the afternoon between 3pm and 8pm, as this is the most common time for leaving work. That is, in the end we will have 90 trips made in the morning between 6-10 am and trips made in the afternoon on the same day between 3-8 pm. We are aware that this is not necessarily the case; on the one hand some of these people may not go to work, and on the other hand we "lose" users who do work but with a different schedule, but we consider that this is an approach that may allow us to infer what the users' working hours are like. Furthermore, for this same reason, we must take into account that not any type of card will be processed, only those that are of the personal title or young person's subscription type will be analysed.

- **Exporting data**: only those cards that meet the above conditions.
- **Create final tables**: generate tables with the necessary data to be able to explain the objectives. To do this, the exported data will be processed in Python, to create the tables that will contain the necessary data to answer the objectives.
- **Representation and conclusions**: finally, with the final tables created from the mobility data, the final results were represented by means of graphs made in Python and the questions raised in this study were answered.

Technology

The amount of information that is generated around the validations of these cards (in the metro, EMT, intercity buses) in a single year is immense, tens of gigabytes of information, which is not possible to process in a personal computer as it is limited in resources such as the amount of RAM memory. For this reason, in this type of problem, Big Data platforms are used [3], this term is defined as the situation that occurs at the moment the data set has grown in such a way that it is difficult to handle and even more

difficult to analyze to obtain a value on them. Within this Big Data framework there is the Cloudera distribution [4] which is an integrated system based on Apache Hadoop [5], which offers a series of tools that facilitate the processing and treatment of the data. In this project, a storage tool called Hive has been used, which allows the execution of a series of queries in SQL format to reduce the information and obtain the final data on which the analysis has been performed.

As the amount of data exported from the Cloudera platform was considerably less, for the exploration and analysis of the data, Python [6] was used as a programming language and libraries such as Pandas, Numpy and Sklearn to process and analyze the data.

In the following chapter we will explain how to generate the tables with the necessary data to carry out the study. In chapter 3 we will present the analysis of these data that will allow us to answer the objectives. Finally, Chapter 4 presents the conclusions and future work.

Chapter - Conclusions and future work

Obtaining the final tables containing the data necessary to carry out the study that was intended in this end-of-master work, was a longer and more complex job than expected, due to the fact that we had to export a reasonable amount of trips and cards from the BigData platform, since each query took several minutes and, in addition, it was necessary to check that the data met the requirements before making the final export. Once the first task was done, the next step was to separate the trips in the morning and the trips in the afternoon, to finally put them together in one trip (round trip) and be able to obtain the time that passed from the time I left home until I returned from work. From this data, it was possible to make consultations with groups to obtain the data that really allowed to respond to the objectives described at the beginning of the document:

- **What day of the week do you work most?**

  In fact, not only is there a day when you work more, but from Monday to Thursday you work more hours than you do at the weekend (Friday, Saturday, and Sunday).

- **How many hours is a person away from home on average?**

  On average one works more than 9 hours. No significant differences were found in relation to the area where a person resides and the hours worked, so I cannot say that the neighbourhoods with higher income work fewer hours per day than the areas with lower income.

- **Is there any relationship between living in an area and working on weekends?**

  People who live in areas where the per capita income is lower are more likely to work on weekends, as well as on holidays.

- **How far do you travel to work?**

  We have calculated an average distance of 5 km. Although in this case we have not sought a significant difference in averages, it can be seen from the

table that once again passengers from lower income stations travel the most kilometres.

The final summary is that we do observe an influence between income and work habits, and that these differences are especially significant between the lowest income and the rest.

As to future work, we propose to work on what are the main limitations of work:

1. Per capita income is by district, and can vary considerably between different areas of the same district. It would be good to have a per capita income from the streets surrounding the metro station.
2. A second limitation is the assumption that people who take the subway live in the area. In the case of interchanges such as Avenida de América, Plaza de castilla, etc. These stations pick up many passengers who come from other locations in the Madrid region. We believe that this does not completely detract from the work, as Puerta del Sur in Madrid (a low income district) receives travellers from the southern part of Madrid, which on average has a lower per capita income than the northern municipalities, such as Chamartín or Plaza de Castilla stations, which belong to higher average income districts. In other words, the interchanges are located in districts with an income that is not too different from that of the municipalities whose journeys arrive at the interchange. In any case, this is a subjective assessment and this could be improved by adding information on intercity journeys, so that we know which travellers have entered Puerta del Sur for the first time, for example, and which are making a transfer from another town.
3. Finally, the study could be extended to more metro stations and other locations in the Community of Madrid, giving an overall view of the relationship between habits and income level in the region.

The data could also be exported using the same criteria, but for the period from 1 January 2020 to 30 June 2020, and study the effect that COVID19 has had on each of the stations and see if the number of validations has been reduced and perhaps try to see how many people have stopped going to work and have stayed at home teleworking

BIBLIOGRAFÍA

[1] Informe anual de CRTM año 2017, «La tarjeta Transporte Público», página 37: Available: https://www.crtm.es/media/651608/informe_anual.pdf

[2] Proyecto Mobiam, <<Movilidad Optimizada mediante Big-data, Integración integrada y Algoritmia Multi-modal>>. Available: https://www.cedint.upm.es/es/proyecto/mobiam

[3] Ohlhorst, Frank J. (2012). Turning Big Data into Big Money.. 2018, de SAS Sitio web: https://www.sas.com/storefront/aux/en/spbdabm/65113_excerpt.pdf

[4] Cloudera, Wikipedia. Available: https://es.wikipedia.org/wiki/Cloudera

[5] White, T. (2012). Hadoop: The definitive guide. " O'Reilly Media, Inc.".

[6] McKinney, W. (2012). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. " O'Reilly Media, Inc.".

[7] Borthakur D: HDFS architecture guide: Hadoop Apache Project.

[8] Sitio Big Data, <<¿Cómo se almacena los datos en DataNodes? Bloques de datos HDFS>>. Aviable: https://sitiobigdata.com/2019/06/25/hadoop-para-principiantes/#

[9] T. Condie, N. Conway, P. Alvaro, and J. M. Hellerstein. MapReduce online. In NSDI '10, May 2010.

[10] Hadoop MapReduce Tutrial: Bear, Deer, River and Car Example. Aviable: https://www.dezyre.com/hadoop-tutorial/hadoop-mapreduce-tutorial-

[11] Mehta, S., & Mehta, V. (2016). Hadoop ecosystem: An introduction. International Journal of Science and Research (IJSR), 5(6), 557-562.

[12] Shubham Sinha, (2020). Hadoop Ecosystem: Hadoop Tools for Crunching Big Data. Aviable: https://www.edureka.co/blog/hadoop-ecosystem

[13] Portal de datos abiertos de Madrid. https://datos.madrid.es/portal/site/egob

[14] Triglav, J. (2009). Geolocation and Time.

[15] McInerney, D., & Kempeneers, P. (2015). Image (re-) projections and merging. In Open source geospatial tools (pp. 99-127). Springer, Cham.

[16] Portal del Insituto Nacional de Estadística, datos de <<Indicadores de renta media>>. Aviable: https://www.ine.es/dynt3/inebase/index.htm?padre=5690&capsel=5690

[17] John Gosset, April Wright (eds): "Data Carpentry Python Ecology lesson." Version 2017.04.0, April 2017. Aviable: https://datacarpentry.org/python-ecology-lesson/05-merging-data/index.html

[18] Sánchez Turcios, R. A. (2015). t-Student: Usos y abusos. Revista mexicana de cardiología, 26(1), 59-61.

[19] Bisong, Ekaba. "Introduction to Scikit-learn." Building Machine Learning and Deep Learning Models on Google Cloud Platform. Apress, Berkeley, CA, 2019. 215-229.

[20] Weisberg, S. (2005). Applied linear regression (Vol. 528). John Wiley & Sons.

[21] Rodríguez, E. M. (2005). Errores frecuentes en la interpretación del coeficiente de determinación lineal. Anuario jurídico y económico escurialense, (38), 315-331.

[22] Garre, M., Cuadrado, J. J., Sicilia, M. A., Rodríguez, D., & Rejas, R. (2007). Comparación de diferentes algoritmos de clustering en la estimación de coste en el desarrollo de software. REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software, 3(1), 6-22.

APÉNDICES

Apéndice A - Código utilizado para el desarrollo

El código realizado en Python que se ha usado para la desarrollo y obtención de las tablas y gráficas necesarias para la exposición del trabajo, están disponibles en Google Drive:

https://drive.google.com/drive/folders/1_xEdYOFiu8HnUSU8phz29EGcFs6ryVNa?usp=sharing

Los ficheros y carpetas contienen la siguiente información:

- Fichero **Notebook1.ipynb**: contiene el código en Python desarrollado para obtener las tablas descritas en la Capítulo 2 y 3
- Fichero **etapas_raw.csv**: contiene los datos en crudo exportados desde el entorno BigData de Cloudera.
- Carpeta **imgs**: contiene las imágenes usadas en el trabajo (tanto las imágenes consultadas como las creadas).
- Carpeta **datos_finales**: contiene los datos generados mediante el notebook (Notebook1.ipynb) descritos a lo largo del documento.
- Carpeta **distritos_madrid**: contiene los datos descargados correspondientes a la renta y los distritos de Madrid.
- Fichero **Notebook2.ipynb**: contiene el código en Python desarrollado para obtener los resultados correspondientes a los algoritmos de machine learning