

Ingeniería Informática Superior  
2016-2017

*Trabajo Fin de Carrera*

# “Sistema Recomendador de Taxis para *Big Data*”

---

Jorge Barata González

Tutor/es

Pablo Basanta Val

Leganés, 20 de Septiembre de 2017



[Incluir en el caso del interés de su publicación en el archivo abierto]  
Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento – No Comercial – Sin Obra Derivada**



Título: SISTEMA RECOMENDADOR DE TAXIS PARA BIG DATA  
Autor: Jorge Barata González  
Director: Pablo Basanta Val

## EL TRIBUNAL

Presidente: \_\_\_\_\_

Vocal: \_\_\_\_\_

Secretario: \_\_\_\_\_

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día \_\_\_\_ de \_\_\_\_ de 20\_\_\_\_ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE



# **Agradecimientos**

Gracias a Pablo, que aún sin conocerme no dudó en ofrecerse a ayudarme a sacar adelante mi última asignatura pendiente: el Proyecto de Final de Carrera.



# Resumen

Este documento propone un sistema para maximizar los ingresos de los taxistas, recomendando las zonas de la ciudad con mayor frecuencia de viajes y más cercanas a la posición del taxista en ese momento.

Analizando 10M de trazas de viajes realizados por los Taxis Amarillos de Nueva York con un cluster Spark, encontramos correlaciones entre los beneficios y el tiempo, distancia y número de pasajeros. También se encuentran que los lugares más frecuencia de viajes cambian cada hora y cada día.

Mediante clustering, el sistema computa las agrupaciones más lucrativas para cada hora y día de la semana, dando una puntuación a cada uno de los grupos basado en las correlaciones encontradas. El sistema se ejecuta varias veces sobre un clúster Spark, buscando la configuración más óptima.

Los resultados se guardan en una base de datos geoespacial, y puede consultarse mediante una aplicación web introduciendo la hora, día de la semana, y ubicación. El sistema recomienda las diez ubicaciones más cercanas, ordenadas por beneficio.

El sistema puede ser interesante para los Taxistas Amarillos de Nueva York, como una forma de incrementar los beneficios influyendo en sus desplazamientos de forma directo, libertad que los servicios competidores como Uber y Cabify no ofrecen, ya que los objetivos de tales taxistas les son fijados por la compañía.

**Palabras clave:** big data, data mining, Spark, spatial clustering, python, taxi



# Abstract

This document proposes a system to maximize the income of taxi drivers, recommending the areas of the city with more frequency of trips and closer to the position of the taxi driver at the time of the query.

Analyzing 10M traces of trips made by the New York Yellow Taxis with a Spark cluster, we found correlations between the benefits and the time, distance and number of passengers. We also found that more frequent travel places change every hour and every day.

Through clustering, the system computes the most profitabla groups for each hour and day of the week, giving a score to each of the groups based on the correlations found. The system runs several times on a Spark cluster, looking for the most optimal configuration.

The results are stored in a geospatial database, and can be viewed through a web application by entering the time, day of the week, and location. The system recommends the top ten closest locations, sorted by profit.

The system may be of interest to the Yellow Taxi drivers in New York, as a way to increase profits by influencing their wandering, something that competing services like Uber and Cabify do not offer, since the objectives of such taxi drivers are fixed by the company.

**Keywords:** big data, data mining, Spark, spatial clustering, python, taxi

# Índice general

<b>INTRODUCCIÓN .....</b>	<b>2</b>
<b><u>1. INTRODUCCIÓN Y OBJETIVOS .....</u></b>	<b>3</b>
1.1 Introducción .....	3
1.2 Objetivos .....	4
1.3 Estructura de la memoria .....	4
<b>ESTADO DEL ARTE .....</b>	<b>7</b>
<b><u>2. ESTADO DEL ARTE.....</u></b>	<b>8</b>
2.1 Recomendadores para Taxistas .....	8
2.2 Big Data Frameworks .....	10
2.3 Base de Datos Espacial .....	11
2.4 Aplicaciones Web .....	11
<b>TRABAJO REALIZADO .....</b>	<b>13</b>
<b><u>3. ANÁLISIS DE LOS DATOS .....</u></b>	<b>14</b>
3.1 Introducción .....	14
3.2 Tecnología.....	14
3.3 Dataset.....	15
3.4 Procesado de los datos .....	15
3.5 Correlacciones de características .....	16
3.6 Correlacciones de clústeres .....	19
<b><u>4. DISEÑO DEL SISTEMA.....</u></b>	<b>27</b>
4.1 Introducción .....	27
4.2 Modelo recomendador .....	27
4.3 Aplicación Web .....	28
<b><u>5. PRUEBAS Y RESULTADOS .....</u></b>	<b>31</b>
5.1 Ejecución en PC Portátil .....	31
5.2 Ejecución en clúster .....	32
<b>CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO .....</b>	<b>34</b>
<b><u>6. CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO .....</u></b>	<b>35</b>
6.1 Mejoras en el clúster .....	35

## ÍNDICE GENERAL

6.2 Validación del modelo .....	35
6.3 Mejoras del modelo.....	36
6.4 Riesgos del modelo .....	36
<b>PLANIFICACIÓN Y PRESUPUESTO .....</b>	<b>38</b>
<b><u>7. PLANIFICACIÓN .....</u></b>	<b>39</b>
7.1 Fases de desarrollo .....	39
<b><u>8. PRESUPUESTO .....</u></b>	<b>41</b>
8.1 Medios Empleados.....	41
8.2 Presupuesto del trabajo .....	43
<b><u>9. ANEXOS .....</u></b>	<b>44</b>
<b><u>A CAPTURAS APLICACIÓN WEB .....</u></b>	<b>45</b>
<b><u>B PROPIEDADES COMPLETAS DE LAS TRAZAS .....</u></b>	<b>48</b>
<b><u>C CÓDIGO .....</u></b>	<b>49</b>
<b><u>D NORMATIVA Y MARCO REGULADOR.....</u></b>	<b>51</b>
<b>REFERENCIAS .....</b>	<b>53</b>

# Índice de figuras

Figura 1. Esquema general del sistema big data desarrollado.	4
Figura 2. Distritos en GeoJSON	16
Figura 3. Correlaciones con el pago total	17
Figura 4. Correlación del rendimiento	18
Figura 5. Correlación del dinero total producido	18
Figura 6. Diagrama de dispersión de coordenadas iniciales	19
Figura 7. Detalle del diagrama de dispersión de coordenadas iniciales	20
Figura 8. Frecuencia de viajes por hora del día	21
Figura 9. Correlación de frecuencias entre días de la semana	21
Figura 10. Clústeres obtenidos por franja horaria	23
Figura 11. Polígonos de los clusters con mayor puntuación	25
Figura 12. Aplicación web tras la carga de las recomendaciones computadas	32
Figura 13. Tiempo de lectura del archivo	33
Figura 14. Tiempo de computación de clústeres	33
Figura 15. Polinomio de grado 9 y curva agregada de tiempo	36
Figura 16. Página principal	45
Figura 17. Trayecto en Google Maps	46
Figura 18. Administración de recomendaciones: listado	46
Figura 19. Administración de recomendaciones: edición	47

# Índice de tablas

Tabla 1. Costes personales .....	43
Tabla 2. Costes materiales .....	43
Tabla 3. Coste total .....	43
Tabla 4. Propiedades completas de las trazas .....	48

Bloque I

## **INTRODUCCIÓN**

# Capítulo 1

## Introducción y objetivos

### 1.1 Introducción

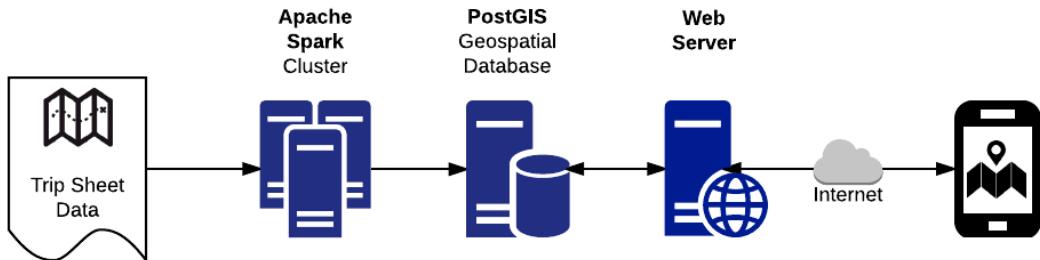
Cada vez es más fácil procesar grandes volúmenes de datos. Aplicaciones de código abierto como Spark son cada vez más populares, a la vez que maduras. Hay tecnologías que permiten desplegar un clúster en minutos<sup>1</sup>.

Por otro lado, con el advenimiento de la economía colaborativa, servicios como Uber y Cabify han provocado que los coches de dichos servicios superen en 4 a 1 a los taxis amarillos de Nueva York<sup>2</sup>.

En este proyecto vamos a realizar un sistema que ayude a los taxistas amarillos de Nueva York a ser más competitivos. La NYC Taxi & Limousine Commission publica regularmente un dataset con trazas de millones de viajes realizados mensualmente en la ciudad de Nueva York. Con un sistema big data podemos analizar qué características se correlacionan con un mayor beneficio económico para los taxistas, y ver qué lugares son más lucrativos en diferentes momentos del día y de la semana<sup>3</sup>.

Una vez tengamos la información, la guardaremos en PostGIS, una base de datos geoespacial basada en PostgreSQL<sup>4</sup>. Para que esté disponible online, crearemos una aplicación web que acceda a ella, de tal forma que los taxistas puedan consultar qué

lugares son más recomendables para su ubicación, día de la semana y momento del día. Además la integraremos con Google Maps, para que al escoger el lugar de preferencia, comience la navegación guiada.



**Figura 1. Esquema general del sistema big data desarrollado.**

## 1.2 Objetivos

El objetivo fundamental de este proyecto es el de crear un sistema que ayude a mejorar el beneficio de los conductores de taxis mediante recomendaciones, analizando el popular dataset de trazas de viajes ofrecido por NYC Taxi & Limousine Commision<sup>5</sup>. En base a ese objetivo principal, se proponen los siguientes objetivos parciales:

- Estudio del contexto y la motivación que llevan a la realización del proyecto.
- Estudio de las tecnologías que van a ser utilizadas en el desarrollo del sistema.
- Estudio de las trazas de los taxis.
- Diseño de un sistema big data que se adapte a las necesidades del proyecto.
- Implementación del sistema en base al diseño del mismo.
- Visualización de los resultados del procesamiento de los datos.
- Realización de pruebas de rendimiento del sistema en diferentes entornos.
- Extracción de conclusiones del desarrollo del proyecto.
- Planteamiento de líneas futuras ofreciendo continuidad al proyecto.

## 1.3 Estructura de la memoria

Para poder facilitar la estructura de la memoria se detalla a continuación una estructura de la misma:

- Bloque I: Introducción.
  - Capítulo 1: Introducción y objetivos.  
Motivaciones del proyecto y establecimiento de los objetivos del mismo recogiendo la idea general del trabajo que se va a realizar. Además, se enuncia la estructura de la memoria.
- Bloque II: Estado del arte.

- Capítulo 2: Estado del arte.  
Detalle de las tecnologías utilizadas en el desarrollo del proyecto las cuales deben ser conocidas por el lector previamente a la exposición de la solución técnica.
- Bloque III: Trabajo realizado.
  - Capítulo 3: Análisis de los datos.  
Nos familiarizamos con el dataset y buscamos correlaciones.
  - Capítulo 4: Diseño del sistema.  
Pasos que se han seguido en el desarrollo del sistema diseñado, desde la carga inicial de datos hasta la visualización de los resultados obtenidos en el procesado.
  - Capítulo 5: Pruebas y resultados.  
Pruebas que se han realizado para conocer los límites del entorno y los tiempos de ejecución alcanzados.
- Bloque IV: Conclusiones y futuras líneas de trabajo.
  - Capítulo 6: Conclusiones y futuras líneas de trabajo.  
Análisis de la consecución de los objetivos y las posibles líneas futuras de trabajo a realizar.
- Bloque V: Planificación y presupuesto.
  - Capítulo 7: Planificación.  
Fases de desarrollo del proyecto y su diagrama de Gantt correspondiente.
  - Capítulo 8: Presupuesto.  
En este capítulo se exponen los recursos empleados y el presupuesto para la realización del proyecto.
- Bloque VI: Anexos.
  - Anexo A: Capturas Aplicación Web.
  - Anexo B: Propiedades completas de las trazas.
  - Anexo C: Código
  - Anexo D: Normativa y marco regulador.
- Referencias



Bloque II

## **ESTADO DEL ARTE**

# **Capítulo 2**

## **Estado del Arte**

### **2.1 Recomendadores para Taxistas**

#### **T-Finder**

En 2012, la revista IEEE Transactions on Knowledge and Data Engineering publicó “T-Finder: A Recommender System for Finding Passengers and Vacant Taxis”<sup>6</sup>. El objetivo es el más parecido que he encontrado a este proyecto.

En dicho estudio se presentó un sistema de recomendación tanto para los taxistas como para las personas que esperan tomar un taxi, utilizando el conocimiento de: 1) los patrones de movilidad de los pasajeros y 2) los comportamientos de recogida / abandono de los taxistas aprendidos de las trayectorias GPS de los taxis.

En primer lugar, este sistema de recomendación proporciona a los taxistas algunas ubicaciones y las rutas a estos lugares, hacia los cuales es más probable que recojan pasajeros rápidamente (durante las rutas o en estos lugares) y maximicen los beneficios del próximo viaje. En segundo lugar, recomienda a las personas con algunas ubicaciones (a poca distancia) donde pueden encontrar fácilmente taxis vacantes.

Analizan las trayectorias GPS de los taxis utilizando un modelo probabilístico que estima el beneficio de las ubicaciones candidatas para un conductor particular basado en donde y cuando el conductor solicita la recomendación. Se basaron en trayectorias históricas generadas por más de 12.000 taxis durante 110 días y validaron el sistema con

evaluaciones extensivas incluyendo estudios de campo. Desarrollaron una aplicación móvil para el sistema.

Es el sistema más parecido que he encontrado, pero es mucho más sofisticado porque calculan trayectos, y además lo validaron con un estudio de campo.

## A cost-effective recommender system for taxi drivers

Se trata de un estudio publicado en 2014 en la revista KDD '14<sup>7</sup>.

El paper propone desarrollar un sistema de recomendación renTabla para los taxistas. El sistema se diseña para maximizar sus ganancias al seguir las rutas recomendadas para encontrar pasajeros. En concreto, diseñan primero una función de objetivo de beneficio neto para evaluar los beneficios potenciales de las rutas de conducción. A continuación, desarrollan una representación gráfica de las redes de carreteras mediante la extracción de los rastreos históricos GPS de taxis y proporcionar una estrategia de fuerza bruta para generar una ruta de conducción óptima para la recomendación.

Sin embargo, un desafío crítico a lo largo de esta línea es el alto costo computacional. Entonces desarrollan una nueva estrategia de recursión basada en la forma especial de la función de beneficio neto para buscar rutas candidatas óptimas de manera eficiente. En particular, en lugar de recomendar una secuencia de puntos de recogida y dejar al conductor decidir cómo llegar a esos puntos, el sistema recomendador es capaz de proporcionar una ruta de conducción completa, y los conductores son capaces de encontrar un cliente para el mayor beneficio potencial siguiendo las recomendaciones.

Esto hace que el sistema de recomendación sea más práctico y renTabla que otros sistemas de recomendación existentes. Finalmente, realizan experimentos en un conjunto de datos del mundo real recolectados en el área de la Bahía de San Francisco y los resultados experimentales validan claramente la efectividad del sistema de recomendación propuesto.

La diferencia principal es que buscan dar la ruta más óptima, cuando en este proyecto se abstrae de dicho cálculo.

## hubcab

*hubcab*<sup>8</sup> es una aplicación web que permite investigar exactamente cómo y cuándo los taxis recogen o dejan a los individuos en la ciudad de Nueva York. El sistema visualiza las condensadas las zonas de recogida y bajada del taxi. Se pueden ver los lugares donde sus viajes de taxi comienzan y terminan y para descubrir cuántas otras personas en su área siguen los mismos patrones de viaje.

Su objetivo principal es encontrar patrones que permitan ayudar a compartir taxis para reducir la huella de carbono. Es un poco diferente de este proyecto, pero el procesamiento de los datos y construcción del modelo es interesante.

Analizaron 170 millones de viajes en taxi de los taxis amarillos en la ciudad de Nueva York en 2011. Usando OpenStreetMap y Python, las calles fueron cortadas en más de 200.000 segmentos de 40 m de longitud. Los puntos de recogida y devolución se compararon con los segmentos de calle más cercanos. Se excluyeron tipos de calles que pudieran contener descargas de taxi o camionetas, tales como senderos, troncos, carreteras de servicio, etc. Los anchos de línea de segmentos de calle de color amarillo y azul en niveles de zoom bajos se diseñaron en una escala logarítmica. Los puntos de recogida y devolución, representados como puntos en los niveles de zoom altos, se generaron a través de un script Arcpy. La base de datos es MongoDB, y contiene todos los segmentos de calle y sus coordenadas, y todos los flujos entre cada par de segmentos de calle. El número de todos los pares de segmentos de calle posibles es de más de 40 mil millones (200.000 veces 200.000) por mapa.

## 2.2 Big Data Frameworks

El término Big Data es muy amplio, especialmente cuando se hace referencia a frameworks. En concreto, buscamos un sistema que nos permita procesar un gran volumen de datos de forma distribuida.

### 3.1.1 Apache Spark

Apache Spark proporciona una interfaz de programación de aplicaciones centrada en una estructura de datos llamada Resilient Distributed Dataset (RDD), un conjunto múltiple de sólo lectura de elementos de datos distribuidos en un grupo de máquinas, que es tolerante a fallos.<sup>9</sup>

La disponibilidad de RDD facilita la implementación de algoritmos iterativos, que visitan su conjunto de datos varias veces en un bucle, y el análisis de datos interactivo o exploratorio. Requiere de un sistema de archivos compartido. Puede usarse NFS, HDFS, y S3, entre muchos otros. Tiene un gestor de cluster nativo, pero también se puede usar Hadoop.

La librería de Spark ofrece de forma nativa algunos modelos de Machine Learning<sup>10</sup>. Se puede programar en Scala, Java o Python, lo que expande aún más su entorno de librerías. Python ofrece una gran cantidad de librerías de matemáticas, ciencia, e ingeniería.

### 3.1.2 Apache Hadoop

Apache Hadoop ha sido sobrepasado en popular por Apache Spark. En Github, podemos ver que Spark tiene hasta 4 veces más estrellas que Hadoop<sup>11</sup>.

El núcleo de Apache Hadoop consiste en su modelo de almacenamiento, conocida como Hadoop Distributed File System (HDFS), y el procesado siguiendo el sistema MapReduce<sup>12</sup>. Hadoop divide los archivos en bloques grandes y los distribuye entre los nodos de un clúster. A continuación, transfiere el código empaquetado a los nodos para procesar los datos en paralelo. Este enfoque se aprovecha de la localidad de datos, donde los nodos manipulan los datos a los que tienen acceso.

## 2.3 Base de Datos Espacial

Para almacenar los lugares de forma que podamos realizar consultarlas, necesitamos una base de datos espacial.

### 3.1.3 MongoDB

Es una de las bases de datos orientada a documentos más populares. Los describe en formato JSON, y al contrario que una base de datos relacional, no necesita un esquema, aunque también es posible. Existen interfaces para los lenguajes más populares.

Esta base de datos incluye consultas espaciales de forma nativa<sup>13</sup>.

### 3.1.4 PostGIS

Es una extensión para PostgreSQL<sup>14</sup>. Añade soporte para objetos geográficos permitiendo que las consultas de ubicación se ejecuten en SQL. El lenguaje SQL tiene interfaces para los lenguajes más populares.

PostgreSQL es una base de datos relacional, ACID y transaccional.

## 2.4 Aplicaciones Web

Todas las aplicaciones web están construidas sobre JavaScript, CSS, HTML, y se transmiten por el protocolo HTTP. Hay multitud de sistemas que facilitan su desarrollo.

Se suelen diferenciar dos partes, *Frontend* y *Backend*.<sup>15</sup>

### 3.1.5 Backend

El backend se encarga de acceder a los datos y enviar los documentos necesarios al cliente para que los visualice. Suelen seguir una arquitectura MVC<sup>16</sup>. Hay frameworks

para todos los lenguajes. Algunos ejemplos son Django para Python, RAILS para Ruby, Symfony para PHP.

### **3.1.6 Frontend**

El frontend es la capa de presentación. Consiste en el pintado del documento HTML, aplicando los estilos CSS, y la dinaminación de la página en JavaScript.

Señalar que la integración con Google Maps se realiza a este nivel.

Existen frameworks para crear frontends muy sofisticados, pero los requisitos del frontend en este proyecto son muy sencillos y no serán necesarios.

Bloque III

## **TRABAJO REALIZADO**

# **Capítulo 3**

## **Análisis de los datos**

### **3.1 Introducción**

Nos familiarizamos con el dataset y buscamos correlaciones que podamos aplicar en un modelo de recomendación.

### **3.2 Tecnología**

Hemos decidido utilizar Apache Spark principalmente por cuatro razones:

- Contiene librerías de matemáticas y machine learning que podremos aplicar en el proceso.
- Como hemos visto en el Estado del Arte, es especialmente útil para exploraciones interactivas de los datos.
- Parte del trabajo que hagamos aquí lo podremos aplicar posteriormente en un cluster.

- Ofrece interfaz en Python, permitiendo usar librerías de matemáticas e ingeniería con la que el autor tiene algo de experiencia previa: la librería SciPy<sup>17</sup>.

La exploración se realiza sobre un portátil MacBook Pro, procesador 2.2 GHz Intel Core i7 y 16 GB 1600 MHz DDR3 de memoria. Se configura Spark para usar 10 GB.

### 3.3 Dataset

Por recomendación directa del tutor, usaremos el dataset de NYC Taxi & Limousine Commision que mencionamos anteriormente.

Los taxis de Nueva York se dividen principalmente en dos:

- Taxis Amarillos o Medallón, que operan en los cinco distritos de la ciudad de Nueva York: Manhattan, Brooklyn, Queens, The Bronx, y Staten Island.
- Taxis Verdes, que operan en Upper Manhattan, The Bronx, Brooklyn, Queens (excluyendo el aeropuerto LaGuardia y el aeropuerto internacional John F. Kennedy) y Staten Island.

Nos centraremos en los Taxis Amarillos. Se añaden trazas cada cierto tiempo, y están divididas por meses. Cada mes tiene unos 10 millones de trazas de taxis. Nos centraremos en las siguientes propiedades:

- `tpep_pickup_datetime`: Fecha y hora inicial
- `tpep_dropoff_datetime`: Fecha y hora final
- `Passenger_count`: Número de pasajeros
- `Trip_distance`: Recorrido en millas
- `Pickup_longitude`: Coordenada longitud
- `Pickup_latitude`: Coordenada latitud
- `Total_amount`: Pago total

Las propiedades completas las podemos ver en el Anexo B.

### 3.4 Procesado de los datos

Los datos deben sufrir un proceso de trasformación y limpieza. Aquí trataremos de dar formato a los datos y desechar aquellos que no son válidos mediante reglas de negocio y manipulaciones requeridas por el sistema de destino. Los registros

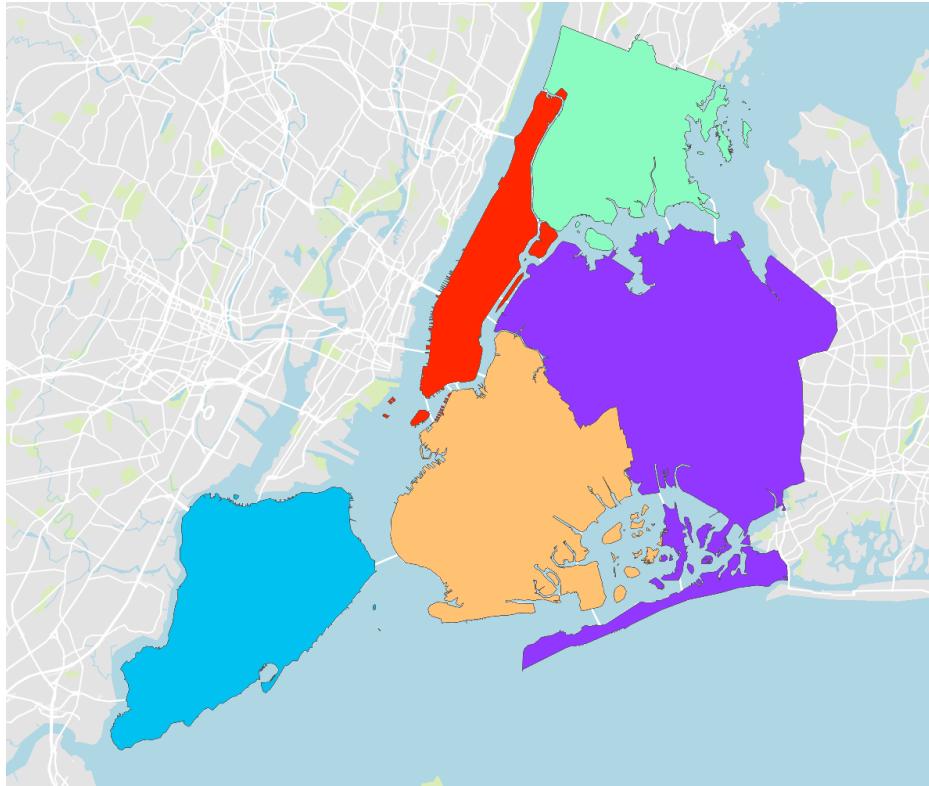
provenientes de los taxis no tienen transformaciones directas sobre los campos de la traza, pero se eliminan los registros considerados no válidos.

Tras un sondeo inicial encontramos trazas tan imposibles como con localizaciones en mitad del mar, con 24 horas de tiempo recorrido, o con ningún pasajero, entre otros. Decidimos filtrar los datos de tal forma que cumplan las siguientes condiciones:

- Las coordenadas deben de estar dentro de las áreas de los distritos en los que operan los taxis amarillos.
- La distancia recorrida debe ser mayor que cero y menor que 100 millas.
- El pago total debe ser mayor que cero.
- La fecha de recogida debe ser menor que la fecha final.
- El tiempo de recorrido debe ser inferior a 3 horas.

Para detalles sobre la implementación, por favor consultar el módulo “lib/process.py” en el Anexo C Código.

La comprobación de los puntos pertenecientes a los distritos no fue trivial. Usamos una distribución libre de los datos en GeoJSON<sup>18</sup>, que leemos en Python con Descartes y Shapely<sup>19</sup>. Por favor consultar el módulo “lib/boros.py” en el Anexo C para más detalles.

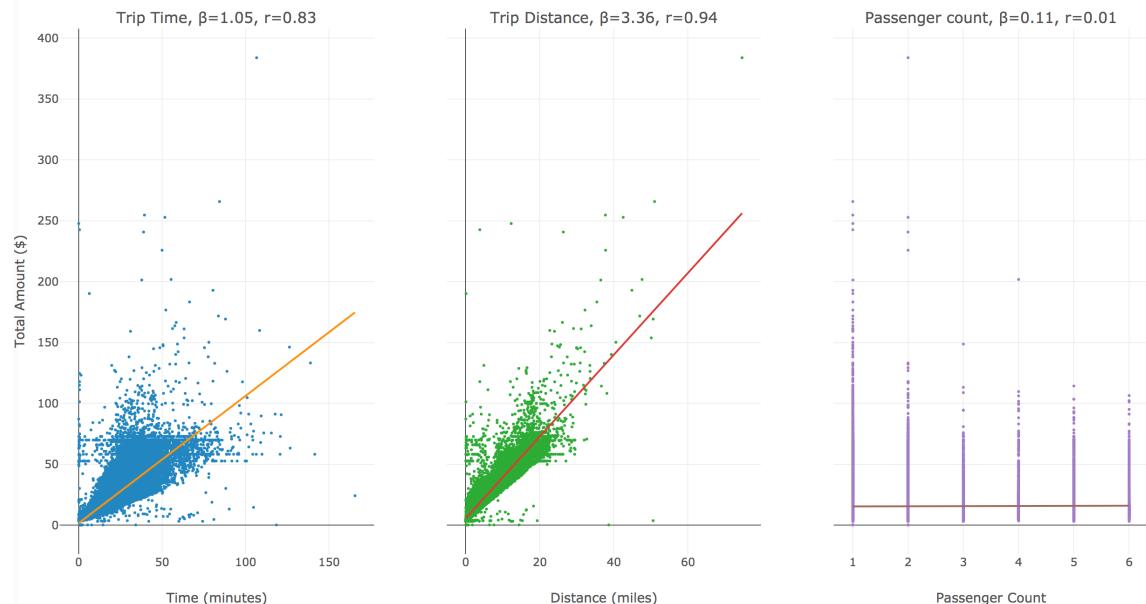


**Figura 2. Distritos en GeoJSON**

## 3.5 Correlacciones de características

Para explorar decidimos tomar una muestra aleatoria del 1% de las trazas de Enero 2017. Si analizamos todas las trazas, el tiempo de computación haría imposible el análisis interactivo.

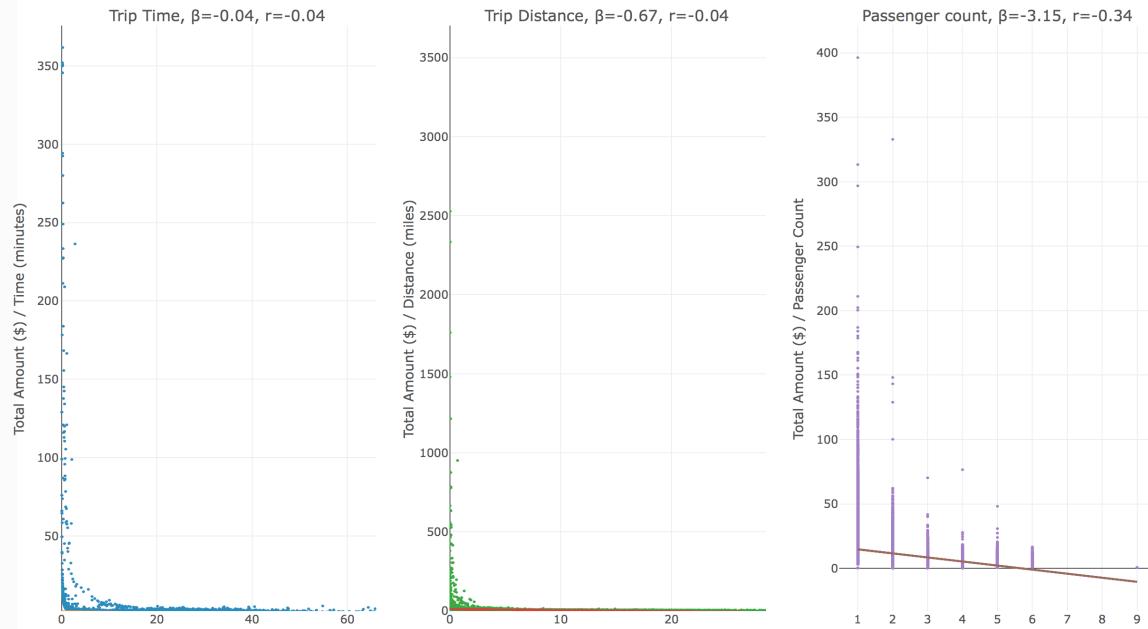
Si comparamos el tiempo, la distancia, y el número de pasajeros directamente con el pago en cada traza, vemos que aparentemente sólo las dos primeras están relacionadas. En la Figura 3. Correlaciones con el pago total vemos los diagramas de dispersión junto con la regresión lineal.



**Figura 3. Correlaciones con el pago total**

$\beta$  es el coeficiente de la regresión lineal, y  $r$  la correlación.

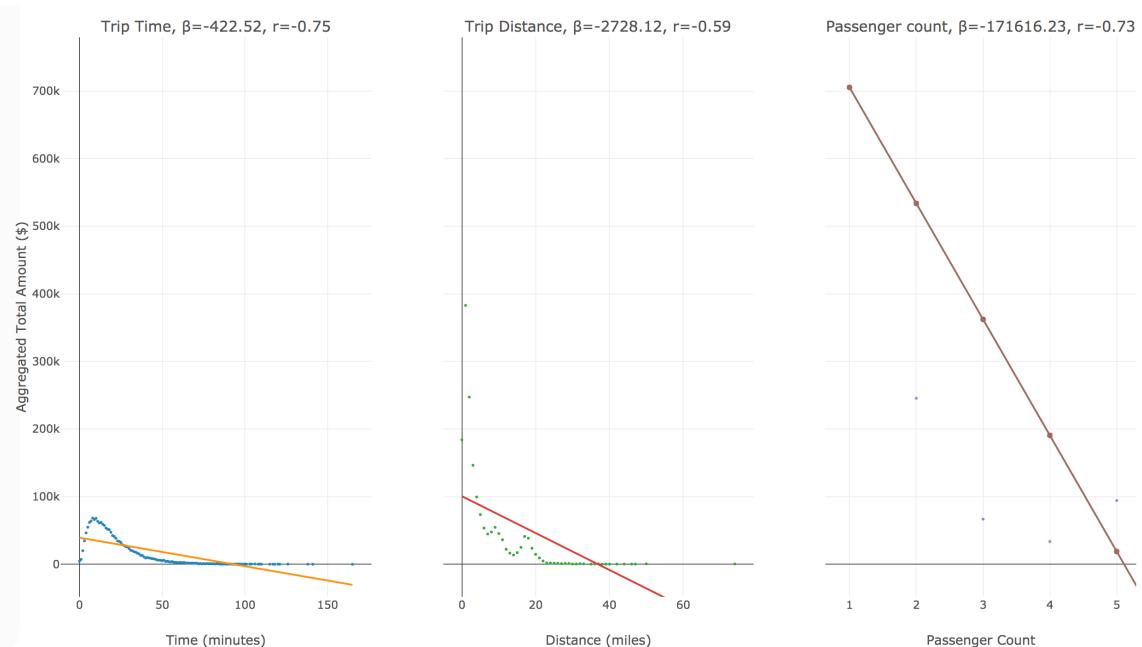
Sin embargo, si comparamos el ratio de precio por cada característica, veremos que cuanto mayor sea la última, menor rendimiento ofrece. Por ejemplo, cuanta más distancia recorrida, menor el pago por milla. En la Figura 4. Correlación del rendimiento, vemos que hay una caída exponencial del rendimiento para las tres características.



**Figura 4. Correlación del rendimiento**

$\beta$  es el coeficiente de la regresión lineal, y  $r$  la correlación.

Para terminar, si sumamos los pagos hechos por cada valor entero de cada característica, podemos observar qué franjas obtienen mejores resultados. En la figura, vemos que las trazas que han llevado sólo un pasajero han sido las más productivas comparadas con las demás características. La segunda característica más significativa es la distancia, seguida por el tiempo.

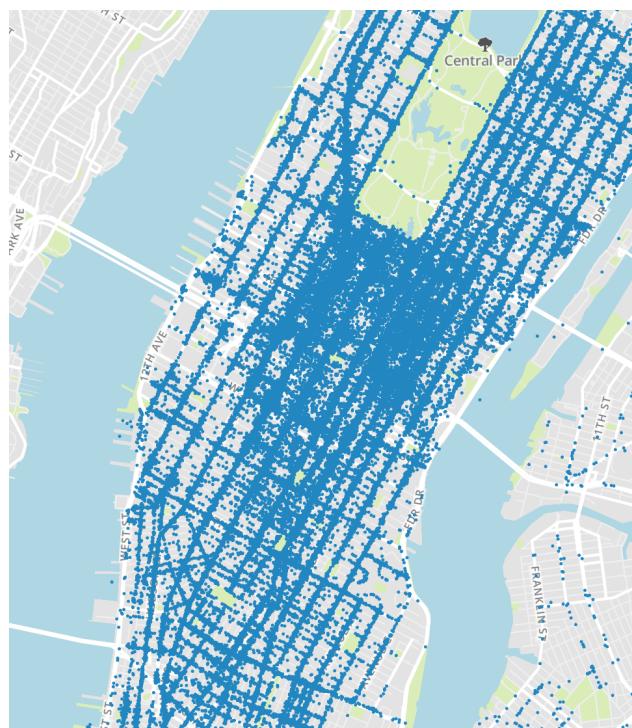


**Figura 5. Correlación del dinero total producido**

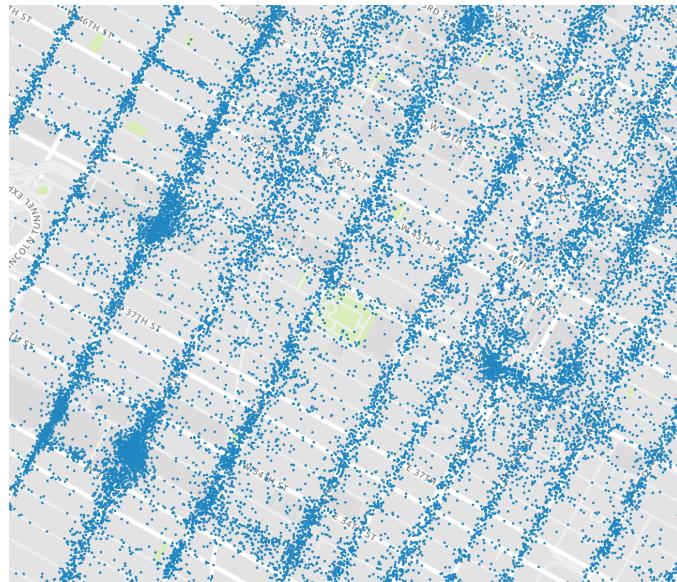
$\beta$  es el coeficiente de la regresión lineal, y  $r$  la correlación.

## 3.6 Correlaciones de clústeres

Si observamos al diagrama de dispersión de la Figura 6. Diagrama de dispersión de coordenadas iniciales, podemos detectar agrupaciones de lugares de recogida a simple vista. Y si hacemos zoom, veremos que algunas agrupaciones son más más circulares, y otras totalmente alargadas. En la Figura 7. Detalle del diagrama de dispersión de coordenadas iniciales, vemos que en algunas calles la densidad es tan alta que podemos ver cómo la agrupación toma la forma de la calle.



**Figura 6. Diagrama de dispersión de coordenadas iniciales**



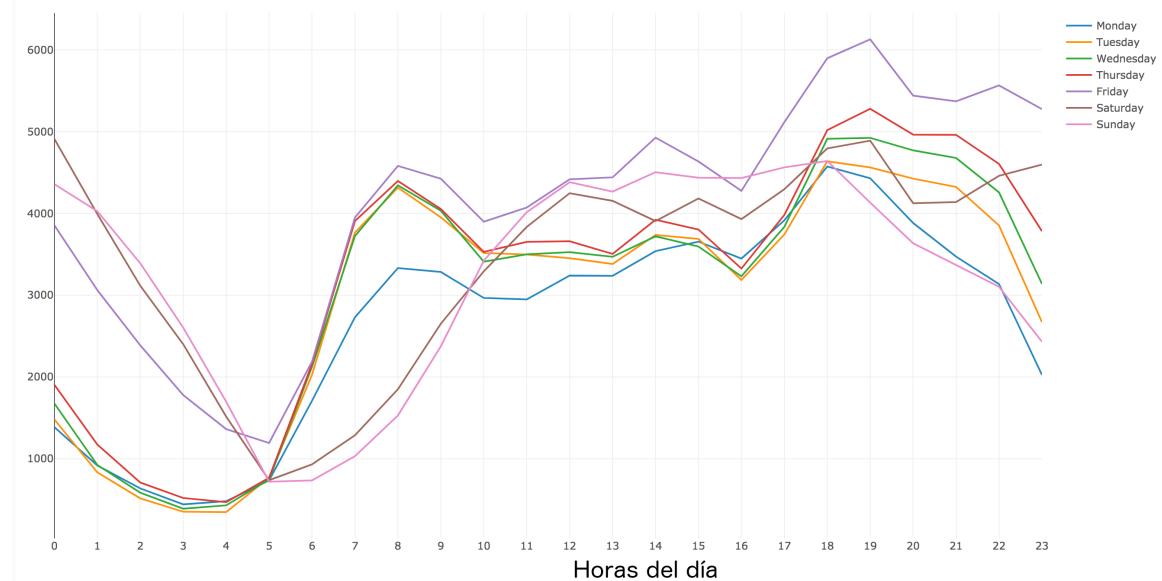
**Figura 7. Detalle del diagrama de dispersión de coordenadas iniciales**

Además, las frecuencias de las trazas cambian para cada hora y día de la semana. En la Figura 8. Frecuencia de viajes por hora del día podemos ver que los horarios de más actividad fluctúan entre las 8 de la mañana y las 12 de la noche, y que la actividad cae especialmente a las 5 de la mañana.

También podemos apreciar que:

- La noche de los lunes es similar a la de martes, miércoles y jueves.
  - Martes, miércoles y jueves son prácticamente idénticos.
  - Los viernes noche y sábado noche son muy parecidos.
  - Los sábados y domingos son parecidos por la mañana, pero la noche del domingo es muy inferior.

En la tabla se ve numéricamente estas similitudes.



**Figura 8. Frecuencia de viajes por hora del día**

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Monday	1.0	0.97	0.96	0.95	0.89	0.52	0.49
Tuesday	0.97	1.0	0.99	0.98	0.89	0.42	0.33
Wednesday	0.96	0.99	1.0	1.0	0.92	0.46	0.33
Thursday	0.95	0.98	1.0	1.0	0.94	0.5	0.34
Friday	0.89	0.89	0.92	0.94	1.0	0.74	0.56
Saturday	0.52	0.42	0.46	0.5	0.74	1.0	0.89
Sunday	0.49	0.33	0.33	0.34	0.56	0.89	1.0

**Figura 9. Correlación de frecuencias entre días de la semana**

Coefficiente  $r$  de correlación Pearson

Sabiendo esto, podemos suponer que las agrupaciones cambian también con el día y la hora. Vamos a tratar de encontrarlos mediante algoritmos de clustering.

Para clustering espacial en minería de datos, DBSCAN es uno de los algoritmos más citados<sup>20</sup> en las publicaciones científicas, pero no está disponible en la librería estándar de Spark. Las opciones disponibles son K-Means|| y el Gaussian Mixture Model (GMM). Las distribuciones observadas parecen gausianas, de modo que el GMM puede ser útil.

Como estamos explorando de forma interactiva, dividimos el día en franjas de 4 horas y computamos los clusters para cada una de las franjas. De esta forma tendremos

un número manejable de combinaciones para comparar de forma intuitiva, y además reducimos el tiempo de computación.

Computamos 1000 clústeres para cada franja horaria. En la Figura 10. Clústeres obtenidos por franja horaria se pueden apreciar que hay ligeras variaciones entre las zonas más frecuentes en cada momento del día. Al noreste, justo debajo del Central Park, la actividad aumenta por la tarde y la noche.



**Figura 10. Clústeres obtenidos por franja horaria**

Utilizando algoritmos *convex hull* podemos encontrar los vértices mínimos necesarios para representar las formas de los clústers. Además, podemos dar una puntuación a cada uno de ellos basándonos en las correlaciones encontradas con la distancia, tiempo y pasajeros, y escoger los 10 más renTablas.

Siendo  $t$  el tiempo de la traza,  $d$  la distancia de la traza, y  $p$  los pasajeros de la traza, podemos computar  $S$ , la puntuación de la traza:

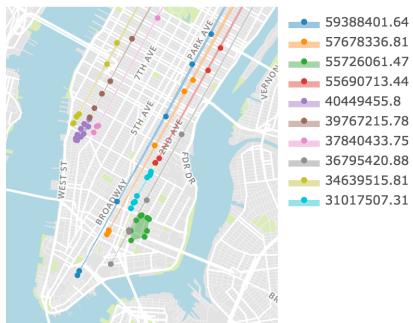
$$S = 422.52 t + 2728.12 d + 171616.23 p$$

Los coeficientes son las  $\beta$  computadas en la Figura 5. Correlación del dinero total producido.

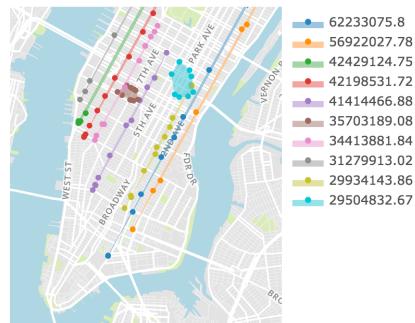
La puntuación total del cluster será igual a la suma de las puntuaciones de las trazas contenidas.

Así, en la Figura 11. Polígonos de los clusters con mayor puntuación podemos ver los polígonos computados para los top 10 mejores clusters de cada franja horaria:

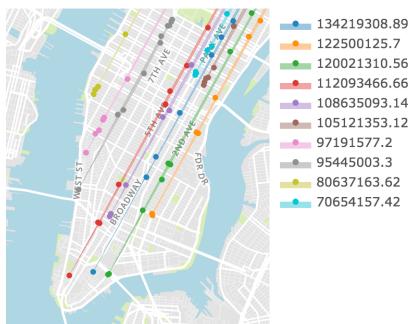
0-4 hours



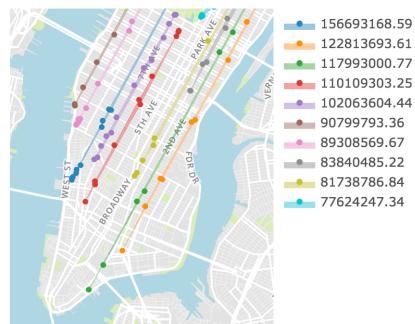
4-8 hours



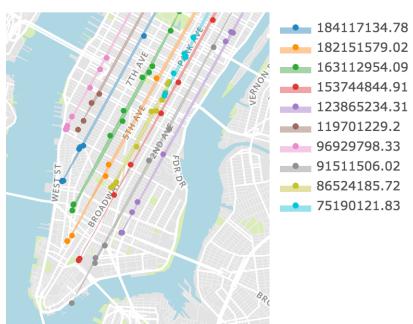
8-12 hours



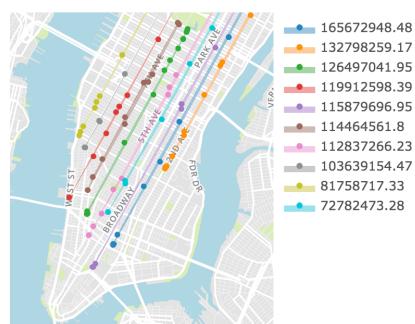
12-16 hours



16-20 hours



20-24 hours

**Figura 11. Polígonos de los clusters con mayor puntuación**

Cada leyenda indica la puntuación obtenida por el clúster



# **Capítulo 4**

## **Diseño del sistema**

### **4.1 Introducción**

El sistema estará compuesto por dos partes principales:

- El modelo de recomendador, que calculará las recomendaciones. Dicho modelo se implementará como una aplicación para Spark.
- La aplicación web, que interpretará los resultados y permitirá a los taxistas interactuar con ellos.

A continuación se detallará su diseño.

### **4.2 Modelo recomendador**

Seguiremos con el sistema de puntuación que concebimos en la última fase del análisis. Recordemos que la puntuación de un clúster es la suma total de las puntuaciones de cada una de las trazas que contiene. La puntuación de una traza la calculamos a partir de los coeficientes obtenidos en la fase de análisis.

Si volvemos a mirar las puntuaciones de los clústers calculados por éste método en la Figura 11, veremos que su orden de magnitud es muy alto y para los humanos es difícil de comparar. Optamos por dividir la función que ya teníamos por 100 para tener unas puntuaciones más manejables:

$$S = 0.01 (422.52 t + 2728.12 d + 171616.23 p)$$

Siendo  $t$  el tiempo de la traza,  $d$  la distancia de la traza, y  $p$  los pasajeros de la traza, y  $S$  la puntuación de la traza.

## Estrategia

Implementaremos el modelo en forma de un script en Python que Spark se encargará de ejecutar. Los resultados se serializarán en un archivo mediante la librería *pickle*<sup>21</sup>, disponible en la librería standard de Python.

La estrategia a seguir con el planificador Spark es la siguiente:

1. Para cada día de la semana
  - a. Para cada hora
    - i. Filtrar el Dataset por día y hora
    - ii. Computar los clústeres con GMM sobre los resultados filtrados
    - iii. Obtener las trazas, cada una de ellas con el clúster asignado

A partir de aquí, los vértices de los polígonos y las puntuaciones totales se realizan en el nodo localmente.

Para más detalles sobre su implementación, consultar el módulo “compute\_clusters.py”.

## 4.3 Aplicación Web

Para la aplicación web, tras revisar el estado del arte escojemos las siguientes tecnologías debido a que el autor ya tiene experiencia con ellas:

- PostgreSQL + PostGIS para la base de datos espacial
- Django (librería en Python) como web framework
- Nginx como servidor web

No esperamos un tráfico muy alto, por lo que consideramos razonable desplegar todos los componentes en un solo VPS de mínimas características. Escogemos la opción más básica en DigitalOcean<sup>22</sup>:

- 512 MB de RAM

- 1 vCPU
- 20GB de disco duro SSD
- 1TB de transferencia

Para la carga de las recomendaciones computadas por Spark, se implementa un script usando el propio Django. Para más detalles, por favor consultar el módulo “`project/driver/taxi/management/commands/import_recommendations.py`”.

El frontend web mostrará un formulario con la hora, día de la semana, y las coordenadas latitud y longitud, que estarán por defecto configuradas para la fecha actual. Se integrará con Google Maps<sup>23</sup> para mostrar un mapa con la ubicación dada y las recomendaciones disponibles a 10 millas alrededor, indicando cada una su puntuación en el ícono sobre el mapa. Al hacer click sobre cualquiera de ellas, se redirigirá a Google Maps con las coordenadas de la recomendación para que el conductor pueda poner el navegador.



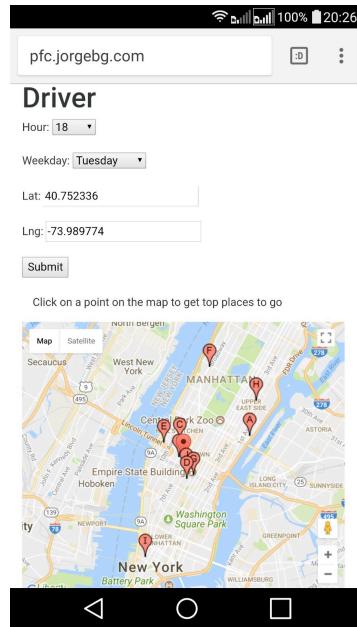
# **Capítulo 5**

## **Pruebas y resultados**

### **5.1 Ejecución en PC Portátil**

Primero se ejecutará el sistema completo sobre el 1% de las trazas de Enero de 2017 para comprobar que funciona. Se ejecutará sobre un MacBook Pro, con procesador 2.2 GHz Intel Core i7, 16 GB 1600 MHz DDR3 de memoria, y disco duro SSD.

La ejecución se resuelve favorablemente. Las recomendaciones se computan en 4 horas, y su carga en la base de datos tarda 4 segundos. Dichas recomendaciones son los usados en la aplicación web desplegada en <http://pfc.jorgebg.com>. Para más detalles por favor consulte el Anexo A Capturas Aplicación Web.



**Figura 12. Aplicación web tras la carga de las recomendaciones computadas**

## 5.2 Ejecución en clúster

Se procede a ejecutar la computación de clusters para 24 horas de un sólo día. Se dispone de cuatro servidores dados por el departamento de Ingeniería Telemática con las siguientes características:

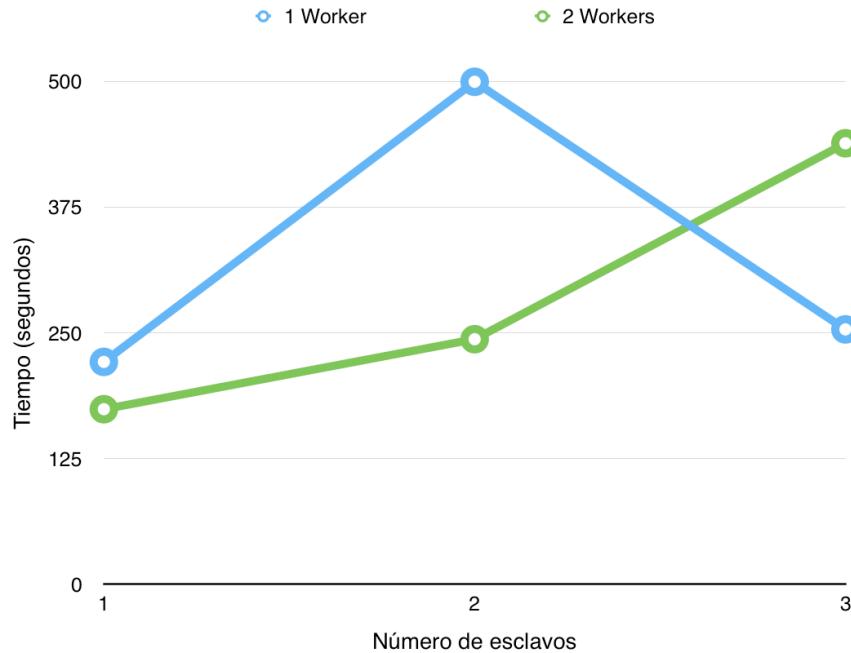
- Procesador Intel(R) Core(TM) i5-4460 CPU @ 3.20GHz
- 16 GB de RAM
- Disco duro con 30GB de cuota, compartido por NFS

Sabiendo que son máquinas compartidas, se va a evitar sobrepasar los 10GB de uso de memoria RAM. Se van a ejecutar las pruebas en múltiples iteraciones probando diferentes conjuntos de máquinas:

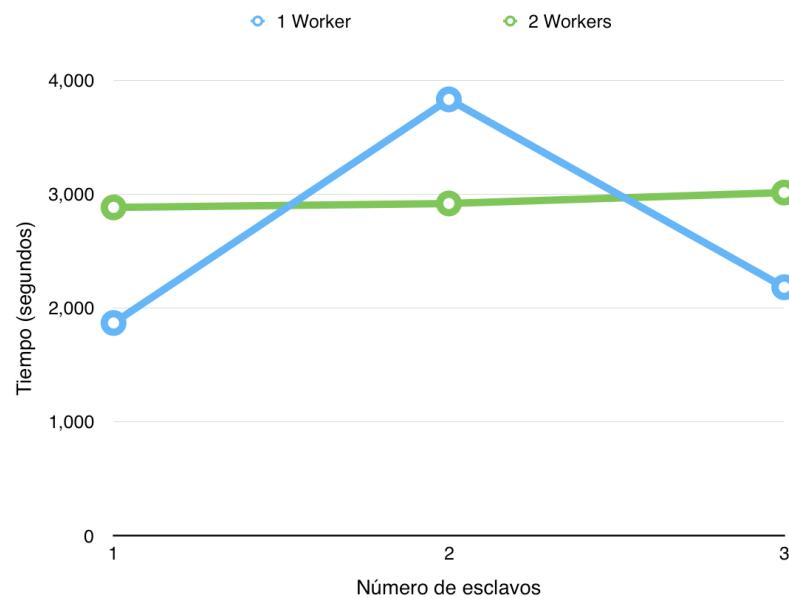
- 1 Maestro, 1 Esclavo
- 1 Maestro, 2 Esclavos
- 1 Maestro, 3 Esclavos

Para cada una de estas iteraciones, se van a probar las siguientes configuraciones:

- 1 Worker por esclavo, 10GB
- 2 Workers por esclavo, 5 GB cada uno (10GB en total)



**Figura 13. Tiempo de lectura del archivo**



**Figura 14. Tiempo de computación de clústeres**

Los resultados son decepcionantes. Se esperaba encontrar un equilibrio entre la latencia introducida por la comunicación entre los nodos y la capacidad de computo añadida, pero no hemos superado dicho umbral y las ejecuciones más rápidas han sido las que se han realizado con menos esclavos y con menos trabajadores. Estudiaremos cómo mejorar los resultados en el siguiente capítulo, conclusiones.

Bloque IV

## **CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO**

# **Capítulo 6**

## **Conclusiones y futuras líneas de trabajo**

### **6.1 Mejoras en el clúster**

Ha sido interesante ver cómo el rendimiento del clúster Spark decrecía a medida que añadíamos esclavos y workers. Muy probablemente esto es fruto de la latencia de usar NFS como sistema de archivos distribuido. En el futuro se debería considerar utilizar un sistema de archivos distribuido alternativo. Apache Spark recomienda usar HDFS, el sistema de archivos integrado en Hadoop.

### **6.2 Validación del modelo**

El siguiente paso sería hacer un estudio para validar que el modelo incrementa los beneficios de los taxistas que lo usan.

La forma más sencilla desde el punto de vista estadístico es hacer un estudio de campo, pero requiere de muchos recursos para encontrar los candidatos, realizar las pruebas y recoger los datos. Con una muestra lo suficientemente grande, dividida en dos

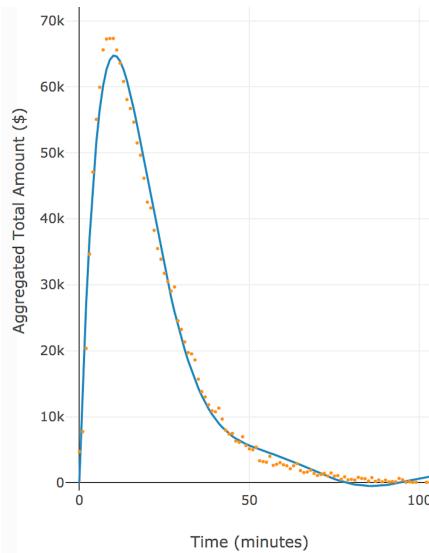
grupos, un grupo de control y otro de intervención, podríamos ver si produce efecto. Y si lo produce, podemos cambiar los parámetros y las funciones de las puntuaciones y repetir el experimento dividiendo más grupos.

La validación del modelo En el capítulo Estado del Arte, vimos dos publicaciones científicas que habían usado dicho método. Uno de ellos usó un método híbrido, añadiendo al estudio de campo simulaciones de trazas con fuentes de datos diferentes.

## 6.3 Mejoras del modelo

Si recordamos la Figura 5. Correlación del dinero total producido del Capítulo 3 Correlaciones de características, tomamos el coeficiente de la regresión lineal como indicador de puntuación, pero las curvas están lejos de ser lineales. Podemos mejorar el modelo buscando las funciones de las curvas y combinándolas en una nueva función de puntuación de traza.

En la Figura 15. Polinomio de grado 9 y curva agregada de tiempo vemos un polinomio de grado 9 encajaría muy bien para predecir la característica “tiempo”.



**Figura 15. Polinomio de grado 9 y curva agregada de tiempo**

## 6.4 Riesgos del modelo

El uso masivo del recomendador en si mismo podría impactar en los taxis de varias formas:

- Saturación, al ir tantos taxis para el mismo sitio pueden provocar tráfico repentino en la zona.

- Las zonas recomendadas podrían dejar de ser rentables al quedarse sin clientes a los que coger debido al exceso de taxistas.
- Otras zonas podrían sufrir una carencia de taxistas debido a que no aparezcan recomendadas en el sistema.

Bloque V

## **PLANIFICACIÓN Y PRESUPUESTO**

# **Capítulo 5**

## **Planificación**

### **7.1 Fases de desarrollo**

La duración de la realización del proyecto ha sido alrededor de tres meses. A continuación se resumen las fases en las que se ha dividido el desarrollo y el tiempo empleado para cada una de ellas:

#### **1. Planteamiento de la necesidad y descripción del problema**

En esta fase se pretende centrar la idea general del proyecto. Se plantea la necesidad de probar nuevas tecnologías y se acuerda cuáles se van a utilizar en concreto para el desarrollo.

- Tiempo estimado: 5 días.
- Participantes: Tutor y desarrollador del proyecto.

#### **2. Estudio de las tecnologías utilizadas**

Análisis de las tecnologías que se van a emplear. Se hizo un estudio de Spark.

- Tiempo estimado: 20 días.
- Participantes: Tutor y desarrollador del proyecto.

#### **3. Análisis de los datos**

Se exploran los datos y se evalúa la viabilidad.

- Tiempo: 30 días.
- Participantes: Tutor y desarrollador del proyecto.

#### 4. Diseño del sistema

Fase en la que se definen los requisitos del sistema y la arquitectura de componentes que lo forman, así como la interacción entre ellos y la elección del diseño final.

- Tiempo: 10 días.
- Participantes: Tutor y desarrollador del proyecto.

#### 5. Pruebas y resultados

Pruebas de validación del correcto funcionamiento del sistema en diferentes entornos. También se miden tiempos de ejecución y se contrastan los resultados.

- Tiempo: 10 días.
- Participantes: Tutor y desarrollador del proyecto.

#### 6. Redacción de la memoria

Documentación final de todo el trabajo realizado en el desarrollo completo del proyecto.

- Tiempo: 5 días.
- Participantes: Tutor y desarrollador del proyecto.

# Capítulo 8

## Presupuesto

### 8.1 Medios Empleados

Los recursos empleados en el desarrollo del proyecto son los siguientes:

#### Recursos humanos

- 1 Ingeniero Senior
- 1 Ingeniero Senior

#### Recursos materiales

- 1 PC portátil de gama alta
- Microsoft Office 365

#### Otros recursos

- Conexión a Internet durante 3 meses
- 4 Servidores de gama alta (Cluster Spark)
- VPS Digital Ocean (Aplicación Web)
- Software
  - Django
  - Nginx
  - PostgreSQL, PostGIS
  - SciPy
  - Sklearn

- o Apache Spark

## 8.2 Presupuesto del trabajo

1. Autor: Jorge Barata González
2. Departamento: Ingeniería Informática
3. Descripción del proyecto
  - Título: Sistema Recomendador de Taxis para Big Data
  - Duración: 3 meses
  - Tasa de costes indirectos: 20%
4. Presupuesto total del proyecto 13920€
5. Desglose del presupuesto, costes directos:

En la realización del proyecto se han visto involucradas dos ingenieros senior: uno con una dedicación de media jornada y otro con dedicación del 1% de la jornada.

Tarea	Días
Planteamiento de la necesidad y descripción del problema	5
Estudio de las tecnologías utilizadas	20
Análisis de los datos	30
Diseño del sistema	10
Pruebas y resultados	10
Redacción de la memoria	5
<b>Total Días</b>	80
<b>Coste por día (€)</b>	120
<b>Coste total (€)</b>	9600

**Tabla 1. Costes personales**

Concepto	Coste (€)	Dedicación (meses)	Periodo de depreciación (meses)	Coste imputable (€)
PC Portátil	2000	3	48	125
			<b>Total</b>	<b>125</b>

**Tabla 2. Costes materiales**

Concepto	Coste (€)
Material	2000
Personal	9600
<b>Total</b>	<b>11600</b>

**Tabla 3. Coste total**

Bloque VI

## **ANEXOS**

# Anexo A

## Capturas Aplicación Web

Podemos visitar la aplicación en <http://pfc.jorgebg.com>.

De cara al usuario final, la web sólo tiene una página (Figura 16. Página principal). Se inicia automáticamente con la hora y día actuales, y un punto arbitrario de la ciudad de Nueva York, junto con los lugares recomendados. La prioridad de las recomendaciones las dan las letras en los iconos, ordenadas alfabéticamente.

Pueden cambiarse los parámetros y consultar las recomendaciones haciendo click en “Submit”. Al pinchar en cualquier punto de la ciudad, actualiza las coordenadas y recomendaciones al lugar pinchado.

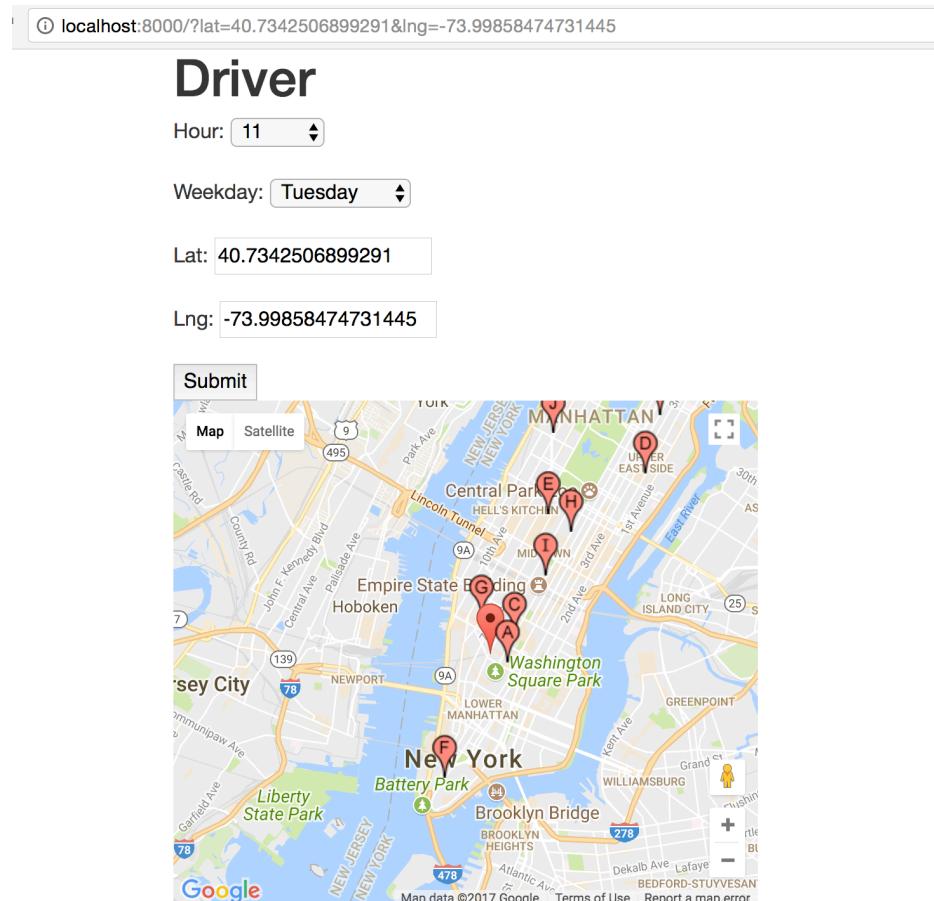
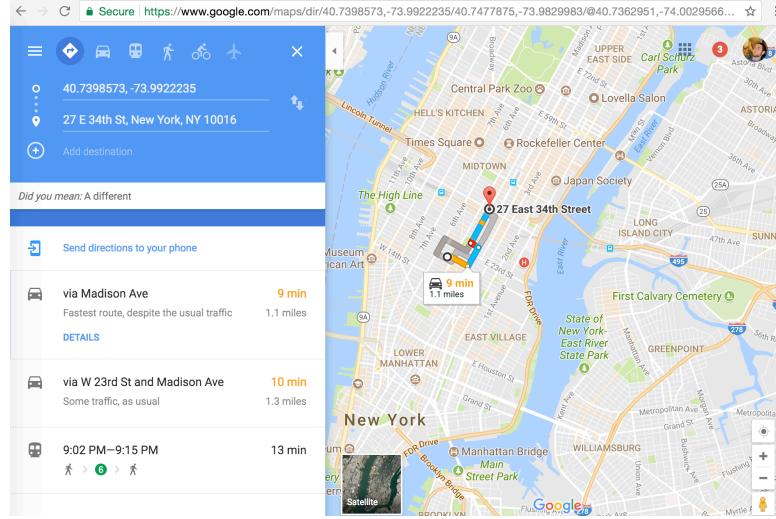


Figura 16. Página principal

Al pinchar en cualquiera de las recomendaciones del mapa, se abre Google Maps y se calcula la ruta desde la ubicación actual (Figura 17. Trayecto en Google Maps).



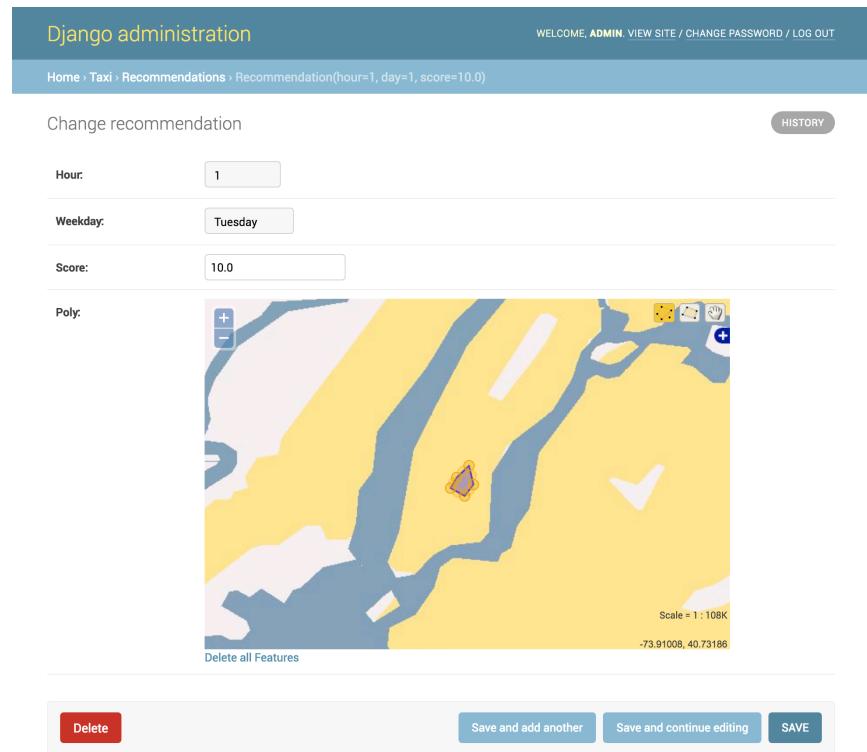
**Figura 17. Trayecto en Google Maps**

En el panel de administración se puede ver y ordenar un listado de las recomendaciones (Figura 18. Administración de recomendaciones: listado).

Django administration					WELCOME, ADMIN   VIEW SITE / CHANGE PASSWORD   LOG OUT
Home > Taxi > Recommendations					<a href="#">ADD RECOMMENDATION +</a>
Select recommendation to change					
Action: <input type="button" value="-----"/> Go 0 of 100 selected					
Action	WEEKDAY	HOUR	SCORE	POLY	
<input type="checkbox"/>	Saturday	23	101.843855261803	SRID=4326;POLYGON ((-73.95613098144531 40.77148818969727, -73.94046783447266 40.79325103759766, -73.95626083740234 40.76332092285156, -73.97148132324219 40.7512092590332, -73.98944854736328 40.72624588012695, -73.99276733398438 40.72146606445312, -73.98470306396484 40.7323112487793, -73.95613098144531 40.77148818969727))	
<input type="checkbox"/>	Wednesday	12	100.095680117607	SRID=4326;POLYGON ((-73.97147369384766 40.76659774780273, -73.99605560302734 40.73271560668945, -73.99496459960938 40.7341461184604, -73.9866943359375 40.74520111083984, -73.95414733886719 40.78987121582031, -73.9571838789062 40.78594589233998, -73.96219635009766 40.77922821044922, -73.97147369384766 40.76659774780273))	
<input type="checkbox"/>	Sunday	16	92.6398453712463	SRID=4326;POLYGON ((-73.95400238037109 40.80303573608398, -73.96623229980469 40.80474090576172, -73.98095703125 40.78250503540039, -74.00365447998047 40.73214340209961, -74.00748443603516 40.70848846435547, -74.0074462890625 40.70796203613281, -73.93388885498047 40.71796798706055, -73.98865509033203 40.72250747680664, -73.95400238037109 40.80303573608398))	
<input type="checkbox"/>	Sunday	13	87.920200586319	SRID=4326;POLYGON ((-73.97769165039062 40.746655151367188, -73.98004150390625 40.74308119921875, -73.96094512939453 40.7691622436523, -73.94911193847656 40.78538131713867, -73.9542922936328 40.77860260009766, -73.95880126953125 40.77245330810547, -73.97769165039062 40.74655151367188))	
<input type="checkbox"/>	Wednesday	7	87.5019159317017	SRID=4326;POLYGON ((-73.95389556884766 40.80619049072266, -73.95555877685547 40.80435943603516, -73.97887420654297 40.77258682250977, -73.99420166015625 40.75146665844727, -73.9993676025391 40.74417877197266, -74.00064086914062 40.7423095703125, -73.9852538037109 40.76321029663086, -73.95389556884766 40.80619049072266))	

**Figura 18. Administración de recomendaciones: listado**

Al pinchar en cualquiera de las entradas, podemos modificar los parámetros de la recomendación, incluyendo los vértices del polígono Figura 19. Administración de recomendaciones: edición:



**Figura 19. Administración de recomendaciones: edición**

# Anexo B

## Propiedades completas de las trazas

VendorID	código que indica el proveedor TPEP que proporcionó el registro. 1 = Creative Mobile Technologies, LLC; 2 = VeriFone Inc.
tpep_pickup_datetime	Fecha y hora en que se activó el medidor.
tpep_dropoff_datetime	Fecha y hora en que se desconectó el medidor.
Passenger_count	Número de pasajeros en el vehículo.
Trip_distance	Distancia de viaje transcurrida en millas informada por el taxímetro.
Pickup_longitude	Longitud en la que el medidor estaba ocupado.
Pickup_latitude	Latitud en la que el medidor estaba ocupado.
RateCodeID	El código de tasa final en vigor al final del viaje. 1 = Tasa estándar 2 = JFK 3 = Newark 4 = Nassau o Westchester 5 = Precio negociado 6 = Paseo en grupo
Store_and_fwd_flag	Este indicador indica si el registro de viaje se mantuvo en la memoria del vehículo antes de enviar al vendedor, también conocido como "store and forward" porque el vehículo no tenía una conexión al servidor. Y = almacenar y reenviar viaje N = no almacenar y pasar un viaje hacia adelante
Fare_amount	La tarifa de tiempo y distancia calculada por el contador.
extra	Diversos extras y recargos. Actualmente, esto sólo incluye los \$ 0.50 y \$ 1 hora punta y cargos de nocturnidad.
MTA_tax	\$ 0.50 El impuesto MTA que se activa automáticamente basado en el contador en uso.
Improvement_surcharge	\$ 0,30 recargo de mejora de viajes evaluados en la bajada de la bandera.
Tip_amount	Propina. Sólo tarjeta de crédito.
Tolls_amount	Cantidad total de todos los peajes pagados en el viaje
Total_amount	La cantidad total cobrada a los pasajeros. No incluye consejos de efectivo.

**Tabla 4. Propiedades completas de las trazas**

# Anexo C

## Código

A continuación, se explican los diferentes módulos que se programaron para el desarrollo del proyecto.

Se puede consultar en <https://github.com/jorgebg/taxi-recommendation-system>.

### **download\_raw\_data.sh**

Descarga los datos necesarios:

- Las trazas de los taxis
- Los bordes de los distritos en GeoJSON

### **fabfile.py**

Gestiona el cluster Spark remotamente. Las acciones disponibles son:

- config
  - Configura el cluster. Admite parámetros para las diversas configuraciones utilizadas en la sección Pruebas y Resultados.
- cluster
  - Inicia o para el cluster
- info
  - Muestra el modelo de procesador y la memoria total y libre.
- notebook
  - Inicia un notebook con Jupyter
- pyspark
  - Inicia una shell de pyspark
- venv
  - Instala las dependencias de Python usadas en el proyecto
- ping
  - Comprueba la latencia entre máquinas

### **lib/**

Incluye las librerías compartidas por los diversos módulos del proyecto:

### **bilos.py**

Lee el fichero GeoJSON con los bordes de los distritos de NYC y los transforma en polígonos para la librería Shapely de Python.

### **plotly.py**

Configuraciones usadas recurrentemente en Plotly.

### **mapbox.py**

Configuraciones usadas recurrentemente para la integración de Plotly en Mapbox.

**spark.py**

Configuración de Spark.

**process.py**

Filtrado y transformación del dataset.

**timer.py**

Librería de mediciones de tiempo.

**project/**

Contiene la aplicación web desarrollada en Django

**remote-shell.sh**

Comprime la librería lib para ser enviada a Spark y abre una shell pyspark.

**show\_boros.py**

Muestra los bordes de los distritos para comprobar que cargamos el GeoJSON correctamente.

**show\_process.py**

Muestra con precisión qué cantidad de registros se están filtrando por lib/process.py

**show\_correlations.py**

Muestra las correlaciones entre pago total y distancia, tiempo y pasajeros.

**show\_clusters.py**

Muestra que se pueden computar clusters para diferentes grupos de hora, y que cada resultado es distinto.

**compute\_clusters.py**

Es el programa final que computa los clusters para cada hora del día y de la semana y guarda los resultados en result.pickle

**result.pickle**

Los polígonos de los cluster computados junto con la hora del dia, de la semana, y la puntuación. Se serializan usando pickle, parte de la librería estándar de Python.

# Anexo D

## Normativa y Marco regulador

En relación al uso de las nuevas tecnologías, no existen inconvenientes legales directos. Sin embargo, pueden existir problemas de violación de la privacidad de las personas. De tal forma que estos problemas son tratados por los organismos correspondientes.

Toda aquella información que posibilite la identificación directa o indirectamente de cualquier persona se considera un dato de carácter personal. Por esto, el derecho fundamental a la protección de datos consiste en otorgar al ciudadano la capacidad de disponer, controlar y decidir sobre sus datos personales.<sup>24</sup>

A nivel nacional es la Agencia Española de Protección de datos es la autoridad de control independiente encargada de velar por el cumplimiento de la normativa que hace referencia a la protección de datos. Además, asegura y protege el derecho fundamental a la protección de datos personales.

La Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de carácter personal tiene como objeto garantizar y proteger todo lo relacionado con el tratado de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas así como el honor e intimidad personal y familiar<sup>25</sup>. Y los derechos que se engloban en esta ley son<sup>26</sup>:

- Derecho de información: Cuando se procede a la recogida de datos el interesado tiene que ser informado.
- Derecho de acceso: El interesado puede conocer y obtener de forma gratuita los datos de carácter personal que van a ser tratados.
- Derecho de rectificación: Se permite la corrección de errores o la modificación de datos que sean inexactos o incompletos.
- Derecho de cancelación: Se puede suprimir los datos considerados inadecuados o excesivos.
- Derecho de oposición: Derecho del afectado a que no puedan ser tratados sus datos personales.

En el ámbito europeo, se han fortalecido los derechos de los ciudadanos adaptando las reglas para los negocios a consecuencia de la en la era digital en la que nos encontramos. Se ha considerado que las empresas no pueden compartir datos de los usuarios sin una previa autorización en la que ofrecen su consentimiento al intercambio de datos. Y en el caso de violación de los derechos de privacidad de sus usuarios la multa

a la que tendrán que enfrentarse las empresas puede ascender al 4% de los ingresos de la compañía.<sup>27</sup>

# Referencias

- 
- <sup>1</sup> Flintrock, a command-line tool for launching Apache Spark clusters. URL: <https://github.com/nchammas/flintrock> [19 de Septiembre del 2019]
- <sup>2</sup> Yellow Cab, Long a Fixture of City Life, Is for Many a Thing of the Past. URL: <https://www.nytimes.com/2017/01/15/nyregion/yellow-cab-long-a-fixture-of-city-life-is-for-many-a-thing-of-the-past.html> [19 de Septiembre del 2019]
- <sup>3</sup> TLC Trip Record Data. URL: [http://www.nyc.gov/html/tlc/html/about/trip\\_record\\_data.shtml](http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml) [19 de Septiembre del 2019]
- <sup>4</sup> PostGIS, Spatial and Geographic objects for PostgreSQL. URL: <http://postgis.net/> [19 de Septiembre del 2019]
- <sup>5</sup> TLC Trip Record Data, NYC Taxi & Limousine Commission. URL: [http://www.nyc.gov/html/tlc/html/about/trip\\_record\\_data.shtml](http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml) [19 de Septiembre del 2019]
- <sup>6</sup> Nicholas Jing Yuan, Yu Zheng, Liuhang Zhang, Xing Xie. “T-Finder: A Recommender System for Finding Passengers and Vacant Taxis”. 2010. URL: <https://www.microsoft.com/en-us/research/publication/t-finder-a-recommender-system-for-finding-passengers-and-vacant-taxis/> [19 de Septiembre del 2019]
- <sup>7</sup> Meng Qu Rutgers, Hengshu Zhu, Junming Liu, Guannan Liu, Hui Xiong. “A cost-effective recommender system for taxi drivers”. 2014. URL: <http://dl.acm.org/citation.cfm?id=2623668> [19 de Septiembre del 2019]
- <sup>8</sup> HubCab. URL: <http://hubcab.org/#12.00/40.7257/-73.8915> [19 de Septiembre del 2019]
- <sup>9</sup> Apache Spark, Wikipedia. URL: [https://en.wikipedia.org/wiki/Apache\\_Spark](https://en.wikipedia.org/wiki/Apache_Spark) [19 de Septiembre del 2019]
- <sup>10</sup> Machine Learning Library Guide, Spark. URL: <https://spark.apache.org/docs/latest/ml-guide.html> [19 de Septiembre del 2019]
- <sup>11</sup> Apache Hadoop, Github URL: <https://github.com/apache/spark> [19 de Septiembre del 2019]
- <sup>12</sup> MapReduce, Wikipedia. URL: <https://en.wikipedia.org/wiki/MapReduce> [19 de Septiembre del 2019]
- <sup>13</sup> Geospatial Queries, MongoDB, URL: <https://docs.mongodb.com/manual/geospatial-queries/> [19 de Septiembre del 2019]
- <sup>14</sup> PostGIS, Wikipedia URL: <https://en.wikipedia.org/wiki/PostGIS> [19 de Septiembre del 2019]
- <sup>15</sup> Front and back ends, Wikipedia. URL: [https://en.wikipedia.org/wiki/Front\\_and\\_back\\_ends](https://en.wikipedia.org/wiki/Front_and_back_ends) [19 de Septiembre del 2019]
- <sup>16</sup> Model-View-Controller, Wikipedia. URL: <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller> [19 de Septiembre del 2019]
- <sup>17</sup> SciPy, URL: <https://www.scipy.org/> [19 de Septiembre del 2019]

---

<sup>18</sup> Borough Boundaries, CitiOFNewYork, URL: <https://data.cityofnewyork.us/City-Government/Borough-Boundaries/tqmj-j8zm/data> [19 de Septiembre del 2019]

<sup>19</sup> Shapely, Python library for geometric analysis, URL:  
<https://pypi.python.org/pypi/Shapely> [19 de Septiembre del 2019]

<sup>20</sup> Top-ranked Papers in "Data Mining", Microsoft Research.  
[https://web.archive.org/web/20100421170848/http://academic.research.microsoft.com/CS\\_Directory/paper\\_category\\_7.htm](https://web.archive.org/web/20100421170848/http://academic.research.microsoft.com/CS_Directory/paper_category_7.htm) [19 de Septiembre del 2019]

<sup>21</sup> pickle, Python Documentation, URL: <https://docs.python.org/3/library/pickle.html> [19 de Septiembre del 2019]

<sup>22</sup> Droplet, DigitalOcean. URL: <https://www.digitalocean.com/products/compute/> [19 de Septiembre del 2019]

<sup>23</sup> Google Maps Developers, Google. URL: <https://developers.google.com/maps/> [19 de Septiembre del 2019]

<sup>24</sup> Agencia española de protección de datos. Tu derecho fundamental a la protección de datos. URL:

<https://www.agpd.es/portalwebAGPD/CanalDelCiudadano/derechos/index-ides-idphp.php> [19 de Septiembre del 2019]

<sup>25</sup> Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal. URL:

[http://noticias.juridicas.com/base\\_datos/Admin/lo15-1999.t1.html#t1](http://noticias.juridicas.com/base_datos/Admin/lo15-1999.t1.html#t1) [19 de Septiembre del 2019]

<sup>26</sup> Agencia española de protección de datos. Principales derechos. URL:  
[https://www.agpd.es/portalwebAGPD/CanalDelCiudadano/derechos/principales\\_derchos/index-ides-idphp.php](https://www.agpd.es/portalwebAGPD/CanalDelCiudadano/derechos/principales_derchos/index-ides-idphp.php) [19 de Septiembre del 2019]

<sup>27</sup> El país. La UE aprueba la ley de protección de datos, bloqueada desde 2013. URL:  
[http://internacional.elpais.com/internacional/2015/12/15/actualidad/1450208377\\_400556.html](http://internacional.elpais.com/internacional/2015/12/15/actualidad/1450208377_400556.html) [19 de Septiembre del 2019]