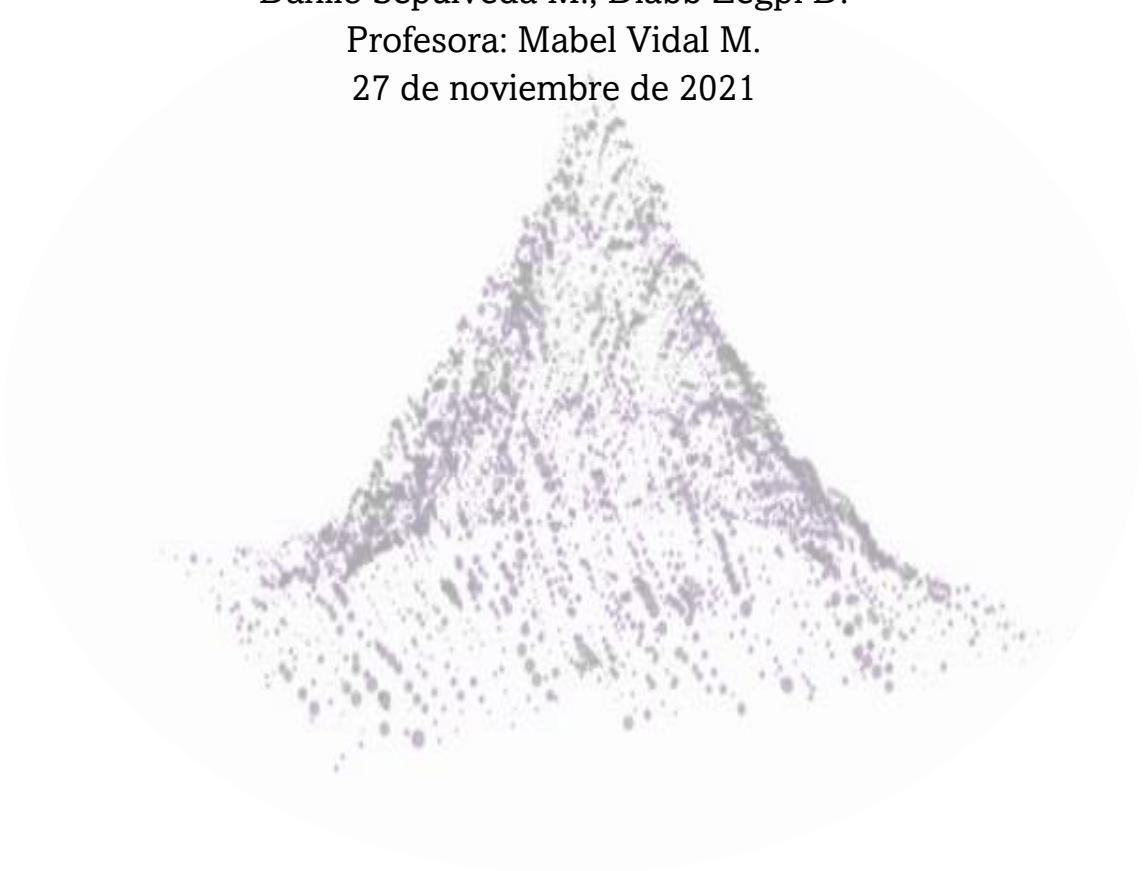
UNIVERSIDAD SAN SEBASTIAN

Tarea 2 de Big Data

Implementación de una Arquitectura de Big Data

Danilo Sepúlveda M., Diabb Zegpi D.
Profesora: Mabel Vidal M.
27 de noviembre de 2021



Índice de contenidos

Descripción del problema 3
Descripción de los datos 3
Exploración de los datos 4
Propuesta de solución 7

Índice de figuras

Figura 1: Serie de tiempo de la cantidad de viajes mensuales en Chicago 4
Figura 2: Cambio en la popularidad de los colores de acuerdo con el año 5
Figura 3: Curvas de densidad para cantidad de viajes 6

Descripción del problema

El problema está definido en el contexto de las cuatro compañías Proveedoras de Redes de Transporte (TNPS en inglés) en Chicago, servicio que ofrecen mediante sus aplicaciones para teléfonos móviles. Específicamente, el problema consiste en el desconocimiento de los gustos y preferencias de los clientes de estas aplicaciones (Uber, Lyft y otras) por los vehículos de los conductores. Se estima que conocer esta información permitiría a los respectivos departamentos de marketing, la toma de decisiones efectivas para mejorar el rédito que obtienen de sus clientes.

Las preguntas directrices que conducirán el desarrollo de la metodología son

- ¿existe una tendencia de vehículos populares entre los años 2015 y 2019?,
- ¿existen relaciones entre el rendimiento de un vehículo y la cantidad de viajes que realiza?, y
- ¿qué estrategia de marketing tiene potencial para aumentar las ventas de las compañías de viajes compartidos?

El conjunto de datos escogido para desarrollar este proyecto está compuesto por la cantidad de viajes compartidos que realizan los vehículos registrados en Chicago, y está disponible en kaggle. Un recurso complementario son los viajes realizados por estos vehículos, los que están disponibles en el portal de datos de Chicago.

Descripción de los datos

Los viajes compartidos en Chicago por vehículo son reportados mensualmente, como se indica en las columnas `REPORTED_YEAR` y `REPORTED_MONTH`. Por cada mes y vehículo registrado, está disponible la siguiente información:

1. `REPORTED_YEAR`: el año en que el vehículo fue reportado.
2. `REPORTED_MONTH`: el mes en que el vehículo fue reportado.
3. `STATE`: el estado de la placa de matrícula.
4. `MAKE`: la marca del vehículo.
5. `MODEL`: el modelo del vehículo.
6. `COLOR`: el color del vehículo.
7. `MODEL_YEAR`: el año del modelo del vehículo.
8. `NUMBER_OF_TRIPS`: número de viajes realizados en el mes.
9. `MULTIPLE_TNPS`: si el vehículo ha sido reportado por múltiples TNPS.

Exploración de los datos

El dataset de *ridesharing* tiene 9 columnas y 1.376.115 filas. 4 de las columnas son categóricas, 4 numéricas y 1 booleana. Ninguna columna registra valores faltantes, porque el dataset fue preprocesado antes de ser compartido en kaggle.

En general, los vehículos del dataset fueron registrados en cada uno de los 51 estados del país, con predominancia de Ilinois (96%). Todos los vehículos provienen de 46 marcas y 630 modelos, siendo la marca más popular Toyota (25,5%) y el modelo más popular el Toyota Camry (9,37%). Cada vehículo se identifica con uno de 137 colores, de los que el más común es el negro (32,6%).

La mayor cantidad de viajes fue observada en el año 2019, con 371.909 (27%) viajes en total, cuyo mes con más viajes es marzo, con 32.572 (2,37%). En la Figura 1 no se observa que la cantidad de viajes mensuales tenga un componente estacional, como también es evidente el efecto que tuvo el coronavirus en 2020. Adicionalmente, un modelo no paramétrico de suavizamiento con ventana móvil se ajusta a los datos, para visualizar el efecto del tiempo en la evolución de la demanda por TNPS.

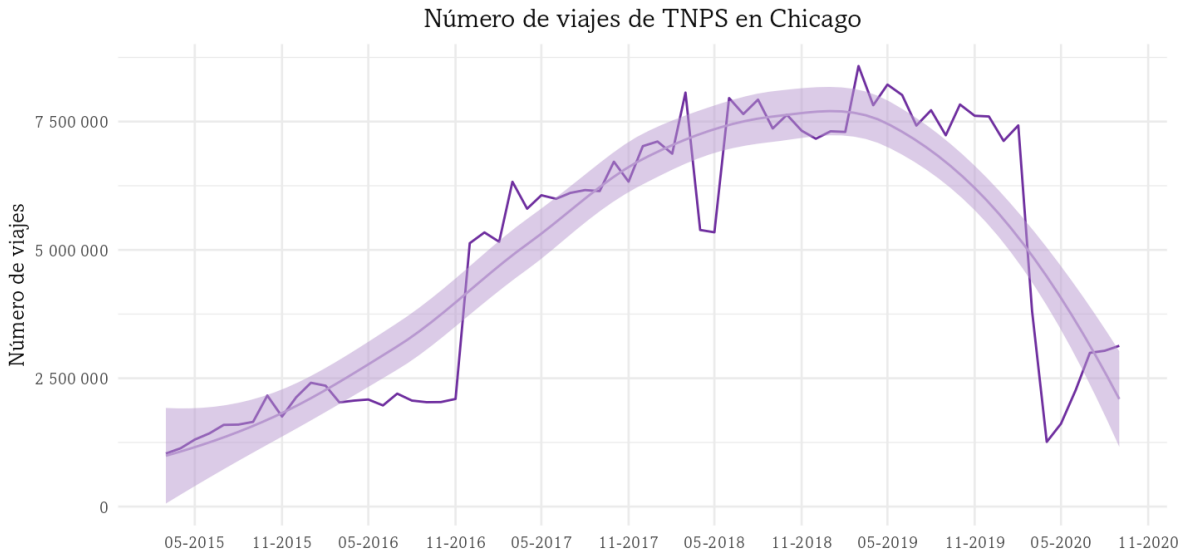


Figura 1: Serie de tiempo de la cantidad de viajes mensuales en Chicago

Paralelamente, puede investigarse la existencia de una tendencia entre los colores de los vehículos y su popularidad, medida en cantidad de viajes, para cada año.

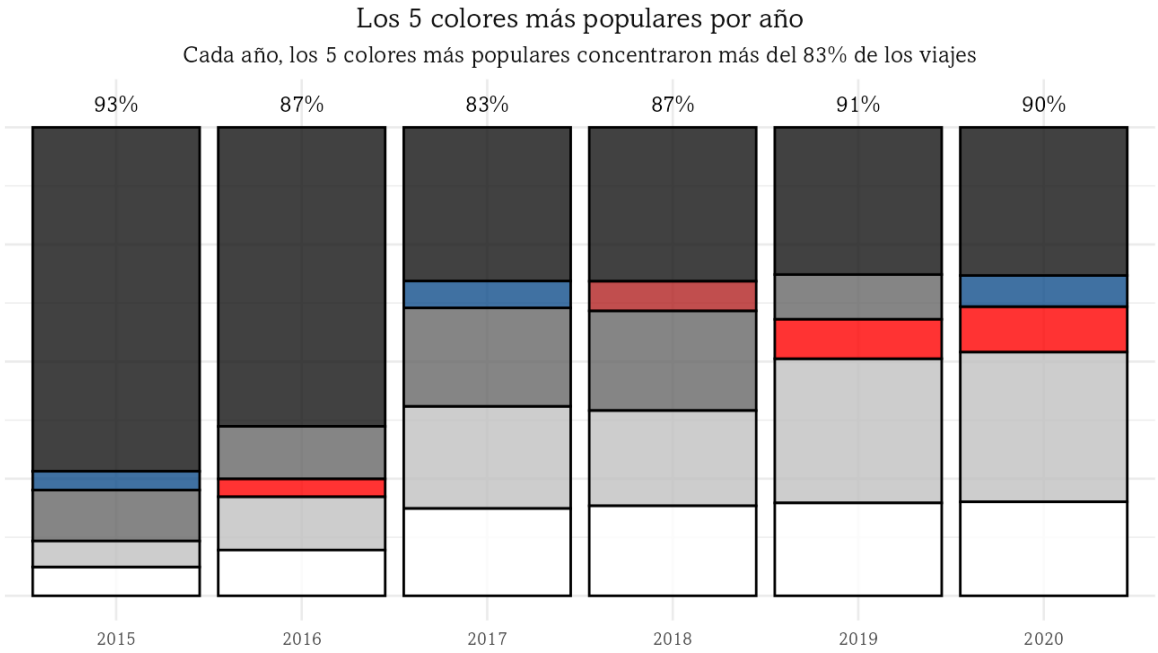


Figura 2: Cambio en la popularidad de los colores de acuerdo con el año

En la Figura 2 se aprecia la disminución de la popularidad del negro. Esto puede deberse al menos a estos dos factores: hay menos vehículos registrados con el color negro o bien, los clientes prefieren menos este color. Cabe destacar que los cinco colores más populares por año concentran más del 83% de los viajes anuales, indicando que la variable `COLOR` tiene gran poder predictivo sobre la demanda por TNPS.

Asimismo, la distribución de la cantidad de viajes se distingue entre usuarios registrados en una o más TNPS. La Figura 3 evidencia las distinciones entre las curvas de densidad de estos usuarios: aquellos usuarios con inscripción en solo una compañía tienden a realizar menos viajes mensuales, con una media de 223, mediana de 192, curtosis de 4,86 y coeficiente de asimetría igual a 1,33. En comparación, la curva de los usuarios inscritos en más de una compañía tiene una cola más gruesa, indicando así que estos usuarios realizan una mayor cantidad de viajes mensuales y tomando la ocupación con más profesionalismo. Las métricas de tendencia central y asimetría son: media igual a 286, mediana de 237, curtosis de 5,15 y coeficiente de asimetría de 1,47.

Ambas curvas presentan asimetría positiva, lo que puede aumentar el error estándar de los parámetros estimados de un eventual modelo de regresión. Por lo tanto, previo a aplicar métodos de machine learning, se recomienda una transformación matemática que centralice las distribuciones (logaritmo o raíz cuadrada).

Respecto de la distribución de los datos en torno a su media, ambas distribuciones son platicúrticas, con gran varianza. Esta característica también puede ser mitigada mediante una transformación matemática.

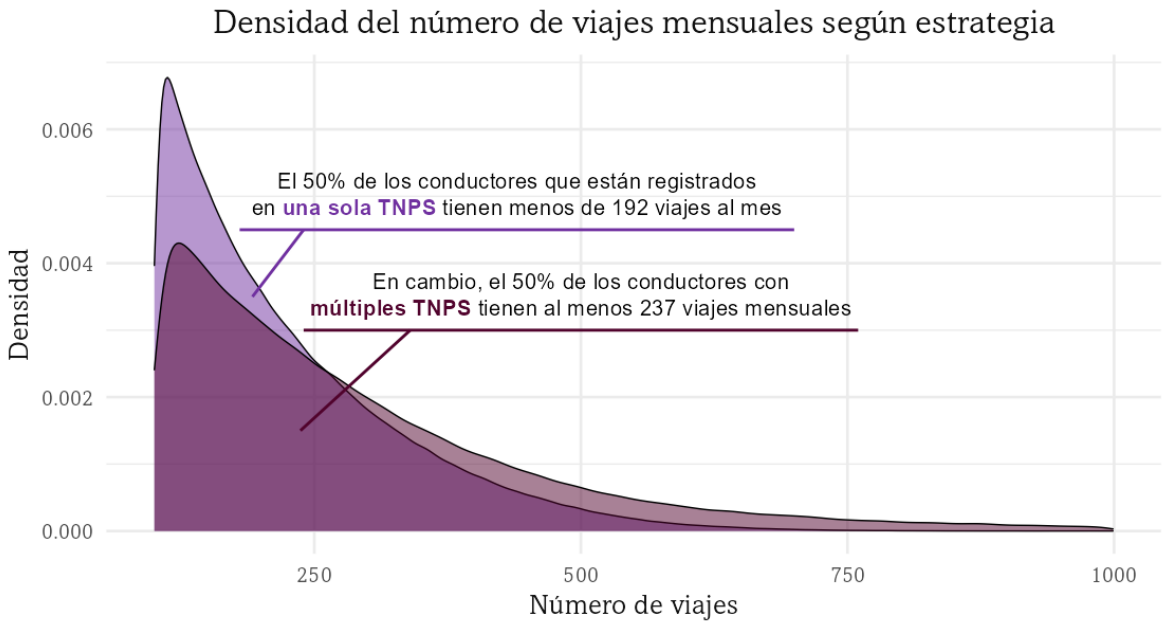


Figura 3: Curvas de densidad para cantidad de viajes

El código completo de la limpieza de datos y creación de los gráficos se encuentra en este repositorio de github.

Propuesta de solución

La estrategia de Big Data escogida para dar forma a la propuesta de solución es *"predicción de la demanda de usuarios para compañías de viajes compartidos"*, utilizando métodos de pronóstico de aprendizaje automático implementados en Apache Spark.

Las condiciones necesarias para desarrollar la solución se resumen en identificar los hábitos de consumo de los clientes (identificar las variables con poder predictivo), levantar la plataforma Apache Spark con el lenguaje de programación R y albergar los datos en ella, ajustar modelos de predicción para la demanda de usuarios de acuerdo con características de vehículos de interés y, por último, seleccionar el modelo con mejor métrica de desempeño y analizar los resultados obtenidos. De esta manera, se podrá presentar información relevante para formular una estrategia de marketing con gran potencial de efectividad.

Para el desarrollo de la solución será necesario aplicar técnicas de feature engineering y machine learning, con tal de obtener el máximo provecho de las variables disponibles y potenciales. Además, también serán útiles métodos de extracción de datos, puesto que en internet hay gran cantidad de información disponible sobre cualidades de cada modelo y marca de vehículo específica. Finalmente, la implementación del modelo de big data requerirá el aprendizaje de estas tecnologías, objetivo que se espera alcanzar con el curso de la asignatura.

Implementación de arquitectura de Big Data

Se escogió la arquitectura Apache Spark, mediante su interfaz para R (spark.rstudio.com), accesible a través del paquete `sparklyr`. *"Apache Spark es un motor multilenguaje para ejecutar ingeniería de datos, ciencia de datos, y aprendizaje automático en clústeres o máquinas de un solo nodo"* (spark.apache.org).

La implementación consiste en levantar una conexión a bases de datos mediante sparklyr, cargar datos a esta base usando la conexión, realizar transformaciones de datos utilizando un *pipeline* de spark y entrenar un modelo de aprendizaje automático en la arquitectura.

El dataset resultante de las transformaciones aplicadas tiene dimensiones (1.376.115, 46); hay 45 variables predictoras y 1 variable respuesta: la cantidad de viajes que realiza un conductor por mes, en escala logarítmica.

Con el fin de aprovechar la arquitectura master-slave de Spark, se escogió el algoritmo de Random Forest, debido a su facilidad de paralelización. La implementación del algoritmo en sparklyr se encuentra en la función `ml_random_forest_regressor()`.

En el siguiente script se muestra el flujo de trabajo completo de levantamiento de una conexión con Spark, preprocesamiento en la arquitectura de Big Data, entrenamiento y prueba de un modelo y, finalmente, descarga de los datos a la memoria de trabajo.

```
library(tidyverse)
library(tidymodels)
library(here)
library(sparklyr)
# Importar los datos
clean_data <- read_csv(file = here("Big Data", "Data", "clean_data.csv"))
%>%
  janitor::clean_names()

# Crear conexión con Apache Spark y cargar una tabla
sc <- spark_connect(master = "local")
cars_tbl <- copy_to(sc, clean_data)

# spark_dplyr <- cars_tbl %>%
#     muchas transformaciones...

# Flujo de machine learning
cars_pipeline <- ml_pipeline(sc) %>%
  ft_dplyr_transformer(tbl = spark_dplyr) %>%
  ft_r_formula(number_of_trips ~ .) %>%
  ml_random_forest_regressor()

# Separación de entrenamiento y prueba
spark_split <- sdf_random_split(cars_tbl, training = .6, testing = .4)

# Entrenamiento y prueba
spark_model <- ml_fit(cars_pipeline, spark_split$training)
spark_pred <- ml_transform(spark_model, spark_split$testing)

# Recolectar los resultados en la memoria RAM
results <- spark_pred %>%
  select(number_of_trips, prediction,
         starts_with("make_"),
         starts_with("color_")) %>%
  collect()
```

Discusión de resultados

El procesamiento de datos en el entorno simulado de Big Data es lento, en comparación con el procesamiento local en una sesión de RStudio. Se piensa que esto se debe a que el entorno simulado no cuenta con las bondades de una arquitectura real de Big Data: hardware especializado. Sin embargo, esta aparente ventaja del procesamiento local puede ser eclipsada en un escenario en que la cantidad de datos es masiva y remota, sin poder ser cargados en la memoria RAM de manera eficiente.

Conclusiones

El enfoque de trabajo con arquitecturas de Big Data puede ser provechoso si se cuenta con tres condiciones básicas:

1. Hardware especializado (clusters),
2. gran cantidad de datos y
3. la necesidad de procesar todos los datos.

Las dos primeras son evidentes, pero la tercera merece una explicación. Si un científico de datos cuenta con una cantidad abrumadora de datos, perfectamente puede extraer una o varias muestras y ajustar modelos sobre ellas. Por tanto, el volumen de datos no es razón suficiente que justifique la utilización de Spark, por ejemplo. La excepción está en el caso de uso; si una compañía desarrolla un buscador inteligente, se verá en la obligatoriedad de indexar una gran cantidad de sitios web, preprocesarlos, entrenar modelos de texto y probar sus modelos con usuarios reales. En este caso y debido a la variedad de búsquedas que podrían atender, la compañía en cuestión requiere tener acceso con gran velocidad a muchos datos, procesarlos con rapidez y ejecutar búsquedas cuasi instantáneas.

Pese a que la época moderna se caracteriza por lo instantáneo y efímero, una fracción pequeña de profesionales trabajará en proyectos tan grandes como desarrollar un buscador web masivo. Por esta razón, concluimos que la implementación de una tecnología de Big Data, que es disruptiva y costosa, ha de justificarse en el uso que le den sus usuarios y en los perfiles de éstos, pero no en los movimientos tendenciales de la industria.