Maestría en ciencias de los datos y analítica

Anteproyecto

Minería de Datos para Grandes Volúmenes de Información

ANÁLISIS Y PREDICCIÓN DE LA DEMANDA DE USUARIOS DE SERVICIOS DE TRANSPORTE UTILIZANDO MACHINE LEARNING Y SERIES DE TIEMPO

Presentado por: Paula Andrea Cárdenas López, Alejandro Velásquez Arango, Jeison Erley Giraldo Toro.

Profesor: Carlos Alzate

Lugar: Medellín

Fecha: 17 de mayo de 2023

1. Pregunta de investigación y objetivos

1.1. Pregunta de investigación

Este proyecto utiliza como fuente un conjunto de datos de Kaggle llamado “Uber NYC for-hire vehicles trip data (2021)”, el cual cuenta con información del registro diario de viajes a través de taxis amarillos, verdes, o servicios prestados a través de plataformas como Uber, Lyft y Via. (Kaggle, 2023)

El objetivo principal es crear un modelo de predicción de demanda diaria de servicios de transporte, abordado desde las series temporales ARIMA y PROPHET, además de técnicas de Machine Learning como LSTM. También se pretende realizar un análisis de los factores que influyen en la demanda de servicios de transporte, como el día de la semana, hora del día, ubicación geográfica, tarifa base del pasajero, entre otros.

Por tanto, es importante resaltar que la identificación de factores que afectan la demanda genera beneficios para los diferentes entes prestadores de servicios, con el fin de mejorar la experiencia de usuarios.

2. Revisión de la literatura

Predecir la demanda de transporte puede ayudar a organizar la flota de taxis y compañías de plataformas, con el fin de minimizar el tiempo de espera de los pasajeros y conductores, por tanto para cumplir este propósito se utilizan comúnmente estas técnicas:

- **LSTM (Long Short Term Memory networks)**
  Es un modelo de aprendizaje secuencial que permite predecir las futuras solicitudes de transporte, basándose en la demanda reciente y otra información relevante. Este tipo de red neural debe predecir:

  $$yt = et + 1$$

  Donde, cada paso de tiempo es t y et represents the number of pickups. (Xu, Rahmatizadeh, Bölöni, & Turgut, 2017)

- **ARIMA**
  Viene de auto regresivo (AR) y media móvil (MA), este modelo permite capturar patrones estacionales en los datos, como la hora del día o el día de la semana, a través del análisis de correlación (ACF) y correlación parcial (PACF). Adicionalmente, el test de Dickey-Fuller (ADF), permite interpretar la no estacionariedad de la serie por medio del valor-p. (Faghiha, Shahb, Wangb, Safikhanic, & Kamgaa, 2020)

- **PROPHET**
  Es un procedimiento de pronóstico automatizado creado por Facebook para series temporales, es eficaz en series con efectos estacionales y múltiples temporadas de datos históricos, además de ser fuerte ante valores atípicos y cambios en la tendencia. (Beneditto, Satrio, Darmawan, Unrica, & Hanafiah, 2021)

Las métricas de evaluación comunes para estos modelos de series de tiempo son:

- **RMSE** (Root Mean Squared Error)

$$\mathrm{RMSE} = \left[\frac{1}{n}\sum_{i=1}^{n}\left(|f_i - \hat{f}_i|\right)^2\right]^{\frac{1}{2}}$$

  Donde, f_i es el valor observado y $\hat{f}_i$ es la predicción. (Lv, Duan, Kang, Li, & Wang, 2014)

- **MAPE** (Mean Absolute Percentage Error)

$$\mathrm{sMAPE}_k = \frac{1}{t}\sum_{i=1}^{t}\frac{|R_{k,i} - X_{k,i}|}{\varrho_{k,i}}$$

$$\varrho_{k,i} = \begin{cases} R_{k,i} + X_{k,i} & \text{if } (R_{k,i} > 0 \vee X_{k,i} > 0) \\ 1 & \text{if } (R_{k,i} = 0 \wedge X_{k,i} = 0) \end{cases}$$

  Donde, R_k es una serie de tiempo discreto con un número k previsto de parada de taxis. (Moreira, Gama, Ferreira, Mendes, & Damas, 2013)

3. Metodología de investigación

3.1. Entendimiento del negocio

La comisión de taxis y limosinas de New York público el registro diario de los viajes de taxis amarillos y verdes, además de los datos de las empresas de alquiler de vehículos a gran volumen como Uber, Lyft y Via. Con el objetivo de generar información que contribuya a mejorar la planificación del transporte público y proporcionar una experiencia de viaje más eficiente y satisfactoria para los usuarios.
Esta información es valiosa para comprender el negocio de transporte en la ciudad, ya que permite analizar y estudiar patrones de demanda, tendencias de viaje, rutas populares, horarios de mayor demanda, áreas geográficas más transitadas, entre otros aspectos relevantes.

3.2. Entendimiento de los datos

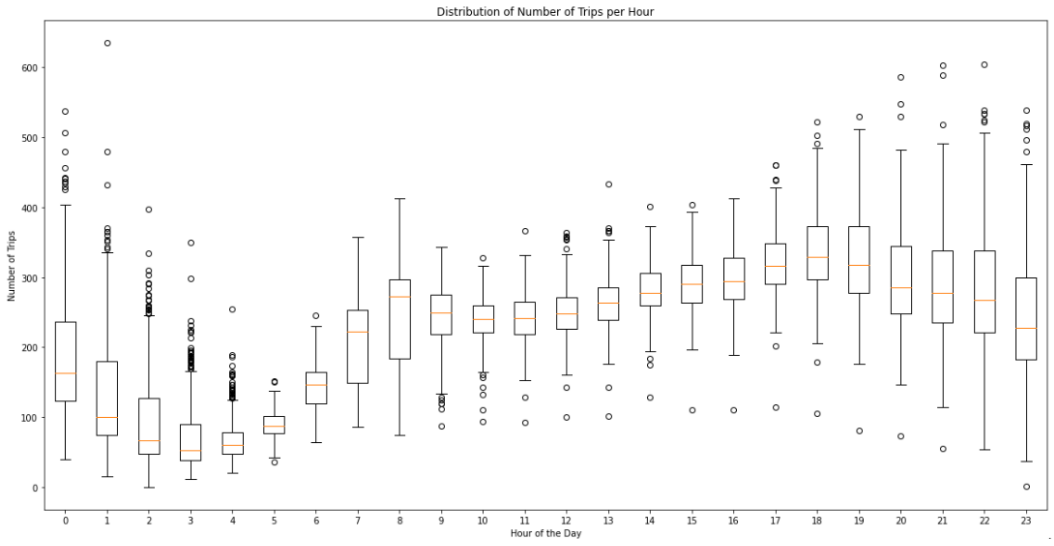
- **Recopilación datos**
  Se configuro la conexión para descargar los datos del registro de viajes desde Kaggle, el archivo comprimido contiene 12 archivos tipo parquet que

comprenden la información de viajes mensuales del 2021 de 3 compañías proveedoras de servicio de transporte a través de aplicativos móviles, estas empresas serán identificadas como HV0003: Uber, HV0004: Via y HV0005: Lyft.
En total se cuenta con 174.596.652 de registros de viajes.
Los datos están disponibles en el siguiente enlace de Kaggle: https://www.kaggle.com/datasets/shuhengmo/uber-nyc-forhire-vehicles-trip-data-2021?select=taxi_zones

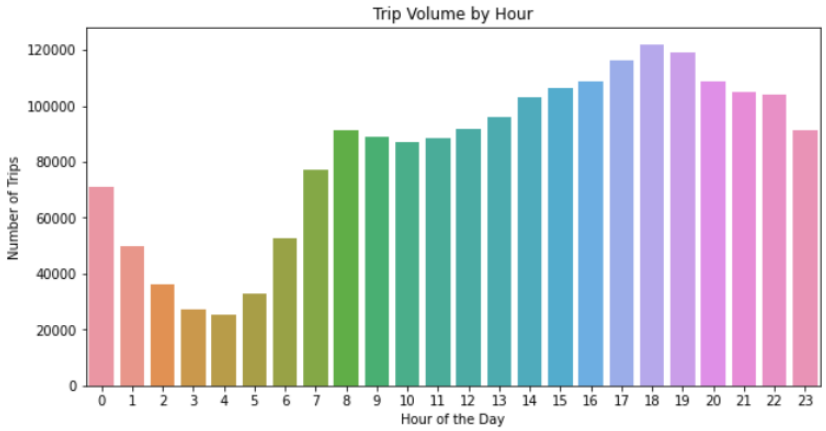
- **Descripción de los datos**
  En esta etapa se revisa la cantidad de variables y su significado, además de la cantidad de variables con valores nulos y el tipo de formato de cada una. Ver anexo.
  De acuerdo a lo anterior, el conjunto de datos contiene un total de 24 variables con los siguientes formatos: datetime64[ns] (4), float64 (9), int64 (3) y object (8).
  A continuación, se mencionan las variables con datos incorrectos o vacíos que son tratados posteriormente:
    - originating_base_num: missing 28%
    - on_scene_datetime_: missing 28%
    - airport_fee:

- **Exploración de datos**
  En la exploración de datos se realizaron tablas de frecuencia, gráficos de distribución y boxplot para entender el comportamiento de las variables:

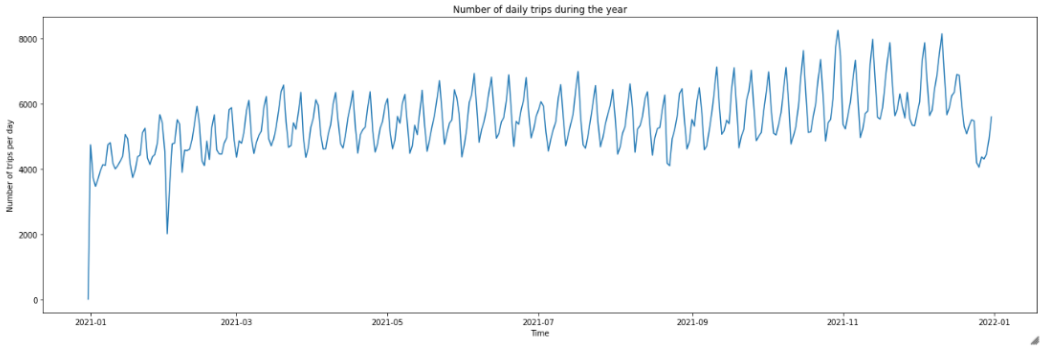
  - **Distribución del número de viajes por hora:**



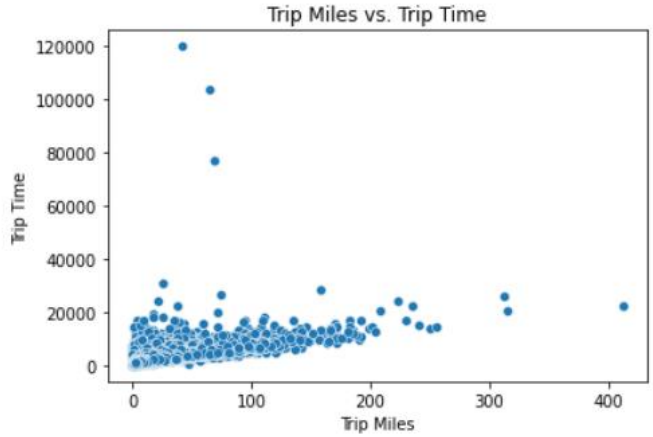
- **Cantidad de viajes por hora:**



- **Cantidad de viajes durante el año:**

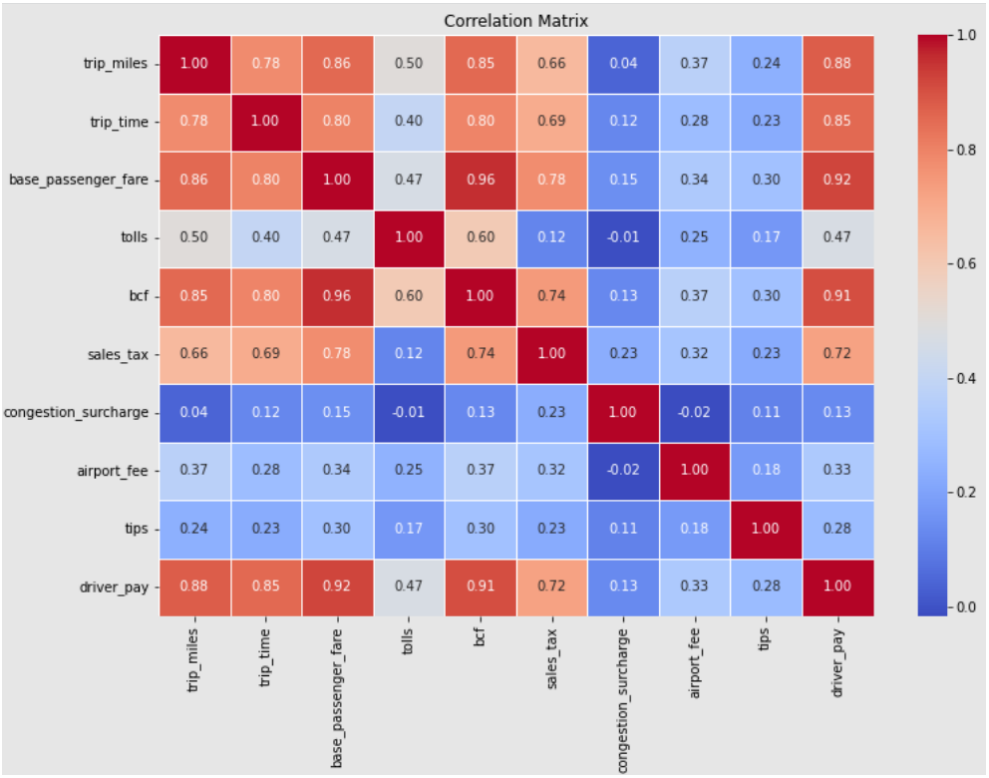


- **Correlación entre el tiempo de viaje y millas recorridas:**



- **Análisis univariante:** se realizó el describe de todas las variables del dataframe, en el notebook anexo se encuentra el desarrollo completo.

- **Análisis multivariante:** se construyó un mapa de calor para ver la correlación lineal entre cada par de variables:



Encontrando las siguientes interpretaciones de correlación:

- Entre mayor sea la cantidad de millas recorridas mayor es la cantidad de tiempo en el viaje.
- Entre mayor es la cantidad de millas recorridas mayor es la base del viaje, impuestos y pago al conductor.

3.3. Preparación de los datos

En esta etapa el objetivo es preparar los datos para adaptarlos a las técnicas de series de tiempo y aprendizaje automático que se utilizaran en el modelado.

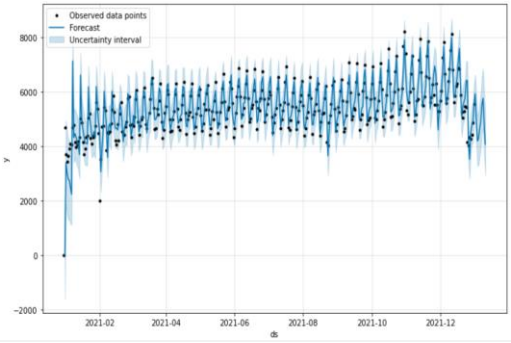
- **Transformación de variables**
  Se aplica get_dummies a las variables categóricas, eliminado la primera categoría para evitar multicolinealidad.
- **Limpieza de datos**
  En este punto se eliminan variables que no aportan al modelo según el problema a resolver.
- **Ingeniería de características**
  En este paso se reemplazan los valores nulos vistos anteriormente por 0. Adicionalmente, se identifica mucha dispersión en la variable PULocationID y DOLocationID, por ello se toma la decisión de tratarlas,

generando un nuevo valor score que proporciona información sobre qué tan frecuentadas son las zonas de embarque y desembarque.

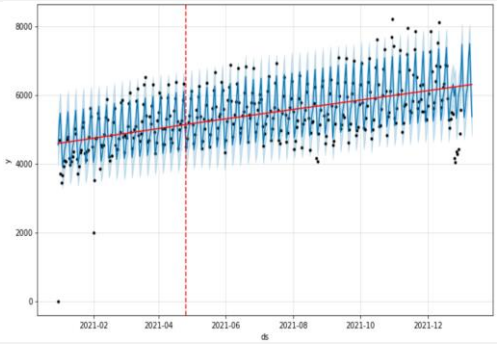
3.4. Modelado

Durante esta etapa del proyecto, se decidió desarrollar tres modelos diferentes: ARIMA, Prophet y LSTM (Long Short-Term Memory), siendo ARIMA el modelo de referencia principal.
Para el modelo ARIMA y Prophet se creó un experimento ML en Databricks de tipo forecast, donde la variable objetivo fue la cantidad de viajes a predecir diariamente en un horizonte de tiempo de 10 días, validado bajo la métrica de evaluación RMSE.

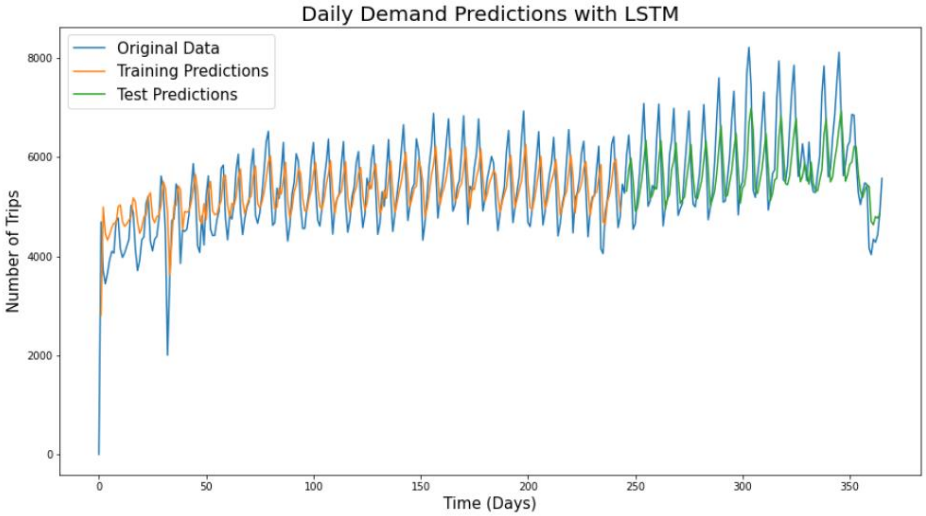
Serie de tiempo ARIMA



Serie de tiempo Prophet



Adicionalmente se entrena una red neuronal LSTM con una capa LSTM, con una función de pérdida de error cuadrático medio como criterio de optimización. Se ajusta el modelo utilizando los datos de entrenamiento durante 100 épocas, es decir, se hace una pasada completa por todo el conjunto de datos de entrenamiento durante el entrenamiento de la red neuronal.



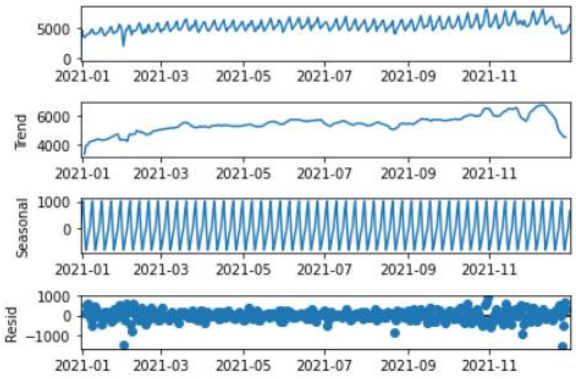
3.5. Evaluación

- **Métricas de evaluación**
  El modelo LSTM obtuvo el mejor resultado con un RMSE de 604.15, lo cual indica que este modelo tiene mejor rendimiento y precisión en la predicción de la demanda en comparación con ARIMA y Prophet.

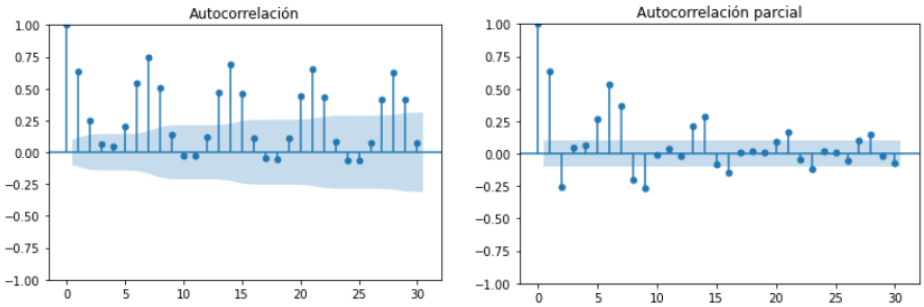
Métrica / Modelo	ARIMA	PROPHET	LSTM
RMSE	765.13	634.36	604.15

4. Análisis de los datos

Se llevó a cabo un análisis de descomposición de series temporales con el objetivo de comprender la tendencia en la cantidad de viajes diarios a lo largo del año, identificar patrones estacionales que se repiten de forma predecible y analizar el residuo para identificar la variabilidad aleatoria presente en los datos.



Posteriormente, se aplicó el análisis de autocorrelación para evaluar la correlación entre los valores de una serie temporal y sus valores pasados. Además, se utilizó la autocorrelación parcial para medir la correlación entre los valores, teniendo en cuenta los efectos de los valores intermedios. En otras palabras, se examinó la relación directa entre dos puntos de datos, controlando los efectos de los valores intermedios.

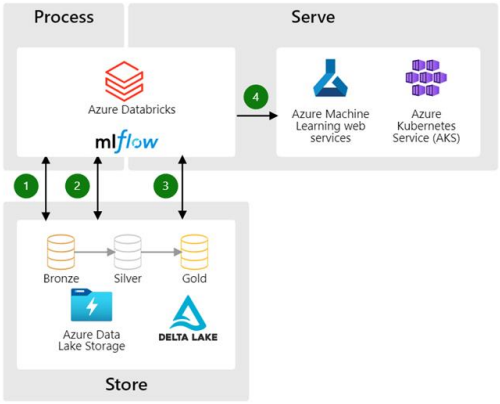


Por último, se realizó el test de Dickey-Fuller para evaluar la estacionariedad de la serie temporal, en este caso dado que el Valor p es mayor que 0.05 y el valor del estadístico ADF no es más negativo que los valores críticos correspondientes, no hay suficiente evidencia para rechazar la hipótesis nula de no estacionariedad en la serie temporal. En este caso, la serie no se consideraría estacionaria.

```
Estadístico ADF: -2.3331599550065873
Valor p: 0.16149610527520136
Valores críticos:
        1% : -3.44911857009962
        5% : -2.8698097654570507
        10% : -2.5711757061225153
```

5. Uso de herramientas de Big Data

Los datos serán almacenados en Data Lake Storage utilizando el formato Delta Lake, organizados mediante la arquitectura de medallas Bronze (datos sin procesar), Silver (datos validados), y Gold (datos enriquecidos). Estos datos son procesados en Databricks con PySpark para finalmente generar modelos de Machine Learning que son implementados en el repositorio de modelos de MLflow. Estos modelos pueden ser desplegados en Azure Machine Learning y en Azure Kubernetes Service. A continuación, se ilustra el pipeline de la arquitectura:



Para llevar a cabo el proyecto, se utilizó la cuenta educativa de Databricks con un clúster de runtime 12.1 ML, que incluye Apache Spark 3.3.1 y Scala 2.12, con 14 GB de memoria y 4 cores.

Se utilizaron los siguientes recursos del clúster para llevar a cabo el proceso de preparación y exploración de datos:

Parquet IO Cache

Data Read from External Filesystem (All Formats)	Data Read from IO Cache (Cache Hits, Compressed)	Data Written to IO Cache (Compressed)	Cache Misses (Compressed)	True Cache Misses	Partial Cache Misses	Rescheduling Cache Misses	Cache Hit Ratio	Number of Local Scan Tasks	Number of Rescheduled Scan Tasks	Estimated Size of Repeatedly Read Data	Cache Metadata Manager Peak Disk Usage
49.2 GiB	0.0 B	0.0 B	0.0 B	0.0 B	0.0 B	0.0 B	0 %	0	0	4.3 GiB (8 %) - 4.3 GiB (8 %)	0.0 B

La lectura de los datos directamente en Kaggle contenía 12 archivos parquet por cada mes, alcanzando en total 174.596.652 de registros, por tanto, se tomó un sample de 2.000.000, con el que se realizó todo el proyecto. Adicionalmente, en algunas celdas se aplicó la operación spark.cache(), con el fin de almacenar el DataFrame en la memoria de los nodos de Spark que están ejecutando el código.

Como resultado del proceso anterior, se generaron 117 bloques de RDD con un total de 10.180 tareas completadas:

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Excluded
Active(1)	117	1 GiB / 3.3 GiB	0.0 B	4	0	0	10180	10220	33 min (31 s)	49.2 GiB	3.7 GiB	3.3 GiB	0
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B	0
Total(1)	117	1 GiB / 3.3 GiB	0.0 B	4	0	0	10180	10220	33 min (31 s)	49.2 GiB	3.7 GiB	3.3 GiB	0

Después de preparar los datos, se procedió a almacenarlos en un sistema de archivos distribuido llamado DBFS (Databricks File System) utilizando el formato Parquet, con el fin de ser el insumo para los modelos.

Con los datos preparados, se generó la serie de tiempo del número de viajes que hubo por día. Esta serie se guardó en una tabla en el DBFS para poder llevar a cabo experimentos de AutoML de Databricks. AutoML creó múltiples modelos PROPHET y ARIMA que sirven de baseline para la comparación de otros modelos. Además, generó notebooks editables que contienen información sobre la creación y desempeño de estos modelos.

6. Entregables y su descripción

Este documento y los anexos de código están ubicados en el repositorio de Github, con el respectivo README que indica el procedimiento que se llevó a cabo para el desarrollo del proyecto. En el enlace a continuación se accede al repositorio público de Github: https://github.com/Paulacardenas/user-demand-Forecast-and-LSTM.git
El proyecto contiene lo siguiente:

Formato archivo	Nombre Archivo
Pdf	Proyecto final Anexo 1. Diccionario_variables_trip_data
Notebooks (dbc)	Anexo 2. Data_Prep (1) Anexo 3. ARIMA Model - Uber-nyc-trip-data (AUTOML) Anexo 4. PROPHET Model - Uber-nyc-trip-data (AUTOML) Anexo 5. LSTM Model - Uber-nyc-trip-data

7. Conclusiones y trabajo futuro

- Se produjo un marcado contraste en relación a proyectos que habíamos realizado. La discrepancia se hizo evidente durante el procesamiento y la exploración de los datos. Al tratar de manejar más de 170 millones de

registros, se notó una clara diferencia en los tiempos de procesamiento y en la ejecución de los procesos con Pyspark. En esta ocasión, optamos por tomar una muestra de 2 millones de registros.

- El período de mayor volumen de viajes se observa entre las 9 de la mañana y las 6 de la tarde, lo cual coincide con la jornada laboral en Estados Unidos.
- LSTM demostró una gran capacidad para modelar y predecir la demanda diaria de viajes. Modelos de aprendizaje profundo, específicamente LSTM, pueden aprender de patrones de viaje históricos y capturar dependencias temporales de manera efectiva, esto es clave para predicciones más precisas.
- Para trabajos futuros se podría generar un modelo de regresión lineal para identificar todos los betas correspondientes a las variables que impactan la demanda de transporte, para posteriormente aplicar los modelos de series de tiempo.

8. Ejecución del plan

La variación con respecto al cronograma planteado inicialmente fue que durante dos semanas se suspendió, por lo que la intensidad horaria en la preparación de datos fue mayor a lo estipulado en un primer momento.

Cronograma proyecto															
		Marzo					Abril				Mayo				
		Sem 1	Sem 2	Sem 3	Sem 4	Sem 5	Sem 1	Sem 2	Sem 3	Sem 4	Sem 1	Sem 2	sem 3	sem 4	
Comprender Negocio															
1	Definir Equipo	X													
2	Entendimento del negocio	X													
3	Definir alcance de la necesidad		X												
Comprender Datos															
4	Recopilacion de datos iniciales		X	X											
5	Descripción y exploración de datos						X								
Preparar Datos															
6	Seleccionar datos a trabajar							X							
7	Limpieza de datos							X	X						
8	Construcción de nuevos datos								X						
Modelar Datos															
9	Selección de técnicas									X					
10	Generación de técnicas										X				
11	Evaluación de modelos										X				
Evaluar Datos															
12	Realizar pruebas & validar resultados										X	X			
Distribuir Datos															
10	Paso a producción														
11	Entrega reporte												X		
12	Presentación														

Se ha confirmado como una lección fundamental que la etapa de preparación de datos representa aproximadamente el 70% del tiempo en un proyecto de análisis de datos. No obstante, es importante reconocer que la complejidad de este proceso puede variar según el tipo de problema que se esté abordando.

9. Implicaciones éticas

Asumiendo que este proyecto fuera utilizado para propósitos comerciales por una compañía de transporte, tanto los trabajadores como los usuarios de las plataformas de servicios se verían afectados positiva o negativamente, dependiendo del éxito del proyecto. Si el modelo logra predecir correctamente los momentos de mayor y menor demanda, los conductores de la plataforma podrían utilizar la información dada por el modelo y obtener mayores ganancias al poder atender constantemente pedidos. Consecuentemente, los usuarios podrían encontrar con mayor facilidad un conductor

que los pueda transportar en tiempos de alta demanda, reduciendo el tiempo de espera mientras la plataforma encuentra un conductor. En general, la calidad de vida de tanto conductores como usuarios podría mejorar en caso de éxito del modelo.

En el caso en que el modelo prediga erróneamente los momentos de demanda y prediga baja demanda cuando en realidad es alta, los usuarios experimentarían mayores tiempos de espera debido a que habrá menos conductores dispuestos a trabajar en ese momento si deciden guiarse por la información dada por el modelo. Esto disminuiría el dinero que los conductores (y por tanto la empresa) puedan generan. En caso contrario, es decir, el modelo prediga alta demanda cuando en realidad es baja, habría tiempos de espera bajos para los usuarios, pero los conductores no encontrarían pasajeros frecuentemente y generarían menos ingresos, potencialmente afectando su calidad de vida.

10. Aspectos legales y comerciales

El potencial comercial de este proyecto para la compañía Uber, e inclusive las demás empresas proveedoras de servicios de movilidad como Juno, Via y Lyft, reside en la posibilidad de diseñar un algoritmo de Machine Learning que permita identificar los patrones de uso para predecir la demanda e identificar las variables que más influyen en ella. Logrando así, generar estrategias para mejorar la experiencia de usuario, asociado al tiempo de espera, calidad del servicio y facilidad de uso de la aplicación.

Por otro lado, los aspectos legales que se deben considerar en este tipo de algoritmos se derivan de la existencia de decisiones que puedan afectar a los usuarios de las plataformas de servicios de transporte o terceros.

En esta investigación en particular los datos son públicos, por tanto, no hay acuerdos de confidencialidad y los resultados que se obtengan serán de código abierto.

11. Bibliografía

Beneditto, C., Satrio, A., Darmawan, W., Unrica, B., & Hanafiah, N. (2021). Time series analysis and forecasting of coronavirus disease in Indonesia using ARIMA model and PROPHET. *Procedia Computer Science*, 524-532.

Faghiha, S., Shahb, A., Wangb, Z., Safikhanic, A., & Kamgaa, C. (2020). Taxi and Mobility: Modeling Taxi Demand Using ARMA and Linear Regression. *The 11th International Conference on Emerging Ubiquitous Systems and Pervasive Networks* (págs. 186-195). Madeira: Elsevier B.V.

Kaggle. (01 de enero de 2023). *Kaggle*. Obtenido de Kaggle: https://www.kaggle.com/datasets/shuhengmo/uber-nyc-forhire-vehicles-trip-data-2021?select=taxi_zones

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F.-Y. (2014). Traffic Flow Prediction With Big Data: A Deep Learning Approach. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, 1-9.

Moreira, L., Gama, J., Ferreira, M., Mendes, J. -M., & Damas, L. (2013). Predicting Taxi–Passenger Demand Using Streaming Data. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, 1393-1402.

Xu, J., Rahmatizadeh, R., Bölöni, L., & Turgut, ,. (2017). Real-Time Prediction of Taxi Demand Using Recurrent Neural Networks. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, 2572 - 2581.

12. Anexos

- Anexo 1. Diccionario_variables_trip_data
- Anexo 2. Data_Prep (1)
- Anexo 3. ARIMA Model - Uber-nyc-trip-data (AUTOML)
- Anexo 4. PROPHET Model - Uber-nyc-trip-data (AUTOML)
- Anexo 5. LSTM Model - Uber-nyc-trip-data