Ciencia de Datos Aplicada – Segunda entrega del proyecto

Jose F. Corzo Manrique, Felipe A. Gutiérrez Naranjo, Alejandro Mantilla Redondo
Reporte técnico del segundo *sprint* del proyecto
Universidad de los Andes, Bogotá, Colombia
{j.corzom, fa.gutierrez, a.mantillar}@uniandes.edu.co
Fecha de presentación: noviembre 15 de 2021

Tabla de Contenido

1 Enlaces relevantes de solución

Esta sección contiene los múltiples enlaces necesarios para soportar la segunda entrega del proyecto del curso:

- Repositorio específico donde se encuentra la solución de esta tarea: https://github.com/MantiMantilla/Ciencia-de-datos-aplicada-2021-20/tree/main/Proyecto/Sprint_2

2 Enfoque analítico

Al sintetizar el contexto del proyecto, se resalta cómo los taxistas en la ciudad de Nueva York en los Estados Unidos se enfrentan a numerosos obstáculos (inseguridad, gente que huye sin pagar, competencia intensa, la pandemia, etc.) que dificultan realizar su trabajo de manera que sea genuinamente rentable. Así, de define al problema como: **generar un plan de identificación de zonas y rutas que maximicen las ganancias de los taxistas, a partir del histórico de trayectos de taxi amarillo en la ciudad de Nueva York del TLC.**

De modo similar, se designan las siguientes preguntas:

- Confirmar, por medio del monto total recolectado por trayecto, si la producción de los conductores de taxi ha disminuido a través de los años.
- Comprobar si, en efecto y por medio del tipo de viaje, los trayectos hacia el aeropuerto de John F. Kennedy en verdad son más lucrativos frente a los demás.
- Identificar, por medio de las horas de los trayectos, cuáles suelen ser los más rápidos.

Con base en lo anterior, la tarea predictiva específica en esta entrega es: **predecir qué circunstancias maximizan en verdad el monto recolectado por un trayecto de taxi amarillo en la ciudad de Nueva York**. La variable objetivo del ejercicio es "Total_amount" (representativo del costo final que el (los) pasajero(s) paga al concluir el trayecto), mientras que las variables predictoras son "total_time_min", "total_trip_distance", "total_fare_amount", "total_tip", "total_toll", "total_amount", "total_taxis" (todas el resultado de agregaciones de las variables descritas en el documento anterior).

3 Entendimiento de los datos

Ahora se debe describir el reporte de los problemas de calidad encontrados en el data set, junto a las soluciones para éstos, y cualquier transformación que se haya realizado también.

3.1 Errores y soluciones para incrementar la calidad en el *dataset* original

Se identificaron las siguientes situaciones:

- **Filas en los años erróneos**: se identificó que un porcentaje mínimo de filas se encontraban en años donde no correspondían. Como se considera que es más probable que estas lecturas fueran un error causado por los dispositivos de recolección de información, se eliminaron estas filas por medio de una selección exclusiva de los datos del 2019. Este año es el más reciente del que se tenía información, que problemas de tiempo impidieron complementar con otros**.**
- **Valores de cero, o negativos, en las columnas del tiempo, la tarifa o la distancia de los trayectos**: se identificaron este tipo de datos en estas columnas que, a simple vista, son obvios errores. Mientras es posible que los datos de las tarifas sean representativos de los robos que se sabe que ocurren, esto no puede confirmarse y sólo perjudicaría a los modelos. Por lo tanto, se elimina toda fila con estos datos.
- **Limpieza de *outliers* en las columnas de tarifa y distancia de los trayectos**: al utilizar gráficos de caja y bigote, se identifica que estas columnas presentan outliers bastante marcados; se eligieron límites a partir de las gráficas de cuarenta mil para el total de los precios y veinte mil para la distancia, a fin de deshacerse de los *outliers* más evidentes.

3.2 Enriquecimiento de los datos

Con base en retroalimentación obtenida en la entrega anterior, se realiza un proceso de enriquecimiento del *dataset* con los siguientes *datasets* que podrían ser relevantes, junto a las variables que introducen:

- Boros y zonas para los trayectos: este dataset, proveniente de la misma página de donde se obtuvieron los datos originales (1), indica la zona específica donde se realiza cada trayecto. Contiene las siguientes variables útiles:
    - Zone: variable con el nombre de la zona específica en sí.

- Días festivos en Nueva York: este *dataset*, provisto también por *Public Holidays Global* (2), indica los días festivos relevantes para la ciudad de Nueva York. Contiene las siguientes variables útiles:
    - Holiday: variable booleana que indica si el día es feriado o no.
- Tormentas severas en NY: este *dataset*, provisto por *NOAA* (3), indica la presencia de tormentas fuertes en el área de Nueva York, representativa del clima durante los días que afectaron. Contiene las siguientes variables útiles:
    - Storm_Event: variable booleana que indica si el día presentó clima severo o no.

3.3 Transformaciones sobre los datos

Como último paso en el análisis de los datos antes de la creación de los modelos, se realizan las siguientes transformaciones:

- **Agregación del modelo**: este es, en verdad, el primer cambio que se realizó sobre los datos, a fin de tener una cantidad apropiada de datos que no requirieran de mayor capacidad de cómputo que el uso de *Jupyter Notebook*. El resultado es un archivo de 60 MB, fundamental para esta entrega.
- **Selección del año 2019**: más que una transformación, fue una limitación genuina para el proyecto. Debido a problemas de tiempo, se optó por utilizar filas exclusivamente de este año.
- **Decodificación de los meses**: se creó una función, “cat_month”, que reemplazaba el número del mes por su nombre correspondiente, en inglés, por los clientes esperados.
- **Decodificación de los días de la semana**: de igual que la función anterior, se creó la función “cat_semana” para obtener el nombre los días de la semana en inglés.
- **Representación categórica del día del mes y la hora**: se utilizó el string representativo de los valores numéricos de estas variables para utilizarlas como categóricas.
- **Eliminación de columnas duplicadas y nulos**: se eliminaron las columnas con las que se realizaron los JOIN de enriquecimiento, ya que no resultan de valor para la entrega. De igual modo, se eliminaron los valores nulos.
- **Relleno de booleanos en las columnas enriquecidas**: se rellenaron los campos de las columnas enriquecidas donde fuera necesario, con cero.
- **Creación de variable categórica objetiva**: se creó la columna “total_dollars_avg” al dividir el valor del total del cobro entre el total de taxis en operación para esa fila, luego se identificaron las medidas de sus cuartiles uno y tres, para designar los valores de bajo, medio y alto, en inglés, en la nueva columna objetivo “amount_level”.
- **Creación de variable de peso para los modelos**: se creó una columna representativa del peso que cada fila tendría en los modelos al multiplicar la velocidad promedio (calculada a partir de las columnas de tiempo y distancia) con el producto del total de propinas, peajes y monto total, todo dividido entre la cantidad de taxis. El resultado se dividió entre su promedio para normalizar.

4 Preparación de los datos

Como se menciona en el enunciado, el conjunto de datos se divide en conjuntos de prueba y entrenamiento por medio de la operación “train_test_split” que se ha visto a lo largo del semestre, para crear nuestras variables X – Y de entrenamiento y de prueba. Se optó que el balance entre ambas fuera de 75%: 25% respectivamente, con una semilla de 200 para recrear el modelo. Las distribuciones de ambos conjuntos son similares. Esto puede verse a continuación:

```
Training data shape: (858678, 336)
Training label shape: (858678,)
Testing data shape: (286226, 336)
Testing label shape: (286226,)
```

Figura 1. Distribuciones del modelo

5 Construcción del modelo

La sección muestra los detalles de los modelos diseñados.

5.1 Modelo creado por *Decision Tree*

El primer modelo desarrollado se hizo a través de *Decision Tree*. Este algoritmo se basa en una estructura de flujo similar a un árbol, donde cada nodo indica una prueba sobre los atributos predictores, cada rama indica un resultado de esas pruebas, y cada hoja indican una clase resultado del proceso (4). Se seleccionó por sus beneficios teóricos: la exploración de distintas posibilidades a través de lógica de grafos, considerando el costo de cada decisión, ofrece un prospecto bastante útil. Por otro lado, el tratar con un único árbol en vez de un *random forest* por ejemplo es muchísimo más barato en términos de tiempo de computación.

6 Evaluación del modelo

La sección muestra los resultados de la evaluación de los modelos de la sección anterior.

6.1 Modelo creado por *Decision Tree*

En cuanto al entrenamiento, los resultados demuestran que el modelo tiene una amplia oportunidad de mejora; la precisión es bastante aceptable para los valores de ganancias “medios” identificados en el *dataset*, pero es bastante flojo en términos de los “altos” o “bajos”, posiblemente porque éstos no son tan numerosos como el primero, a pesar de la introducción de la variable de peso para el modelo.

Por el lado de la validación, se encuentra que el modelo tiende particularmente hacia los falsos negativos en los “altos” y “bajos” al llevar los valores al promedio, de nuevo, posiblemente por el desbalance de los datos, mientras que las pruebas revelan que el modelo requiere de una sesión de búsqueda de mejores parámetros en pro de la optimización y el encontrar de la manera adecuada de balancear los datos en el preprocesamiento. Las medidas con las cuales se llegó al mejor modelo conocido son una semilla de 200 para replicar resultados, junto al criterio de entropía, una máxima profundidad de tres y un tamaño mínimo de muestras para las hojas de cinco.

```
Matriz de confusión:  [[  4234  10630  53390]
 [    31  21331  48081]
 [    78  17295 131156]]
Precisión :  54.75428507542991
Reporte :                precision    recall  f1-score

        High       0.97      0.06      0.12     68254
         Low       0.43      0.31      0.36     69443
      Medium       0.56      0.88      0.69    148529

    accuracy                           0.55    286226
   macro avg       0.66      0.42      0.39    286226
weighted avg       0.63      0.55      0.47    286226
```

Figura 2. Resultados del modelo

En general, puede afirmarse que el modelo es satisfactorio para los valores comunes de los datos, por decirlo así, pero deja mucho que desear para los casos de mayores ganancias, que es la situación que se espera brindarle a los taxistas. Al pensar en posibilidades de mejora, se destaca la búsqueda de mejores parámetros, el correcto balanceo de los datos en preprocesamiento, y la búsqueda de otros modelos que puedan ajustarse mejor a los datos.

7 Cierre de *sprint*

Después de lo que se consigue con este *sprint*, esta sección representa las consideraciones de cierre de esta entrega.

7.1 Segundas conclusiones

La ejecución con grandes volúmenes de datos puede ser verdaderamente problemática; el tiempo que se necesita tan solo para verificar si un modelo siquiera se crea en primer lugar puede tomar varias horas, lo que lleva a una inversión considerable de tiempo que termina siendo infructuosa. Las medidas que se tomaron para desarrollar un conjunto de datos con el cual trabajar que fuera mucho más manejable que el de la propuesta original no fue suficiente.

La automatización puede ser una opción que considerar para realizar pruebas bajo un esquema simple de entender, con bastante tiempo de antelación, en pro de comprobar si los modelos que son más costosos en términos de tiempo de cómputo sí dan mejores resultados, antes de perder la oportunidad de siquiera probarlos por falta de tiempo para ejecutar.

El enriquecimiento puede ser más difícil de lo esperado; a pesar de intentar con distintas fuentes de datos, si los resultados se ven demasiado relacionados con la variable objetivo, más que apoyar en la predicción, simplemente generan columnas extra a computar que no ayudan en verdad.

7.2 Acciones sugeridas

Se necesita de un mejor manejo del tiempo en general: la ejecución de modelos puede complicarse mucho más de lo esperado y el tiempo invertido en pruebas que al final terminan en un error, o no terminan de ejecutar en un tiempo apropiado, es tiempo perdido para el desarrollo.

No subestimar los requisitos de Big Data; hay casos donde los equipos con los que se trabajó se congelan mientras ejecutan algún modelo, por lo que debe encontrarse un mejor balance para el volumen de datos, de forma que sean los suficientes como para tener valor, pero no tantos que abrumen la capacidad de cómputo de la que dispone el equipo de trabajo.

Dividir mejor las tareas, de forma que se cumpla con más del proyecto, como el desarrollo REST, en vez de intentar conseguir un resultado de distintos modelos que no dan en el tiempo necesario.

7.3 Contribución del equipo

Debido al mal manejo del tiempo, el equipo terminó desarrollando el proyecto a través de sesiones síncronas de trabajo de búsqueda de información, desarrollo de código y ejecución de distintos modelos, todo en conjunto y al mismo tiempo. La división de roles no fue verdaderamente concisa.

8 Bibliografía

1. NYC Taxi & Limousine Commission. *Taxi Zona Maps and Lookup Tables.* [En línea] [Citado el: 13 de noviembre de 2021.] https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page.

2. Public Holidays Global. *Start Planning Nueva York.* [En línea] [Citado el: 15 de noviembre de 2021.] https://publicholidays.com/us/es/new-york/2019-dates/.

3. National Center For Environmental Information. *Storm Event Database.* [En línea] [Citado el: 15 de noviembre de 2021.] https://www.ncdc.noaa.gov/stormevents/details.jsp.

4. GeeksforGeeks. *Decision Trees.* [En línea] [Citado el: 15 de noviembre de 2021.] https://www.geeksforgeeks.org/decision-tree/.

5. Vittal, Ram y Muppala, Sireesha. Amazon Web Services. *Machine learning on distributed Dask using Amazon SageMaker and AWS Fargate.* [En línea] 18 de febrero de 2021. [Citado el: 21 de septiembre de 2021.] https://aws.amazon.com/blogs/machine-learning/machine-learning-on-distributed-dask-using-amazon-sagemaker-and-aws-fargate/.

6. NYC OpenData. *Bus Breakdowns and Delays.* [En línea] [Citado el: 13 de noviembre de 2021.] https://data.cityofnewyork.us/Transportation/Bus-Breakdown-and-Delays/ez4e-fazm.