Data-Driven Decision Making

Prof. Marcelo Olivares
Prof Ayudante Ian Malgarini

Assignment 2

Fecha de Entrega 15 de Enero.

Pregunta 1 – Sistema de Recomendación

Se busca analizar el dataset con canciones de Spotify (spotify.csv) para encontrar canciones "similares" y utilizarlo para generar un sistema de recomendación simple. El dataset fue obtenido de Kaggle (https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset).

1. Analice el dataset e indique algunos estadísticos descriptivos (numero de canciones, valores promedio de los atributos para cada canción, entre otros).
2. Utilice k-NN para encontrar similitudes entre canciones. Especifique la métrica de distancia que utilizó.
3. Cada miembro del grupo escoge una canción que le parezca interesante y desea evaluar que tan probable es que sea de su agrado. Cada alumno del grupo realiza lo siguiente.
    a. Para la canción seleccionada, encuentre las 10 canciones mas cercanas a esa canción.
    b. Evalue con nota de 1 a 5 esas diez canciones, donde 5 es mayor agrado y 1 mayor desagrado. Si no conoce la canción, escúchela antes de evaluarla o déjela en blanco.
    c. En base a las canciones evaluados, realice una predicción de rating para la canción seleccionada.
    d. Escuche la canción y evalúe la calidad de la predicción realizada por el modelo.

Pregunta 2 - Uber

El dataset 'data_near_laguardia.csv' contiene todos los viajes de Uber realizados durante el mes de agosto 2014 en un radio de 5 kms alrededor del aeropuerto La Guardia. El dataset incluye la fecha hora, la latitud y longitud de la localización del pasajero y la distancia al centro del aeropuerto.[1]

1. Construya un histograma de la distancia de los pasajeros al aeropuerto.
2. Filtre el dataset para usar los viajes en un radio de 2000 mts. Use este dataset para el resto de la pregunta.
3. Construya un nuevo dataset que calcule el número de viajes realizados en periodos de una hora y genere un gráfico de serie de tiempo. En base al gráfico, identifique que tipo de estacionalidades son relevantes para modelar esta serie de tiempo.
4. Construya un modelo de serie de tiempo para este dataset que permite a Uber predecir el número de viajes durante la próxima hora en esta zona.
5. Muestre un gráfico de serie de tiempo que permita visualizar la predicción y valores reales de número de pasajeros
6. Calcule el Root Mean Square Error de la predicción. Calcule el indicador de forma separada para las predicciones entre 7am-10pm y entre 10:01pm y 6:59am.

En lo que resta de la pregunta, resolveremos el problema de matching. Para el análisis usaremos solo el día 6 de agosto entre las 7am y las 10pm (ultimo bloque es 9-10pm).

Uber se planifica para tener disponible un numero de autos al comienzo de cada bloque horario listos para asignar a pasajeros. La cantidad de autos esta dada por la predicción de su modelo de serie de tiempo. Por ejemplo, si el 6 de agosto a las 5pm el modelo predice 43.84 viajes, Uber tendrá listo 44 autos a las 5pm. La ubicación de estos autos es random en el mapa, pero siempre dentro de un radio de 2km alrededor del centro del aeropuerto.

La siguiente función entrega una localización al azar dentro de un radio alrededor del punto dado por lat,lon.

```
def random_point_in_circle(radius, lat, lon):
    # Convert radius from meters to degrees (approximately)
    radius_in_degrees = radius / 111320

    # Generate random angle and radius
```

[1] Este dataset es un subconjunto de los datos publicados por FiveThirtyEight en https://github.com/fivethirtyeight/uber-tlc-foil-response/blob/master/uber-trip-data/uber-raw-data-aug14.csv

```
    theta = random.uniform(0, 2 * math.pi)
    r = radius_in_degrees * math.sqrt(random.uniform(0, 1))

    # Calculate the new latitude and longitude
    new_lat = lat + r * math.cos(theta)
    new_lon = lon + r * math.sin(theta)

    return new_lat, new_lon
```

Las coordenadas (latitud,longitud) del aeropuerto La Guardia son: 40.7769, -73.8740

7. Programa una asignación que asigna cada pasajero entrante al vehículo más cercano disponible (heurística greedy). Una vez asignado, el vehículo desaparece. Si ya no quedan vehículos disponibles, el pasajero no es asignado. Por ejemplo, si en el bloque de las 5pm se planificaron 44 autos pero llegaron 45 clientes durante las 5-6pm, el ultimo cliente quedó sin asignar. Los vehículos que quedan sin asignar durante un bloque horario desaparecen al final del bloque (no se acumulan para el siguiente bloque). Calcule la distancia promedio para todos los pasajeros asignados y el porcentaje de pasajeros que no pudo ser asignado.
8. Explique: (i) como se podría realizar una asignación más eficiente que la heurística greedy utilizada en la pregunta anterior? ; (ii) que cambios haría en esta simulación para hacerlo más realista? (e.g. es realista que lo autos estén todos disponibles al comienzo de cada hora? Que se ubiquen al azar alrededor del aeropuerto?). No es necesario que cambie la simulación, solo argumente como podría ser mejorada.