****

**Open Dataset Report**

**Hemshikha Sultoo**

**May 26, 2022**

**Introduction**

The purpose of this report is to discuss the planning and ideation process of using the Open Dataset available at <https://data.transportation.gov/stories/s/i5zb-xe34> to develop models about efficient and effective systems that will benefit the traffic flow.

**Steps:**

The links provided by the employer were used to download the Open Dataset and the supporting documents. After analyzing the dataset, I decided to work with the Lankershim CSV file. The following are the steps that I would follow to build models for this dataset using the Jupyter Notebook and Python:

1. **Data Wrangling**

* Import the following libraries:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

import seaborn as sns

sns.set()

from sklearn.cluster import KMeans

from sklearn.linear\_model import LinearRegression

* Use pd.read\_csv() from pandas to read the CSV files.
* To view the content, use dataframe.head(); it will show the first 5 rows in the CSV
* print(df.dtypes) will allow us to see if the data types are compatible and comparable.
* Use dataframe..isnull().sum().to\_frame() to see if there are any missing values; there were none.
* Use dataframe.corr(method = ‘pearson’) to see the correlation between the variables; anything > 0.6 can be used for further analysis. Higher values are preferred as they suggest stronger correlations.
* Then, eliminate data/columns that are irrelevant to our purpose.

dataframe.drop(['Total\_Frames', 'Global\_Time', 'Local\_X', 'Local\_Y', 'Global\_X', 'Global\_Y'], axis = 1, inplace = True)

1. **Model Development**

From this point, we will start creating models for the dataset that we have and in the end we will apply comparison methods to see which is better. Our model will focus on finding a patterns about how the start and destination of vehicles vary and why.

Let's start with Simple Linear Regression. I chose to proceed with this model as I want to use multiple factors to predict how vehicle types are related to all the factors.

X **=** df1[['O\_Zone']]

Y **=** df1['D\_Zone']

lm.fit(X,Y)

Ypred**=**lm.predict(X)

Ypred[0:5]

array([207.0060707, 207.0060707, 207.0060707, 207.0060707, 207.0060707])

lm.intercept\_

278.67568713025673

lm.coef\_

array([-0.7026433])

width **=** 12

height **=** 10

plt.figure(figsize**=**(width, height))

sns.regplot(x**=**"O\_Zone", y**=**"D\_Zone", data**=**df1)

plt.ylim(0,)

(0.0, 211.516335579332)

The negative coefficient is a big giveaway; it’s already telling us that this will not be a good model for our dataset.

Let’s move to Multiple Linear Regression.

col **=** df1[['v\_Class', 'Int\_ID', 'Direction']]

Fit Linear model

In [108]:

lm.fit(col, df1['D\_Zone'])

Out[108]:

LinearRegression()

Find intercept a and coefficients b1, b2, b3

In [109]:

lm.intercept\_

Out[109]:

210.92843433036586

In [110]:

lm.coef\_

Out[110]:

array([ 0.45305925, 0.04938094, -2.36392501])

Now let's use a regression plot to see how the model is working with our data

destination = lm.predict(col)

import seaborn as sns

%matplotlib inline

width = 12

height = 10

plt.figure(figsize=(width, height))

ax1 = sns.distplot(df1['O\_Zone'], hist=False, color="r", label="Actual Value")

sns.distplot(destination, hist=False, color="b", label="Fitted Values" , ax=ax1)

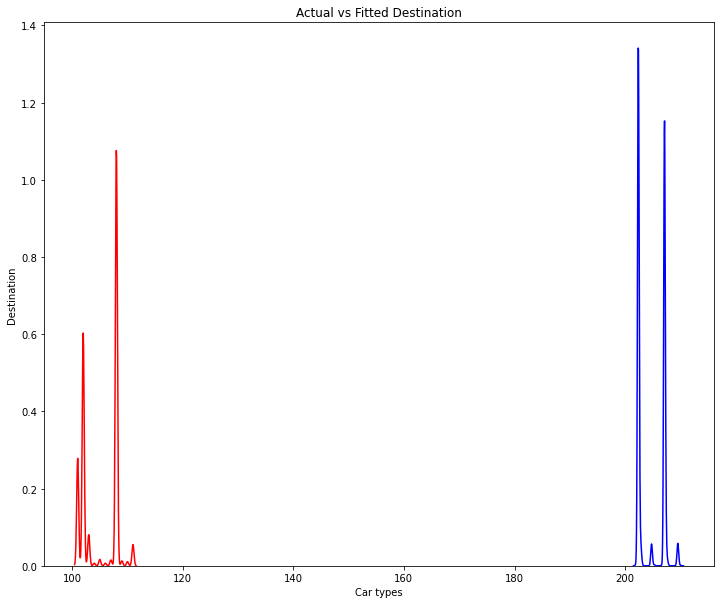
plt.title('Actual vs Fitted Destination')

plt.xlabel('Car types')

plt.ylabel('Destination')

plt.show()

plt.close()



This model is not suitable for our dataset either and the actual and fitted plot were supposed to be overlapping.

Therefore, these 2 models have been eliminated from the list.

1. **Improve efficiency and Analysis**

Another way to make this prediction more accurate is to combine multiple datasets (different locations together) and use K-Means clustering instead. The merged dataset can be fed to clustering methods such as the K-Means clustering or Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clustering with the aim of finding clusters (similar data points) about most frequent destination point or most frequent entry point or most frequent vehicle type exiting destination X and so on.

After standardizing the data, the elbow method and the K-Means inertia value could be used to find the maximum and optimal number of clusters for the analysis. Then, new cluster plots are generated.

The Silhouette Score/Coefficient can be calculated in order to compare whether the clusters are worth analyzing. The way that the Silhouette Score works is that:

If the Silhouette Score = 1, then the clusters are well apart from each other and clearly distinguished.

If silhouette score = 0, then clusters are too close and undistinguishable.

**Source:**

i) The elaboration above is from experience that I’ve gained from using K-Means clustering in an individual project for the Modeling and Simulation course. This project entails the simulation of a queuing system. Precisely, the queuing system is for the scenario where the people have to wait to use workout equipment at the gym. The project’s title is Workout Equipment Queuing System.

Imagine going to the gym during its peak hours and having to wait to use the workout equipment, E. Logically, we would perform another set of workout and circle back to this workout equipment, E, later. Then, when we come back, equipment, E, is taken by someone else and we have to wait again. The purpose of this simulation is to depict a solution to this problem and potentially utilizing it for a myriad of other applications in the future.

K-Means Code: [K-Means](https://drive.google.com/file/d/1G6zoEraGs3mtFyZ66u2ApuU2Q52Xmx2a/view?usp=sharing)

Report link: [Workout Equipment Queuing System Report](https://docs.google.com/document/d/1reQ5P3RlNNgZF40lMBcMTCl1TTLsRDszjKIsKebx8gY/edit?usp=sharing)

ii) Similarly, I’ve applied K-Means to our Intro to AI group project. This project was more about working with CSV files about neighborhood information in Toronto, try to find a correlation between them and predict how one variable affects the other. For example, how are employment and crime rates in a neighborhood codependent.

Detailed Code Explanation Video: [Code Explanation Video](https://drive.google.com/file/d/1q_4TEhiZg2NnZ2utzUothtA50m6FnCRZ/view?usp=sharing)

[Data Wrangling, Wiki Scrapping, FourSquare and Folium](https://drive.google.com/file/d/1zw7-Zu7uO2-wrEmpnQCFN4jVA53QJVs-/view?usp=sharing)

[Segmentation and Clustering](https://drive.google.com/file/d/1x165x9BOWOYVUTIwuB8F1s2mH2Ll-nko/view?usp=sharing)

Report link: [Intro to AI Project Report](https://docs.google.com/document/d/1wcxa0O4Vnq2T3uAG-vSVG3aLx1ogLdpjgoEwlHlYQYQ/edit?usp=sharing)

Link to all the files: <https://drive.google.com/drive/folders/1nWhr23R3m3vqk9H72l5413h_Dmz3qFFB?usp=sharing>

I focused more on the solution rather than getting the code ready. I tried to give my best. I initially wanted to do the microcontroller task. I ordered a raspberry pi and it’s still in transit. Therefore, this is the best that I can provide to you right now. I do have one microcontroller project that’s in the process; it is connected to my aquarium. I will be more than happy to share my aim about this project with you.

Thank you!