ANALYSIS OF UBER DATA USING MACHINE LEARNING ALGORITHM

Rohan Lahane[1],Lokendra Jagat KC[2],Krunal Pandya[3], Prof.Prakash bhise[4]

Students, Department of Information Technology, Pillai college of Engineering, New Panvel, India

[4] Professor, Department of Computer Engineering, Pillai college of Engineering, New Panvel, India

---***---

Abstract -Uber Technologies, Inc., commonly known as Uber, offers vehicles for hire, food delivery (Uber Eats), package delivery, couriers, freight transportation, and, through a partnership with Lime, electric bicycle and motorized scooter rental. The company is based in San Francisco and has operations in over 900 metropolitan areas worldwide. It is one of the largest providers in the gig economy and is also a pioneer in the development of self-driving cars. Uber is estimated to have over 110 million monthly active users worldwide. Looking at Data, we find that the data is increasing day by day and approx. 2.5 quintillion bytes of data are generated every day. Now, from this data analysis and get useful information which is most important and to understand that here we perform data analysis on UBER data using machine learning in Python. Talking about our Uber data analysis project, data storytelling is an important component of Machine Learning through which companies are able to understand the background of various operations. With the help of visualization, companies can avail the benefit of understanding the complex data and gain insights that would help them to craft decisions. The process of cleaning, transforming, manipulating data into useful information that is Data analysis. When we take a particular decision based on previous data that is data analysis. We can make future decisions using data analysis.

***Key Words*:**k-means, uber analysis, dataset, machine learning

1.INTRODUCTION

According to Gartner, by 2021, a quarter billion connected vehicles will form a major element of the Internet of Things. Connected vehicles are projected to generate 25GB of data per hour, which can be analyzed to provide real-time monitoring and apps, and will lead to new concepts of mobility and vehicle usage. Uber Technologies Inc is a peer-to-peer ride sharing platform. Uber platform connects the cab drivers who can drive to the customer location. Uber usesmachine learning, from calculating pricing to finding the optimal positioning of cars to maximize profits.Used public uber trip dataset to discuss building a real-time example for analysis and monitoring of car GPS data. Clustering is the process of dividing the datasets into groups, consisting of similar data-points". Clustering is a type of unsupervised machine learning, which issued when you have unlabeled data. Here, we have applied a K-Means clustering algorithm whose main goal is to group similar elements or data points into a cluster.

"K" in K-means represents the number of clusters. The system was mainly written in R programming and used SQL Alchemy as the ORM-layer to the database.

Uber's backend is now not just designed to handle taxies, instead, it can handle taxi, food delivery and cargo also.

The backend is primarily serving mobile phone traffic. uber app talks to the backend over mobile data.

A **neural network** is a network or circuit of neurons, or in a modern sense, an artificial **neural network**, composed of artificial neurons or nodes.

These artificial networks may be used for predictive modelling, adaptive control and applications where they can be trained via a dataset.

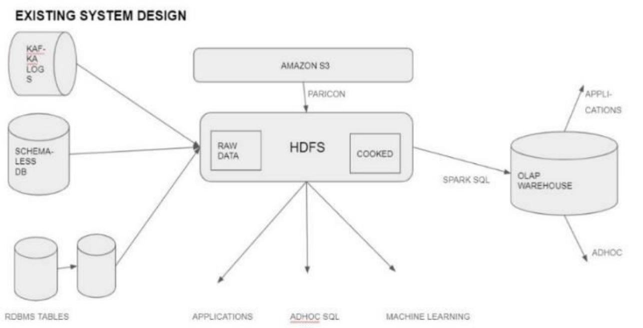


Fig. 1.1: EXISTING DESIGN

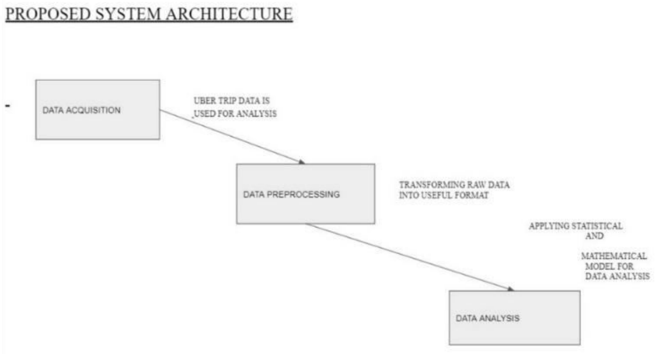
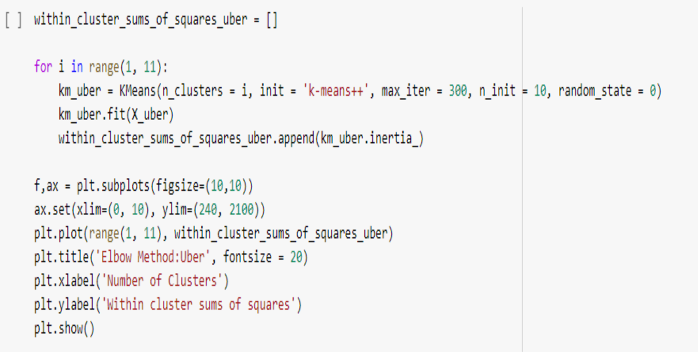


Fig.1.2: SYSTEM ARCHITECTURE

Elbow Method

A fundamental step for any unsupervised algorithm is to determine the optimal number of clusters into which the data may be clustered. The **Elbow Method** is one of the most popular methods to determine this optimal valueofk.We now demonstrate the given method using the K-Means clustering technique using the **Sklearn** library of python.



```
[ ] within_cluster_sums_of_squares_uber = []

    for i in range(1, 11):
        km_uber = KMeans(n_clusters = i, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0)
        km_uber.fit(X_uber)
        within_cluster_sums_of_squares_uber.append(km_uber.inertia_)

    f,ax = plt.subplots(figsize=(10,10))
    ax.set(xlim=(0, 10), ylim=(240, 2100))
    plt.plot(range(1, 11), within_cluster_sums_of_squares_uber)
    plt.title('Elbow Method:Uber', fontsize = 20)
    plt.xlabel('Number of Clusters')
    plt.ylabel('Within cluster sums of squares')
    plt.show()
```

Fig.1.3 Elbow Method code

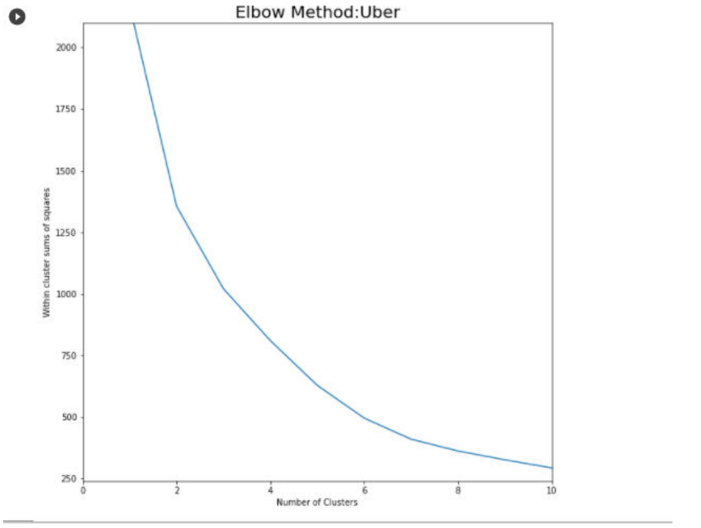
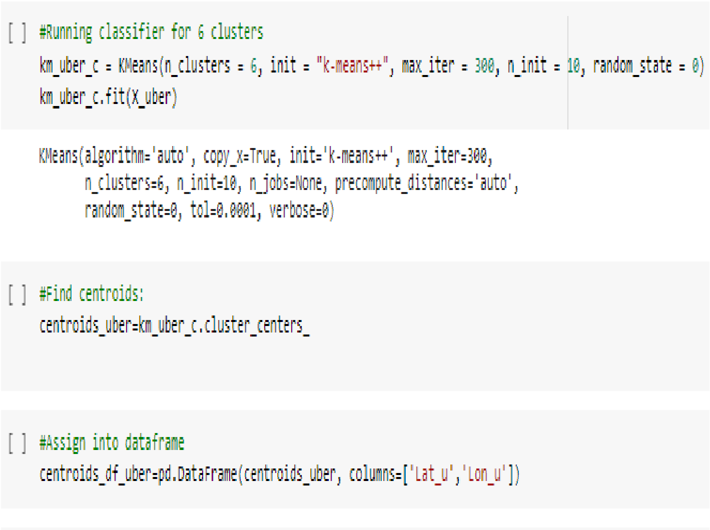


Fig.1.4 Elbow method Graph

K-means Algorithm

K-means algorithm is an iterative algorithm that tries to partition the dataset into *K*-pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to **only one group**. It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and thecluster's centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum. The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster.Specify number of clusters *K*. Initialize centroids by first shuffling the dataset and then randomly selecting *K* data points for the centroids without replacement.Keep iterating until there is no change to the centroids. i.e. assignment of data points to clusters isn't changing. Compute the sum of the squared distance between data points and all centroids. Assign each data point to the closest cluster (centroid). Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.The approach k-means follows to solve the problem is called **Expectation-Maximization**. The E-step is assigning the data points to the closest cluster. The M-step is computing the centroid of each cluster. Below is a breakdown of how we uber data



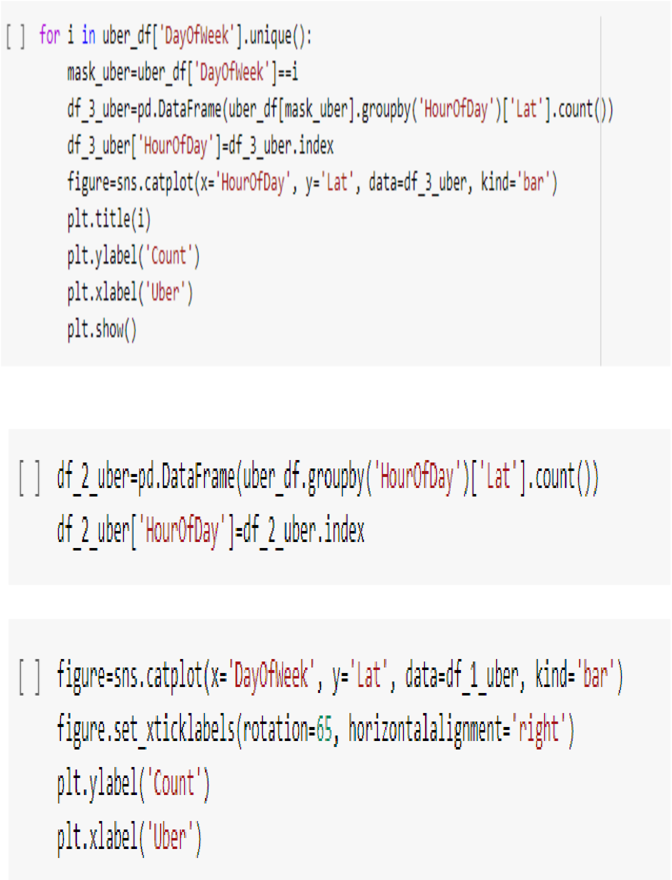
```
[ ] #Running classifier for 6 clusters
    km_uber_c = KMeans(n_clusters = 6, init = "k-means++", max_iter = 300, n_init = 10, random_state = 0)
    km_uber_c.fit(X_uber)

    KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
           n_clusters=6, n_init=10, n_jobs=None, precompute_distances='auto',
           random_state=0, tol=0.0001, verbose=0)

[ ] #Find centroids:
    centroids_uber=km_uber_c.cluster_centers_

[ ] #Assign into dataframe
    centroids_df_uber=pd.DataFrame(centroids_uber, columns=['Lat_u','Lon_u'])
```

Fig.2.0 Project code



```
[ ] for i in uber_df['DayOfWeek'].unique():
        mask_uber=uber_df['DayOfWeek']==i
        df_3_uber=pd.DataFrame(uber_df[mask_uber].groupby('HourOfDay')['Lat'].count())
        df_3_uber['HourOfDay']=df_3_uber.index
        figure=sns.catplot(x='HourOfDay', y='Lat', data=df_3_uber, kind='bar')
        plt.title(i)
        plt.ylabel('Count')
        plt.xlabel('Uber')
        plt.show()

[ ] df_2_uber=pd.DataFrame(uber_df.groupby('HourOfDay')['Lat'].count())
    df_2_uber['HourOfDay']=df_2_uber.index

[ ] figure=sns.catplot(x='DayOfWeek', y='Lat', data=df_1_uber, kind='bar')
    figure.set_xticklabels(rotation=65, horizontalalignment='right')
    plt.ylabel('Count')
    plt.xlabel('Uber')
```

Fig.2.1 Per day count

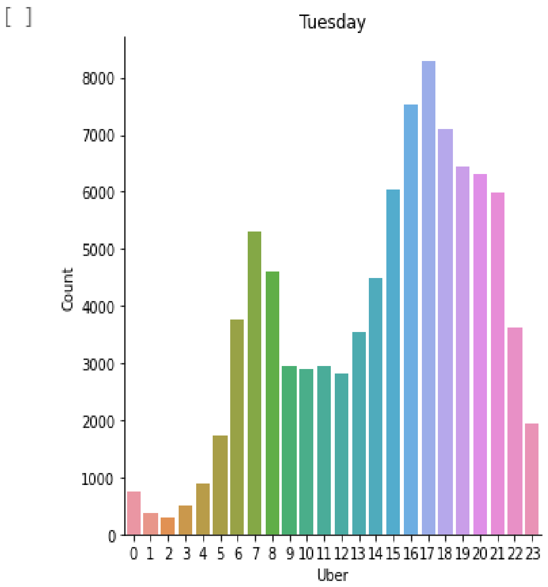


Fig.2.2 Perday Output

III.FUTURE SCOPE

Withinitiatives like UberFresh for grocery deliveries, UberRush for package courier service and UberChopper offering helicopter rides to the wealthy-Uber is all set to revolutionize private transportation globally. Uber knows the popular nightclubs in the city, best in class restaurants and has data about traffic patterns across different regions. Uber's data would soon be combined with customer specific personal data in exchange for benefits making Uber the big Data Company.

3. CONCLUSIONS

Basically,the developer should know about the basics of Python. Data visualization makes it easier to understand the core values of the databases. Machine Learning is very interesting and this is one of the projects which prove it.

This project is easily implemented and very useful for a number of apps.Not only Uber but there is a lot more application which will need to extract information from their huge databases. This project can help in that situation

REFERENCES

[1] An approach to predict taxi-passenger demand using quantitative histogram on uber data **A Bharathi, S SuryaPrakash**

[2] Real-time Prediction of Taxi Demand Using Recurrent Neural Networks **Jun Xu, Rouhollah Rahmatizadeh,LadislauBoloni andDamla Turgut**

[3] Analysis Of UberPickups In New YorkCity Using K-MEANS ClusteringAlgorithm.**Abhishek Upadhyay,Siddharth Nanda**

[4] Analysis Of Uber Pickups In New YorkCity Using K-MEANS ClusteringAlgorithm.**Abhishek Upadhyay, Siddharth Nanda**

[5] Travel Time Prediction using Machine Learning and Weather Impact onTraffic Conditions.**Bilash Deb,SalehinRahman Khan,Khandker Tanvir Dr.MdAshikulHaqueKhan.AshrafulAlam**

[6] Predicting Taxi Demand at HighSpatial Resolution: Approaching theLimit of Predictability**Kai Zhao, DenisKhryashchev,Juliana FreireClaudio Silva AndHuy**