BioGecko

Vol 12 Issue 03 2023
ISSN NO: 2230-5807

GPS Ecosystem Security Issue Implementation

Siddhi Saxena[1], Anuradha Misra[2*]

[1]UG student, Department of Computer Science& Engineering, Amity School of Engineering and Technology, Amity University, Lucknow Campus, India,
[2*]Assistant Professor, Department of Computer Science & Engineering, Amity School of Engineering and Technology, Amity University, Lucknow Campus, India,

Abstract
Taxi demand prediction plays a crucial role in urban transportation planning, resource allocation, and improving passenger experience. In this article, we delve into the topic of taxi demand prediction in New York City (NYC). We discuss various techniques employed for predicting taxi demand, the challenges associated with accurate prediction, and potential future directions to enhance the accuracy and efficiency of these predictions. This article provides a comprehensive overview of the taxi demand prediction domain in NYC, highlighting the significance of accurate predictions for effective urban transportation management. Taxi demand prediction in New York City is a complex task that involves analyzing various factors such as historical data, weather conditions, events, and time of day. While I can provide you with a general overview of the process, please note that the specifics may require more advanced techniques and access to real-time data.

Keywords: taxi demand prediction, urban transportation planning, techniques,

1. INTRODUCTION

Taxi Demand Prediction Using Machine Learning

Taxi demand prediction is a key area of application for machine learning (ML) in the transportation industry. The goal is to predict the number of taxi rides that will be requested in a given area and time period, in order to improve the efficiency of taxi services and reduce wait times for passengers. There are several approaches to predicting taxi demand using ML. One common method is to use regression models. Regression models are used to predict a continuous value, such as the number of taxi rides in a given time period. The algorithm is trained on historical data, which includes factors that may influence taxi demand, such as time of day, day of the week, weather conditions, and events taking place in the area. The model learns to map the input features to the output value, and can then make predictions about the number of taxi rides for new inputs.

Another approach to taxi demand prediction is to use time series analysis. Time series analysis involves modeling the time-dependent structure of the data, and using this model to make predictions about future values. In the context of taxi demand prediction, time series analysis can be used to model the daily or weekly patterns in taxi demand, as well as any trends or seasonal variations. This approach can be especially useful for predicting demand during holidays or special events.

Neural networks are also commonly used for taxi demand prediction. Neural networks can learn complex relationships between the input features and the output value, and can adapt to changes in the data over time. In the context of taxi demand prediction, neural networks can be used to learn patterns in the data that may be difficult to capture with traditional regression models.

To train ML models for taxi demand prediction, high-quality data is required. This data can include historical data on taxi rides, as well as data on factors that may influence taxi demand, such as weather and

traffic conditions. It is important to preprocess the data to ensure that it is in a format that can be easily used by the ML algorithm.
Once the ML model has been trained, it can be used to make predictions about future taxi demand. This can be done in real-time, using data from sensors and other sources to continuously update the predictions. The predictions can then be used to optimize the deployment of taxis and improve the efficiency of taxi services.
One key benefit of using ML for taxi demand prediction is the ability to adapt to changing conditions. ML models can be trained on a wide range of historical data, allowing them to capture the complex relationships between different factors and taxi demand. As new data becomes available, the models can be updated to improve their accuracy and make more accurate predictions.
Overall, ML is a powerful tool for predicting taxi demand and improving the efficiency of taxi services. By training algorithms to learn patterns and relationships in the data, and using those patterns to make predictions or decisions about future events, ML can provide valuable insights and improve the efficiency of transportation systems. However, it is important to carefully consider the quality and quantity of data used to train the algorithm, as well as the choice of algorithm and its parameters, in order to achieve the best possible performance.

2. BUSSINESS PROBLEM

Problem Statement-
The problem statement is to predict the number of pickups and demand for taxi in New York at a given time interval.

Constraints-
The end user or customer is a taxi driver
Latency- Demand of taxi in a few seconds

Objectives-
To predict the number of pickups in the nearby reigns in a 10 minute interval for each region in New York City
(region, 10 min interval) ---> #pickups
Most of the taxis have GPS, given the location information we can assign a taxi to a region with more demand in real time

Data Overview-
This is a CSV file provided by the TCS office in New York made available to all its other branches for case study purposes-

file name	file name size	number of records	number of features
yellow_tripdata_2016-01	1. 59G	10906858	19
yellow_tripdata_2016-02	1. 66G	11382049	19
yellow_tripdata_2016-03	1. 78G	12210952	19
yellow_tripdata_2016-04	1. 74G	11934338	19
yellow_tripdata_2016-05	1. 73G	11836853	19
yellow_tripdata_2016-06	1. 62G	11135470	19
yellow_tripdata_2016-07	884Mb	10294080	17
yellow_tripdata_2016-08	854Mb	9942263	17
yellow_tripdata_2016-09	870Mb	10116018	17
yellow_tripdata_2016-10	933Mb	10854626	17
yellow_tripdata_2016-11	868Mb	10102128	17
yellow_tripdata_2016-12	897Mb	10449408	17
yellow_tripdata_2015-01	1.84Gb	12748986	19
yellow_tripdata_2015-02	1.81Gb	12450521	19
yellow_tripdata_2015-03	1.94Gb	13351609	19
yellow_tripdata_2015-04	1.90Gb	13071789	19
yellow_tripdata_2015-05	1.91Gb	13158262	19
yellow_tripdata_2015-06	1.79Gb	12324935	19
yellow_tripdata_2015-07	1.68Gb	11562783	19
yellow_tripdata_2015-08	1.62Gb	11130304	19
yellow_tripdata_2015-09	1.63Gb	11225063	19
yellow_tripdata_2015-10	1.79Gb	12315488	19
yellow_tripdata_2015-11	1.65Gb	11312676	19
yellow_tripdata_2015-12	1.67Gb	11460573	19

3. FEATURES IN THE DATA

Machine Learning Problem Formulation

Time-series forecasting and Regression-

Time-series forecasting and regression are two types of statistical analysis techniques used to model and analyze data.Time-series forecasting is the process of predicting future values of a variable based on its historical behavior. This technique is commonly used in areas such as finance, economics, and weather forecasting. Time-series models typically use past values of a variable and other external factors to predict future values.Regression analysis, on the other hand, is a statistical technique used to study the relationship between two or more variables. It is commonly used to model the relationship between an outcome variable and one or more predictor variables. Regression models can be used to make predictions about the outcome variable based on the values of the predictor variables.While both techniques can be used for prediction, there are some key differences between them. Time-series forecasting is specifically designed for predicting future values of a variable based on its historical behavior, while regression analysis is more general and can be used to model the relationship between any two variables.

Additionally, time-series models typically use past values of a variable and other external factors to predict future values, while regression models can use any number of predictor variables to model the relationship between the outcome variable and the predictor variables.

In summary, time-series forecasting and regression are both valuable statistical techniques used for modeling and predicting data, but they have different applications and use different methods to make predictions.The given data is time series stationary.

To find number of pickups, given location coordinates(latitude and longitude) and time, in the query region and surrounding regions.Break the whole city of NYC into regions.

Given a region and some time interval, We're trying to predict the number of pickups at time(t+1) using previous pickup values in that region.

$$P_{t+1} = P_t + \alpha$$

This can be posed as a *Time Series Regression problem*.

To solve the above we would be using data collected in Jan - Mar 2015 to predict the pickups in Jan - Mar 2016.

4. PERFORMANCE METRIC

4.1 Mean Absolute Percentage Error

Mean absolute percentage error (MAPE) is a statistical measure that is used to evaluate the accuracy of predictions or forecasts. It is a widely used method for evaluating the performance of forecasting models in various fields such as economics, finance, engineering, and science.

The MAPE is a relative measure of the forecasting accuracy that calculates the percentage difference between the actual and predicted values. This metric is used to measure the magnitude of errors in a prediction model. The MAPE is calculated as the average of the absolute differences between the actual and predicted values, expressed as a percentage of the actual values.

For example, if the actual value of a variable is 100 and the forecasted value is 80, the absolute error is 20 (|100-80|), and the percentage error is 20% (20/100). If the actual value is 0, then the percentage error is undefined. To avoid this, a common approach is to replace the denominator with the average of the actual values, known as the mean absolute percentage error.

The MAPE Formula Can Be Expressed as Follows:

MAPE = (1/n) * Σ(|(actual - predicted)/actual|) * 100% ; where n is the number of observations, actual is the actual value, and predicted is the predicted value. The absolute difference between the actual and predicted values is divided by the actual value, and the result is multiplied by 100% to obtain the percentage error. This calculation is repeated for all the observations, and the average of the percentage errors is calculated to obtain the MAPE.

However, the MAPE has some limitations that should be taken into account when interpreting the results. One limitation is that it can be sensitive to extreme values or outliers in the data, which can inflate the error rate. For example, if there is a large outlier in the actual values, the MAPE may be high even if the forecasting model is accurate for the majority of the data. Therefore, it is important to interpret the MAPE in conjunction with other measures of accuracy, such as mean squared error or root mean squared error, and to use caution when interpreting results based on the MAPE alone.Therefore we will be using MAPE method as primary performance metric.

4.2 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is the process of analyzing data sets to summarize their main characteristics. The aim is to identify patterns and relationships in the data, and to understand its distribution and variability. EDA is a crucial step in the data analysis process, as it allows analysts to gain insights into the data, identify potential issues, and determine appropriate statistical techniques for further analysis.

The EDA process typically involves several steps, including data collection, cleaning, visualization, and statistical analysis. These steps are discussed in more detail below

In [1]:

```
import warnings
warnings.filterwarnings("ignore")

import dask.dataframe as dd  #read large csv files
import pandas as pd
import folium #plot maps
import datetime
import time #convert to unix time
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns

# this lib is used while we calculate the straight line distance
#between two (lat, lon) pairs in miles
from gpxpy.geo import haversine_distance#Get the haversine distance
import pickle
import os

%matplotlib inline
```

In [2]:

```
matplotlib.rcParams['figure.dpi'] = 100
```

In [3]:

```
jan_2015_dask_df = dd.read_csv('yellow_tripdata_2015-01.csv')
```

In [4]:

```
jan_2015_dask_df.head(5)
#Each row corresponds to one trip
```

Out[4]:

In [4]:

```
len_df = len(jan_2015_dask_df)
print(f"Number of trips in Jan 2015 data    : {len_df}")
print(f"Number of features in Jan 2015 data : {len(jan_2015_dask_df.columns)}")
print()
print(jan_2015_dask_df.columns)
```

```
Number of trips in Jan 2015 data    : 12748986
Number of features in Jan 2015 data : 19

Index(['VendorID', 'tpep_pickup_datetime', 'tpep_dropoff_datetime',
       'passenger_count', 'trip_distance', 'pickup_longitude',
       'pickup_latitude', 'RateCodeID', 'store_and_fwd_flag',
       'dropoff_longitude', 'dropoff_latitude', 'payment_type', 'fare_amou
nt',
       'extra', 'mta_tax', 'tip_amount', 'tolls_amount',
       'improvement_surcharge', 'total_amount'],
      dtype='object')
```

- We've 12M trips data and those are only from Jan 2015.

In [5]:

```
# However unlike Pandas, operations on dask.dataframes don't trigger immediate computat
ion,
# instead they add key-value pairs to an underlying Dask graph. Recall that in the diag
ram below,
# circles are operations and rectangles are results.

# to see the visulaization you need to install graphviz
# pip3 install graphviz if this doesnt work please check the install_graphviz.jpg in th
e drive
jan_2015_dask_df.visualize()
```

In [6]:

```
# in the data we have time in the format "YYYY-MM-DD HH:MM:SS" we convert this string t
o python time format and then into unix time stamp
# https://stackoverflow.com/a/27914405
def convert_to_unix(s):
    return time.mktime(datetime.datetime.strptime(s, "%Y-%m-%d %H:%M:%S").timetuple())
```

In [7]:

```
# we return a data frame which contains the columns
# 1.'passenger_count' : self explanatory
# 2.'trip_distance' : self explanatory
# 3.'pickup_longitude' : self explanatory
# 4.'pickup_latitude' : self explanatory
# 5.'dropoff_longitude' : self explanatory
# 6.'dropoff_latitude' : self explanatory
# 7.'total_amount' : total fair that was paid
# 8.'trip_duration' : duration of each trip
# 9.'pickup_times : pickup time converted into unix time
# 10.'Speed' : velocity of each trip

def return_with_trip_times(df):

    #fits the columns into memory for faster ops
    duration_cols = df[['tpep_pickup_datetime', 'tpep_dropoff_datetime']].compute()

    #pickups and dropoffs to unix time
    time_of_pickup = duration_cols['tpep_pickup_datetime'].apply(convert_to_unix)
    time_of_drop = duration_cols['tpep_dropoff_datetime'].apply(convert_to_unix)


    #load these cols into memory
    new_df = df[['passenger_count', 'trip_distance', 'pickup_longitude', \
                 'pickup_latitude', 'dropoff_longitude', 'dropoff_latitude', \
                 'total_amount']].compute()

    #add small offset(0.0001) to prevent zero division
    new_df['trip_duration'] = (time_of_drop - time_of_pickup) / 60.0
    new_df['pickup_time'] = time_of_pickup
    #multiply by 60 to convert time(minutes) to time(hours)
    #this way speed will be miles/hours
    new_df['speed'] = 60.0 * (new_df['trip_distance'] / new_df['trip_duration'])

    return new_df
```

In [8]:

```
jan_2015_df = return_with_trip_times(jan_2015_dask_df)
jan_2015_df.head(5)
```

In [9]:

```
num_trips_before_removal = jan_2015_df.shape[0]
print(f"Number of trips before removing outliers : {num_trips_before_removal}")
```

```
Number of trips before removing outliers : 12748986
```

.

Data Collection

The first step in EDA is to collect the data. Data can come from a variety of sources, such as databases, spreadsheets, or APIs. The data can be structured or unstructured and may be stored in different formats,

such as CSV, Excel, JSON, or SQL. It is important to ensure that the data is complete, accurate, and consistent before proceeding with the analysis.

Data Cleaning
The next step is to clean the data. This involves identifying and correcting errors, missing values, and inconsistencies in the data. Common data cleaning techniques include imputing missing values, removing outliers, and correcting data entry errors. Data cleaning is an iterative process, and may require multiple passes to ensure that the data is free from errors and inconsistencies.

Data Visualization
Once the data has been cleaned, the next step is to visualize it. Visualization techniques are used to explore the data and identify patterns and relationships. Common visualization techniques include histograms, scatter plots, box plots, and heat maps. Visualization can help identify trends and patterns in the data, and can be used to communicate findings to stakeholders.

Statistical Analysis
After visualizing the data, the next step is to conduct statistical analysis. This involves summarizing the data using statistical measures such as mean, median, and standard deviation. Statistical analysis can be used to test hypotheses and identify relationships between variables. Common statistical techniques include correlation analysis, regression analysis, and hypothesis testing.

Communication
The final step in the EDA process is to communicate thefindings. This involves presenting the data and analysis in a clear and concise manner, using visualizations and other techniques to communicate key insights. The communication should be tailored to the intended audience, and should provide actionable recommendations based on the analysis.

I. Data Cleaning
Data cleaning is a crucial step in the data analysis process that involves identifying and correcting errors, inconsistencies, and missing values in the data. The goal of data cleaning is to ensure that the data is accurate, complete, and consistent before proceeding with further analysis. In this article, we will discuss the process of data cleaning in more detail.

Step 1: Identify Missing Data
The first step in data cleaning is to identify missing data. Missing data can be due to a variety of reasons, such as data entry errors, system failures, or incomplete surveys. Missing data can be problematic because it can lead to biased results, reduce the power of statistical tests, and make it difficult to draw conclusions from the data.
There are several methods for identifying missing data. One common method is to create a missing data report that summarizes the number and percentage of missing values for each variable. This report can help identify variables that have a high proportion of missing values and may require further investigation.

Step 2: Impute Missing Data
Once missing data has been identified, the next step is to impute or fill in the missing values. There are several methods for imputing missing data, including:

1. Mean imputation: Replace missing values with the mean value of the variable.
2. Median imputation: Replace missing values with the median value of the variable.
3. Mode imputation: Replace missing values with the mode (most frequent) value of the variable.
4. Regression imputation: Use regression analysis to predict missing values based on other variables.
5. Multiple imputation: Generate multiple imputations of the missing values using statistical models.

The method of imputation used will depend on the nature of the data and the type of analysis being conducted. It is important to document the method of imputation used and to consider the potential impact of imputed values on the analysis results.

Step 3: Identify Outliers

Outliers are data points that are significantly different from the other data points in the sample. Outliers can be due to data entry errors, measurement errors, or other factors. Outliers can have a significant impact on the analysis results, and it is important to identify and handle them appropriately.

There are several methods for identifying outliers, including:

1. Box plots: Plot the data using box plots and identify any data points that fall outside the whiskers.
2. Z-scores: Calculate the z-score for each data point and identify any data points with z-scores greater than a certain threshold (e.g., 3).
3. Tukey's method: Use Tukey's method to identify outliers based on the interquartile range.

Once outliers have been identified, the next step is to decide how to handle them. Depending on the nature of the data, outliers may be removed, transformed, or kept in the data set.

Step 4: Check for Duplicates

Duplicates are data points that are identical in all variables. Duplicates can be due to data entry errors, system failures, or other factors. Duplicates can lead to biased results and reduce the power of statistical tests.

To identify duplicates, the data set can be sorted by one or more variables and checked for identical values. Once duplicates have been identified, they can be removed from the data set or combined into a single data point.

Step 5: Check for Inconsistencies

Inconsistencies are data points that are contradictory or violate logical constraints. Inconsistencies can be due to data entry errors, measurement errors, or other factors. Inconsistencies can lead to biased results and reduce the power of statistical tests.

To identify inconsistencies, the data can be checked for logical constraints. For example, if the data set contains information on the age and gender of participants, it can be checked for inconsistencies

The sub observations under data cleaning are as follows-

Pickup latitude and longitude

Dropoff Latitude &Dropoff Longitude

Trip Durations

Speed

Trip Distance

Total Fare

Ii.Data Preparation

Data preparation is the process of transforming raw data into a format that can be easily analyzed by a machine learning algorithm or other data analysis tools. This process involves several steps, including data cleaning, data integration, data transformation, and data reduction. In this article, we will explore these steps in more detail and provide tips for effectively preparing data for analysis.

Data Cleaning

Data cleaning is the process of identifying and correcting errors in the data, such as missing values, duplicates, or inconsistencies. The goal of data cleaning is to ensure that the data is accurate and complete, so that the results of any analysis are reliable. Some common techniques used in data cleaning include:

1. Removing duplicates: When the same data is recorded multiple times, it can create problems in analysis. By removing duplicates, you can ensure that each data point is unique.
2. Handling missing data: Missing data can occur due to a variety of reasons, such as incomplete surveys or technical issues. Depending on the extent of missing data, you can choose to either remove the data points or fill in the missing values using imputation techniques.

3. Correcting errors: Errors in the data can occur due to human error, technical issues, or other factors. By identifying and correcting errors, you can ensure that the data is accurate and reliable.

Iii.Data Integration

Data integration involves combining data from multiple sources into a single dataset. This is often necessary when working with large datasets that are spread across multiple databases or files. Some common techniques used in data integration include:

1. Joining tables: Joining tables involves combining data from two or more tables that share a common field. This is a common technique used in relational databases.
2. Merging datasets: Merging datasets involves combining data from two or more datasets that share a common variable. This is often used in data analysis to combine data from different sources.
3. Appending data: Appending data involves adding new data to an existing dataset. This is often used when new data becomes available after the initial dataset has been created.

Iv.Data Transformation

Data transformation involves converting the data into a format that is suitable for analysis. This often involves converting data into numerical values, so that they can be used in statistical analysis. Some common techniques used in data transformation include:

1. Scaling data: Scaling data involves converting data into a common scale, such as between 0 and 1 or -1 and 1. This is often used to standardize data that has different units of measurement.
2. Encoding categorical variables: Categorical variables are variables that take on a limited set of values, such as gender or ethnicity. These variables can be encoded as numerical values so that they can be used in analysis.
3. Feature engineering: Feature engineering involves creating new features from the existing data. This can include creating interaction terms, polynomial features, or other transformations that can improve the accuracy of the analysis.

V.Data Reduction

Data reduction involves reducing the size of the dataset while retaining as much information as possible. This is often necessary when working with large datasets that are too large to analyze directly. Some common techniques used in data reduction include:

1. Sampling: Sampling involves selecting a subset of the data to analyze. This can include random sampling, stratified sampling, or other techniques.
2. Dimensionality reduction: Dimensionality reduction involves reducing the number of variables in the dataset while retaining as much information as possible. This can include techniques such as principal component analysis (PCA) or t-distributed stochastic neighbor embedding (t-SNE).
3. Clustering: Clustering involves grouping similar data points together into clusters. This can be used to reduce the size of the dataset by analyzing each cluster separately.

The sub analysis of data preparation are mentioned as follows-
Clustering/Segmentation of NYC into Regions
Time Binning
Smoothing
Time series and Fourier Transforms

5. DATA MODELLING

Data modeling is the process of designing a representation of the data that will be stored, processed, and analyzed within a system. The purpose of data modeling is to ensure that the data is organized in a logical and efficient manner, which makes it easier to manage and analyze.
There are several types of data models, including conceptual, logical, and physical models. A conceptual data model describes the high-level relationships between different entities in a system. A logical data model takes the conceptual model and further refines it by adding more detail and structure, such as defining the attributes of each entity and the relationships between them. Finally, a physical data model describes how the data will be stored in a specific database or system, taking into account the technical details of the hardware and software. One of the primary benefits of data modeling is that it helps ensure data consistency and accuracy. By defining clear relationships between different entities, data modeling can help prevent errors and inconsistencies that can arise when different parts of a system store data in different ways. This can be especially important in larger systems, where many different people may be accessing and manipulating the data.Another important benefit of data modeling is that it can help improve system performance. By organizing the data in an efficient and logical manner, data modeling can help reduce the amount of processing power and storage space required to work with the data. This can result in faster and more efficient system performance, which can be especially important in high-traffic systems or those that require real-time processing.
There are several different approaches to data modeling, depending on the specific needs of the system. One common approach is entity-relationship modeling, which involves identifying the different entities within a system and their relationships to one another. Another approach is object-oriented modeling, which is often used in software development to represent the different objects and classes within a system.

Regardless of the approach, there are several key steps involved in the data modeling process. These typically include:
1. Identifying the data that will be stored and processed within the system.
2. Defining the relationships between different entities or objects within the system.

3. Refining the data model to include attributes and other details that will be necessary for data storage and analysis.
4. Creating a physical data model that specifies how the data will be stored within the system.

I. Cluster diagrams, also known as clustering diagrams, are graphical representations of data points that have been organized into clusters or groups based on their similarities. In other words, cluster diagrams are used to visually represent the clustering of data points into groups, where the members of each group are more similar to each other than they are to members of other groups

II-Simple Moving Average
SMA on Ratios
SMA on previous pickup densities
A simple moving average (SMA) is a technical analysis tool that is used to analyze financial data, such as stock prices, exchange rates, or commodity prices. It is a widely used indicator to track trends and identify potential buy or sell signals in the market. The SMA is calculated by taking the average of a specific number of closing prices over a specified period of time. For example, a 20-day SMA would be calculated by adding up the closing prices of the last 20 days and dividing the total by 20. The resulting value is the SMA for that day. SMA is a lagging indicator, which means that it uses past price data to generate signals. It can be used to identify the general trend of a security or asset, as well as support and resistance levels. When the price is above the SMA, it is considered a bullish signal, while when the price is below the SMA, it is considered a bearish signal.

III-Weighted Moving Averages
WMA on Ratios
WMA on previous pickup densities
A weighted moving average (WMA) is a type of technical analysis tool that is used to smooth out price movements of a security or financial instrument over a period of time. Unlike simple moving averages (SMA), which give equal weight to all data points, a weighted moving average gives greater weight to more recent data, resulting in a more responsive indicator.The calculation of a WMA involves multiplying each data point by a predetermined weight, summing the products, and then dividing the total by the sum of the weights. The weights are usually determined by a mathematical formula that assigns a higher value to the most recent data and a lower value to older data.

IV-Exponential Weighted Moving Averages
EMA on Ratios
EMA on previous pickup densities
Exponential Weighted Moving Averages (EWMA) is a statistical technique used to analyze time series data by giving more weight to recent observations while still considering past data points. It is a variation of the simple moving average, where each data point is assigned a weight that decreases exponentially as it moves further into the past.
The concept of EWMA is widely used in finance, engineering, economics, and other fields where trends over time are important.
How it works:
The EWMA method takes a weighted average of all the past data points, where the most recent data points are given more weight than the older data points. The weighting factor is determined by a smoothing parameter (λ) that ranges from 0 to 1.

To calculate the EWMA, the following formula is used:
EWMA = λ * Current Observation + (1-λ) * Previous EWMA
Where,
- λ is the smoothing parameter
- Current Observation is the most recent data point
- Previous EWMA is the exponentially weighted moving average of the previous period
The EWMA formula implies that the weight of the most recent observation is λ, and the weight of the previous EWMA is (1-λ). As λ approaches 1, the importance of the recent data points increases, and as λ approaches 0, the importance of the older data points increases.
The choice of λ is critical as it determines the level of smoothing and responsiveness of the EWMA to changes in the data. A higher λ value gives more weight to recent data points, making the EWMA more sensitive to changes, while a lower λ value gives more weight to older data points, making the EWMA less sensitive to changes.

4. CONCLUSION

In this we have used ML, AI and Data Analyitcs. The csv. File contains all the details of the New York City taxi details and maps and locations of certain regions of the city. This study has actually been used by the company for its own purpose in cab facilities for its employess. The code used in this project has been provided in the pdf and html file along with the diagrams and graphs.
Observations and result-
We have built a total of nine regression models using the pickup_densities. Some models used the time series property while the other models ignored it and was treated as a proper regression problem.
Out of which 6 of the models, moving averaging models were built using only Jan 2015 and Jan 2016 pickup densities.
The other three models were built using [Jan, Feb, Mar] 2016 pickup densities.
The results came out pretty well, Even Linear Regression was able to achieve a similar MAPE to XGBoost.Out the features we engineered, the most weighted feature/the feature which had most impact on the outcome was the pickupdensity followed by the EMA Model using previous pickup_densities.
Predicting taxi demand is a common use case of machine learning, AI, and data analytics in the transportation industry. In this example, we will use Python to build a machine learning model to predict taxi demand.

Here are the steps to build a taxi demand prediction model using ML, AI, and data analytics:
1. Collect and clean data: Collect historical taxi demand data and clean the data by removing any missing values and
outliers.
2. Feature engineering: Create new features that can be used to predict taxi demand, such as time of day, day of the week, and weather conditions.
3. Split the data: Split the data into training and testing sets. The training set will be used to train the machine learning model, and the testing set will be used to evaluate the model's performance.
4. Train the model: Use a machine learning algorithm, such as Random Forest or XGBoost, to train the model on the training set.
5. Evaluate the model: Use the testing set to evaluate the model's performance. Common evaluation metrics include Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).
6. Deploy the model: Once the model is trained and evaluated, it can be deployed to predict taxi demand in real-time.

5. REFERENCES

1. Taxi Demand Prediction Based on a Combination ForecastingModel in HotspotsZhizhen Liu,1Hong Chen ,1Yan Li,2and Qi Zhang
2. Jun Xu, RouhollahRahmatizadeh, LadislauBoloni and DamlaTurgut. ”Real-time Prediction of Taxi Demand Using Recurrent Neural Networks”, IEEE, 2017
3. IouliaMarkoua *, Filipe Rodriguesa , Francisco C. Pereiraa Multi-step ahead prediction of taxi demand using time-series and textual data”, IEEE,2018.
4. Daqing Zhang, Nan Li, Zhi-Hua Zhou, Chao Chen†, Lin Sun, ShijianLi ”: Detecting Anomalous Taxi Trajectories from GPS Traces”,IEEE,2011.
5. UkrishVanichrujee, TeerayutHoranont, WasanPattara-atikom, ThanarukTheermunkong, Takahiro S ”Taxi Demand Prediction using Ensemble Model Based on RNNs and XGBOOST”, IEEE,2018.
6. JuntaoWang ,Xiaolong Su ”An Improved K-Means Algorithm”, IEEE,2018.
7. N. J. Yuan, Y. Zheng, L. Zhang, X. Xie, ”T-finder: A recommender system for finding passengers and vacant taxis”, IEEE,2013
8. Taxi Demand Prediction using ML; Authors: Dr. A. VenkataRamana, AsiyaBatool, ManishaRamavath , PindrathiViveka