NYC – Yellow Taxi Prediction  
An End-to-End Approach

Problem Statement:

Given location coordinates (latitude & longitude) and time as input, the goal is to build a model which can predict the number of pickups by an yellow taxi in the query region and surrounding regions (Hint : Time-series Forecasting and Regression).

Data Description:

Dataset consists of 3 types of taxi data. Yellow Taxi, FHVs (For Hire Vehicles) and Green Taxi: Street Hail Livery (SHL). For this task I are only considering yellow taxi data betIen Jan - Mar 2022 & Jan - Mar 2023.

*Data Download Link*:  
<https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page>

Data Dictionary Link: <https://www.nyc.gov/assets/tlc/downloads/pdf/data_dictionary_trip_records_yellow.pdf>

NYC yellow taxi trip data consists of detailed records on individual taxi rides. This data, provided by the NYC Taxi and Limousine Commission (TLC), includes information like pick-up and drop-off locations and times, trip distance, passenger count, fare details, and even payment methods.

***P.S****: Due to data privacy policies, the dataset now only contains LocationID instead of Latitudes and Longitudes. Since there is now way to calculate bounding boxes of these location IDs, I have made an assumption that the Lat/Long input in problem statement now refers to the LocationID*

Solution Approach

1. Exploratory Data Analysis(EDA):  
     
   Before diving into model building for NYC taxi trip prediction, a crucial step is Exploratory Data Analysis (EDA). EDA helps us understand the underlying structure of the NYC yellow taxi trip data, identify patterns and trends, and uncover potential issues.  
   1. Data Overview:   
      I begin by examining the data schema, identifying the features present and their data types (numerical, categorical, etc.). This provides a basic understanding of the information available.
   2. Descriptive Statistics:   
      For numerical features like trip distance, fare amount, and passenger count, I have calculated summary statistics like mean, median, standard deviation, minimum, and maximum values. This helps me gauge the spread and potential outliers within the data.
   3. Data Visualization:   
      Visualizations play a key role in EDA. I have created boxplots to explore the distribution of features and identify potential outliers in the data.
   4. Feature Derivation:  
      Based on available data, I’ll derive some extra relevant variables like trip duration, speed and total fare that can help us identify outliers in data.
2. Data Processing:  
   1. Outlier Removal:  
      Since I have determined from EDA that the data consists of values at extremities acting as outliers, I’ll be removing them as it will hinder the performance of our model.
   2. Timestamp conversion:  
      Our model cannot take a timestamp object as input. Hence, I’ll be converting the timestamp into unix time stamp, and bin the entire dataset into intervals of 10 mins so that I can predict the number of pick ups within that time frame.
   3. Dealing with Missing Values:  
      Upon Visualization, I have identified that the data is very sparsely populated with respect to the number of pickups in 10 minute intervals per Region/Location ID. Hence, I need to fill these values with zeros. I haven’t performed any smoothing technique on data due to the sparsity. Forcefully smoothing will lead to poor model performance in this case.
3. Data Modelling:  
     
   The objective is to predict a quantity in the future. The quantity may also exhibit some periodical trends based on time attributes. Hence the best way to solve this problem is by using Time Series Analysis to obtain predictions based on moving averages of previous intervals. We can further use these averages as a feature in Regression techniques which will help model accuracy.  
     
   1. Time Series Analysis:  
        
      I have explored 3 time series analysis techniques (SMA, WMA, EMA) and chose Exponential moving averages technique (EMA). Here’s why:  
        
      **SMA (Simple Moving Average**): This is the most basic approach. It adds all the data points in the chosen period and divides by the number of points. SMA treats all data points equally.  
        
      **WMA (Weighted Moving Average):** WMA assigns weights to each data point, with more recent data points receiving higher weights. This gives WMA a better idea of recent trends compared to SMA.  
        
      **EMA (Exponential Moving Average):** EMA applies an exponential weighting scheme, where the weight decreases exponentially with each older data point. This puts significantly more emphasis on the most recent data compared to WMA and SMA.  
        
      For predicting taxi trips, EMA is the preferred choice due to its focus on recent trends:  
        
      Taxi demand fluctuates significantly:   
      EMA's strong weighting on recent data allows it to capture these short-term changes effectively, leading to more accurate predictions.  
        
      Responds to changing patterns:   
      Since taxi trip patterns can evolve over time, EMA's adaptability to recent trends proves valuable.  
        
      While WMA also considers recent data, EMA's exponential weighting places an even stronger emphasis on the most recent information, making it ideal for capturing the dynamic nature of NYC taxi trip data.  
        
      The Emphasis is being on placed on most recent information because unlike some data sets with gradual changes, NYC taxi trip demand experiences significant fluctuations throughout the day, week, and even year. Factors like rush hour commutes, weekend nights, holidays, and weather events can cause sharp increases or decreases in demand compared to typical patterns.
   2. Regression Model:  
        
      To perform regression, I chose a tree-based regressor because it offers greater flexibility and can capture complex relationships between features, potentially leading to more accurate predictions. I have explored the following techniques:  
       **Random Forest:**  
      Ensemble of Decision Trees: Builds multiple independent decision trees on random subsets of data and features. Final prediction is the average of these individual tree predictions.  
        
      Strength: Handles complex relationships, robust to missing data, interpretable results due to individual trees.  
        
      Weakness: Can be computationally expensive to train, prone to overfitting if not carefully tuned.  
        
      **XGBoost (Extreme Gradient Boosting):**  
        
      Sequential Ensemble: Builds decision trees sequentially, with each tree focusing on improving the errors of the previous one.  
        
      Strength: More powerful learning algorithm, can handle complex interactions and missing data, often leads to higher accuracy.  
        
      Weakness: More complex to tune than Random Forest, can be less interpretable due to the sequential boosting process.  
        
      While both Random Forest and XGBoost are strong contenders, XGBoost's focus on accuracy, sparse data handling, and built-in regularization (to avoid overfitting) make it a potentially better choice for capturing the complexities of NYC taxi trip data.
4. Error Metrics:  
     
   For evaluating the model metrics I have used 2 techniques:  
     
   **MAPE:**   
   Mean Absolute Percentage Error (MAPE) expresses the average prediction error as a percentage of the actual value. A 10% MAPE signifies an average prediction that deviates from the real value by 10%. This makes it easy to interpret the model's performance, especially for metrics like taxi fare where percentages are meaningful. Additionally, MAPE is less sensitive to outliers in the data, making it a good choice for taxi trip predictions.  
    **MSE:**Mean Squared Error (MSE) represents the average squared difference between actual and predicted values. While a lower MSE indicates better prediction accuracy, the squared error itself isn't directly interpretable. However, MSE is always non-negative and allows for easier comparison of model performance across different datasets, regardless of the value ranges. This can be beneficial for general model evaluation. However, MSE is sensitive to outliers, meaning a single very different prediction can skew the overall error measure.
5. Deployment:  
     
   The model is now ready for deployment. In order to make it an End to End solution, I have developed a basic flask app which gets LocationID and Pickup Datetime as input and responds with no of pickups as output. I have also built a docker image for quick deployment of the project.