SquareShift Technologies Inc

SQUARESHIFT

# Chicago Taxi Driver Services : Predicting Driver tips using Machine Learning

# TABLE OF CONTENTS

## Objective

To predict individual customer tipping patterns and behavior in taxi services, specifically focusing on predicting tip amounts using machine learning techniques. This analysis will help drivers and service providers optimize their strategies, improve service quality, and enhance customer engagement by accurately forecasting tipping behavior. It aims to increase driver satisfaction, promote fairness, and maximize earnings while fostering customer loyalty. Additionally, it highlights the potential of machine learning to revolutionize decision-making in the transportation industry by offering actionable insights for sustainable growth, equitable practices, and innovation.

## Predicting Driver tips for the Benefit of Driver’s Earnings

Understanding tipping behavior in the taxi industry presents a valuable opportunity to improve driver earnings and enhance the operational efficiency of service providers. Tips not only serve as a significant component of a driver’s income but also provide a lens into customer satisfaction and service quality. To achieve a balance between optimizing driver earnings and improving customer experiences and increasing the revenue for taxi service providers, it is crucial to analyze the factors that influence tipping patterns across various ride scenarios across the Chicago city for this analysis.

By leveraging historical data, this analysis uncovers trends and predicts tipping behavior, enabling taxi companies to achieve dual objectives: supporting drivers in maximizing their earnings and maximizing profits for taxi service providers. Through the use of machine learning, taxi service providers can:

- Maximize tipping opportunities by predicting the tips for different rides across the Chicago city.
- Tailor service recommendations and strategies to align with patterns in tipping behavior, such as time of day, location, or trip distance.

The overarching business goal is to empower drivers and service providers to improve performance and earnings through a data-driven understanding of tipping behavior. By delivering a transparent and equitable experience, machine learning ensures that taxi companies stay competitive and foster trust among drivers and customers alike.

## Dataset

The Chicago Taxi Services dataset, which captures customer and driver costs during the taxi rides from 2013 to the present, reported to the City of Chicago, includes 23 columns and around 211,655,459 rows, totaling approximately 76.75 GB. The large volume of data requires advanced analytical methods to uncover tipping trends, insights across various communities, and factors influencing taxi rides. The dataset and the columns it includes are shown in the table below.

| Field Name | Description and Example |
|---|---|
| unique_key | Unique identifier for the trip (Example: 5003bdd51918…) |
| taxi_id | A unique identifier for the taxi (Example: 2130bc5fd239..) |
| trip_start_timestamp | When the trip started, rounded to the nearest 15 minutes (Example: 2013-02-25 14:15:00 UTC) |
| trip_end_timestamp | When the trip ended, rounded to the nearest 15 minutes (Example: 2013-02-25 14:15:00 UTC) |
| trip_seconds | Time of the trip in seconds (Example: 120) |
| trip_miles | Distance of the trip in miles (Example: 0.04) |
| pickup_census_tract | The Census Tract where the trip began. For privacy, this Census Tract is not shown for some trips (Example: 17031320400) |
| dropoff_census_tract | The Census Tract where the trip ended. For privacy, this Census Tract is not shown for some trips (Example: 17031839100) |
| pickup_community_area | The Community Area where the trip began (Example: 32 for a specific mapping of community area) |
| dropoff_community_area | The Community Area where the trip ended (Example: 30 for a specific mapping of community area) |
| fare | The fare for the trip (Example: 4) |
| tips | The tip for the trip. Cash tips generally will not be recorded (Example: 1) |
| tolls | The tolls for the trip (Example: 0.5) |

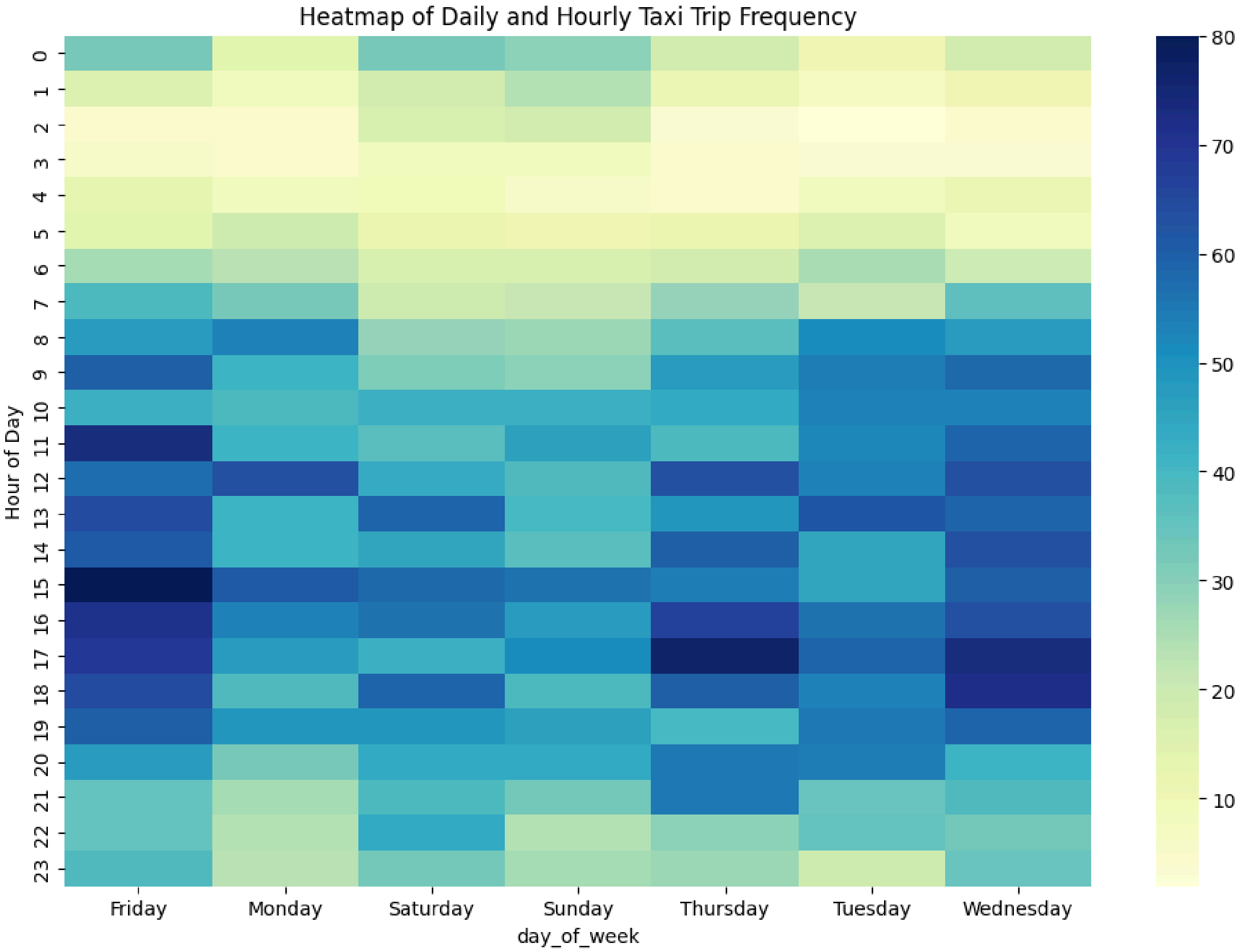
| | |
|---|---|
| extras | Extra charges for the trip (Example: 0.25) |
| trip_total | Total cost of the trip, the total of the fare, tips, tolls, and extras (Example: 4.25) |
| payment_type | Type of payment for the trip (Example: Cash) |
| company | The taxi company (Example: 303 Taxi) |
| pickup_latitude | The latitude of the center of the pickup census tract or the community area if the census tract has been hidden for privacy (Example: 41.97907082) |
| pickup_longitude | The longitude of the center of the pickup census tract or the community area if the census tract has been hidden for privacy (Example: -87.9030) |
| pickup_location | The location of the center of the pickup census tract or the community area if the census tract has been hidden for privacy (Example: POINT(41.979,-87.9030) |
| dropoff_latitude | The latitude of the center of the dropoff census tract or the community area if the census tract has been hidden for privacy (Example: -87.9030) |
| dropoff_longitude | The longitude of the center of the dropoff census tract or the community area if the census tract has been hidden for privacy (Example: 41.97907082) |
| dropoff_location | The location of the center of the dropoff census tract or the community area if the census tract has been hidden for privacy (Example: POINT(-87.9030,41.979)) |

## Data Exploration: Provide Insights for Taxi Companies

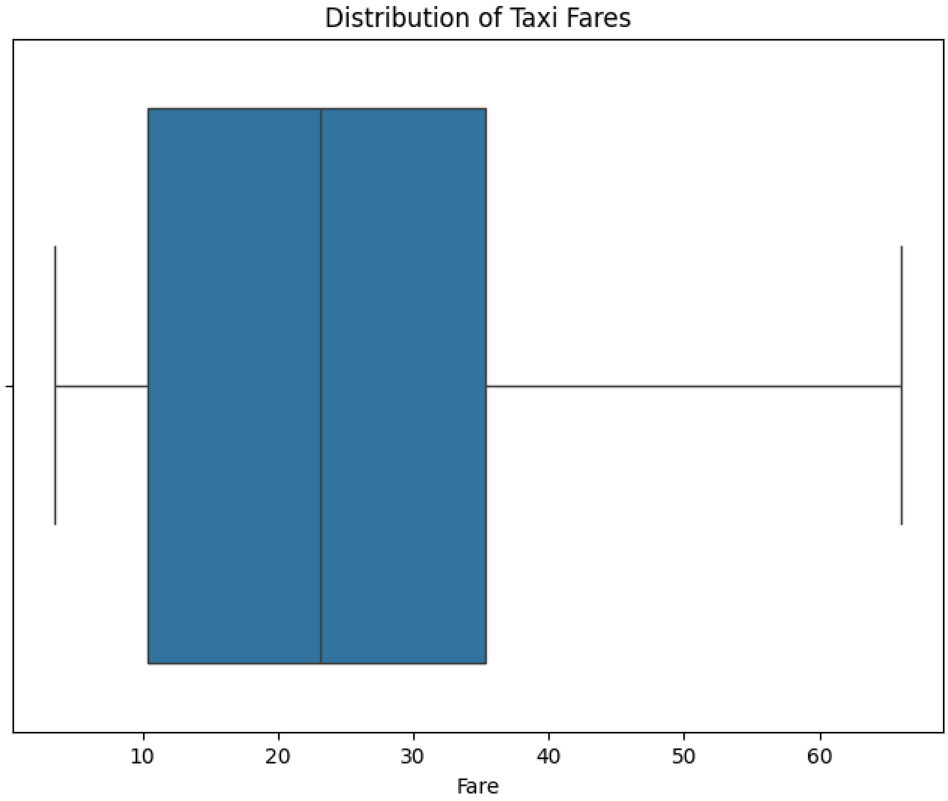
A thorough rundown of the several data exploration techniques utilized to find hidden patterns and trends in the dataset is given in this section. Descriptive statistics, trend analysis, and comparisons were among the methods used to gain understanding of fare tactics, consumer behavior for tipping, and the dynamics of the taxi industry. These are the main conclusions we drew from the examination of our thorough data research.

**Insights over Popular Pickup Spots and High Demand Timings**:

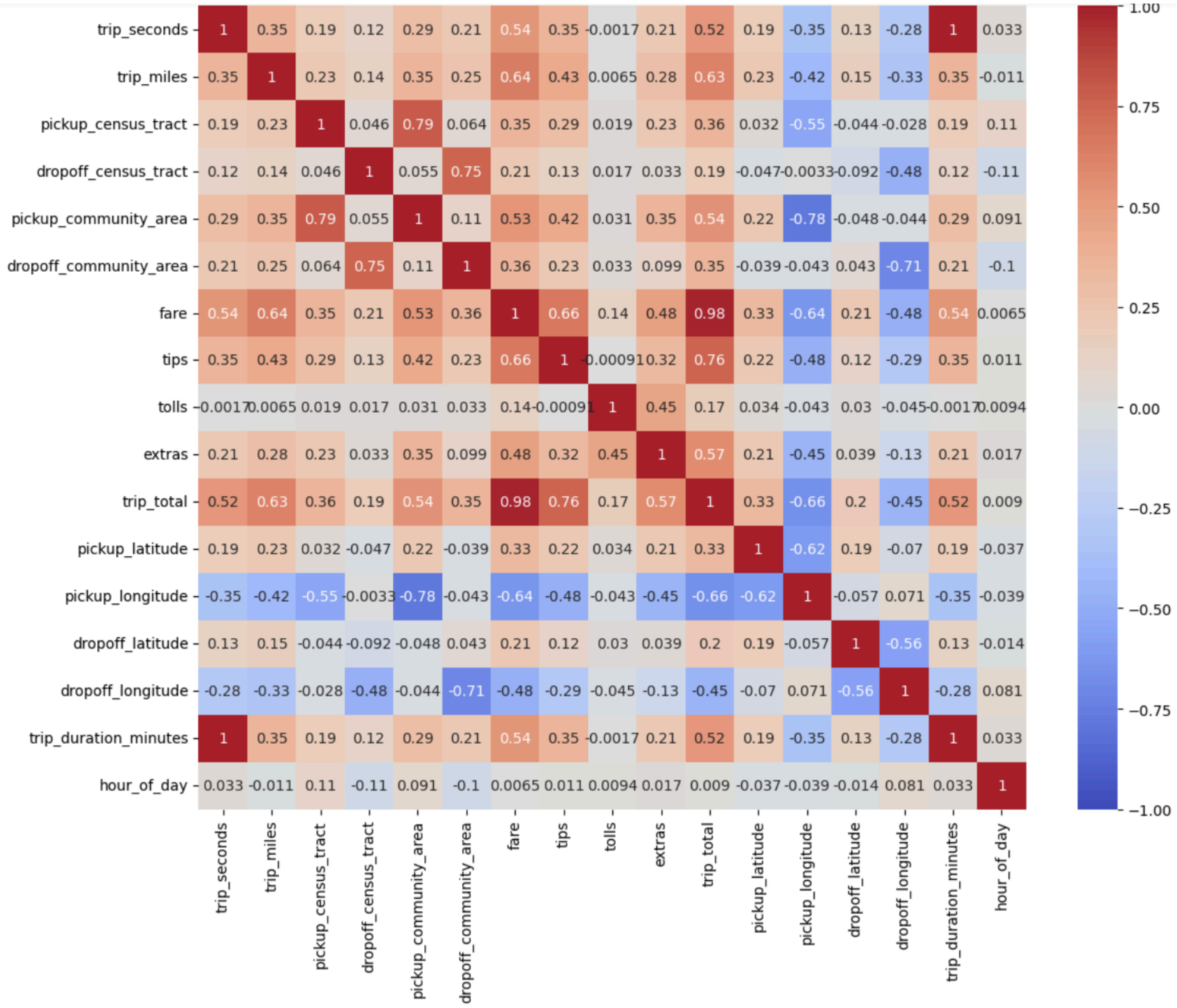
Analysis revealed that rides are busy at morning and evening hours and high usage of taxis towards the weekend.



**Distribution of Taxi Fares for Maximizing Revenue** :
This analysis offers a clear overview of the central tendency, variability, and skewness of fare data, providing quick insights into typical fare ranges and identifying any unusual pricing patterns that may impact tip predictions.

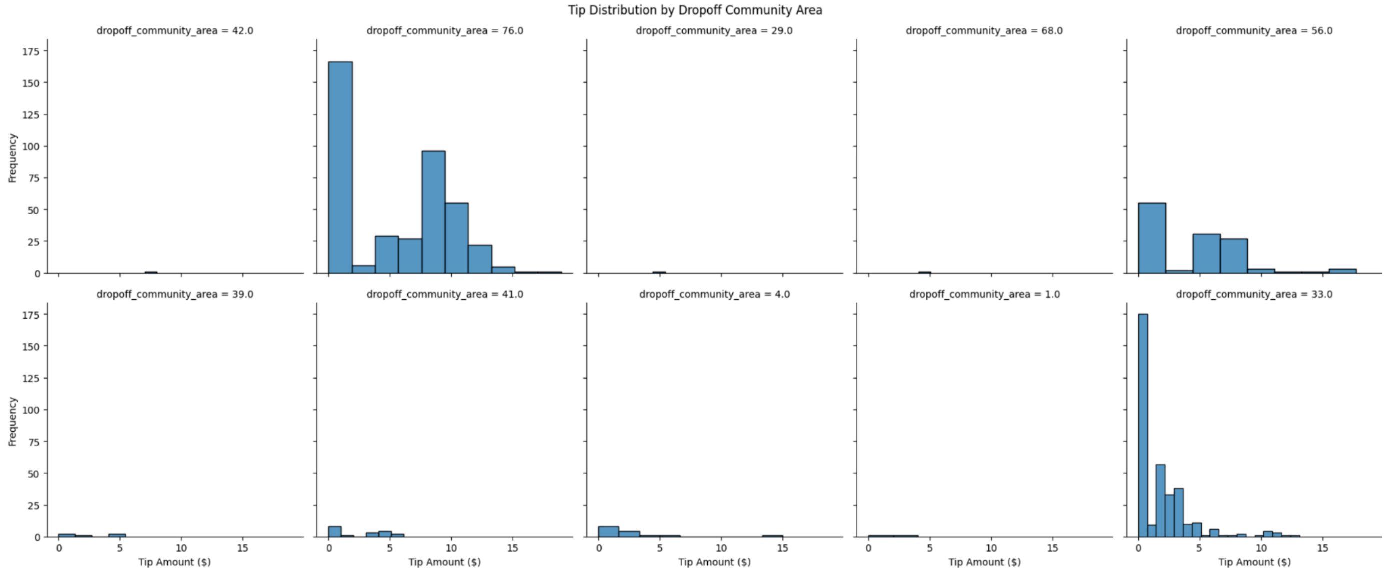


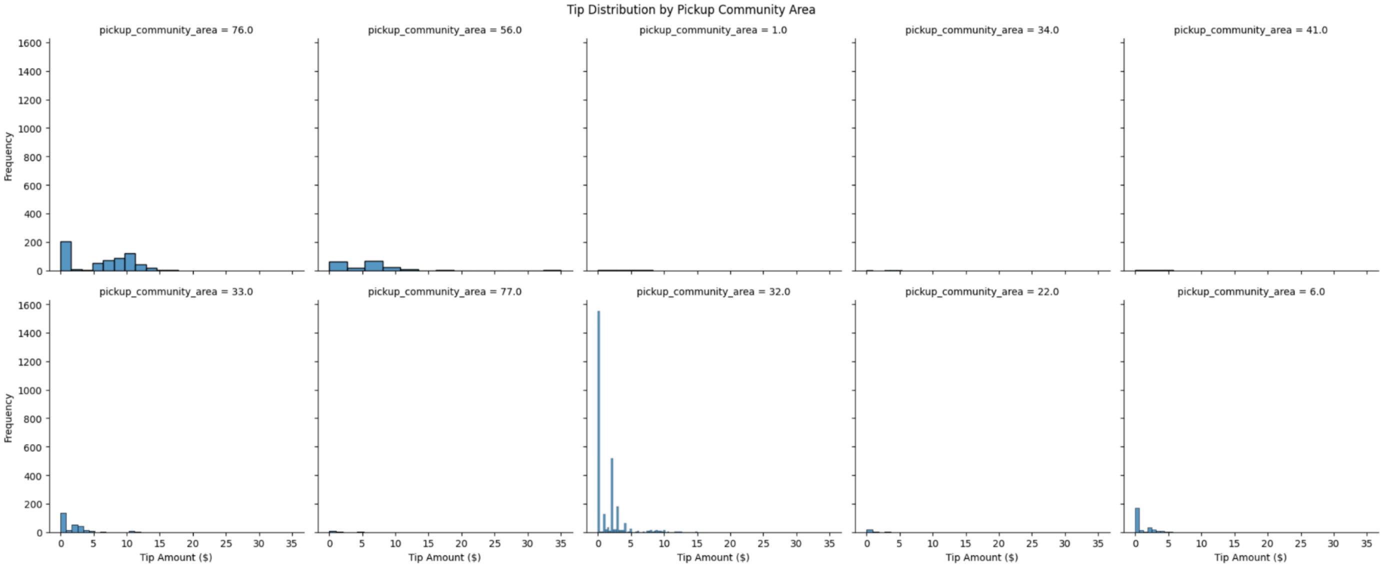
**Features Correlation Matrix**:
Using color intensity to show correlation coefficients, a correlation heatmap will show us the correlations between several variables in the dataset. In order to guide feature selection and provide insights into the elements that most influence tipping behavior in taxi rides, this robust tool enables you to rapidly determine which features are most strongly connected with tip amounts and detect potential multicollinearity among predictors.



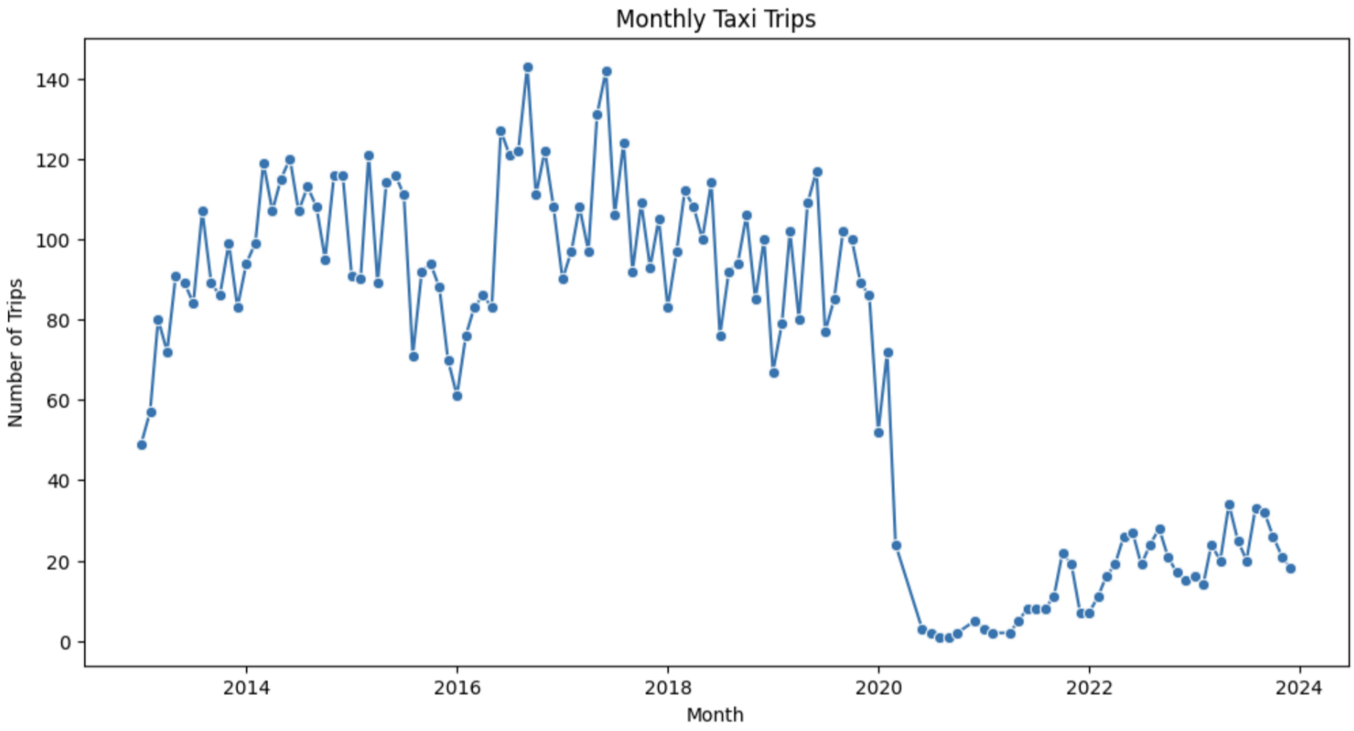
On analysis we can see that there is stronger correlation between fare and trip_total, trip_miles features which are bright in color and we can see that the tips are weakly correlated with tolls and drop off locations.

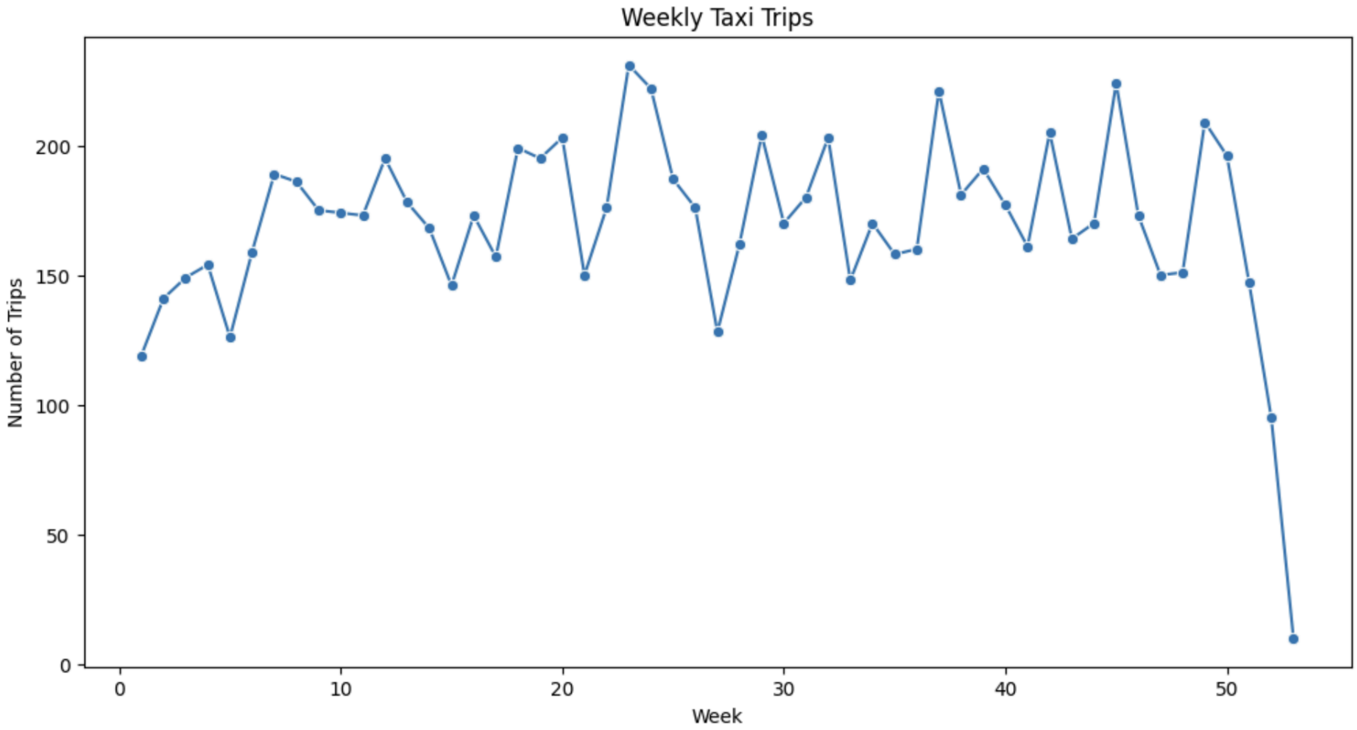
**Tip Amounts Variation across Different Communities:**

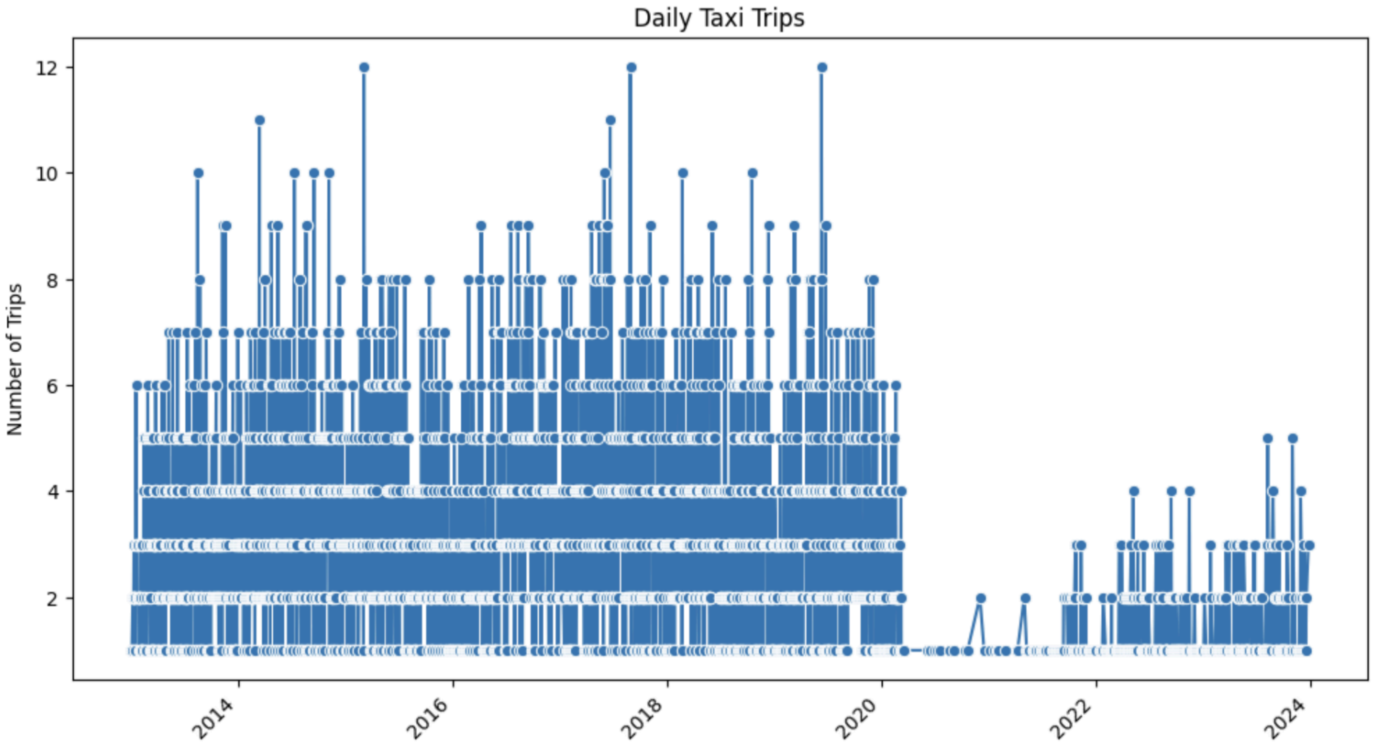




**Number of Rides that differ for every month, every week and every day:**

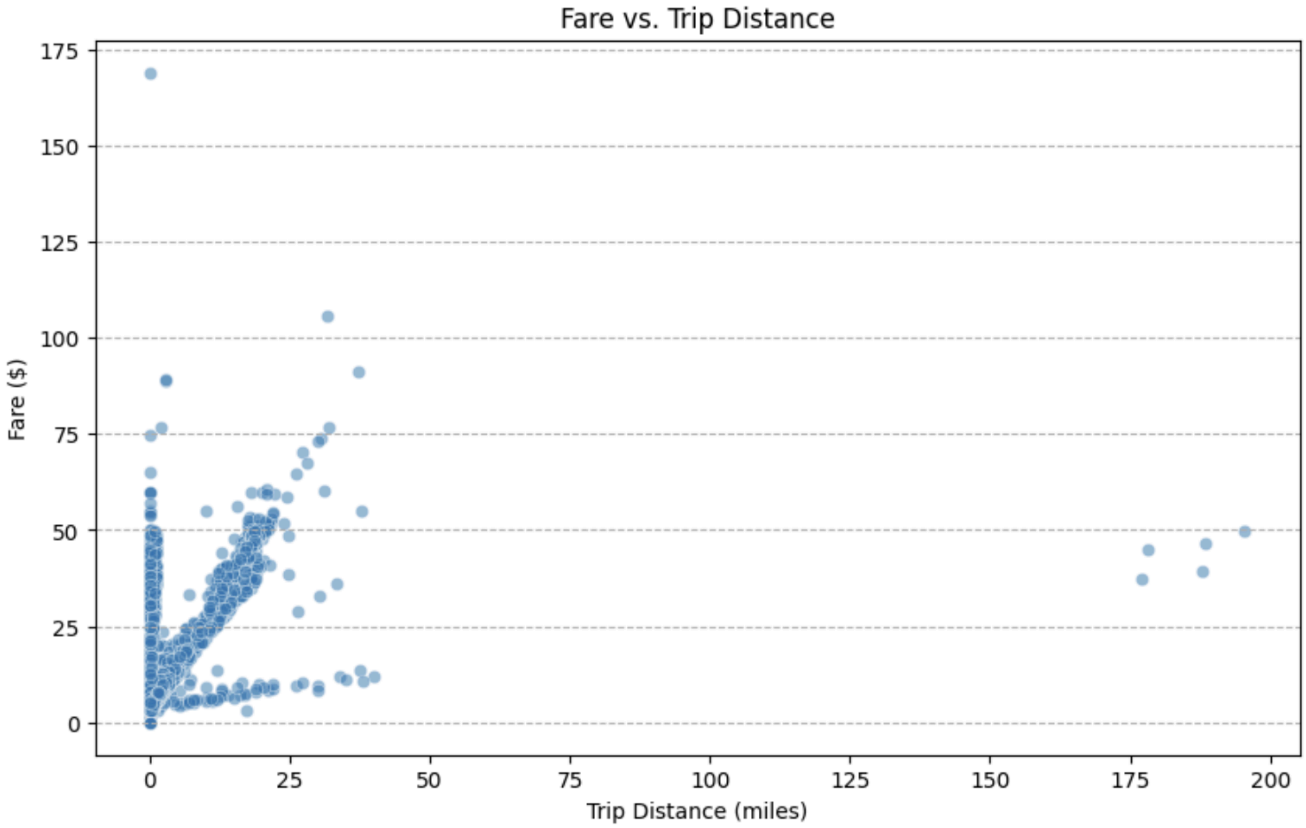

Monthly Taxi Trips
Number of Trips
0
20
40
60
80
100
120
140
Month
2014
2016
2018
2020
2022
2024

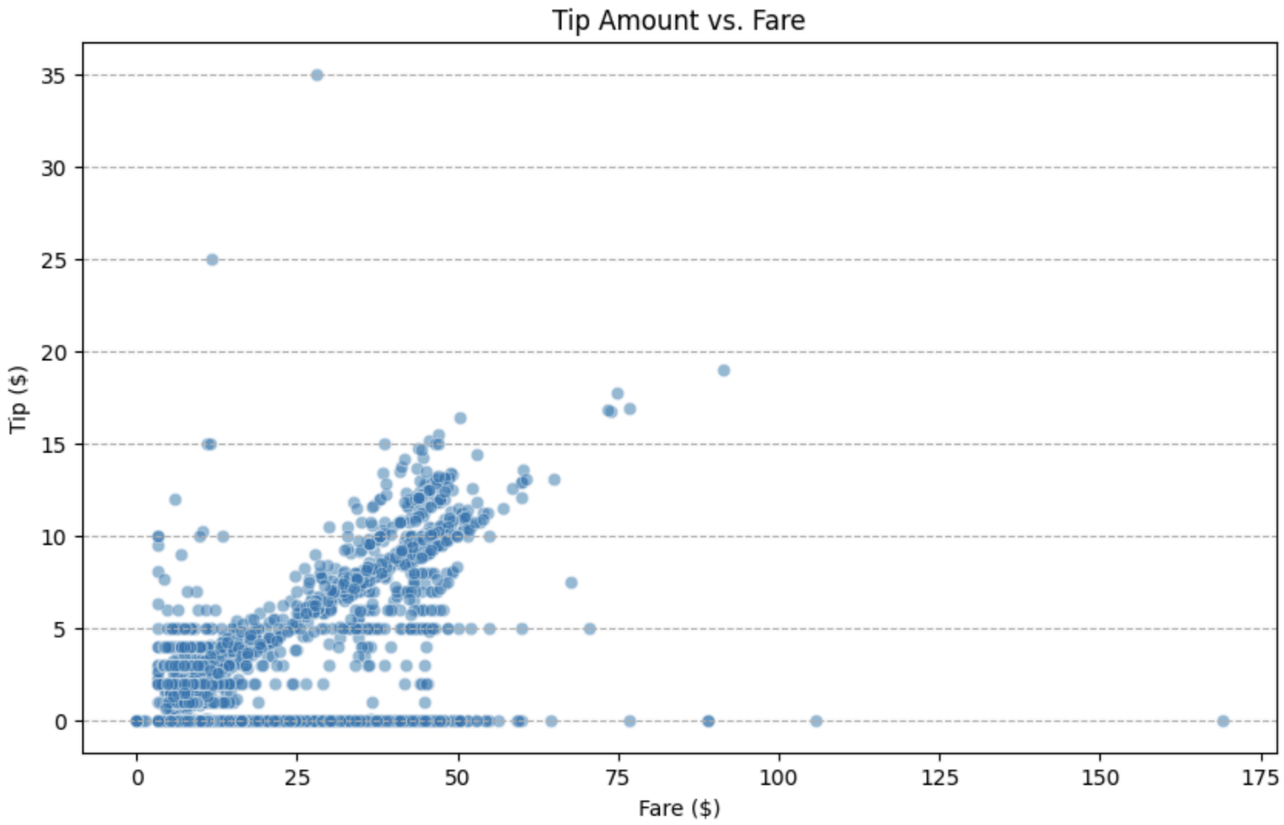

Weekly Taxi Trips
Number of Trips
0
50
100
150
200
Week
0
10
20
30
40
50



Based on daily, weekly, and monthly trends, the frequency of taxi journeys varies. Seasonal trends are frequently reflected in monthly fluctuations, such as higher ride counts during tourist seasons or holidays. Weekly trends often indicate that demand is higher on weekends than on weekdays, primarily due to social gatherings and recreational activities. Daily variances draw attention to times when commuting demands are greatest, such as morning and evening rush hours. Optimizing resource allocation and enhancing service efficiency require an understanding of these trends.

**Relationship between Two Features for Overview:**


Fare vs. Trip Distance
Fare ($)
Trip Distance (miles)
175
150
125
100
75
50
25
0
0
25
50
75
100
125
150
175
200



Analyzing relationships among features in a taxi trips dataset can reveal valuable insights into travel patterns and influencing factors. For example, examining the correlation between trip distance and fare amount can highlight pricing trends, while the relationship between pickup time and demand can identify peak hours. Additionally, studying the interaction between trip locations and tip amounts can uncover areas with higher tipping behavior, helping optimize driver strategies and service offerings. Such feature relationships provide a deeper understanding of the dataset, enabling more accurate predictions and informed decision-making.

## Feature Engineering Methods Used

A number of feature engineering techniques were used to make sure the data was reliable and machine learning-ready before it was used for predictive analysis.

- **Dropping Null Values:** Eliminating missing values from columns like ‘pickup_census_tract’ and ‘extras’ ensures that the analysis or model relies on complete and accurate data. Removing these entries helps prevent errors or biases that could arise from incomplete information.

- **Feature Construction:** Converting a couple of features into new features to utilize them for the machine learning model. Like we can convert the ‘trip_seconds’ from seconds to minutes to reduce the size of the feature.
- **Feature Scaling:** The range of features is normalized or standardized to bring them onto a similar scale. For instance, ‘trip_miles’ (trip distance in miles) and ‘fare’ amount (measured in dollars) might have vastly different scales. Scaling ensures that both features contribute equally to the model's learning process
- **Dimensionality Reduction:** Simplifying our data by transforming it into a smaller set of variables (principal components), making the data easier to visualize, analyze, and model. It also helps remove noise and redundancy, improving model performance and reducing computational cost.

## Feature Selection and its importance

Seven features 'trip_miles', 'fare', 'trip_total', 'trip_seconds', 'pickup_community_area', 'dropoff_community_area', 'trip_duration_minutes' have been chosen for the analysis because they are highly correlated with tip amounts, making them essential for predicting tipping behavior. These features have been identified as the most impactful due to their statistically significant relationship with tips. By concentrating on these key features, the accuracy and efficiency of the predictive model are enhanced. Feature selection is important as it simplifies the model, reduces the impact of irrelevant or weakly related features, and helps prevent overfitting. This method ensures that the model is easier to interpret and better equipped to generalize to new data, ultimately providing more reliable and actionable insights.

## Data Pre-Processing Pipelines

The data source was initially made available as a public dataset in Google BigQuery. The dataset was then fetched from BigQuery, and comprehensive feature engineering methods were applied to prepare it for model training. This process was carried out using a PySpark job on Google Cloud Dataproc, with all data preprocessing and feature engineering conducted through the scikit-learn library in Python.

Once the data was processed, the PySpark job was integrated with a Cloud Function API, which allows for dynamic parameterization. Parameters such as dataset size, table name, and dataset name are passed to the API, enabling flexibility in data processing.

After the data preparation, the model building, training, and testing phases were executed and wrapped as reusable APIs. These APIs, created as Cloud Functions, form the core of an end-to-end machine learning pipeline. This comprehensive pipeline was deployed on a Compute Engine Cluster, ensuring the smooth execution of all steps—from data ingestion to model creation, prediction, and evaluation—making the entire workflow automated and scalable.

## Machine Learning Models : Scikit-Learn and TensorFlow

Four Machine Learning Models: Polynomial Regression, Gradient Boosting, Support Vector Regression, and Artificial Neural Network were assessed in order to appropriately handle. Metrics like MSE, MAE, RMSE and R2 scores were used to evaluate each model's performance.

### Scikit-Learn Models

1. Polynomial Regression: Polynomial regression can capture non-linear relationships while remaining relatively simple. Unlike linear regression, which fits a straight line, polynomial regression fits a curve that can be more complex.
2. Gradient Boosting: This model reduces residual errors (or gradients) by concentrating on the areas where previous models underperformed. This iterative approach enhances prediction accuracy, making Gradient Boosting a robust and highly effective algorithm, particularly for handling complex datasets.

### Tensorflow Models

3. Support Vector Regression(SVR): This model is mostly used for regression tasks, where the goal is to predict a continuous value, which is tips in our case. SVR is based on the principles of Support Vector Machines (SVM), but instead of classifying data points, it aims to find a function that best fits the data within a specified margin of tolerance.
4. Artificial Neural Network(NN): This is an artificial Neural Network which is a more complex model than a machine learning model, so it can handle large data and more features which will help in more accurate predictions of our tips.

## Model Training and Validation

1. **Polynomial Regression:**

   This Machine Learning model pipeline includes standardizing the input features, generating polynomial features up to a specified degree, and fitting a linear regression model. The dataset was split into training and test sets with a ratio of 4 to 1, and the model was trained on the scaled training data, while predictions were evaluated on the test set. Multiple degrees of polynomial features (1 to 4) were tested to identify the optimal degree using cross-validation, with mean squared error (MSE) as the primary evaluation metric. The performance of each model was assessed using cross-validated MSE (CV MSE). The model with the lowest CV MSE was selected as the best model which is of degree 2.

2. **Gradient Boosting:**

   A Gradient Boosting Regressor model was implemented to capture complex relationships in the dataset. This model was trained using 100 estimators with a learning rate of 0.1 and a maximum tree depth of 3, optimizing for performance on scaled training data.

3. **Support Vector Regression(SVR):**

   It leverages the capability to model complex, non-linear relationships using a radial basis function (RBF) kernel. The model was trained on the scaled training data with a regularization parameter ( C ) set to 1.0 and an epsilon margin of 0.1, which balances the trade-off between accuracy and generalization. This approach highlights the SVR's strength in handling non-linear patterns while minimizing prediction errors within a defined tolerance.

4. **Artificial Neural Network(NN):**

   This Multi-Layer Perceptron (MLP) neural network is designed for accuracy and reliability in predicting continuous outcomes. The model architecture included three dense layers with 64, 32, and 16 neurons respectively, each using the ReLU activation function to effectively learn non-linear patterns. Dropout layers were added after the first two dense layers, with a dropout rate of 20%, to prevent overfitting by randomly deactivating neurons during training. The output layer consisted of a single neuron to generate predictions. The model was trained using the Mean Squared Error (MSE) loss function and the Adam optimizer, known for its efficiency in optimizing neural networks. Training was conducted over 20 epochs with a batch size of 64, and 20% of the training data was reserved for validation, enabling the monitoring of performance on unseen data and ensuring generalization.

## Model Evaluation and Selection

Finally evaluation of all of these models are done to pick the best model that can be used for different subsets of the dataset.

## Testing and Evaluation Metrics

These Models are tested upon on the 0.2 percent of data that is split for test and are compared based on the evaluation metrics Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and R-squared Score(R2 Score).

**Model Evaluation Metrics Comparison:**

| | Mean Squared Error | Mean Absolute Error | Root Mean Squared Error | R-squared Score |
|---|---|---|---|---|
| SVR | 1.7884 | 0.6880 | 1.3373 | 0.7413 |
| Neural Network | 2.4729 | 1.1905 | 1.5726 | 0.5508 |
| Gradient Boost | 1.0517 | 0.5913 | 1.0255 | 0.8504 |
| Polynomial Regression | 0.8477 | 0.5606 | 0.9207 | 0.8912 |

## Model Selection Criteria

Based on the regression model results, the SVR (Support Vector Regression) model emerges as the best choice for scaling to a 184 million row dataset in Google Cloud's Dataflow. While Gradient Boost shows slightly better accuracy with the highest R-squared score (0.8504), SVR's strong performance (R-squared 0.7413) combined with its superior scalability makes it more suitable for large-scale deployment. SVR offers a good balance of accuracy and generalization, handles high-dimensional data efficiently, and is well-suited for distributed computing environments. Its robustness to outliers, lower memory requirements compared to ensemble methods, and ability to be trained incrementally are crucial advantages when dealing with such a massive dataset. These characteristics make SVR the most practical and efficient choice for scaling up in Google Cloud's Dataflow architecture, despite the marginal trade-off in accuracy compared to Gradient Boost on the smaller dataset.

## Business Use case

The model predicts how much a driver is going to get a tip from historical data. Using this information, the taxi service provider can improve their business to generate more revenue and keep taxi drivers busy at the same time and also helps in the increase of taxi driver’s wages . Example Scenario: For a driver predicted to receive a tip of  $10 for a particular route and time will help me get to the busy locations and get more of the tips.

### Conclusion

This Machine Learning use case highlights the transformative potential of machine learning in predicting tipping behavior within the taxi service industry. By leveraging historical data, comprehensive feature engineering, and advanced predictive models, the analysis provides actionable insights to enhance driver earnings and service provider efficiency. Among the models evaluated, Support Vector Regression (SVR) emerged as the optimal choice for scalability and performance, making it suitable for deployment in large-scale cloud environments. These predictions not only enable taxi companies to strategize resource allocation and service improvement but also empower drivers with data-driven tools to maximize their earnings. Overall, this work demonstrates the critical role of machine learning in fostering innovation, equity, and sustainable growth in the transportation industry, delivering benefits to both service providers and their customers.