Uber Fare Prediction Using Random Forest Regression

Abstract

The task for this project is to predict Uber fares with the help of dataset provided on Kaggle. We want to build a robust predictive model that could predict the fare price for future rides given a combination of pickup and drop off locations, number of passangers, and the distance traveled. It was selected as the main model, the Random Forest Regressor, which can perform best at the complex relationship and give reliable result (Pedregosa, F., Varoquaux, G., Gramfort, A., et al, 2011). We ended up with a Mean Squared Error (MSE) of 22.02 and an R^2 score of 0.78, showing that it predicts well. This report touches on every step of the process, from EDA to preprocessing data to training and evaluating the model, to extracting insights.

Introduction

Background

One of the world's largest ride-sharing companies, Uber, processes millions of trips a day. Fare estimation that is accurate is crucial to sustaining operational efficiency. Fare prediction, however, is difficult due to dynamic factors such as traffic, time of day and location. Through the use of historical ride data machine learning models can accurately predict the fare, enabling Uber to optimize its pricing strategy and improve customer satisfaction. (Kaggle, 2021)

Objective

The primary objective of this project is to:

1. Understand the factors influencing fare prices.
2. Develop a predictive model capable of accurately estimating Uber fares.
3. Gain insights into key features contributing to fare variability.

Dataset Overview

The dataset used in this project contains information about Uber rides, including the following columns:

- **key**: Unique identifier for each trip.
- **fare_amount**: The cost of the trip in USD.
- **pickup_datetime**: Date and time when the trip started.
- **passenger_count**: Number of passengers.

- **pickup_longitude** and **pickup_latitude**: Geographical coordinates of the pickup location.
- **dropoff_longitude** and **dropoff_latitude**: Geographical coordinates of the dropoff location.

Data Preprocessing

Handling Missing Values

Missing values can compromise the quality of the analysis. In this dataset, columns with missing values were identified, and appropriate measures were taken:

1. **Fare Amount**: Rows with missing or negative fare values were removed.
2. **Passenger Count**: Outliers (e.g., passenger counts exceeding 6) were filtered out, assuming realistic limits for ride-sharing vehicles.

Removing Outliers

Outliers in fare amounts and geographical coordinates were removed:

1. **Fare Amount**: Trips with fares above $500 or below $1 were excluded, as they were likely to be outliers.
2. **Latitude and Longitude**: Coordinates outside valid ranges were filtered out to ensure data consistency.

Feature Engineering

1. **Datetime Features**:
    - The `pickup_datetime` column was converted into datetime format, and new features such as `hour`, `day`, `month`, and `weekday` were extracted.
2. **Distance Calculation**:
    - A new feature was calculated using the geodesic function from the geopy library, which represents a straight line distance between pickup and dropoff points in kilometers named distance_km.

Exploratory Data Analysis (EDA)

1. Distribution of Fare Amount

- Shape: It is heavily right skewed. Since most riders are low fare riders, a few high fare riders have disproportionately high average cost.
- Peak: We see that the distribution has a sharp peak in the lower fare range (below $50). This implies that the extent of rides that occurred within this range are high.

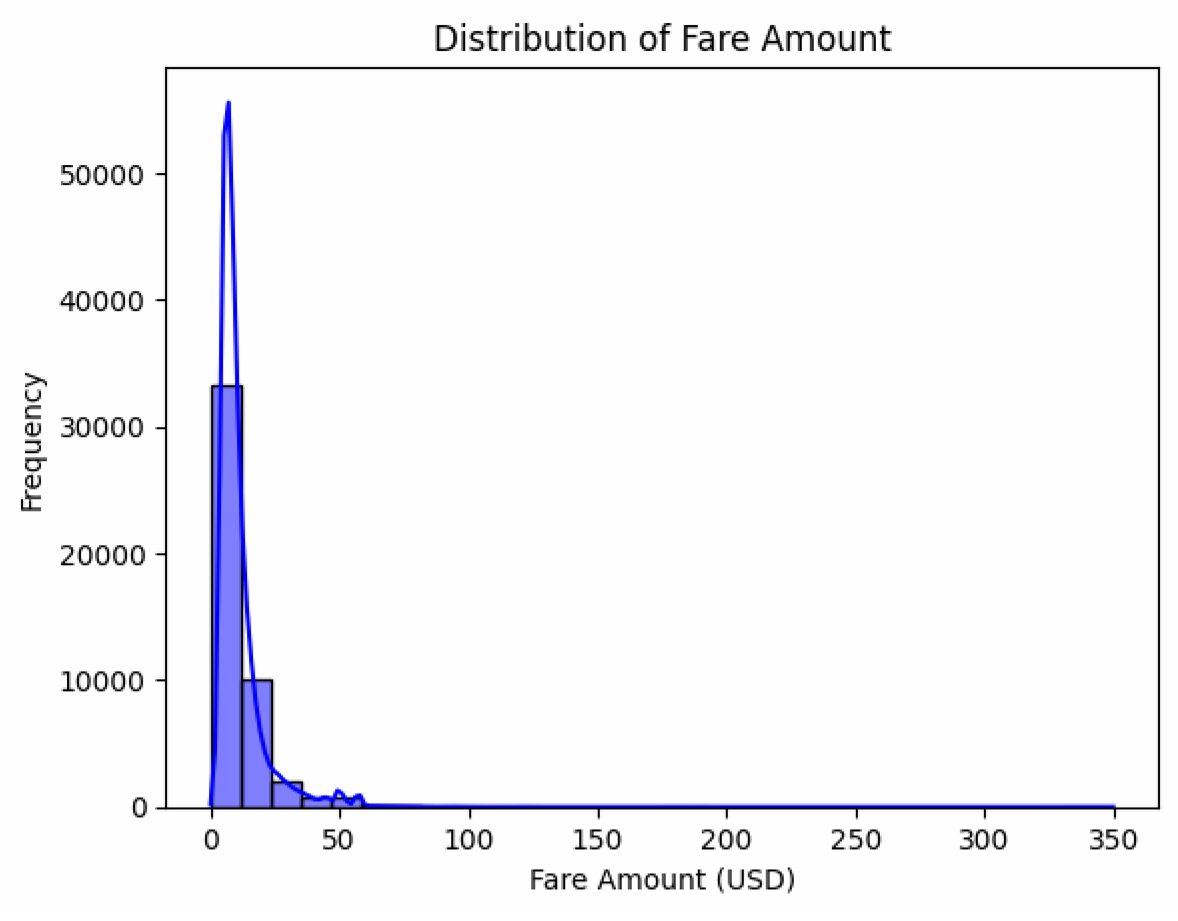
- Outliers: We see there's a long tail on the distribution with respect to fare amount, and so there must be some pretty expensive rides, which can, perhaps, be considered outliers. They may also be outliers for rides that took place at peak time or that involved charges on top of the base rate.

Insights:

Pricing Strategy: The distribution can be utilized to get useful insights regarding the pricing strategy for the transportation service. This implies that, at least for a sufficiently large number of rides, the majority of revenue will be from a large set of low fare rides and a small set of high fare rides contributes substantially to overall revenue.

Business Decisions: Understanding this distribution can help businesses make informed decisions, such as:

- **Setting price ranges:** Identifying appropriate fare ranges for different types of rides.
- **Identifying potential outliers:** Investigating the reasons for unusually high fares.
- **Improving pricing models:** Developing more dynamic pricing strategies that better reflect demand and cost factors.



2. Passenger Count vs. Fare

Passenger count had minimal impact on fare amount, as Uber charges fares based on distance and time rather than the number of passengers.

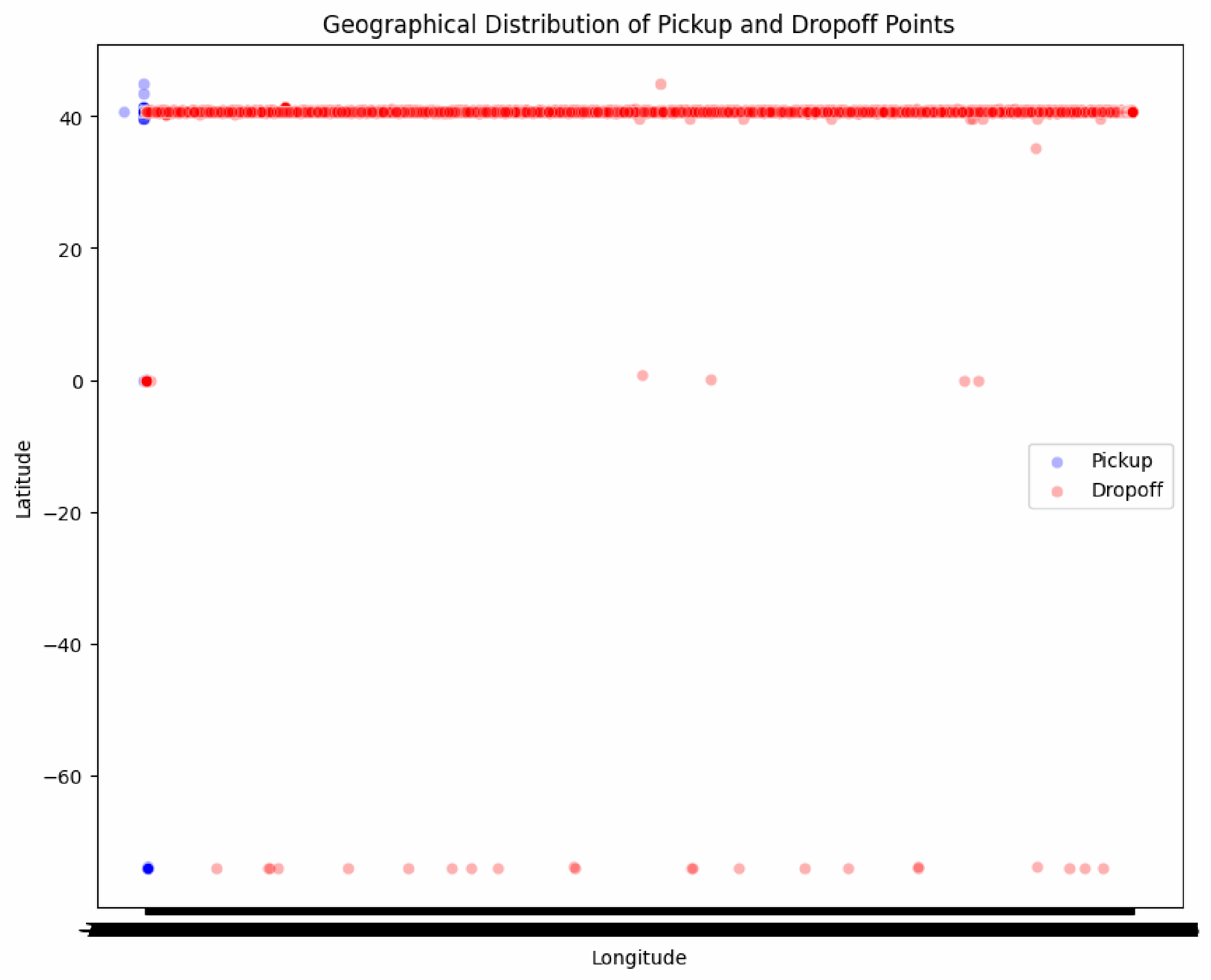
- Boxplot Structure: The distribution of fare amounts for a particular passenger count is represented in each box.
- Median Line: Within each box, the horizontal line is the median fare amount that passenger count fares.
- Box Limits: By first and third quartiles, I mean the bottom and top of the box, which cover the middle 50 percent of the fare amounts.
- Whiskers: Finally, the lines off of the box are the range of the data, without the outliers.
- Outliers: Any individual data points outside the whiskers are considered outliers meaning that the average price charged to the passenger for the same transport is unusually high.

Insights:

- Trend: It has been found that as the number of passengers growso does the fare amounts. That's to be expected; more passengers generally equal higher fares.
- Variability: As passenger count increases, the spread of fare amounts (box height) appears to also increase. It means that for bigger groups of passengers, there are more degree of variation in fares.
- Outliers: Outliers present indicate that there are certain rides with the same number of passengers that heavily charge fare compared to other rides. There are also outliers, which could be caused by the amount traveled, the time of day, or special requests.

3. Pickup and Dropoff Locations

- **Two Sets of Points:** The plot shows two distinct sets of points: one in blue representing pickup locations and another in red representing drop-off locations.
- **Clustering:** Both pickup and drop-off points seem to exhibit some clustering, suggesting that certain areas are more frequently used for transportation.
- **Longitude and Latitude:** The x-axis likely represents longitude, and the y-axis represents latitude, indicating the geographical coordinates of the locations.
- **Limited Latitude Range:** The visible range of latitude is relatively narrow, suggesting that the data might be focused on a specific region or city.



Model Evaluation

Metrics

The following metrics were used to evaluate model performance:

- **Mean Squared Error (MSE)**: Measures the average squared difference between predicted and actual values.
- **R² Score**: Indicates the proportion of variance in the target variable explained by the model.

Results

- **MSE**: 22.02

- **R²**: 0.78

The results indicate that the model performs well, explaining 78% of the variance in fare amounts and maintaining a relatively low prediction error.

Feature Importance

The top contributing features were:

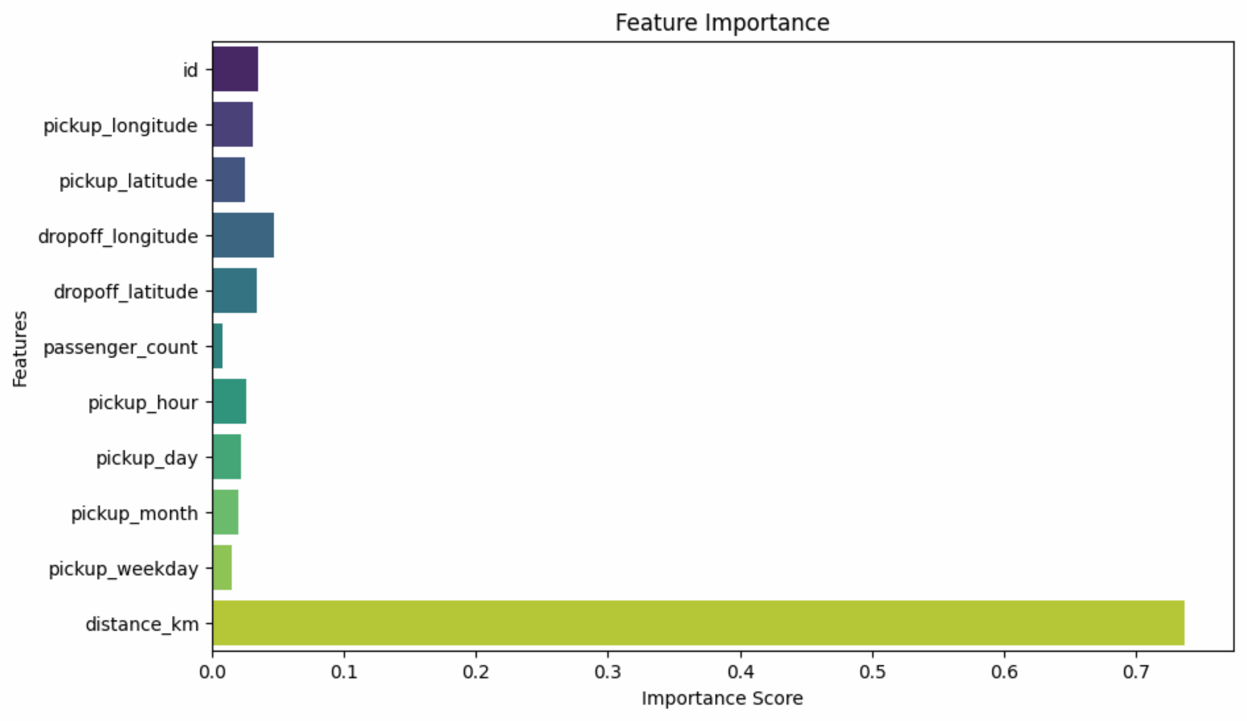
1. **Distance**: The most significant predictor of fare amount.
2. **Pickup Hour**: Reflects surge pricing during peak hours.
3. **Day of Week**: Indicates fare variations based on weekdays and weekends.

Visualization of Results

1. Feature Importance

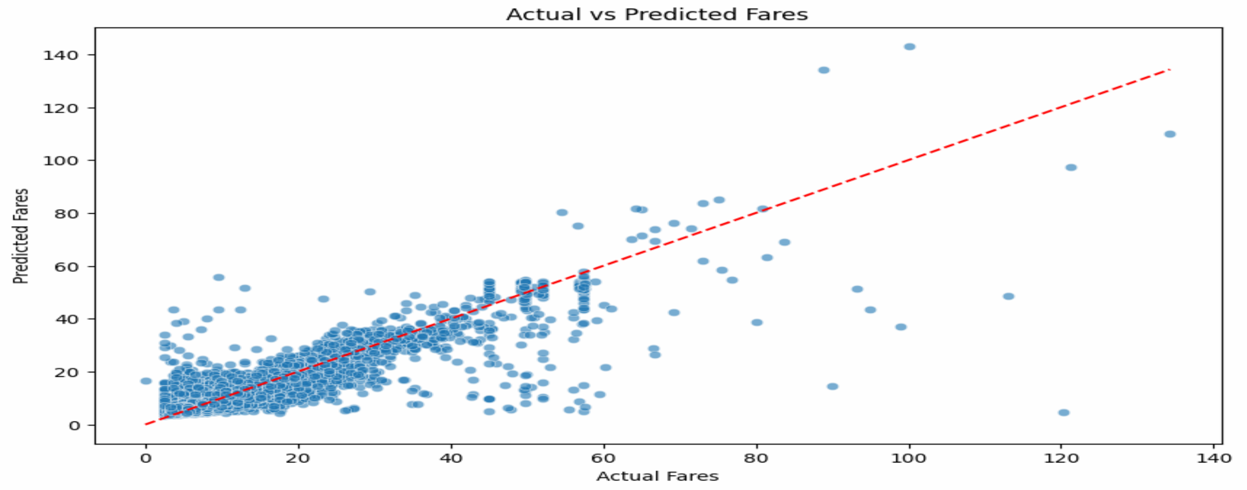
Feature importance was measured through a bar chart where distance_km was the most important feature around which time related features as pickup_hour had an influence.

According to the score of each bar, we have the most important feature as the length of distance_km and then some geographical coordinates (pickup and dropoff longitudes and latitudes). We found that passenger count and the time related features (pickup_hour, pickup_day, pickup_month, pickup_weekday) have moderate importance (i.e. lead to moderate change in predictions) and 'id' seems to have the least impact.



2. Actual vs. Predicted Fares

The distribution of this predicted versus actual fares on a scatter plot formed a good linear trend, and most of the points were close to the diagonal line which indicated good predictions. Discussion of the results includes comparing actual fares to the fares predicted by a machine learning model. The x coordinate of each point is actually the fare and the y coordinate is the predicted fare. The ideal scenario is represented by the red dashed line which represents the line that predicted fares perfectly match actual fares. In general, the points scatter around this line and we see both underpredictions and overpredictions, largely for higher fare amounts.



Insights and Business Applications

1. **Distance and Fare Relationship**:

- Distance was confirmed as the primary driver of fare prices. This insight reinforces the need for accurate distance estimation in fare calculation.

2. **Impact of Time**:

- Fares were higher during peak hours and weekends, highlighting the importance of dynamic pricing models.

3. **Urban Clusters**:

- High-density areas for pickups and dropoffs can be used to optimize driver allocation and reduce waiting times.

Conclusion

This project proved a success in using machine learning to make predictions of Uber fares. Random Forest Regressor proved to be robust predictors with an R² score of 0.78. Key takeaways include:

- Distance and time are critical factors in fare prediction.
- Outlier handling and feature engineering significantly improved model performance.

Future Work

1. **Incorporate Additional Features**:
   - Include weather data, traffic conditions, and ride demand to enhance predictions.
2. **Model Optimization**:
   - Experiment with advanced models like Gradient Boosting or neural networks for potentially better performance.
3. **Real-Time Deployment**:
   - Implement the model in a production environment to estimate fares in real time.
4. **Geospatial Analysis**:
   - Use clustering techniques to identify high-demand areas for better resource allocation.

References

1. Kaggle. (n.d.). Uber Fare Prediction Dataset. Retrieved from Kaggle.
2. (Pedregosa, F., Varoquaux, G., Gramfort, A., et al, 2011). (2011). Scikit-learn: Machine Learning in Python.
3. Haversine Formula and Geopy Library Documentation. Available at Geopy.

GitHub Guidelines:

1. **Fork the Repository**:

   Create your own copy of the project on GitHub.

   git fork

2. **Clone Your Fork**:

   Clone your forked repository to your local machine.

   git clone

3. **Create a Feature Branch**:

   Create a new branch to work on your feature or bug fix.

```
git checkout -b main
```

4. **Commit and Push Changes**:
   Save your changes and push them to your GitHub fork.

   ```
   git commit -m "Changes to be commited"
   ```

   ```
   git push origin main
   ```

5. **Submit a Pull Request**:
   Submit a pull request to the original repository for review and inclusion.

Licensing

The code and documentation of this project is governed under the MIT License, and to use, modify and distribute.

It allows free use. See LICENSE file in the repository for more details.