Optimising Daily Revenue

Underlying trends and fare amount prediction

Dong Li
Github Repository

August 25, 2024

1 Introduction

There are only so many hours a person can work, so how can taxi drivers optimise their time to maximise their revenue. On average, a taxi driver works for 9.5 hours a day, 6 days a week [1], this gruesome schedule demands efficiency and a strategic approach to find customers to generate and maximise daily revenue.

Understanding which taxi zones to target and which conditions leverage themselves to higher revenue is important to any driver looking to make more money. Fare amount was identified as the main predictor of revenue as it makes up the majority of a taxi driver's income and is the most accurate representation of the general income of a taxi driver. In this paper, we will look at the time period from the 1st of October, 2022, to the 31st of March, 2023, look at the geospatial representation of the data and explore and evaluate two machine learning models on their ability to predict average daily fare amount.

The goal of this paper is to provide taxi drivers with insights into features that can guide their decision-making and help them understand underlying relations that maximize their earnings. By leveraging these findings, we hope that drivers can enhance their operational efficiency and improve their overall revenue.

2 General Preprocessing

In this section, we will look at the general preprocessing methods used to data wrangle the TLC taxi [2], NOAA weather [3] and NYC Open traffic data [4] into daily representations. Feature selection methods and further transformations for the modelling of the data will be further explored in section 4, Modelling.

2.1 Preprocessing TLC data

The initial total amount of records from the two datasets was 20121302 for the 6 months from 2022-10-01 to 2023-03-31 It was found that there were many semantic errors as well as inconsistencies in the entry and recording of the data. The steps used to merge the two taxi datasets, address these issues and then process the data into a daily representation are described below.

2.1.1 Merging of yellow and green taxi data

Due to discrepancies in the features of the yellow and green taxi datasets, irrelevant features such as trip type and, store and forward flag, were removed from both datasets. Poorly defined features, namely the E-hail fee, were also excluded. Following this, the pick-up and drop-off times, which had identical descriptions but different feature names, were generalised. These steps ensured both datasets had the same features allowing for the merging of the two datasets.

2.1.2 Removal of semantic errors and outliers

The data had many semantic errors and inconsistencies which resulted in many incorrect records. Due to the sheer volume of the data, instances with errors were filtered out as outlined below.

- **Records outside the target date range** were removed as we were focused on the time period, 2022-10-01 to 2023-03-31.
- **Negative or zero values** in passenger count, trip distance were removed as these can only take values ≥ 0.
- **Negative values** in the taxi fees, mta tax, tip amount, tolls amount, improvement surcharge, total amount, congestion surcharge and airport fee were removed as these values are ≥ 0.
- **Values exceeding a set maximum** of \$1.25 in airport fees were removed.
- **Values outside set range**, seen in Rate code Id and Payment type were removed as these values could only take integer values in the range 1 to 6.
- **Fare amounts below the minimum fare amount**, the starting fare amount is \$2.50 so values less than this were removed.
- **Severe mismatch between recorded total amount and calculated total of summed fees** were found and removed, indicating inconsistency and error in the record. The threshold for the mismatch between the summed total and the recorded total was set to \$3 as many records did not sum the airport fee which is \$1.25 as well as smaller errors in the summation of fees.
- **Negative or very short trip time** were removed, this threshold was set to 0.02 hours which is 1.2 minutes.

Following this, outlier detection and removal was conducted. As there is a substantial amount of data and the data records are, in practice, a random representation of the true records, according to the central limit theorem the overall distribution of the data will be approximately normal [5]. Using this assumption we calculated the mean and the standard deviation for each of the numerical features and removed any values that lay beyond $\pm\sqrt{2\log(N)}$ standard deviations, N is the number of instances in the data. This resulted in the removal of a further 180258 records from the dataset leaving a total of 17932561 records (about 89% of the original data).

2.2 Preprocessing Weather data

An initial look at the weather data showed there were 15825 instances over 2022 and 2023, and that it was generally well-defined and consistent. However, there were random maximum values that were inserted and evidence of incorrectly inputed data. The steps used to process this were similar to the techniques used for the taxi data sets. The weather data was generally well-defined and thus the outliers were a super-set of the semantic errors therefore removing outliers also acted as a means to remove semantic errors. We proceeded with preprocessing and outlier removal as such.

- **Filtering out primary numerical attributes** was done as there were 8 primary numerical features in the dataset with each attribute having several columns describing specific information about the main primary feature. These columns were often empty or beyond the scope of the research question and thus were dropped.
- **Records outside the target date range** were removed to leave only records in the target date range of 2022-10-01 to 2023-03-31.
- **Removal of outliers** was implemented by removing data that lay beyond $\pm\sqrt{2\log(N)}$ standard deviations for each of the numeric columns respectively.

After this filtering and outlier removal, there were 3858 instances (about 25% of the original data) left in the dataset, this showed that not many outliers were found and that most of it was filtering to the date range.

2.3 Preprocessing Traffic Data

There was a lot of traffic data, a single csv with 1673725 instances, however, a lot of data was missing at the daily scale, thus this dataset was used to see general visualisation of the traffic in the 5 boroughs of New York City. The steps taken to preprocess the data were as such.

- **Records outside the target date range** were removed to leave only records in the target date range of 2022-10-01 to 2023-03-31.
- **Removal of outliers** was implemented by removing data that lay beyond $\pm\sqrt{2\log(N)}$ standard deviations for each of the numeric columns respectively.

The final shape indicated sparsity of data as there were only 32305 (about 2% of the original data) with most records having a lot of records for one day and then no records for a few days.

2.4 Aggregation of data into a daily representation

For the TLC data, the data was grouped by pick-up location ID and date and then the averages for all numerical values related to revenue were calculated. As previously mentioned a summation of the fees was calculated and compared to the recorded total amount, as there was often a discrepancy and the average between these two values was taken and recorded as the average total amount.

The weather data was grouped by date and the averages for temperature, dew point temperature, station level pressure, sea level pressure, wind speed, precipitation, relative humidity and wet bulb temperature were all averaged.

Unfortunately, the traffic data per borough was too sparse and did not allow for reasonable imputation of values to create a useful day-to-day representation thus the data was grouped by borough and the sum of the total volume of the recorded days was aggregated together.

3 Analysis

This section aims to analyse the datasets, illustrate the distribution and spread of the data from the datasets and gain a general overview of how the datasets correlate to each other.

3.1 Geospatial analysis

The focus of this paper is on the average daily fare amount, we first need to get an understanding of the data. Looking at Figure 1 (left) we see the average distribution of fare amounts across taxi zones over the entire time period. From this initial look at the data we see several taxi zones, particularly in Queens, exhibit higher average fare amounts per hour compared to other boroughs. In contrast, when looking at Figure 1 (right), we can see that there appears to be an inverse relationship between traffic volume and average fare amount per hour.

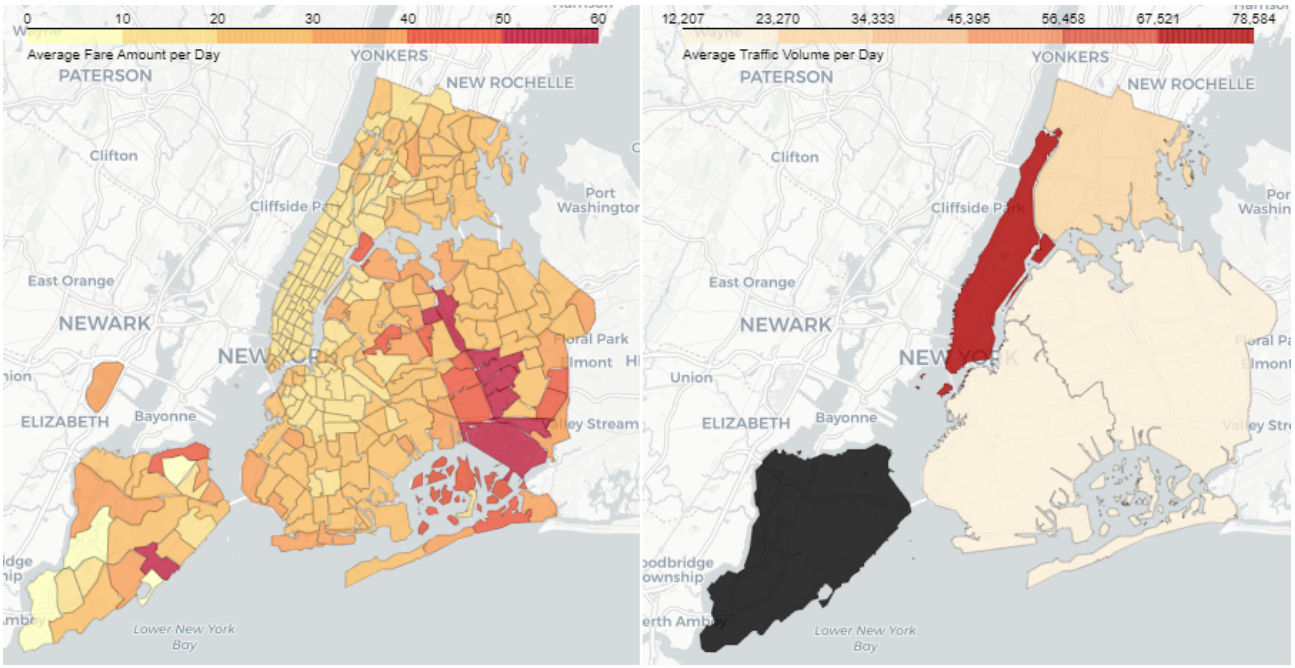

Figure 1: Average fare amount (USD) per day by taxi zone (left) vs average traffic volume per day by New York City borough (right).

The traffic data seems very promising and correlated to average fare amount but as the traffic data was very sparse it was decided that it was best used to show a general representation of how traffic operates in New York City, moving forward we will primarily look at the taxi dataset and weather data set in relation to modelling and recommendations.

3.2 Trends of Daily Average Fare Amount Over Time

Following this, we constructed a representation of how the average daily fare amount in New York City fluctuated over the time period 2022-10-01 to 2023-03-31. In Figure 2, it is evident that there is a sharp increase in fare revenue during the period of Christmas and New Year's. However, what was more unexpected was the continuance of this high fare amount into February and March. Temperature was chosen as a representation of the weather dataset as temperature acted as a broad representation of the trends in weather data.

Looking at Figure 2, we see that temperature has a slight negative trend, with a sharp drop occurring around Christmas and New Year's, coinciding with the sharp increase in fare amount over this period. Overall, we can see that there is a slight inverse correlation between average daily fare amount and average temperature.

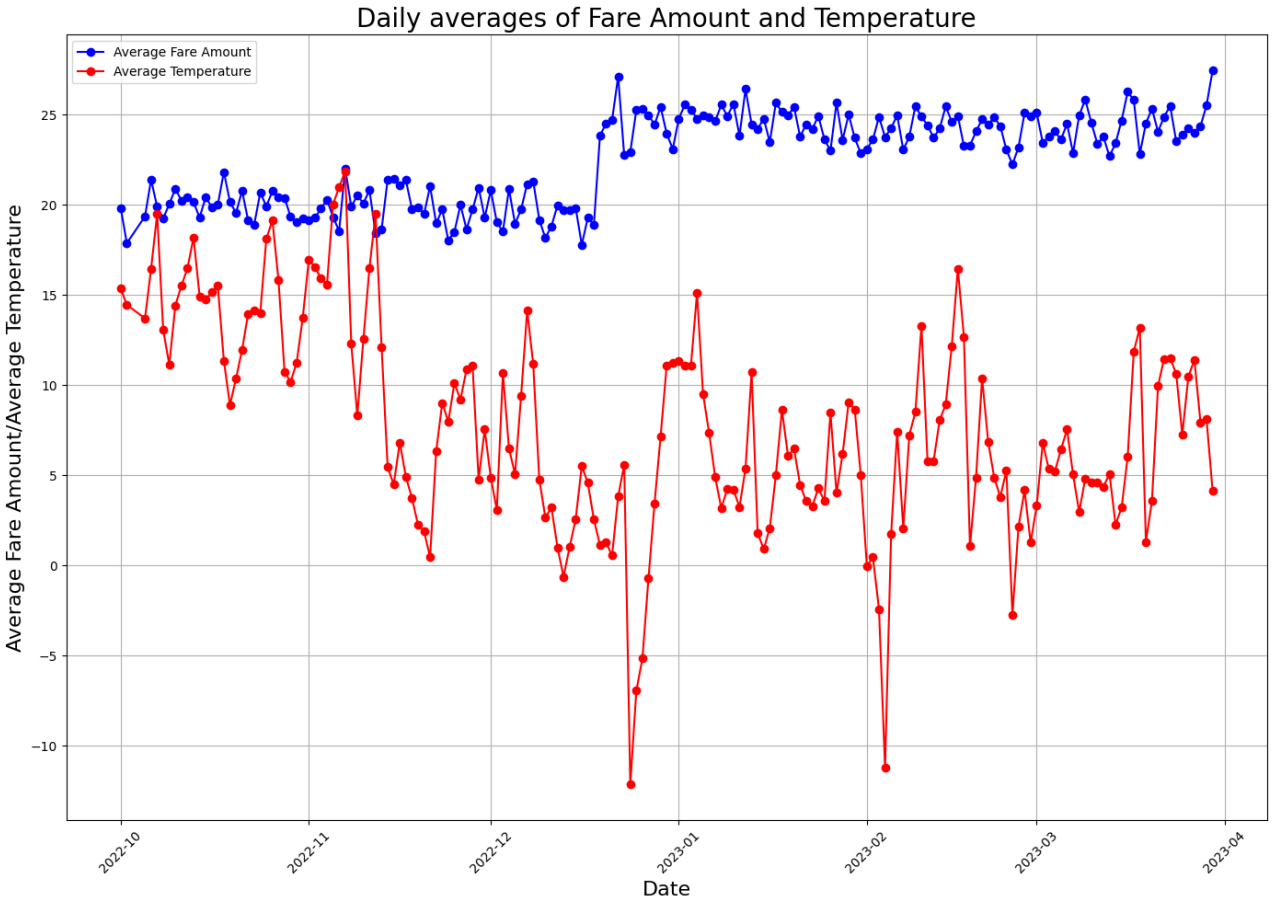

Figure 2: The trends of daily average temperature (°C) and fare amount (USD) over the time period 2022-10-01 to 2023-03-31

4 Modelling

To model the average daily fare amount, two different models were implemented, Linear Regression (LR) and a Neural Network (NN). These models were trained on the preprocessed taxi and weather datasets, and were evaluated on their performance of predicting daily fare amount on a future set of taxi and weather datasets from the period 2023-04-01 to 2023-06-30.

Linear Regression was chosen for its capability to identify linear relationships within the data, particularly after performing some transformations. However, with the limitations of linear models in capturing complex relationships, a neural network was also implemented. The neural network provides a more sophisticated framework for modelling intricate patterns and complex underlying relations in the data.

4.1 Linear Regression

Here the aim was to do feature selection and fit a linear regression model to our data, the steps in how we implemented this are described below.

4.1.1 Further Processing

Given the nature of fare amount and the goal of finding the underlying relationships that predict daily average fare amount, extra processing steps were employed to avoid data leaks and refine the data for machine learning.

Under the assumptions of a linear model, the errors of the model must be distributed multivariate normal [6]. From Figure 3, we can see that many of the numeric features have a positive tail, so to better normalise it we performed log transforms on average airport fee, average amount per hour, average extra, average fare amount, average passenger count, average tip amount, average tolls amount, average total amount, average trip distance, average trip time, daily zone revenue, mean precipitation and number of pickups. These transformations resulted in a better distribution of the data and thereby the errors hence satisfying the assumptions of a linear model.

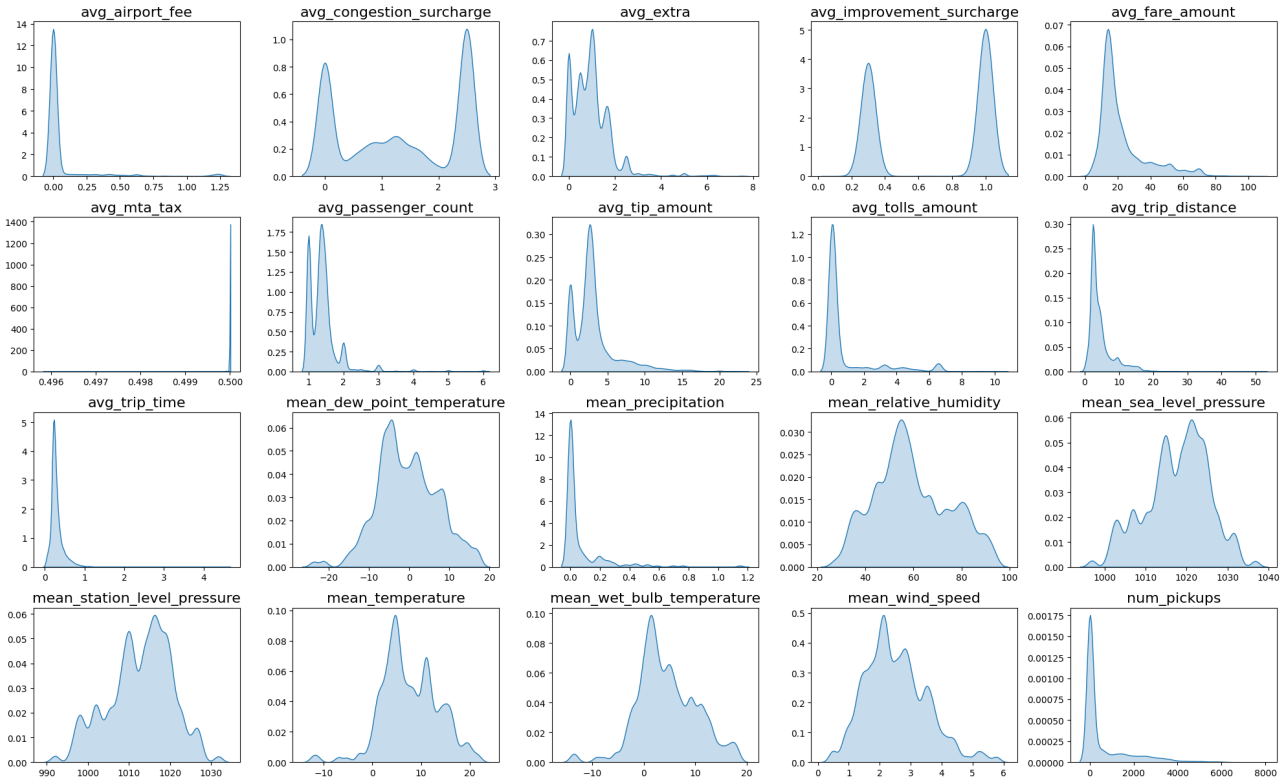

Figure 3: The density plots of the numeric features

4.1.2 Feature Selection

Fare amount was defined as "The time-and-distance fare calculated by the meter" [2], thus to predict this value without data leaks, the average trip time and average trip distance columns were removed as fare amount is directly derived from these two features. Other columns were also dropped, average total amount, daily zone revenue, and average amount per hour. These were also either directly derived from fare amount or were nearly perfect proxies for the feature, namely the average total amount.

In the training dataset, there were 278 features with one-hot encoded pick-up locations making up 259 of these features. From Figure 1, we can see that pick-up location is relevant but it is unclear which locations are relevant enough to generalise the model. To do feature selection, recursive feature elimination (RFE) was used. RFE recursively considers smaller and smaller sets of data removing

features based on the importance of each feature [7]. Feature importance is defined differently depending on the model. For linear models, importance is typically measured by the absolute value of the coefficients associated with each feature. In contrast, for neural networks, the concept of feature importance is less straightforward and is often evaluated by analyzing how the removal or perturbation of a feature influences the model's performance [8]. RFE was used in preference to RFE with cross validation as that was too computationally expensive and time-consuming to run to find the optimal number of features. A brute force search between initial guesses was ran to find the number features that produced the highest R^2 value and this was taken to be the best number of features to select.

4.1.3 Linear Regression Model

From RFE, it was found that the number of features that constructed the most optimal was 142. The general form of the linear model was

$$\log(\mathbf{y}) = X\beta + \epsilon$$

Where X contains the log-transformed variables and the one hot encoded pick-up location variables that are described above.

4.2 Neural Networks

Unlike Linear Regression which has certain assumptions about the data, neural networks do not need this and as such were trained without transformations to the data. A feed-forward neural network was implemented with an input layer of 113 features that were derived using RFE and were chosen based on the maximum R^2 value. Following the input layer two hidden layers with 64 and 32 units, respectively, were chosen, both using Rectified Linear Unit (ReLU) activation to introduce non-linearity and capture underlying complex patterns. The output layer is comprised of a single unit for predicting the average fare amount. The optimiser that was chosen was "Adam" as it is good at adaptive learning and the loss function that was chosen was MSE [9].

5 Results and Discussion

To evaluate both of the models Mean Squared Error was used, in terms of predicting an average value like fare amount it is highly interpretable.

Linear Regression performed well with a true MSE of 1.050054 and an R^2 0.822124. It was expected that linear regression would perform well but these results were still surprisingly good. RFE determined that much of the other fee data as well as pick-up location was very important to constructing an effective linear model. Using the support function of RFE we were able to derive the most important features, shown on Figure 4, is the graphical representation of how important pick-up locations were in the feature selection phase of linear regression. Much of the graph is red indicating that pick-up location was a significant variable in determining the average daily fare amount. Furthermore, looking at Figure 5 we can see that date, day, other fee features as well as pickup location and weather features had significant effects on the model.

Neural networks obtained an MSE of 0.043963 and an R^2 of 0.839894 which was better than linear regression. This was expected as neural networks are able to model non-linear relationships. For the training data for the neural network, none of the variables were transformed and the optimal amount of features was 129. From these results, we see that the Neural Network model was able to capture

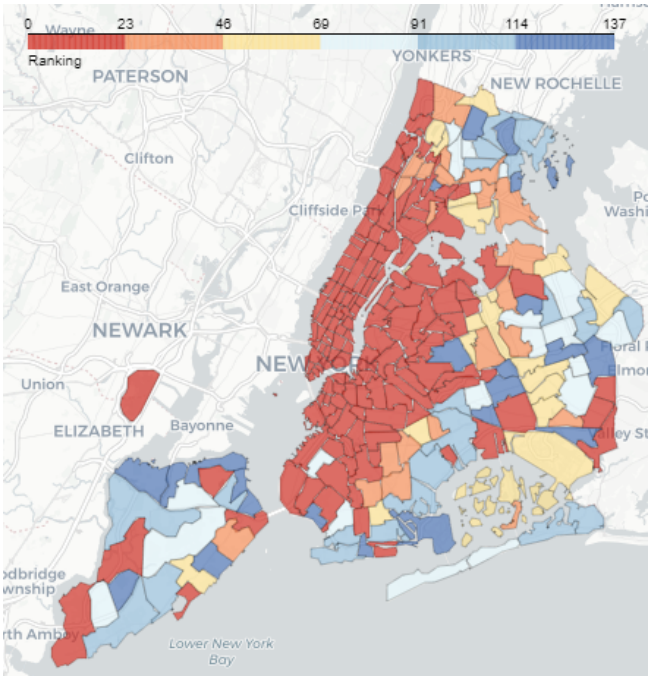

Figure 4: Ranking of the zones relevance in feature selection

Feature	Coefficient
date	0.4871
day_of_week	-0.4838
avg_congestion_surcharge	0.1940
avg_passenger_count	0.1312
avg_mta_tax	0.1285
avg_extra	0.1100
pu_location_id_163	-0.0812
pu_location_id_114	-0.0788
pu_location_id_162	-0.0785
pu_location_id_180	-0.0773
pu_location_id_165	-0.0769
pu_location_id_116	-0.0766
pu_location_id_157	-0.0760
pu_location_id_115	-0.0758
mean_station_level_pressure	-0.0747

Figure 5: Sorted top 15 features by coefficient magnitude

more of the underlying patterns connecting features to fare amount. However, due to the nature of neural networks, it is much harder to interpret and draw conclusions about which features specifically caused this outcome.

6 Recommendations and Conclusion

From the results obtained from our models, we can see that both neural networks and linear regression were effective in predicting the daily average fare amount with both models having low MSE and high R^2 values. This in conjunction with the geospatial analysis leads me to suggest taxi drivers to target high revenue zones, adjust driving patterns according to day and date and analyse the weather conditions.

The results of the models indicate that taxi zones are significant to fare amount, from the geospatial analysis we see that there are zones with consistently higher average fare amounts. Near the centre of Queens is the main area of interest with it having a high average fare amount, making it a prime area for drivers seeking higher returns. Furthermore, the daily fare trends and the importance of the day of the week and date indicate that drivers should adjust their driving patterns to capitalize on these peak periods by working longer hours or focusing on areas with historically high fares during these times. Finally, it is seen that there is an inverse correlation between average fare amount and weather conditions. Drivers should consider adjusting their schedules to work more during colder periods when fares are generally higher.

Ultimately, by strategically targeting high-revenue zones, adjusting driving patterns to peak times, and considering weather conditions, drivers can effectively optimise their routes and significantly increase their daily earnings.

References

[1] I. Mujawar et al. "Sleep behavior of New York City taxi drivers compared to the general US population". In: *Journal of Transport & Health* 22 (2021), p. 101237. DOI: 10.1016/j.jth.2021.101237. URL: https://doi.org/10.1016/j.jth.2021.101237.

[2] New York City Taxi and Limousine Commission. *TLC Trip Record Data*. https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page. Accessed: 2024-08-01.

[3] Matthew J. Menne et al. *Global Historical Climatology Network-Hourly (GHCNh)*. https://www.ncei.noaa.gov/metadata/geoportal/rest/metadata/item/gov.noaa.ncdc:C01688/html. 2023. Accessed: 2024-08-02.

[4] NYC Open Data. *Automated Traffic Volume Counts*. https://data.cityofnewyork.us/Transportation/Automated-Traffic-Volume-Counts/7ym2-wayt/about_data. Accessed: 2024-08-02.

[5] William J. Welch. *Statistical Inference: A Primer on Likelihood and Bayesian Methods. Stat 305 Introduction to Statistical Inference*. Course notes. Department of Statistics, University of British Columbia.

[6] Yao-Ban Chan. *Linear Statistical Models - MAST30025*. Lecture notes. Department of Mathematics and Statistics, University of Melbourne.

[7] Loyford Mwenda. *Recursive Feature (RFE) Elimination with Scikit-learn*. Medium. Accessed: 2024-08-17. URL: https://medium.com/@loyfordmwenda/recursive-feature-rfe-elimination-with-scikit-learn-d0d29e96273d%5C#:~:text=RFE%5C%20searches%5C%20through%5C%20the%5C%20training.

[8] Scikit-Learn. *sklearn.feature_selection.RFE — scikit-learn 0.23.1 documentation*. https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html. Accessed: 2024-08-17.

[9] TensorFlow. *tf.keras.optimizers.Adam — TensorFlow Core r2.0*. https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam. Accessed: 2024-08-17. 2019.