# Executive Summary

This report is set out to find the best model for accurately predicting taxi price at the time of booking, with the aim of helping to enable A2B’s prepaid fare service. After conducting comparisons across different high performing machine learning models, the optimal model is found to be the Gradient Boosting model with 15 features, for its high accuracy and robustness. The optimal model has a validation RMSE of 3.265. 84.9% of the predicted values have 10% deviation from the true target values, and 93.6% of the predicted values are within ± 20% of the true target values. To further improve the model’s prediction accuracy, it would be recommendable to include the type of taxi as a regressor which affects both the fixed flagfall and the charged rate. It is also suggested to use as regressors the forecasted trip distance and time data from Google map, instead of using the recorded distance and time information which were not known at the time of booking.

# Overview

The first part of the report is dedicated for data pre-processing and feature engineering, with the main purposes of denoising and selecting/transforming the features to be used in the model building process. In the second part of the report, different machine learning models were fit into the training and validation dataset with the aim of selecting the best one in terms of the prediction accuracy and model interpretability. Simple benchmark models were trialed followed by the more complex ones. Section 4 compares the performance of the models, using the measures of model accuracy and error. A brief discussion follows the model results to suggest practical limitations and some potential ways of further improving the model. Section 5 provides further insights into the features used in the optimal gradient boosting model for better model interpretation. Section 6 concludes the report and suggests scope for improvement.

# Prediction Model Methodology

## Prediction Measures

This report presents predictor performance using three measure: Root-Mean-Square Error (RMSE), accuracy within 20% and accuracy within 10%. The RMSE is calculated as,

where is the number of samples being tested, and and are the true and predicted values of the -th sample, respectively. Similarly, the Mean Absolute Error (MAE) is calculated,

The accuracy within 10% (and 20%) is the proportion of samples correctly predicted to within 10% (and 20%) of the true value. Except where otherwise indicated, reported measures are calculated on the validation sample.

## Pre-processing and Feature Engineering

Before the data can be used in prediction models, certain pre-processing steps were performed. They can be categorised into general, time-related, and location related processes.

### General pre-processing

Outliers were removed from the dataset by excluding the most extreme 1% value of the distance, time, and the charge price data. The set was randomly split into training and validation sets where 70% of the data was placed into the training set.

### Time

The day of the week on which trips occurred was extracted and transformed into six dummy variables. The time of day was also extracted and scaled such that the range is between 0 and 1 rather than 0 and 24. One public holiday exists in the dataset: Labour Day on 7th October 2019. Whether or not a trip occurred on this public holiday and between the hours of 6am and 10 pm (as define by the taxi fare legislation, NSW Government, 2018) is defined and set to be a binary variable.

### Location

The city where trips took place is recorded as either Sydney or Tamworth, however many trips recorded in Tamworth took place in Coff’s Harbour. Therefore, this was corrected using the criterion that if a trip labelled as occurring in Tamworth has a longitude of the start of the trip greater than 152°, then the trip is set to be in Coff’s Harbour. This split point was chosen from above and is clearly identified in Figure 1. Since the city is a categorical variable, it is transformed into two dummy binary variables.

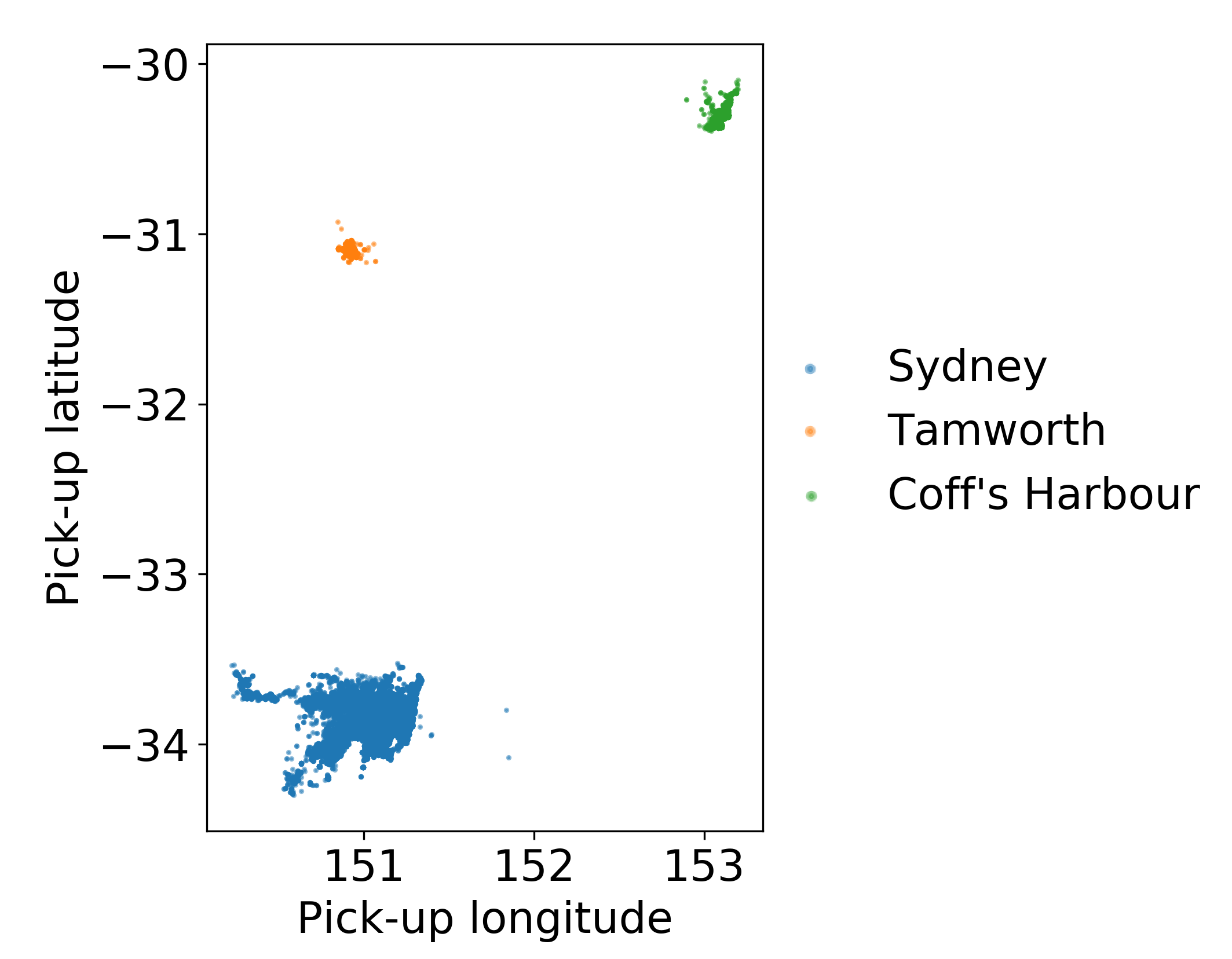


Figure 1. Starting locations of trips.

The geodesic distance of the starting points from their respective city centres (defined for Sydney as -33.87097, 151.207285, see Figure 2, and as the centroids of Tamworth and Coff’s Harbour trips respectively) were calculated following (“Calculate distance and bearing between two Latitude/Longitude points using haversine formula in JavaScript”, n.d.). Additionally, the bearing of start point of trips from the city centres were also calculated. Similarly, the journey directions (the bearing of the end point from the starting point) were calculated.

A close up of a map

Description automatically generated

Figure 2. Sydney's city centre as used for distance and bearing calculations.

Directions were scaled from a range of 0 to 360 to a range of 0 and 1 for simplicity of calculations. Distances were standardised such that they have a standard deviation of 1, but the distribution is unchanged, and all values remain positive.

## Benchmark Model

Linear models are desirable for a number of reasons. They are simple, computationally cheap to apply, easy to interpret, and tend to generate relatively stable predictions (James, Witten, Hastie, & Tibshirani, 2017). However, they tend to be prone to under-fit (reduction in prediction performance caused by oversimplification) and are sensitive to the underlying assumptions (James *et al,* 2017). Nonparametric models (which include K-Nearest Neighbours, Decision Trees, Random Forest, Extreme Gradient Boosting, and Neural Networks) are more challenging to interpret, prone to over-fit (a reduction in prediction performance due to unnecessary complexity and fitting to noise), have the potential to generate less stable predictions, and are more computationally expensive to apply (James *et al,* 2017). However, they are very flexible and don’t rely on any assumptions (James *et al,* 2017). As a result, the process of specifying nonparametric models tends to be automatable and data-driven (James *et al,* 2017).

The choice between the two types of models is often a trade-off between the above characteristics and the observable prediction performance. Some clients will prefer models with high interpretability over accuracy. Other clients view performance accuracy as paramount, as is the case for A2B and their stakeholders.

### Relevant pre-processing

A total of fifteen features (including: Trip distance, trip time, time of day, geodesic distance of the pick-up location from the city centre, bearing of the pick-up location from city centre, journey direction, city as dummy variables, day of the week as dummy variables, and public holiday as a binary variable) were used in all benchmark models. Whilst the trip distance and time are not known at booking, they can be estimated with Google Maps or similar application programming interface (API). Log transformation was performed on the trip distance, trip time and fare (our target variable). Exploratory data analysis demonstrated that those variables were highly skewed (see Figure 3, Figure 4, and Figure 5), with outliers in the upper tails of their respective distributions. Log transformation of the affected variables appears to have addressed this issue. Standardisation was also performed in an effort to address the differences in the scale of the variables and improve the computational time.

A number of models were constructed for consideration as a benchmark. These can be grouped into linear and K-Nearest Neighbour models. In general, these types of models are highly interpretable.

|  |  |
| --- | --- |
|  |  |
| 1. Fare | 1. Log fare |

Figure 3. Histograms of fare and log fare.

|  |  |
| --- | --- |
|  |  |
| 1. Trip distance | 1. Log trip distance |

Figure 4. Histograms of trip distance and log trip distance.

|  |  |
| --- | --- |
|  |  |
| 1. Trip time | 1. Trip distance |

Figure 5. Histograms of trip time and log trip time.

### Linear models

A number of linear models were explored. Linear-linear, linear-log, log-linear and log-log (referring to the transformation of the Fare-Trip distance / Trip time combinations) models were trained. LASSO models were attempted first. LASSO is capable of performing dimension-reduction - which generates simpler models (when compared to Ordinary Least Squares (OLS)) during training, reducing the risk of over-fit at an earlier stage in the modelling process. OLS models were also attempted and assessed. Each linear model’s results are presented in Table 1. The LASSO Linear-Linear Model has the best performance of the linear models, with the highest accuracy - the difference between the validation RMSE with that of the OLS Linear-Linear model is insubstantial.

Table 1. Linear model results. Training and Validation RMSE. (smaller is better)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **Feature-Target Scales** | **In-sample RMSE** | **Validation RMSE** | **10% Accuracy** | **20%**  **Accuracy** |
| **LASSO** | **Linear-linear** | **4.014** | **3.977** | **0.726** | **0.930** |
| Linear-log | 8.709 | 8.648 | 0.187 | 0.373 |
| Log-Linear | 17.957 | 17.823 | 0.243 | 0.573 |
| Log-log | 19.967 | 19.866 | 0.093 | 0.188 |
| OLS | Linear-linear | 3.974 | 3.937 | 0.707 | 0.912 |
| Linear-log | 8.462 | 8.431 | 0.187 | 0.384 |
| Log-Linear | 17.008 | 16.881 | 0.265 | 0.576 |
| Log-Log | 4.322 | 4.277 | 0.704 | 0.916 |

Figure 6 demonstrates that the least-squares assumptions are clearly invalid, as the errors are not Gaussian - there is a large excess kurtosis (and we have already mentioned the significant skewness of the continuous variables). As such, the linear model is unreliable, and we must examine non-parametric models (the reliability of which does not depend on such assumptions).

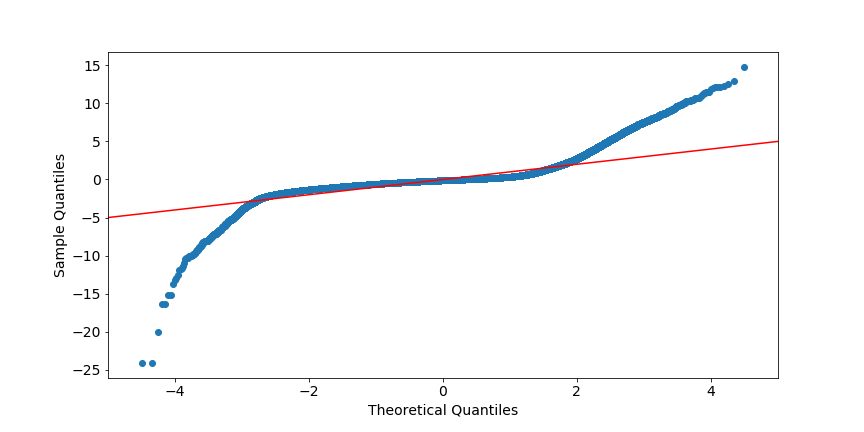


Figure 6. Quantile-quantile plot of standardised residuals.

### K-Nearest Neighbours (KNN) models

A range of KNN models were specified, each with a different number of neighbours (k) model setting () and assessing prediction performance on validation data. According to Figure 7, the optimum value for k was 29 - the resulting KNN model has a training RMSE of 3.744 and a validation RMSE of 3.850. The test accuracy within 10% was 0.758, and within 20% was 0.930. The Mean absolute error was 2.009, and the mean absoulte percentage error was 8.411%. Model fitting computational times were each less than 10 minutes, and prediction computational times were, similarly, less than 30 seconds (using 289,924 data entries).

### Chosen benchmark model

The benchmark model chose moving forward was the KNN model with . Note that its performance was comparable to that of the best linear model but unlike the latter, its reliability does not depend on the least-squares assumptions.

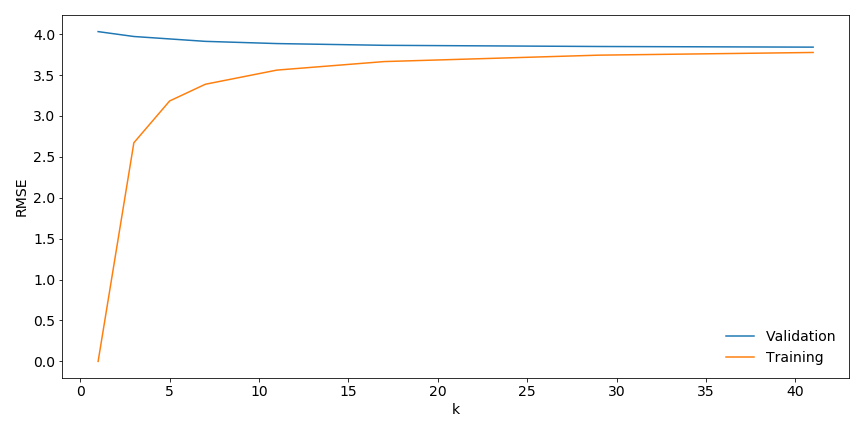


Figure 7. KNN (linear) model tuning using training and validation.

## Random Forest

Random forest consists of many decision trees and take the mode of the output class which consists of all outputs from all individual trees. Unlike bagging of trees where trees may be intercorrelated due to similar choice of features, random forest model manages to lower the correlations between trees by splitting up features and allocating one subset of the features to each tree.

The choices of features for the random forest model are simple: city (as dummy variables), distance, time, holiday (as dummy variable) and daylight (as dummy variable, which indicates whether the trip took place between 6am and 10pm). Figure 8 shows the importance of different features in the model.

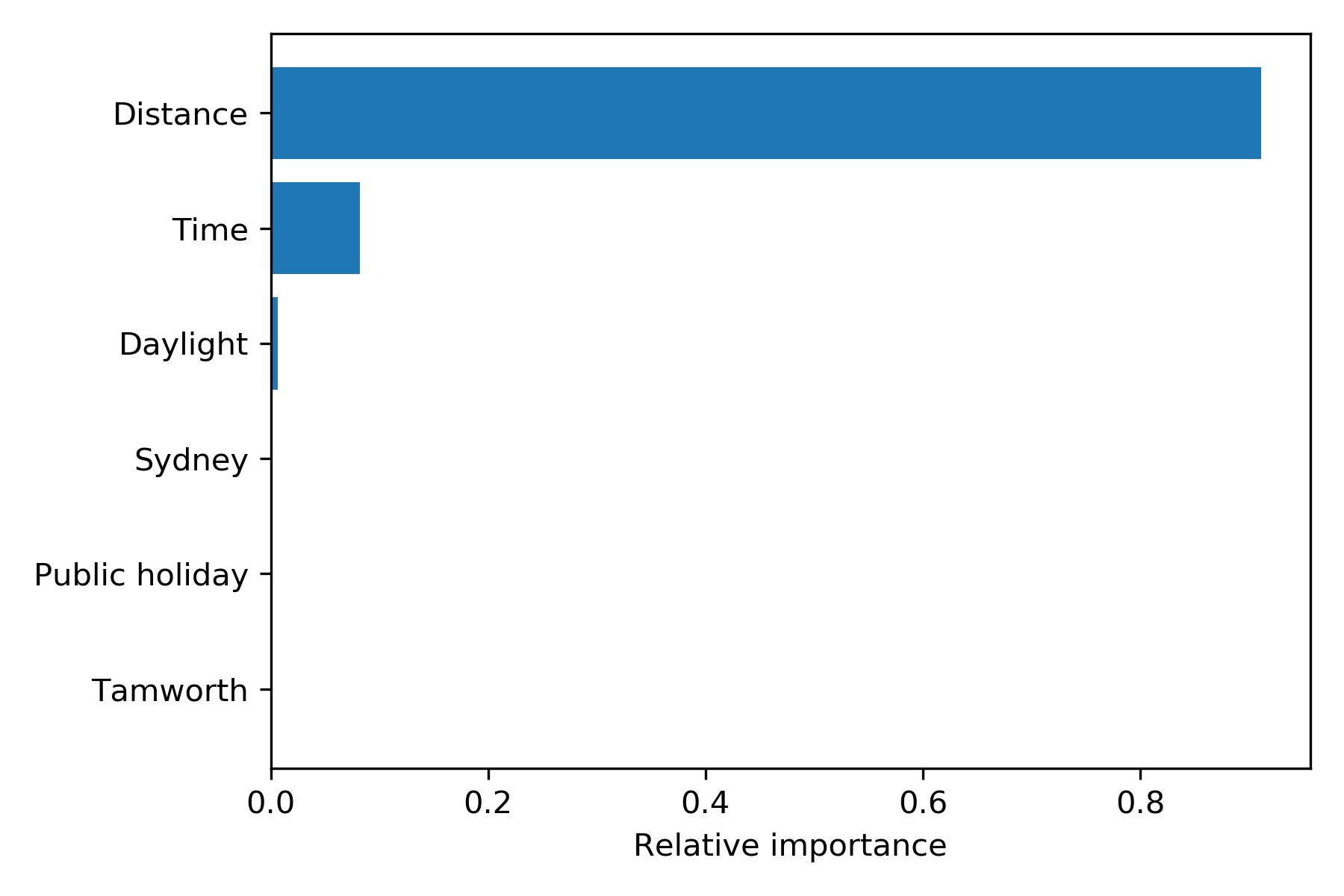


Figure 8. Feature importance for the Random Forest model.

## Decision Tree

A decision tree is a highly interpretable and effective predictor whereby the predicted value is the outcome of passing the predictors through a hierarchical decision-making process. Figure 9 shows a decision tree with three layers of decisions. Because of the shallow tree size, only two predictors are implemented in the tree: distance and time. In this model, time is only a factor for shorter journeys.

A close up of a piece of paper

Description automatically generated

Figure 9. Decision tree regressor with a depth of 3.

A more advanced and accurate decision tree can also be used. To determine the optimal tree depth, a decision tree was trained and validated with various maximum depths and the following features: city (as dummy variables), day of week (as dummy variables), distance, time, time of day, public holiday, starting geodesic distance from city centre, starting direction from city centre, and journey direction. Figure 10 shows the tuning results: a decision tree with a depth of 13 has the lowest RMSE. The increasing RMSE for larger trees is likely due to overfitting on the training set.

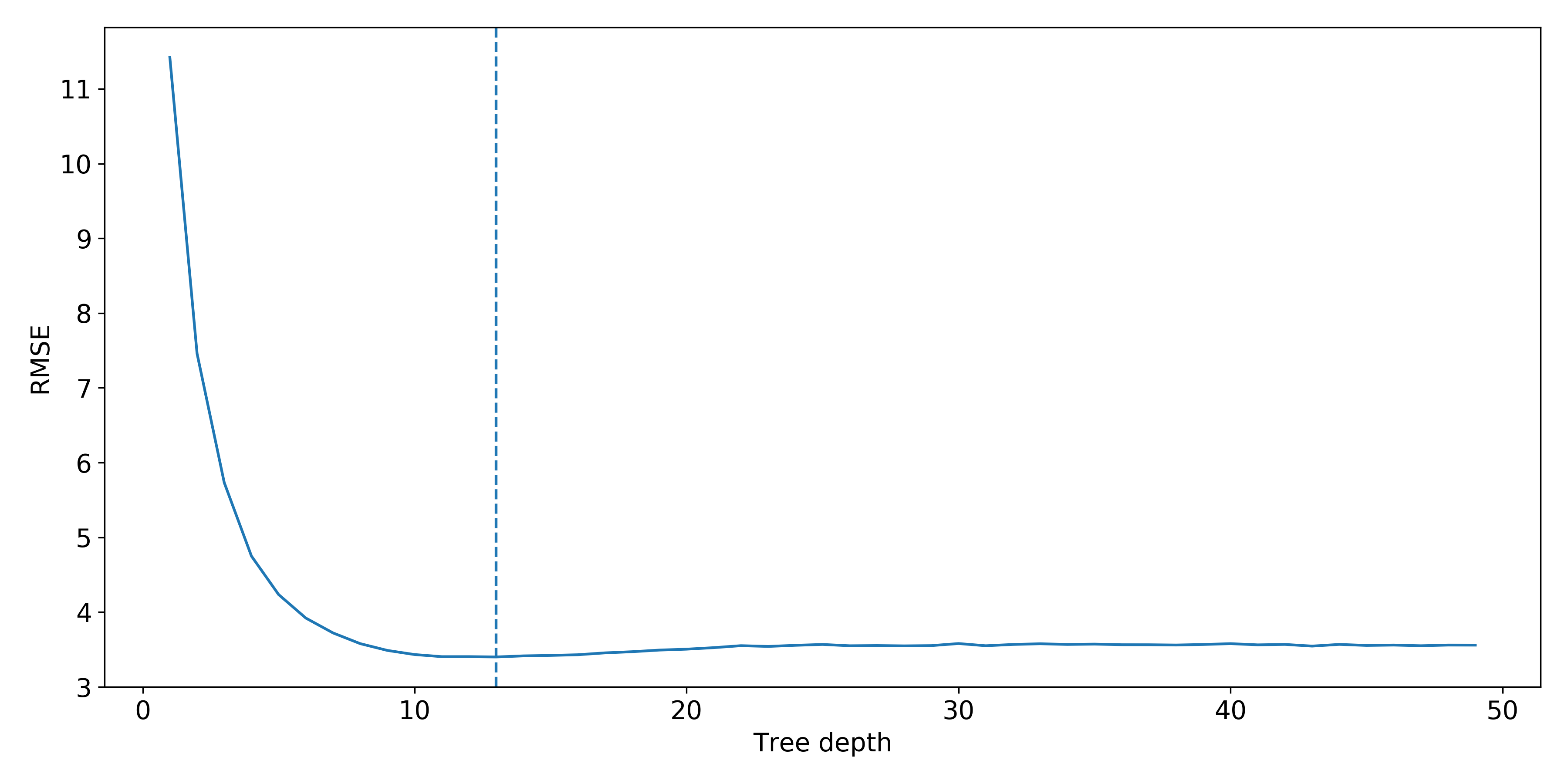


Figure 10. RMSE of decision tree regressors of varying depth.

Using the decision tree with a maximum depth of 13 yields the features’ importance as depicted in Figure 11. Note that all feature importance values are non-zero and reducing the number of features does not improve validation results.

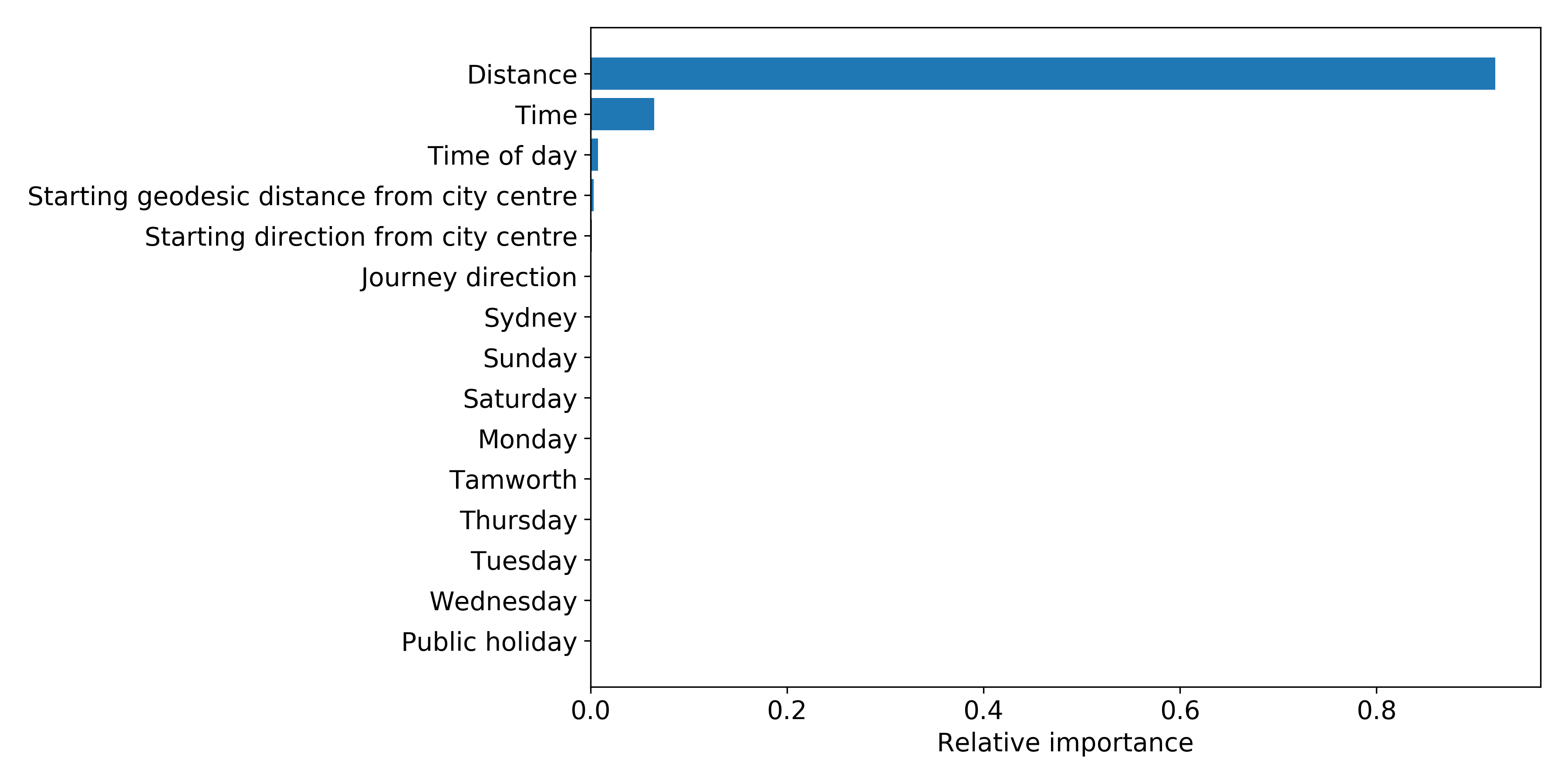


Figure 11. Feature importance for decision tree regressor of depth 13.

## Gradient Boosting

Gradient boosting uses multiple decision trees to regress. Multiple simple trees are built on the residuals of the trees before, with optimisations occurring at each step to minimise the error. Then, the ensemble of simple trees makes up the final model. It has been implemented using XGBoost, an algorithm which performs additional optimisation and regularisation to improve performance.

The parameters were tuned on a randomly selected sample 10 000 points from the training set. This resulted in much faster tuning and no or little difference to the chosen parameters. Table 2 shows the optimised parameters after using cross validation on the model trained on every combination of a set of each of these parameters. The maximum depth is the maximum allowable depth of the individual decision trees. The number of estimators is how any trees are trained in the model. The subsample size is the ratio of data to use in training each boosting iteration to avoid over-fitting. Finally, the learning rate is how much the step size reduces in each iteration, which also prevents over-fitting.

Table 2. Tuned parameters of the gradient boosting model.

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Subsample size | 0.8 |
| Number of estimators | 1200 |
| Maximum depth | 4 |
| Learning rate | 0.8 |

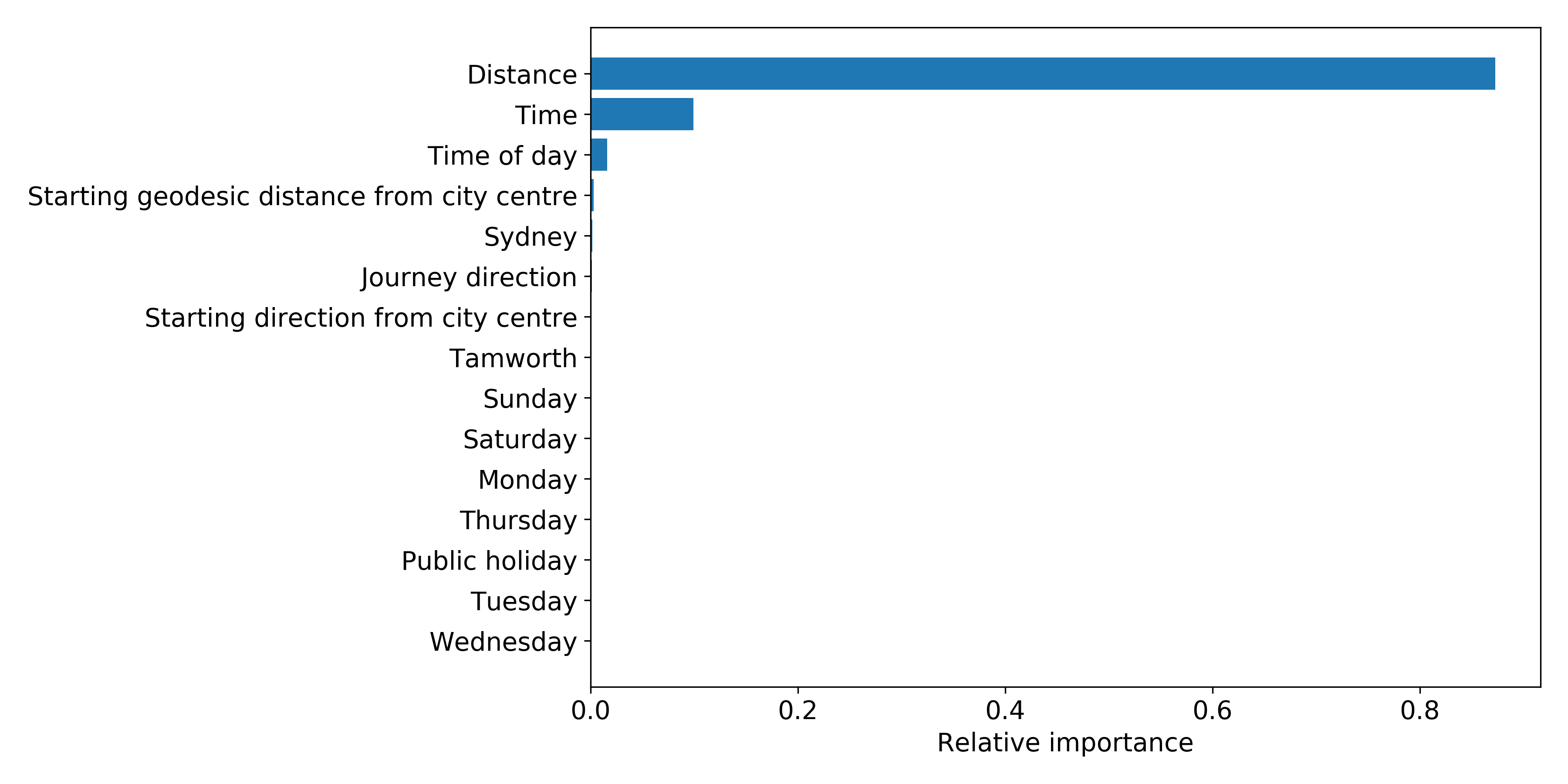
Similar to Figure 11, Figure 12 shows the feature importance values for the trained model. Again, all importance values are non-zero.

Figure 12. Feature importance for gradient boosting model.

## Neural Network

Finally, a neural network was trained on the data with the same 15 features as the decision tree regressor and gradient boosting model. The network has 3 hidden layers of 128, 64 and 32 nodes respectively, and a linear activation layer. This architecture is summarised in Figure 13.

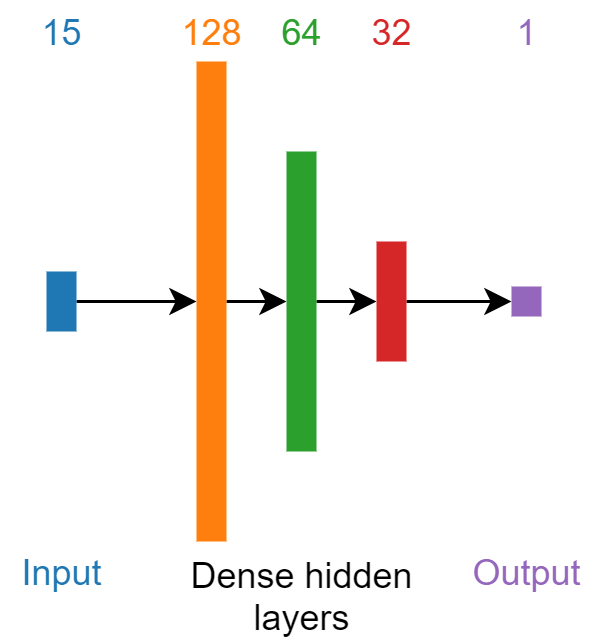


Figure 13. Neural network architecture.

The neural network was trained over 20 epochs in batches of 256, and the loss function is the Mean Squared Error (MSE), which will consequently minimise the RMSE. A learning curve of the model being trained on a subset of the training set and validated on the remainder of the training set is depicted in Figure 14. Evidently, the model quickly converges. Unfortunately, the neural network has very low interpretability.

A close up of a map

Description automatically generated

Figure 14. Learning curve of the neural network.

# Results

Table 3 presents a summary of the results obtained by the prediction models and benchmark discussed in this report.

Table 3. Results of prediction models.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Features** | **In-sample RMSE** | **Validation RMSE** | **MAE** | **Accuracy within 20%** | **Accuracy within 10%** |
| Benchmark: KNN | 15 | 3.744 | 3.850 | 2.099 | 0.930 | 0.758 |
| Decision Tree (depth - 3) | 3 | 5.738 | 5.733 | 3.969 | 0.607 | 0.347 |
| Decision Tree (depth - 13) | 15 | 2.937 | 3.400 | 1.472 | 0.934 | 0.847 |
| Random Forest | 6 | 1.605 | 3.204 | 1.437 | 0.929 | 0.828 |
| **Gradient Boosting** | **15** | **3.236** | **3.265** | **1.424** | **0.936** | **0.849** |
| Neural Network | 15 | 4.054 | 4.022 | 2.345 | 0.925 | 0.676 |

There are several points worth discussing. Firstly, the prediction accuracy could be improved if the information on Flagfall charges is allowed as a feature in the model training process, given that the Flagfall price is included in the taxi price. Second, an increase in prediction accuracy could also be achieved if access was granted to Google map for retrieving the predicted trip distance and time data. Third, to enable the industrial practicability of the model, interpretability is attached high importance in our model selection process. Despite that the random forest model with 6 features has the lowest RMSE, it is not robust as its validation RMSE double its in-sample RMSE. Gradient boosting model with 15 features seems to be more optimal because its RMSEs do not largely differ from those of the random forest model, its accuracy measures are higher, and its robustness makes it better for generalisation.

# Insights

## FEATURE ATTRIBUTION

Interpreting sophisticated non-parametric models is often difficult (James *et al*, 2017). It is possible to examine the importance of various features in the model on deriving the prediction of an observation, and to get a sense of the importance of that feature to overall model performance (Lundberg, 2018; Molner 2020). The Shapley value is one such metric that enables this comparison, and has many favorable properties, such as consistency, which contribute to its reliability (Molner, 2020). An observation’s total Shapley value is the sum of the Shapley values across all features for that observation (Molner, 2020). A single feature’s Shapley value (for a single observation) could be thought of as the contribution of that feature towards the prediction (Molner, 2020). Positive Shapley values indicate a positive correlation between a prediction and the feature, and negative Shapley values indicate a negative correlation between a prediction and the feature (Choudhary, 2019).

Figure 15 summarises the Shapley Values for the best Gradient Boosting Model as applied to the training data set. Each row represents a feature; each observation appears as a dot on each row; it’s row-wise position indicates the respective feature Shapley value; whilst the color indicates the overall feature value (or attribution – the contribution of that feature to the total Shapley value of the observation) (Lundberg, 2018). The density of the dots indicates the frequency of observations clustering around a feature Shapley value (Lundberg, 2018).

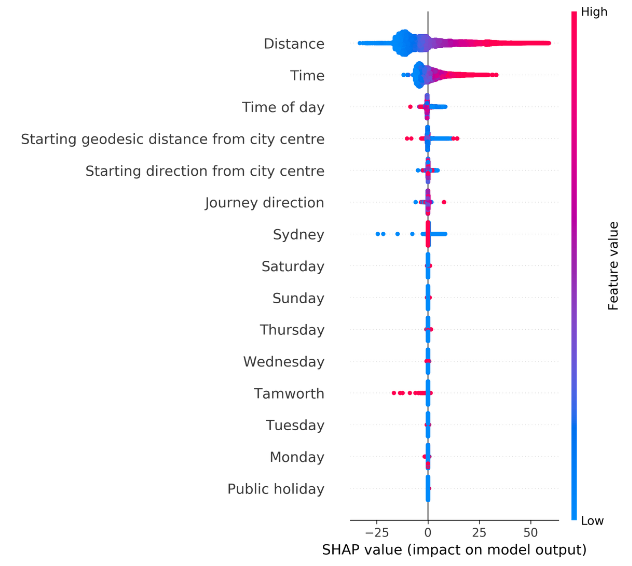


Figure 15. Shapley values summary.

Analysis of Figure 15 yields that the top five variables have the most substantial impact on the model’s predictions. There are a large number of trips for which the fare prediction is not attributed by distance nor time. However, there are a few trips for which the fare prediction is substantially attributed to distance and time – resulting in an overall high importance of trip distance and trip time to prediction. This is confirmed in Figure 16 and Figure 17, which examine the relationship between Shapley values and trip distance and trip time on respective scatter plots.

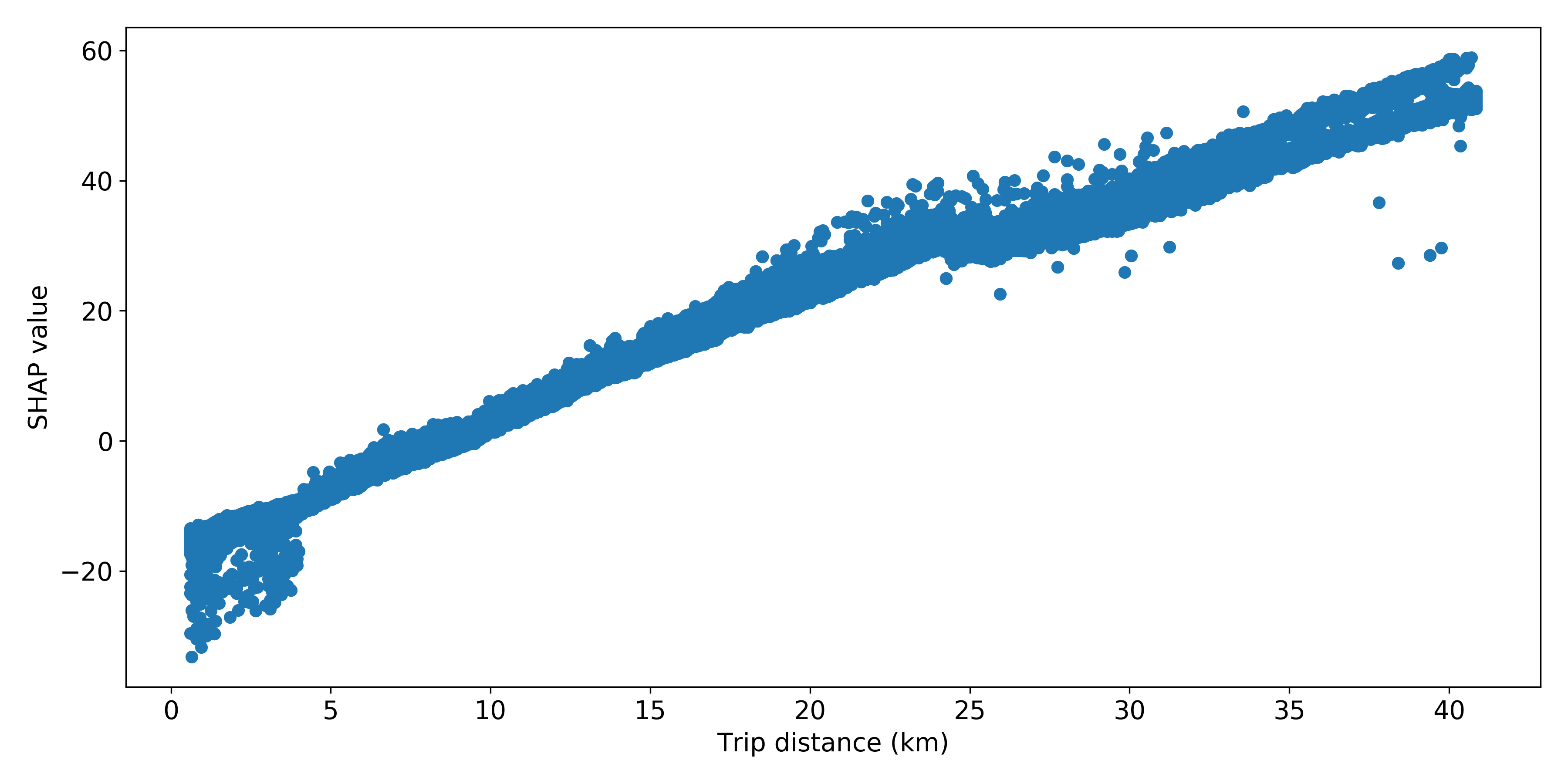


Figure 16. Shapley values for trip distance.



Figure 17. Shapley values for trip time.

Figure 16 and Figure 17 also confirm the positive correlation between trip distance/time and total Shapley value (respectively) predicted by Figure 15. Symmetrical data would result in the mean feature value having a Shapley value of 0, with the extreme lower and upper features values having the same absolute value Shapley values (the lower Shapley value being negative for positively correlated data). However, our data set is skewed, with extreme data points extending the upper tails of the distributions for both distance and time (as well as fare). The higher feature value (of trip distance and time) of those upper tail observations (see Figure 15) possibly indicates that the model is appropriately capturing their influence.

Figure 15 also predicts that time of day and origin displacement from City center are negatively correlated with their respectively Shapley values. Figure 18 and Figure 19 confirm this.



Figure 18. Shapley values for time of day.

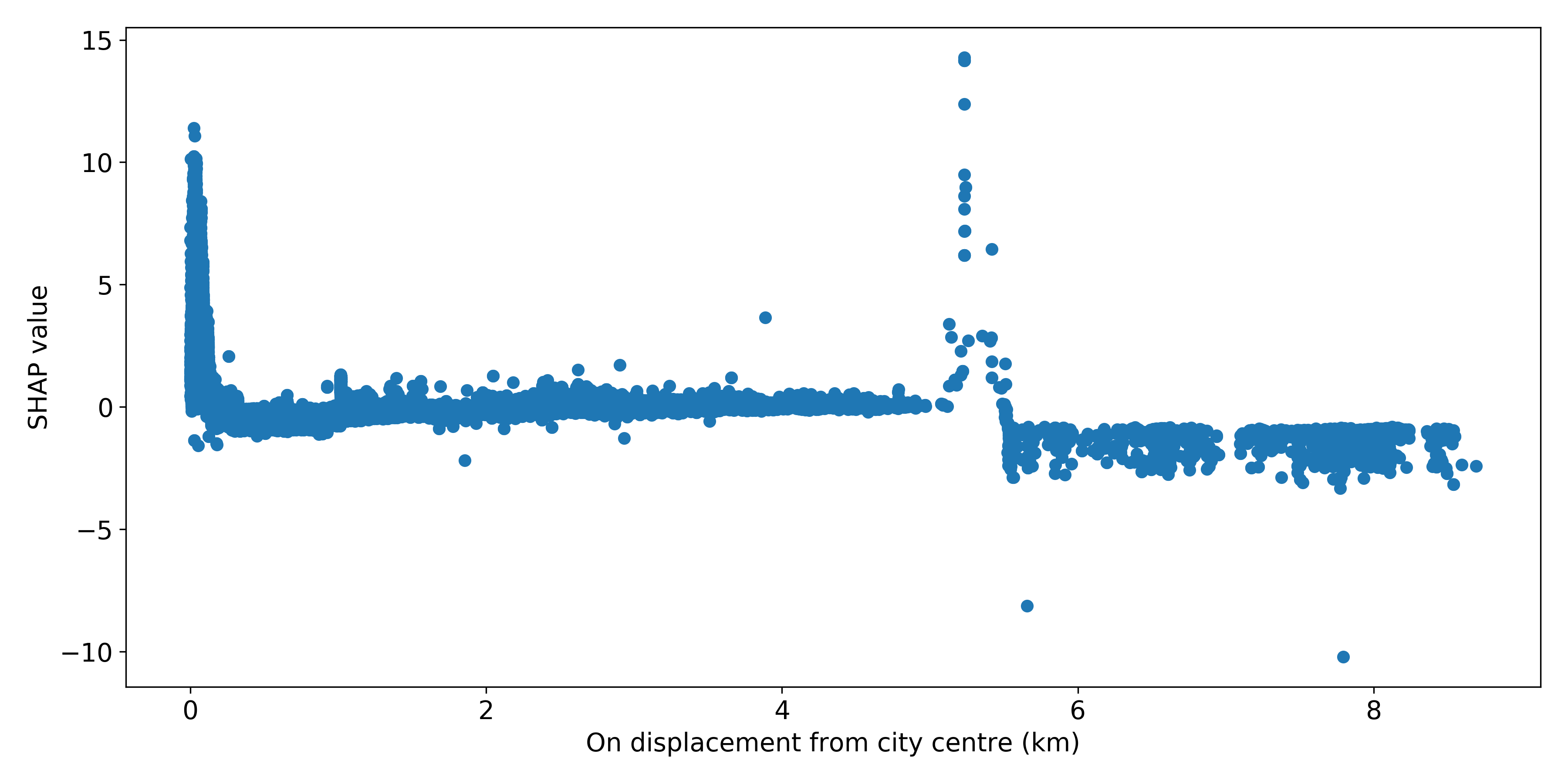


Figure 19. Shapley values for pick-up displacement from city centre.

In fact, the night-time tariff (as per the legislation, NSW Government, 2018) coincides with the added feature value of time of day when time is between 10pm and 6 am. When origin displacement from city center is low (less than 300 m) the feature value is substantial.

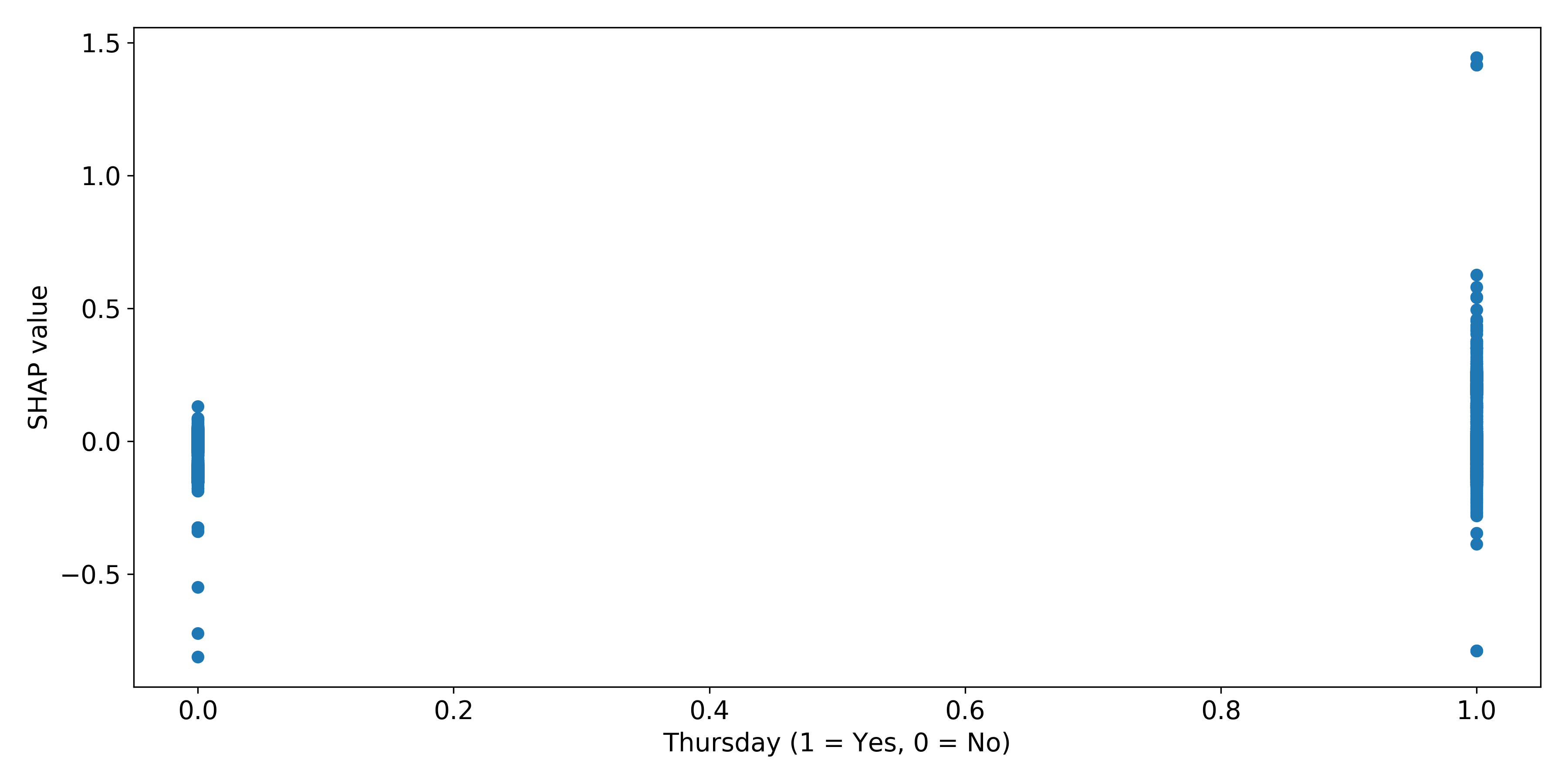


Figure 20. Shapley values for Thursdays.

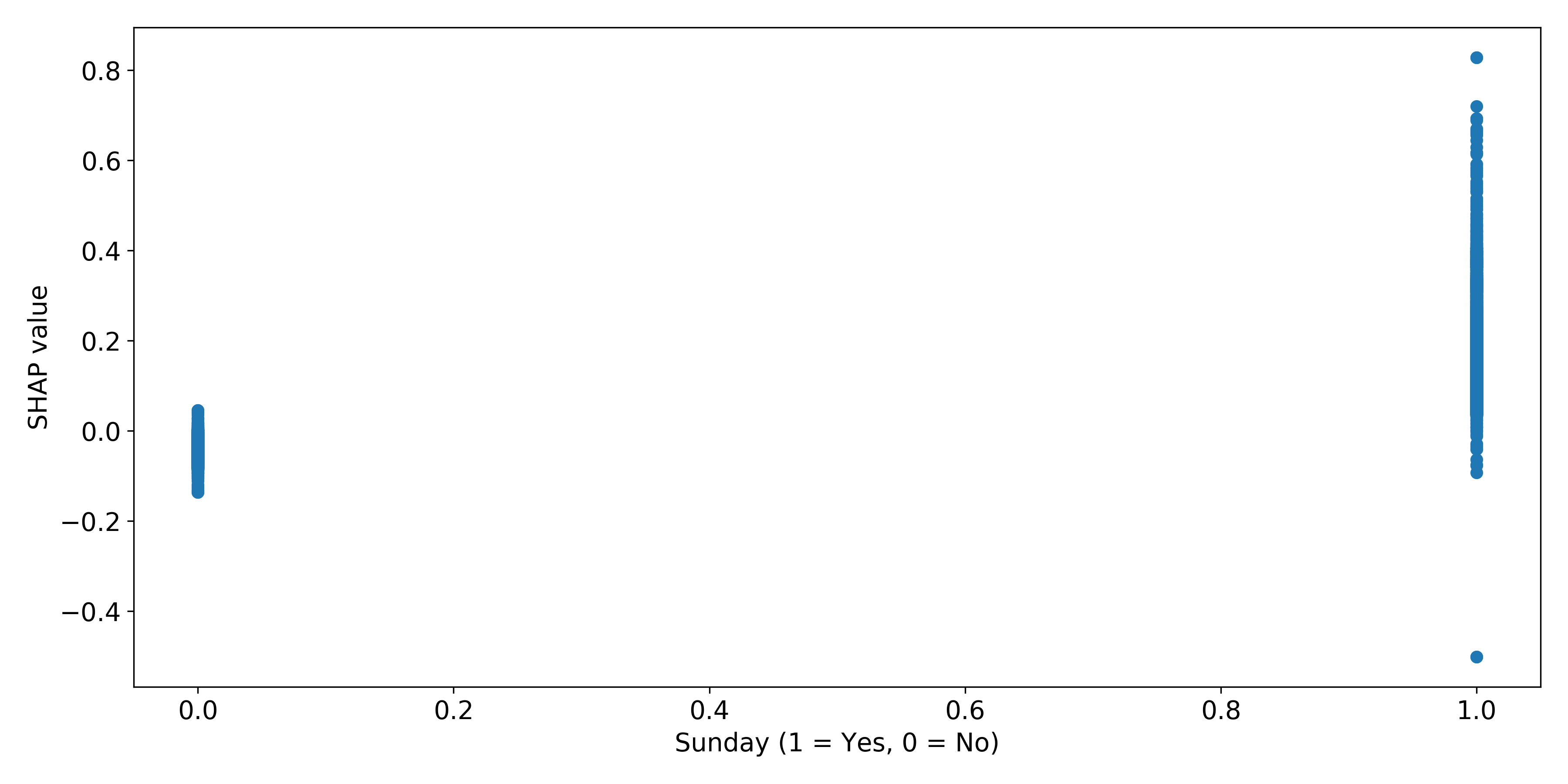


Figure 21. Shapley values for Sundays.

The remaining ten features appear to contribute only slightly to model prediction. Figure 21 and Figure 21 show that the feature value of days of the weeks (Thursday and Sunday included for representation) is quite low.

## Case Studies

To further explore the outcome of the final prediction model, this section presents some case studies highlighting different performance in various segments of data.

Sydney trips were segmented into 30 clusters using K-Means clustering, based on the pick-up time of day and the pick-up and drop-off locations (as latitude and longitude values). The features of each of these clusters as well as the model performance were explored briefly for the purpose of extracting five case studies for further illustration. The distribution of pick-up times for each case are illustrated in Figure 22, and the pick-up and drop-off locations are illustrated in Figure 23 and Figure 24, respectively. In general, Case 1 (blue) represents morning commuters in the North-West area of Sydney; Case 2 (orange), the morning commute from the Western Suburbs towards the city; Case 3 (green), the afternoon commute in Sydney’s East; Case 4 (red), early morning trips in the Greater Western Sydney area; and Case 5 (purple), afternoon and evening trips in the Penrith area.

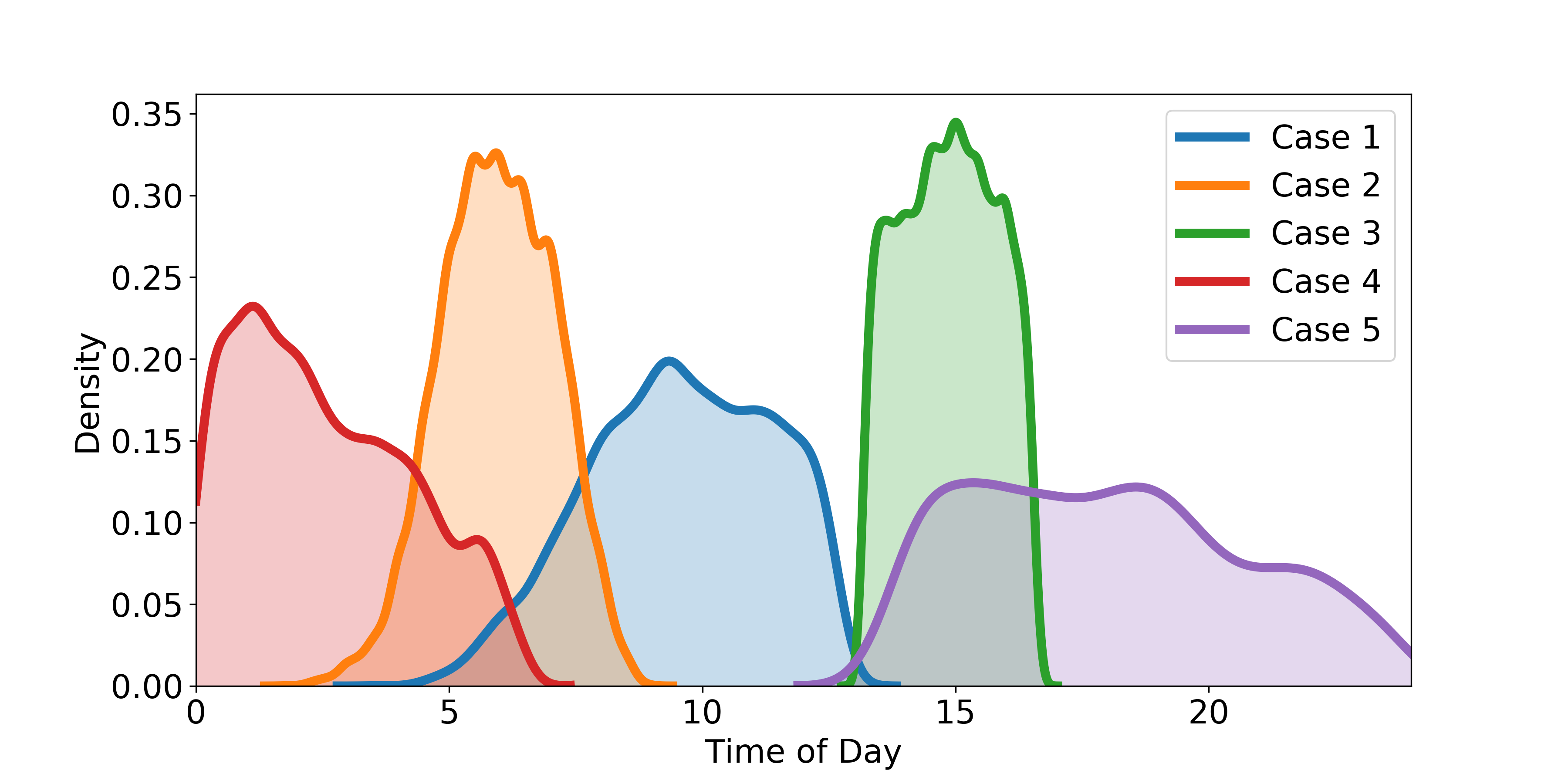


Figure 22. Distributions of pick-up times of day for each case study case.

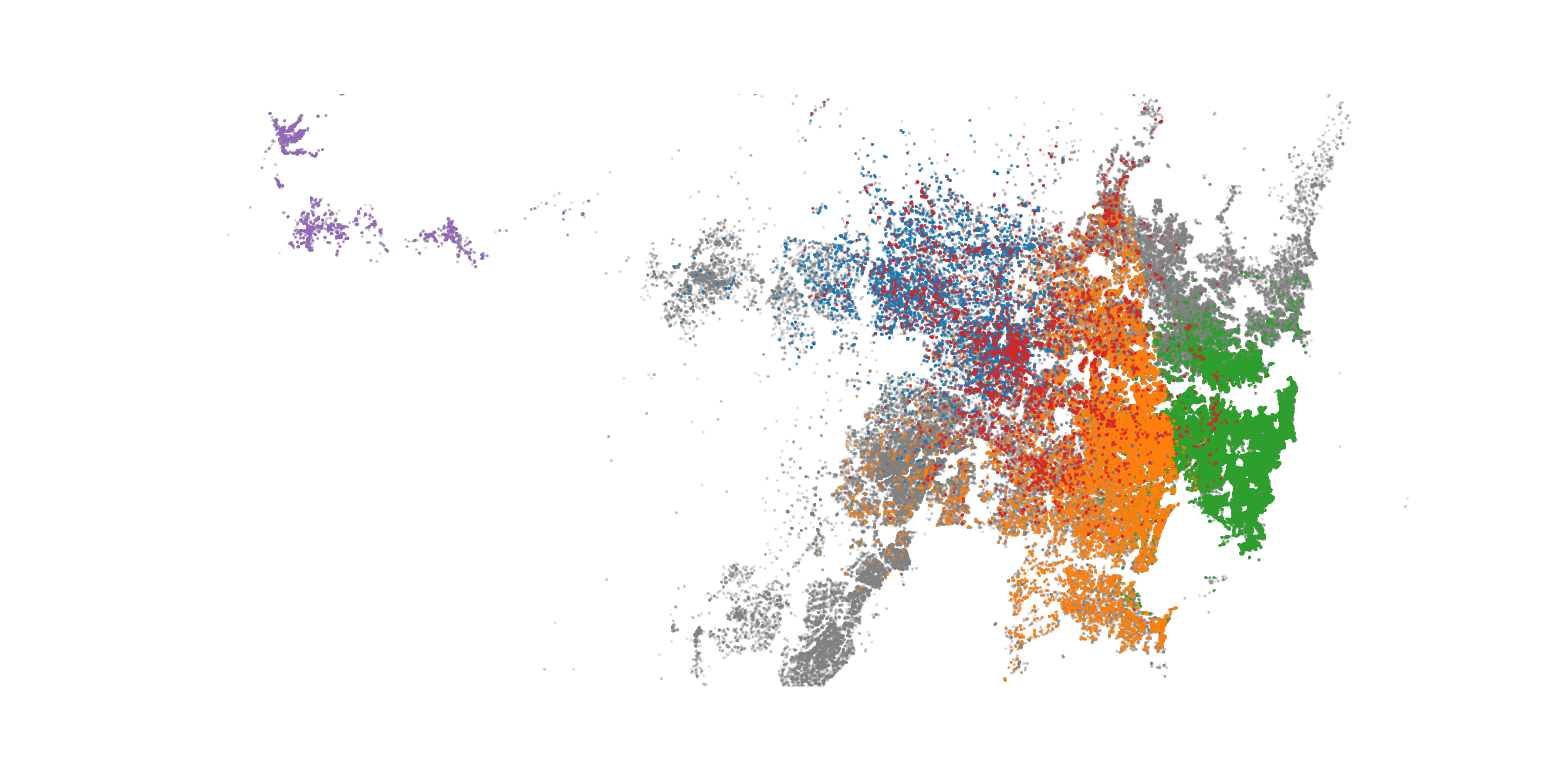


Figure 23. Pick-up locations of each case.

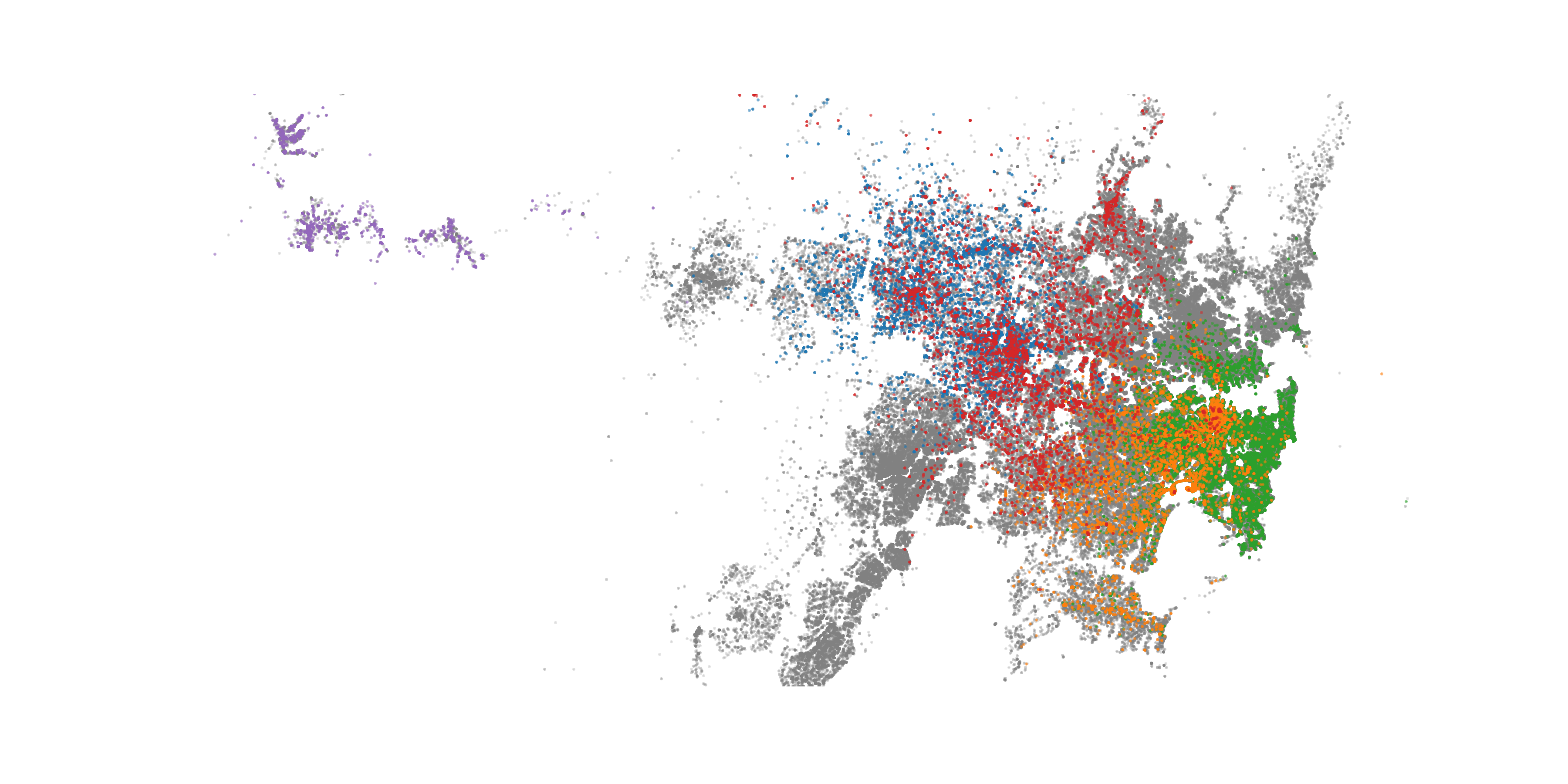


Figure 24. Drop-off locations of each case.

Figure 25 illustrates some properties of each case. Namely, the average prices, price per kilometer, and price per minute. A grey line in each figure shows the average values over all of Sydney’s trips. The high price of Case 2 in Figure 25 (a) is due to the longer trip distances. It is expected that trips with higher prices per kilometer would have lower prices per minute since these would be shorter trips, and vice versa. This result is reflected in Figure 25 (a) and (b) by all cases except Case 5, which is not in an urban area and so is charged at a lower rate (NSW Government, 2018

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 1. Average price | 1. Average price per kilometer | 1. Average price per minute |

Figure 25. Properties of the case study cases.

Figure 26 and Figure 27 highlight the performances of each case. Figure 26 shows the distribution of the prediction errors, where for each sample,

This information is also summarised by the accuracy and RMSE. It is expected that a generally better set of predictions would have a high accuracy and a low RMSE and vice-versa. This is true for Cases 1, 3, and 4. However, Case 2 has a high accuracy (Figure 27 (a)) and a high RMSE (Figure 27 (b)), which can be explained by Figure 26 where there is a large amount of well-predicted (large spike in prediction errors near zero) resulting in high accuracy, but many outliers draw up the RMSE. Conversely, the spread of the majority of data in Case 3 is flatter, lowering the accuracy; while the presence of fewer outliers retains a low RMSE.

In summary, the predictive performance of the final model is varied based on the data’s features.

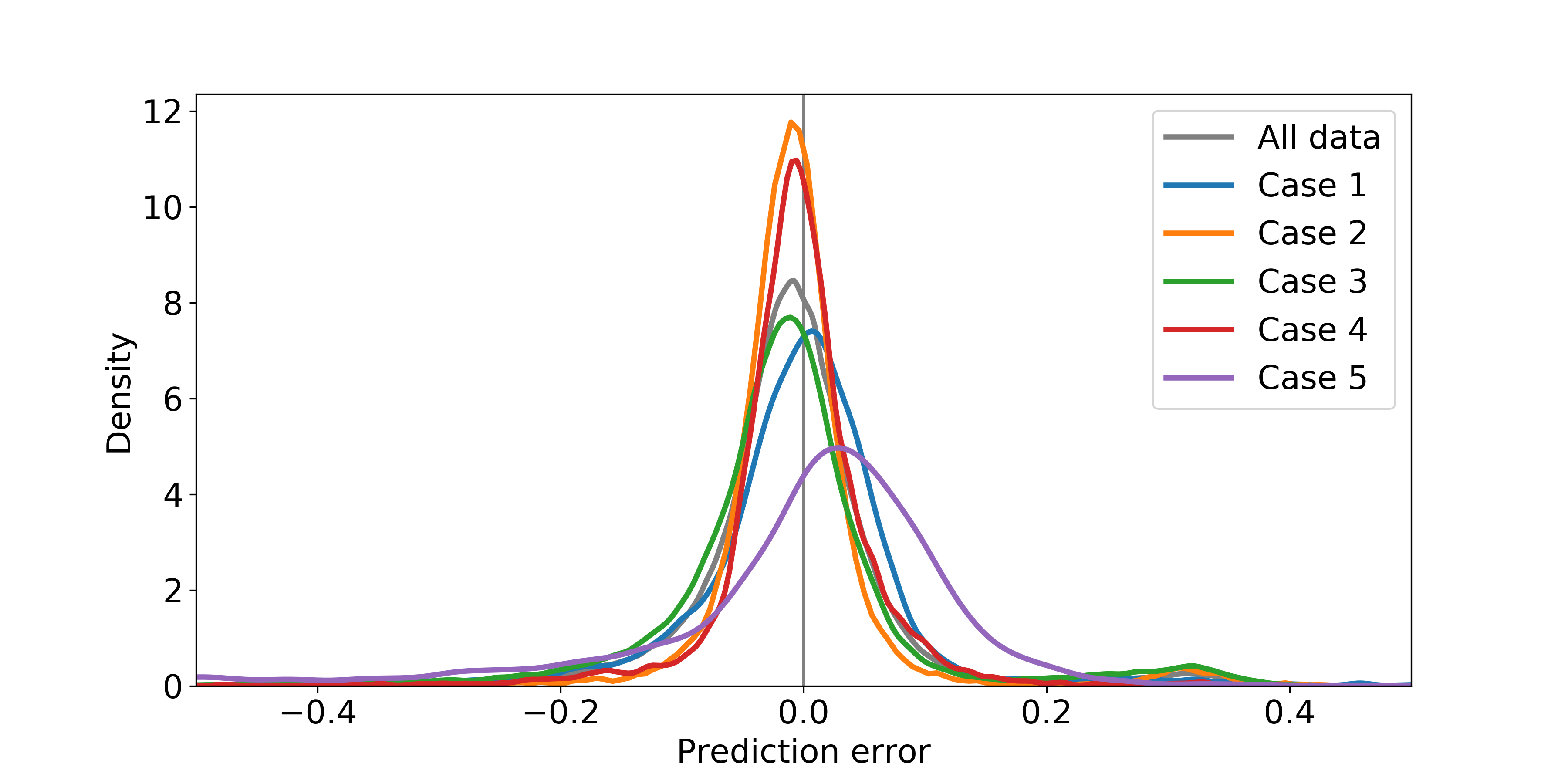


Figure 26. Distributions of the prediction errors for each case.

|  |  |
| --- | --- |
|  |  |
| 1. Accuracy within 10% | 1. RMSE |

Figure 27. Prediction measures for each case.

## Flagfall

The performance of the prediction model is hindered by the exclusion of information relating to the type of taxis. The fixed price of trips, flagfall, captures whether trips are taken in the country verses the city, night or day, and whether the taxi is a maxi taxi or a handicap vehicle. Figure 28 depicts the distributions of the prediction errors categorized by the flagfall values. There is a clear indication that the charged price of maxi taxi and handicap trips are being consistently under-estimated. This finding is in alignment with maxi and handicap taxis charging at a higher rate, on top of the extra fixed start fee. Unfortunately, no predictor variables in the data set capture whether the tax is a maxi or handicap vehicle, despite this attribute being fixed and known for respective taxis.

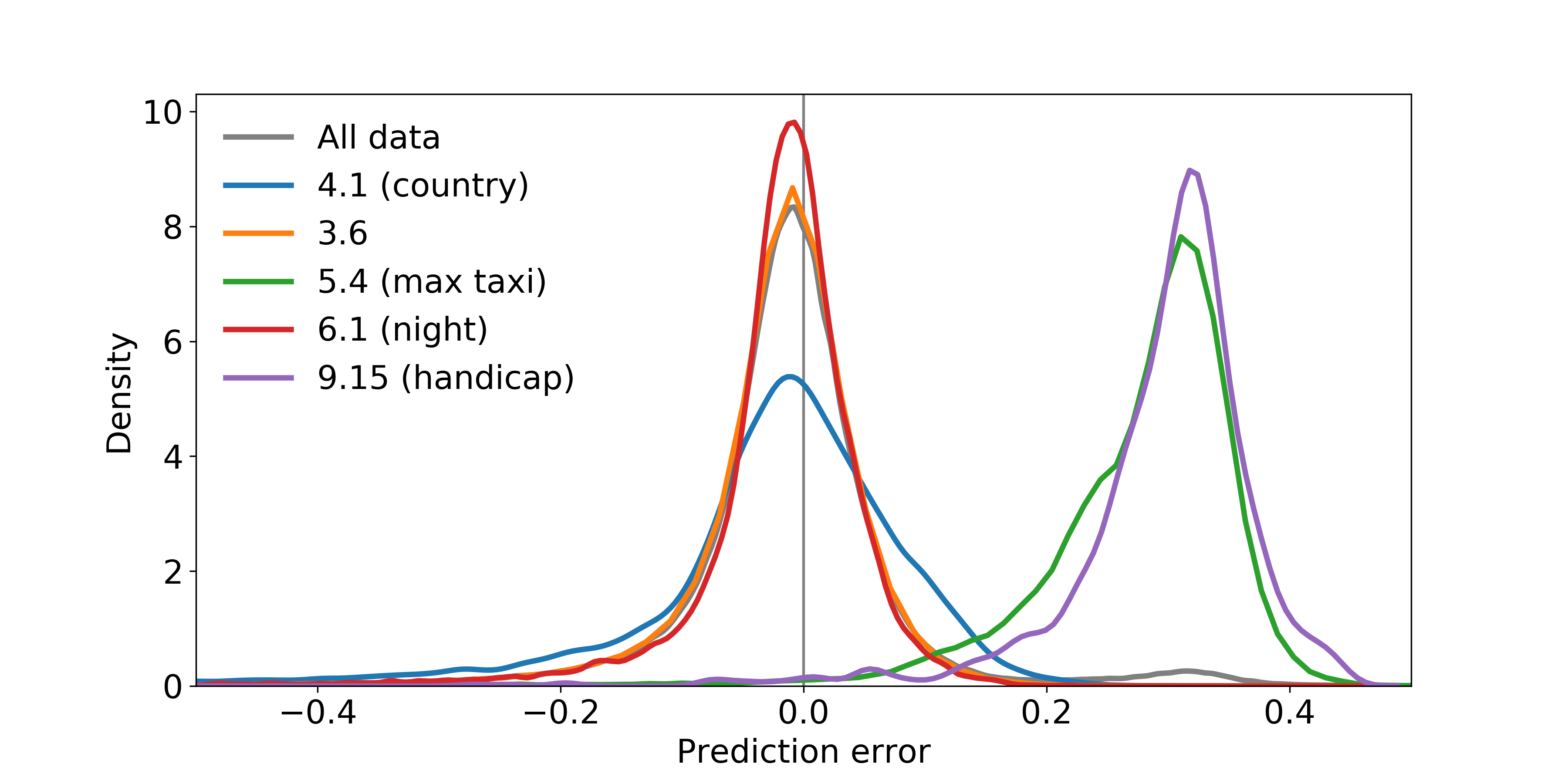


Figure 28. Distribution of prediction errors categorised by flagfall.

With the addition of a binary variable to indicate if a taxi is a maxi or handicap taxi based on the flagfall the prediction model can be improved to obtain a validation RMSE of 2.087, which is a 36% improvement. The accuracies within 10% and 20% improve to 88.9% and 96.5%, respectively (from 84.9% and 93.6%).

## Limitations

While trip distance and time measures were used as predictors in the models, this information is not strictly available at the time of booking taxis since it is measured during the trip. Therefore, the assumption is made that these values will be estimated externally. These estimations may be obtained by integrating the Google Maps or similar API into the system to calculate routes and subsequent predicted distance and time of the trip. However, this will add variance into the model, and will likely weaken the performance to an extent.

# Conclusion and recommendations

Through the comparisons across 8 different models, the best model is decided to be the gradient boosting model with 15 features. This final model results in a RMSE score of 3.265. Using the model, trips can be predicted to within 10% of the true value in 84.9% of cases, and within 20% in 93.6% of cases.

To further improve predictive performance, data pertaining to the type of taxi (a regular or a maxi or handicap taxi) should be included as a predictor. This would be reasonable, since the taxi’s type is fixed and can be known at the time of booking, especially if maxi or handicap taxis are requested in the booking. This addition would improve the model’s RMSE by 36% and bring the number of fares correctly estimated within 10% up by 5%.

An API such as Google Maps should be integrated into the booking system so that distance and time can be estimated before the respective trips and used to predict the fare. Although the estimation of parameters will reduce the performance of the model, the outcome will be more realistic.