Transportation Research Part C 114 (2020) 377–390

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

A class-specific soft voting framework for customer booking prediction in on-demand transport☆

Le-Minh Kieu[a,*], Yuming Ou[b], Long T. Truong[c], Chen Cai[d]

[a] University of Auckland, Auckland, New Zealand
[b] University of Technology Sydney, Sydney, NSW, Australia
[c] La Trobe University, Melbourne, VIC, Australia
[d] Data61, CSIRO, Sydney, NSW, Australia

ARTICLE INFO

Keywords:
Ensemble
Customer booking prediction
Soft voting

ABSTRACT

Customer booking prediction is essential for On-Demand Transport services, especially for those in rural and suburban areas where the demand is low, variable and often regarded as unpredictable. Existing literature tends to focus more on the prediction of demand for traffic, classical public transport, and urban On-Demand Transport service such as taxi, Uber or Lyft, in areas with higher and less variable demand, in which popular time-series prediction methods can be employed. This paper proposes an ensemble learning framework to predict the customer booking behaviour and demand using the observed data of a suburban On-Demand Transport service where data scarcity is a challenge. The proposed method, which is called as Class-specific Soft Voting, is found to be the most accurate prediction method when compared to popular supervised classification methods such as Logistic Regression, Random Forest, Support Vector Machine and other ensemble techniques.

1. Introduction

On-Demand Transport (ODT), also known as dial-a-ride or demand responsive transport, is a shared mobility transport service. ODT provides door-to-door transport to individual passengers. It is also a strong solution for the first mile and last mile problem for getting to and from a nearest transit hub. ODT has recently flourished in many cities and become an important component of modern transportation systems.

ODT services can generally be classified into two types. The first type of ODT focuses mainly on metropolitan areas of high population densities, which can be defined as Urban On-Demand Transport (UODT). In UODT, passengers only request for a service minutes before the desired pick up time, or hail for a ODT vehicle in real-time. Examples of UODT services are taxis, Uber and Lyft. Vehicles of UODT are usually small passenger cars, and in case of Uber and Lyft, belong to the drivers. The vehicle fleet and their exact routing are not strictly optimised in advance. Passenger assignments to vehicle are usually done in real-time by transmitting the customer travel request to all available vehicles nearby and waiting for a voluntary commitment from a requested vehicle.

The second type generally operates in suburban and rural areas where fixed-route public transport are limited (Li and Quadrifoglio, 2010), which we define in this paper as Suburban On-Demand Transport (SODT). Due to the limited customer demand

☆ The majority of this work has been done while the first author Le-Minh Kieu was in Data61, CSIRO.

* Corresponding author.
E-mail addresses: minh.kieu@auckland.ac.nz (L.-M. Kieu), yuming.ou@uts.edu.au (Y. Ou), L.Truong@latrobe.edu.au (L.T. Truong), chen.cai@data61.csiro.au (C. Cai).

https://doi.org/10.1016/j.trc.2020.02.010
Received 11 September 2018; Received in revised form 25 May 2019; Accepted 11 February 2020
Available online 25 February 2020
0968-090X/

within a less populated area and a large service coverage, SODT often asks customers to book their trips a few hours to a few days in advance, so that SODT vehicles would be scheduled for servicing. This type of service is usually regulated and subsided. SODT usually has a predetermined vehicle fleet with the vehicle capacity varying from a small car to a minibus. Some SODT systems have a set of compulsory stops and some reference schedules, also known as *semi-flexible systems* (Errico et al., 2013), while others operate with no predefined schedules and stops.

This paper focuses the problem of anticipating near future customer demand for SODT services, for a medium-term period of 1–14 days in advance. This problem is important because the vehicle fleet and crew schedules in SODT are often determined in advance to maximise their efficiency. This problem is also unique, because a short-term demand prediction, such as in UODT, may not be as effective to SODT, considering that passengers of SODT usually book their trips a few hours to a few days in advance. A long-term demand prediction, on the other hand, is better for systems that require strategic planning decisions such as definition of service area, transfer points, frequency and fleet size. SODT would benefit from a medium-term demand prediction since customers of SODT usually book their trips a few hours to a few days in advance. Medium-term demand forecasting would assist operational service planning, e.g. routing, vehicle scheduling and crew scheduling, given that demand for SODT in low-density areas can be highly variable in time and space. In the literature, while various studies have attempted to predict demand at an aggregated level, for instance traffic flow (Vlahogianni et al., 2004), public transport passenger demand (Wei and Chen, 2012) and UODT demand (Moreira-Matias et al., 2013; Ke et al., 2017), there is a very limited number of studies on the prediction of customer booking behaviour for SODT. In fact, the demand for SODT is generally regarded as low, variable and unpredictable Gomes et al. (2014). This paper tackles the medium-term demand prediction problem for SODT using a customer booking prediction approach. Instead of predicting aggregated demand (i.e. predicting the sum of passengers within a time period – such as in a regression problem), this paper aims to anticipate the booking behaviours of individual customers (i.e. whether a passenger would book another trip or not – such as in a classification problem). This unique approach brings two major benefits. First, customer booking prediction enables operators to anticipate whether a passenger will book their next trip, so that vehicles may be strategically located near customer locations. Second, analysing customer booking behaviour enables operators to understand the mobility needs, the factors that lead to customer booking or perform customer churn analysis for SODT services.

This paper is one of the first studies that aims at the demand prediction for SODT using a customer booking prediction approach. The contribution of this paper is twofold: (1) we develop a predictive ensemble learning framework that deals with predictive analysis of customer demand, and (2) we provide an analysis of SODT passenger booking behaviour from realworld data.

The rest of this paper is organised as follows. Section 2 reviews the related studies in literature. Section 3 describes the prediction algorithms that will be used, before a new ensemble method are proposed for the predictive analysis of passenger demand for SODT services. Section 4 shows the numerical experiment setup and results, follows by the discussion in Section 5, and the conclusion in Section 6.

2. Related work

Demand prediction is an emerging topic in transportation due to its importance in transportation planning, control and demand management. Demand prediction of both private transport (Vlahogianni et al., 2004) and public transport (Wei and Chen, 2012) are of high interest.

The prediction of UODT demand is also an emerging topic in the literature, as the result of the proliferation of taxi and car-hailing services such as Uber and Lyft. New data sources from these services have provided unprecedented opportunities to gain insights into ODT systems and augment decision making. For instance, there are an emerging number of studies in the literature on predicting ODT demand using a time-series analysis technique, because the aggregated customer demand at each time step are often assumed to be temporally correlated. Algorithms such as Auto-Regressive Integrated Moving Average (ARIMA) and some recent deep learning methods such as Long Short-Term Memory (LSTM) are generally effective for predictive analysis of time-series variables. Moreira-Matias et al. (2013) proposed a sliding-window ensemble framework, which consists of a time-varying Poisson Regression, a weighted time-varying Poisson Regression and an ARIMA model to predict the real-time taxi demand at individual taxi rank. The demand at each rank is considered as a time-series. An experiment using the taxi data in Porto, Portugal shows that the proposed Ensemble framework outperforms the base models. On a similar problem, Yao et al. (2018) recently proposed a Deep Learning framework to predict the taxi demand in Guangzhou, China. The proposed Multi-View Spatial–Temporal Network (DMVST-Net) framework consists of a temporal view by LSTM, a spatial view by convolution neural network (CNN) and a semantic view by structural embedding. Using a similar dataset from the online car hailing service Didi Chuxing (in China), Ke et al. (2017) developed another fusion of LSTM and CNN, which consists of multiple convolutional LSTM layers, standard LSTM layers and CNN layers. A random forest is then used for feature selection. Similar to Yao et al. (2018), Ke et al. (2017) also concluded that the Deep Learning architecture significantly improves the prediction performance.

The majority of existing studies related to SODT aim to solve optimisation problems in ODT services, such as vehicle routing (Berbeglia et al., 2007). Nevertheless, there are studies providing insights into the demand of SODT systems. For instance, there is a rich body of literature focusing on finding a balance of perspective passenger demand where SODT is most effective compared to a traditional fixed-route public transport system. Daganzo (1984) proposed a checkpoint SODT system where pick-up and drop-off locations are limited to check-points. The author concluded that fixed-route transit system is more suitable for areas of high demand, whereas ODT performs best in areas of low demand. Similar conclusion has been drawn from Nourbakhsh and Ouyang (2012) and Quadrifoglio and Li (2009), among others.

The existing predictive analysis developed for UODT services are not directly applicable to the prediction of SODT demand,

especially when customer booking behaviour is of main interest. Existing aggregated prediction models tackles the problem as a regression model, in which they aim to predict the sum of passengers within a short time period of minutes to hours. This aggregated approach is not suitable for the sparse, highly variable and low demand of a SODT service, where the sum of demand in a short time period will be very low. Instead, passengers of SODT tend to rebook multiple trips within a multiple days period. This fact leads to a good, unique longitudinal data about customer booking behaviour of SODT. This paper aims to exploit this unique data to develop a new approach to predict the passenger demand for SODT from the customer booking behaviour perspective. By looking at the trip history from each passenger, we define this problem as a classification problem, to predict whether the passenger would book another trip within a certain number of days. This analysis would enable SODT operators to anticipate the near future demand and understand customer booking behaviours.

Perhaps the most similar studies to ours are those aiming at modelling the customer behaviours of ODT services. The majority of these studies used a stated preference survey data to investigate the customer behaviour or susceptibility to use ODT services. Benjamin et al. (1998) developed a logit model with latent factors to evaluate the impact of ODT services passenger mode choice behaviour. Al-Ayyash et al. (2016) developed a mixed logit model to exploit customer socioeconomic characteristic, stated preference and perception indicator to evaluate the potential customer demand of a Shared-Ride Taxi service. The generality of studies are limited to the case study in the stated preference surveys and fail to capture the long-term impacts of changes in customer behaviour. Among the survey-independent studies, Jain et al. (2017) estimated the long-term customer demand for ODT using the spatial variation of demographic characteristic and travel behaviours.

The approach in this paper is novel because it aims to predict a short-term customer demand for SODT services. The proposed approach is generic, survey-independent and applicable to any other SODT services, given data availability. We will also develop an adaptive ensemble method to exploit and incorporate the predictive power of multiple classification algorithms to predict the customer booking behaviour for SODT services. Finally, customer booking behaviours are analysed from the data, showing insights into the reasons why or why not passengers book a SODT trip.

3. Methodology

The customer booking prediction problem in this paper answers the question of whether a customer i would book another trip within D days. Here D can be any integer number. For instance, D equals one means that the problem will be the prediction for tomorrow's booking, and D equals seven means the prediction for the next week.

We define this problem as a binary classification problem with two classes, i.e. whether the customer i would book another trip within D days, or not. A historical trip that customer i made on day d is viewed as an object represented by feature vectors X_i^d, so that it is possible to apply statistical classifiers for the defined classification problem. The feature vector $X_i^d = [x_{1i}^d, x_{2i}^d, ...,x_{Ki}^d]$ may include variables like travel time, delay, actual waiting time, booking time and fare. The training set for the classifier is simply compiled by looking at all the trips made by the customer i from the day d to day $d + D$ to identify whether the customer i booked another trip or not.

The following sections describe different approaches that we will employ as the supervised classifier to solve aforementioned customer booking prediction problem. We will then introduce the concept of an ensemble algorithm to incorporate the predictive power of multiple classifiers and finally propose the Class-specific Soft Voting (CSV) algorithm for the ensemble.

3.1. Supervised classifiers for customer booking prediction problem

Let Y be a binary response variable where $Y_i^d = 1$ if the customer i book another trip after D days from the day d, and $Y_i^d = 0$ otherwise. Recall that $X_i^d = [x_{1i}^d, x_{2i}^d, ...,x_{Ki}^d]$ is a feature vector of explanatory variables. Naturally X_i^d would be highly correlated with Y_i^d, e.g. heavily delayed trips may discourage the customer to book another trip. A good binary classifier should find the correlation between X_i^d and Y_i^d during training and perform accurate prediction of Y_i^d on unseen X_i^d.

The most popular binary classifier is **Logistic Regression (LR)**. This model is closely related to the logit models proposed in (Benjamin et al., 1998 and Al-Ayyash et al., 2016). LR estimates the probability that the customer i would book another trip within D days. The probability is estimated as:

$$\pi_i^d = \Pr\left(Y_i^d = 1 \middle| X_i^d\right) = \frac{\exp(\beta_0 + \beta_1 x_{1i}^d + ... + \beta_K x_{iK}^d)}{1 + \exp(\beta_0 + \beta_1 x_{1i}^d + ... + \beta_K x_{iK}^d)} \tag{1}$$

The next algorithm to be applied is **Random Forest (RF)** (Liaw et al., 2002). RF builds multiple Decision Trees, where each is a tree-like graph to show possible consequences. A Decision Tree when being trained with a dataset with targets Y and features X will formulate some set of rules, i.e. each node in a Decision Tree has a decision based on whether or not a feature $x_{ki} > a$ for a fixed a. A RF is a combination of multiple Decision Trees where each tree provides its "vote" for the classification task. In order to add randomness into the model and avoid over-fitting problem, RF only takes a random subset of the features into consideration in training each tree, and enhances classification accuracy by having many more trees in its structure. It takes N members of the training set $\mathcal{S} = \{(x_i, y_i)_{i=1}^N\}$ where $x \in \mathcal{X} \subset \mathbb{R}^F$ to train a model $\mathcal{F}(x)$.

$$\mathcal{F}(x) = \text{argmax} \frac{1}{T} \sum_{j=1}^{T} p_j\left(y \middle| x\right) \tag{2}$$

where T is the number of Decision Tree and $p_j, j \in \{1, ..., T\}$ is a vote from an individual Decision Tree. Eq. 2 shows that RF simply chooses the class with more votes as its prediction result.

Another supervised classifier in consideration is **Support Vector Machine (SVM)** (Steinwart and Christmann, 2008). SVM is a discriminative classifier, which is originally designed for linear two-class classification with margin. The margin is the minimal distance from the separating hyperplane to the closest data points. A linear hyperplane can be formulated as

$$f(x) = \beta_0 + \beta^T x \tag{3}$$

where β is known as a weight vector and β_0 is the bias. They can be scaled to $|\beta_0 + \beta^T x| = 1$ for simplicity. One important and unique feature of SVM is that x represents the training samples closest to the hyperplanes, and are called **support vectors**.

Let the two classes of Y be $\mathbf{A}_+$ and $\mathbf{A}_-$. SVM aims at finding a separating hyperplane to categorise new samples into these two classes. If $\mathbf{w}$ is the normal vector to a pair of bounding planes, then:

$$\mathbf{w}^T x + b \geqslant +1, \textit{ for } x \in \mathbf{A}_+ \tag{4}$$

$$\mathbf{w}^T x + b \leqslant -1, \textit{ for } x \in \mathbf{A}_- \tag{5}$$

We then can convert the problem of training a SVM to a maximisation problem of the margin $\frac{2}{||\mathbf{w}||_2^2}$, which in turn can be done by minimising $\frac{1}{2}||\mathbf{w}||_2^2$. This is a quadratic program and can be solved to optimality.

$$\min_{(\mathbf{w},b)\in\mathbb{R}^{n+1}} \frac{1}{2}\left|\left|\mathbf{w}\right|\right|_2^2 \tag{6}$$

s.t. $y_i(\mathbf{w}^T x + b \geqslant +1)$ for $i = 1, 2, \ldots, m$.

3.2. Majority voting ensemble algorithms

Having multiple classifiers with different strengths and weaknesses, a reasonable way to enhance the prediction performance of the models is to integrate their predicting power to achieve a better classification result. Ensemble learning is a popular divide and conquer technique approach in Machine Learning aiming to combine these algorithms. For classification problem, the most popular ensemble method is majority or plurality voting. The ensemble model simply chooses the class that has the majority of votes from M classifers $\mathcal{F}_m$:

$$\widehat{Y} = mode\{\mathcal{F}_1(x), \mathcal{F}_2(x), ..., \mathcal{F}_M(x)\} \tag{7}$$

In our binary classification problem, assuming that the classifiers have uncorrelated error, the following Lemma can also be proven.

Lemma 1. *In a binary classification problem, the majority voting ensemble will always lead to an improvement in classification results if each classifier has a minimum 0.5 probability of giving a correct classification.*

Proof. Given M classifiers for a binary classification problem, the ensemble algorithm will give a correct answer if at least $\lfloor \frac{M}{2} + 1 \rfloor$ classifiers give the correct classification. Assume that each classifier has a probability p of giving the correct classification, the probability of having K correct classifiers out of M classifiers is:

$$P_{ens} = \sum_{K=(M/2)+1}^{M} \binom{M}{K} p^k (1-p)^{M-K} \tag{8}$$

as $M \to \infty$ we have

$$P_{ens} \to 1 \quad \text{if } p > 0.5 \tag{9}$$

$$P_{ens} \to 0 \quad \text{if } p < 0.5 \tag{10}$$

In majority voting, each classifier gives a vote first, which is later linearly combined in the ensemble algorithm. The ensemble model then relies on good individual classifiers. Majority voting reduces the variance in classification outcomes, because multiple classifiers are making decisions instead of one. However, majority voting would only improve the classification accuracy if multiple similar classifiers are employed. For instance, a majority voting ensemble will follow the classification outcomes of the majority, rather than follow a strong and proven classifier among its members.

A straight-forward extension to the majority voting algorithm when classifiers have different probability of giving the correct answer is weighted majority voting (MV):

$$\widehat{Y} = \underset{i}{\text{argmax}} \sum_{m=1}^{M} w_m \chi_C\left(\mathcal{F}_m(x) = i\right) \tag{11}$$

where χ_C is the characteristic function $[\mathcal{F}_m(x) = i \in C]$, and C is the set of class labels. In a binary classification problem, C equals 2. A common weight for MV is $w_m \propto \mathcal{P}_m/(1-\mathcal{P}_m)$ where individual classifier has the probability $\mathcal{P}_m$ of giving the correct classification. We can simply calculate $\mathcal{P}_m$ by dividing the sum of correct classification outcomes from model m to the total sample size.

Most of supervised classifiers are probabilistic, which means that they are able to output p_{im}, which is the probability that classifier $\mathcal{F}_m$ giving i as the classification label ($i \in C$). This leads to another extension called soft voting (SV):

$$\widehat{Y} = \underset{i}{\text{argmax}} \sum_{m=1}^{M} w_m p_{im} \tag{12}$$

SV combines the outcomes of each classifier before they gives a classification outcome. Besides reducing the variance, SV gives higher weights to strong classifiers (higher $\mathcal{P}_m$), so that the final classification outcome from the ensemble is more likely to follow the strong classifiers.

3.3. Class-specific soft voting (CSV)

MV and SV method, also referred as Generalised Ensemble Method, often yield better performance than classical majority voting and always give a better classification than individual classifiers (Perrone and Cooper, 1995). However, both MV and SV are linear and use a static set of weights w_m. There are several methods that use an adaptive procedure to change w_m, such as AdaBoost (Rätsch et al., 2001). However, the linear ensemble logic in weighted majority voting and soft voting means that the final classification outcome is also affected by weak classifiers, which may give wrong outcomes.

We propose in this paper a framework for ensemble learning by leveraging the Class-specific Predictive Accuracy Value. Let us first define the Class-specific Predictive Accuracy Value.

Definition 1. Class-specific Predictive Accuracy Value is the probability that a model gives accurate classification for a specific class.

If a model F_m gives an answer of class $i \in C$, Class-specific Predictive Accuracy Value ζ_{im} is the probability that i is the actual class among C classes.

In a binary classification problem, ζ_{im} is often referred as Positive Predictive Value (PPV) and Negative Predictive Value (NPV). PPV is the probability that a model gives an accurate positive classification and NPV is the probability that a model gives an accurate negative classification. PPV is slightly more important for the customer booking problem being tackled in this paper, because we should prepare vehicle fleets and crews if customer wants to book a service. Class-specific Predictive Accuracy Value gives a class-specific confidence level for each predictive model.

Our proposed method is best explained as an Expert Voting technique. Instead of linearly combining the classification outcomes of individual classifiers as in MV and SV, it compares a weighted probability of each classification outcome. In particular, the probability outcome p_{im} of model m to each class i is multiplied by the Class-specific Predictive Accuracy Value ζ_{im}. A high ζ_{im} means that the model m is usually correct when giving a label i, while a high p_{im} means that the model m is certain about giving class i as the outcome for the inputs in consideration. We then compared these products between ζ_{im} and p_{im} for every i and m and choose the highest. We define this method as Class-specific Soft Voting (CSV). The following formulation can be proposed:

$$\widehat{Y} = \underset{i \in C}{\text{argmax}} \left\{ \underset{m \in M}{\text{argmax}} \left\{ \zeta_{im} p_{im} \right\} \right\} \tag{13}$$

CSV is an adaptive method because ζ_{im} changes with each classifier's performance against training data. CSV exploits the hypothesis that ζ_{im} are different when model m gives an outcome of different class. For binary classification, this is the same with saying PPV is different to NPV. Some model is more accurate at predicting the positive class, and *vice versa. If* $\zeta_{im} > \zeta_{jm}$, *model m has more confidence in giving a class i classification than a class j outcome. By comparing, instead of combining the results of multiple classifiers, CSV should at least be as accurate as the best classifier of each class.*

For a better explanation, consider a binary classification example with 3 classifiers $\mathcal{F}_1$, $\mathcal{F}_2$ and $\mathcal{F}_3$.

Example 1. The models give the following classifications:

$$\mathcal{F}_1 \rightarrow [0.9, 0.1] \quad \text{with } \zeta_{i1} = [0.85, 0.65]$$
$$\mathcal{F}_2 \rightarrow [0.7, 0.3] \quad \text{with } \zeta_{i2} = [0.5, 0.75]$$
$$\mathcal{F}_3 \rightarrow [0.7, 0.3] \quad \text{with } \zeta_{i3} = [0.65, 0.85]$$

In this example, all model outcomes are more towards the class 1 than the class 2, with the model 1 being highly accurate for class 1 and model 2 and 3 are more accurate when giving class 2 outcomes. The weighted average probabilities are calculated as:

$$p(i_1|x) = \text{argmax}(0.9 * 0.85, 0.7 * 0.5, 0.7 * 0.65) = 0.765$$
$$p(i_2|x) = \text{argmax}(0.1 * 0.65, 0.3 * 0.75, 0.3 * 0.85) = 0.255$$

then the final outcomes of the ensemble can be concluded:

$$\widehat{Y} = \underset{i}{\text{argmax}} \left[p\left(i_1 \middle| x\right), p\left(i_2 \middle| x\right) \right] = 1 \tag{14}$$

Table 1
Description of explanatory and target variables.

Parameters	Description
ID	Passenger ID
BookingDay	Day from 1st January (31 to 83)
BookingTime	Minute from 0:00 (1 from 1440)
BookingType	Whether it is Phone call, Mobile Application or Website
BookingWait	Time from booking to confirmation of booking (s)
ActualWait	Actual waiting time from desired pick-up time to actual pick up time
TravelTime	Actual travel time of the ODT trip (min)
Fare	Binary variable, 0 for Concession and 1 for Standard type of fare
SchedPickUpTime	Scheduled pick up time (min from 0:00)
ActualPickUpTime	Actual pick up time (min from 0:00)
SchedDropOffTime	Scheduled drop off time (min from 0:00)
ActualDropOffTime	Actual drop off time (min from 0:00)
Delay	Actual delay from schedule to actual drop off time (min)
PickUpLocation	Location of the pick up
DropOffLocation	Location of the drop off
Distance	Actual distance of the ODT trip (km)
Num_Trips	Number of trips made by the passenger from the first day
TotalCharges	Total fare charged from the passenger from the first day (AUD)
Rebooked	Target variable: whether the passenger book another trip within D days (Rebooked $\in \{0, 1\}$)

4. Numerical experiments

This section will describe the dataset (Section 4.1). We will then evaluate and compare the base classifiers (Section 3.1), the classical majority and soft voting (Section 3.2) and finally the proposed Class-specific Soft Voting (Section 3.3) using observed data of a SODT service in suburban Sydney, Australia.

4.1. Dataset

The data used for the experiments in this paper is from an SODT service in Northern Beaches, Sydney, Australia. The service is operated by the Transport for NSW and Keolis Downer from November 2017. Interested reader may refer to (TransportNSW, 2018) for more details about the service. We use two-month data of this ODT service from February to March 2018, with an overall sample size of 2845 individual customer bookings. Each data record contains the information of a single trip booking from an individual passenger, with unique trip ID and passenger ID. Other variables include the booking day, time, scheduled pickup time, actual pickup time and other trip related variables. A list of variables included in each data record can be found in Table 1 at the end of this section.

After cleansing the data (removing uncompleted, corrupt or inaccurate data points), the data processing procedure starts with aggregating the raw spatial coordinates. We aggregate passengers' pick up and drop off locations into grid cells to maintain individual's privacy, while maintain the spatial distribution of demand. As shown in Fig. 1, there are hot spots of demand that can be explored. This is the example of customer pick ups for a random day in our data set. The cell with 21 bookings has two stops of a high frequency bus line, where passengers usually book SODT trips to return home.

The next data processing step is to find out for each data whether that particular passenger would rebook another trip after a predefined D days. We varies the value of $D$from 1 day to 14 days. The maximum value of D has been limited to 14 because we need $D + 1$ days to know whether the passenger rebook another trip or not. As D increases, the sample size in the processed dataset reduces.

Fig. 2 shows the probability of rebooking within D days for a random customer, without considering any other modelling or explanatory variables. This plot can be used as a 'random guess' of whether a customer would book another trip when there is no developed model. The figure shows that around 33% of customer would book a trip on the next day, and approximately 70% of them would eventually come back within 8 days.

Table 1 shows the description of variables in the processed data.

4.2. Model development

The understanding of data is an important stepping stone to develop the classifiers described in Section 3.1. We first split the observed data into 80% for training the models, and the remaining 20% for evaluation and comparison.

We choose the parameters for Logistic Regression (LR) by firstly choose parameters which are both important and uncorrelated. Fig. 3 shows the correlation of numerical variables. A few correlated variables, such as *SchedPickUpTime* and *ActualPickUpTime* are not shown.

We then use the list of uncorrelated numerical (Fig. 3) and categorical variables to develop a LR model where all explanatory variables are significant and the model has the highest prediction accuracy. Maximum Likelihood Estimation is the method used for parameters estimation. Table 2 shows the coefficients of the final LR model.

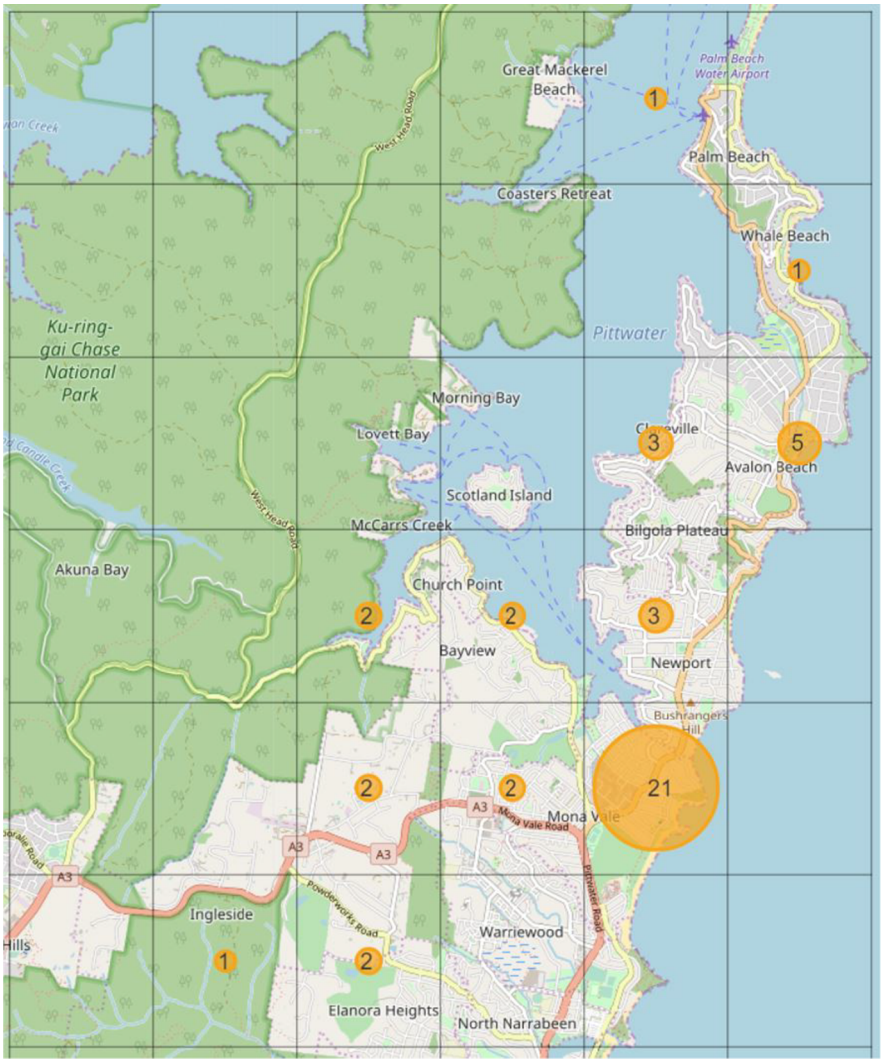


Fig. 1. The study site and the cell-based discretisation of passenger location.

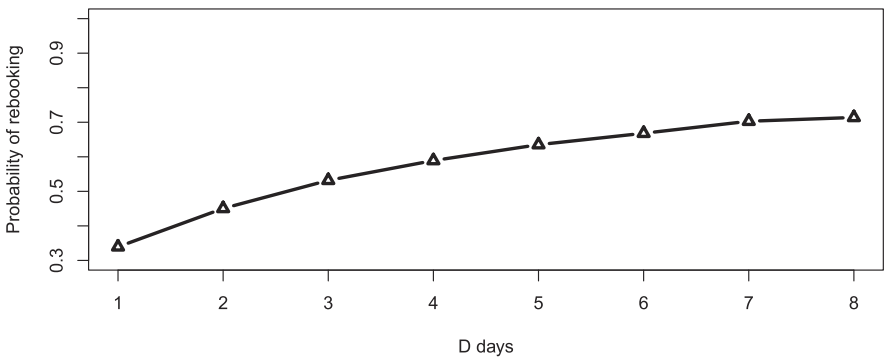


Fig. 2. Probability of rebooking within *D* days.

The Random Forest (RF) model is developed using the **quantregForest** package in R language (Meinshausen and Schiesser, 2007). The model is developed using all variables in Table 1. Fig. 4 shows the top 10 important feature to contribute to the Mean Squared Error (MSE) of the RF model. *Num_Trips* is again the most important variable, followed by *ActualPickUpTime*, *BookingTime* and *Distance*.

The Support Vector Machine (SVM) model is developed using the **e1071** package in R language (Meyer and Wien, 2001). We

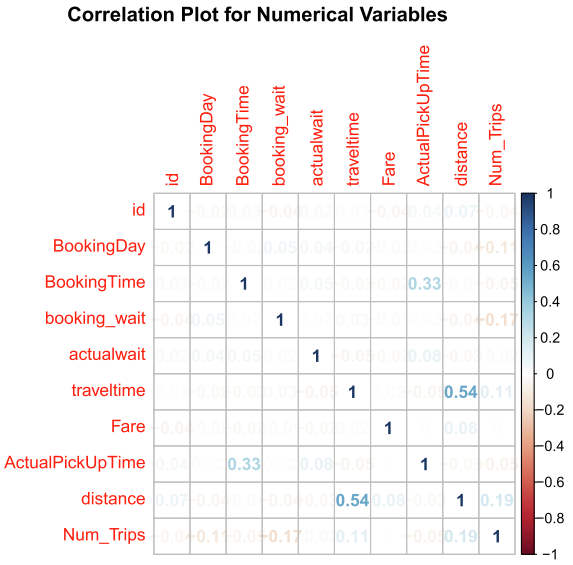


Fig. 3. Correlation of numerical explanatory variables.

Table 2
Coefficients of Logistic Regression model.

Coefficients	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	−0.92657	0.268055	−3.457	0.000547	***
Num_Trips	0.156764	0.01015	15.445	<2e−16	***
ActualPickUpTime	−0.00116	0.000279	−4.145	3.39e−05	***
BookingTime	0.000963	0.000251	3.835	0.000126	***

Significant codes: *** is <0.001.

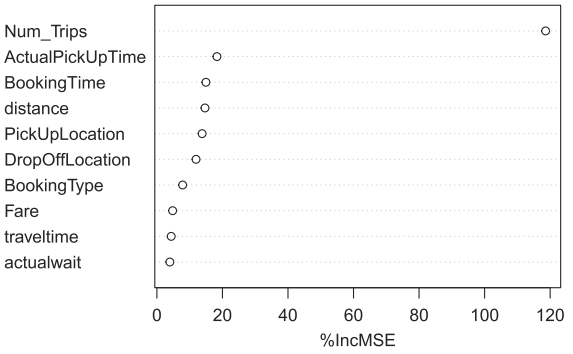


Fig. 4. Top 10 important features for Random Forest classifier.

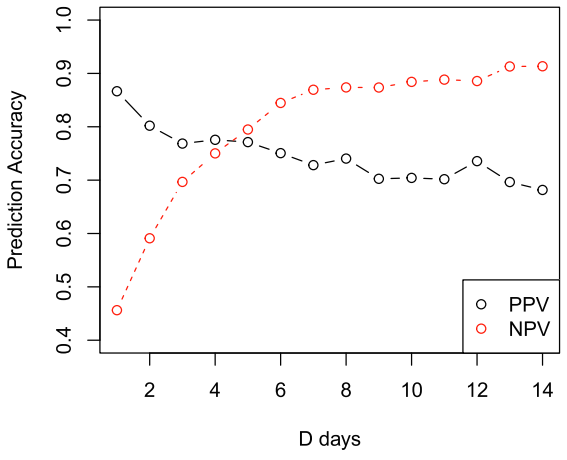
choose *C*-classification as the type of SVM because of its classicaltion performance in the binary problem. The formulation for *C*-classification SVM for binary classification can be written as: (Meyer and Wien, 2001)

$$\min_{\alpha} \frac{1}{2}\alpha^T Q\alpha - \mathbf{e}^T\alpha \tag{15}$$

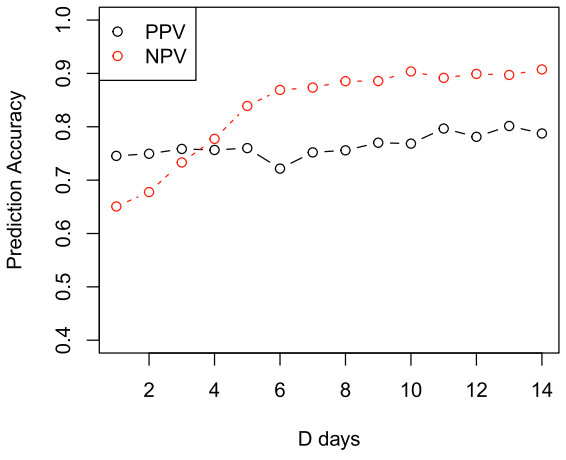
s.t. $0 \leqslant \alpha_i \geqslant UBD,\ \ i = 1,\ ...,l,\quad$ and $y^T\alpha = 0$where $\mathbf{e}$ is the unity vector, *UBD* is the upperbound, Q is an l by l positive semi-definite matrix with $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, and the kernel is $K(x_i, x_j) \equiv \phi(x_i)^T\phi(x_j)^T$. We also choose the linear kernel because of its simplicity and high performance for a binary classification problem.

4.3. Evidence of the difference in class-specific predictive accuracy value

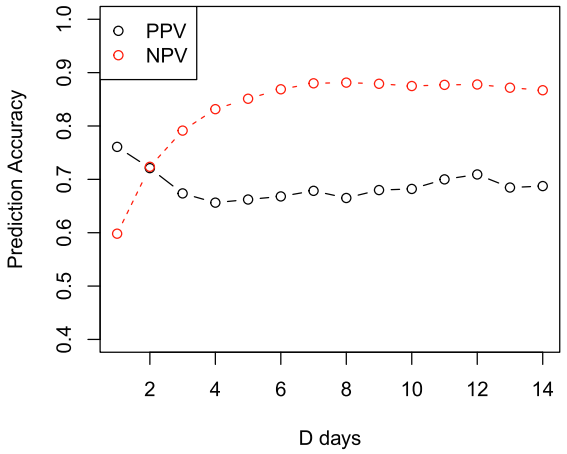
The proposed Class-specific Soft Voting (CSV) ensemble method relies on a hypothesis that the models have reasonably different



(a) Logistic Regression



(b) Random Forest



(c) Support Vector Machine

Fig. 5. Evidence of difference in PPV and NPV.

accuracy when predicting each class in the data. CSV exploits this to provide a more reliable soft majority voting for ensemble models. Fig. 5 shows the progression of PPV and NPV when the number of days D for prediction increases from 1 to 14 days. To increase the reliability of the prediction results, we make 40 replications of each model for each value of D, where the data is randomly split between training and testing data at each replication. The final results in Fig. 5 are the mean values from those replications.

Fig. 5 shows strong evidence of the difference in Class-specific Predictive Accuracy Value. At any value of D, PPV is different to NPV for every model. As D increases from 1 to 14 days, NPV generally increases to nearly 0.9 in all models. Among the three models, LR shows the most variation in PPV and NPV at $D = 1$ to $D = 14$ days.

4.4. Results and comparison

This section compares the prediction accuracy of the base classifiers (LR, RF and SVM), the classical majority voting (MV), soft voting (SV) and the proposed CSV ensemble model. Both MV and SV are implemented using the popular setting for w_m, in which $w_m = P_m/(1 - P_m)$. The prediction accuracy is calculated as the number of accurate prediction over the total number of data points.

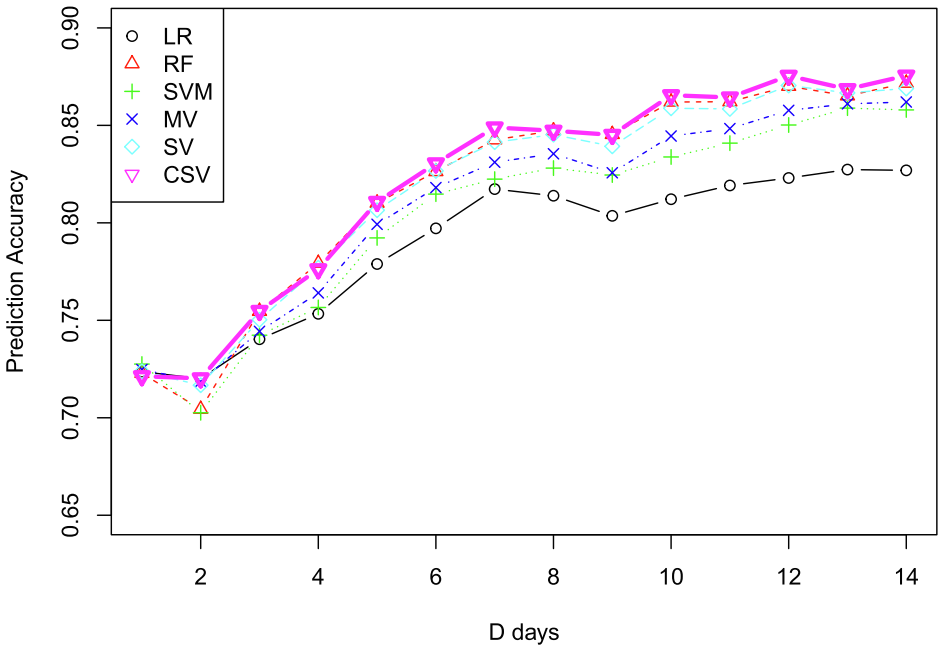


Fig. 6. Comparison of prediction accuracy.

Similar to the previous section, each model has been replicated 40 times before their mean accuracy can be estimated. Fig. 6 shows the mean performance of each model being considered.

Fig. 6 illustrates that the proposed CSV ensemble model has the highest probability of giving correct classification. The RF, which itself is also an ensemble model, has the second best performance and is slightly better than the SV model. The MV ensemble has a significantly lower accuracy than the top three models, but is still better than the SVM and LR model. The LR model is the model with the lowest prediction accuracy, especially as D increases from 5 to 14 days.

The proposed CSV ensemble model only slightly improves the prediction accuracy compares to the base algorithms, especially when compared to a powerful algorithm such as the RF. However, CSV is a stable method that always among the best algorithms at our extensive test of 40 replications and with D varies from 1 to 14. A way to see this is to see the minimum prediction accuracy in any of the 40 replications, as illustrated in Fig. 7.

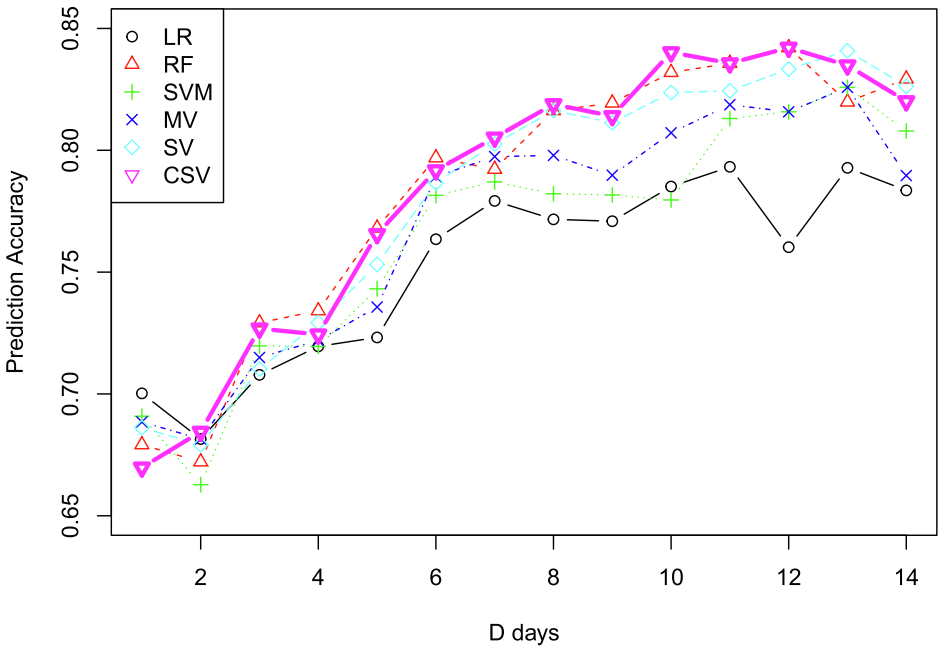


Fig. 7. Comparison of minimum prediction accuracy on any replication.

Fig. 7 shows that CSV is also among the top when it comes to the minimum accuracy in all replications. It shows the high reliability of this method at different predictive experiments.

5. Discussion and implications

This section analyses the processed data from Section 4.1 to provide some insights into passenger booking behaviour.

We first use the Pearson correlation to find out how much each variable contributes to passenger booking. Fig. 8 shows the pairwise correlation between each explanatory variable (in Table 1) and the target variable *Rebooked*. The results for D equals 7 are demonstrated here, but similar results are found for different values of D.

The variables with negative correlation (red color) contributes against the customer booking behaviour, and *vice versa for the green color. Fig. 8 shows that passengers with high TotalCharges and Num_Trips are likely to book another trip within 7 days. This is reasonable because customer who has already made a significant number of trips or have paid more for the service are likely to be more loyal and will be likely to book more trips. In addition, it can be seen that customers may prefer booking via the website. Some hot spots for pick up and drop off can also be discovered using Fig. 8.*

Fig. 8 also shows that waiting for a booking confirmation is undesirable for customers. Customers who have to experience long booking wait time ($booking_wait$) are unlikely to book another trip within 7 days. Customers who book using a phone call and Concession customer are also less likely to rebook. Customer who book late trips (large *SchedPickUpTime*, *SchedDropOffTime*, *ActualPickUpTime* and *ActualDropOffTime*) are also less likely to rebook. This is because these late trips are less recurrent compared to trips during the day.

Along that line of thinking, we then look at the distribution of each important variable versus the value of the Target variable *Rebooked*. Fig. 9 shows the distribution of variables *Num_Trips*, *SchedPickUpTime* and *Traveltime* in the two classes of *Rebooked*. The gray shaded in each sub-figure of Fig. 9 shows the density distribution, while the blue dots are actual data points. Each data point is classified into $Rebooked = NO$ and $Rebooked = YES$ according to the value of its target variable *Rebooked*.

Fig. 9(a) shows a clear difference between the two classes of *Rebooked* regarding the distribution of *Num_Trips* . Customers who have made only a few trips are much less likely to rebook within 7 days. Fig. 9(a) reveals an opportunity to ODT providers to target customers who have made only a few trips because they are less loyal to the service.

Fig. 9(b) shows the scheduled pick up time of customers who rebooked and did not rebook within 7 days. Customers who rebooked ($Rebooked = YES$) are more likely to schedule a pick up within peak periods (around 8AM and 5-6PM), while customers who did not rebook ($Rebooked = NO$) are more likely to schedule a trip within off-peak periods. Providers should value customers who take ODT for commuting trips because they are the most loyal to the service.

In its current form, the proposed CSV method provides an adaptive, extendable predictive analytic framework to forecast the future passenger demand for an SODT service. CSV is one of the first algorithms that aims at the prediction of SODT demand, and the first survey-independent work that predicts the passenger demand from individual behaviours. The method is adaptive and extendable, because more predictive algorithms can be added and their relative Class-specific Predictive Accuracy Value can be updated.

The forecasted passenger demand brings key intelligence for operators to provide better SODT services. Customer booking prediction enables operators to strategically locate their vehicles close to passengers that are likely to book a trip. This is beneficial for both UODT and SODT, but is especially important for SODT because of its nature of a subsided, community service in suburban or rural areas, where there are more disadvantaged customers with mobility needs. Operators may target passengers that are not likely to rebook to provide incentives and encourage more usage. A change of behaviour can also be easily observed from these predictions of the likelihood to book. Improved efficiency through booking prediction would minimise booking rejections, enhancing customer service level and customer loyalty.

6. Conclusion

This paper proposes a framework to predict customer booking behaviour for On-Demand Transport (ODT) services, especially those in suburban areas (SODT) where the demand is low, variable and often regarded as unpredictable Gomes et al. (2014). We use the observed data of a SODT service in Northern Beaches, Sydney, Australia. We further leverage the understanding of the data to propose an ensemble method to predict the customer booking behaviour for On-Demand Transport services, called Class-specific Soft Voting (CSV). The comparison with the base classifiers Logistic Regression, Random Forest and Support Vector Machine, as well as two common ensemble methods: weighted majority voting (MV) and soft voting (SV) shows that the proposed CSV ensemble model provides the most accurate prediction. An analysis of customer booking behaviour shows that the number of trips travelled, the total charges and the website booking positively encourage the customer booking in the data, while variables, such as booking wait time, discourage future booking of customers.

Future research direction includes the inclusion of more supervised classification techniques, and investigation of a more adaptive ensemble method for classification.

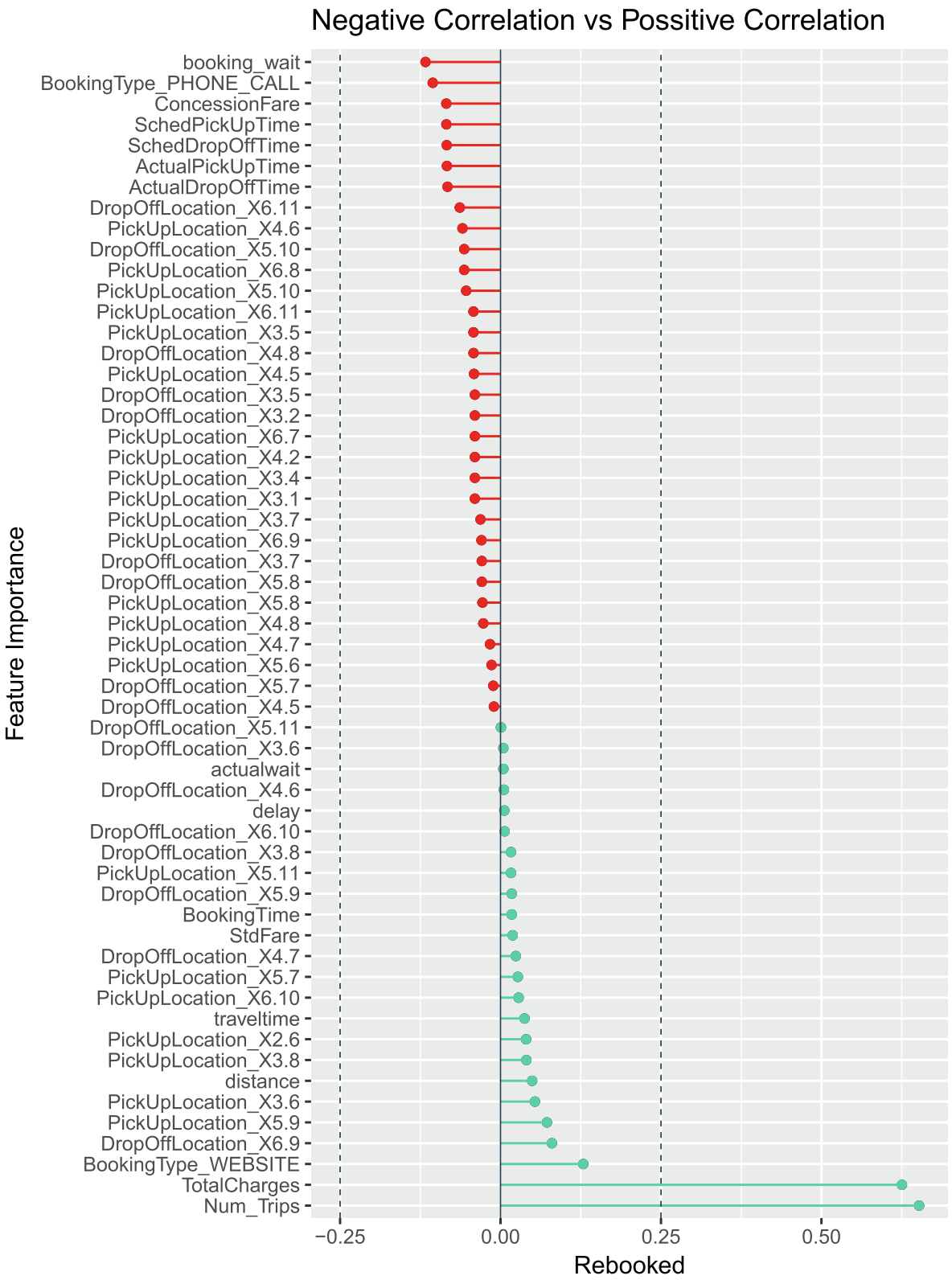
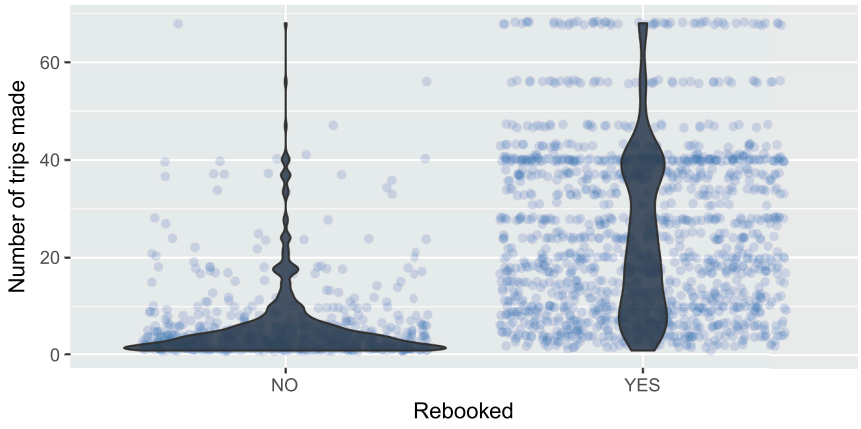
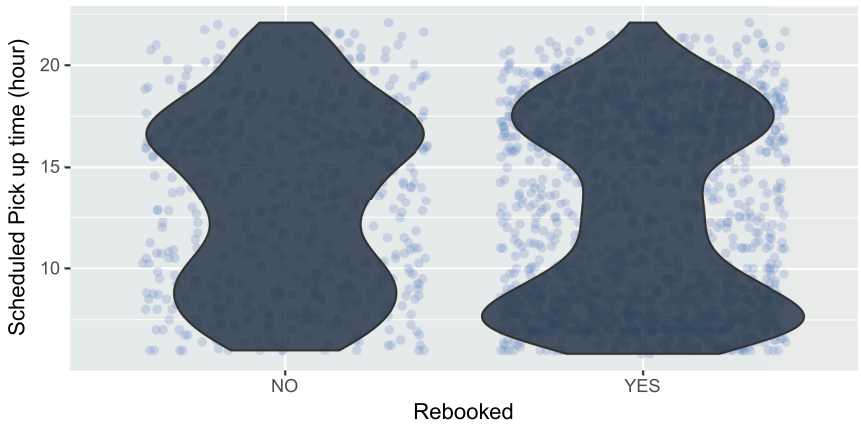


Fig. 8. Feature importance analysis.



(a) Number of trips made before travelling



(b) Scheduled pick up time

Fig. 9. Variable contribution to customer booking behaviour.

Acknowledgement

This research is partially supported by the Strategic Research Funding at CSIRO and the NSW On-Demand Transport Pilot Project. The authors would like to express their gratitude to industry partner Keolis Downer who provided the data; and Dr. Hoang Nguyen at CSIRO who provided insightful and valuable feedback to this study.

References

Al-Ayyash, Z., Abou-Zeid, M., Kaysi, I., 2016. Modeling the demand for a shared-ride taxi service: an application to an organization-based context. Transp. Policy 48, 169–182.
Benjamin, J., Kurauchi, S., Morikawa, T., Polydoropoulou, A., Sasaki, K., Ben-Akiva, M., 1998. Forecasting paratransit ridership using discrete choice models with explicit consideration of availability. Transport. Res. Rec.: J. Transport. Res. Board 1618, 60–65.
Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., Laporte, G., 2007. Static pickup and delivery problems: a classification scheme and survey. Top 15 (1), 1–31.
Daganzo, C.F., 1984. Checkpoint dial-a-ride systems. Transport. Res. Part B: Methodol. 18 (4–5), 315–327.
Errico, F., Crainic, T.G., Malucelli, F., Nonato, M., 2013. A survey on planning semi-flexible transit systems: methodological issues and a unifying framework. Transport. Res. Part C: Emerg. Technol. 36, 324–338.
Gomes, R., de Sousa, J.P., Dias, T.G., 2014. A grasp-based approach for demand responsive transportation. Int. J. Transport. 2 (1), 21–32.
Jain, S., Ronald, N., Thompson, R., Winter, S., 2017. Predicting susceptibility to use demand responsive transport using demographic and trip characteristics of the population. Travel Behav. Soc. 6, 44–56.
Ke, J., Zheng, H., Yang, H., Chen, X.M., 2017. Short-term forecasting of passenger demand under on-demand ride services: a spatio-temporal deep learning approach. Transport. Res. Part C: Emerg. Technol. 85, 591–608.
Li, X., Quadrifoglio, L., 2010. Feeder transit services: choosing between fixed and demand responsive policy. Transport. Res. Part C: Emerg. Technol. 18 (5), 770–780.
Liaw, A., Wiener, M., et al., 2002. Classification and regression by randomforest. R News 2 (3), 18–22.
Meinshausen, N., Schiesser, L., 2007) Quantregforest: quantile regression forests. R package version 0.2-2.
Meyer, D., Wien, F.T., 2001. Support vector machines. R News 1 (3), 23–26.
Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., Damas, L., 2013. Predicting taxi–passenger demand using streaming data. IEEE Trans. Intell. Transp. Syst. 14 (3), 1393–1402.
Nourbakhsh, S.M., Ouyang, Y., 2012. A structured flexible transit system for low demand areas. Transport. Res. Part B: Methodol. 46 (1), 204–216.

Perrone, M.P., Cooper, L.N., 1995. When networks disagree: Ensemble methods for hybrid neural networks. In How We Learn; How We Remember: Toward an Understanding of Brain and Neural Systems: Selected Papers of Leon N Cooper, pages 342–358. World Scientific.
Quadrifoglio, L., Li, X., 2009. A methodology to derive the critical demand density for designing and operating feeder transit services. Transport. Res. Part B: Methodol. 43 (10), 922–935.
Rätsch, G., Onoda, T., Müller, K.-R., 2001. Soft margins for adaboost. Mach. Learn. 42 (3), 287–320.
Steinwart, I., Christmann, A., 2008. Support Vector Machines. Springer Science & Business Media.
TransportNSW (2018). Northern beaches on demand service. <https://www.transport.nsw.gov.au/data-and-research/nsw-future-mobility-prospectus/nsw-future-mobility-case-studies/procurement-as-3>. accessed on 25 may 2019.
Vlahogianni, E.I., Golias, J.C., Karlaftis, M.G., 2004. Short-term traffic forecasting: overview of objectives and methods. Transp. Rev. 24 (5), 533–557.
Wei, Y., Chen, M.-C., 2012. Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks. Transport. Res. Part C: Emerg. Technol. 21 (1), 148–162.
Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., Gong, P., Ye, J., Li, Z., 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In: 2018 AAAI Conference on Artificial Intelligence (AAAI'18).