Transportation Research Part C 109 (2019) 289–304

Contents lists available at ScienceDirect

Transportation Research Part C

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

Evaluation of machine learning methodologies to predict stop delivery times from GPS data

Sebastián Hughes, Sebastián Moreno, Wilfredo F. Yushimito*, Gonzalo Huerta-Cánepa

Faculty of Engineering and Sciences, Universidad Adolfo Ibáñez, Viña del Mar, Chile

ARTICLE INFO

Keywords:
Machine learning
Stop delivery time
Classification
Regression
Hazard duration
GPS

ABSTRACT

In last mile distribution, logistics companies typically arrange and plan their routes based on broad estimates of stop delivery times (i.e., the time spent at each stop to deliver goods to final receivers). If these estimates are not accurate, the level of service is degraded, as the promised time window may not be satisfied.

The purpose of this work is to assess the feasibility of machine learning techniques to predict stop delivery times. This is done by testing a wide range of machine learning techniques (including different types of ensembles) to (1) predict the stop delivery time and (2) to determine whether the total stop delivery time will exceed a predefined time threshold (classification approach). For the assessment, all models are trained using information generated from GPS data collected in Medellín, Colombia and compared to hazard duration models.

The results are threefold. First, the assessment shows that regression-based machine learning approaches are not better than conventional hazard duration models concerning absolute errors of the prediction of the stop delivery times. Second, when the problem is addressed by a classification scheme in which the prediction is aimed to guide whether a stop time will exceed a predefined time, a basic K-nearest-neighbor model outperforms hazard duration models and other machine learning techniques both in accuracy and F_1 score (harmonic mean between precision and recall). Third, the prediction of the exact duration can be improved by combining the classifiers and prediction models or hazard duration models in a two level scheme (first classification then prediction). However, the improvement depends largely on the correct classification (first level).

1. Introduction

Last mile distribution is the final step of the delivery part of the order-to-delivery process. This part of the journey is typically the least efficient and most expensive, representing up to 50 percent of the total logistics cost and totaling 70 billion euros per year worldwide, with a growth rate of 10 percent (Joerss et al., 2016) and with China, Germany, and the USA accounting for more than 40 percent of the market.

An essential aspect of last mile distribution is its tour-centric nature: a trip starts in a warehouse and performs a distribution tour that requires multiple stops for delivery before returning to the warehouse. This characteristic implies that the total distribution time of a tour consists of different elements such as the in-vehicle travel time and the stop delivery time. The latter involves tasks such as unloading and walking times to deliver the goods to end customers. In many cases and depending on several factors (e.g., the size of the delivery), multiple clients can be delivered in a single stop sequentially (implying several unloading and walking trips at that stop), or all clients can be delivered

* Corresponding author.
E-mail addresses: shughes@alumnos.uai.cl (S. Hughes), sebastian.moreno@uai.cl (S. Moreno), wilfredo.yushimito@uai.cl (W.F. Yushimito), gonzalo.huerta@uai.cl (G. Huerta-Cánepa).

https://doi.org/10.1016/j.trc.2019.10.018
Received 10 October 2018; Received in revised form 2 October 2019; Accepted 29 October 2019

in a single walking trip. The process can last minutes or hours and the wrong prediction of the time spent in a delivery can impact the total cost of the tour distribution, especially in countries where labor is expensive relative to the cost of fuel. This is because routing planners usually consider that routes have a fixed time windows at each stop, implicitly this is the same as assuming that the stop delivery duration is less than a pre-defined threshold of time (e.g. a time window of 40 min). However, if the assumption on the stop delivery times falls short, a cascading effect of the duration of the total tour would lead to an increase in the total cost of the tour distribution (e.g., re-scheduling or extending hours-of-service). Thus, a good prediction of the stop delivery times might be critical in route planning in determining whether a sequence of customers can be served without violating regulations on hour-of-service or in reducing the need to re-schedule the deliveries. With a suitable model, the proposed routes can be revised, improving planning and avoiding delays in the delivery.

However, stop delivery times are hard to predict because of their considerable variability (Figliozzi, 2007; Zou et al., 2016). Zou et al. (2016), Schmid et al. (2018) use survival models to predict stop delivery times based on several characteristics of a parked freight vehicle and built environment characteristics (the type of vehicle, the location of legal parking, and the neighborhood). Both studies use limited number of observations, Schmid et al. (2018) used 177 observations from four neighborhoods in Manhattan, while Zou et al. (2016) calibrated their model using 44 observations, including the time of the day as a variable. Other studies in literature are slightly related to stop delivery times, for instance Yang et al. (2014) and Gong et al. (2015) used support vector machines (SVMs) to identify whether a truck has performed a stop from Global Positioning System (GPS); but, the majority of the literature on parking duration focus on parking times for private cars (Caicedo et al., 2012; Vlahogianni et al., 2016).

In this paper, we propose the use of machine learning models to predict the value of the stop delivery time and whether the stop delivery times surpasses a certain time threshold. The resulting contributions are twofold. First, we leverage operational data collected directly from GPS traces provided by a company that provides routing solutions in Latin America to evaluate the machine learning models. This approach also allows us to include operations-related data in the prediction (i.e., demand, number of clients visited in a stop), an important difference with respect to previous works that included only built environment variables (Zou et al., 2016; Schmid et al., 2018). Second, by performing a comprehensive and robust evaluation of machine learning techniques, we test whether such techniques can be used instead of well-known approaches such as hazard duration models. With this evaluation, our final goal is to provide guidance over the use of machine learning models for duration analysis in the context of logistics and transportation, for instance, selecting the appropriate tool, or combination of tools to predict the time or classify the stopping time.

Our procedure starts by processing the available GPS traces, which are enhanced with data obtained from the planning database (e.g., the demand and the actual number of deliveries) and other sources available (e.g., OpenStreetMap). The obtained data are then used to identify the zone of the stops, beginning/end of the stop (stop delivery times), and walking times from the stop to a client's location. Once the variables are obtained, the prediction process is achieved in two ways. In the first approach, we predict the stop delivery time using regression models and compare them with estimates achieved by hazard duration models. In the second approach, we estimate whether the stop delivery time reaches a certain threshold (i.e., the time threshold used by the company that provided the data, in addition to two other times: median and mean times according to the data). In both approaches, we leverage a broad range of primary and sophisticated machine learning techniques, including several ensemble combinations. To the best of our knowledge, this is the first time that such a variety of machine learning techniques has been used and evaluated in this type of problem. All results are compared to those of hazard duration models, which are a family of models constituting the conventional approach for duration analysis. In addition, we propose a hybrid scheme to predict the stop delivery times composed of two levels: a classification model in the first level, followed by either a regression or hazard duration model.

Our results are threefold. First, although it is difficult to predict the stop delivery time due to the (exponential-like) nature of the data, in general, the best hazard duration model (which is log-normal) predicts better results than sophisticated machine learning models in terms of absolute errors between the real and predicted time. However, if mean square errors are used, machine learning models are better. This distinction is an essential aspect because hazard duration models fail in the prediction of extreme values. The second insight is that when the problem is addressed as a classification in which the objective is to determine whether the stop delivery time will be exceeded, most machine learning models (including ensembles) are better than the classification based on the best hazard duration model (i.e., when the output is binarized to evaluate the threshold). However, the machine learning model K-nearest neighbors is significantly better than all other machine learning models. Finally, the application of a two level model could improve largely the prediction of the stop delivery times. Unfortunately, this improvement is largely related to the correct classification in the first level of the model.

The remainder of the paper is organized as follows. Section 2 presents an overview of machine learning techniques for classification problems, covering a wide variety of models that are discussed as succinctly as possible. Section 3 presents an overview of hazard duration models (as they are used as a benchmark). Section 4 presents the data used and a case study. Section 5 describes time duration analysis, as it will be used as a benchmark for the machine learning models. In Section 6, we present the procedure and the results of machine learning models and hazard duration models for comparison, while the hybrid scheme combining classification and regression or hazard duration models is shown in Section 7. Finally, we show our conclusions in Section 8.

2. Machine learning classifiers

Machine learning models have gained popularity due to their ability to capture nonlinear relationships between variables, making them capable of high-quality prediction in several domains. Machine learning models can be broadly categorized as unsupervised and supervised. While unsupervised models find the natural grouping of objects given unlabeled data, supervised models learn the underlying distribution of the data to predict an object class from prelabeled (classified) objects (Aggarwal, 2015). As the final goal of this paper is to predict whether the stop delivery time exceeds a certain time threshold, our interest is in supervised learning models.

Supervised learning models can be separated into two types of problems based on the type of the variable to predict (output

variable). In the case of a categorical output variable, the prediction is typically called **classification**. In contrast, for a numerical output variable, the prediction is typically called **regression**.

The next subsections present a brief description of the models used in this evaluation. This description includes the most relevant machine learning models that vary from simple ones to ensembles (which are a combination of models). Since the latter group is less widely applied, more details are included.

Classification models: There are several machine learning models that vary depending on how the classification is done. Taking into consideration our problem of interest, which is whether the stop delivery time of a truck will exceed a time threshold while stopped to deliver goods, the decision can be reduced to a binary classification problem with diverse types of input variables (categorical and numerical). Then, we briefly describe the most relevant machine learning models of the literature used for binary classification (for a more comprehensive description, see Aggarwal (2015)), including naive Bayes, logistic regression, K-nearest neighbors, support vector machines, classification trees, and neural networks.

Naive Bayes (Langley et al., 1992) is a probabilistic model that assumes independence of the dependent variables given the output variable. **Logistic regression** is another probabilistic model that fits a logistic function to predict the output variable. **K-nearest neighbors (KNN)** (Bhavsar and Ganatra, 2012) is a lazy learner model that classifies a data point based on the distance and class of the KNN.

Support vector machines (SVMs) (Trustorff et al., 2011) create a binary separation through a hyperplane in N_K dimensions (where N_K can be greater than the dimension of the data). Similarly, **classification trees** (Quinlan, 1986) use one-dimensional planes to separate the input data space into multiple N_d-dimensional hyperrectangles (where N_d is the dimension of the data), assigning all data points in the same hyperrectangle a specific class. Finally, **neural networks** (Kruppa et al., 2013; Barboza et al., 2017) are widely considered a black box, where layers with multiple neurons are interconnected among them to model the output variable.

Regression models: We describe the most relevant regression machine learning models of the literature (lasso regression, ridge regression, and elastic net). These models are extensions of linear regression but seek to overcome some of its common limitations such as overfitting (by means of penalization or regularization) or multicollinearity. **Lasso regression** (Tibshirani, 1996) is a linear regression with a L_1 penalization of the parameters to reduce overfitting. Similarly, **ridge regression** (Tibshirani, 1996) is a linear regression with an L_2 penalization of the parameters. **Elastic net** (Zou and Hastie, 2005) is a generalization of lasso and ridge regression combining both penalizations. In addition, for the evaluation described later in the paper, we included neural networks as an additional regression model (which is largely considered both as a classification and regression model). Finally, it is also possible to apply a classification model for regression, such as classification trees. However, in most cases, the performance is even worse than that of linear regression models, so we consider only an ensemble approach of classification trees (XGboost).

Note that in addition to predicting the numeric value of the output variable, regression models can be used to address a classification problem (e.g., whether the time threshold is exceeded based on their predicted time). For this purpose, first, regression can be used to perform the prediction; then, the predicted value can be transformed into a response that can be classified. The specific case of interest in this work is explained later in Section 6.3.

2.1. Ensemble models

Ensemble models are techniques that combine (similar or different) multiple models called base learners. An ensemble's goal is to make a system robust and stable, incorporating the predictions of all learners. Since each algorithm has its own prediction, structure, and parameters, ensembles generate a greater global understanding of the problem and data. It is through this variety that a robust final classification can be made, improving the precision and reducing the bias. Possible combinations between algorithms vary from simple averages to dynamic algorithms that learn from the errors of a previous iteration. One of the advantages of ensembles is that they tend to improve accuracy significantly. In contrast, their disadvantages are that their interpretation of real-life situations is not clear and that their computation time is high. Next, we present brief explanations of each of the ensembles that will be used in this research.

- **Average ensemble** uses the outputs of multiple models of the same or different types (logistic regression, classification tree, or KNN, among others) and calculates the average probability values obtained by every single model. This type of ensemble suppresses the variability of single models, reducing the variance error and leading to better prediction capacity.
- **Stacked generalizer** is an ensemble that captures the outputs of a group of base classification models of different types and uses them as input in a "superior" model to obtain the final prediction (Jonsson et al., 2016). This superior model typically involves logistic regression and takes as input the probability values of the class label.
- **Bootstrap aggregation (bagging)** is an ensemble of independent algorithms typically derived from a single type of classifier. The variability of the models is obtained through the sample of a determined number of different training datasets from the original data. Then, each model learns over these different data samples. Once all models and predictions are generated, the simple majority or the average of the outputs is used to produce the final prediction, obtaining the most efficient results (Breiman, 1996; Barboza et al., 2017). This method is used to reduce the variance and to control the over-training of a class within a model.
- **Boosting** is an ensemble of a single type of classifier. It is a sequential technique, where the first model is trained with the complete dataset and where each subsequent model is constructed, taking into account the residuals of the last trained model, thus giving greater weight to those observations that are misclassified (Begley et al., 1996). The goal is to create weak models in each iteration, which may not be favorable for all data but just for some part of it. In this way, each model improves the performance of the ensemble incrementally. One of the most important implementations based on classification trees is **XGboost** (Chen and Guestrin, 2016).

3. Hazard duration models

Hazard-based duration models are based on the concept of conditional probability of termination of an event given that its duration has lasted some specific time. These models are used to predict, in a simple, efficient, and effective manner, the time duration of events (Wang et al., 2017). Hensher and Mannering (1994) provides an overview of the application of such models. In general, these models have been limited to duration of events or activities that affect traffic or the demand of transportation such as the time spent shopping (Bhat, 1996), daily activities (Kharoufeh and Goulias, 2002), the time spent at home between trips (Hamed and Mannering, 1993), road pavement duration (Svenson, 2014), traffic delays (Paselk and Mannering, 1994), duration of accidents/incidents (Jovanis and Chang, 1989; Tavassoli et al., 2013; Li, 2015), evacuation events (Hasan et al., 2013), and recently the duration of truck parking delivery segments (Zou et al., 2016; Schmid et al., 2018).

In the case of stop delivery duration, the probability that a vehicle (truck) stop duration T is greater than t can be defined by the survival function $S(t) = 1 - F(t) = P[T > t]$, while the hazard function $\lambda(t) = \frac{f(t)}{S(t)}$ gives the departure from the parking spot. Using these two definitions, the survival function can be expressed solely in terms of the hazard function, resulting in

$$S(t) = \exp\left\{-\int_0^t \lambda(t)dt\right\} \tag{1}$$

This survival function accommodates different underlying distributions in the models as the distribution of the hazard function defines them. The most common distributions are the parametric models that use a priori distribution models such as exponential, Weibull, and log-normal, among others. When the distribution is not known or the number of samples is small, semiparametric and nonparametric models are more appropriate, as the baseline hazard function can take any form (not necessarily a distribution). The Cox model (Cox, 1972) is the most well-known semiparametric model that entails the specification of the log hazard function as a linear model. That is, the log hazard function is expressed as a linear function of its covariates $\mathbf{X}$:

$$\log\lambda(t) = \alpha + \mathbf{X}'\beta \tag{2}$$

The hazard function can be directly obtained, resulting in a model based on the exponential distribution where the α is replaced by the baseline hazard function $\log\lambda_0(t)$, expressed as

$$\lambda(t) = \lambda_0(t)\exp(\mathbf{X}'\beta) \tag{3}$$

where the baseline hazard $\lambda_0(t)$ can take any functional form and the vector of coefficients β is estimated by Cox regression. The estimated β can be interpreted similarly to the case of multiple logistic regression.

One advantage of both types of hazard duration model is the inclusion of covariates that can affect the stop delivery times. For instance, Zou et al. (2016) used the Cox model to determine the factors that affect the duration of truck parking delivery times in New York City (NYC). They used 44 observations and identified relevant covariates such as the location (neighborhood) and the type of cargo. Schmid et al. (2018) recently performed a similar analysis, also in NYC, using a Weibull model with 177 observations, including the type of vehicle and some other variables related to illegal parking areas and the distance to legal parking areas. None of them included operations-related data as we include in this paper.

4. Data description

We focused our analysis on the information available from a planning and tracking software in South America, called RoutePro (http://www.citymovil.cl/route-pro/). The software plans routes and schedules the deliveries of goods through a vehicle routing meta-heuristic, providing a sequence of customers which are associated to a given truck. The data provided for our analysis consists of daily route plans for a fleet of 15 medium (1 ton) trucks of a dairy products company that operates in Medellín (Colombia). The software stores data of the dispatching, information on the vehicles (trucks), delivery volumes, truck characteristics, and customer information. The software also uses a mobile application that displays the information related to the delivery and stores GPS traces. Since the mobile application is installed in a device kept in the truck, it continuously send information back to the server, even when the driver is out of the truck making a delivery.

The available data included for this study contains the equivalent of 1.87 GB of GPS traces collected between February 18th, 2016, and April 13th, 2017, with information from 2760 clients (Fig. 1). To provide an idea of the type of data collected, Appendix A shows the GPS information and the extracted information from the database. For each delivery, the mobile application generates a GPS trace composed of multiple GPS points, each enhanced with metadata consisting of an ID, timestamp, geographical coordinates, instantaneous speed, direction, and status of the tour (tour initiated, in progress, and if the delivery worker is on foot, among others).

4.1. Data pre-processing

As in previous studies (Pfoser and Jensen, 1999; Yanhong and Xiaofa, 2013; Gingerich et al., 2016), we found measurement errors and inaccuracies in the GPS traces. As expressed by Brakatsoulas et al. (2007), the uncertainty of an object's movement is related to the frequency with which position samples are taken (the sampling rate), and includes errors based on the loss of connection to satellites. To address these issues, we implemented a two-step filtering algorithm to eliminate inconsistencies. The general idea of the procedure is to identify fixed or valid points in a sequence of GPS pulses, and discard those that are not logical, possible, or can be considered as a single point due to lack of movement.

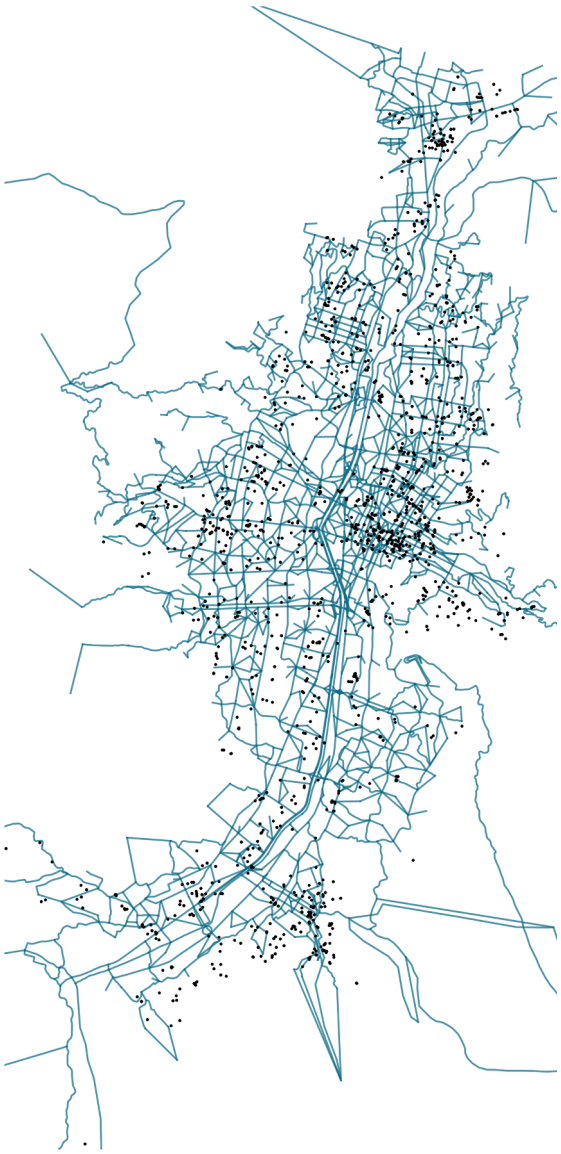

Fig. 1. Locations of clients identified in the study area (2760) between February 18th, 2016, and April 13th, 2017.

We used a multi-layer filter to remove invalid or unnecessary data points from our collection. This multi-layer filter consists of a location, speed, and stop detection filter. The location-based filter removes locations that are invalid due to their geographical position (positions that are outside Medellín's radius, which was fixed at 19 kilometres from downtown). The speed-based searches for consecutive points where the speed between them is higher than 100 km per hour (kph). When a case is detected, we analyze their previous and subsequent points to determine its feasibility (maximum speed in Colombia is 60 kph and 80 kph on urban public road and rural areas respectively). In case that the change in the speed is improbable, such as an increment from 30 kph to 100 kph in a couple of seconds, the point that triggered this peak is removed. Finally, stop-detection filter clusters multiple location points into a single point based on two criteria: distance and time. Specifically, the data points must be closer than 100 meters and their timestamps differences must be less than minutes.

The last step in the pre-processing flow consists of path rectification. Path rectification was a two-step process: first, we rectify a point to the closest street using Open Source Routing Machine (OSRM) (Luxen and Vetter, 2011). Then, we simplify the representation of the defined paths for each route, removing points that did not provide relevant information (same street and not close to a corner).

4.2. Data processing

After the pre-processing process, a clustering algorithm groups the activities by stopping point, i.e., a location where the truck stopped to make deliveries to one or multiple clients. We obtain the stopping points in two steps. The first step determines places where the vehicle (truck) stopped for more than five minutes, based on the previous clustered points from data pre-processing. The

second step takes those places and marks potential stops using client addresses obtained from the company's database: stopping points that are walking distance from a client are marked as delivery stops. This step removes stall points due to traffic jams.

For the second step, the maximum radius is set to 500 meters (approximately 0.3 miles), and assuming a maximum walking speed of 4.5 kph, the maximum time between deliveries to customers inside a cluster is fixed to 15 min (13.5 min roundtrip + waiting times) (Munuzuri et al., 2012). Then, the algorithm generates a circle at each stop point, covering customer locations.

As a result of the process, from the original 2760 client locations, a total of 344 clusters and their corresponding stop locations were identified. Fig. 2 shows a sample clusters of clients assigned to a stop for a given tour. Note that clusters are built per route and are not generic, meaning that on one day, a cluster can contain client A and B, whereas on another day, client A can be in a cluster by its own.

We characterize each cluster by its stop point, the number of clients, and their demand, and we enhance this baseline with information related to the period of the day, walking distance to deliver the goods in the cluster, distance from the previous stop, type of street, and traffic. The latter two values were obtained using the API from OpenStreetMap (Weber and Haklay, 2008).

4.3. Summary of features of GPS data

After the filtering process, we identified a total of 202 tours with valid information. From these tours, we extracted mobility patterns of urban freight distribution to distinguish variables, which are included in the survival and machine learning analysis.

Table 1 presents a summary of the resulting information. Each tour has an average number of clients of 27.98 with a high standard deviation (16.01 clients per tour). The average number of stops per tour is 7.38, for a total number of 1487 stops from the 202 tours analyzed. The average number of clients visited per stop is 4.64 but with a high standard deviation (3.88 number of clients visited per stop). The average time spent at a stop is 37.4 min with a high standard deviation in the stop delivery times (30.6 min), which is consistent with the results of other studies (e.g., Figliozzi (2007)). Regarding the demand, the average number of pallets delivered on one tour is 114.22 with an average of 21.09 pallets delivered per client. Note that the demand could be obtained only in terms of pallets due to the large weight variability, preventing us from converting the demand into weight units, which would have been the ideal case.

In terms of the duration of the stops, Fig. 3 shows the distribution of stop delivery times. Note that the distribution is right-skewed, which is consistent with the results of previous studies (e.g., Zou et al. (2016)). Approximately 70% of the observed delivery trucks have an on-street parking duration of less than 40 min, with a few outliers with very long parking durations. The next section presents the results of hazard-based duration models to complement the analysis of the delivery time stop duration.

5. Duration analysis

Three hazard duration parametric models and the Cox semiparametric model are tested to analyze the duration of the stop delivery times. This analysis is made for comparative purposes as these models are used as benchmark for the machine learning models tested. Also, as mentioned in Section 3, the advantage of hazard duration models is the inclusion of covariates that can affect the stop delivery times (Table 2 shows the list of evaluated covariates).

Covariates from Table 2 were selected because they might influence stop delivery times. Some of these covariates are directly obtained from the GPS data, such as: Stop Number, Walking Distance, and Distance from Previous Stop. We also included other covariates indirectly related to the GPS, including: Clients, Demand, and Time of Day (3 time slots). Finally, other covariates were extracted from OpenStreetMaps: Type of Street, Traffic, and Location Area.

Notice that we are including time and built environment variables as in Zou et al. (2016) and Schmid et al. (2018). Beside these types of variables, we are also including operational variables, as well as, other variables that were obtained from the GPS data. Our reasoning is that operation related variables might affect the stopping time. For instance, if multiple trips are required to be done from the truck to the clients location, the stopping time might increase and this can be due to the size of the cargo (demand) or the number of clients visited. Similarly, the order of the stop might be relevant, as in some cases, in order to complete the tour within certain time, last stops might be shorter.

To identify the relevant variables for each model, we leverage a stepwise procedure, selecting variables according to a level of significance of 0.10. The results of the variables selected in each model are shown in Table 3, including goodness of fit measures such as the p-value for the whole model, the AIC value (which measures the entropy of the information), and the log likelihood of the model.

As shown in Table 3, the Demand, Stop Number, Type of Street, and Time of Day are significant for all models. Recall, that Demand is a continuous variable, Stop Number is discrete, and Type of Street and Time of Day are categorical. For Type of Street, "primary road" was included as the base, while for Time of Day, the morning period (8am to 12 pm) was chosen as base. Only the log-normal model includes an additional variable, which refers to Location Area with Medellín chosen as the base. However, while all models are significant, the best model is the log-normal one as its log-likelihood is less negative than all other models, and it also has the lowest AIC criterion.

As in Zou et al. (2016) and Schmid et al. (2018), Time of Day is significant for all models (Table 3), while Location Area is significant only for the log-normal model. Results also show the importance of operational variables in stop delivery times, as Demand and Stop Number are selected as important variables. Note that when a stop includes multiple clients, the total demand of the stop is more relevant. Note also that the hazard ratio is slightly over 1.00, which indicates that the duration is reduced slightly with increasing demand; possibly because a single stop is typically made to attend large clients. The other operational variable is Stop Number, whose effect is similar to Demand, implying that deliveries to later clients might be hurried to return within schedule (to the warehouse). This implication might also be related to Time of Day, where the stop delivery times are reduced for the evening period. Variables that increase the duration according to the hazard rate include location outside the main district (Medellín) and if the truck is parked in a secondary road, which is typically farther from commercial areas.

Fig. 2. Sample of stops corresponding to February 20, 2017. (Clusters are shown as big circles, their assigned client's in solid small circles and the x marks the stop.).

Table 1
Data description of the 202 tours.

	Average	Standard deviation	Source
Number of clients per tour	27.98	16.01	GPS
Demand (#pallets/tour)	114.22	60.53	GPS/Database
Number of stops per tour	7.38	4.64	GPS
Demand per stop (#pallets/stop)	21.09	20.52	GPS/Database
Number of clients per stop	4.64	3.88	GPS
Delivery tour duration (hour)	14.27	4.11	GPS
Time per stop (minutes)	37.4	30.6	GPS

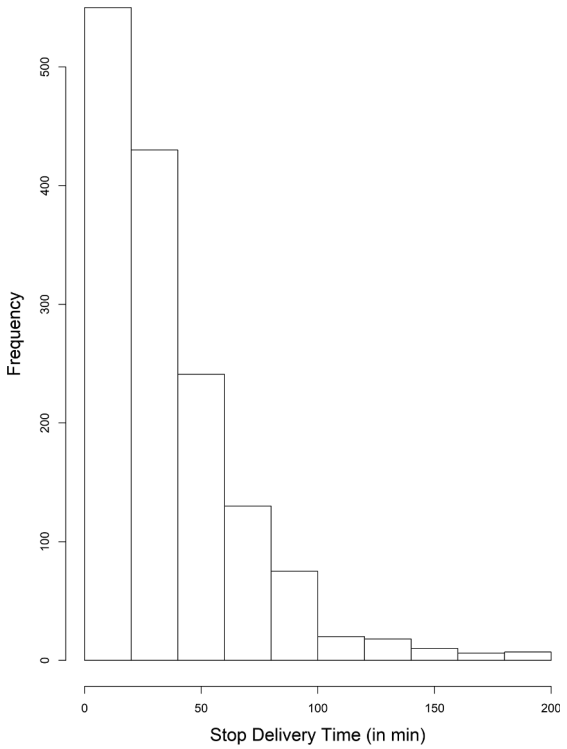

Fig. 3. Histogram of the stop delivery times for the total data set.

Table 2
Covariates included in the hazard duration models.

Variable	Data type (Source)	Value
Clients	Numeric/Discrete (GPS)	Number of clients per stop
Demand	Numeric/Discrete (GPS)	Total demand in pallets per stop
Type of Street	Binary (OpenStreetMaps)	0 if a primary road; 1 if another type of road
Traffic	Categorical (OpenStreetMaps)	Three levels: Normal, Low, High
Location Area	Categorical (OpenStreetMaps)	Four levels: Medellín, Itagui, Envigado, Other
Stop Number	Numeric/Discrete (GPS)	Stop number (order) in the tour sequence
Time of Day	Categorical (GPS)	Three levels: Morning (8am-12 pm), Afternoon (12 pm-4 pm), Evening (4 pm-8 pm)
Walking Distance	Continuous (GPS)	Total walking distance in the stop (in km)
Distance from Previous Stop	Continuous (GPS)	Distance from previous stop (in km)

6. Machine learning models' results

6.1. Variable selection and experiment procedure

We performed a feature and parameter selection process for each machine learning model. Specifically, we applied a sequential forward generation search method (a greedy approach) with the wrapper evaluation method (the evaluation is done by the learned model over the test data).

Table 3
Results and performance measures of hazard duration models for significant variables at a level of significance of 10% or less.

	Exponential			Weibull			log-normal			Cox		
Attribute	Coef	Z	Hazard Ratio	Coef	Z	Hazard Ratio	Coef	Z	Hazard Ratio	Coef	Z	Hazard Ratio
(Intercept)	3.96	15.77	52.62	4.04	21.86	56.99	3.77	19.39	43.55	0	0	0
Location Area												
Medellín (base)	–	–	–	–	–		–	–	–	–	–	–
Itagui	–	–	–	–	–		–0.16	–2.47	0.85	–	–	–
Envigado	–	–	–	–	–		–0.03	–0.55	0.97	–	–	–
Other	–	–	–	–	–		–0.16	–1.79	0.86	–	–	–
Type of Street												
Primary Road (base)	–	–	–	–	–		–	–	–	–	–	–
Secondary Road	−0.42	−1.68	0.66	−0.41	−2.21	0.66	−0.44	−2.3	0.64	0.53	2.1	1.7
Time of Day												
Morning (8am-12) (base)	–	–	–	–	–		–	–	–	–	–	–
Afternoon (12–4 pm)	−0.04	−0.42	0.96	−0.04	−0.42	0.96	−0.01	−0.12	0.99	0.03	0.38	1.04
Evening (4 pm–8 pm)	0.45	2.09	1.57	0.47	1.97	1.59	0.32	1.6	1.38	−0.27	−1.22	0.77
Tour related variables												
Stop Number	0.02	1.22	1.02	0.02	1.25	1.02	0.02	1.56	1.02	−0.03	-1.91	0.97
Demand (10^3)	0.01	2.84	1.01	0.01	3.98	1.01	0.01	3.15	1.01	−0.01	-3.64	0.99
Log-likelihood	−4830			−4750			−4680			−6190		
AIC	9660			9510			9380			12400		

The method is as follows. Let $\mathcal{P} = \{X_1, \cdots, X_d\}$ be the set of possible variables for the model, $\mathcal{S} = \{\}$ be the set of selected variables for the model, and $\mathcal{M}_{best}$ be the best model using the variables from $\mathcal{S}$ (the best model initially uses none of the variables). At each iteration, $N_{\mathcal{P}} = |\mathcal{P}|$ new models are learned by moving each of the variables from $\mathcal{P}$ to $\mathcal{S}$, training the model using the variables from $\mathcal{S}$ and returning the selected variable to $\mathcal{P}$. Let $\mathcal{M}_{P_i}$ be the best model from the $N_{\mathcal{P}}$ possible models, where moving the variable i generates the best model. If the performance of $\mathcal{M}_{P_i}$ is lower than or equal to that of $\mathcal{M}_{best}$, then $\mathcal{M}_{best}$ is selected as the trained element. In contrast, if the performance of $\mathcal{M}_{P_i}$ is better than that of $\mathcal{M}_{best}$, the variable i is moved permanently from $\mathcal{P}$ to $\mathcal{S}$, $\mathcal{M}_{best}$ is updated to $\mathcal{M}_{P_i}$, and a new iteration begins. The iterations are repeated until no variables are available at $\mathcal{P}$ or the performance is not longer increased.

To evaluate the models in both experiments, we applied the same procedure, stratified K-fold cross validation with $K = 10$. This sampling method is applied to overcome the underestimation of the error standard deviation obtained by the random sampling method (Dietterich, 1998). From the 1,489 data points, corresponding to the whole data set, we separate the data in ten folds with 149 randomly selected data points (one fold has only 148 data points), where the percentage of the data points belonging to each of the classes is kept similar among all data folds. Then, the mean ± one standard deviation error is based on K iterations. At each iteration, the i-th fold data are left as test data, and the other $K - 1$ folds are combined generating the training data, where each model is trained.

In all experiments, we apply the F-test and t-test hypothesis test with $\alpha = 1\%$. The F-test determines if the differences among the estimated variance for the models are statistically significant. On the other hand, the t-test, with unknown variance and similar/different variance (according to the result of the F-test), is applied to determine if the differences among mean errors are statistically significant.

6.2. Regression models

For the regression models described in Section 3, we evaluated whether the models can predict the actual stop delivery time of the trucks. We tested **lasso regression** (Tibshirani, 1996), **ridge regression** (Hoerl and Kennard, 1970), **elastic net** (Zou and Hastie, 2005), NeuralNetR, and XGBoost, as briefly reviewed in Section 2.

To evaluate the performance of the trained models, we evaluated the mean squared error (Eq. (4)) and mean absolute percentage error (Eq. (5)) over the test data.

$$MSE = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} (Y_i - \widehat{Y}_i)^2 \tag{4}$$

$$MAPE = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \left| \frac{Y_i - \widehat{Y}_i}{Y_i} \right| \tag{5}$$

The MSE is the average squared deviation of the predicted values ($\widehat{Y}_i$) with respect to the actual values (Y_i). Given the same data, the best model is expected to have the lowest MSE. However, the MSE is affected by extreme values (an extreme stop delivery time);

for this reason, we also calculate the MAPE.

The MAPE is the mean absolute percentage error of the predicted values ($\widehat{Y_i}$) with respect to the actual values (Y_i). A MAPE of x implies that, on average, the prediction has an error of $x\%$ with respect to the real value. This measure avoids the problem of extreme values given the standardization over Y_i.

6.2.1. Results

Table 4 presents the results for the best hazard duration (log-normal) and the regression models used in this paper. As can be observed, according to the MSE, the best model is NeuralNetR. However, all models show a high MSE error and standard deviation, so none of them is better from a statistical point of view (t-test applied). In contrast, using MAPE, the best model is the log-normal hazard duration model, which is significantly better than the rest of the models (t-test applied). This difference can be explained by extreme values. Specifically, the log-normal model does not capture the extreme stop delivery times, increasing their MSE error. In contrast, the other models could be giving more importance to the higher stop delivery times rather than the lower stop delivery times.

As indicated by the errors, the prediction of the exact stop delivery time is hard mainly due to the exponential-like distribution of the values. Thus, the selection of the best model would depend on which values one is trying to predict (e.g., for high values of time, the log-normal model is not recommended). The type of data also explains the low performance obtained by all models (recall that the exponential distribution has mean and variance equal to λ). The next subsection presents the evaluation of the classification model, including NeuralNetR as the best regression model (even though it is not significantly better from a statistically point of view) and the log-normal hazard duration model.

6.3. Machine learning classifiers

All classification and ensemble models described in Section 3 are tested by converting the predicted stop delivery time into a binary classification problem. The binary decision is whether the stop delivery time is higher than a specific threshold. For the case of the NeuralNetR and the log-normal hazard model, their numeric prediction of the stop delivery time is binarized to compare it against the threshold.

Three time thresholds are used, representing the median and mean of the stop delivery time (27 min and 37.4 min, respectively) obtained from the data in addition to the actual time threshold set in the software (40 min). To evaluate the performance of the models, we measure the accuracy (Eq. (6)) and the F_1 score (Eq. (7)) over the test data.

$$Accuracy = \frac{TP + TN}{N_{test}} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$F_1\ score = \frac{2 * \mathrm{TP}}{2 * \mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{7}$$

Accuracy is computed as the percentage of correctly classified points, which is equivalent to the number of true positives (TP: number of positive points classified correctly, i.e., the time is lower than the threshold) plus the number of true negatives (TN: number of negative points classified correctly, i.e., the time is higher than the threshold) over the total number of test points (N_{test}). The accuracy varies between 0 and 1, where a high accuracy implies that the model can predict most of the data points correctly. Unfortunately, the accuracy behaves improperly when a class is biased. To avoid this problem, we also evaluate the F_1 score.

The F_1 score is the harmonic average of the precision and recall, which avoids the TN values. As shown in Eq. (7), the F_1 score is calculated based on TP, the number of false positives (FP: number of data points incorrectly classified as positive), and the number of false negatives (FN: number of data points incorrectly classified as negative). The F_1 score varies between 0 and 1, where a high F_1 score implies that the model can classify the positive class and (most importantly) generates a low number of FN and FP.

For the ensembles, boosting and bagging use a combination of classification trees. The stacked generalization ensemble model (SG) leverages four classification models (classification tree, logistic regression, naive Bayes, and KNN), and their outputs are combined using a logistic regression model. The average ensemble uses the same classification model as the SG, but the output is averaged to determine the classification. In addition to the selected ensembles, we also tested a different type of ensemble presented by He et al. (2014) to predict clicks on Facebook ads. This ensemble is a two-level ensemble that combines the boosted classification tree with logistic regression. First, the classification tree is boosted, and for each tree produced, the ending leaf for each register is saved as a binary vector (the length of the vector is equal to the total sum of leaves of all the tree generated by the boosting process). Then, this new dataset of size $N_{train} \times N_{\ell}$, where N_{train} is

Table 4
Performance measures MSE and MAPE of the hazard duration and regression models.

Method	MSE	MAPE
NeuralNetR	918.80 ± 197.25	105.00 ± 12.40
log-normal	1013.00 ± 234.20	75.80 ± 9.00
XGBoost	932.00 ± 179.07	111.60 ± 12.80
Ridge	928.20 ± 204.68	105.70 ± 12.10
Lasso	928.10 ± 202.25	105.40 ± 12.10
ElasticNet	928.30 ± 202.72	105.50 ± 12.20

the number of registers in the original dataset, and N_ℓ is the total number of leaves in the groups of trees created by the boosting process, is learned by a logistic regression to then produce a final classification. For the particular case of this ensemble, we used ten trees in a boosting scheme that are used later in a logistic regression model, making the final classification.

6.3.1. Results

Table 5 presents the results for each classifier, where the positive class correspond to a time lower than the specified threshold. In addition to the classification, we include the log-normal and NeuralNetR as the baseline models for hazard duration and regression. Columns present the mean ± one standard deviation of the accuracy and the F_1 score for three different thresholds, a 40-min threshold, the median time (27.0 min) and the mean time (37.4 min).

Both the accuracy and F_1 scores increase with higher thresholds, reaching a maximum F_1 score of approximately 0.825 when the 40-min threshold is used. However, for the lowest threshold value (27 min), the classification models obtain an F_1 score of approximately 0.707.

For the lowest threshold (27 min), KNN is significantly better than all other models except stacked generalization and average ensemble (t-test applied). Note that the last two models are not significantly better than other basic classification models. Similarly, all models show better performance than the random model except NeuralNetR, which obtains a similar performance. Similar results are obtained for the F_1 score, as all models (including NeuralNetR) are better than the random model with KNN as the best model. However, KNN is significantly better only against models based on trees (classification tree, boosting tree, bagging tree and the model by He et al.). Considering that KNN is superior to the classification trees and that classification trees did not generate sufficient variation among the models, it is not surprising that the ensembles behave practically the same as the classification trees.

For the 37.4-min threshold, the results show that KNN is the best model. In both cases, KNN exhibits better performance than all other models considering the accuracy and F_1 score (t-test applied). Furthermore, NeuralNetR and LogNormal are the worst models (t-test applied) except when they are compared against the classification tree and their ensembles. Finally, the 40-min threshold leads to similar conclusions as KNN as the best model in the accuracy and F_1 score, but it is not significantly better than logistic regression, neural net, stacked generalization, and average ensemble. Similar to the 27-min case, and given the performance of logistic regression and neural net, it is expected that the ensembles based on these methods also improve their behavior.

It is quite surprising that none of the ensembles overcomes the KNN model. This result can be explained by the exponential-type distribution form of the data, which leads to similar results among all basic models. Consequently, each basic model of an ensemble extracts the same information from the data, reducing the variability among them and leading to the same type of results. In contrast, KNN can adapt to the data and model the distribution.

Note that even though the prediction of the time gives us more information about the delivery, when the problem is to classify whether the time of the stop will exceed a predefined time, it is considerably better to use a classification model than to use the time predicted by either a regression model or a hazard duration model. For those particular classification cases, KNN appears to be the best model. This finding is important from a practitioner's perspective, as KNN is a very simple model relative to the complex ensemble models or neural networks. Moreover, considering that this model does not have a training process and uses all the available data, as new data is added in the system, the model can be continuously upgraded. The only limitation of KNN might be the slow estimation process based on a large number of data points, but from a planning perspective, this process can be implemented day to day.

7. Integrating machine learning classifiers with hazard duration/regression models

Given the considerable high performance of the classification models, we explored the possibility of improving the predictions of

Table 5
Mean and standard deviation of accuracy and F_1 score for classification models, log-normal hazard duration, and NeuralNetR regression model.

	Time Threshold Selection					
	27 min (median)		37.4 min (mean)		40 min	
Method	Acc	F_1 Score	Acc	F_1 Score	Acc	F_1 Score
Random	0.496 ± 0.030	0.495 ± 0.034	0.485 ± 0.026	0.540 ± 0.035	0.502 ± 0.029	0.571 ± 0.035
Logistic Regression	0.543 ± 0.024	0.686 ± 0.010	0.644 ± 0.018	0.779 ± 0.006	0.669 ± 0.013	0.799 ± 0.007
Classification Tree	0.525 ± 0.018	0.680 ± 0.006	0.632 ± 0.004	0.775 ± 0.003	0.659 ± 0.004	0.795 ± 0.003
Naive Bayes	0.546 ± 0.026	0.686 ± 0.008	0.638 ± 0.008	0.777 ± 0.004	0.662 ± 0.005	0.796 ± 0.003
SVM	0.539 ± 0.029	0.685 ± 0.007	0.632 ± 0.004	0.775 ± 0.003	0.659 ± 0.004	0.795 ± 0.003
KNN	0.605 ± 0.044	0.707 ± 0.021	0.744 ± 0.061	0.815 ± 0.028	0.739 ± 0.063	0.825 ± 0.026
Neural Net	0.541 ± 0.025	0.687 ± 0.008	0.644 ± 0.010	0.78 ± 0.005	0.673 ± 0.016	0.800 ± 0.006
Staked Generalization	0.557 ± 0.033	0.692 ± 0.012	0.646 ± 0.013	0.780 ± 0.005	0.667 ± 0.011	0.797 ± 0.004
Average Ensemble	0.568 ± 0.062	0.697 ± 0.023	0.659 ± 0.028	0.783 ± 0.010	0.679 ± 0.024	0.803 ± 0.009
Boosting Tree	0.537 ± 0.029	0.684 ± 0.010	0.644 ± 0.022	0.779 ± 0.010	0.662 ± 0.008	0.796 ± 0.004
Bagging Tree	0.526 ± 0.016	0.680 ± 0.007	0.640 ± 0.010	0.778 ± 0.005	0.659 ± 0.004	0.795 ± 0.003
He et al. (2014)	0.540 ± 0.028	0.683 ± 0.007	0.645 ± 0.014	0.780 ± 0.007	0.661 ± 0.006	0.796 ± 0.004
NeuralNetR	0.510 ± 0.006	0.676 ± 0.001	0.550 ± 0.029	0.636 ± 0.026	0.623 ± 0.037	0.731 ± 0.028
lognormal	0.523 ± 0.031	0.449 ± 0.040	0.628 ± 0.006	0.772 ± 0.004	0.653 ± 0.012	0.790 ± 0.008

both, hazard duration and regression models (see Table 4), by developing a new model consisting of two levels. In this model, given an observation of a stopping time duration, the classification model (first level) determines if the stop delivery time will be higher or lower than a specific threshold. Then, a prediction model (second level), trained only with the data corresponding to the classified class, predicts the value of the stop delivery time. Considering that KNN had the highest performance according to the previous section, we selected KNN as the first level classifier. For the second level model, we compare the six model from Table 4 (NeuralNetR, XGBoost, Ridge, Lasso, ElasticNet, and the log-normal hazard duration model). In order to fairly compare the new results with the results from Table 4, we applied the same training and testing process used on previous sections including the same folds of the 10-fold cross validation process.

For the evaluation of the new prediction results, we tested the mean time threshold (37.4 min). This threshold was selected for three main reasons. First, KNN obtained the highest accuracy among all models for this specific threshold. Second, the mean value balances the data points belonging to positive and negative classes (points correctly classified and incorrectly classified respectively), reducing the possibility of over-fitting some of the prediction models. Finally, the value of the mean is close to the 40 min threshold. The results are shown in Table 6, the left side shows the new results while the right side replicates previous results from Table 4 to facilitate the comparison. As it can be observed, the MSE is higher on the two level model. However, MAPE is considerable lower, being almost half of the previous results. To further analyze the reasons of these inconsistent results, we proceed to decompose the results based on the classification realized by the first level model. That is, we looked to the predictions of the observations that were either correctly and incorrectly classified by KNN in the first level.

Table 7 shows the decomposition of the MSE and MAPE for the two level model based on KNN and prediction models. Left side of the table shows MSE for the 37.4 threshold, while right side shows the MAPE. Each error is separated into four columns ($C > 37.4$, $C < 37.4$, $I > 37.4$, $I < 37.4$) based on whether they were correctly classified (C) or incorrectly classified (I) in the first level. For example, $I > 37.4$ represents all stop delivery times that were incorrectly classified (i.e. the total stop times of these points were less than 37.4 but were classified as more than 37.4). As it can be observed, the MSE and MAPE for correctly classified observations are considerably lower than previous results, showing a considerably better performance. Moreover, KNN classified correctly all points less than 37.4, explaining the value of NA in this cell. However, the incorrectly classified points for $I > 37.4$ boosted all errors, specially the MSE for all models. This shows the importance of a correct classification, as the classification in the first level influences largely the predictions of the second level: the error produced of the incorrectly classified points affects considerably the final predictions.

8. Conclusions

The stop delivery time is a critical component for planning routes for delivery distribution. However, the prediction of such times is hard due to many factors that increase the variance in the data such as the time of day, location, and amount of cargo, among others.

This work presented an evaluation of the state-of-the-art techniques in machine learning to predict stop delivery times in two ways. First, we predict the duration of the stop. Second, we estimate whether the stop time will exceed a target time threshold. In both cases, we use information generated from GPS traces (gathered by a routing software developer from Medellín, Colombia) and planning data and data from OpenStreetMaps.

For the time prediction with machine learning models, we showed that considering the MAPE, machine learning regression models are not better than the best hazard duration model fitted to the data (the log-normal model). This result can be explained by the fact that regression models give more importance to extreme values than the log-normal model. In contrast, when the MSE is considered, machine learning regression models provide better results than the log-normal model.

In the case of the classification process (whether the stop delivery time surpasses a specific threshold), the machine learning classification models perform better than the binarization of the time predicted by the regression and hazard duration models. Moreover, K-nearest neighbors (KNN) behaves significantly better than the other models tested. The adaptability of KNN to the data could explain this result. The results have some important practical applications, as KNN is a straightforward model that requires minimal resources; as long as the data are kept up to date, the predictions can be fully automated. The automation can be also included into tour construction procedures in order to, for instance, accomplish hours of regulation or fulfill time windows.

We also tested a hybrid scheme of two levels (first level) combining the KNN classifier with regression or hazard duration models

Table 6
Performance measures MSE and MAPE of two level model based on KNN and prediction models (left) and prediction models (right) for the mean value of the stopping time (37.4 min).

	Time Threshold Selection			
	Two Level Model		Prediction Models	
Method	MSE	MAPE	MSE	MAPE
NeuralNetR	2256.20 ± 352.77	95.20 ± 0.30	918.80 ± 197.25	105.00 ± 12.40
log-normal	1298.51 ± 289.73	53.80 ± 4.40	1013.00 ± 234.20	75.80 ± 9.00
XGBoost	1268.46 ± 282.44	56.70 ± 4.90	932.00 ± 179.07	111.60 ± 12.80
Ridge	1236.13 ± 284.48	56.50 ± 5.30	928.20 ± 204.68	105.70 ± 12.10
Lasso	1236.11 ± 284.27	56.50 ± 5.20	928.10 ± 202.25	105.40 ± 12.10
Elastic Net	1236.26 ± 284.46	56.50 ± 5.20	928.30 ± 202.72	105.50 ±12.20

Table 7
Decomposition of the MSE and MAPE for the two level model, based on their classification and threshold (Note: C/I means correctly and incorrectly classified, > / < means if the classification was higher or lower than the 37.4 threshold, and NA means Not Applicable.)

	MSE for Mean Threshold (37.4 min)				MAPE for Mean Threshold (37.4 min)			
Method	C > 37.4	C < 37.4	I > 37.4	I < 37.4	C > 37.4	C < 37.4	I > 37.4	I < 37.4
NeuralNetR	3976.79 ± 1435.30	402.74± 29.78	5466.70 ± 950.90	NA	98.30 ± 0.30	93.40 ± 0.50	98.30 ± 0.10	NA
log-normal	124.76 ± 115.34	70.70 ± 4.53	3501.70 ± 799.25	NA	15.40 ± 4.80	45.00 ± 6.80	70.50 ± 1.40	NA
XGBoost	156.79 ± 122.01	76.76 ± 4.97	3406.00 ± 778.15	NA	16.90 ± 4.80	50.40 ± 7.80	69.00 ± 1.90	NA
Ridge	166.15 ± 84.53	67.96 ± 3.92	3332.08 ± 784.97	NA	18.80 ± 9.10	50.90 ± 8.30	67.50 ± 1.60	NA
Lasso	158.79 ± 88.86	67.84 ± 4.05	3333.18 ± 784.21	NA	18.50 ± 9.40	50.90 ± 8.30	67.60 ± 1.60	NA
Elastic Net	158.02 ± 91.65	67.98 ± 3.85	3331.73 ± 783.92	NA	18.40 ± 9.60	50.90 ± 8.20	67.50 ± 1.60	NA

(second level) to improve stop delivery time prediction. Results are promising as error measures are reduced significantly when the first level classifies correctly whether or not they exceed a time threshold, but the error produced in the incorrectly classified cases could affect considerably the final predictions.

Finally, this evaluation can guide practitioners in the use of machine learning models in problems related to duration analysis in logistics (in which the time needs to be either predicted or classified) and other related problems such as parking times, accidents, or road pavement duration.

Acknowledgements

The authors would like to thank the Corporación de Fomento de la Producción (CORFO) of the Government of Chile 16VIP-71524 for their financial support and Citymovil for providing the data. The authors also appreciate the insightful comments and suggestions of four reviewers who helped us to improve the quality of the paper.

Appendix A. GPS and other data used

The data obtained from the clients extracted from the data base and the GPS data collected are shown in Tables A.1–A.7.

Table A.1
Customer information stored in the database. The empty fields, such as the arrival time, contain information only if the customer imposes a restriction.

customer_id	Name	external_id	service_duration	location_id	arrival_time	demand	sequence
1	MERCADOS KENEDY	400622		2		0.0	0
2	SALSAMENTARIA LOS AGUDELOS	400926		3		0.0	0

Table A.2
Selected order stored in the database.

order_id	Date	Demand	external_id	customer_id	scheduled_customer_id
1535	2017-01-17 00:00:00	1.0	13994663	2083	1445
1536	2017-01-17 00:00:00	1.0	13994747	1996	1446

Table A.3
Selected data related to delivery locations from the database. The different ways users input data can be observed when comparing both addresses.

location_id	Address	Route	Locality	administrative_area_level_1	Latitude	Longitude
1	Cr 49 # 17 1, medellin	Calle 17	Medellín	Antioquia	6.2187444	−75.5767293
2	CL 89A # 75 B - 33 Kennedy, Doce De Octubre Medellín, Antioquia 050040, Colombia				6.28679627	−75.58207113

Table A.4
Scheduling information. This table is filled with information from the scheduling software application (RoutePro) and complemented with data obtained from the application. This information is available to drivers through the mobile application. Other information not shown in this snapshot includes the full destination address (including the geo-coordinates), distance and time from the previous location and next standstill, and zone.

scheduled_customer_id	external_id	Name	Sequence	ready_time	due_time	arrival_time	service_duration	real_arrival_time	total_demand	location_id
1445	2041191	GRANERO LA PLACITA	1	21600	36000	22428	480		1.0	2084
1446	299155	GRANERO LOS SUAREZ	2	21600	50400	23056	480		1.0	1997

Table A.5
Example of a route stored in the database.

route_id	pdf_location	total_customers_in_route	total_kilograms	total_meters	schedule_id	vehicle_id	end_of_trip	trip	start
70		23	23.0	25833	3	36	61615	1	21600
71		41	46.0	40758	3	37	61439	1	21600

Table A.6
Vehicle information extracted from the database. Only trucks are shown (vehicle_type = 1).

vehicle_id	Capacity	external_id	driver	ready_time	due_time	depot_id	active	load_time	max_trips
1	339	SKH858	HECTOR CASTAÑO	25200	57600	1	1	1800	1
2	216	SLP102	DARIO HERRERA	25200	57600	1	1	1800	1

Table A.7
Database data describing the location of a device associated with a truck.

vehicle_position_id	vehicle_id	latitude	longitude	date	heading	created	accuracy
1	1	−33.4115238	−70.5438308	2017-02-17 21:07:45	0	2017-02-18 00:06:55	26.422
2	1	−33.4115496	−70.5440318	2017-02-17 21:08:19	0	2017-02-18 00:07:30	21.038

References

Aggarwal, C.C. 2015. Data Mining: The Textbook. Springer. doi:https://doi.org/10.1007/978-3-319-14142-8.

Barboza, F., Kimura, H., Altman, E., 2017. Machine learning models and bankruptcy prediction. Expert Syst. Appl. 83, 405–417. https://doi.org/10.1016/j.eswa.2017.04.006.

Begley, J., Ming, J., Watts, S., 1996. Bankruptcy classification errors in the 1980s: an empirical analysis of altman's and ohlson's models. Rev. Acc. Stud. 1, 267–284. https://doi.org/10.1007/BF00570833.

Bhat, C.R., 1996. A hazard-based duration model of shopping activity with nonparametric baseline specification and nonparametric control for unobserved heterogeneity. Transp. Res. Part B: Methodological 30 (3), 189–207. https://doi.org/10.1016/0191-2615(95)00029-1.

Bhavsar, H., Ganatra, A., 2012. A comparative study of training algorithms for supervised machine learning. Int. J. Soft Comput. Eng. (IJSCE) 2, 74–81. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.492.6088 .

Brakatsoulas, S., Pfoser, D., Wenk, C. 2007. Creating a data mart for floating car data. Technical Report COOP-CT-2006-032823, TRACK&TRADE consortium. https://www.researchgate.net/publication/242126088_Creating_a_Data_Mart_for_Floating_Car_Data. [Online; accessed 14-January-2019].

Breiman, L., 1996. Bagging predictors. Mach. Learn. 24 (2), 123–140. https://doi.org/10.1007/BF00058655.

Caicedo, F., Blazquez, C., Miranda, P., 2012. Prediction of parking space availability in real time. Expert Syst. Appl. 39 (8), 7281–7290. https://doi.org/10.1016/j.eswa.2012.01.091.

Chen, T., Guestrin, C. 2016. Xgboost: A scalable tree boosting system. CoRR, abs/1603.02754. doi:https://doi.org/10.1145/2939672.2939785.

Cox, D.R., 1972. Regression models and life-tables. J. Roy. Stat. Soc. Ser. B (Methodological) 34 (2), 187–220.. http://www.jstor.org/stable/2985181 .

Dietterich, T.G., 1998. Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput. 10 (7), 1895–1923. https://doi.org/10.1162/089976698300017197.

Figliozzi, M.A., 2007. Analysis of the efficiency of urban commercial vehicle tours: Data collection, methodology, and policy implications. Transp. Res. Part B: Methodological 41 (9), 1014–1032. https://doi.org/10.1016/j.trb.2007.04.006.

Gingerich, K., Maoh, H., Anderson, W., 2016. Classifying the purpose of stopped truck events: An application of entropy to gps data. Transp. Res. Part C: Emerging Technol. 64, 17–27. https://doi.org/10.1016/j.trc.2016.01.002.

Gong, L., Sato, H., Yamamoto, T., Miwa, T., Morikawa, T., 2015. Identification of activity stop locations in gps trajectories by density-based clustering method combined with support vector machines. J. Mod. Transp. 23 (3), 202–213. https://doi.org/10.1007/s40534-015-0079-x.

Hamed, M.M., Mannering, F.L., 1993. Modeling travelers' postwork activity involvement: toward a new methodology. Transp. Sci. 27 (4), 381–394. https://doi.org/10.1287/trsc.27.4.381.

Hasan, S., Mesa-Arango, R., Ukkusuri, S., 2013. A random-parameter hazard-based model to understand household evacuation timing behavior. Transp. Res. Part C: Emerging Technol. 27, 108–116. https://doi.org/10.1016/j.trc.2011.06.005.

He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., Shi, Y., Atallah, A., Herbrich, R., Bowers, S., Candela, J.Q. 2014. Practical lessons from predicting clicks on ads at facebook. In: Proceedings of the Eighth International Workshop on Data Mining for Online Advertising, ADKDD'14. ACM, pp. 5:1–5:9. doi:https://doi.org/10.1145/2648584.2648589.

Hensher, D.A., Mannering, F.L., 1994. Hazard-based duration models and their application to transport analysis. Transp. Rev. 14 (1), 63–82. https://doi.org/10.1080/01441649408716866.

Hoerl, A.E., Kennard, R.W., 1970. Ridge regression: biased estimation for nonorthogonal problems. Technometrics 12 (1), 55–67. https://doi.org/10.1080/00401706.1970.10488634.

Joerss, M., Schroder, J., Neuhaus, F., Klink, C., Mann, F. 2016. Mckinsey & company parcel delivery: The future of last mile. Technical report, McKinsey & Company. http://www.mckinsey.com/industries/travel-transport-and-logistics/our-insights/how-customer-demands-are-reshaping-last-mile-delivery. Online; accessed 5-July-2018.

Jonsson, L., Borg, M., Broman, D., Sandahl, K., Eldh, S., Runeson, P., 2016. Automated bug assignment: ensemble-based machine learning in large scale industrial contexts. Empirical Softw. Eng. 21 (4), 1533–1578. https://doi.org/10.1007/s10664-015-9401-9.

Jovanis, P.P., Chang, H., 1989. Disaggregate model of highway accident occurrence using survival theory. Acc. Anal. Prevent. 21 (5), 445–458. https://doi.org/10.1016/0001-4575(89)90005-5.

Kharoufeh, J.P., Goulias, K.G., 2002. Nonparametric identification of daily activity durations using kernel density estimators. Transp. Res. Part B: Methodological 36 (1), 59–82. https://doi.org/10.1016/S0191-2615(00)00038-2.

Kruppa, J., Schwarz, A., Arminger, G., Ziegler, A., 2013. Consumer credit risk: individual probability estimates using machine learning. Expert Syst. Appl. 40 (13), 5125–5131. https://doi.org/10.1016/j.eswa.2013.03.019.

Langley, P., Iba, W., Thompson, K., 1992. An analysis of bayesian classifiers. In: Proceedings of the Tenth National Conference on Artificial Intelligence. AAAI, pp. 223–228. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.135.7718.

Li, R., 2015. Traffic incident duration analysis and prediction models based on the survival analysis approach. IET Intel. Transport Syst. 9 (4), 351–358. https://doi.org/10.1049/iet-its.2014.0036.

Luxen, Dennis, Vetter, Christian, 2011. Real-time routing with openstreetmap data. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, pp. 513–516.

Munuzuri, J., Cortes, P., Grosso, R., Guadix, J., 2012. Selecting the location of minihubs for freight delivery in congested downtown areas. J. Comput. Sci. 3 (4), 228–237. https://doi.org/10.1016/j.jocs.2011.12.002.

Paselk, T.A., Mannering, F.L., 1994. Use of duration models for predicting vehicular delay at a us/canadian border crossing. Transportation 21 (3), 249–270. https://doi.org/10.1007/BF01099213.

D. Pfoser and C.S. Jensen. Capturing the uncertainty of moving-object representations. In Ralf Hartmut Güting, Dimitris Papadias, and Fred Lochovsky, editors, Proceedings of the 6th International Symposium on Advances in Spatial Databases, pages 111–131. Springer, 1999. doi:10.1007/3-540-48482-5_9.

Quinlan, J.R., 1986. Induction of decision trees. Mach. Learn. 1 (1), 81–106. https://doi.org/10.1007/BF00116251.

Schmid, J., Wang, X.C., Conway, A., 2018. Commercial vehicle parking duration in new york city and its implications for planning. Transp. Res. Part A: Policy Pract. 116, 580–590. https://doi.org/10.1016/j.tra.2018.06.018.

Svenson, K., 2014. Estimated lifetimes of road pavements in Sweden using time-to-event analysis. J. Transp. Eng. 140 (11), 040140561–040140568. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000712.

Tavassoli, A., Ferreira, L., Washington, S., Charles, P., 2013. Hazard based models for freeway traffic incident duration. Acc. Anal. Prevent. 52, 171–181. https://doi.org/10.1016/j.aap.2012.12.037.

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. J. Roy. Stat. Soc. Ser. B (Methodological) 58 (1), 267–288.

Trustorff, J.H., Konrad, P.M., Leker, J., 2011. Credit risk prediction using support vector machines. Rev. Quant. Financ. Acc. 36 (4), 565–581. https://doi.org/10.1007/s11156-010-0190-3.

Vlahogianni, E.I., Kepaptsoglou, K., Tsetsos, V., Karlaftis, M.G., 2016. A real-time parking prediction system for smart cities. J. Intell. Transp. Syst. 20 (2), 192–204. https://doi.org/10.1080/15472450.2015.1037955.

Wang, P., Li, Y., Reddy, C.K. 2017. Machine learning for survival analysis: a survey. CoRR, abs/1708.04649, 2017. http://arxiv.org/abs/1708.04649.

Weber, P., Haklay, M., 2008. Openstreetmap: user-generated street maps. IEEE Pervasive Comput. 7, 12–18. https://doi.org/10.1109/MPRV.2008.80.

Yang, X., Sun, Z., Ban, X., Holguin-Veras, J., 2014. Urban freight delivery stop identification with gps data. Transp. Res. Rec.: J. Transp. Res. Board 2411, 55–61. https://doi.org/10.3141/2411-07.

Yanhong, F., Xiaofa, S., 2013. Research on freight truck operation characteristics based on gps data. Procedia – Soc. Behav. Sci. 96, 2320–2331. https://doi.org/10.1016/j.sbspro.2013.08.261.

Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. J. Roy. Stat. Soc. Ser. B 67, 301–320. https://doi.org/10.1111/j.1467-9868.2005.00527.x.

Zou, W., Wang, X.C., Conway, A., Chen, Q., 2016. Empirical analysis of delivery vehicle on-street parking pattern in manhattan area. J. Urban Plann. Dev. 142 (2), 04015017. https://doi.org/10.1061/(ASCE)UP.1943-5444.0000300.