Discussion Paper

Machine-learning based transport mode prediction in a smartphone-based travel and mobility survey

Quinty Boer
Yvonne Gootzen
Jonas Klingwort
Daniëlle Remmerswaal
Peter Lugtig

February 2026

Abstract

Reliable statistics on travel behavior are important for national infrastructure planning, transport policy-making, and understanding mobility patterns. Recent advances in smartphone-based travel surveys enable passive data collection via smartphone Global Positioning System (GPS) sensors. These smart surveys offer an alternative to traditional travel diary surveys, which are typically impacted by a high response burden. This study examines the application of supervised machine learning models to automatically identify transport modes from GPS measurements obtained through a travel app developed by Statistics Netherlands. We compare Random Forest and Extreme Gradient Boosting classification models trained on GPS-based features in combination with contextual location-based features from OpenStreetMap, as well as temporal features derived from time-related information and previous travel behavior. The Extreme Gradient Boosting model trained on the complete feature set achieved the highest accuracy (0.91) and macro-averaged F1-score (0.84), while also achieving the best accuracy (0.84) when evaluated on external validation data. Although these results suggest that fully automated transport mode classification in the travel app may not be feasible, a semi-automated approach with targeted prompts could be used to balance transport mode classification accuracy and response burden.

Keywords: transport mode detection, smart survey, Random Forest, Extreme Gradient Boosting, GPS

Contents

1 Introduction

Reliable statistics on travel behavior are important for national infrastructure planning, transport policy-making, and understanding mobility patterns. Traditionally, data on travel behavior are collected by national statistical institutes through diary surveys (e.g., McCool et al. 2021). In these surveys, respondents are asked to report their daily activities over a specified period. The requested information typically includes the duration of a trip, the trip purpose (e.g., work, shopping), and the transport mode used (e.g., car, walking). While diary-based travel surveys have long been the standard for collecting travel behavior data, they pose considerable challenges. As diary-based travel surveys rely on the respondent's ability to remember their travel activities, they are susceptible to recall bias, underreporting, and response-over-time bias (McCool et al. 2021; Prelipcean et al. 2018). For example, short trips are typically under-reported (Stopher and Greaves 2007; Nahmias-Biran et al. 2018), and the duration of public transport trips is regularly overestimated compared to car trips (Bohte and Maat 2009). Furthermore, diary surveys are time-intensive and impose a high burden on respondents, which is associated with an increased risk of non-response and dropout (Kitamura and Bovy 1987; Hu et al. 2020; Wang et al. 2024).

To address these challenges, researchers and statistical institutes are increasingly exploring smart survey approaches. In smart surveys, data collection is facilitated by utilizing sensors available on smartphones. In the context of travel surveys, the Global Positioning System (GPS) sensors in smartphones can be used to collect information on individual travel behavior (e.g., Yang et al. 2018). The idea behind this approach is that the collected GPS measurements are used to define segments, where each segment is a cluster of consecutive GPS points representing either a trip or a stop (Safi et al. 2016). Based on these defined segments, travel statistics such as the trip distance, duration, and speed can be computed (Xiao et al. 2012; Calabrese et al. 2013; Nahmias-Biran et al. 2018; Li et al. 2021). If done well, respondents no longer need to manually record their trips, which may reduce respondent burden. Additionally, GPS data may provide higher data granularity and more detailed information on travel behavior that is difficult to obtain from a traditional diary survey. In this study, we assume that all trip segments are single-modal trips.

In 2018 and 2022/2023, Statistics Netherlands performed large-scale field tests of a smart survey application (McCool et al. 2021; Schouten et al. 2024). A travel app was developed that passively collected GPS data from participants. There were three reasons for in-house development: transparency of technology and algorithms, flexibility in training and retraining machine learning methods, and flexibility in monitoring and evaluation of in-app respondent paradata (Klingwort et al. 2026). The GPS measurements were algorithmically grouped into trips and stops (Klingwort et al. 2025c). Participants were then prompted to annotate these trips and stops with the corresponding transport mode or stop purpose (see Remmerswaal et al. (2025) for the role of the respondent). This semi-automated approach no longer requires respondents to provide details about their travel times and durations, unlike traditional diary surveys. Only when the stop/track segmentation algorithm failed did respondents still have to add all trip information manually. However, even when the app accurately defined the stop/track segments, the semi-automated approach still relied on respondent input for annotating the collected data.

A potential next step for the travel app is to infer the transport mode from GPS data automatically. If done well, this may eliminate the need for respondents to annotate the data and enable entirely passive data collection. Automatic transport mode detection is a relevant topic in international transportation research (Sadeghian et al. 2021; Yang et al. 2024), and

several studies have been conducted in the Netherlands using data collected via the travel app from Statistics Netherlands. Smeets et al. (2019) applied a Random Forest classifier to predict transport mode based on data collected in 2018, and Fourie et al. (2025) developed a rule-based algorithm using data collected in 2022 and 2023. Both studies combined GPS data collected in the app with location context data from OpenStreetMap (OSM), such as proximity to public transport stations, to classify transport modes.

The studies used different approaches to handling the nested structure of the travel data, in which respondents participating in the survey make multiple trips and may collect GPS data over multiple days. Fourie et al. (2025) accounted for the nested data structure during the training and testing of the algorithm by grouping trips from the same respondent into either the train or test data. However, they did not include information on earlier travel behavior as potential input for their model. Contrastingly, Smeets et al. (2019) considered all trips as independent events, thereby not explicitly accounting for the nested structure of the data. However, they included an indicator in their model indicating whether a transport mode had been used previously, thereby acknowledging potential recurring patterns among respondents. Ultimately, it is not yet clear whether the longitudinal structure of data collected in travel surveys can be used to improve the predictive performance of transport mode classifiers.

In contrast to Smeets et al. 2019; Fourie et al. 2025, we study machine learning algorithms on the 2022/2023 dataset. We compare the performance of two commonly used supervised machine learning algorithms in the field of transportation research: Random Forest and Extreme Gradient Boosting (XGBoost) (Xiao et al. 2017; Bedogni et al. 2016; Bjerre-Nielsen et al. 2020; Hasan et al. 2022; Sadeghian et al. 2022). We build on previous work by combining GPS data with context-location information from OSM. Additionally, we aim to capture potential temporal patterns in the data by adding features based on prior travel behavior. Moreover, we analyze how respondent prompts and respondent interaction could be applied in practice. With this, we aim to answer the research question: how well can machine learning algorithms predict the mode of transport in a smartphone-based travel survey, considering different algorithms and feature sets?

The following section includes a brief overview of standard features and models in transportation mode detection literature. After providing information on the data and methods, we compare the predictive performance of two machine learning algorithms with different feature sets: GPS-based, OSM-based, and temporal. We show that including temporal features provides a slight increase in prediction performance with both algorithms. However, the added value differs per model and is considerably lower than that of location-based information from OSM.

2 Research Background

Given the increased prevalence of smartphones in society, many researchers are exploring the potential of using smartphone sensors for data collection as an addition or alternative to traditional diary surveys (Zhou et al. 2022). Smartphone sensors such as the GPS, Global System for Mobile Communication (GSM), and Wi-Fi Positioning System (WPS) can be used to collect real-time location data and discover individual mobility patterns (Xiao et al. 2012; Calabrese et al. 2013; Lu et al. 2019; Li et al. 2021). Data collected from these sensors may then be used to infer mobility patterns, including the transport mode.

This section provides a brief overview of methods commonly used for transport mode detection in the transportation research literature. The data collected by Statistics Netherlands in the field

test of their travel app have been labeled by the respondents with the relevant transport mode. Consequently, we focus particularly on supervised machine learning approaches that require labeled data. Although semi-supervised and unsupervised machine learning approaches are becoming increasingly popular in the literature because they can be applied when labeled data is limited or unavailable (Zhang et al. 2021), supervised approaches are preferred when labeled data is available (Yang et al. 2024).

We also briefly discuss rule-based algorithms, which have previously been successfully applied to travel app data collected by Statistics Netherlands, see for example Klingwort et al. (2025a) and Fourie et al. (2025). A summary of the predictive performance of smart features is given by Klingwort et al. (2025b).

2.1 Supervised machine learning algorithms

Supervised machine learning algorithms have been a popular choice for transport mode detection in the transportation research literature (Sadeghian et al. 2022; Yang et al. 2024). Machine learning algorithms are well-suited for high-dimensional data and can capture complex, non-linear relationships in data (e.g., Boulesteix and Schmid 2014). Some of the algorithms used for transport mode classification include Support Vector Machines (Bolbol et al. 2012), Naive Bayes (Nour et al. 2016), and Bayesian Networks (Xiao et al. 2015). Decision tree-based ensemble models, such as gradient-boosted trees and Random Forests, are also commonly used for transport mode detection (e.g., Li et al. 2021; Lu et al. 2019; Shafique and Hato 2016; Liu et al. 2022). As tree-based ensemble models have been found to regularly outperform other supervised machine learning models (Stenneth et al. 2011; Li et al. 2021; Sadeghian et al. 2021; Liu et al. 2022), these will be discussed more in depth.

2.1.1 Random Forest and Extreme Gradient Boosting

Random Forest and Extreme Gradient Boosting (XGBoost), a specific implementation of gradient-boosted trees, are both ensemble models based on decision trees. However, these models differ in how the trees are constructed and combined. Random Forests typically create a large number of independent decision trees using bootstrapped samples of the data, also known as bagging. The predictions are then aggregated through majority voting (Breiman 2001). In contrast, XGBoost creates trees sequentially, where each new tree is trained to correct the residual errors of the previous ones, also known as boosting (Chen and Guestrin 2016). This approach may allow for learning more complex relationships and patterns in the data, potentially achieving higher accuracy. However, it also results in higher model complexity and greater hyperparameter sensitivity, which can lead to overfitting, especially when hyperparameters are not properly tuned (e.g., Bentéjac et al. 2021). Both algorithms have the advantage of being non-parametric and are capable of handling strongly correlated features (Pappu and Pardalos 2014; Zhao et al. 2020).

Multiple studies have found that the Random Forest algorithm performs well in classifying transport modes. Li et al. (2021) compared a Random Forest classifier against a Decision Tree, AdaBoost, XGBoost, LightGBM, and Artificial Neural Network and found it provided the highest accuracy rate on a test dataset with five transport modes with 91.1% accuracy. Furthermore, Stenneth et al. (2011) found that a Random Forest algorithm outperformed Naive Bayes, Decision Tree, and Bayesian Network with an accuracy of 93% in the classification of 6 transport modes. Additionally, Smeets et al. (2019) used a Random Forest algorithm on earlier travel app data collected by Statistics Netherlands, achieving 64% accuracy across eight transport modes and 77% when collapsing modes into four overarching categories (walk, bike, motorized, and

public transport). In contrast, XGBoost has been used far less frequently for transport mode detection. Still, it has been used successfully in the literature, with Liu et al. (2022) reporting that both LightGBM and XGBoost slightly outperformed Random Forest with a macro-averaged F1-score of 0.92 when classifying six transport modes. Considering these reported results, both Random Forest and XGBoost models will be used and compared in this study.

2.2 Rule-based Algorithms

Although machine learning algorithms have been most prevalent in the literature on transport mode detection (Sadeghian et al. 2021), rule-based algorithms have also been successfully used to predict transport mode based on GPS data (Xiao et al. 2019; Fourie et al. 2025). As previous transport mode detection approaches based on data from Statistics Netherlands used rule-based algorithms, they are also briefly discussed here.

Rule-based algorithms have the advantage of high interpretability, especially in comparison to machine learning algorithms. However, these approaches often require researchers to manually adjust decision rules, which may be time-intensive and require expert knowledge (Sadeghian et al. 2021). On travel app data collected by Statistics Netherlands in 2022 and 2023, which is also used in this study, Fourie et al. (2025) implemented a rule-based algorithm with an accuracy of 85% and balanced accuracy of 70% on a test set comprising seven transport modes. Besides GPS-based features, their study considered a large number of OSM-based features, including public transport routes, public transport stations, traffic indicators, and parking amenities.

2.3 Temporal Features

Travel surveys may have data collection windows ranging from a single day to multiple days or multiple weeks or months (Axhausen et al. 2007). In the field test of the travel app developed by Statistics Netherlands, respondents were asked to collect and annotate their travel behavior data for either one or seven days. Consequently, the collected data can be considered longitudinal. As a result, it is possible to explore whether recurring trips or other temporal patterns in the data improve the accuracy of automatic transport mode detection.

However, few studies that apply traditional supervised machine learning algorithms for automatic transport mode detection have incorporated the longitudinal nature of the data into their models. Although there are studies explicitly making use of the longitudinal structure of data, they typically make use of deep learning models like Long Short-Term Memory (LSTM) (Liu and Lee 2017; Yu 2021; Qin et al. 2019; Jiang et al. 2023). These models typically support automatic feature extraction, including temporal dependencies, unlike most traditional machine learning models for transport mode detection, which rely on manual feature extraction. Deep learning models also allow for greater flexibility in the model input. Whereas traditional supervised learning algorithms often require input at the trip segment level, deep learning models allow GPS measurement-level input. Creating features at the trip segment level requires summarizing and aggregating GPS measurements to the trip segment level, resulting in information loss (Yang et al. 2024). However, a major disadvantage of these algorithms is that they require a huge amount of data to be trained successfully. The data available to this study was considered too small to train an LSTM model.

Consequently, we aim to incorporate manually constructed temporal features into a traditional supervised machine learning approach for automatic transport mode detection. We are

interested in whether information from previous trips and stops, and potential daily or weekly patterns, may improve prediction performance, in addition to features based on GPS or OSM data that are primarily used in the transportation research literature.

3 Data

3.1 Travel App Data

From November 2022 to February 2023, Statistics Netherlands collected smartphone GPS data from a sample of the general population as part of a field test of a smartphone travel app designed to measure travel behavior in the Netherlands. The field test used an experimental design, with data collected across three phases. The phases differ in three ways: whether respondents were asked to collect data for 1 or 7 days, whether they could edit their trips and stops in the travel app, and whether they received an additional web-based questionnaire (Schouten et al. 2024). The data collected in the travel app used for this study is based on a total of 501 respondents randomly sampled from the Dutch population.

After respondents were invited to participate in the field test and install the app, the smartphone's GPS sensors were used to collect information on respondents' locations. An algorithm was implemented to segment the data into stationary events ('stops') and travel events ('tracks'). Stops were determined when a respondent stayed within a 100-meter radius for at least 300 seconds (Gootzen et al. 2025; Klingwort et al. 2025c). The radius was adjusted based on geolocation measurement accuracy, with lower accuracy resulting in a larger radius. Accuracy is represented by a radius in meters in which the actual geolocation is expected to lie. As a result, small values indicate more precise measurements.

Based on the algorithm-generated segmented tracks and stops, a travel diary was automatically generated and presented to the respondent. The respondent could label the presented tracks and stops by selecting a transport mode for the tracks (e.g., bus, car, walking) and a stop purpose for the stops (e.g., home, work, shopping) from provided options. For stop purpose predction, we refer to Zahroh et al. (2025). Figure 3.1 provides an overview of the app's user interface, showing the homepage, a generated travel diary, and an annotated travel diary.

3.2 Data Pre-processing

Pre-processing of the travel app data was performed in two stages. The first stage was conducted before the start of this study and consisted of removing low-quality observations. Data from respondents were excluded when they had recorded less than one hour of data or fewer than 2,000 GPS measurements (Schouten et al. 2024; Gootzen et al. 2025). Approximately 25% of respondents and their corresponding data were excluded based on these quality criteria. This ensures the analysis is based on high-quality data, and the potential effects on performance have not yet been studied.

The remaining data is stored in two separate datasets: one containing the annotated events and the other containing the geolocations collected in the travel app. Each event includes a respondent ID, a start time, an end time, an indication of whether the event is a track or a stop, and, if provided by the respondent, the transport mode or stop purpose. The geolocations

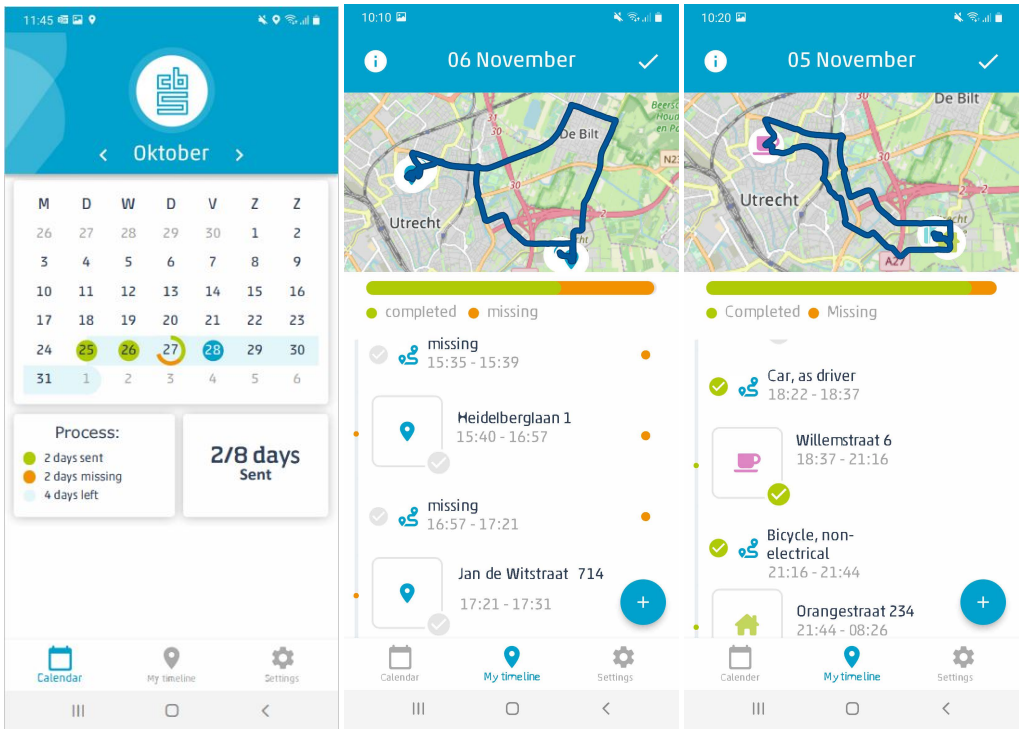

Figure 3.1 User interface of the travel app. The left image shows the app's homepage, with an overview of the days on which data was collected and/or labeled by the respondent. The middle image shows the automatically generated travel diary presented to the respondent. The right image shows the travel diary after the respondent annotated it with transport modes and stop purposes. The data is based on internal testing and does not reflect an actual respondent.

consist of a longitude, a latitude, a timestamp, and a respondent ID. These datasets were made available at the start of this study.

The second stage of data pre-processing includes all the steps performed in this study. In this stage, all duplicate GPS measurements were removed. Duplicate measurements were identified as observations with the same respondent ID and timestamp. Typically, these duplicates occur when more than one sensor collects data simultaneously, or when multiple measurements are made within the same second. When duplicates were found, the GPS measurement with the lowest accuracy value was kept. Afterward, the geolocations were linked to their corresponding events through the respondent ID and the event start and end times.

In addition, transport mode categories such as 'car driver', 'car passenger', and 'delivery van driver' were combined into a single 'car' category, and 'bike' and 'e-bike' were combined into a single 'bike' category. Combining bike and e-bike categories was also done in Smeets et al. (2019), resulting in a higher prediction accuracy. In contrast to their study, however, public transport and motorized modes were not combined in this study. Unlabelled trips and miscellaneous transport modes were excluded from the analysis. The miscellaneous category, where respondents could enter free text, was excluded due to its heterogeneity and limited interpretability. The remaining transport mode labels include bike, bus, car, metro, train, tram, and walking. As the respondents provided transport mode labels in the travel app, these are used as ground truth for transport mode classification.

Lastly, tracks exceeding 10 hours in duration or containing fewer than 10 geolocation measurements were excluded from the analysis. Very long tracks may reflect measurement or processing errors, whereas tracks with few GPS measurements may yield unreliable event-level statistics and model features. Only minimal filter criteria were applied, resulting in some remaining trips with very high trip durations or very few GPS measurements. However, stricter filter criteria were not applied, as it is expected that potential future travel app data may exhibit similar patterns.

3.3 Target Variable

After preprocessing, the remaining data consisted of 2,916,696 geolocation measurements representing 4,101 labeled trips made by 251 users. The target variable is the respondents' transport mode label. The distribution of transport modes is shown in Table 3.1. Car trips are most frequent, followed by walking and biking. Public transport trips occur considerably less frequently, with less than 1% of labeled trips made by tram. Consequently, the data is highly imbalanced. The average trip duration for respondents is 25 minutes. For comparison, the row 'Netherlands 2023 %' contains the percentages based on the regular survey. The categories bus, tram, and metro are grouped in the official statistics and constitute together 2.2%. In this study, each of these categories is predicted separately. Figure 3.2 shows a boxplot with the trip duration in minutes for each of the seven transport modes.

Table 3.1 Distribution of transport modes in the pre-processed travel app data

Class	Car	Walking	Bike	Train	Bus	Metro	Tram	Other	Total
N	1842	987	932	158	94	56	32	0.0	4101
%	44.9	24.1	22.7	3.9	2.3	1.4	0.8	0.0	100
Netherlands 2023 %	44.5	20.6	26.8	2.6	*	*	*	3.7	100

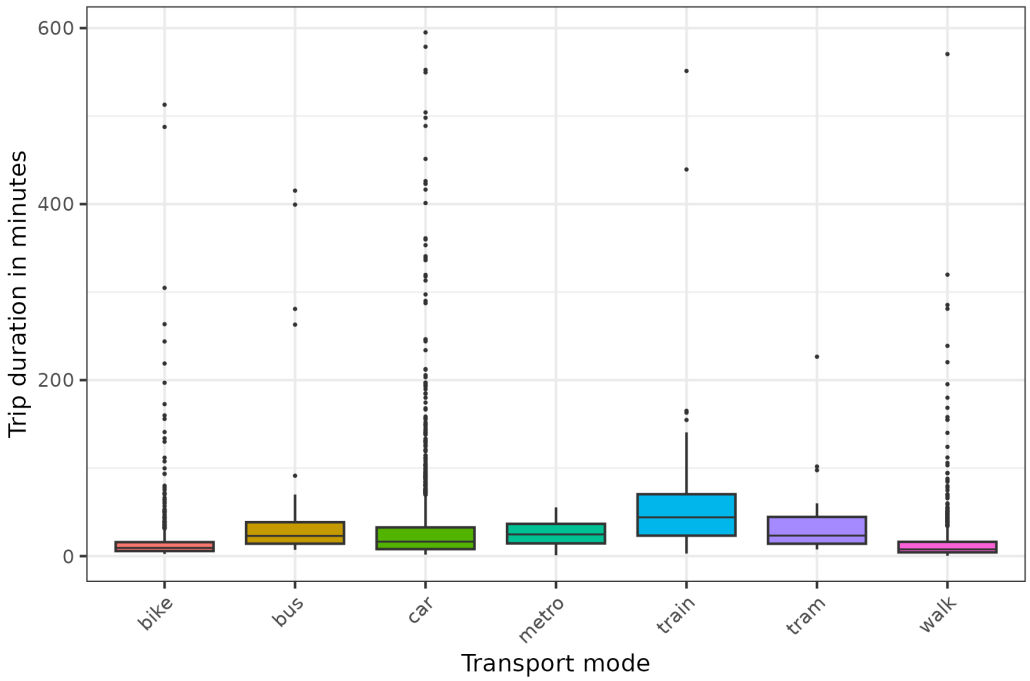

Figure 3.2 Boxplot of trip duration for each of the seven transport modes

3.4 GPS-based Features

GPS-based features were created from the pre-processed geolocation data in two steps. First, features were calculated at the GPS measurement level, which consists of timestamps and longitude/latitude value pairs. Clusters of consecutive GPS measurements that were classified as a track by the stop-algorithm were grouped. For these clusters, the distance between consecutive points was computed using the Haversine formula for great-circle distance to determine the distance traveled between GPS measurements Bansal 2021. The distance traveled and the time difference between points were then used to calculate speed and its higher-order derivatives: acceleration (m/s^2), jerk (m/s^3), and snap (m/s^4). Acceleration measures the rate of change of speed, jerk measures the rate of change of acceleration, and snap measures the rate of change of jerk. These higher-order derivatives were computed to monitor changes in motion that may help distinguish between transport modes. Additionally, the bearing was calculated from the difference in longitude and latitude between the two points. The bearing was calculated only for GPS measurements with a speed of at least 0.6 m/s, as lower speeds yielded highly divergent consecutive bearing values, which were deemed uninformative. All computations were performed with the `geosphere` package in R (Hijmans 2024; R Core Team 2024).

In a second step, GPS measurement-level features were aggregated to the event-level by computing summary statistics. These summary statistics include the mean, median, minimum, maximum, standard deviation, coefficient of variation, autocorrelation coefficient, skewness, kurtosis, interquartile range (IQR), and values at the 1st, 5th, 10th, 90th, 95th, and 99th percentiles. These summary statistics were computed for speed, acceleration, jerk, snap, difference in bearing, altitude, and measurement accuracy. Additionally, the frequency and size of outliers were taken into account for speed, acceleration, jerk, and snap. Outliers were defined as observations that fall more than 1.5 IQR above or below the third or first quartile. For speed and difference in bearing, the proportion of values within pre-specified intervals were computed as well. Finally, the total trip duration is computed as the sum of distances between consecutive GPS points. An overview of all computed features is provided in Appendix A.

Figure 3.3 shows a boxplot of median speed for each of the seven transport modes, one of the summary features extracted from the GPS data. Some of the outliers represent implausible values, such as a median walking speed greater than 50 km/h, which can occur when measurement inaccuracy is high or when trips are mislabeled.

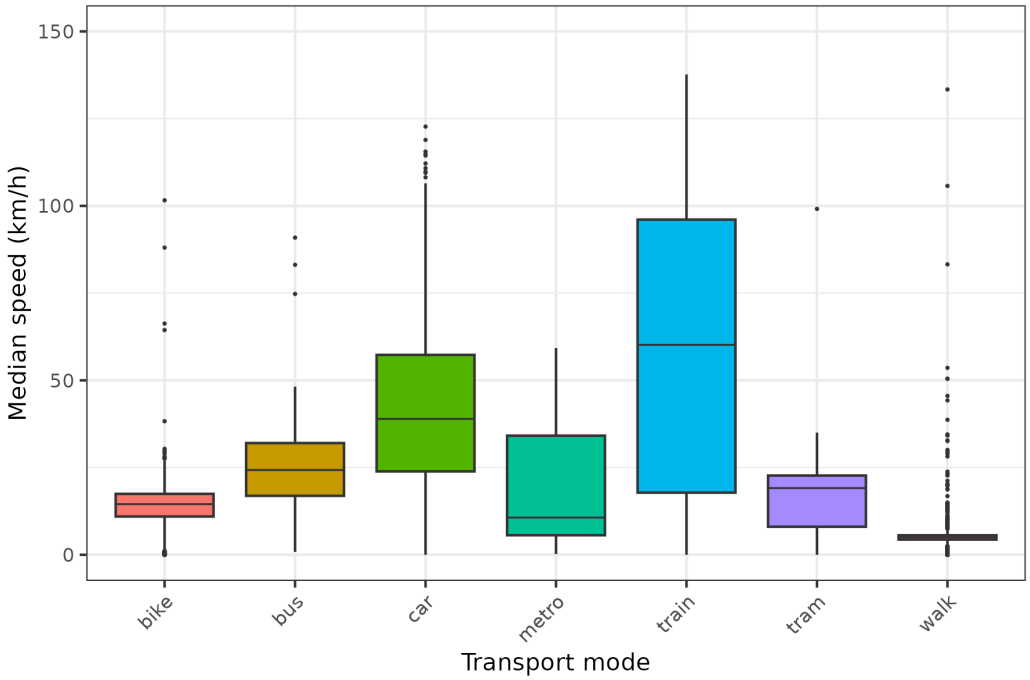

Figure 3.3 Boxplot of median speed for each of the seven transport modes

3.5 OpenStreetMap-based Features

In addition to the GPS data collected through the travel app, external geospatial data from OpenStreetMap (OSM) was used to enrich the GPS data with contextual location-based information. OSM is an open-source platform maintained by a large user community that provides detailed location data, including information on roads, buildings, public transport infrastructure, and other elements relevant to travel behavior analysis (OpenStreetMap Contributors 2025). For this study, OSM data were sourced from Geofabrik (Geofabrik 2025), which provides extracted OSM data for specific regions. OSM data on public transport routes, public transport stations, traffic indicators, and parking amenities are used for the Netherlands and the German regions Baden-Württemberg, Niedersachsen, and Nordrhein-Westfalen. The OSM-based features used in this study are derived from the methods described by Fourie et al. (2025). Their approach for extracting context-location features to enhance transport mode prediction has been adopted with minimal modifications. Features such as the proximity to public transport lines and intersection counts were computed as outlined in their study. An overview is provided in Appendix A.

3.6 Temporal Features

In addition to the GPS and OSM features, multiple temporal features were created to enrich the input data. This is new in contrast to all other transport mode prediction studies conducted at Statistics Netherlands. The work by Zahroh et al. 2025 to predict travel motives also used such temporal features, which were found to be relevant in this context. With these features, we aim to capture patterns in data from previous tracks and stops, and to situate tracks within a daily or weekly context. Such features include indicators of the day of the week, whether a trip was made during morning or evening rush hour, the number of trips made so far on a given day, and the daily cumulative duration and distance traveled. Furthermore, the duration and purpose of previous stops were added as features, along with the duration and transport mode of the last three tracks made by the respondent.

Finally, to capture respondent-specific patterns, respondent baselines were constructed by taking the cumulative mean of various speed-related statistics. These include the cumulative mean of the average, median, standard deviation, interquartile range, and 90th percentile of speed. Additionally, the cumulative mean of trip duration and distance is constructed. For each new track, the difference from the respondent baseline is then computed. For the most common transport modes (car, bike, and walking trips), mode-specific respondent baselines of the mean were also computed.

Besides GPS-based temporal features, OSM-based temporal features have also been created. These consist of user baselines or cumulative means of the number of encounters with public transport stations and public transport lines along tracks, as well as the difference from these user baselines for each new track. An overview of these features can be found in Appendix A.

3.7 External Validation Data

To assess the external validity of the transport mode classification models, an external validation dataset is used. The data were collected using the Statistics Netherlands travel app in the summer of 2024. As this is a later version of the travel app than the one used for the 2022/2023 data collection, the sensor configuration differs slightly. The data were collected as part of Eurostat's Smart Surveys Implementation project (Eurostat 2024) and are publicly available on the project's GitHub (essnet-ssi 2025). It has also been used in various discussion papers published by Statistics Netherlands (Fourie et al. 2025; Klingwort et al. 2025c).

The data were collected specifically to obtain high-quality transport mode labels with no errors. A small group of employees from Statistics Netherlands and Utrecht University was instructed to collect data on trips that were expected to be challenging to classify, such as public transport trips with multiple transfers. Data was collected by five respondents in the Netherlands and Germany, resulting in 137 trips to be classified.

Two differences between the data collected in 2022/2023 and 2024 should be mentioned. First, the available transport modes differ slightly. Whereas the car is the most common transport mode in the 2022/2023 data, no car trips have been made in the 2024 data. Additionally, the 2024 validation data includes trips labeled as 'ferry', a category not included in the 2022/2023 data. Second, no stops were recorded in the 2024 validation data, and the data do not represent regular travel days, unlike the 2022/2023 data. Consequently, temporal features that use information about previous stop duration and purpose have not been added.

4 Methodology

This section describes the training process and evaluation of the machine learning models used for the transport mode classification. A total of eight models have been trained: four Random Forest models and four XGBoost models. The models are trained on four different sets of features. The first feature set includes only GPS-based features. The second set includes both GPS-based and OSM-based features. The third set contains GPS-based and temporal features. Lastly, the fourth feature set contains all features. The remainder of this section describes the modeling approach, including data splitting, model training, and the evaluation metrics used.

4.1 Machine Learning Classification Task

The goal of automatic transport mode detection can be formulated as a multi-class classification task. The prediction instance is a single annotated track. The to-be-classified transport mode labels contain seven classes: bike, bus, car, metro, train, tram, and walk. The input for the classification models are the four feature sets: (1) GPS-based, (2) GPS combined with OSM, (3) GPS combined with temporal features, and (4) the complete set of features. The two classification algorithms are Random Forest and XGBoost.

4.2 Class Imbalance

Given the large class imbalance in the data, inverse class weights were computed and provided to the model to penalize misclassifications of underrepresented classes. The weights assigned to the classes were 0.59 for bike, 6.30 for bus, 0.32 for car, 10.11 for metro, 3.75 for train, 20.71 for tram, and 0.63 for walk. Although a downsampling strategy was also considered for the Random Forest model, inverse class weights were ultimately chosen for ease of implementation with XGBoost. This allows for easier comparison between the two sets of models.

4.3 Training Procedure

As preparation, the data was split into a training and a testing dataset at a 75/25 ratio. A blocking strategy was used to ensure that all trips from a single respondent were included in either the training or test data, but not both. This strategy ensures that the models are evaluated on unseen respondents and may prevent overly optimistic model evaluation when similar trips from a single respondent are used for both training and testing the model.

4.4 Hyperparameter Tuning

Model hyperparameters were tuned using a grid search and 10-fold cross-validation on the training data. The same type of blocking strategy used to split the train and test data was used to create the 10 folds. As a result, tracks from the same respondent were all assigned to the same fold for validation.

4.4.1 Random Forest

Tuning and training of the Random Forest classifier was performed in R (R Core Team 2024) with the tuning pipeline from the `caret` package (Kuhn 2008). The `ranger` implementation of the random forest algorithm was used for training the model parameters (Wright and Ziegler 2017). A grid search was performed for two key model parameters: `mtry`, which represents the number of potential features randomly selected at each tree split, and `min.node.size`, which specifies the minimum number of observations required in a terminal node. The grid search values for the randomly selected features were set at 20, 30, and 40. The minimum node sizes considered were 200, 300, and 400. Whereas the number of randomly sampled features impacts the feature diversity of the Random Forest model, the minimum node size influences the tree depth. Low values for the randomly sampled selected features typically result in a higher variety of features being selected into the model, which may decrease prediction performance if they are minimally correlated with the classification target variable. On the other hand, high values typically allow for stronger variables to be selected more often, but may therefore also result in trees with

higher correlation and an increased risk of overfitting. In contrast, larger minimum node sizes lead to shallower trees and therefore regularize the model, reducing the risk of overfitting. In contrast, lower minimum node sizes result in deeper trees that are more prone to overfitting the data.

The hyperparameter grid search for the random forest model was performed with 200 trees. Afterward, model hyperparameters were selected based on the macro-averaged recall of the cross-validation sets. With these hyperparameters, a final model with 2000 trees was trained on the complete training data. For the Random Forest model with only GPS features, the number of randomly sampled features was set to 20, and the minimum node size was set to 400. For the other three models, the selected hyperparameter values were 300 for the minimum node size and 30 for the number of potential features to be selected. An overview of the four Random Forest models with the final hyperparameter values can be found in Table 4.1.

Table 4.1 Final hyperparameters for the Random Forest classification models

Model	1	2	3	4
Algorithm	Random Forest	Random Forest	Random Forest	Random Forest
Features	GPS	GPS, OSM	GPS, temporal	All
Feature subsample	20	30	30	30
Minimum node size	400	300	300	300
Number of trees	2000	2000	2000	2000

Figure 4.1 illustrates the hyperparameter sensitivity for the Random Forest model trained on the complete feature set. To better illustrate the hyperparameter sensitivity, additional values were added for the minimum node size (0 and 100) and the number of randomly selected features (10 and 50). The figure illustrates that the minimum node size has a larger effect on the mean recall in the cross-validation loop than the number of randomly sampled features. Although the models with minimum node sizes of 0 and 100 achieve visibly lower mean recall scores than those with a minimum node size of 200 or more, the mean recall scores of the models with minimum node sizes of 200, 300, and 400 are much closer to each other.

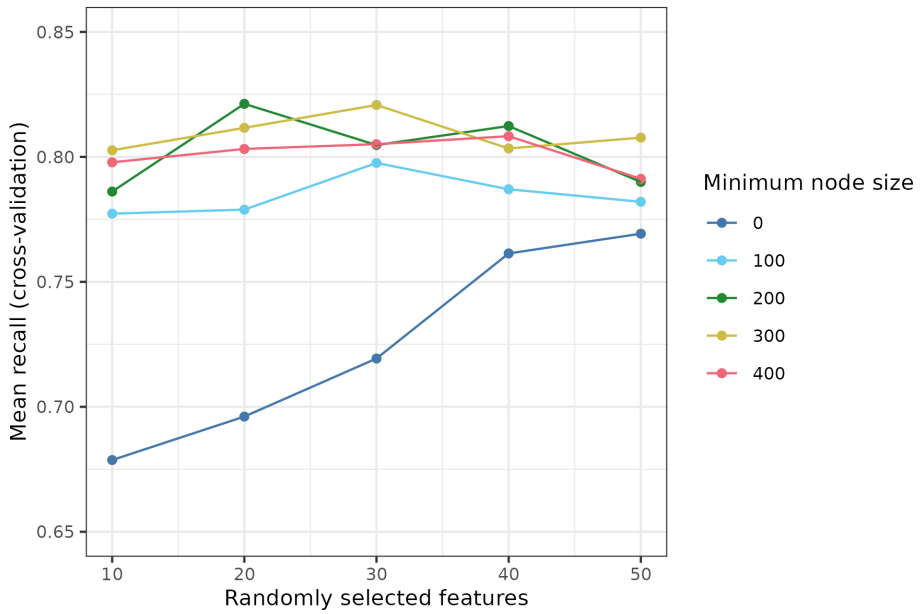

Figure 4.1 Average recall in the cross-validation for different hyperparameter values of the Random Forest model

4.4.2 XGBoost

Unlike the Random Forest algorithm, XGBoost has many tunable hyperparameters. Although there is no consensus in the literature on XGBoost about optimal hyperparameter settings or tuning strategies (Bentéjac et al. 2021; Kapoor and Perrone 2021), the learning rate is often considered one of the key hyperparameters for model performance (Chen and Guestrin 2016).

The XGBoost model hyperparameters were tuned using 10-fold cross-validation in three steps, with the same blocking strategy as for the Random Forest cross-validation. All XGBoost tuning and training was done with the `xgboost` package in R (Chen and Guestrin 2016). In the first step, an initial grid search was performed for the hyperparameters `gamma`, `max_depth`, and `min_child_weight`. These hyperparameters affect the model complexity and regularization. `gamma` represents the minimum loss reduction required to keep branching the tree. Consequently, higher gamma values have a regularizing effect on the model, helping reduce overfitting. Both the maximum depth and the minimum child weight influence the model's complexity. High maximum depth values result in deeper, more specialized trees, whereas high minimum child weights prevent the creation of small, highly specific tree branches. The grid search values for these hyperparameters were 1, 2, and 5 for gamma, 3, 4, and 6 for maximum depth, and 10, 20, and 30 for minimum child weight.

In the second step, a grid search was performed for the hyperparameters `subsample` and `colsample_bytree`. These hyperparameters affect the model's sampling behavior. The `subsample` hyperparameter represents the proportion of observations that is randomly selected from the data to be used in fitting the tree, and the `colsample_bytree` represents the proportion of features that are randomly selected to serve as potential input for the tree. The grid search values for these hyperparameters were 0.5, 0.7, and 0.9 for both the `subsample` and the `colsample_bytree`.

A final grid search was performed to determine the learning rate `eta` and the number of boosting iterations. Learning rates of 0.05, 0.1, 0.2, and 0.3 were tested in a cross-validation setting. The maximum number of boosting iterations was set to 200 trees, with the option to stop the algorithm earlier if no improvement in the evaluation metric was observed in the last 20 iterations.

In all three tuning steps, the hyperparameters were selected based on the multi-class logistic loss in the cross-validation. Table 4.2 shows the final hyperparameter values for each of the four XGBoost models.

Table 4.2 Final hyperparameters for the XGBoost classification models

Model	5	6	7	8
Algorithm	XGBoost	XGBoost	XGBoost	XGBoost
Features	GPS	GPS, OSM	GPS, temporal	All
Gamma	2	1	5	1
Maximum depth	3	3	3	3
Minimum child weight	20	20	30	10
Data subsample	0.9	0.9	0.9	0.9
Feature subsample	0.7	0.7	0.7	0.9
Learning rate (eta)	0.1	0.1	0.3	0.2
Boosting Iterations	42	54	14	23

Figure 4.2 shows the sensitivity of the XGBoost model with all feature sets to different settings of the learning rate hyperparameter. In a 10-fold cross-validation loop, the multi-class logistic loss was tracked for up to 200 boosting iterations (200 trees). If the multi-class logistic log loss did not

decrease in the last 20 iterations, the algorithm was allowed to stop earlier. The log loss is averaged over the validation sets, and four different values of eta were tested in the cross-validation loop. The plot shows that the higher learning rates converge much faster to a stable level within the first 50 iterations. Contrastingly, the lowest learning rate converges more slowly, with the multi-class log loss decreasing considerably more slowly. Although low learning rates converge much more slowly and require a larger number of boosting iterations, and therefore more computational power and time, they also have a lower risk of over-fitting. On the other hand, the fast convergence of the high learning rates can also be seen as a considerable advantage in terms of speed and computational power. The computational power and time required for training the model are minor issues when training the model once. Overfitting is a more severe issue in this case. When the model is retrained during fieldwork, computational power and time become more important issues.

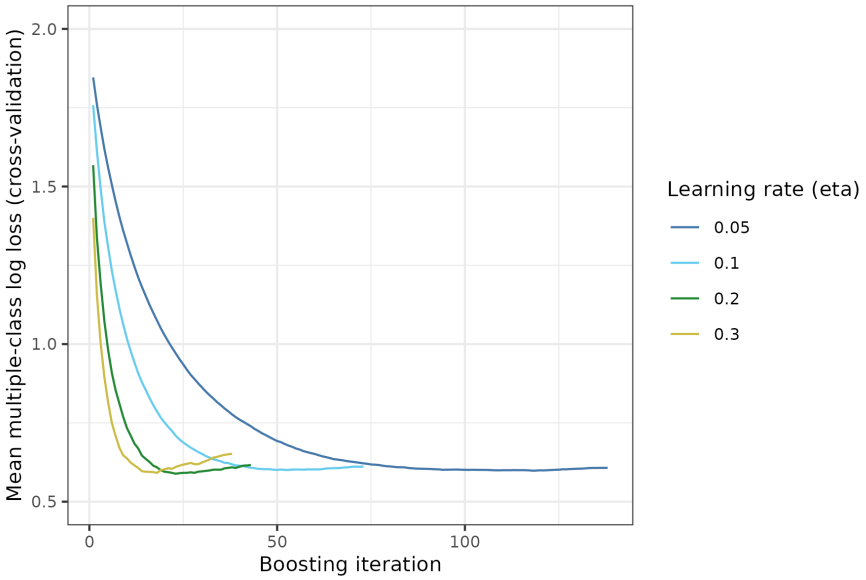

Figure 4.2 Averaged multi-class log loss in cross-validation for different learning rates (*eta*)

4.5 Model Evaluation

4.5.1 Confusion Matrix-based Evaluation Methods

We measured model performance using accuracy, balanced accuracy, macro-averaged F_1 score, and average recall. The last three evaluation criteria are confusion-matrix-based metrics that are considered suitable for multi-class classification in imbalanced datasets (Luque et al. 2019). Whereas balanced accuracy for each class is the mean of the true positive rate (recall) and the true negative rate, the F_1 score balances precision and recall. F_1 scores close to 1 indicate high precision and recall, whereas scores below 0.5 indicate a poor prediction performance.

Although accuracy is typically not recommended for imbalanced datasets because it often favors the correct prediction of large classes, it was chosen as an evaluation metric for comparing datasets. The validation dataset contains the class ‘ferry’, which was not included in the 2022/2023 data used to train the models, and it does not include any car tracks. Consequently, any evaluation metrics requiring the recall and precision cannot be correctly computed or are not directly comparable. The overall accuracy is simply the number of correct predictions divided by the total number of predictions.

The F_1 score for a single class is the harmonic mean of precision and recall:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

where the recall is given by $\frac{\text{TP (true positives)}}{\text{TP+FN (false negatives)}}$

and the precision is given by $\frac{\text{TP}}{\text{TP+FP (false positives)}}$.

The macro-averaged F_1 is then computed by taking the mean of the F_1 scores over all classes.

The balanced accuracy for a single class is the mean of the true positive rate and true negative rate:

$$\text{Balanced Accuracy} = \frac{1}{2}\left(\frac{\text{TP}}{\text{TP} + \text{FP}} + \frac{\text{TN}}{\text{TN} + \text{FN}}\right)$$

4.5.2 Feature Importance

For the Random Forest models, feature importance is evaluated using the permutation variable importance measure (Altmann et al. 2010). This measure can be understood as the mean decrease in prediction accuracy and calculated by looking at the decrease in accuracy when the values of a single feature are randomly shuffled, thereby isolating the effect of that particular feature on the model's prediction performance. This method is often preferred over the default impurity or Gini importance measure (Janitza et al. 2018).

On the other hand, XGBoost computes feature importance with three importance metrics. First, the gain is the improvement in prediction accuracy resulting from splitting based on that feature. Second, the cover is the number of observations in the training data that were affected by a branch split based on that feature. Lastly, the frequency is the number of times a feature is used in all the grown trees.

4.5.3 Predicted Pseudo-Probabilities

In addition to the predicted transport modes, pseudo-probabilities for each transport mode were also extracted from the Random Forest and XGBoost models[1)]. These predicted probabilities were used to visualize the relationships among the probability assigned to the predicted class, the probability assigned to the second-most-likely class, and whether the initial predicted mode was correct.

This exploratory analysis was conducted to assess whether higher predicted probabilities are associated with greater prediction accuracy and whether these probabilities are informative for a potential implementation of respondent prompts in the travel app. The idea behind such prompts is that whenever there is high uncertainty in a transport mode classification, a targeted prompt could be sent in the travel app, asking respondents to either validate the classification or choose between two or more transport modes.

1) We refer to Puts and Daas (2021), for further reading on pseudo-probabilities.

Although both Random Forest and XGBoost models can return predicted class probabilities, these probabilities are computed differently. Whereas Random Forest calculates the predicted class probability as the proportion of decision trees in the full ensemble that vote for that particular class, XGBoost obtains predicted probabilities by applying a softmax function to the logits produced at the leaves of the trees and accumulated over boosting iterations, normalizing these to sum to one across the different classes. Even though these model outputs are expressed as probabilities, neither Random Forest nor XGBoost is a probabilistic generative model and therefore does not return actual likelihoods or true probability estimates. Consequently, predicted probabilities are understood as pseudo-probabilities in this study. Although not true probabilities, they may still provide information about the classification models' predictive uncertainty.

5 Results

This section shows the prediction performance of the eight machine learning classification models. Confusion matrices, variable importance and hyperparameter effects are visualised for the best performing models. Additionally, these models are evaluated on the external validation dataset from 2024.

5.1 Predictive Performance

In this section, the predictive performance of the models developed in this study is presented. Table 5.1 shows the prediction performance of the eight trained machine learning models evaluated on the test data according to the accuracy rate, mean balanced accuracy, mean recall, and macro-averaged F_1 score. Overall, the best-performing models on the test set are the Random Forest and XGBoost models trained on all feature sets. Whereas the XGBoost model has the highest overall accuracy (.91) and macro-averaged F_1-score (.84) out of the eight models, the Random Forest has the highest balanced accuracy (.93) and mean recall (.87). However, the difference in the evaluation scores is larger for the metrics where XGBoost outperforms the Random Forest (accuracy and F_1-score). For the metrics where the Random Forest outperforms the XGBoost (balanced accuracy and recall), the difference between the two models is smaller. When comparing the Random Forest and XGBoost models with other feature sets, XGBoost consistently scores better on the overall accuracy and macro-averaged F_1-score than the Random Forest alternative, except for the models with both GPS and OSM-based features, where the scores are approximately the same. XGBoost also slightly outperforms Random Forest on balanced accuracy and recall for the GPS-only and GPS + temporal models. However, for the models that combine GPS with OSM features, the Random Forest achieves higher scores for these two metrics than the XGBoost. Regarding the feature sets, adding temporal features slightly improves prediction performance. For both the Random Forest and XGBoost models, those with GPS and temporal features achieve higher evaluation scores than those with only GPS features, and those with the complete feature set perform better than those with GPS and OSM features. This difference in prediction performance is very minimal for the Random Forest, where none of the metrics improve by more than .03 in the evaluation score. However, we observe a large difference with XGBoost, where adding temporal features alongside GPS and OSM features increases the macro-averaged F_1-score from .75 to .84.

Table 5.1 Evaluation metrics for the Random Forest (RF) and XGBoost (XGB) machine learning models, evaluated on the test data

Model	Algorithm	Features	Accuracy	Mean balanced accuracy	Mean recall	Macro-averaged F_1-score
1	RF	GPS	.70	.78	.62	.50
2	RF	GPS, OSM	.86	.92	.86	.75
3	RF	GPS, temporal	.81	.81	.65	.57
4	RF	All	.87	.93	.87	.78
5	XGB	GPS	.80	.80	.63	.57
6	XGB	GPS, OSM	.88	.88	.78	.75
7	XGB	GPS, temporal	.86	.85	.72	.66
8	XGB	All	.91	.92	.85	.84

Table 5.2 and 5.3 present the confusion matrices for the Random Forest and XGBoost models trained on all feature sets, as well as the recall, precision, and F_1-score for each class. For both models, we see that the classification of the dominant classes is typically better than that of the less represented classes, in particular, the tram and metro. When comparing the two confusion matrices, the largest discrepancy between the two models is in the smallest tram class. Whereas the XGBoost predicts the transport mode as tram for 13 observations, 9 of which are correct, the Random Forest predicts tram 33 times. Although this results in perfect recall, a large number of car trips have also been misclassified as tram trips, thereby lowering precision. Additionally, the Random Forest's recall for the bus class is considerably higher than the XGBoost's, but lower for the other classes.

Table 5.2 Confusion matrix of the Random Forest model with all features, evaluated on the test data

	Actual Mode							
Predicted Mode	Bike	Bus	Car	Metro	Train	Tram	Walk	Total
Bike	171	0	18	0	2	0	26	217
Bus	2	22	4	0	0	0	0	28
Car	10	2	426	0	2	0	14	454
Metro	0	1	2	10	0	0	4	17
Train	1	0	1	2	36	0	2	42
Tram	4	0	15	1	1	11	1	33
Walk	7	0	10	0	1	0	248	266
Total	195	25	476	13	42	11	295	1057
Recall	.88	.88	.89	.77	.86	1.00	.84	.87
Precision	.79	.79	.94	.59	.86	.33	.93	.75
F_1-score	.83	.83	.92	.67	.86	.50	.88	.78

Table 5.3 Confusion matrix of the XGBoost model with all features, evaluated on the test data

	Actual Mode							
Predicted Mode	Bike	Bus	Car	Metro	Train	Tram	Walk	Total
Bike	184	0	10	0	2	2	17	215
Bus	1	19	2	1	0	0	0	23
Car	3	5	446	0	1	0	13	468
Metro	0	0	2	10	0	0	3	15
Train	1	1	1	2	36	0	2	43
Tram	1	0	2	0	0	9	1	13
Walk	5	0	13	0	3	0	259	280
Total	195	25	476	13	42	11	295	1057
Recall	.94	.76	.94	.77	.86	.82	.88	.85
Precision	.86	.83	.95	.67	.84	.69	.92	.82
F_1-score	.90	.79	.94	.71	.85	.75	.90	.84

5.2 Feature Importance

Figures 5.1 and 5.2 show the 20 most important features in the Random Forest and XGBoost models trained on the complete feature set. In the Random Forest, the most important features are determined by the permutation variable importance method. In contrast, in the XGBoost model, the average gain in accuracy determines which features are most important. When comparing the two models, we see that features related to speed are essential in both. For the Random Forest model, almost all features deemed important are related to speed. The exceptions are two OSM-based features on the proximity to bus routes and two temporal features that contain information on the previous trip mode. Consequently, we can consider that speed summary statistics play an important role in distinguishing between transport modes.

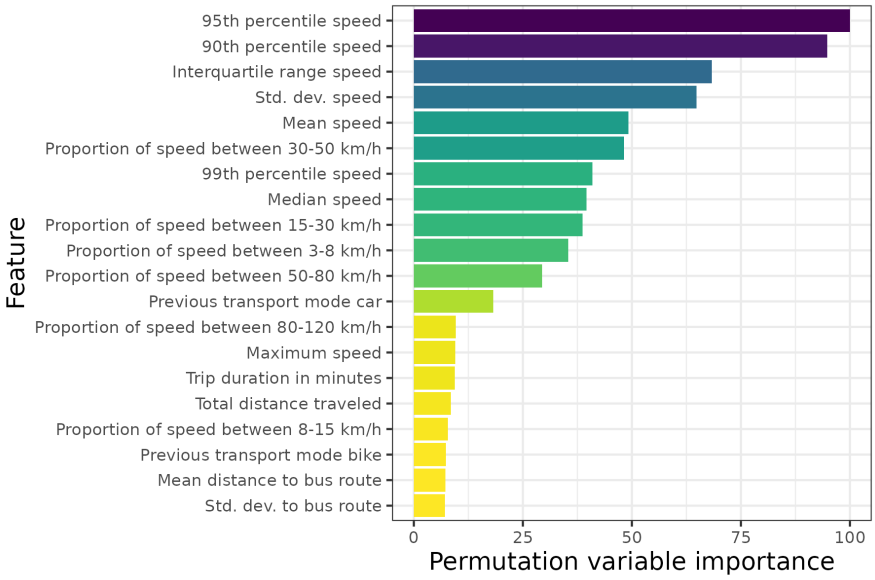

Figure 5.1 Top 20 most important variables in the Random Forest model based on the permutation method

Whereas the speed variables are highly represented among the top 20 most important variables in the Random Forest model, the XGBoost model includes a broader variety of feature types. Proximity to public transport lines is highly influential in this model, with proximity to metro, bus,

tram, and train routes ranking high on the list. While speed-based features remain clearly still very important in the model, OSM-based features play a larger role in distinguishing between transport modes.

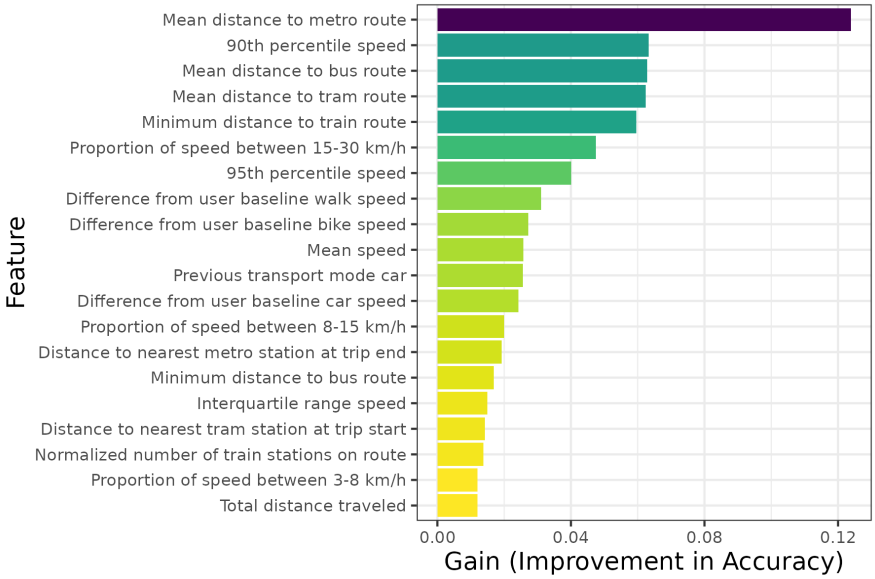

Figure 5.2 Top 20 most important variables in the XGBoost model based on the average gain in accuracy

Additionally, some temporal features are among the top 20 most influential features. Besides information on the previous transport mode, which was also among the Random Forest's most important features, XGBoost also considers the trip's average speed difference relative to the cumulative mean of a respondent's car, bike, and walking speeds. Compared to the Random Forest model, the XGBoost model considers more features that provide specific information or context to distinguish between single transport modes. In contrast, the Random Forest model prioritizes speed-related features that can distinguish between a larger number of modes simultaneously.

5.3 Model Validation on Additional Data

To check generalisability, the Random Forest and XGBoost models were evaluated on an external validation dataset. Table 5.4 compares the overall accuracy rate achieved on the test data and the external validation data. For the Random Forest and XGBoost models trained on the complete feature set, confusion matrices and class-specific metrics are also available for comparison in Table 5.5 and 5.6.

Table 5.4 shows that each of the eight models achieves a considerably lower accuracy on the external evaluation data compared to the test data. The sharp drop in accuracy may indicate that these models are specifically trained on data collected in a specific context and time, and therefore do not generalize fully to respondents or time periods outside this context.

Among the eight models, the XGBoost model trained on the complete feature set achieves the highest accuracy (.84) on the external validation data. This is considerably higher than the accuracy of the Random Forest models trained on all features (.74). Furthermore, the difference in accuracy between the model evaluated on the test data and on the external validation data is smallest for the XGBoost model with all features. This may indicate that the final XGBoost model generalizes best outside the specific context in which the original data was collected.

Table 5.4 Overall accuracy of the eight machine learning models evaluation on the test and external validation data

Model	Algorithm	Features	Accuracy on test data	Accuracy on external validation data
1	Random Forest	GPS	.70	.56
2	Random Forest	GPS, OSM	.86	.73
3	Random Forest	GPS, temporal	.81	.57
4	Random Forest	All	.87	.74
5	XGboost	GPS	.80	.63
6	XGboost	GPS, OSM	.88	.77
7	XGboost	GPS, temporal	.86	.69
8	XGboost	All	.91	.84

In the confusion matrices of Tables 5.5 and 5.6, we observe that both models perform relatively well at classifying train and walking trips, with XGBoost also achieving good results for the metro class. Even though bike trips were one of the larger classes in the original training data, neither model performs well at classifying this mode, with the Random Forest achieving a modest precision of .37 and the XGBoost achieving 0.60. In general, the XGBoost model outperforms the Random Forest, achieving higher precision and F_1-scores across most classes.

Table 5.5 Confusion matrix of the Random Forest model with all features, evaluated on the external validation data

	Actual Mode								
Predicted Mode	Bike	Bus	Car	Ferry	Metro	Train	Tram	Walk	Total
Bike	7	1	0	3	0	0	1	7	19
Bus	1	4	0	0	0	1	0	0	6
Car	0	0	0	0	0	0	0	1	1
Ferry	0	0	0	0	0	0	0	0	0
Metro	0	0	0	0	4	1	2	7	14
Train	0	0	0	0	0	7	1	0	8
Tram	2	0	0	0	1	0	23	7	33
Walk	0	0	0	0	0	0	0	56	56
Total	10	5	0	3	5	9	27	78	137
Recall	.70	.80	-	0	.80	.78	.85	.72	-
Precision	.37	.67	-	-	.29	.88	.70	1.00	-
F_1-score	.48	.73	-	-	.42	.82	.77	.84	-

Table 5.6 Confusion matrix of the XGBoost model with all features, evaluated on the external validation data

	Actual Mode								
Predicted Mode	Bike	Bus	Car	Ferry	Metro	Train	Tram	Walk	Total
Bike	9	1	0	1	0	0	0	4	15
Bus	0	3	0	0	0	0	0	2	5
Car	0	0	0	0	0	0	0	0	0
Ferry	0	0	0	0	0	0	0	0	0
Metro	0	0	0	0	4	0	0	0	4
Train	0	0	0	0	0	9	2	0	11
Tram	1	1	0	0	1	0	23	5	31
Walk	0	0	0	2	0	0	2	67	71
Total	10	5	0	3	5	9	27	78	137
Recall	.90	.60	-	0	.80	1.00	.85	.86	-
Precision	.60	.60	-	-	1.00	.82	.74	.94	-
F_1-score	.72	.60	-	-	.89	.90	.79	.90	-

5.4 Misclassification

A brief comparison of misclassification across the XGBoost and Random Forest models with all features indicates considerable overlap in misclassified trips between the two models. Of the 94 misclassifications made by the XGBoost model on the internal test data, 76 (81%) were also misclassified by the Random Forest model.

These trips may be difficult to classify in general due to data quality or ambiguous travel patterns. Misclassification and data quality were briefly explored in terms of the number of geolocations and the average accuracy (in meters). The 20% of trips with the fewest GPS points (fewer than 142 measurements) accounted for 45% of misclassifications in the XGBoost model evaluated on the test data. Similarly, the 20% of trips with the highest average accuracy (greater than 12 meters) accounted for 39% of misclassification in the XGBoost model. Additionally, very long trips also seemed more difficult to classify. Among the 11 trips in the test data with a travel duration of 5 hours or longer, 27% were misclassified, compared with the overall misclassification rate of 9%.

Potentially ambiguous travel patterns are evident in some misclassifications by the XGBoost model for classes where we expect distinct trip characteristics. For example, the average mean speed of all the labeled walking trips in the test data was 7.26 km/h. However, the 13 walking trips in the test data that were classified as car trips by the XGBoost model (see Table 5.3) had an average mean speed of 27.03 km/h. Similarly, the average speed of labeled train trips in the test data is 170.20 km/h, while the average speed of the three labeled train trips classified as walking trips is 13.87 km/h.

5.5 Predicted Pseudo-Probabilities

In this subsection, the pseudo-probabilities predicted on the test data by the Random Forest and XGBoost models with all features are visualized to explore model uncertainty. Figures 5.3 and 5.4 show scatterplots of the predicted pseudo-probabilities for the top two classes with the highest probability, for each of the original transport mode labels. The x-axis represents the predicted pseudo-probability of the predicted transport mode, and the y-axis indicates the probability of the second most likely transport mode. Figure 5.3 (Random Forest) shows that for most of the

correctly classified trips, the model assigned a relatively high probability to the predicted class. The under-represented transport modes, tram, metro, and bus, seem to be an exception, with both correct and incorrect classifications generally having lower predicted probabilities for the predicted class. Although most misclassified points appear where the predicted probability is low and the gap between the two probabilities is small, misclassifications with a high pseudo-probability for the predicted class also occur. This suggests that model prediction uncertainty could, to some extent, be informative for prediction accuracy and may be addressed via potential respondent prompts. However, it is unlikely that all misclassifications could be targeted through such a strategy.

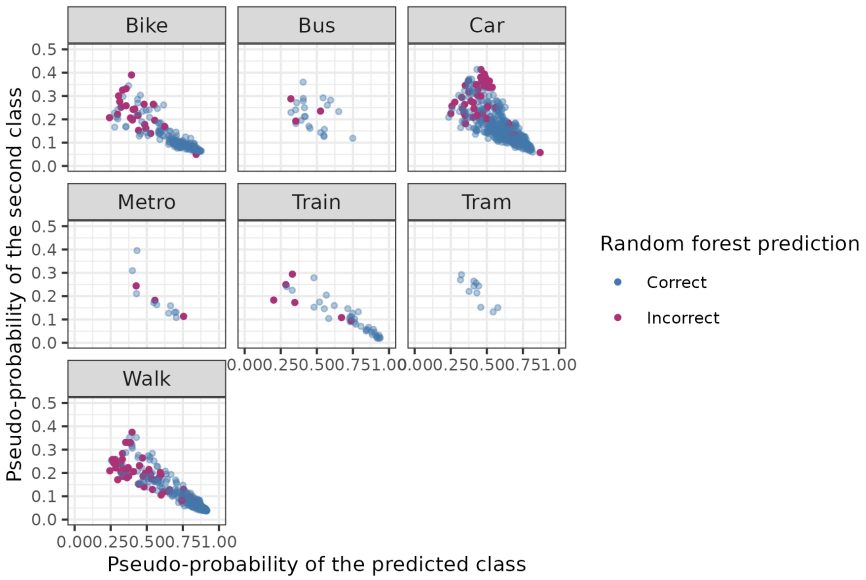

Figure 5.3 Predicted pseudo-probabilities of the predicted transport mode against the predicted pseudo-probabilities of the second most likely transport mode in the Random Forest model (all features), evaluated on the test data

Similar patterns can be observed in Figure 5.4 (XGBoost), although we also see that XGBoost pseudo-probabilities are typically higher than those from the Random Forest model. The density of correct predictions in the bottom-right corner is high across many classes, indicating that many correct predictions receive high predicted pseudo-probabilities in the XGBoost model. The tram class is the largest exception, since both correct and incorrect predictions have relatively low predicted probabilities compared to other classes. As with Random Forests, incorrect predictions are more often assigned lower pseudo-probabilities than correct ones. However, there are multiple instances of incorrect predictions with high pseudo-probabilities as well.

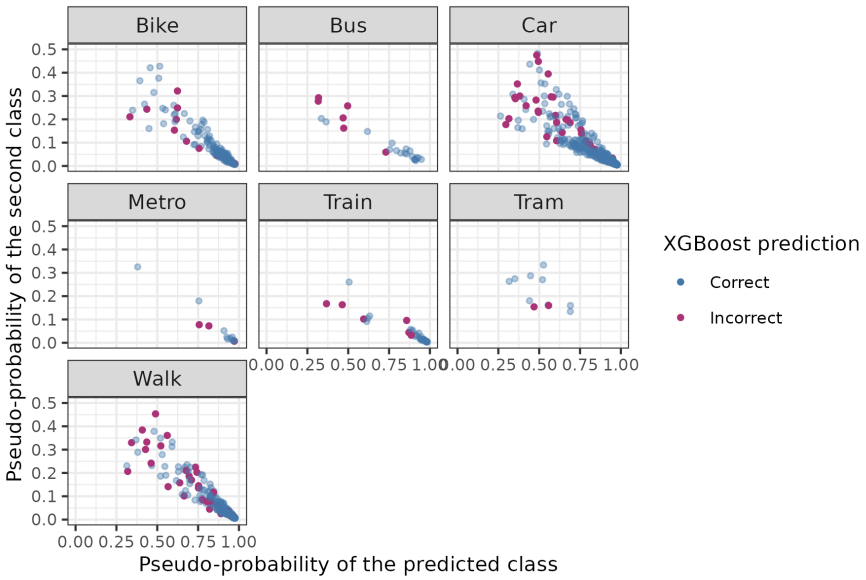

Figure 5.4 Predicted pseudo-probabilities of the predicted transport mode against the predicted pseudo-probabilities of the second most likely transport mode in the XGBoost model (all features), evaluated on the test data

5.6 Respondent prompts

We simulated the potential effect of asking respondents to choose between the top two predicted classes based on the model's pseudo-probabilities. For this exploratory analysis, incorrect test set predictions were corrected by replacing the original predicted class with the second-most-likely class. The results for both models are shown in Figure 5.5. The x-axis represents the proportion of the test data that was subjected to the adjustment, based on quantiles of the predicted pseudo-probability for the predicted class. For example, at 10%, the adjustment is applied to the initially incorrect predictions that fall within the 10% of all observations that received the lowest predicted probability for the predicted class. Recalculated metrics are presented on the y-axis and include the overall prediction accuracy and macro-averaged F_1-score for both the Random Forest and XGBoost models with all features.

When adjusting the incorrect predictions based on the relative value of the predicted probabilities, both the accuracy and macro-averaged F_1-score initially increase considerably. When incorrect predictions within the 30% lowest predicted probabilities are changed to the second most likely class, the Random Forest accuracy changes to .93 compared to .87 when no adjustments are made. Additionally, the macro-averaged F_1-score increases from .78 to .87. The XGBoost results show similar patterns, with the accuracy increasing from .91 to .96 and the macro-averaged F_1-score increasing from .84 to .91. However, the increase in the evaluation metrics stagnates and sometimes even diminishes above 30% of adjustment. This is mainly due to many misclassifications receiving relatively lower predicted probabilities and therefore being adjusted early on. Another potential reason for the stagnation is that misclassifications with high predicted probabilities have low probabilities for the remaining classes, leading to slight differences and high uncertainty among the second, third-, and fourth-most-likely classes. Therefore, second-best predictions are less likely to be informative. Additionally, when misclassified predictions are changed to second-best predictions that are also incorrect, the macro-averaged F_1-score can be negatively affected when incorrect predictions change from a higher to a lower represented class.

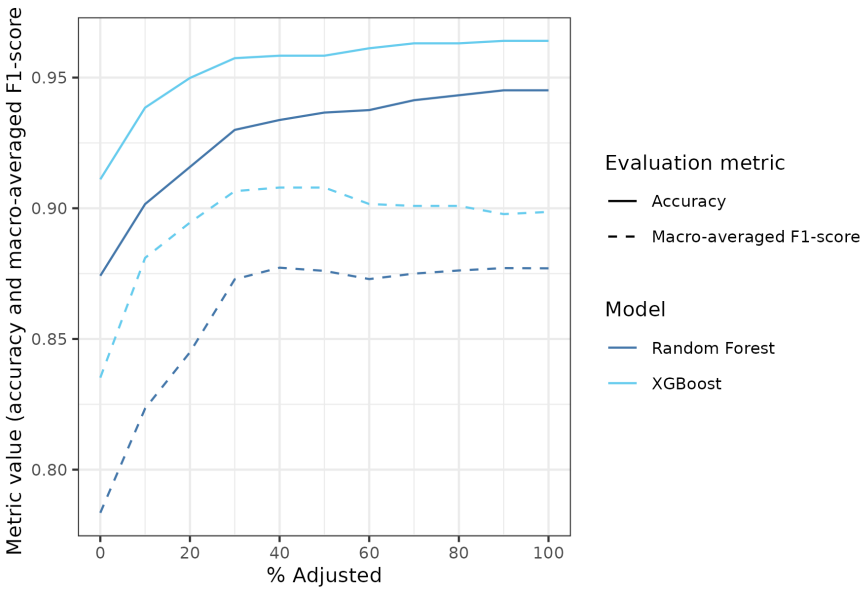

Figure 5.5 Accuracy and macro-averaged F_1-score for the Random Forest and XGBoost models (all features) evaluated on the test set, when adjusting the misclassified instances to the second most likely transport mode based on the pseudo-probability of the predicted class

6 Discussion

In this study, we trained and evaluated eight supervised machine learning models for automatic transport mode detection using GPS data collected in a smart survey travel app. We aimed to evaluate the performance of two supervised machine learning techniques: Random Forest and Extreme Gradient Boosting (XGBoost). Additionally, it was investigated whether including temporal features alongside GPS and OpenStreetMap (OSM) features improved the model's prediction performance.

The XGBoost model trained on the complete feature set achieved the highest overall prediction performance, with an accuracy of .91 and macro-averaged F_1-score of .84 on the test data. This model seemed to generalize moderately well when evaluated on an external validation set collected in a different time period and context, with different travel app settings, achieving an accuracy of 84%. However, the confusion matrices and F_1-scores indicate that some classes were more challenging to predict in the test and external validation data. This was often the case for the under-represented transport modes, including the tram, metro, and bus.

6.1 Value of Temporal Features

The temporal features had a modest contribution to the model performance, especially compared to the OSM features. For both the Random Forest and XGBoost models, models that included temporal features outperformed those using only GPS features, and models that included all features outperformed those combining GPS with OSM features. However, this improvement was more substantial in the XGBoost models than in the Random Forest models, where it was small. The external validation results indicated that the temporal features generalized well to a different context, especially for the XGBoost model.

However, a considerable limitation is that some of these temporal features rely on labeled data from previous events. If this information is correct and available, these temporal features may improve prediction performance, especially when data are collected over a more extended period. Nonetheless, if correctly labeled data is unavailable, these features can only be constructed by using prediction results from earlier trips as input for predicting subsequent trips, which is likely to decrease performance. Consequently, the merit of implementing a transport mode classification model with label-dependent temporal features in a travel app may depend on how many days of data are collected and whether respondents are asked to inspect or validate the collected data.

6.2 Related Work

Compared to previous studies, the XGBoost model with all features performs reasonably well. It improves upon the prediction accuracy of the Random Forest models in Smeets et al. (2019), who reported a prediction accuracy of .64 when classifying a similar number of transport modes, and an accuracy of .85 when collapsing the classes into walking, motorized transport modes, and public transport. The 2018 app used for this study was limited in respondent retractions. For example, the start and end times of a stop/track could not be edited.

Compared to the study by Fourie et al. (2025), who implemented a rule-based algorithm to predict transport mode on the same data that was used in this study, the XGBoost model with all features achieved slightly higher prediction accuracy. Their study reported an accuracy of .85 and a balanced accuracy of .70. Their rule-based algorithm was also evaluated on the external validation data that was used in this study, resulting in an accuracy of .80, with the minor difference that their study excluded the ferry trips from the analysis and allowed for multiple transport mode predictions. The rule-based algorithm used only 13 GPS and OSM-based features, resulting in a model with lower complexity and higher interpretability than those in this study.

A specific limitation of this study regarding interpretability concerns feature importance. Due to the high dimensionality of the data, some of the features are strongly correlated. Although the Random Forest and XGBoost algorithms appear relatively robust to multicollinearity (e.g., Lindner et al. 2022; Neyra et al. 2024), the reliability of feature importance metrics may be compromised, leading to an inaccurate representation of individual feature effects. Although the feature importance plots of these relatively complex models can provide a general understanding of which feature types are important for transport mode classification, specific individual-feature effects should be interpreted with caution.

6.3 Travel App Respondent Prompts

Although the XGBoost model with the complete feature set performed relatively well on both the test and external validation data, it did not achieve consistent predictive performance across all transport mode classes. Therefore, a fully automated travel app approach without any respondent interaction or validation may not be feasible. A potential alternative may be a semi-automated approach with targeted respondent prompts. Respondents could be prompted to confirm transport mode classifications based on prediction uncertainty, data quality, or other variables or considerations. Although such an approach is not fully automated, respondents may experience less burden than in a travel app implementation, where all data needs to be validated.

This study briefly explored the possibility of using predicted pseudo-probabilities from classification models to inform targeted respondent prompts. The simulated respondent prompt example in Figure 5.5 indicates that sending out respondents' prompts for 10% to 30% of the recorded trips could already increase accuracy considerably, assuming respondents select the correct transport mode.

Future work could investigate potential pseudo-probability thresholds for targeted prompts, which differ for various models. Additionally, data quality could be considered as information for respondent prompts as well. Misclassification in the XGBoost model occurred more often when a relatively low number of GPS measurements were collected for a trip, or when the average GPS measurement accuracy (in meters) was high. For these data quality-related measures, there may also be thresholds that could be optimized for sending out targeted respondent prompts. Although not explored in this study, other variables or patterns may also be informative for prompt strategies that balance response burden and classification accuracy.

In addition to increasing classification accuracy, targeted respondent prompts could also serve other purposes. If a machine learning classification model were implemented in a travel survey app, additional labeled and validated data would be beneficial for periodic retraining and re-evaluation. Respondent prompts could then be optimized to collect additional labeled data on trips that exhibit atypical travel patterns. Alternatively, respondent prompts could be sent out whenever an under-represented public transport mode is expected, to reduce class imbalance or better understand travel patterns for these modes.

6.4 Limitations and Future Work

This study has several additional limitations that should be addressed. First, all transport mode labels were treated as ground truth, even though the data may contain incorrectly labeled trips. These may affect the reliability of temporal features that rely on previously labeled events and the overall model performance. As many transport mode detection studies typically use non-probability sampling with respondents who have been specifically instructed or trained to collect GPS data (Sadeghian et al. 2021), there has been limited investigation into the effect of potentially incorrect labels in transport mode detection. Second, although the trained classification models were evaluated on external validation data in addition to internal test data, the evaluation was still limited to two specific contexts. Third, to better understand model generalisability, evaluating model performance on broader contexts would be beneficial. Currently, there is no standard framework for comparing transport mode detection models developed in different demographic, temporal, and geographical contexts. Future work on transport mode detection may benefit from the development of a standard evaluation dataset or framework. Fourth, the model fit is likely dependent on the number of available data points. This was not varied during the study, and additional data may yield more optimal parameter choices. Fifth, prior to transport mode prediction, the location data is split into segments of stops and tracks. In the current study, these track segments were assumed to be fixed. In contrast, in reality, improvements to track segmentation may be possible, which will likely influence the results of this study. Lastly, this work did not consider multi-modal trips, because the annotations were single-modal labels. In reality, multi-modal trips occur and are currently reflected in this model only if the segmentation splits multi-modal trips into separate segments.

7 Conclusion

This study evaluated and compared the performance of Random Forest and XGBoost models for transport mode detection using GPS data collected in a smartphone travel survey app. In addition to GPS-based information, context-location information from OpenStreetMap (OSM) and temporal features capturing past travel behavior were used to train the models. The XGBoost model trained on all features achieved the highest overall accuracy and macro-averaged F_1-score, and also performed best on an external validation dataset. Although OSM-based features yielded the largest performance improvements, adding temporal features yielded modest improvements in the XGBoost models. However, the choice of model and features may also depend on practical considerations, such as computational costs and speed.

Based on the results of this study, a fully automated approach to transport mode classification may not be feasible. In particular, underrepresented transport modes are challenging to classify consistently. Alternatively, smartphone-based travel surveys may explore semi-automated approaches that utilize targeted respondent prompts, balancing classification accuracy with minimizing response burden.

Acknowledgments

The views expressed in this paper are those of the author(s) and do not necessarily reflect the policies of Statistics Netherlands.

We would like to thank Barry Schouten for his time and valuable feedback that improved the manuscript.

References

Altmann, A., L. Toloşi, O. Sander, and T. Lengauer (2010). "Permutation importance: A corrected feature importance measure." In: *Bioinformatics* 26.10, pp. 1340–1347. DOI: 10.1093/bioinformatics/btq134.

Axhausen, K. W., M. Löchl, R. Schlich, T Buhl, and P. Widmer (2007). "Fatigue in long-duration travel diaries." In: *Transportation* 34, pp. 143–160. DOI: 10.1007/s11116-006-9106-4.

Bansal, R (2021). "Deriving and testing the great circle theory." In: *International Journal of Statistics and Applied Mathematics* 6.5, pp. 16–24. DOI: 10.22271/maths.2021.v6.i5a.722.

Bedogni, L., M. Di Felice, and L. Bononi (2016). "Context-aware Android applications through transportation mode detection techniques." In: *Wireless Communications and Mobile Computing* 16.16, pp. 2523–2541. DOI: /10.1002/wcm.2702.

Bentéjac, C., A. Csörgő, and G. Martínez-Muñoz (2021). "A comparative analysis of gradient boosting algorithms." In: *Artificial Intelligence Review* 54.3, pp. 1937–1967. DOI: 10.1007/s10462-020-09896-5.

Bjerre-Nielsen, A., K. Minor, P. Sapieżyński, S. Lehmann, and D. D. Lassen (2020). "Inferring transportation mode from smartphone sensors: Evaluating the potential of Wi-Fi and Bluetooth." In: *PLOS ONE* 15.7, e0234003. DOI: 10.1371/journal.pone.0234003.

Bohte, W. and K. Maat (2009). "Deriving and validating trip purposes and travel modes for multi-day GPS-based travel surveys: A large-scale application in the Netherlands." In: *Transportation Research Part C: Emerging Technologies* 17.3, pp. 285–297. DOI: 10.1016/j.trc.2008.11.004.

Bolbol, A., T. Cheng, I. Tsapakis, and J. Haworth (2012). "Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification." In: *Computers, Environment and Urban Systems* 36.6, pp. 526–537. DOI: 10.1016/j.compenvurbsys.2012.06.001.

Boulesteix, A.-L. and M. Schmid (2014). "Machine learning versus statistical modeling." In: *Biometrical Journal* 56.4, pp. 588–593. DOI: 10.1002/bimj.201300226.

Breiman, L. (2001). "Random forests." In: *Machine learning* 45, pp. 5–32. DOI: 10.1023/A:1010933404324.

Calabrese, F., M. Diao, G. Di Lorenzo, J. Ferreira Jr, and C. Ratti (2013). "Understanding individual mobility patterns from urban sensing data: A mobile phone trace example." In: *Transportation Research Part C: Emerging Technologies* 26, pp. 301–313. DOI: 10.1016/j.trc.2012.09.009.

Chen, T. and C. Guestrin (2016). "XGBoost: A scalable tree boosting system." In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: Association for Computing Machinery, pp. 785–794. ISBN: 9781450342322. DOI: 10.1145/2939672.2939785.

essnet-ssi (2025). *geoservice-ssi*. GitHub repository. URL: https://github.com/essnet-ssi/geoservice-ssi#.

Eurostat (2024). *Smart surveys implementation*. URL: https://cros.ec.europa.eu/dashboard/trusted-smart-surveys.

Fourie, J., J. Klingwort, and Y. Gootzen (2025). *Rule-based transport mode classification in a smart travel and mobility survey*. Discussion Paper. Statistics Netherlands. DOI: 10.13140/RG.2.2.18231.23204.

Geofabrik (2025). *OpenStreetMap data extracts*. URL: https://download.geofabrik.de/.

Gootzen, Y., J. Klingwort, and B. Schouten (2025). *Data quality aspects for location-tracking in smart travel and mobility surveys*. Discussion Paper. Statistics Netherlands. DOI: 10.13140/RG.2.2.33690.76480.

Hasan, R. A., H. Irshaid, F. Alhomaidat, S. Lee, and J.-S. Oh (2022). "Transportation mode detection by using smartphones and smartwatches with machine learning." In: *KSCE Journal of Civil Engineering* 26.8, pp. 3578–3589. DOI: 10.1007/s12205-022-1281-0.

Hijmans, R. J. (2024). *geosphere: spherical trigonometry*. R package version 1.5-20. URL: https://github.com/rspatial/geosphere.

Hu, M., E. R. Melipillán, B. T. West, J. A. Kirlin, and I. Paniagua (2020). "Response patterns in a multi-day diary survey: Implications for adaptive survey design." In: *Survey Research Methods* 14.3, pp. 289–300. DOI: 10.18148/srm/2020.v14i3.7465.

Janitza, S., E. Celik, and A.-L. Boulesteix (2018). "A computationally fast variable importance test for random forests for high-dimensional data." In: *Advances in Data Analysis and Classification* 12.4, pp. 885–915. DOI: 10.1007/s11634-016-0276-4.

Jiang, Z., A. Huang, G. Qi, and W. Guan (2023). "A framework of travel mode identification fusing deep learning and map-matching algorithm." In: *IEEE Transactions on Intelligent Transportation Systems* 24.6, pp. 6401–6415. DOI: 10.1109/TITS.2023.3250660.

Kapoor, S and V Perrone (2021). *A simple and fast baseline for tuning large XGBoost models*. Technical Report. ArXiv. DOI: 10.48550/arXiv.2111.06924.

Kitamura, R. and P. H. Bovy (1987). "Analysis of attrition biases and trip reporting errors for panel data." In: *Transportation Research Part A: General* 21.4-5, pp. 287–302. DOI: 10.1016/0191-2607(87)90051-3.

Klingwort, J., Y. Gootzen, and J. Fourie (2025a). *Development and performance of a transport mode classification algorithm for smart surveys*. Smart Survey Implementation (SSI). Report number: WP3: Developing Smart Data Microservices. DOI: 10.13140/RG.2.2.30160.83203.

Klingwort, J., Y. Gootzen, M. Kompier, and V. Toepoel (2025b). *How smart are smart travel surveys? Evaluating trip segmentation, travel motive, and travel mode predictions*. Conference paper. Mobile Apps and Sensors in Surveys (MASS).

Klingwort, J., Y. Gootzen, D. Remmerswaal, and B. Schouten (2025c). "Algorithms versus survey response: Comparing a smart travel and mobility survey with a web diary." In: *Transportation Research Interdisciplinary Perspectives* 31, p. 101436. DOI: 10.1016/j.trip.2025.101436.

Klingwort, J., B. Schouten, D. Remmerswaal, D. McCool, Y. Gootzen, J. de Groot, P. Lugtig, and M. Schulte (2026). *Eight years of developing a general population smart travel survey. Lessons learned from two large field tests*. Discussion Paper. Statistics Netherlands.

Kuhn, M. (2008). "Building predictive models in R using the caret package." In: *Journal of Statistical Software* 28.5, pp. 1–26. DOI: 10.18637/jss.v028.i05.

Li, J., X. Pei, X. Wang, D. Yao, Y. Zhang, and Y. Yue (2021). "Transportation mode identification with GPS trajectory data and GIS information." In: *Tsinghua Science and Technology* 26.4, pp. 403–416. DOI: 10.26599/TST.2020.9010014.

Lindner, T., J. Puck, and A. Verbeke (2022). "Beyond addressing multicollinearity: Robust quantitative analysis and machine learning in international business research." In: *Journal of International Business Studies* 53.7, pp. 1307–1314. DOI: 10.1057/s41267-022-00549-z.

Liu, H. and I. Lee (2017). "End-to-end trajectory transportation mode classification using Bi-LSTM recurrent neural network." In: *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pp. 1–5. DOI: 10.1109/ISKE.2017.8258799.

Liu, Y., E. Miller, and K. N. Habib (2022). "Detecting transportation modes using smartphone data and GIS information: Evaluating alternative algorithms for an integrated smartphone-based

travel diary imputation." In: *Transportation Letters* 14.9, pp. 933–943. DOI: 10.1080/19427867.2021.1958591.

Lu, Z., Z. Long, J. Xia, and C. An (2019). "A random forest model for travel mode identification based on mobile phone signaling data." In: *Sustainability* 11.21, p. 5950. DOI: 10.3390/su11215950.

Luque, A., A. Carrasco, A. Martín, and A. de Las Heras (2019). "The impact of class imbalance in classification performance metrics based on the binary confusion matrix." In: *Pattern Recognition* 91, pp. 216–231. DOI: 10.1016/j.patcog.2019.02.023.

McCool, D., P. Lugtig, O. Mussmann, and B. Schouten (2021). "An app-assisted travel survey in official statistics: Possibilities and challenges." In: *Journal of Official Statistics* 37.1, pp. 149–170. DOI: 10.2478/jos-2021-0007.

Nahmias-Biran, B.-h., Y. Han, S. Bekhor, F. Zhao, C. Zegras, and M. Ben-Akiva (2018). "Enriching activity-based models using smartphone-based travel surveys." In: *Transportation Research Record* 2672.42, pp. 280–291. DOI: 10.1177/0361198118798475.

Neyra, J., V. Siramshetty, and H. Ashqar (2024). *The effect of different feature selection methods on models created with XGBoost*. DOI: 10.48550/arXiv.2411.05937.

Nour, A., B. Hellinga, and J. Casello (2016). "Classification of automobile and transit trips from smartphone data: Enhancing accuracy using spatial statistics and GIS." In: *Journal of Transport Geography* 51, pp. 36–44. DOI: 10.1016/j.jtrangeo.2015.11.005.

OpenStreetMap Contributors (2025). *OpenStreetMap*. URL: https://www.openstreetmap.org.

Pappu, V. and P. M. Pardalos (2014). "High-dimensional data classification." In: *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*. Ed. by F. Aleskerov, B. Goldengorin, and P. M. Pardalos. New York, NY: Springer New York, pp. 119–150. ISBN: 978-1-4939-0742-7. DOI: 10.1007/978-1-4939-0742-7_8.

Prelipcean, A. C., Y. O. Susilo, and G. Gidófalvi (2018). "Collecting travel diaries: Current state of the art, best practices, and future research directions." In: *Transportation Research Procedia* 32, pp. 155–166. DOI: 10.1016/j.trpro.2018.10.029.

Puts, M. J. H. and P. J. H. Daas (2021). *Unbiased Estimations based on Binary Classifiers: A Maximum Likelihood Approach*. arXiv: 2102.08659 [stat.ML]. URL: https://arxiv.org/abs/2102.08659.

Qin, Y., H. Luo, F. Zhao, C. Wang, J. Wang, and Y. Zhang (2019). "Toward transportation mode recognition using deep convolutional and long short-term memory recurrent neural networks." In: *IEEE Access*, pp. 142353–142367. DOI: 10.1109/ACCESS.2019.2944686.

R Core Team (2024). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. URL: https://www.R-project.org/.

Remmerswaal, D., B. Schouten, J. Bakker, J. van den Heuvel, and J. Klingwort (2025). *A smart Travel Survey: What is the role of the respondent?* Discussion Paper. Statistics Netherlands. DOI: 10.13140/RG.2.2.16084.31364.

Sadeghian, P., J. Håkansson, and X. Zhao (2021). "Review and evaluation of methods in transport mode detection based on GPS tracking data." In: *Journal of Traffic and Transportation Engineering (English Edition)* 8.4, pp. 467–482. DOI: 10.1016/j.jtte.2021.04.004.

Sadeghian, P., X. Zhao, A. Golshan, and J. Håkansson (2022). "A stepwise methodology for transport mode detection in GPS tracking data." In: *Travel Behaviour and Society* 26, pp. 159–167. DOI: 10.1016/j.tbs.2021.10.004.

Safi, H., B. Assemi, M. Mesbah, and L. Ferreira (2016). "Trip detection with smartphone-assisted collection of travel data." In: *Transportation Research Record* 2594.1, pp. 18–26. DOI: 10.3141/2594-03.

Schouten, B., D. Remmerswaal, A. Elevelt, J. Groot, J. Klingwort, T. Schijvenaars, M. Schulte, and M. Vollebregt (2024). *A smart travel survey: Results of a push-to-smart field experiment in the Netherlands*. Discussion Paper. Statistics Netherlands. DOI: 10.13140/RG.2.2.30248.38404.

Shafique, M. A. and E. Hato (2016). “Travel mode detection with varying smartphone data collection frequencies.” In: *Sensors* 16.5, p. 716. DOI: 10.3390/s16050716.

Smeets, L., P. Lugtig, and B. Schouten (2019). *Automatic travel mode prediction in a national travel survey*. Discussion Paper. Statistics Netherlands. URL: https://www.cbs.nl/-/media/_pdf/2019/51/dp%20smeets-lugtig-schouten%20-%20vervoermiddelpredictie.pdf.

Stenneth, L., O. Wolfson, P. S. Yu, and B. Xu (2011). “Transportation mode detection using mobile phones and GIS information.” In: *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. GIS ’11. Chicago, Illinois: Association for Computing Machinery, pp. 54–63. ISBN: 9781450310314. DOI: 10.1145/2093973.2093982.

Stopher, P. R. and S. P. Greaves (2007). “Household travel surveys: Where are we going?” In: *Transportation Research Part A: Policy and Practice* 41.5, pp. 367–381. DOI: 10.1007/s11116-007-9126-8.

Wang, K., Y. Liu, S. Hossain, and K. Nurul Habib (2024). “Who drops off web-based travel surveys? Investigating the impact of respondents dropping out of travel diaries during online travel surveys.” In: *Transportation*, pp. 1–37. DOI: 10.1007/s11116-024-10510-8.

Wright, M. N. and A. Ziegler (2017). “ranger: A fast implementation of random forests for high dimensional data in C++ and R.” In: *Journal of Statistical Software* 77.1, pp. 1–17. DOI: 10.18637/jss.v077.i01.

Xiao, G., Q. Cheng, and C. Zhang (2019). “Detecting travel modes using rule-based classification system and Gaussian process classifier.” In: *IEEE Access* 7, pp. 116741–116752. DOI: 10.1109/ACCESS.2019.2936443.

Xiao, G., Z. Juan, and C. Zhang (2015). “Travel mode detection based on GPS track data and Bayesian networks.” In: *Computers, Environment and Urban Systems* 54, pp. 14–22. DOI: 10.1016/j.compenvurbsys.2015.05.005.

Xiao, Y. et al. (2012). “Transportation activity analysis using smartphones.” In: *2012 IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 60–61. DOI: 10.1109/CCNC.2012.6181051.

Xiao, Z., Y. Wang, K. Fu, and F. Wu (2017). “Identifying different transportation modes from trajectory data using tree-based ensemble classifiers.” In: *ISPRS International Journal of Geo-Information* 6.2, p. 57. DOI: 10.3390/ijgi6020057.

Yang, N., C. Al Haddad, I. Yamnenko, and C. Antoniou (2024). *Machine learning for data-centric transport mode detection: A systematic review*. Available at SSRN. DOI: 10.2139/ssrn.4960556.

Yang, X., K. Stewart, L. Tang, Z. Xie, and Q. Li (2018). “A review of GPS trajectories classification based on transportation mode.” In: *Sensors* 18.11, p. 3741. DOI: 10.3390/s18113741.

Yu, J. J. Q. (2021). “Travel mode identification with GPS trajectories using wavelet transform and deep learning.” In: *IEEE Transactions on Intelligent Transportation Systems* 22.2, pp. 1093–1103. DOI: 10.1109/TITS.2019.2962741.

Zahroh, S., P. Lugtig, Y. Gootzen, J. Klingwort, and B. Schouten (2025). *Predicting trip purpose in a smartphone-based travel survey*. Discussion Paper. Statistics Netherlands. DOI: 10.13140/RG.2.2.26114.80329.

Zhang, C., Y. Zhu, C. Markos, S. Yu, and J. J. Yu (2021). “Toward crowdsourced transportation mode identification: A semisupervised federated learning approach.” In: *IEEE Internet of Things Journal* 9.14, pp. 11868–11882. DOI: 10.1109/JIOT.2021.3132056.

Zhao, X., X. Yan, A. Yu, and P. Van Hentenryck (2020). “Prediction and behavioral analysis of travel mode choice: A comparison of machine learning and logit models.” In: *Travel behaviour and society* 20, pp. 22–35. DOI: 10.1016/j.tbs.2020.02.003.

Zhou, Y., Y. Zhang, Q. Yuan, C. Yang, T. Guo, and Y. Wang (2022). “The smartphone-based person travel survey system: Data collection, trip extraction, and travel mode detection.” In: *IEEE*

Transactions on Intelligent Transportation Systems 23.12, pp. 23399–23407. DOI: 10.1109/TITS.2022.3207198.

Appendices

A Feature List

A.1 GPS-based Features

- Speed
  - Summary statistics: mean, median, minimum, maximum, skewness, kurtosis, interquartile range, standard deviation, coefficient of variation, and autocorrelation coefficient
  - Speed values at nth percentiles: 1, 5, 10, 90, 95, and 99
  - Outlier summaries: proportion of outliers (>1.5 IQR above the third quantile), and mean outlier value
  - Proportion of speed values within specified intervals: 0 - 3 km/h, 3 - 8 km/h, 8 - 15 km/h, 15 - 30 km/h, 30 - 50 km/h, 50 - 80 km/h, 80 - 120 km/h, and >120 km/h
- Acceleration, jerk, and snap
  - Summary statistics: mean, median, minimum, maximum, skewness, kurtosis, interquartile range, standard deviation, coefficient of variation, and autocorrelation coefficient
  - Values at nth percentiles: 1, 5, 10, 90, 95, and 99
  - Outlier summaries: proportion of high outliers (>1.5 IQR above the third quantile), mean high outlier value, proportion of low outliers (<1.5 IQR below the first quantile), and mean low outlier value
- Difference in bearing between consecutive points
  - Summary statistics: mean, median, minimum, maximum, skewness, kurtosis, interquartile range, standard deviation, coefficient of variation, and autocorrelation coefficient
  - Values at nth percentiles: 1, 5, 10, 90, 95, and 99
  - Proportion of bearing difference values within specified intervals: 0 - 5, 5 - 10, 10 - 15, 15 - 30, 30 - 45, 45 - 90, and >90
- Total distance
  - Total distance traveled in km
- Trip duration
  - Trip duration in minutes
- Accuracy
  - Summary statistics: mean, median, minimum, maximum, skewness, kurtosis, interquartile range, standard deviation, coefficient of variation, and autocorrelation coefficient
  - Speed values at nth percentiles: 1, 5, 10, 90, 95, and 99
- Altitude
  - Summary statistics: mean, median, minimum, maximum, skewness, kurtosis, interquartile range, standard deviation, coefficient of variation, and autocorrelation coefficient
  - Values at nth percentiles: 1, 5, 10, 90, 95, and 99

A.2 OSM-based Features

- Route and public transport station counts (based on a 20-meter buffer)
  - Number of overlaps with bike, bus, metro, train, and tram routes
  - Number of overlaps with bike, bus, metro, train, and tram routes, normalized by track size
  - Number of encountered bus, metro, train, and tram stations

    - Number of encountered bus, metro, train, and tram stations, normalized by track size
- Route and public transport station proximity
    - Minimum, maximum, mean, and standard deviation of the distance to bike, bus, metro, train, and tram routes
    - Distance to nearest bus, metro, train, and tram station at the beginning and end of a trip, taking the average of the first and last five GPS measurements with an accuracy below 20 meters (if all GPS measurements have an accuracy greater than 20, the measurement with the lowest accuracy from the first and last 10 observations is used)
- Proximity to parking amenities (based on a 20-meter buffer)
    - Distance to nearest parking place, parking entrance, charging station, bike parking, and bike rental at the beginning and end of a trip, taking the average of the first and last five GPS measurements with an accuracy below 20 meters (if all GPS measurements have an accuracy greater than 20, the measurement with the lowest accuracy from the first and last 10 observations is used)
- Traffic indicators (based on a 10-meter buffer)
    - Number of encountered traffic signals, street lights, speed cameras, roundabouts, motorway junctions, stops, and turning circles
    - Number of encountered traffic signals, street lights, speed cameras, roundabouts, motorway junctions, stops, and turning circles, normalized by track size

For more extensive information on the computation of these OSM-based features, see Fourie et al. (2025).

A.3 Temporal Features

- Information from the previous stop
    - Binary indicator if the previous event was a stop
    - Duration of the previous stop in minutes
    - One-hot encoded previous stop purpose (education, home, other, paid work, pickup goods, pickup people, shopping, sport, transfer, unpaid work, visit)
- Information from previous tracks
    - Trip duration of the last three tracks (lag 1, lag 2, and lag 3)
    - One-hot encoded transport mode of the last three tracks (lag 1, lag 2, and lag 3)
- Temporal context
    - nth daily trip
    - daily cumulative duration of trips
    - daily cumulative duration of distance traveled
    - Binary indicator if the trip fell (partially) within morning rush hour and/or evening rush hour (morning and evening rush hours as defined by the Dutch ANWB)
    - Proportion of the trip duration that falls within the morning or evening rush hour
    - One-hot encoded day of the week
- Respondent-specific baselines (cumulative means)
    - Respondent-specific cumulative mean for the following GPS-based features: 90th and 95th speed percentiles, median speed, mean speed, interquartile range speed, trip duration in minutes, and total duration in km
    - Difference between the current trip and the respondent-based cumulative means for the previously mentioned GPS-based features
    - Mode-specific cumulative means: mean walking speed, mean bike speed, and mean car speed (mode-specific cumulative means are only updated when respondents make and label these trips)
    - Difference between the current trip and the respondent-based mode-specific cumulative means

B Additional Tables With Evaluation Metrics

Table B.1 F1 scores by class for the eight machine learning models evaluated on the internal test data

	Algorithm	Features	Bike	Bus	Car	Metro	Train	Tram	Walk
1	RF	GPS	.82	.24	.71	.29	.59	.02	.88
2	RF	GPS + OSM	.83	.69	.90	.65	.86	.47	.88
3	RF	GPS + temporal	.83	.50	.88	.25	.64	.05	.88
4	RF	all	.83	.83	.92	.67	.86	.50	.88
5	XGB	GPS	.84	.31	.84	.44	.63	.06	.89
6	XGB	GPS + OSM	.84	.69	.92	.71	.87	.33	.90
7	XGB	GPS + temporal	.89	.55	.90	.43	.67	.30	.90
8	XGB	all	.90	.79	.94	.71	.85	.75	.90

Table B.2 F1 scores by class for the eight machine learning models evaluated on the external validation data

	Algorithm	Features	Bike	Bus	Car	Ferry	Metro	Train	Tram	Walk
1	RF	GPS	.44	.21	-	-	.17	.63	0	.88
2	RF	GPS + OSM	.48	.73	-	-	.42	.75	.78	.84
3	RF	GPS + temporal	.45	.19	-	-	.18	.63	0	.89
4	RF	all	.48	.73	-	-	.42	.82	.77	.84
5	XGB	GPS	.60	.14	-	-	0	.63	0	.95
6	XGB	GPS + OSM	.69	.62	-	-	.50	.89	.81	.83
7	XGB	GPS + temporal	.59	.18	-	-	0	.70	.22	.95
8	XGB	all	.72	.60	-	-	.89	.90	.79	.90

Colophon

Publisher
Statistics Netherlands
Henri Faasdreef 312, 2492 JP The Hague, Netherlands
www.cbs.nl

Prepress
Statistics Netherlands Grafimedia

Design
Edenspiekermann

Information
Telephone +31 88 570 70 70
Via contact form: www.cbs.nl/information