information

Article

Improved Carpooling Experience through Improved GPS Trajectory Classification Using Machine Learning Algorithms

Manish Kumar Pandey [1,2], Anu Saini [3], Karthikeyan Subbiah [2], Nalini Chintalapudi [4] and Gopi Battineni [5,*]

1 Centre for Quantitative Economics and Data Science, Birla Institute of Technology, Mesra, Ranchi 835215, Jharkhand, India; pandey.manish@live.com
2 Department of Computer Science, Institute of Science, Banaras Hindu University, Varanasi 221005, Uttar Pradesh, India; karthinikita@gmail.com
3 Department of Computer Science, G. B. Pant DSEU Okhla-1 Campus, New Delhi 110020, Delhi, India; anuanu16@gmail.com
4 Informatics Centre, School of Science and Technologies, University of Camerino, 62032 Camerino, Italy; nalini.chintalapudi@unicam.it
5 Clinical Research Centre, School of Medicinal and Health Products Science, University of Camerino, 62032 Camerino, Italy
* Correspondence: gopi.battineni@unicam.it; Tel.: +39-3331728206

Abstract: Globally, smart cities, infrastructure, and transportation have led to a rise in vehicle numbers, resulting in an increasing number of problems. This includes problems such as air pollution, noise pollution, high energy consumption, and people's health. A viable solution to these problems is carpooling, which involves sharing vehicles between people going to the same location. As carpooling solutions become more popular, they need to be implemented efficiently. Data analytics can help people make informed decisions when selecting a ride (Car or Bus). We applied machine learning algorithms to select the desired ride (Car or Bus) and used feature ranking algorithms to identify the foremost traits for selecting the desired ride. Based on the performance evaluation metric, 11 classifiers were used for the experiment. In terms of selecting the desired ride, Random Forest performs best. Using ten-fold cross-validation, we obtained a sensitivity of 87.4%, a specificity of 73.7%, an accuracy of 81.0%, a sensitivity of 90.8%, a specificity of 77.6%, and an accuracy of 84.7% using leave-one-out cross-validation. To identify the most favorable characteristics of the Ride (Car or Bus), the recursive elimination of features algorithm was applied. By identifying the factors contributing to users' experience, the service providers will be able to rectify those factors to increase business. It has been determined that the weather can make or break the user experience. This model will be used to quantify and map intrinsic and extrinsic sentiments of the people and their interactions with locality, socio-economic conditions, climate, and environment.

Keywords: carpooling; GPS Trajectory; SMAC; Random Forest; feature ranking; SDG 9

check for updates

Citation: Pandey, M.K.; Saini, A.; Subbiah, K.; Chintalapudi, N.; Battineni, G. Improved Carpooling Experience through Improved GPS Trajectory Classification Using Machine Learning Algorithms. *Information* **2022**, *13*, 369. https://doi.org/10.3390/info13080369

Academic Editor: Willy Susilo

Received: 23 June 2022
Accepted: 2 August 2022
Published: 3 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

By combining Social, Mobile, Analytics, and Cloud (SMAC) technologies, the third computing paradigm is launched, resulting in the massive growth of multifarious data [1,2]. The data are generated by a variety of applications, including sensors, mobility, social apps, and many others. Due to the convergence of technologies, the Internet of Everything (IoE) has gained unprecedented connectivity, where everyone is connected to everything [3].

IoE has been identified as an opportunity by the United Nations (UN) and given the Sustainable Development Goal (SDG) of Industry, Innovation, and Infrastructure that can be realized with efficient utilization of IoE. As part of this goal, resilient infrastructure will be built, sustainable industrialization will be encouraged, and innovation will be fostered. In addition to smart cities and infrastructure, IoE is used in transportation and infrastructure. Some of these developments contribute to achieving the Sustainable Development Goals,

but they also create new challenges. With the increasing number of motor vehicles in cities around the world, motor vehicle-related problems are also on the rise. People's health is also affected by pollution, urban mobility, and urban congestion. Several green initiatives have been taken by most authorities in response to this situation, such as bicycles as a mode of transportation, underpasses, and ridesharing services. One such area, called carpooling or ridesharing, has been the focus of the current work, which is a boon for cities worldwide as their population grows. In the proposed approach, users' experiences are brought together to provide better services. A method is being developed to understand the intricacies of the user's experience, including the crowding of shared rides, the performance on the given day, and the duration of rides. The benefits of ridesharing include improving health, saving the environment, and speeding up traffic commutes. Although ridesharing offers such benefits, people are not enthusiastic about using it as a daily mode of transportation. Considering the traffic woes, particularly for working professionals, adopting environment-friendly and faster commutes via Ridesharing can benefit the administration and citizens alike. There is no doubt that COVID-19 has hampered people's mindset in preferring local confinements when traveling. In addition, there are various socioeconomic, environmental, geographic, and emotional factors that hinder people from choosing cycling as their primary mode of transportation. Using their ratings, the current work seeks to understand the factors that limit the use of ridesharing among people and prescribe ways to promote it. It would not only be a one-time investment, but it would also assist policymakers in framing informed regulations.

While Intelligent Transportation Systems (ITS) are advancing rapidly, the potential for improving the performance of the transportation system still needs to be investigated thoroughly. An understanding of the relationship between the performance of the transportation system and the distinctiveness of travel demand is necessary for this. The practice of carpooling or ridesharing involves sharing a vehicle for a common trip. All of the software works by searching for people who are on the same or similar trajectory as the user and can offer them a ride. The lack of familiarity between passengers and drivers calls for a ride-matching procedure that could resolve this problem. The performance of mining these rides based on trajectories can be used to select effective rides. There are many related works reported around the world, but only a few are listed in the related works section.

Related Work

Carpooling solutions are becoming increasingly popular, but their implementation poses a problem. The datasets should be analyzed efficiently to help riders choose the desired ride (Car or Bus) in an informed manner. The term ridesharing refers to a group of individuals using a car or other vehicle to travel to the same or similar location [4]. Ridesharing can be casual, real-time, or social network-assisted. Using the friend list of social networking sites, the third variant finds potential riders. By doing so, trust can be built and safety can be estimated [4]. Mobile applications such as GO! [5], Uber [6], and BlaBlacar [7] are examples of real-time ridesharing, as they do not require travelers to know each other. Carpooling applications have the advantage of working like a taxi, preserving the concept of a common trip while preserving the character of the service. In [8], a comprehensive review of car sharing services is given, along with their offerings and challenges.

In major cities across the world, traffic congestion is a major concern [9]. The known associated problems of this concern are damage to the environment, economic loss, and health issues [10–12]. In most of these major cities, the road network is not compatible with the growth of vehicles. As a result of delivery delays, these congestion issues sometimes cause significant business losses in several countries [13]. Private vehicle owners were restricted by the Chinese authorities in their latest initiative. Despite an odd-even scheme implemented by the New Delhi government, traffic is still deadly [14]. These congestion issues cause a huge business loss in several countries [15].

M. O. Cruz et al. [15] has worked in grouping similar trajectories for carpooling purpose. The congestion in major cities is a disturbing truth [16]. The indirect burden on society is visible. One example is the increase in the cost of consumer goods caused due to delays in delivery in the supply chain sector. Ridesharing seems to be a feasible solution [16], Ref. [17] taking into consideration the issue of occupancy rate which is pretty low and is evaluated as a fraction of performance shown in transport (passenger-kilometers) and the provided vehicle kilometers which again is a unit that measures the movement of a vehicle over one kilometer. With the trajectory limited to known ones, the goal of ridesharing or carpooling could be achieved which is increasing the value of the occupancy rate. One efficient way of increasing the occupancy rate is by filling up the empty seats in vehicles [18–21].

The basic requirement on which all this software works is that the search needs to be done for people who can offer them the ride if they are moving on the same or similar trajectory. Unfamiliarity between passengers and drivers is a difficulty that calls for a ride-matching procedure that could deal with this issue. Effective rides can be chosen based on the performance evaluation of mining of these rides based on trajectories.

2. Dataset and Methods

The GPS Trajectory dataset was used for experimentation. This dataset is available in the well-recognized UCI (University of California and Irvine) data repository [22]. We extracted the dataset from a mobile application called Go! Track from Google Play Store. The dataset contains 163 instances, each containing details about the ride and its associated features. Table 1 provides statistics about the dataset.

Table 1. GPS Trajectory dataset description.

Attribute Type	Numerical
Number of attributes	15
Number of instances	163
Number of classes	2

2.1. Selection of Input Features Vector

The dataset contains 15 attributes and one class label for each instance (indicating whether it belongs to a Car or Bus). For trajectories with only cars or buses, six attributes were used. For the selection of rides, a binary classifier was used with a class variable (1 indicates Car, 2 indicates Bus). Figure 1 shows the distribution of various attributes from the original dataset, while Table 2 shows details about the feature vectors.

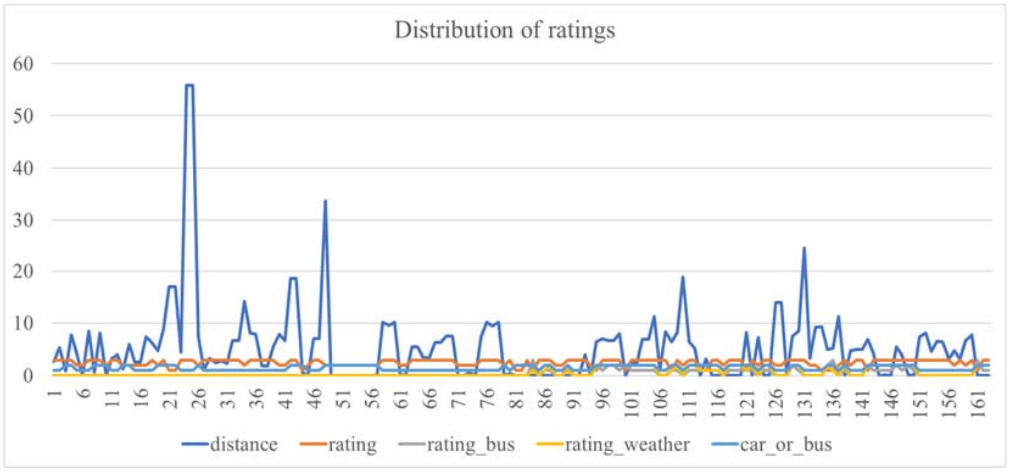


Figure 1. Distribution of attributes between the two classes as Car and Bus.

Table 2. Feature description of the dataset.

Features	Descriptor
d_android	Devices to capture the instances
speed	Speed in km/H is captured
distance	Total distance in km is captured
rating	This is the evaluation parameter of the user's experience in terms of good (2), normal (1), and bad (3).
rating_bus	The evaluation parameter is associated with crowding of the bus, crowded means rating is 1, a little crowded means rating is 2, and not crowded at all is represented with rating 3.
rating_weather	The evaluation parameter is associated with weather; 1 is for sunny and 2 for rainy conditions.
car_or_bus	The overall experience of choosing a Car (2) or Bus (1).

2.2. Proposed Methodology

In the classification protocol section, 11 machine learning (ML) algorithms are described. Table 3 presents a brief description of each algorithm and Figure 2 illustrates the proposed methodology.

Table 3. Brief description of adopted individual algorithms.

N	Algorithm	Definition
1.	Random Tree	A tree is built by considering K randomly chosen attributes at each node without pruning. This permits the assessment of class probabilities based on the training and testing set.
2.	Multi-Layer Perception (MLP)	Instances would be classified in a backpropagation manner.
3.	Polykernel sequential minimal optimization (SMO)	This algorithm trains the support vector classifiers [23,24]
4.	Instance-based learning with k-parameter (IBK)	K-nearest neighbor's classifier picks the most suitable K based on the cross-validation technique as well as based on the weights of distance [25,26].
5.	Rotation Forest (RF)	Classification is done using the base learner [27]
6.	Bagging	This is used mainly to reduce the variance along with the classification of the base learners [28].
7.	Random Forest	A forest of random trees is constructed [29].
8.	RealADABoost	Performance is improved using ensemble learning [30].

Our testing of various ML algorithms allowed us to choose the best-fit algorithm. The term Artificial Neural Network (ANN) refers to an amalgamation of models that mimic the brain's neurons from which complex information is processed. Layers make up the overall model, such as input layers for data entry, hidden layers for in-between processing, and output layers for results. Assigning weights at the beginning of the model facilitates communication between layers, which could later be optimized using backpropagation.

The boosting algorithm is an ensemble of decision tree algorithms. Repeatedly fitting several decision trees improves the accuracy of the models. By using the boosting method, the subset of data is selected. Boosting seeks to minimize the loss function expectation by estimating a regression function with random selection. In the updated decision trees,

they account for errors in the previous steps by selecting poorly modeled data from the previous step.

Random Forests (RFs) are forests of trees in which each tree makes an independent prediction, which is then added up to reach a final value. Averaging the values or taking the maximum value is used to accumulate. Therefore, most of the features are covered randomly. RF is again a tree-based ensemble algorithm that transforms the subsets of attributes and then constructs the tree. Two key differences between Rotation Forest and Random Forest are the transformation of subsets into principal components and the use of the C4.5 decision tree. Although RF includes many add-ons, it usually excels in providing better accuracy in large datasets, estimation of prominent features, and most importantly, not overfitting.

It is generally recommended to train support vector machines (SVM) using poly kernel sequential minimal optimization. To do so, the sub-problems are broken down to the smallest level and then analyzed. We intend to identify the associations between ratings and then select rides based on users' experiences, so we need to explore features exhaustively. Past ML algorithms have proven to be effective in solving these types of problems, so they are selected.

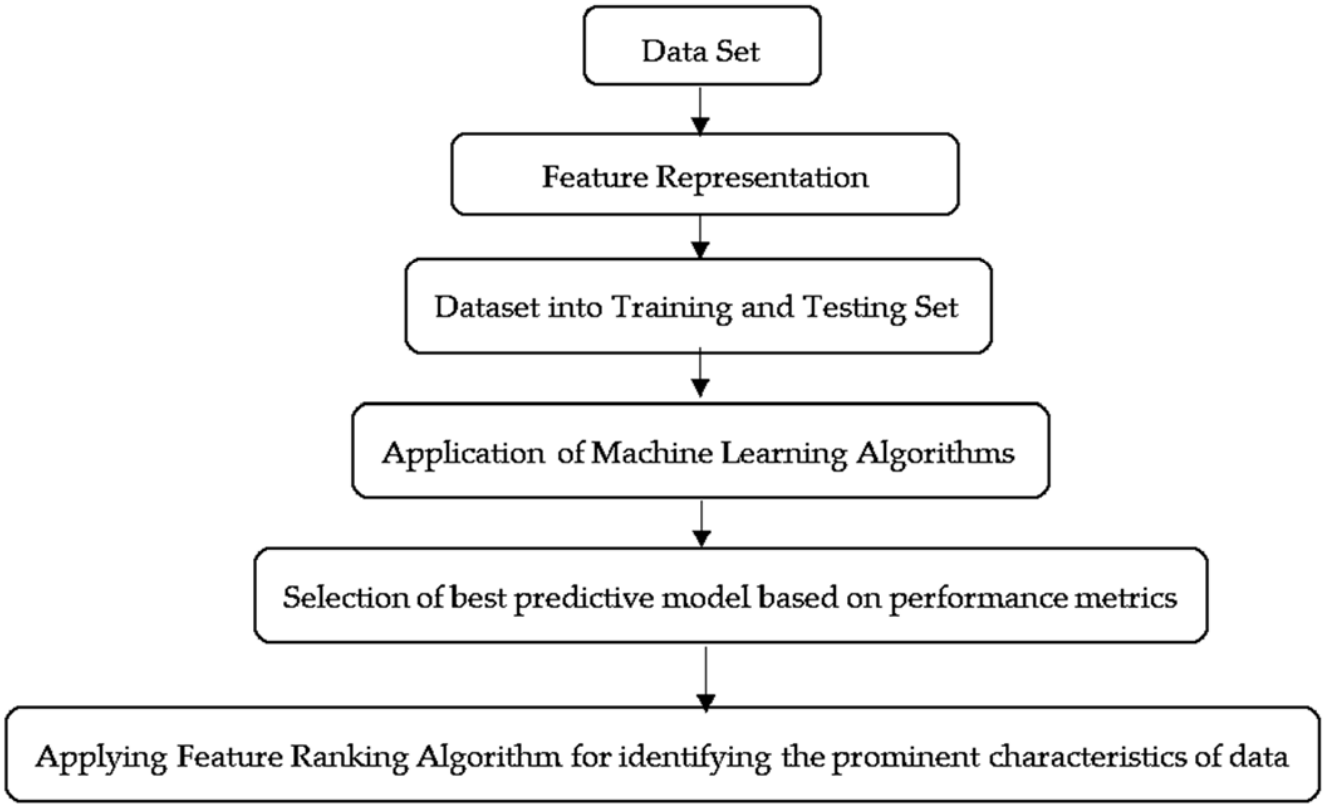


Figure 2. Flow diagram of the proposed methodology.

2.3. Feature Ranking and Reduction Protocol

A feature ranking algorithm that recursively eliminates the features is described by [31]. In the current work, features have been eliminated recursively by identifying their worth in the classification of rides by identifying their importance in the process [26]. Using a repeated sampling of the instances and assigning weights to the rides, the analysis was carried out. Additionally, the weights assigned were used to separate neighboring instances belonging to the same or different rides. In addition, the algorithm rewarded classifiers who predicted values that differed across neighborhoods and penalized classes that predicted values that were the same.

The final set was prepared using these weights as well as the weights that were obtained through threshold value. The algorithm then selected the instances, and they were nearly random (R_d is a randomly sampled instance) on two parameters, one belonging to the same ride and the other belonging to a different ride. These are termed as the nearest hit (H_t) and nearest miss (M_s), respectively. The algorithm was made to update the weights (Wt_X—Weight of x ride) based on the ability of the algorithm to distinguish between these misses. If S_i denotes the total number of randomly sampled instances, then the update of these weights is done as per the formula given in Equation (1).

$$Wt_x = Wt_x - \frac{bal(x, Rd, Ht)^2}{S_i} + \frac{bal(x, Rd, Ms)^2}{S_i} \quad (1)$$

The probability difference [32] of ride x is evaluated using Equation (2).

$$\begin{aligned} Wt_x = P\ (&\text{different value of x}| \text{ nearest instance of different class}) \\ &-P\ (\text{different value of x}| \text{ nearest instance of same class}) \end{aligned} \quad (2)$$

In case, attributes are independent then Equation (2) becomes Equation (3) [32].

$$Relief_x = P(\text{different value of x}| \text{ different class}) - P(\text{different value of x}| \text{ same class}) \quad (3)$$

Equation (3) can be rewritten as Equation (4) if C could be categorized as a class variable,

$$Relief_X = \frac{Gini^i X \sum_{x \in X} p(x)^2}{\left(1 - \sum_{c \in C} p(c)^2\right) \sum_{c \in C} p(c)^2} \quad (4)$$

$$Gini^i = \left[\sum_{c \in C}, p, (c), (, 1, -, p, (c)\right] - \sum_{x \in X}\left(\frac{p(x)^2}{\sum_{x \in X} p(x)^2} \sum_{c \in C} p(c|x)(1 - p(c|x))\right) \quad (5)$$

The information gain during the process is denoted by the value, $Gini^i$ [33] and mathematically could be obtained from Equation (5). The algorithm is described below (Algorithm 1):

Algorithm 1: Recursive Elimination of Features

1: Input: F with feature set (f_1,f_2, … f_p) where an instance X is described by vector with dimension p (x_1,x_2,..x_p), and x_j denotes the feature value f_j of X

2: Output: Relevant Features f_r. Begin with setting the weights (W_j) to 0 followed by a random selection of observations (x_r) iteratively.

3: Evaluate the k-nearest observations (x_q) to the selected observations of the rides followed by an update of their weights.

4: Weights adjustment

- ➢ The weights would be updated as per the below formula in case x_r and x_q correspond to the same rides $W_j^i = W_j^{i-1} - \frac{\Delta_j(x_r, x_q)}{m} d_{rq}$
- ➢ The weights would be updated as per the below formula in case x_r and x_q correspond to different rides $W_j^i = W_j^{i-1} + \frac{P_{y_q}}{1 - P_{y_r}} \frac{\Delta_j(x_r, x_q)}{m} d_{rq}$

5: The value of the j^{th} classifiers about observations x_r and x_q is given by x_{rj} and x_{qj}. The values come out to be

6: For discrete Fj, $\Delta_j(x_r, x_q) = \begin{cases} 0, & x_{rj} = x_{qj} \\ 1, & x_{rj} \neq x_{qj} \end{cases}$

7: For continuous Fj, $\Delta_j(x_r, x_q) = \frac{|x_{rj} - x_{qj}|}{\max(F_j) - \min(F_j)}$

8: If rank(r,q) denotes the state of observation q amid nearest neighbors of observation r and the distance between the neighbors is k, then mathematically the distance function d_{rq} is obtained by $d_{rq} = \frac{d'_{rq}}{\sum_{l=1}^{k} d'_{rl}}$ which is subject to scaling as given by $d'_{rq} = e^{-\left(\frac{rank(r,q)}{sigma}\right)^2}$.

To identify the common traits of the rides and predict the desired ride, we varied the features using classification algorithms. In addition to reducing irrelevant and redundant features, this also improved the accuracy of classification of the desired ride.

2.4. Performance Evaluation Metrics

ML algorithms are evaluated using parameters that are dependent and independent of threshold values. True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FP) are used to calculate the parameters. TP is the number of accurately predicted instances belonging to Ride (Car), FN is the number of inaccurately predicted instances belonging to Ride (Car), TN is the number of accurately predicted instances belonging to Ride (Bus), and FP is the number of inaccurately predicted instances belonging to Ride (Bus).

Sensitivity: This provides the proportion of accurately predicted Ride (Car) instances and is given by $\frac{\text{TP}}{(\text{TP+FN})} \times 100$.

Specificity: This provides the proportion of proper accurately predicted Ride (Bus) instances and is given by $\frac{\text{TN}}{(\text{TN+FP})} \times 100$.

Accuracy: This provides the proportion of accurately predicted Ride (Car) and Ride (Bus) instances and presents as $\frac{\text{TP+TN}}{(\text{TP+FN+TN+FP})} \times 100$.

Area Under Curve (AUC): This is the representation of the zone under the receiver operating characteristic curve (ROC). The zone which is closed to 1 is the one that provides better prediction.

MCC: Mathew's correlation coefficient is used as a performance metric and is obtained using the equation $\frac{\text{TPxTN}-\text{FPxFN}}{\sqrt{(\text{TP+FP})(\text{TP+FN})(\text{TN+FP})(\text{TN+FN})}}$, where an MCC value of 1 is an indicator of the best predictor.

3. Results

3.1. Model Performance Evaluation

A 10-fold and a leave-one-out cross-validation (LOOCV) experiment was conducted on the training set. Performance evaluation metrics showed that Random Forests perform better than linear models. Outcomes mentioned that RF achieved a sensitivity of 90.8%, a specificity of 77.6%, and an accuracy of 84.7%, whereas LOOCV achieved a sensitivity of 90.8%, a specificity of 77.6%, and an accuracy of 84.7%. Table 3 summarizes experimental results with given performance metrics. We evaluated the model's performance using an independent and threshold-independent statistical test, Area Under the Receiving Operator Curve (AUROC).

An ROC graph is a visualization of how well the classifiers perform. Based on validations using 10-fold and LOOCV methods, Figure 3 shows ROC curves for different models. RF outperforms the others with 89.3% (with 10-fold CV) and 92.4% (LOOCV) of accuracies. Using this classifier, new rides can be classified instantly and accurately with over 80% accuracy. Table 4 shows that apart from Random Forest, there is a greater difference between sensitivity and specificity. For 10-fold, MLP has a higher MCC value, but for LOOCV, Random Forest has a higher value. Similarly, MLP is more accurate than 10-fold, but the AUC value is more like Random Forest, indicating that Random Forest performs better.

Table 4. Performance metrics for the classifiers on the training set.

Machine Learning Algorithms	10 Folds					LOOCV				
	Sensitivity	Specificity	Accuracy	AUC	MCC	Sensitivity	Specificity	Accuracy	AUC	MCC
MLP	89.7	80.3	85.3	0.873	0.705	89.7	78.9	84.7	0.874	0.693
SMO-PUK	94.3	69.7	82.8	0.820	0.667	94.3	68.4	82.2	0.813	0.656
IBK	82.8	81.6	82.2	0.815	0.643	82.8	82.9	82.8	0.834	0.656
Rotation Forest	92.0	73.7	83.4	0.876	0.672	92.0	72.4	82.8	0.901	0.661
Bagging	92.0	71.1	82.2	0.876	0.650	94.3	72.4	84.0	0.898	0.689
Random Forest	87.4	73.7	81.0	0.893	0.619	90.8	77.6	84.7	0.924	0.694
Random Tree	79.3	75.0	77.3	0.772	0.544	81.6	81.6	81.6	0.816	0.631
RealADABoost-Decision Stump	83.9	78.9	81.6	0.881	0.630	83.9	80.3	82.2	0.869	0.642
RealADABoost-Random Tree	85.1	73.7	79.8	0.850	0.593	86.2	80.3	83.4	0.882	0.667
RealADABoost- RepTree	85.1	75.0	80.4	0.887	0.605	86.2	76.3	81.6	0.919	0.630
RealADABoost- Random Forest	86.2	75.0	81.0	0.894	0.618	86.2	82.9	84.7	0.904	0.692

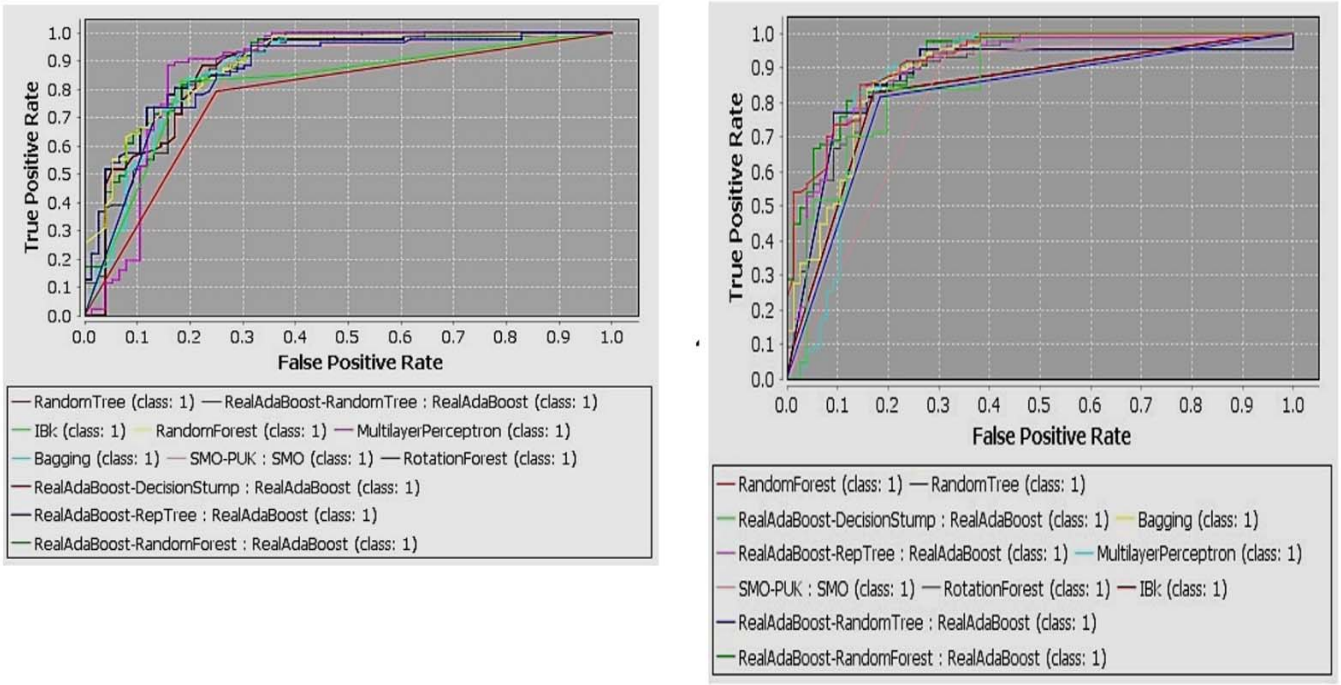


Figure 3. ROC for classifiers trained using 10-fold CV (**left**) and LOOCV (**right**).

3.2. Preventive Analytics through Feature Ranking

Previously recursive elimination of features has been successfully applied for feature ranking [31–34]. In the present work, the same was used to identify prominent features to predict desired rides. Figure 4 shows the rank of the features of the rides and how they are represented as heat maps.

The three most discriminating features for identifying favorite rides were rating_weather (i.e., an evaluation parameter) and rating_bus (i.e., rating 1, if the bus crowd is less, 2 if there is no bus crowd, and 3 if the bus is crowded), and rating evaluates traffic in the form of the user's experience with traffic. A good experience will receive three stars, a normal experience will receive two stars, and a bad experience will receive one star.

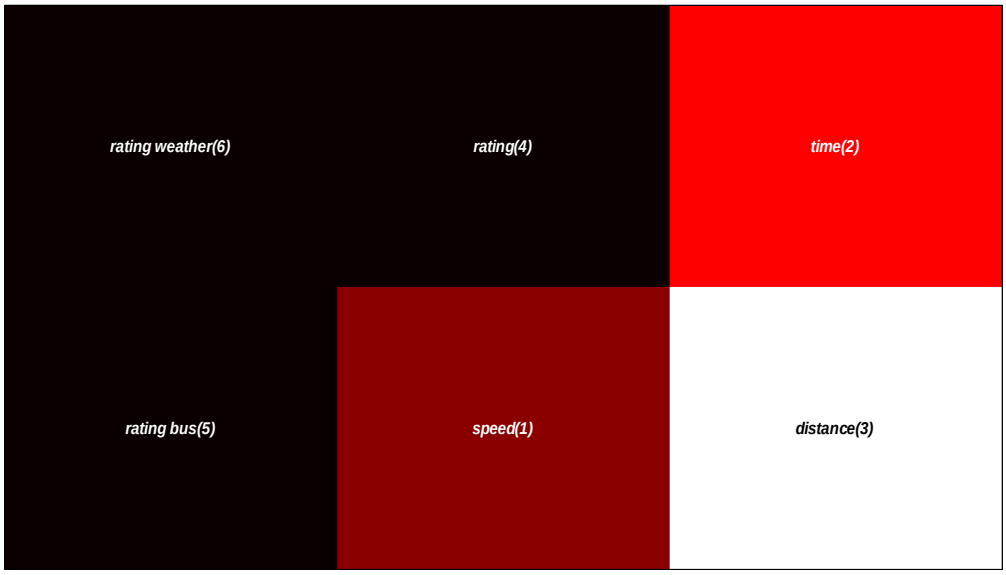


Figure 4. Heatmap representation of feature ranking in discriminating two classes.

4. Discussion

Though Intelligent Transportation System (ITS) technologies have advanced significantly, there is still a need to investigate the prospects of improving the system's performance. This requires a proper understanding of the relationship between transportation systems and travel demand.

SMAC generates huge amounts of data from sensors of Social Internet of Vehicles (SIoV), GPS, etc. To understand insights from this enormous data, it must be efficiently implemented and processed. Carpooling is one such area where efficient analytics of datasets will assist riders in selecting the appropriate ride. Only a few of the features might be important to the true target concept where more deviate from the true representation.

As a result of identifying and reducing features, the computational complexity was reduced, as well as the desired features being obtained, resulting in efficient classification. Financial domains have been quite successful with this approach. Many of the newer domains such as hyperspectral data analytics and sensory applications are huge generators of data which could use the current approach. Identifying the most important features for building the most effective ride selection model is the key takeaway from the proposed approach.

The ultimate goal is not to reach the most accurate model. Identifying and selecting prominent features and inadequate learning are challenges that classifier usually face to perform efficiently. By identifying core features that are crucial to providing inputs in choosing the best ride, the current work offers a solution to these challenges. Using the Recursive Elimination of Features algorithm, this was achieved.

Several ML algorithms based on performance evaluation metrics are used to test the efficacy of the algorithms. It has been demonstrated that feature ranking and selection techniques are important for achieving higher sensitivity values using ML algorithms. A more effective ride selection solution could be offered by taking into account the hyper-local parameters in many scenarios. Hyper-local parameters would also contribute to building brand value and therefore scalability in terms of customer acceptance and promotion.

One of the limitations of the present work is that many of the hyper-local parameters are not available. It is very difficult to provide a solution offered at the root level due to the limited number of datasets available. The parameters involving privacy make learning an improvised feature representation challenging. These hyper-local features will be explored in the future based on variational changes in ranking as well as embedding multiple features.

5. Conclusions

A predictive method to identify the most vital characteristics of the desired ride was proposed using ML algorithms and feature ranking algorithms, and the experiments were conducted using GPS trajectory data. Our first step was to conduct experiments on eleven different classifiers and select the best performing Random Forest. Recursive elimination of features was used to discard redundant and less informative features to identify the most relevant features that predict the most prominent features of the desired ride. To classify the desired ride in an informed way and to avoid future inconvenience, these features can be used in the probabilistic reasoning process.

Policymakers will be able to draft effective regulations against the impact of urbanization based on the deliverables of the current work. It was possible to understand the variability of parameters in detail during the study, enabling the researchers to better understand the complex urban network and associated sentiments. Various scenarios related to environmental impact assessment could be planned and analyzed in the prone areas using the deliverables. This study would also provide a benchmark model for quantification, and mapping of Intrinsic and Extrinsic Sentiments and their associated impact on climate and environment, as well as their adaptability to changing scenarios.

Author Contributions: Data curation, M.K.P. and A.S.; Formal analysis, M.K.P. , K.S. and G.B.; Funding acquisition, M.K.P. and N.C.; Investigation, A.S. and M.K.P. ; Methodology, M.K.P., N.C. and G.B. ; Supervision, K.S.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are available at the well-recognized data repository of the UCI.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pandey, M.K.; Subbiah, K. Performance analysis of time series forecasting of ebola casualties using machine learning algorithm. *Proc. ITISE* **2017**, *2*, 885–898.
2. Pandey, M.K. Novel Application Oriented Problem Solving Approaches in SMAC. Banaras Hindu University. 2017. Available online: http://hdl.handle.net/10603/268444 (accessed on 18 May 2022).
3. Pandey, M.K.; Srivastava, P.K. A Probe into Performance Analysis of Real-Time Forecasting of Endemic Infectious Diseases Using Machine Learning and Deep Learning Algorithms. In *Advanced Prognostic Predictive Modelling in Healthcare Data Analytics*; Springer: Berlin/Heidelberg, Germany, 2021; Volume 64, pp. 241–265. [CrossRef]
4. Chan, N.D.; Shaheen, S.A. Ridesharing in North America: Past, Present, and Future. *Transp. Rev.* **2012**, *32*, 93–112. [CrossRef]
5. Cruz, M.; Macedo, H.; Mendonça, E.; Guimarães, A. GO!Caronas: Fostering Ridesharing with Online Social Network, Candidates Clustering and Ride Matching. In Proceedings of the 2016 8th Euro American Conference on Telematics and Information Systems (EATIS), Cartagena, Colombia, 28–29 April 2016.
6. Kalanick, T.; Camp, G. Uber. 2015. Available online: https://www.uber.com/ (accessed on 18 May 2022).
7. Mazzella, F. Blablacar. Available online: http://www.blablacar.com (accessed on 26 May 2022).
8. Ferrero, F.; Perboli, G.; Rosano, M.; Vesco, A. Car-sharing services: An annotated review. *Sustain. Cities Soc.* **2018**, *37*, 501–518. [CrossRef]
9. He, W.; Li, D.; Zhang, T.; An, L.; Guo, M.; Chen, G. Mining regular routes from GPS data for ridesharing recommendations. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, 12 August 2012; p. 79. [CrossRef]
10. Currie, J.; Walker, R. Traffic Congestion and Infant Health: Evidence from E-ZPass. *Am. Econ. J. Appl. Econ.* **2014**, *3*, 65–90. [CrossRef]
11. Levy, J.I.; Buonocore, J.J.; von Stackelberg, K. The Public Health Costs of Traffic Congestion: A Health Risk Assessment. *Environ. Health* **2010**, *9*, 65. [CrossRef] [PubMed]
12. Hart, J.E.; Laden, F.; Puett, R.C.; Costenbader, K.H.; Karlson, E.W. Exposure to Traffic Pollution and Increased Risk of Rheumatoid Arthritis. *Environ. Health Perspect.* **2009**, *117*, 1065–1069. [CrossRef] [PubMed]
13. Eriksson, H.-E.; Penker, M. *Business Modeling With UML: Business Patterns at Work*; Wiley: Hoboken, NJ, USA, 2000; p. 12. ISBN 978-0471295518.

14. He, W.; Hwang, K.; Li, D. Intelligent Carpool Routing for Urban Ridesharing by Mining GPS Trajectories. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2286–2296. [CrossRef]
15. Cruz, M.O.; Macedo, H.; Guimaraes, A. Grouping Similar Trajectories for Carpooling Purposes. In Proceedings of the 2015 Brazilian Conference on Intelligent Systems (BRACIS), Natal, Brazil, 4–7 November 2015; pp. 234–239. [CrossRef]
16. Carma, S.O. 2015. Dynamic Road Pricing. Available online: https://carmacarpool.com (accessed on 21 May 2022).
17. Yan, S.; Chen, C.Y.; Chang, S.C. A Car Pooling Model and Solution Method with Stochastic Vehicle Travel Times. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 47–61. [CrossRef]
18. Matos, M.L.; Cruz, M.; Guimaraes, A.; Macedo, H. A social network for carpooling. In Proceedings of the 7th Euro American Conference on Telematics and Information Systems, Valparaiso, Chile, 2–4 April 2014; pp. 1–6. [CrossRef]
19. Ghoseiri, K.; Haghani, A.; Hamedi, M. Real-Time Rideshare Matching Problem. Ph.D. Thesis, University of Maryland, College Park, MD, USA, 2011.
20. Arias-Molinares, D.; García-Palomares, J.C. The Ws of MaaS: Understanding mobility as a service fromaliterature review. *IATSS Res.* **2020**, *44*, 253–263. [CrossRef]
21. Dingil, A.E.; Rupi, F.; Esztergár-Kiss, D. An Integrative Review of Socio-Technical Factors Influencing Travel Decision-Making and Urban Transport Performance. *Sustainability* **2021**, *13*, 10158. [CrossRef]
22. Cruz, M.O.; Macedo, H.T.; Barreto, R.; Guimarães, A.P. GPS + Trajectories. Available online: https://archive.ics.uci.edu/ml/datasets/ (accessed on 22 May 2022).
23. Wang, L.P. *Support Vector Machines: Theory and Application*; Wang, L.P., Ed.; Springer: Berlin, Germany, 2005.
24. Platt, J. Fast training Support Vector Machines using parallel Sequential Minimal Optimization. In *Advances in Kernel Methods—Support Vector Learning*; MIT Press: Cambridge, MA, USA, 1998; pp. 41–65. [CrossRef]
25. Aha, D.W.; Kibler, D.; Albert, M.K. Instance-Based Learning Algorithms. *Mach. Learn.* **1991**, *6*, 37–66. [CrossRef]
26. Witten, I.H.; Frank, E.; Hall, M.A. *Data Mining*, 4th ed.; Elsevier: Amsterdam, The Netherlands, 2017.
27. Rodriguez, J.; Kuncheva, L.; Alonso, C. Rotation Forest: A New Classifier Ensemble Method. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 1619–1630. [CrossRef]
28. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
29. Schumacher, R.S.; Hill, A.J.; Klein, M.; Nelson, J.A.; Erickson, M.J.; Trojniak, S.M.; Herman, G.R. From Random Forests to Flood Forecasts: A Research to Operations Success Story. *Bull. Am. Meteorol. Soc.* **2021**, *102*, E1742–E1755. [CrossRef]
30. Friedman, J.; Hastie, T.; Tibshirani, R. Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors). *Ann. Stat.* **2000**, *28*, 337–407. [CrossRef]
31. Kira, K.; Rendell, L.A. A Practical Approach to Feature Selection. In *Machine Learning Proceedings 1992*; Elsevier: Amsterdam, The Netherlands, 1992; pp. 249–256. [CrossRef]
32. Kononenko, I. Estimating Attributes: Analysis and Extensions of RELIEF. In *European Conference on Machine Learning*; Springer: Berlin/Heidelberg, Germany, 1994; Volume 784, pp. 171–182. [CrossRef]
33. Breiman, L. Technical note: Some properties of splitting criteria. *Mach. Learn.* **1996**, *24*, 41–47. [CrossRef]
34. Pandey, M.K.; Mittal, M.; Subbiah, K. Optimal balancing & efficient feature ranking approach to minimize credit risk. *Int. J. Inf. Manag. Data Insights* **2021**, *1*, 100037. [CrossRef]