Machine Learning Approach for Labeling Undetected Planned Trips in Public Transport Operators

Mohammad Amin Zadenoori[1](✉), Marco Calamai[2], Francesca Del Lungo[2], Daria Faucci[2], Andrea Gaffi[2], Lorenzo Sarti[2], and Alessio Micheli[3]

[1] National Research Council (CNR), Pisa, Italy
mohammadamin.zadenoori@isti.cnr.it
[2] MAIOR Srl, Pistoia, Italy
[3] Department of Computer Science, University of Pisa, Pisa, Italy
micheli@di.unipi.it

Abstract. Accurate labeling of undetected trips in public transportation is critical, as it directly affects operational efficiency, cost savings, and service quality. Undetected trips refer to scheduled trips that were either not completed or inaccurately recorded by Automatic Vehicle Location (AVL) systems. These discrepancies can disrupt resource allocation, hinder operational planning, and compromise financial accountability. If undetected trips are not properly classified, they can cause significant financial losses, misallocation of resources, and lower customer satisfaction due to unaddressed service issues.

This paper presents a machine learning approach to automate the classification of undetected trips in public transit. The model categorizes trips into three types: Operated (successfully completed trips), Lost-Deductible (missed trips within operational limits), and Lost - Non-deductible (missed trips outside operational standards and non-compensable). Automating this process enhances operational efficiency, reduces financial losses, and streamlines claim management. By replacing manual classification with AI-driven automation, transit operators can ensure faster, more accurate trip labeling, ultimately leading to optimized resource use, better decision-making, and higher service standards.

Keywords: Machine Learning · Public Transport · Trip Classification · Random Forest · Model Deployment

1 Introduction

In the realm of public transit, undetected trips are a common occurrence. Undetected trips refer to trips (or parts of them) that are not detected by AVL systems. Public Transport Authorities (PTAs) serve as clients who contract Public Transport Operators (PTOs) to provide mass transit services to the public. The relationship between PTAs and PTOs is contractual and often performance-based, with PTAs holding the authority to impose penalties on PTOs if service

A. Kocian et al. (Eds.): INTSYS 2024, LNICST 608, pp. 177–190, 2025.
https://doi.org/10.1007/978-3-031-86370-7_11

levels do not meet the agreed-upon standards. One essential aspect of this collaboration is the need for operators to report any deviations from the planned service and specify the reason behind such deviations.

Manually categorizing these deviations is both time-consuming and prone to human error. There is a pressing need for an automated solution to address this issue, ensuring that deviations are categorized correctly and efficiently.

Leveraging historical data from a transit operator as our foundational dataset, the primary objective of this work is to develop a machine learning-driven classifier that can analyze trip data and predict the reasons for any undetected trips. In today's landscape, it is crucial for public transport agencies and operators to understand their service levels to optimize transport availability and efficiency, ensuring a high-quality service for the public.

This industrial research was conducted within MAIOR, a global market leader in designing and developing software solutions for strategic service planning, resource scheduling and management for both PTAs and PTOs. The collected data for this work comes from one of MAIOR's customers and serves as the foundation for training and validating the machine learning model. The broader vision of this research is to develop a versatile trip classifier that can be applied across various public transit operators, beyond just the current data source. By doing so, we aim to optimize transit operations, reduce manual workloads, ensure accurate reporting, and pave the way for a more efficient and scalable public transportation system.

In the following sections, this paper explores various aspects of the project. Section 2 presents a Literature Review, focusing on the application of machine learning in public transportation systems. Section 3 details the Project Objectives, outlining the goals and guiding research questions. Section 4 explains the methodology, describing the processes of data collection, feature engineering, model training, and model evaluation. Section 5 offers a thorough discussion of the obtained results, along with the model deployment, including its practical implications for public transit operators and its potential to improve operational efficiency and decision-making. Finally, Sect. 6 concludes the paper by summarizing the research's impact and providing insights for future developments in this area.

2 Literature Review: Machine Learning in Public Transportation

Machine learning (ML) is increasingly pivotal in optimizing public transportation systems, leveraging large-scale data to enhance operations, efficiency, and prediction capabilities. One notable application of ML is in trip classification and anomaly detection. Convolutional neural networks (CNNs) are employed to identify anomalies in scheduled trips by analyzing historical AVL data, significantly improving service reliability [5]. Additionally, hybrid models that combine decision trees and gradient boosting were used to classify trips accurately into categories like "On-Time","Delayed", or "Missed", helping transportation authorities make timely decisions [6]. Methods such as Random Forests and Extremely Randomized Trees have further improved classification accuracy [1] [3].

ML was also applied to improve operational efficiency. Reinforcement learning models optimize bus routes by analyzing real-time traffic and passenger data, which reduces operational costs and enhances service delivery [7]. Predictive maintenance models, based on sensor data and vehicle health metrics, can forecast when maintenance is required, preventing breakdowns and reducing vehicle downtime [8].

In terms of passenger experience, ML-based recommender systems personalize route suggestions by analyzing commuter preferences and real-time traffic data, leading to more efficient and satisfying travel experiences [6]. Furthermore, ML models are used to manage crowd flow by predicting congestion in transit stations through real-time data integration from sources such as fare gates and CCTV. This helps reroute passengers and prevent overcrowding, improving safety and comfort [9].

In summary, ML is revolutionizing public transportation through examples such as trip classification, operational efficiency improvements, and passenger experience enhancements, offering significant opportunities for future advancements in the management of urban transit networks.

3 Project Objective

The primary objectives of this work are centered around addressing key challenges in public transit operations. These objectives guide the research questions that this work seeks to answer:

Research Question 1: How can a robust machine learning model be developed to accurately classify trips into one of three categories: “Operated”, “Lost - Deductible”, and “Lost - Non-deductible”?

Research Question 2: How can accurate trip classifications assist in improving decision-making within a transportation company, particularly in identifying potential trip issues and optimizing deductible claims?

Research Question 3: How can the automation of the trip classification process enhance operational efficiency, reduce manual intervention, and streamline public transit operations?

4 Data Modeling

The success of the machine learning model heavily relies on high-quality data gathered from a real public transit operator for the duration of one year (2022). The following steps were taken to collect and prepare the dataset.

Historical trip data was collected from the operator’s database, involving both planned and actual trips. The dataset encompasses details about intended trips and actual trips captured using an AVL system. This data includes trip dates, route information, vehicle information, and any incidents or claims associated with each trip.

4.1 Dataset Overview

In the dataset under consideration, three distinct labels have been assigned to various instances: *"Operated"* with a frequency of 75,005, *"Lost - Deductible"* with a count of 53,210, and *"Lost - Non-deductible"* with an occurrence of 45,675.

A 70-30 train-test split was used, allocating 70% of the data for training and 30% for testing. Additionally, 5-fold cross-validation was applied to the training set, cycling through five subsets to improve model reliability and reduce overfitting.

Furthermore, to comprehensively assess the performance of the classifier, an initial evaluation was conducted on the training set using a 5-fold cross-validation approach.

4.2 Features

The features in the dataset were summarized into key categories:

- **Temporal Features**: Day of the week, type of day (weekday, weekend, holiday).
- **Route-Related Features**: Line number, pattern, direction.
- **Operational Features**: Planned number of stops, planned departure and arrival times.
- **AVL-Related Features**: AVL detection percentages, detection of trip departure and arrival.
- **Delay Measurements**: Delays at start and end stops.
- **Contextual Features**: AVL percentages of previous and next trips with the same pattern or vehicle block.

4.3 Description of Trip Categories

Operated Trips

"Operated" trips refer to successfully executed bus journeys within the transportation system. These are instances where buses have completed their scheduled routes and reached their destinations as intended.

Lost - Not Deductible

"Lost - Not Deductible" trips refer to undetected trips that have been deemed lost due to delays or disruptions but are eligible for deductible claims, allowing the transportation company to recover a portion of the financial loss incurred. Causes of Lost - Not Deductible Trips include:

- **Adverse Weather Conditions**: Heavy rain, snowstorms, or fog causing delays.
- **Traffic Congestion**: Unexpected road closures, accidents, or high traffic volumes.
- **Operational Delays**: Route deviations or driver-related issues.

Lost - Deductible

"Lost - Deductible" trips refer to undetected trips that are lost without eligibility for deductible claims, resulting in a direct financial loss for the company. Causes of Lost - Deductible Trips include:

- **Internal Operational Issues**: Delays due to internal inefficiencies without external factors.
- **Schedule Disruptions**: Conflicts arising from poor scheduling.
- **Non-insurable Events**: Incidents not covered by insurance policies.
- **Vehicle Breakdowns**: Mechanical issues disrupting the trip.

4.4 Model Selection and Training

Several machine learning algorithms were evaluated to classify the undetected trips in public transit operations. The models considered include Random Forest, K-Nearest Neighbors (KNN), XGBoost, and Extra Trees.

Random Forest: Random Forest is an ensemble learning method that constructs multiple decision trees and merges them to obtain more accurate and stable predictions. This model is particularly effective in handling both classification and regression tasks, and it mitigates the risk of overfitting compared to individual decision trees [1].

K-Nearest Neighbors (KNN): KNN is a non-parametric, instance-based learning algorithm. It classifies a data point based on the majority label of its nearest neighbors in the feature space. While simple, KNN can be computationally expensive and sensitive to the choice of distance metrics and the number of neighbors, k [2].

XGBoost: XGBoost (Extreme Gradient Boosting) is a powerful gradient boosting algorithm optimized for both speed and performance. It builds sequential trees, where each new tree attempts to correct the errors made by the previous one. Its efficiency and effectiveness make it well-suited for structured data tasks, but it requires careful tuning of hyperparameters to avoid overfitting [4].

Extra Trees: The Extra Trees algorithm (Extremely Randomized Trees) is similar to Random Forest but differs in how it splits nodes. Instead of choosing the optimal split, Extra Trees chooses splits randomly. This randomness often leads to better generalization but can introduce more variance in some cases [3].

Hyperparameter tuning was performed using a grid search strategy. A grid of potential hyperparameters was defined for each model, and an exhaustive search was conducted, varying the parameters in the potential hyperparameter ranges, to identify the best combination of parameters that maximized performance. The models were further evaluated through 5-fold cross-validation to ensure robustness and avoid overfitting.

4.5 Model Classification Performance

The classification performance of four machine learning models-**Random Forest**, **K-Nearest Neighbors (KNN)**, **XGBoost**, and **Extra Trees**-was evaluated based on several metrics. Table 1 presents the comparison of these models in terms of Training Accuracy, Testing Accuracy, Average Cross-Validation (CV) Accuracy with 5 folds, and the Standard Deviation of CV Accuracy.

Table 1. Performance Metrics for Different Models

Metric	Random Forest	KNN	XGBoost	Extra Trees
Training Accuracy	82%	76%	74%	**90%**
Testing Accuracy	**77%**	73%	74%	76%
Average CV Accuracy (k=5)	**76%**	73%	73%	75%
Std. Dev. of CV Accuracy	2.6%	2.6%	**2.5%**	2.6%

From the table, we can observe the following insights:

- The **Random Forest** model demonstrates the highest testing accuracy (77%), which suggests that it generalizes well to unseen data.
- The **K-Nearest Neighbors (KNN)** model has the lowest testing accuracy (73%) and training accuracy (76%), which might suggest that the model struggles with both the training and test sets.
- **XGBoost** shows consistent performance between training (74%) and testing (74%) accuracies, which may indicate that the model is balanced, neither overfitting nor underfitting.
- The **Extra Trees** model shows the highest training accuracy (90%), but with a significant drop to 76% in testing accuracy. This indicates potential overfitting, where the model performs well on the training data but fails to generalize as effectively.

In terms of cross-validation accuracy, all models perform similarly, with averages ranging from 73% to 76%. The standard deviation of cross-validation accuracy is low for all models, indicating that performance is stable across the folds.

Given these results, we choose **Random Forest** as our primary classifier. In addition to its slightly better performance, Random Forest models are known for their interpretability and the ability to provide **feature importance** scores, which are crucial for understanding the underlying patterns in the data. This makes Random Forest not only effective but also valuable for gaining insights into which features most influence the model's decisions.

4.6 Random Forest-Grid Search Hyperparameter Optimization

Here, we present the results of optimizing the Random Forest model using a Grid Search strategy. The objective was to identify the best combination of hyperparameters to enhance the model's predictive performance. The Random Forest algorithm is an ensemble method that constructs multiple decision trees and aggregates their predictions for more accurate and stable outcomes.

During hyperparameter optimization using random search with 5-fold cross-validation, the training data is divided into 5 equal parts (folds). The model is trained on 4 of these folds and validated on the remaining fold. This process is repeated 5 times, with a different fold used as the validation set each time. This ensures that every part of the data is used for both training and validation, helping to avoid overfitting and providing a reliable estimate of the model's performance. For each random combination of hyperparameters, this cross-validation process is repeated, and the best-performing hyperparameter combination across all splits is selected.

Table 2 shows the optimal hyperparameters selected through the Grid Search process. Each parameter is explained below, emphasizing its importance in improving the model's generalization ability and preventing overfitting.

Table 2. Optimized Random Forest Hyperparameters with Modified Parameter Ranges

Parameter	Optimal Value	Range	Description
n_estimators	10	[2, 4, 6, 10]	The number of trees in the forest
min_samples_split	10	[2, 5, 10]	The minimum number of samples required to split an internal node. A value of 10 means a node will only be split if it contains at least 10 samples
min_samples_leaf	2	[1, 2, 4]	The minimum number of samples required to be at a leaf node. Setting this value to 2 prevents the model from creating overly complex trees with too few samples at the leaf nodes
max_features	auto	[auto, sqrt]	The number of features to consider when looking for the best split. When set to auto, the algorithm automatically determines the optimal number of features
max_depth	15	[7, 8, 9, 15, 100, None]	The maximum depth of the tree. A depth of 15 allows the trees to grow relatively deep, capturing more complexity in the data, but it can also increase the risk of overfitting
bootstrap	True	[True, False]	Whether bootstrap samples are used when building trees. When set to True, the model samples with replacement, which helps in creating more robust models

After tuning the model, we evaluated its performance on the test set. The classification report for this dataset is presented in Table 3, summarizing precision, recall, F1-score, and support for each class. The largest number in each column is highlighted in bold to indicate the best-performing metrics.

Table 3. Random Forest Classification Report - Test Set

Class	Precision	Recall	F1-Score	Support
Lost - Deductible	72%	**73%**	72%	10,503
Lost - Non-deductible	**79%**	74%	**77%**	10,691
Operated	78%	**81%**	**80%**	16,021
Accuracy	**77%** (37,215 samples)			
Macro Avg	**77%**	76%	76%	37,215
Weighted Avg	**77%**	**77%**	**77%**	37,215

Explanation of Results

The performance of the model on the test set reveals valuable insights into its classification ability. As shown in Table 3, the model demonstrates its highest precision for the "Lost - Non-deductible" class, reaching 79%. This indicates that the model was most accurate when predicting this class, with fewer false positives compared to the other categories. The highest recall of 81% is observed in the "Operated" class, which means that the model correctly identified the majority of true positive instances within this class, minimizing the number of missed "Operated" samples.

The F1-score, which balances precision and recall, is highest for both the "Operated" (80%) and "Lost - Non-deductible" (77%) classes. This suggests that these two classes exhibit a good balance between identifying positive instances and minimizing false positives, showcasing the model's ability to handle these categories effectively.

Moreover, the overall accuracy of the model on the test set is 77%, calculated across all classes. While this accuracy reflects the model's reasonable performance on unseen data, the difference between the test accuracy and any previously evaluated training performance (not shown here) might indicate slight overfitting or challenges with generalization. Nonetheless, the classification report reveals consistent performance across the three main classes, with similar precision, recall, and F1-scores, highlighting the robustness of the model's predictions for each class.

4.7 Random Forest Feature Importance Analysis

The model's prediction capability is influenced by a set of features, each with varying degrees of significance. Highest importance is **line number** with an importance score of 0.13, underscoring its important role in the model's outcomes. It is closely followed by **number of stops** and **delay at the starting stop**, registering scores of 0.11 and 0.10, respectively.

These top three features potentially reflect the core aspects of transportation patterns, from line-specific dynamics to stoppage patterns and initial delays. mid-tier features such as **pattern** and **avl percentage**, with scores of 0.09 and 0.08 respectively, also retain significant influence. However, as we traverse down the ranking, diminishing importance scores like that of **weekday** at 0.01 and **direction** at 0.01 indicate their lesser contribution to the model's decisions. interestingly, **virtual stops** with a score of 0.00 underscores its negligible or non-existent impact on predictions.

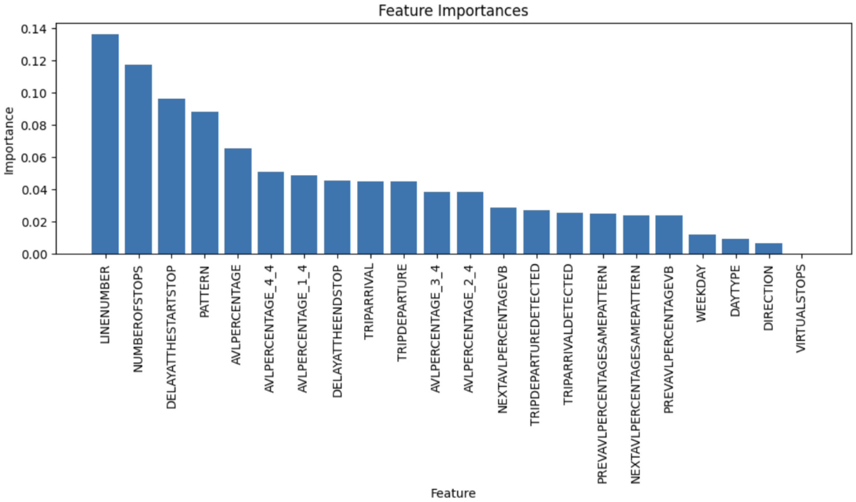

Fig. 1. Feature importance plot showing the relative significance of each feature in the random forest model.

4.8 Insights

As shown in Fig. 1, the *line number* feature emerges as the most influential, indicating that specific transit lines may exhibit unique operational characteristics that significantly affect the overall performance of the system. This could be due to the varying nature of routes, including traffic patterns, ridership levels, or infrastructure quality, which makes this feature critical for accurate trip classification.

Similarly, the *number of stops* and *delay at the starting stop* are highlighted as crucial factors. The *number of stops* directly correlates with the complexity of a trip, while the *delay at the starting stop* is a key indicator of whether the system meets scheduled timetables, which is vital for passenger satisfaction and resource optimization.

Mid-tier features, such as the *pattern* and *AVL percentage*, also play a significant role. These features have relatively high importance scores, reflecting their contribution to the model. The *pattern* represents the predefined route structure, which can influence trip predictability, and the *AVL percentage* indicates the availability of real-time location data, which improves the system's ability to track performance and make informed decisions.

On the other hand, features like *weekday*, *direction*, and *virtual stops* contribute much less, as evidenced by their lower importance scores. Although these features are included in the model, their impact on the classification outcomes is minimal compared to the more dominant features. The *weekday* may introduce some variation in travel patterns, but this effect is relatively minor. Similarly, *direction* and *virtual stops* have limited influence, suggesting they are not as critical in determining the classification of transit trips.

In summary, the model's feature importance rankings provide valuable insights into which aspects of the transit system most influence performance, with operational efficiency and route characteristics emerging as key determinants.

4.9 Confusion Matrix of Random Forest Classifier

The confusion matrix displayed in Fig. 2 represents the performance of the Random Forest classifier on classifying trips into the categories *Lost Deductible*, *Lost Non-deductible*, and *Operated.* Each cell in the matrix corresponds to the percentage of predictions relative to the true labels.

- The classifier correctly identified 69.65% of trips labeled as *Lost Deductible*, while 22.95% were misclassified as *Operated*, and 7.40% as *Lost Non-deductible.*
- For the *Lost Non-deductible* category, 75.44% of the trips were correctly classified, with 12.59% misclassified as *Operated* and 11.97% as *Lost Deductible.*
- In the *Operated* category, the classifier correctly identified 80.50% of trips, while 10.71% were incorrectly classified as *Lost Deductible* and 8.79% as *Lost Non-deductible.*

The confusion matrix highlights that the Random Forest classifier performs reasonably well, particularly in the *Operated* and *Lost Non-deductible* categories. However, there is a noticeable proportion of misclassifications between the *Lost Deductible* and *Operated* labels, which suggests room for improvement in the classifier's ability to distinguish between these categories.

5 Discussion

Based on the results achieved by the Random Forest (RF) classifier, the RF model can be considered to be well-suited for deployment in a real-world application. The RF model demonstrated adequate accuracy during testing, with

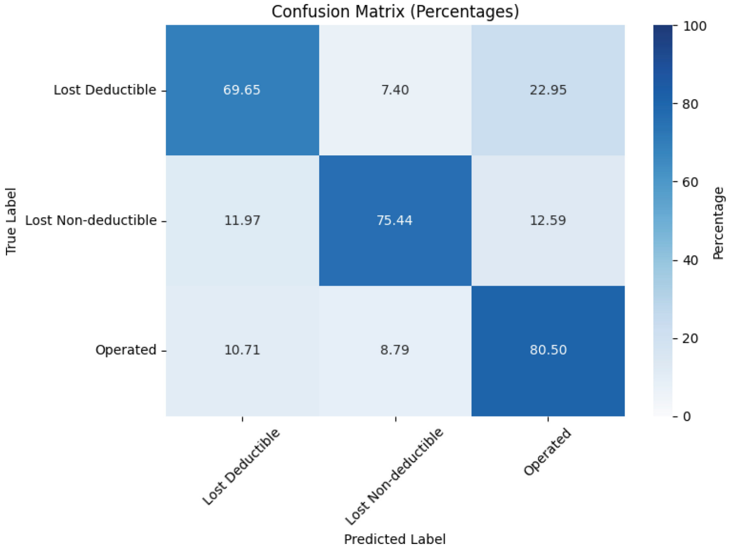

Fig. 2. Confusion Matrix of the Random Forest Classifier (Percentages)

accuracy score of 77%, respectively. This accuracy on testing set indicates that the model can provide reliable predictions on new, unseen data.

Moreover, the RF model's average cross-validation accuracy of 76% and the small standard deviation of 2.6% further validate its robustness and consistency in different data scenarios. The RF model's ability to maintain stable and dependable performance across various data folds increases confidence in its predictive capabilities.

Due to its ability to capture complex relationships in the data, avoid overfitting, and generalize well to new data, the RF model proves to be a powerful and reliable predictive tool. It has demonstrated slightly better performance compared to other classifiers, making it a better choice for model deployment in our application.

5.1 Answering Research Questions Based on Results

Answering RQ1: *How can a robust machine learning model be developed to accurately classify trips into one of three categories: Operated, Lost - Deductible, and Lost - Non-deductible?*

Answer: The Random Forest model demonstrated the slightly better performance among the models tested, achieving high accuracy on both the training (82%), validation (76%), and testing (77%) datasets, with a small standard deviation (2.6%) across cross-validation folds. This suggests that Random Forest can be considered a robust and reliable model for trip classification. The model's ability to generalize well on unseen data highlights its potential to accurately classify trips into the three predefined categories (Operated, Lost - Deductible, and Lost - Non-deductible).

The top five features identified—*NumberOfStops, AVLPercentage, DelayAtTheStartStop, LineNumber*, and *Pattern*—indicate that specific operational characteristics and patterns have a substantial influence on classification outcomes. These features collectively allow the model to distinguish between the trip types efficiently. Therefore, the development of such a machine learning model is feasible and effective in categorizing undetected trips.

Answering RQ2: *How can accurate trip classifications assist in improving decision-making within a transportation company, particularly in identifying potential trip issues and optimizing deductible claims?*

Answer: Accurate classification of trips into categories such as *Lost - Deductible* and *Lost - Non-deductible* provides critical insights into which trips are eligible for deductible claims and which are not. With Random Forest's reliable predictions, transportation companies can better understand trip performance and quickly identify issues that contribute to deductible losses. For instance, features like *DelayAtTheStartStop* and *NumberOfStops* could be key indicators for identifying trips that need operational interventions.

This capability enables decision-makers to prioritize trips for investigation and streamline the claims process. Furthermore, by understanding the patterns of lost trips, corrective actions can be planned, such as revising scheduling or improving vehicle allocation, thereby optimizing the company's operations and financial outcomes.

Answering RQ3: *How can the automation of the trip classification process enhance operational efficiency, reduce manual intervention, and streamline public transit operations?*

Answer: In this work, we have achieved several industrial milestones, leading to improvements in the efficiency of our processes. These achievements are outlined below:

- We have designed a system that operates effectively in real-world scenarios, managing the labeling of approximately 1500 trips daily.
- The manual labeling process, which previously took 5 to 10 s per trip (amounting to 2 to 4 h each day), has been optimized.
- By implementing an automatic labeling system, the time required per trip has been reduced to an 5 to 10 milliseconds, reducing the total time to merely 2 to 4 s.
- Our model achieves a prediction accuracy of about 76%, ensuring an adequate level of reliability.

Overall, clearly, the new process is much faster, even when considering a final human review for verification.

5.2 Model Deployment with Flask

In this work, Flask serves as the web framework for deploying the machine learning model, allowing seamless integration with MAIOR's transit planning system. By deploying the model as a REST API, the Flask server acts as an intermediary between the model and external systems, efficiently handling requests and delivering predictions [11].

The deployment process involves creating RESTful endpoints that accept data for classification and return the corresponding results. Flask's lightweight nature ensures minimal overhead, while also allowing for easy updates or replacements of the model without disrupting the system's functionality [10]. This modularity ensures that the classifier remains flexible and adaptable to changing needs.

Additionally, Flask supports containerization through Docker, enabling scalable deployment and efficient resource management [12]. This approach ensures that the model is not only operational but also optimized for performance, security, and maintainability within a production environment [13].

6 Conclusions

In conclusion, the application of machine learning for classifying discrepancies in public transit operations is a groundbreaking venture that holds significant promise for the industry. The challenges faced due to manual categorization processes have been met with an innovative, data-driven solution that not only addresses the inefficiencies but also offers scalability across different transit operators. Our efforts, underpinned by the rich dataset from a MAIOR's costumer, have yielded a classifier that can pave the way for more reliable, streamlined, and efficient public transportation systems.

As we reflect on this endeavor, it is evident that machine learning has the potential to revolutionize the way public transit agencies approach operational discrepancies, and we remain hopeful about its transformative impact on the future of transit operations.

Acknowledgement. This research has been carried out as part of the ADDSTRES project, funded by Regione Toscana and MAIOR Srl. Thanks to Prof. A. Frangioni to coordinate the project.

References

1. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
2. Altman, N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. Am. Stat. **46**(3), 175–185 (1992)
3. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Mach. Learn. **63**(1), 3–42 (2006)

4. Chen, T., Guestrin, C.: XGBoost. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM (2016)
5. Elvas, L., Nunes, M., Francisco, B., Gonçalves, F., Martins, A.L., Ferreira, J.C.: Points of interest in smart cities and visitor behavior (2023)
6. Caroleo, B., et al.: Machine learning methods to forecast public transport demand based on smart card validations (2023)
7. Lozano Pinilla, J.R., Vicente Chicote, C., Sanchez Cordero, I.: Smart-routing web app: a road traffic eco-routing tool proposal for smart cities (2023)
8. Tokkozhina, U., Martins, A.L., Ferreira, J.C.: Blockchain-based solution for charitable supply chains: network proposal architecture for Portuguese tax consignment program (2023)
9. Elvas, L., Nunes, M., Francisco, B., Domingues, N.: City mobility and night life monitor (2023)
10. Grinberg, M.: Flask Web Development: Developing Web Applications with Python. O'Reilly Media Inc, Sebastopol (2018)
11. Flask Documentation, "Flask," (2024). https://flask.palletsprojects.com/. Accessed 03 Oct 2024
12. Docker Documentation, "Docker," (2024). https://docs.docker.com/. Accessed 03 Oct 2024
13. Kruchten, P.: The 4+1 view model of architecture. IEEE Softw. **12**(6), 42–50 (1995)