Project 2

Team members:

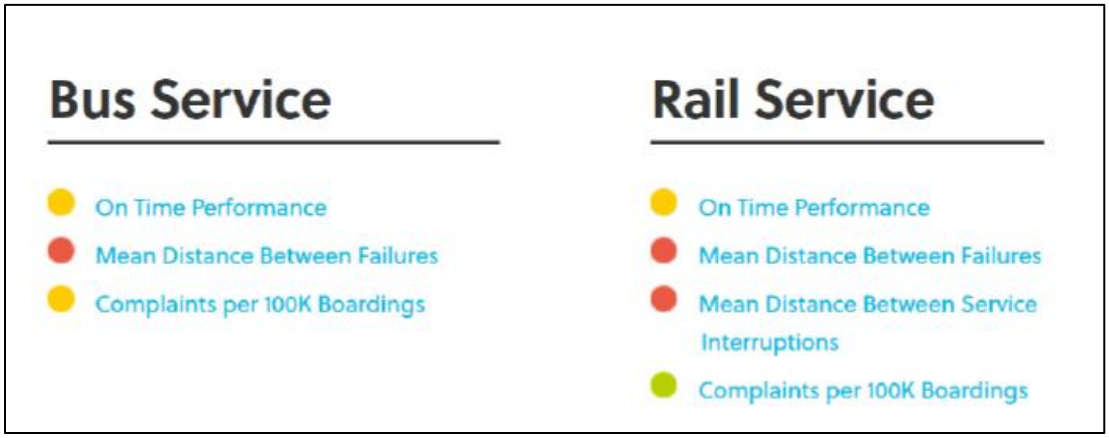
Hari Priya Avarampalayam Manoharan - 002711275 -havarampalayammanoh1@student.gsu.edu
Rida Fathima - 002695685- rfathima1@student.gsu.edu
Sakshi Sachin Agarkar - 002712319 - sagarkar1@student.gsu.edu
Lilia Chebbah - 0027044185 - lchebbah1@student.gsu.edu
Mouayed Lajnef - 002704186 - mlajnef1@student.gsu.edu- (MANAGER)

"Statement of Academic Honesty:

The following code represents our own work. We have neither received nor given inappropriate assistance. We have not copied or modified code from any source other than the course webpage or the course textbook. We recognize that any unauthorized assistance or plagiarism will be handled in accordance with Georgia State University's Academic Honesty Policy and the policies of this course. We recognize that our work is based on an assignment created by the Institute for Insight at Georgia State University. Any publishing or posting of source code for this project is strictly prohibited unless you have written consent from the Institute for Insight at Georgia State University."

1. Motivation and Problem:

MARTA (The Metropolitan Atlanta Rapid Transit Authority) is a public transport operator in the Atlanta metropolitan area. MARTA being the 8th largest transit system in the US, operates bus routes linked to a rapid transit system consisting of 48 miles (77 km) of rail track with 38 train stations. According to sources, in 2021, the entire system (bus and rail) had 50,288,800 rides or about 179,600 per weekday in the second quarter of 2022. It has become the primary mode of transportation thanks to its affordability and connectivity to major areas in Atlanta. However, amidst this, the MARTA schedule suffers from a great problem reported by its daily users which is the frequent delays in the arrival of buses and trains. That led to the loss of trust and satisfaction by its users. Adding to that the fact that customers can sometimes be impacted in their livelihood because the planned bus does arrive on time. The problem here is twofold. From the customer's side, MARTA riders do not know if the scheduled trips arrive on time at a particular stop causing them to miss their appointments and commitments. From the organization's side, their scheduling is not accurate, causing them to lose the trust of their riders and lose customers. In fact, MARTA provides official KPI's that they need to keep in check. This further shows that here is a gap in the arrival times that requires a solution.



2. Proposed Solution

Our proposed solution is to create a classification model that predicts if busses will arrive either on time or not at each particular stop. The definition of in time is whether the bus arrives within 3 minutes early or late. The goal is to integrate the solution into their existing mobile application that already notifies users of their train arrivals but not busses. That would allow MARTA riders to plan their trips in a more accurate manner. While MARTA can fulfill its duties & obligations as a public service by promoting the interest of the public and providing a more consistent method of transportation. Users will not be put at a major disadvantage in their livelihoods for not being able to afford a car.

3. Data Used

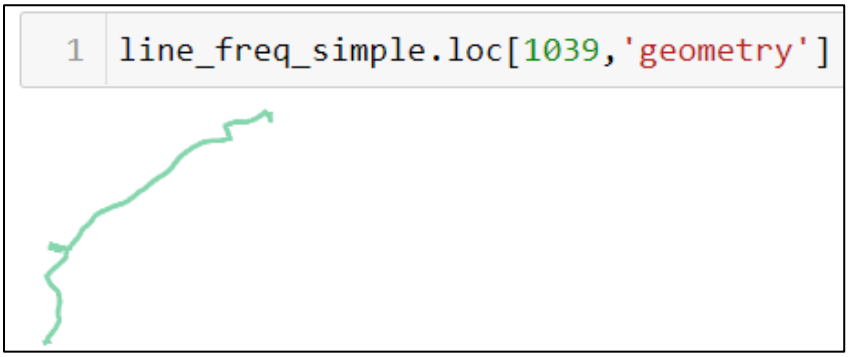
For this project, we used Public "The General Transit Feed Specification (GTFS)" data. Public transport agencies can publish their transit data in a format that can be used by a wide range of software applications according to the General Transit Feed Specification (GTFS), a data specification. Today, tens

of thousands of public transportation companies employ the GTFS data format. A real-time component of GTFS comprises arrival forecasts, vehicle positions, and service advisories in addition to a schedule component that contains schedule, fare, and geographic transit data.

4. Data Preparation

a. Feature Selection

The GFTS data discusses above includes several comma-separated ".txt" files that are connected together through primary keys or ID's. By merging these files, it is possible to extract valuable information and features for our prediction. Information that were used for this model are the following: Day of the week, Scheduled stop times, Vehicle Number, Vehicle Route, Stations, Station Stop Order, Station Sequence in a Day, Temperature, Stop location in a Block & Route information. The route information needed some transformations in order to extract trainable features:



The data type of the geometry is a LINESTRING of pairs of latitude and longitude points. All the pairs are connected together to constitute a line segment, a full route or a mini route. From that we were able to extract the following features:

1- Number of nodes
2- Start position
3- End position
4- Length of the line segment
5- Minimum clearance, which is a measure of the robustness of the geometry
6- Hausdorff-distance from the line centroid.

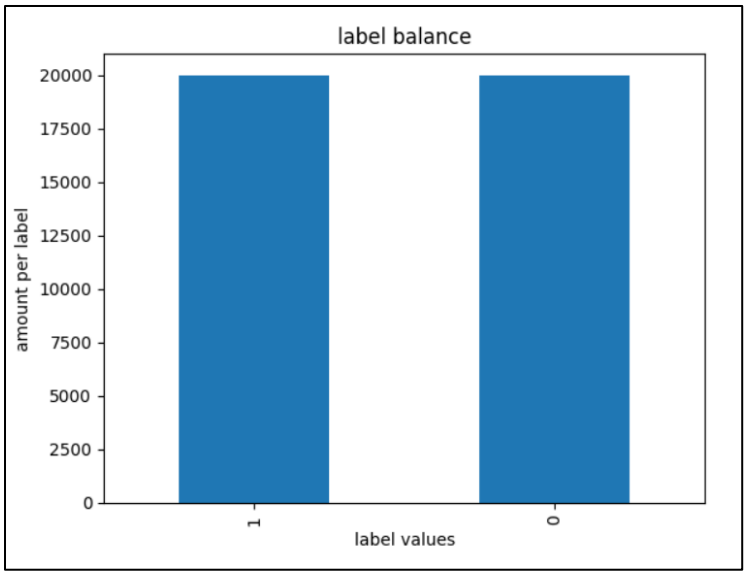
b. Normalization & Cleaning

The Data was subjected to standard normalization & cleaning techniques. All categorical variables were enumerated. Latitude and longitude data were converted to x & y coordinates using the following:

$$x = \cos(Latittude) \times \cos(Longitude)$$

$$y = \cos(Latittude) \times \sin(Longitude)$$

At the end, we performed data under sampling to achieve a balanced dataset in terms of the target variable, leaving us with 40,000 total data points



Finally, the data was split into training and test sets with a 80:20 ratio. Using the prepared data, we tested several classification algorithms and chose the model that gave the best overall performance metrics. The next part will discuss the algorithms used, their fine tuning & the results well be summarized in the end.

5. Machine Learning

a. Logistic Regression & Vector Machine

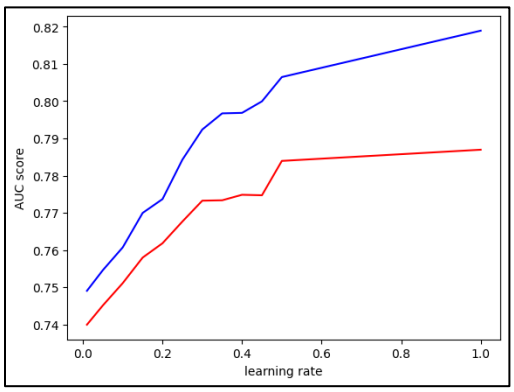
We first chose to use logistic regression and support vector machine because they are the most well-known machine learning techniques for binary classification, which the case of our study. We want to predict if the train is on time or not. But we found that the performance metrics were low. This can be explained by the simplicity of this algorithm that can sometimes not work well with large datasets. So, we chose not to focus on finetuning them since we have other models that outperformed significantly these two.

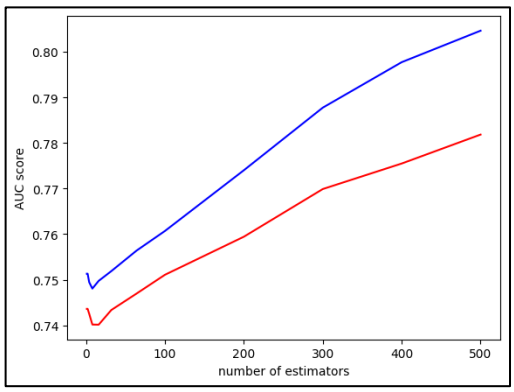
b. Gradient Boosting Trees

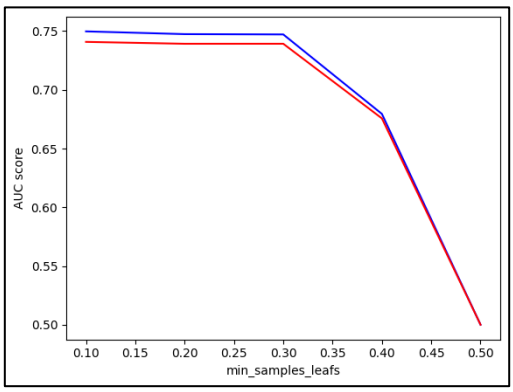
Similar to random forests, gradient boosting trees use a group of trees to predict the target value. This machine learning technique can be used for both classification and regression. Our case is a binary classification, that's why we used the classifier. In training, instead of creating the trees independently like the random forests, the trees are created sequentially, one after another, to measure each time the residuals and use it for the next tree in order to correct the previous errors. So, the most important parameter for this classifier is the loss function result. The output is then the class that is most predicted by the trees.

Since our classification is binary, we can use the AUC-ROC curve to visualize the performance of our models. The higher the area under the curve the better the model is performing. We used the AUC-ROC

curve for the number of estimators, learning rate, minimum sample leaf finetuning for the boosting classifier. We tried to do it for the maximum sample leaf, but it took so long.

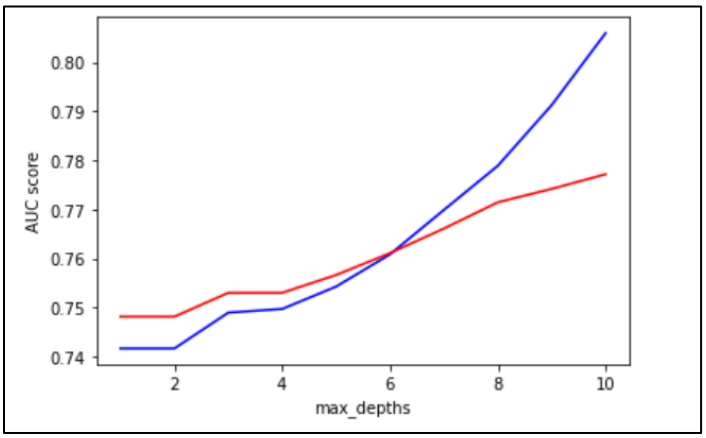






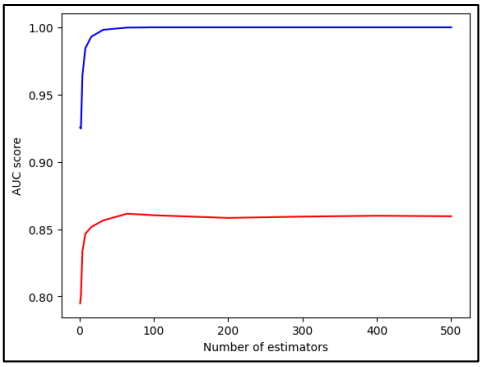
c. Decision Tree

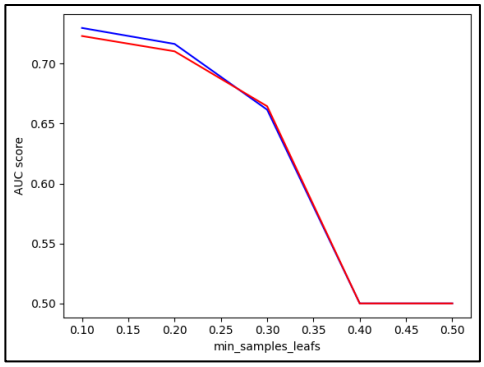
A decision tree is a flowchart-like tree structure where the internal node represents an attribute, the branch represents a decision rule, and the leaf node represents an outcome. It was picked due to it being able to handles skewed classes which suits well for our dataset that is non-linear in nature. It is also intuitive to understand and makes it easier to explain to all stakeholders. The decision tree ran on the both the GINI index & Entropy & hyper-tuned using maximum depth of 9.

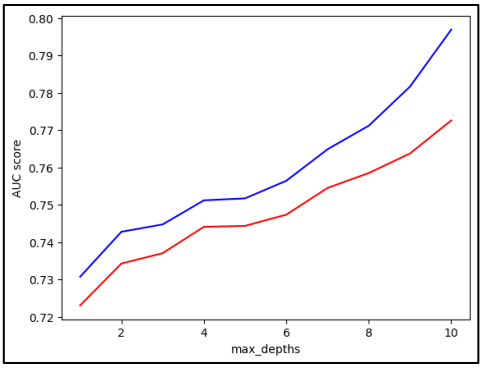


d. Random Forest

Random forest algorithm combines multiple decision-trees, resulting in a forest of trees. In the random forest classifier, the higher the number of trees in the forest results in higher accuracy. It was used due to it working well on large datasets & can overcome overfitting by taking the majority voting. Similar to Decision tree it was implemented for GINI index & Entropy. It was hyper-tuned using maximum depth, minimum sample leaves & number of trees.

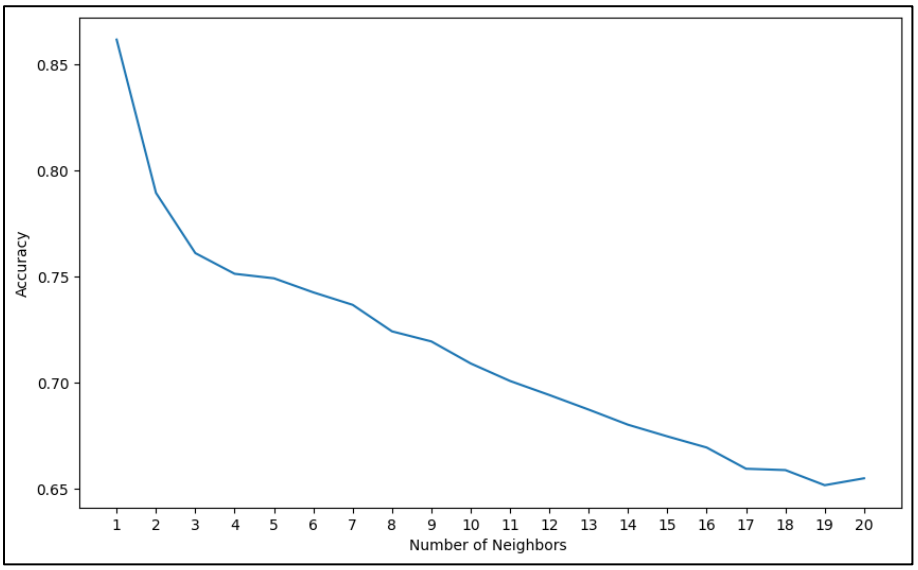






e. KNN

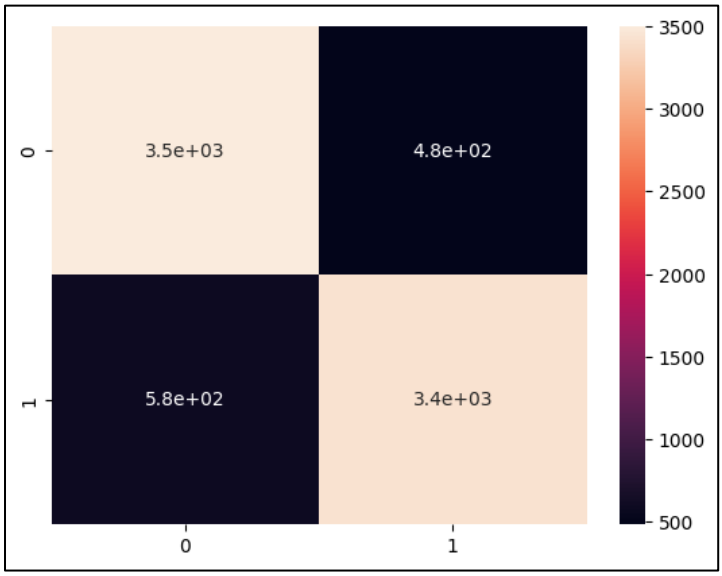
KNN is also another famous algorithm that was considered for our case. The default metrics for this algorithm were promising so we decided to hypertune its parameters using an exhaustive grid search technique. Goes through all the parameters of the model and chooses the model with the best metrics. The best model had the following parameters: n_neighbors = 3, weights = 'distance' and metric = 'Manhattan'.



6. Results & Discussion

In this part, we will present the results of the hypertuned models by summarizing them in a table that has the performance metrics of each machine learning model. We can observe from the table below that the best performing model is the KNN across all the performance metrics. Therefore, for the purpose of this project, it chosen to be the best performing model. Below is the confusion matrix for the KNN model

ML Model	Accuracy	Precision	F1-Score
Logistic Regression	0.63	0.63	0.63
Support Vector Machine	0.52	0.52	0.52
KNN	0.87	0.87	0.87
Decision Trees	0.773	0.884	0.75
Gradient Boosting Classifier	0.83	0.83	0.83
Random forest	0.86	0.86	0.86



This model can be fed the real time data that MARTA has, and in turn it will be able to predict if busses are arriving on time, giving its riders a better read on the arrival times so that they can plan accordingly. The model is still not perfect and preferably for future work, the model will sperate early & late arrivals.

This can definitely serve as a step towards the right direction for the relationship between MARTA & its riders. Providing a better service & customer satisfaction will always be a goal that any public entity.