**NORTHEASTERN UNIVERSITY**

**ALY6040 – Data Mining Applications**

**Module 6**

**Final Report**



**Date:** 06/28/2023

**Instructor**: Prof. Samarasinghe Kasun

**Group 5 Team Members**

Vartika Joshi – [joshi.vart@northeastern.edu](mailto:joshi.vart@northeastern.edu) (002760320)

Pardasaradhi Adapa – [adapa.p@northeastern.edu](mailto:adapa.p@northeastern.edu) (002770618)

Sayali Rajaram Kolte – [kolte.sa@northeastern.edu](mailto:kolte.sa@northeastern.edu) (002669469)

Karan Rajnikant Patel – [patel.karanr@northeastern.edu](mailto:patel.karanr@northeastern.edu) (002681342)

**INTRODUCTION**

In a comprehensive study of the Boston Bluebike dataset, we conducted an extensive Exploratory Data Analysis (EDA), uncovering key usage patterns and behaviors of the bike-sharing system. Our findings revealed that certain stations and time slots exhibited higher usage, notably the night times and weekends, with MIT at Mass Ave / Amherst emerging as the most frequented station.

In addition to EDA, we developed predictive models to estimate peak hour trips, leveraging machine learning algorithms like Decision Tree, Random Forest, Gradient Boosting, and Logistic Regression. Each model showed modest predictive capabilities, emphasizing the need for improved modeling strategies, feature engineering, and consideration of additional factors that may influence bike usage. This study provides a foundational understanding of the Boston Bluebike system and sets the groundwork for enhanced decision-making and strategic planning in bike-sharing systems.

**EXPLORATORY DATA ANALYSIS**

The analysis of the Boston Bluebike dataset involved several steps to gain insights into the usage patterns and characteristics of the bike-sharing system. Here is a summary of the EDA process and the findings -

**Dataset Entries:**

The dataset consists of 1,048,575 rows and 14 columns.

**Data Quality:**

Several aspects regarding data quality were discovered during the investigation -

* Missing Data:

The "postal code" column contains 110,793 missing entries, while the rest of the columns have no missing values.

* Duplications:

There are no duplicate rows in the dataset, ensuring the uniqueness of each observation.

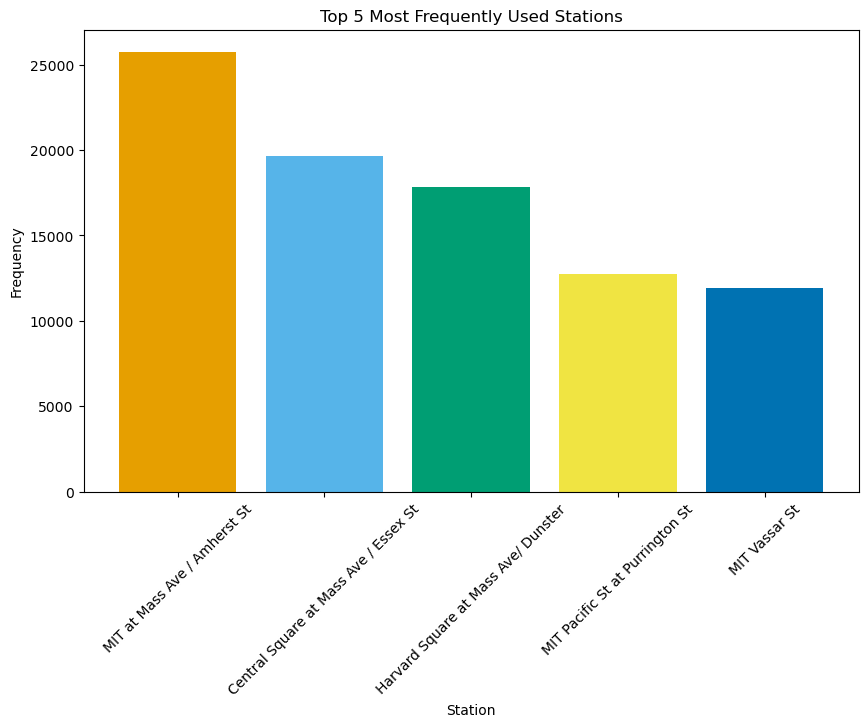
* Outliers and Suspicious Data:

The "tripduration" column exhibits numerous outliers, indicating unusually long or short trips compared to the majority. Outliers in trip duration can be attributed to user behavior or special events.

**Questions Explored and insights from the EDA:**

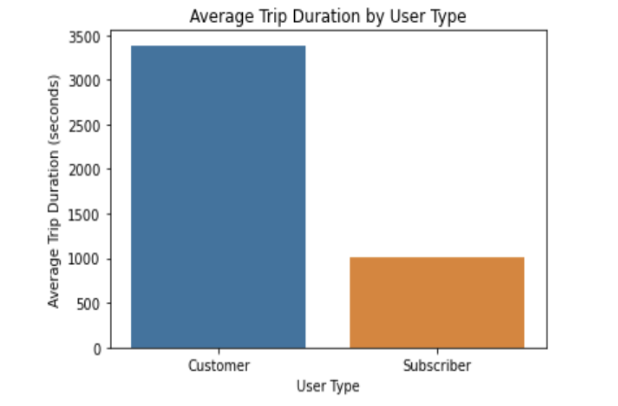
**1. What are the top 5 most frequently used stations?**

The below bar chart illustrates the top 5 most frequently used stations in the Bluebikes dataset.



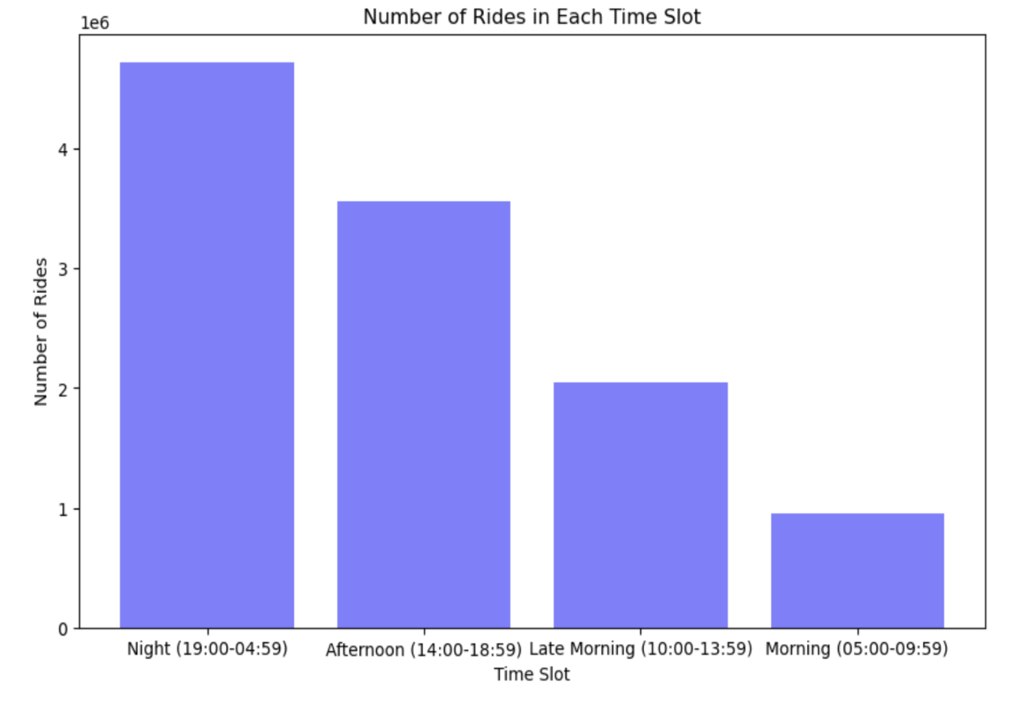
**2. What is the average trip duration for subscribers and customers?**

Regular customers have an average trip duration of approximately 3500 seconds, while subscribers' average trip duration is around 1000 seconds.



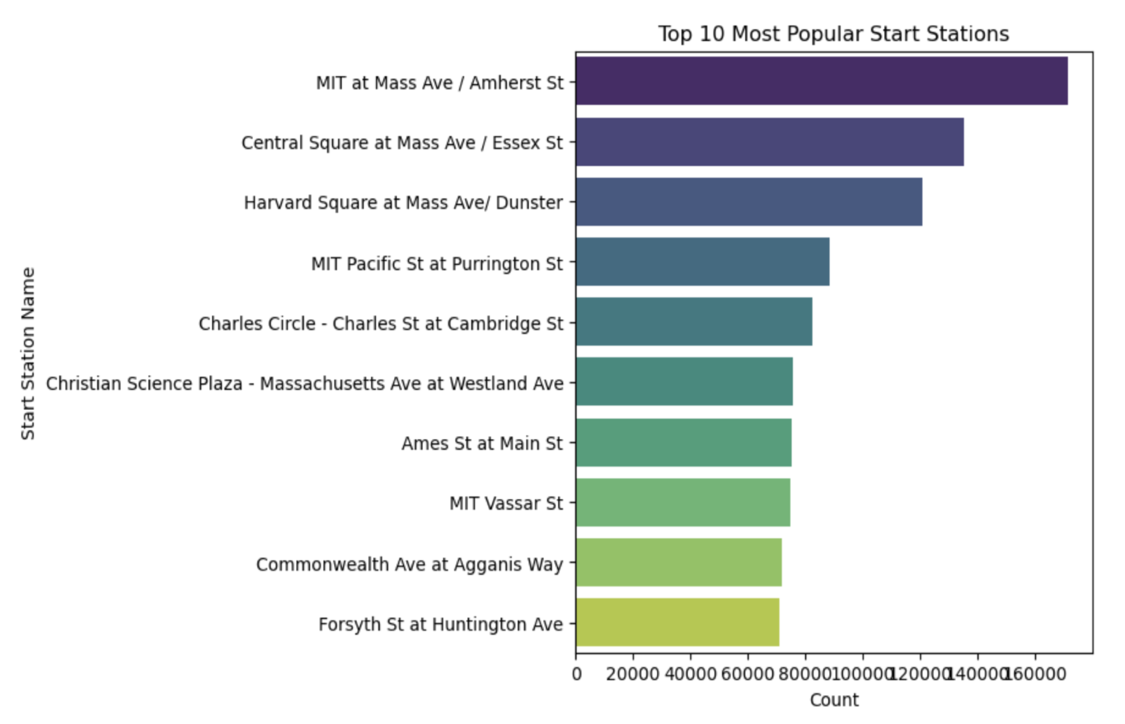
**3. Which time slot is the most popular for bike rides?**

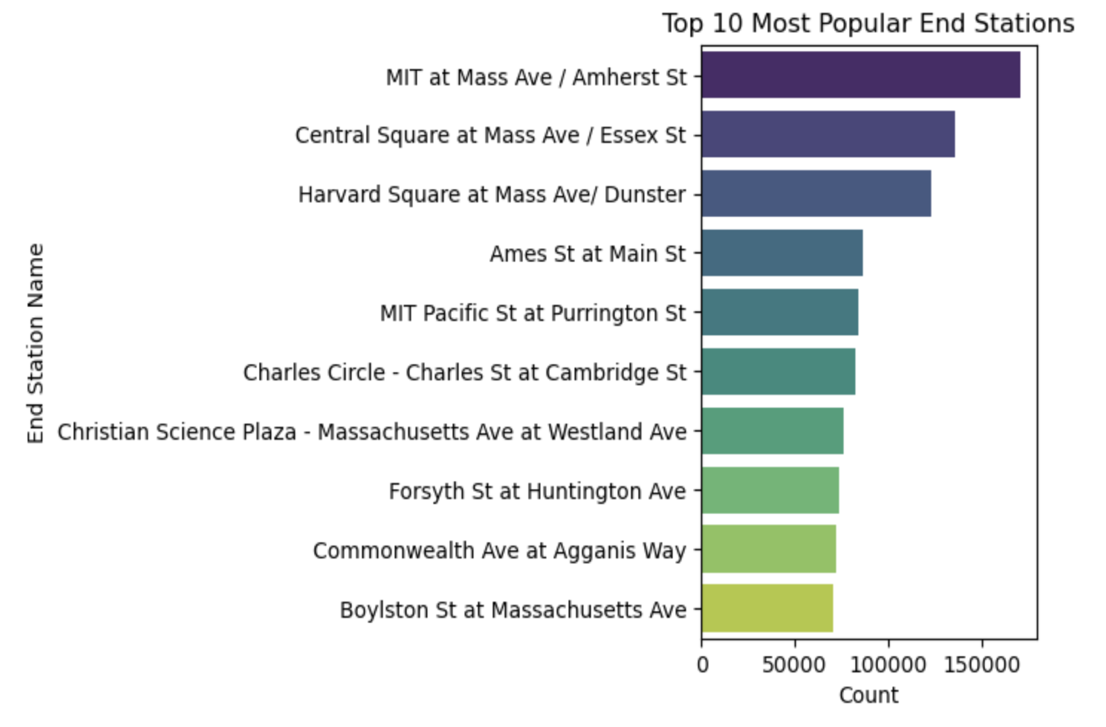
The bar plot of time slots reveals that the night slot, from 19:00 to 5:00, is the most popular time for bike rides.



**4. What are the most popular start and end stations?**

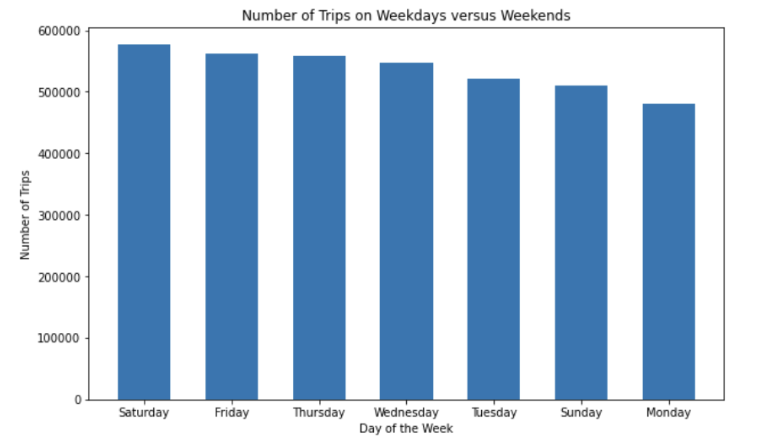
MIT at Mass Ave / Amherst emerges as the most popular start and end station, primarily due to its usage by MIT students.





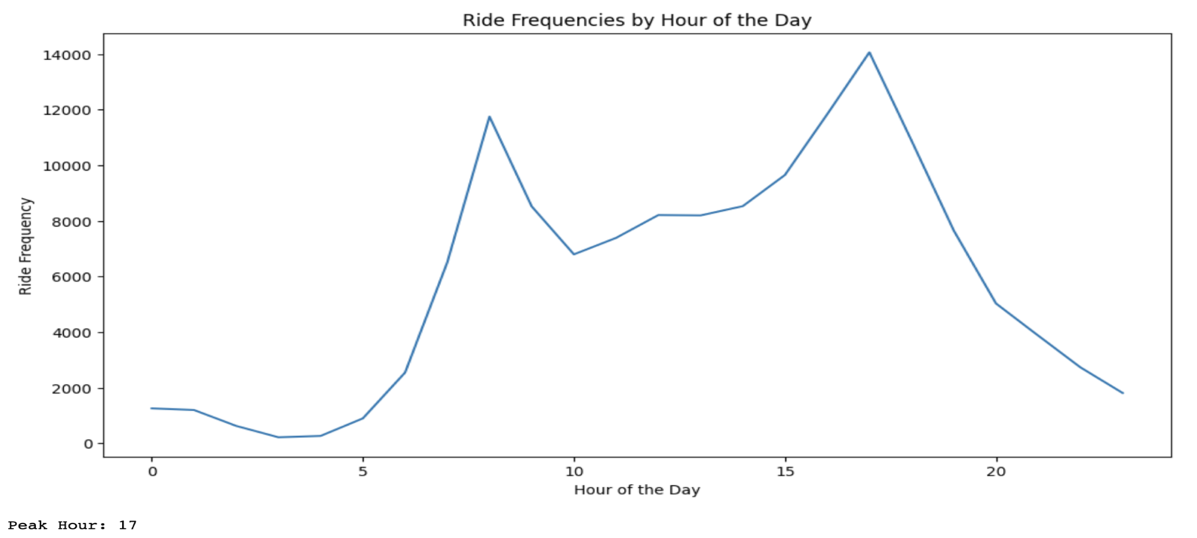
**5. How does the usage of blue bikes vary on different days of the week?**

Saturday and Friday exhibit the highest number of recorded trips, while Monday records the lowest. This indicates varying patterns of bike usage throughout the week.



**6. What are the peak hours of ride frequencies for Boston Blue Bikes?**

The peak hour of ride frequencies is 17:00 (5:00 PM), suggesting the highest number of rides occur during this hour. Identifying peak hours helps optimize resource allocation and maintenance schedules.



**MODEL DEVELOPMENT**

Following the EDA, we developed one predictive model: a classification model to predict peak hour trips.

Peak Hour Prediction: Several machine learning techniques were used to determine the most effective model for predicting whether a trip will begin during peak hours. The goal was to use these algorithms to extract information from the data and then use that information to generate precise predictions.

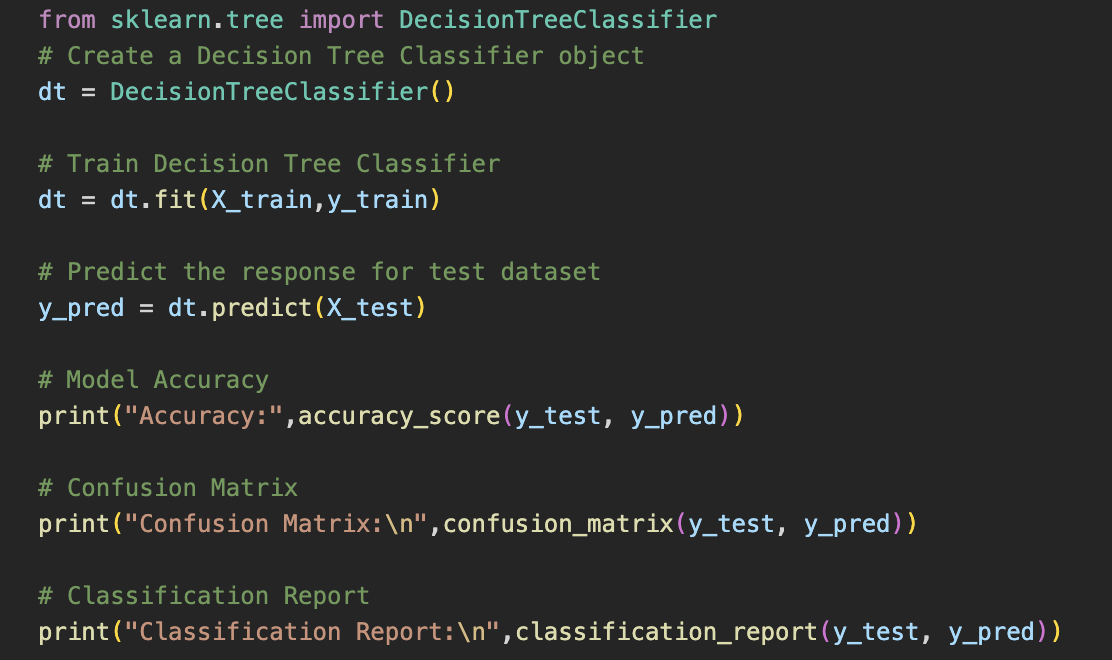
Decision Tree, Random Forest, Gradient Boosting, and Logistic Regression models were among those put to the test. Each team member used and tested their own model, and the outcomes were then compared to determine which model was the most successful.

**DECISION TREE MODEL**

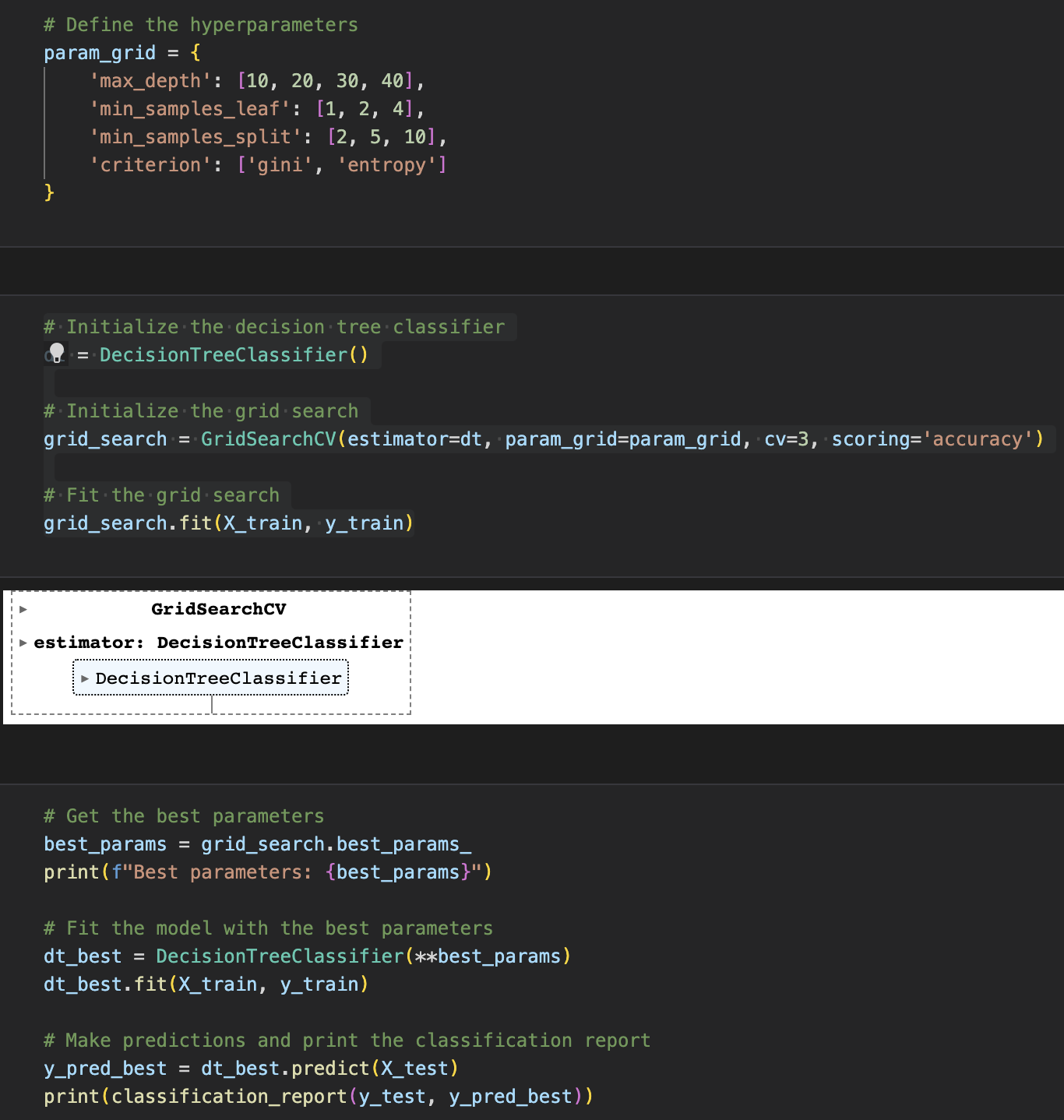
Decision tree model was chosen due to its ability to handle both categorical and numerical data, as well as its interpretability. Our goal was to find a model that not only performs well but also provides insights into the factors influencing bike usage during peak hours.

To analyze the dataset, we utilized the Decision Tree Classifier from the sklearn library. Decision Trees are non-parametric supervised learning methods used for classification and regression tasks.

In the first part of the code, a Decision Tree classifier was trained on the dataset (X\_train and y\_train) using default parameters. The trained model was then used to predict class labels (y\_pred) for the test dataset (X\_test). The evaluation metrics included accuracy, a confusion matrix, and a classification report containing precision, recall, and F1-score.



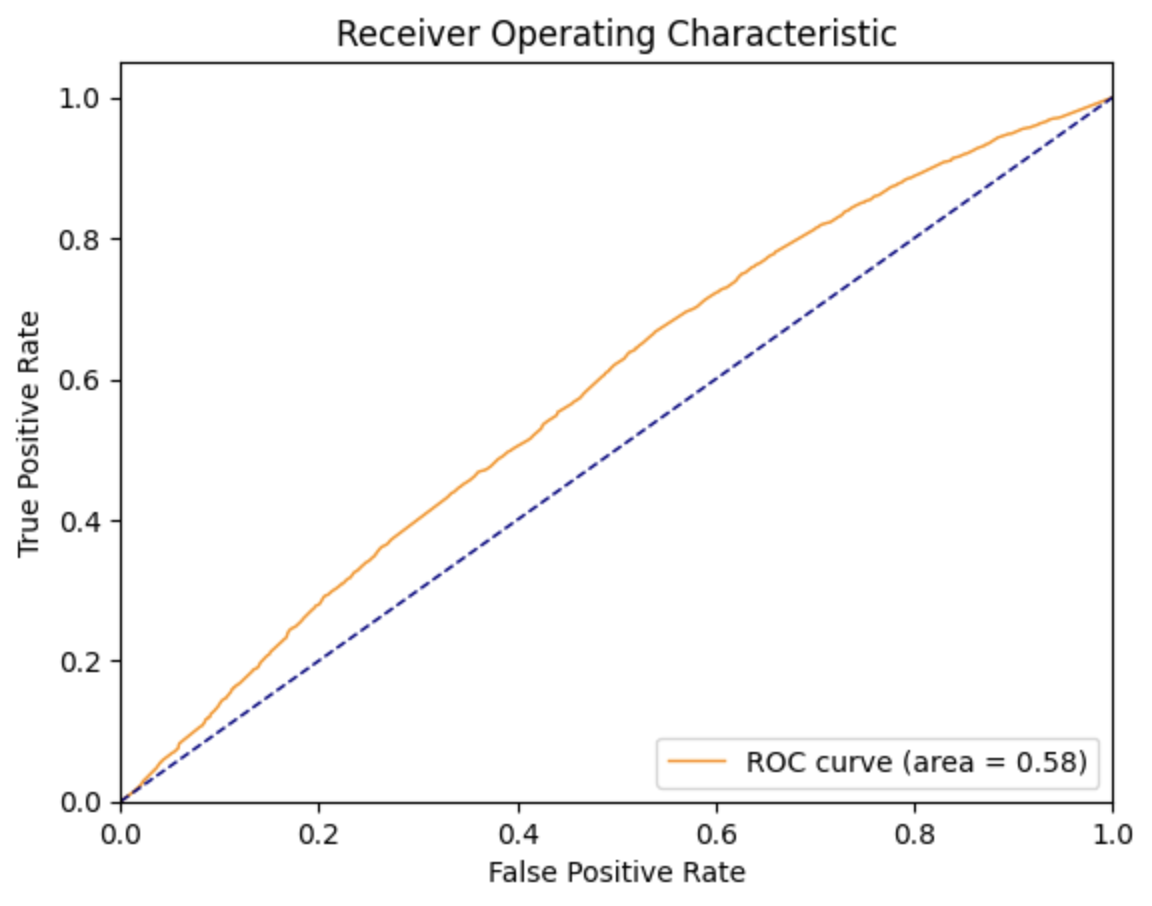
The second part of the code involved hyperparameter tuning for the model. A dictionary of different hyperparameters (param\_grid) was defined, including 'max\_depth', 'min\_samples\_leaf', 'min\_samples\_split', and 'criterion'. We performed a grid search using GridSearchCV to find the best combination of hyperparameters for the model. The optimal parameters were used to train a new Decision Tree classifier, and its performance was evaluated using the same metrics as before.



The initial Decision Tree model, without hyperparameter tuning, achieved an accuracy of approximately 57.16%. The classification report revealed that the model had better performance in predicting class 0, with higher precision, recall, and F1-score compared to class 1.

After performing hyperparameter tuning, the best parameters for the Decision Tree model were determined to be 'gini' for criterion, 30 for max\_depth, 1 for min\_samples\_leaf, and 5 for min\_samples\_split. However, the new model trained with these optimal parameters did not significantly improve the performance. The accuracy remained nearly the same, and the performance metrics for each class did not show noticeable changes.

ROC curve:



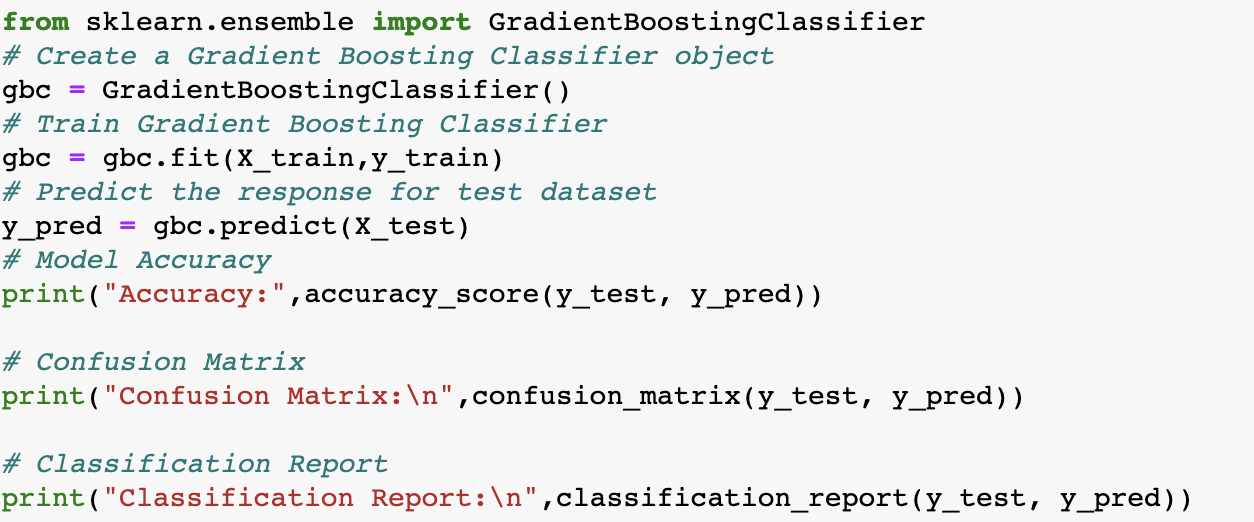
The ROC curve demonstrated that the Decision Tree model's performance in predicting peak hour bike trips based on the provided attributes (start station, user type, and day of the week) was modest, with an accuracy of around 58%. This suggests that the model was essentially making random predictions. It is possible that the attributes used did not fully capture the complexity of predicting peak hour bike trips.

To improve the model's performance, several approaches can be considered. This includes incorporating more advanced models, conducting additional feature engineering, or including more relevant features in the training process. Furthermore, gaining a deeper understanding of the problem's context could also lead to better results. The current Decision Tree model serves as a starting point for these enhancements.

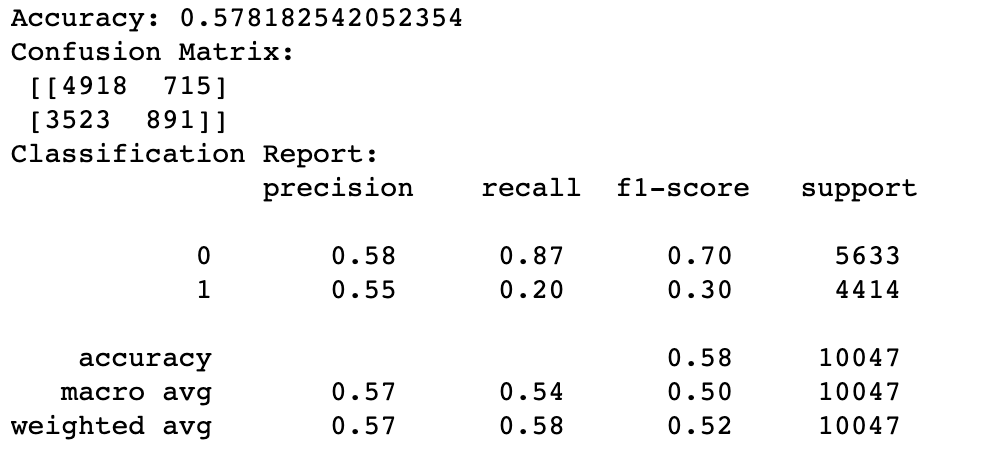
**GRADIENT BOOSTING MODEL**

Gradient Boosting is a powerful machine learning technique that combines multiple weak predictive models, such as decision trees, to create a strong ensemble model. It iteratively trains new models to target the mistakes made by preceding models.

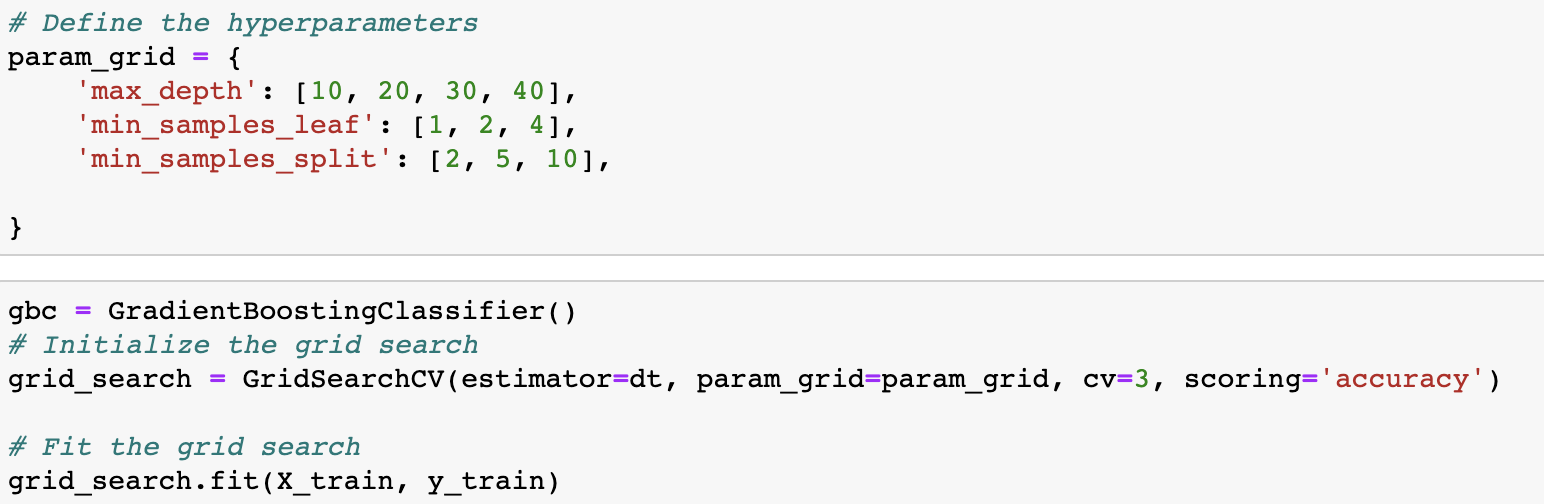
The provided code focuses on the use of the Gradient Boosting Classifier (GBC), which is a supervised learning method for classification and regression tasks. The code begins by training the GBC model using the training data (X\_train and y\_train). Once the model is trained, it is used to predict the responses for the test dataset (X\_test), and the accuracy of the model is calculated.



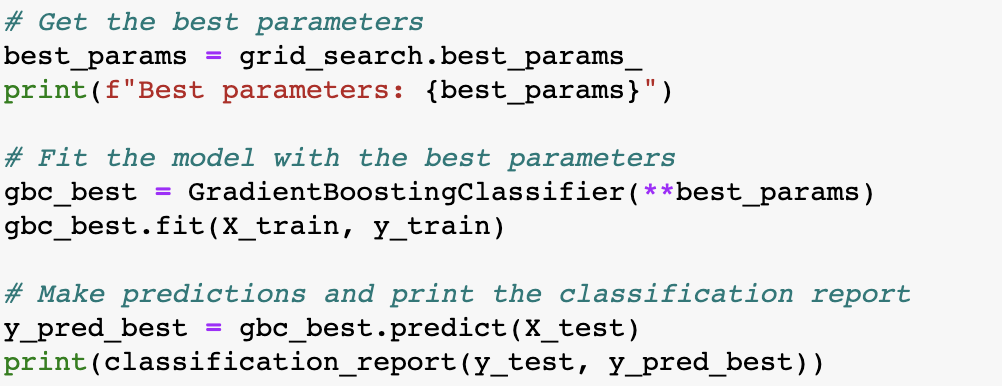
To evaluate the model, precision, recall, F1 score, and accuracy are calculated to measure its ability to correctly classify peak hour trips. Precision indicates how well the model predicts peak hour trips, while recall tells us how many actual peak hour trips it captures. F1 score combines precision and recall into a balanced metric, and accuracy provides an overall measure of the model's correctness. The Gradient Boosting model achieves a modest performance in predicting peak hour bike trips, with an accuracy of approximately 57.81%.



A grid search is performed to find the best hyperparameters for the GBC. The hyperparameters considered in the grid search are max\_depth, min\_samples\_leaf, and min\_samples\_split. The grid search evaluates different combinations of these hyperparameters using cross-validation and selects the best combination based on accuracy.

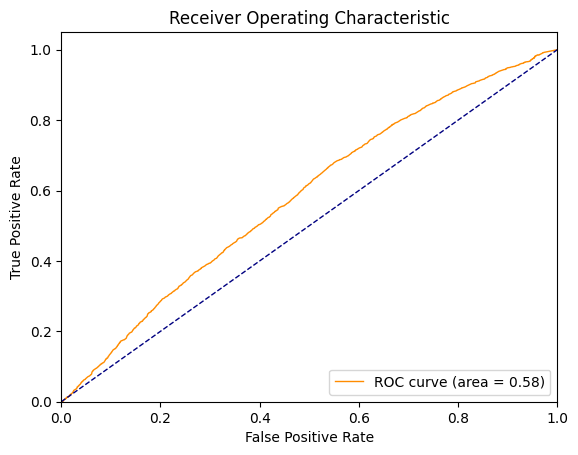


The best parameters found through the grid search are then used to create a new GBC model, which is trained on the training data. Predictions are made on the test data using this model, and the classification report is printed again to compare the performance with the initial model.



ROC Curve:

The analysis of the Gradient Boosting Classifier (GBC) includes the ROC curve, which visualizes the changes in the true positive rate and false positive rate as the classification threshold is adjusted. The GBC achieves an AUC value of 0.58, indicating a moderate level of discriminatory power. This suggests that the model has some ability to accurately classify rush hour and non-rush hour rides, although there is potential for improving its predictive performance.



**RANDOM FOREST MODEL**

We utilized the random forest algorithm to predict peak hour bike trips. To handle categorical variables, we employed one-hot encoding to create dummy variables.

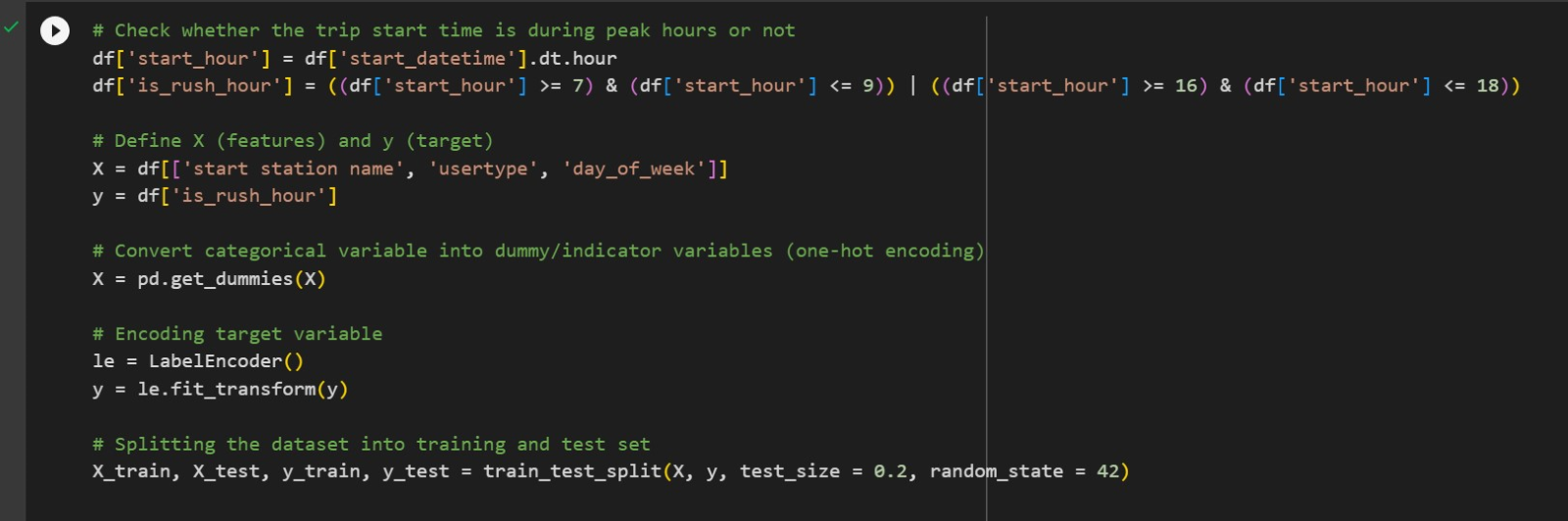


Figure 1: Before performing hyperparameter tuning:

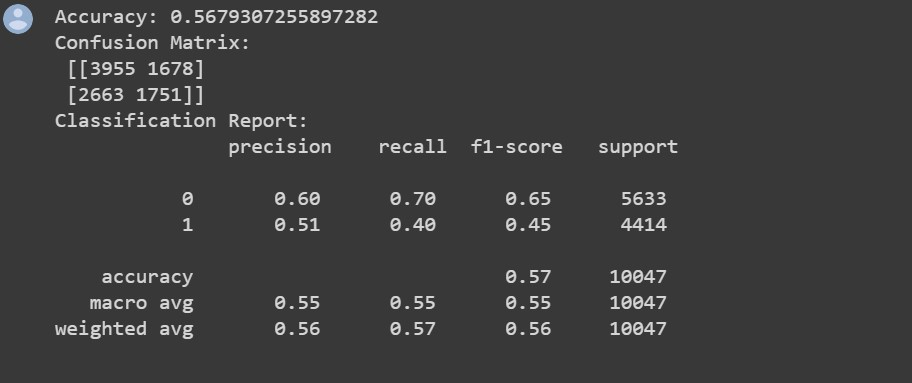


Figure 1 shows the results before hyperparameter tuning. The initial random forest model achieved an accuracy of 0.568, correctly predicting the rush hour/non-rush hour classification for approximately 56.8% of the instances in the test dataset. The confusion matrix revealed that out of 10,047 instances, 3,955 were correctly classified as non-rush hour, and 1,751 were correctly classified as rush hour. However, there were misclassifications, with 1,678 instances wrongly classified as rush hour and 2,663 instances wrongly classified as non-rush hour. The precision was 0.60 for non-rush hour predictions and 0.51 for rush hour predictions. The recall (sensitivity) was 0.70 for non-rush hour and 0.40 for rush hour. The overall F1-score was 0.56, indicating a fair balance between precision and recall.

Figure 2: After performing hyperparameter tuning:

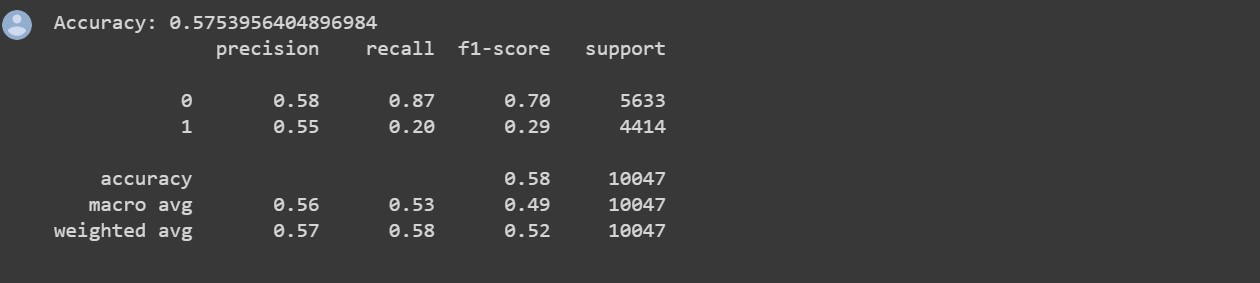


Figure 2 shows the results after hyperparameter tuning. The refined random forest model demonstrated slight improvement with an accuracy of 0.575. The precision for non-rush hour predictions increased to 0.58, while the precision for rush hour predictions increased to 0.55. However, the recall for rush hour instances remained low at 0.20, while the recall for non-rush hour instances improved to 0.87. The F1-score was 0.70 for non-rush hour and 0.29 for rush hour.

Figure 3: ROC Curve:

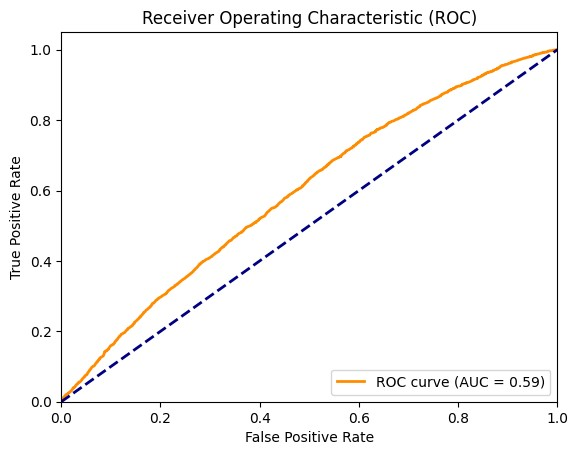


Figure 3 displays the ROC curve, which illustrates the model's ability to differentiate between rush hour and non-rush hour bike rides. The curve demonstrates that as the classification threshold varies, the true positive rate and false positive rate change accordingly. The AUC value of 0.59 suggests that the model has a moderate level of discriminatory power.

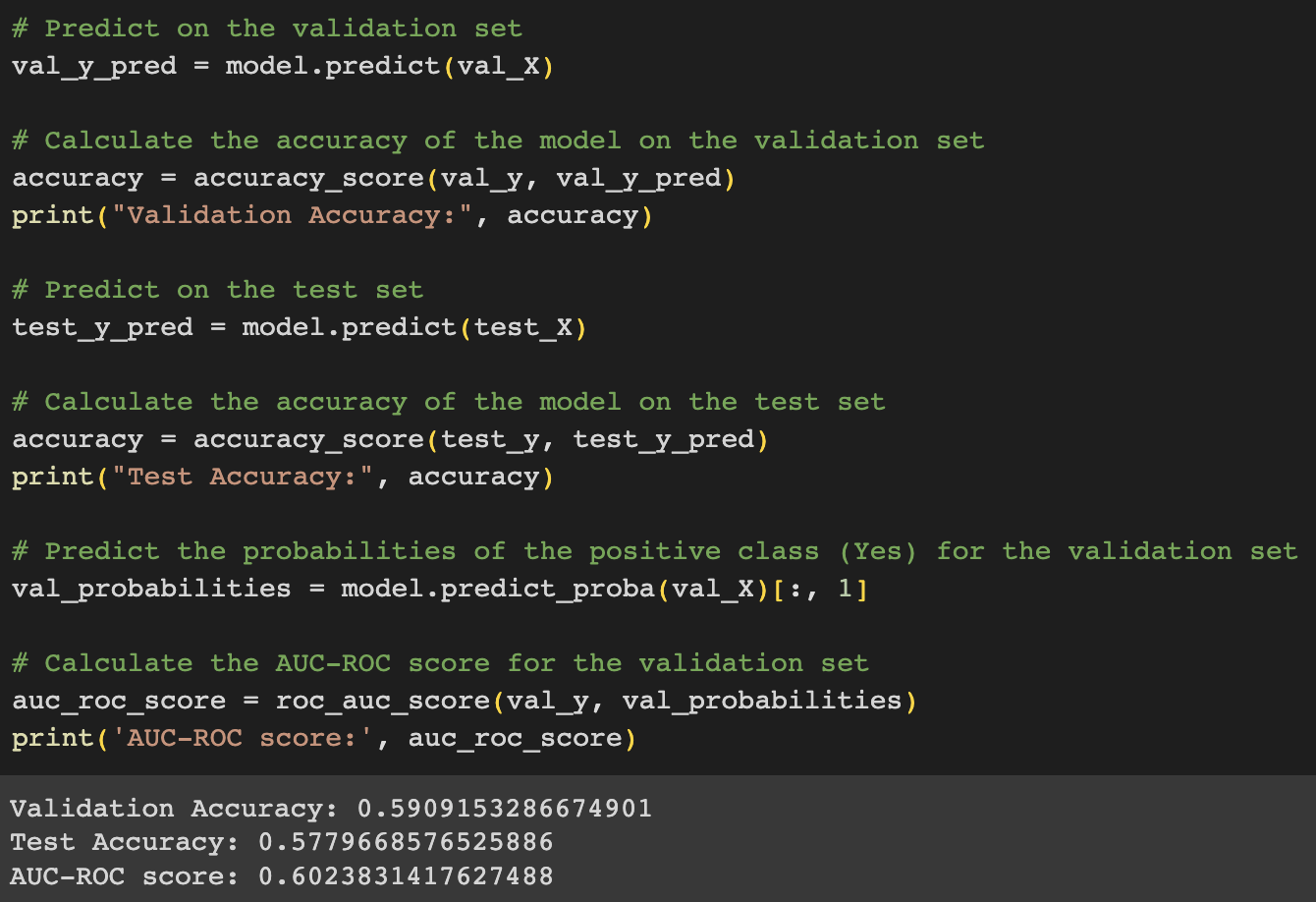
**LOGISTIC REGRESSION**

Data Preprocessing:

In the initial steps of the analysis, the dataset was preprocessed by separating the start and stop times into date and time components. The day of the week was extracted, and a binary variable called "peak\_hour\_trip" was created to indicate whether a trip started during peak hours (7-9 AM and 4-6 PM), with 1 representing "Yes" and 0 representing "No".

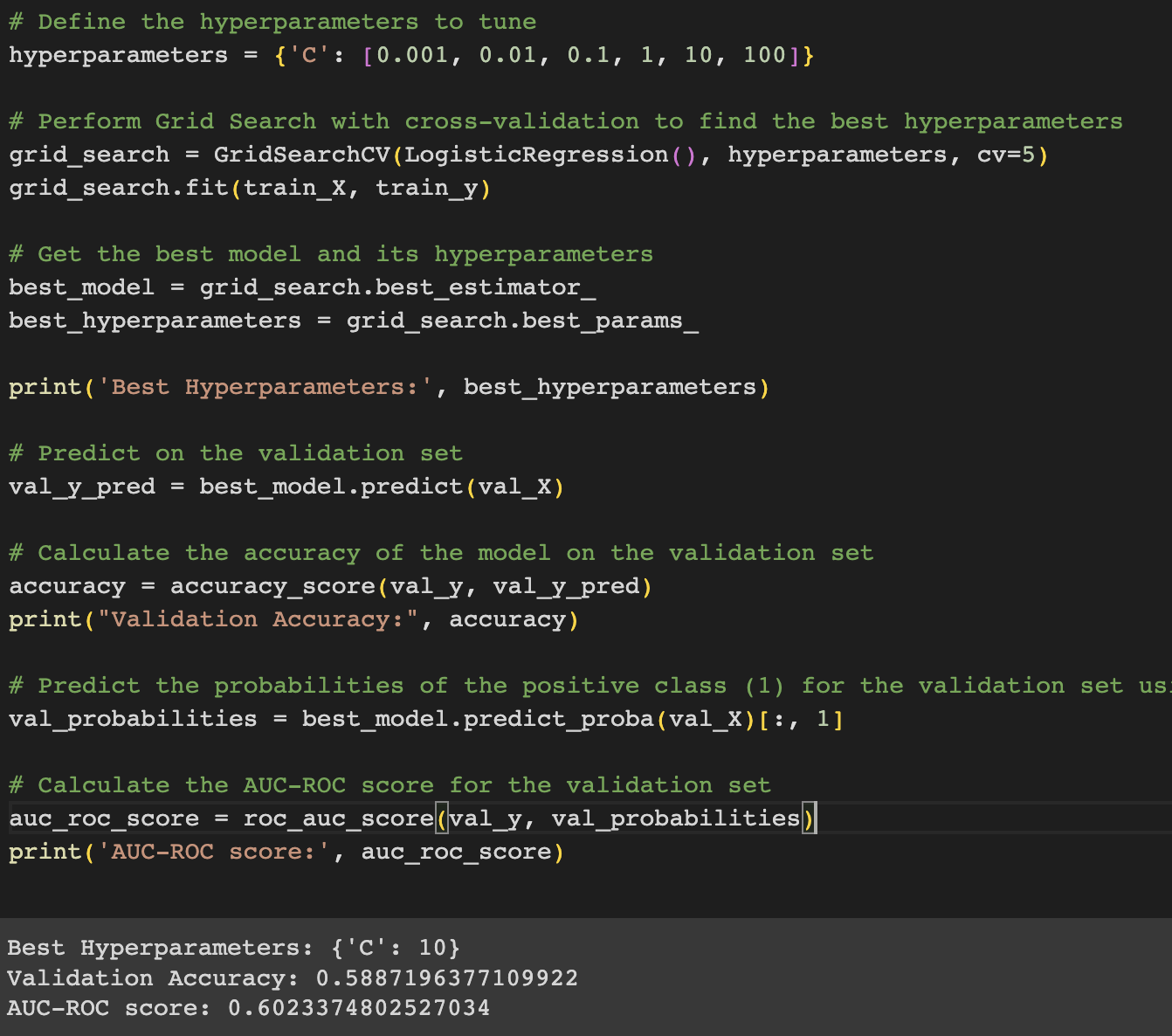
Model Training and Validation:

A logistic regression model was trained on the training set and its performance was evaluated on the validation set. The accuracy of the model on the validation set was calculated as 0.5909, indicating that approximately 59.09% of the samples were correctly classified.



Hyperparameter Tuning:

To further enhance the model's performance, hyperparameter tuning was conducted using GridSearchCV. Different values of the regularization parameter 'C' (0.001, 0.01, 0.1, 1, 10, and 100) were tested, and the optimal hyperparameter value was found to be 'C=10'. This parameter controls the inverse of the regularization strength in logistic regression.



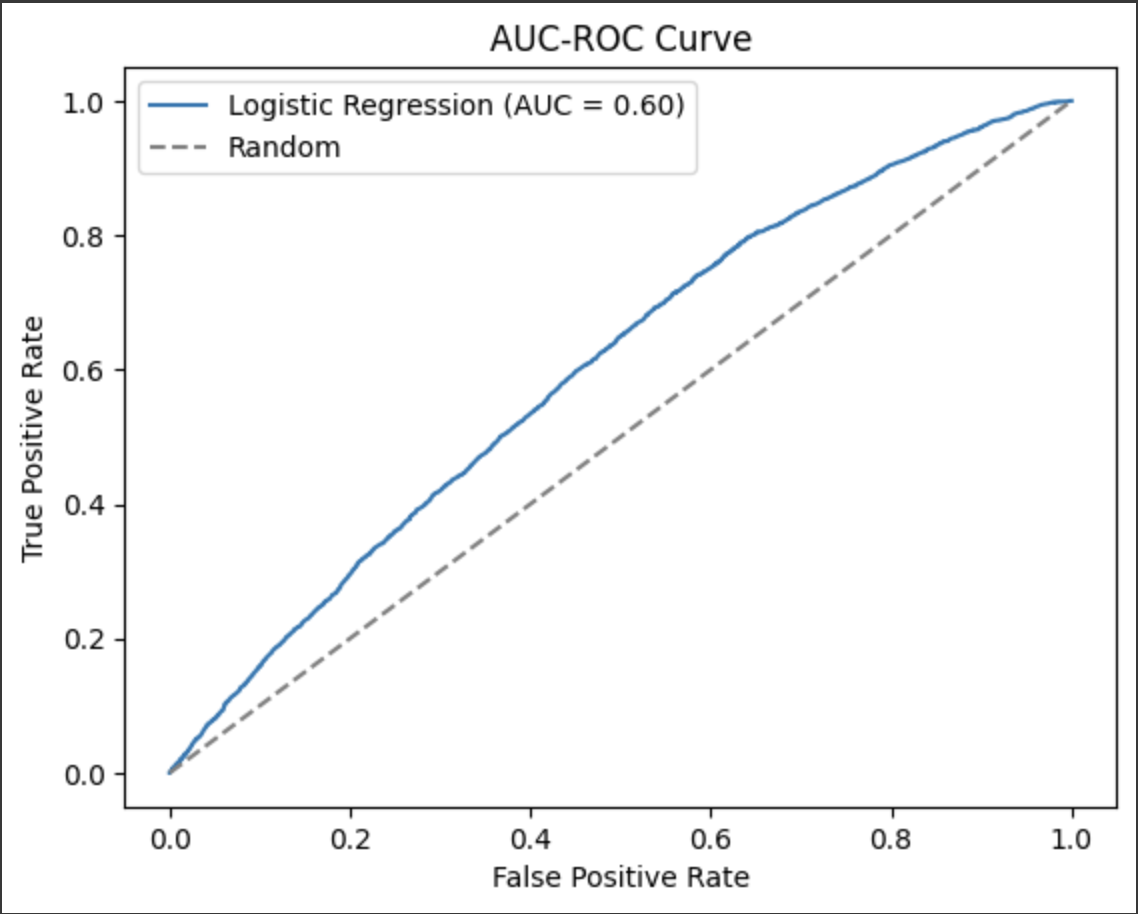
Impact on Accuracy:

After hyperparameter tuning, the model's accuracy on the validation set was evaluated again, resulting in a slight decrease to 0.5887. The hyperparameter tuning did not have a significant impact on the accuracy of the model.

Interpretation and ROC Curve:

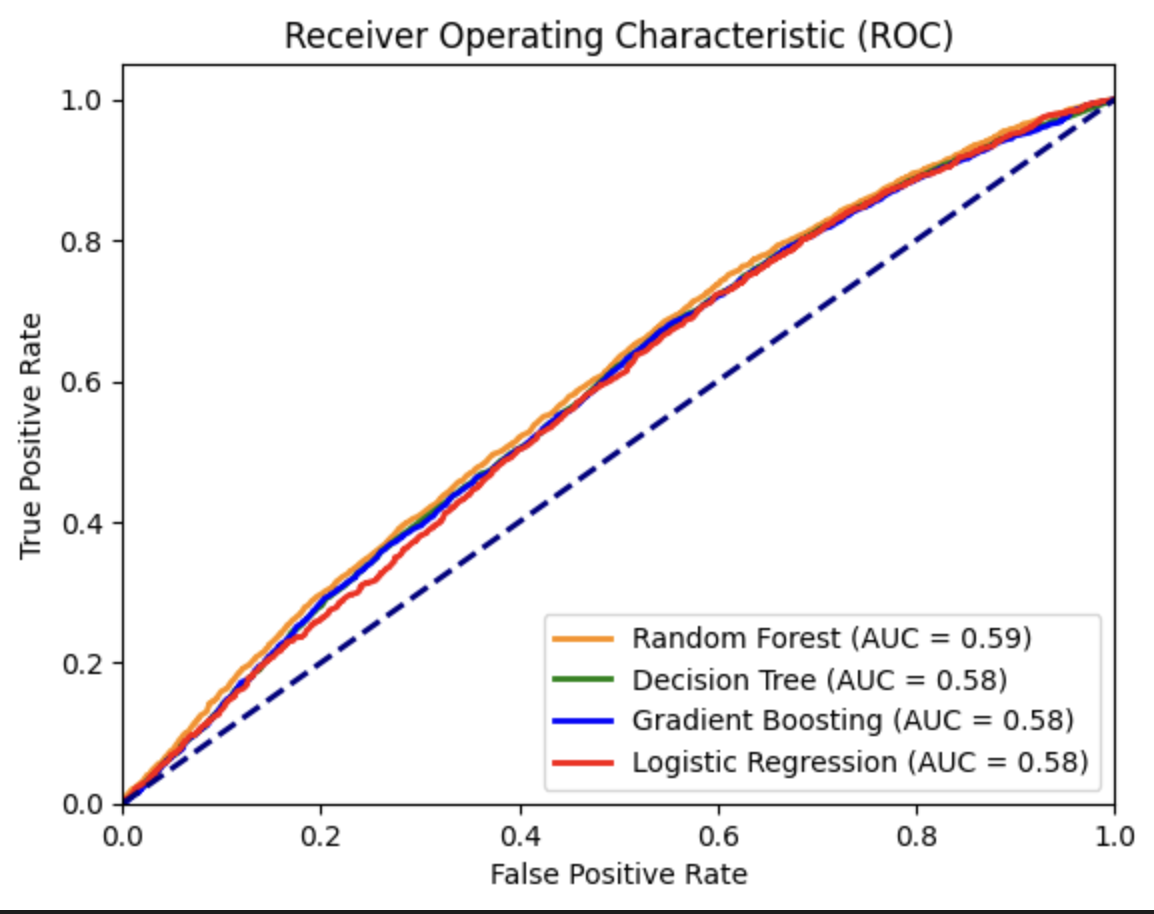
To assess the model's performance in terms of the trade-off between true positive rate and false positive rate, the area under the ROC curve (AUC-ROC) was calculated. The AUC-ROC score for the best model on the validation set was determined to be 0.6023, indicating a moderate level of discrimination ability.

The ROC curve visually represents the performance of the logistic regression model. The plot illustrates the trade-off between the true positive rate (sensitivity) and the false positive rate (1-specificity). The ROC curve for the logistic regression model is displayed, with an AUC value of 0.602, slightly above the random line. This suggests that the model has some predictive power, although there is still room for improvement.



**COMPARISION OF ROC CURVES**

AUC score between 0.58 and 0.59 indicates that the models have some ability to discriminate between rush hour and non-rush hour bike trips, but there is room for improvement.



**INTERPRETATIONS**

The analysis of the Blue Bikes dataset aimed to address the question of whether it is possible to predict whether a trip will start during peak hours based on the start station, user type, and day of the week. Several machine learning models were developed and evaluated for this prediction task, including Decision Tree, Random Forest, Gradient Boosting, and Logistic Regression models.

The initial Decision Tree model achieved an accuracy of approximately 57.16%. However, even after performing hyperparameter tuning, the model's performance did not significantly improve. The Gradient Boosting model achieved a modest performance with an accuracy of approximately 57.81%. The Random Forest model achieved an accuracy of 56.8% before hyperparameter tuning and a slight improvement to 57.5% after tuning. The Logistic Regression model achieved an accuracy of approximately 59.09% on the validation set.

Overall, the results of the analysis suggest that predicting whether a trip will start during peak hours based on the provided attributes is challenging. The models developed in this analysis have demonstrated only modest predictive performance, with accuracies ranging from 56.8% to 59.09%.

As data owners, the results imply that predicting peak hour bike trips based on the provided attributes alone is challenging and may require additional data or more advanced modeling techniques. However, Logistic Regression emerges as the most accurate model among the others, making it the preferred choice for answering the question. This knowledge can aid in planning resource allocation and improving services for Blue Bikes users, enhancing operational efficiency and customer satisfaction.

**RECOMMENDATIONS**

Based on the EDA and model development findings, several recommendations can be made for future work:

* The analysis showed that certain stations and time slots have higher usage. The management could focus on improving infrastructure and service in those areas to cater to the demand.
* Missing data in the "postal code" column should be addressed. More information could be gained on user locations and travel patterns by ensuring that this data is consistently captured.
* Considering the outliers in the "trip duration" column, it would be useful to investigate these outliers in more detail to identify possible reasons behind these unusual durations.
* Given the lower performance of the models in predicting rush hour trips, more advanced models could be explored to increase predictive accuracy. Deep learning techniques or ensemble methods such as XGBoost may be beneficial.
* More features could be included in the models to enhance predictive power. Weather conditions, public events, or specific calendar days (holidays, weekends, etc.) may have an impact on bike usage.
* More feature engineering could be conducted to extract new information from existing variables. For instance, transforming the "start\_time" and "end\_time" into part of day (morning, afternoon, evening, and night) may provide additional insights.
* To address the issue of class imbalance as observed from the confusion matrices, techniques like oversampling, undersampling, or SMOTE can be used.

**PYTHON LIBRARIES AND TECHNIQUES EMPLOYED FOR DATA ANALYSIS**

**Libraries Used**

* Pandas: This library was primarily used for data manipulation and analysis. It offers data structures and operations for manipulating numerical tables and time series.
* NumPy: This library was used for numerical computations and operations on arrays of different dimensions.
* Matplotlib and Seaborn: These libraries were used for data visualization, creating bar plots, histograms, scatter plots, etc.
* Scikit-learn: This machine learning library was used for developing predictive models. Functions from this library were used for pre-processing data, splitting datasets, creating predictive models (Decision Tree, Random Forest, Gradient Boosting, and Logistic Regression), tuning hyperparameters (GridSearchCV), and evaluating model performance.
* SciPy: This library was used for scientific computations such as calculating summary statistics.

**Techniques Used**

* Data Cleaning: Missing values were identified and removed to ensure data quality. Outliers and suspicious data were explored.
* Data Preprocessing: Categorical variables were transformed using one-hot encoding. The dataset was split into a training set and a test set for model development. Feature scaling was done when necessary.
* EDA: Univariate and bivariate analyses were performed using graphical and non-graphical techniques. Trends, patterns, and relationships within the data were explored.
* Predictive Modeling: Decision Tree, Random Forest, Gradient Boosting, and Logistic Regression models were trained on the dataset. Model performances were evaluated using accuracy, precision, recall, F1-score, and ROC-AUC scores. Hyperparameter tuning was performed using GridSearchCV to improve model performance.
* Model Evaluation: Confusion matrices were generated to evaluate model performance in terms of true positives, true negatives, false positives, and false negatives. ROC curves were also drawn to evaluate the trade-off between true positive rate and false positive rate.

**CONCLUSIONS**

The exploratory data analysis provided valuable insights into bike usage patterns and user behavior, revealing the most popular times, days, and locations for bike usage. However, predicting peak hour trips proved to be a challenging task, with all models performing moderately well, having AUC scores between 0.58 and 0.59.

The models provided a starting point for understanding the complexities of the task, but further improvements are required to increase their predictive performance. Implementing the recommendations above could help enhance these models, contributing to more effective decision-making and strategic planning for the Boston Bluebike system.

**REFERENCES**

[1] Blue Bikes Boston. (n.d.). <https://www.bluebikes.com/blog/the-data-challenge-entries>

[2] Motivate International, Inc. (n.d.). Bluebikes System Data. Blue Bikes Boston. <https://www.bluebikes.com/system-data>

[3] Bluebikes. Wikipedia. (2023, April 17). <https://en.wikipedia.org/wiki/Bluebikes>

[4] Working with missing data in Pandas. GeeksforGeeks. (2023, February 9). <https://www.geeksforgeeks.org/working-with-missing-data-in-pandas/>