A red and white logo

Description automatically generated

**Data Mining Applications**

College Of Professional Studies, Northeastern University, Boston.

ALY 6040

Professor Ajit Appari

May 18th, 2024

**INTRODUCTION:**

This assignment aims to work on data regarding bicycle ride records to predict long-ride duration using different machine-learning approaches. This dataset includes bike ride records with ride duration, rideable type, season, and user characteristics, among other details. The major objective here is to build a binary classification model, or a model that will identify whether a ride will likely last more than 2 hours, thereby trying to associate patterns of the factor related to a long bike ride.

It all begins with data cleaning—handling missing values and removing outlier values to preserve the quality of the information. In this regard, we will now have a closer look at the dataset by visualizing it to unleash insights into the ride patterns across features, user segmentation, and period. Consequently, we create long rides as a binary variable and standardize the numeric predictors.

We perform class imbalance by manual oversampling of the minority class. We split the dataset into training and testing sets where the model was well-trained. Support Vector Machine (SVM) and Random Forest classifiers were used to predict long-ride duration. Cross-validation was done to ensure the performance of the model and its robustness.

The purpose of this assignment is to derive actionable insights that bike-sharing companies can implement to understand their users and, in turn, improve services to these important factors. Model outcomes include performance metrics and visualizations of the importance of the factors influencing long rides, demonstrated by the effectiveness of predictive models.

**METHODOLOGY:**

We first move forward with the basic EDA where we preprocess bike ride data for machine learning. It begins by sampling 5% of the data and handling missing values. Outliers are removed using the IQR method. A binary variable long\_ride is created to indicate rides longer than 120 minutes. Non-relevant identifiers are removed, and numeric predictors are standardized. If the minority class (long\_ride = 1) is absent, it is artificially introduced. The data is split into training (70%) and testing (30%) sets. The oversample\_minority function balances the training data by oversampling the minority class, ensuring both classes are equally represented for modeling.

*Visualizations:*

**PLOT NUMBER 1:**

A screenshot of a graph

Description automatically generated

• This bar plot shows the distribution of ride lengths by membership type. The x-axis represents ride length (standardized), and the y-axis represents the count of rides. The teal bars indicate members, while the red bars indicate casual riders. The plot highlights a higher frequency of rides among members than casual riders.

**PLOT NUMBER 2:**

A diagram of different colored squares

Description automatically generated

•This box plot shows the distribution of ride lengths by rideable type and membership. The x-axis categorizes rideable types (classic, docked, and electric bikes), and the y-axis represents ride length in minutes. The plot is faceted by membership type (casual and member). Each box plot displays the median (black line within the box), interquartile range (box), and potential outliers (dots). Casual riders show higher variability in ride lengths, especially for docked bikes, compared to members.

**PLOT NUMBER 3:**

A screenshot of a computer screen

Description automatically generated

•This heatmap visualizes the interaction between the season and the part of the day. The x-axis represents different seasons (Autumn, Spring, Summer, Winter), while the y-axis represents parts of the day (Night, Morning, Evening, Afternoon). The color intensity indicates the number of rides, with red representing the highest frequency and blue the lowest. The plot shows that summer afternoons and evenings have the highest number of rides, while winter nights have the fewest.

**PLOT NUMBER 4:**

A screenshot of a graph

Description automatically generated

• This histogram shows the distribution of ride lengths across different parts of the day (Afternoon, Evening, Morning, Night). The x-axis represents ride length in minutes (standardized), and the y-axis shows the frequency of rides. Each subplot represents a different part of the day. Afternoon and evening have the highest number of rides, followed by morning, while night has the least. The ride lengths appear to be consistently distributed across these periods, with a similar pattern in each subplot.

• Checking for multicollinearity using VIF:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | **GVIF Df GVIF^(1/(2\*Df))** |  |
| **ride\_length\_minutes** |  | **1.329186** | **1** | **1.152903** |
| **num\_rides\_past\_month** |  | **1.154169** | **1** | **1.074322** |
| **rideable\_type** |  | **1.158839** | **2** | **1.037542** |
| **season** |  | **2.569152** | **3** | **1.170303** |
| **part\_of\_day** |  | **2.354031** | **3** | **1.15337** |

•Decision tree:

This decision tree visualizes the classification of bike rides into long rides (1) and not long rides (0). The tree splits based on features like num\_rides\_past\_month, season, part\_of\_day, rideable\_type, and ride\_length\_minutes. Each node shows the predicted class, the probability of the class, and the percentage of data points reaching that node. The root node splits on num\_rides\_past\_month, indicating its importance in predicting long rides. Subsequent splits refine the predictions, highlighting interactions between various features and ride lengths. Terminal nodes (leaves) provide final classifications with associated probabilities and sample percentages.

**PLOT NUMBER 5:**

A diagram of a company

Description automatically generated

**PLOT NUMBER 6:**

This bar plot displays the feature importance of a decision tree model. The x-axis represents the importance score, while the y-axis lists the features: num\_rides\_past\_month, ride\_length\_minutes, season, part\_of\_day, and rideable\_type. num\_rides\_past\_month has the highest importance, indicating it is the most influential feature in predicting long rides, followed by ride\_length\_minutes and other features. This plot helps in understanding which variables most significantly impact the model's predictions.

**A screenshot of a computer

Description automatically generated**

•Random forest:

Random forest accuracy: "Random Forest Accuracy: 0.997709083335813"

The Random Forest model achieved an accuracy of 99.77%. However, the confusion matrix reveals potential overfitting. Despite the high accuracy, the model incorrectly predicted 151 instances of long\_ride = 0 as long\_ride = 1 and failed to predict any true positives for long\_ride = 1. This suggests that the model might be overfitting to the majority class (long\_ride = 0) and not generalizing well to the minority class (long\_ride = 1). To mitigate overfitting, consider adjusting class weights, further tuning hyperparameters, or using a more balanced dataset.

•Confusion matrix:

|  |  |  |
| --- | --- | --- |
|  | **Actual** |  |
| **Predicted** | **0** | **1** |
| **0** | **67068** | **3** |
| **1** | **151** | **0** |

•SVM model:

The SVM model, using a linear kernel, was trained on 70% of the bike ride data to predict whether rides would be long (greater than 120 minutes). The model's performance was evaluated on the remaining 30% of test data. The accuracy was calculated, and a confusion matrix was generated. The confusion matrix shows the distribution of true positives, true negatives, false positives, and false negatives, allowing assessment of the model's classification performance and any potential issues with class imbalance or misclassification.

SVM accuracy: "SVM Accuracy: 0.999955371753295"

The SVM model achieved an accuracy of 99.99%, meaning that it correctly predicted the class for almost all instances in the test set. This high accuracy suggests that the model fits the training data extremely well. Majority Class Bias: The model predicts almost all instances as 0 (non-long rides), ignoring 1 (long rides), thus achieving high accuracy by favoring the majority class.

Confusion matrix:

|  |  |  |
| --- | --- | --- |
|  | **Actual** |  |
| **Predicted** | **0** | **1** |
| **0** | **67219** | **3** |
| **1** | **151** | **0** |

A decision tree is chosen for this project over Random Forest and SVM because of its ease of use and interpretability. Decision trees offer lucid visual aids that facilitate comprehension of the decision-making process about many attributes. For stakeholders that need to understand and believe in the model's results, this openness is essential. Decision trees may be used for fast, real-time decision-making since they require less computing power than Random Forest and SVM. Even though Random Forest and SVM have greater accuracy, situations where model interpretability and usability are crucial render them less useful due to their complexity and "black box" nature.

**CONCLUSION:**

This project aims to predict long bike rides, defined as rides with a duration greater than 120 minutes, using a set of method learning methods. This data deals with features like ride length, number of past rides, rideable type, season, and part of the day among others. Following full data cleaning, missing value treatment, and treatment of outliers, we create a binary target variable for long rides and manually over-sample our data to handle class imbalance.

Exploratory data analysis showed that members tend to ride more frequently than casual riders; distributional insights among different categories are put forward. Histograms, box plots, and heat maps are included to show the pattern of interactions between features like season, part of day, and ride length.

Three machine learning models were implemented: Decision Tree, Random Forest, and Support Vector Machine (SVM). The Decision Tree method was implemented because it is simple and interpretable. It can thus provide results in such a way that even stakeholders can easily understand and trust them. It showed all the important features influencing the ride length, such as the number of rides made in the last month and the type of rideable.

The prediction performance of Random Forest and SVM models was reasonably high; however, both clearly showed that there was overfitting, with almost perfect accuracy at 99.99% for the SVM model. Such differences in performance indicate that probably the models chosen were too complex for the respective dataset in question, leading to evident overfitting and a reduction in generalizability. More explicit evidence of the models being overfitted can be calculated from the resulting confusion matrices of these models. The Decision Tree model appeared to balance both accuracy and interpretability. In the end, it managed to identify what factors led to longer rides, which is highly valuable when gaining some insights while working on the improvement of bike-sharing services. More advanced balancing techniques, feature engineering, and different models under hyperparameter tuning might be the way forward in attaining the best-predicted performance and generalizability. The work here is, in fact, an apt example of how one would select models based on individual project needs, rather than making an a priori blanket statement over accuracy, interpretability, or computational efficiency.

**REFERENCES:**

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32. <https://doi.org/10.1023/A:1010933404324>

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32. <https://doi.org/10.1023/A:1010933404324>

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324