Project 3 Report
Christina Carbajal

Introduction and Overview

The data had mainly numerical data, but some columns were strings or included other symbols along with numbers. Some columns had NaN values and were therefore dropped from the set. For the data to run through the models, all the columns needed to be in numerical form. The data was preprocessed by various methods, such as encoding categorical values, binning, and removing all appearances of non-numerical symbols. After the set was cleaned and split into train and testing sets, it was used within a logistic model. To find the optimal model parameters, GridSearchCV was used and the resulting model was fitted to the set. The model was evaluated using common evaluation metrics, such as accuracy, precision score, recall score, f1 score, and a confusion matrix. The resulting model produced an AUC score, for both test and train sets, of .95. This process was repeated for the Gradient Boosting Classifier.

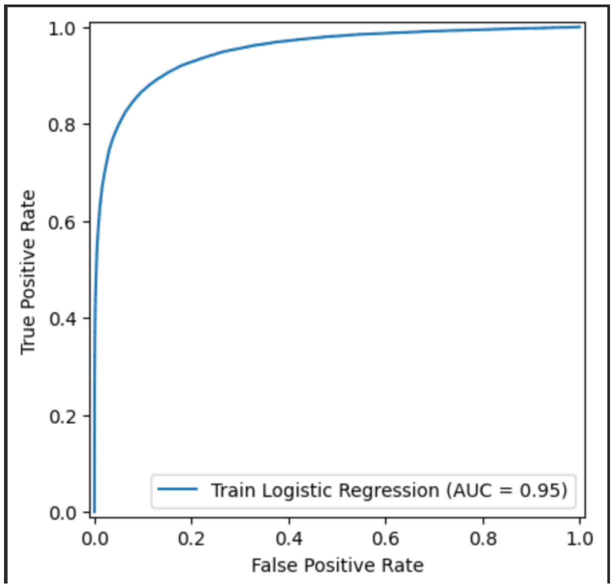
Data Cleaning and Preparation

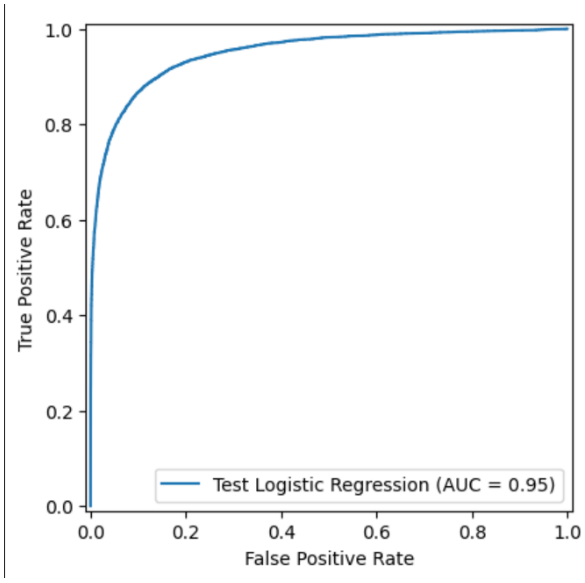
As stated before, various methods were used to clean the data. To identify which columns needed to change, I collected all columns that were of the object type. One of them had two unique values, Male and Female. Each were converted so 1 would represent males and 0 would represent females. Two functions were created to loop through the string values different columns to get either all the numbers or all the characters. This would leave columns with only string or numerical values, which is much easier to handle. A column of all null values was also found and dropped from the set, along with duplicates and other null values from the rest of the data. A correlation was also created to see if any potential columns needed to drop due to high correlation with other values. After viewing the correlation matrix, there was no significant correlation between variables that caused concern, and therefore all columns were kept. We still needed to convert the string values into numerical values, so dummy variables were created to categorize two columns. The original columns were dropped while two more added that included the dummy variables. A column also had infinite values, and so binning was used to separate the values into bins ranging from 0 to 100 with steps of 10. The original column was dropped, and the new bin column was added, and thus the set was cleaned and ready to be used in models.

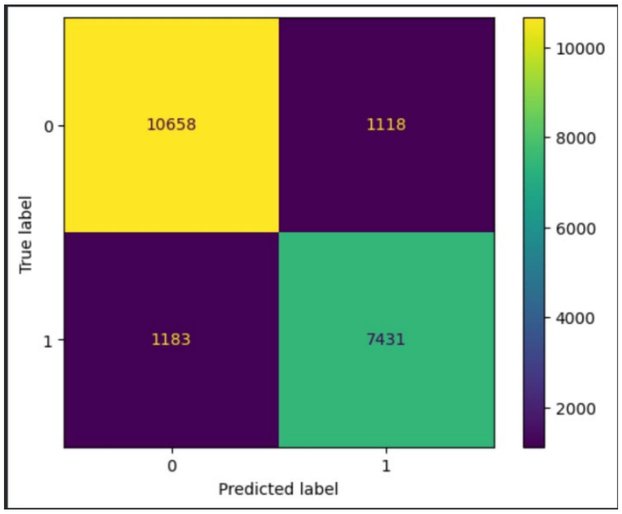
Part A – Logistic Regression Model

Project 3 Report
Christina Carbajal

Logistic regression identifies weights for each feature and produces a binary output, typically 1 or 0. It is very simplistic in its creation and efficient in datasets with linearly separable features. However, since it depends on the data being linear, it may not be applicable to non-linear data. It is also sensitive to features that are highly correlated with others. Feature selection and engineering was used to create the cleaned dataset, as described before. Hyperparameter tuning was also used to find the best logistic model. GridSeachCV was used to find each optimal parameter, which was then used in the final model. All the model evaluations on the test set, which included accuracy, precision, recall, and F1 score, showed around an 87% score. Both ROC curves showed that the AUC value is at .95. This is extremely close to 1, meaning that the logistic model was able to separate the two classes (0 and 1) highly correctly. The only challenge when creating this model was the amount of time it took for the GridSearchCV algorithm to run through all hyperparameters.



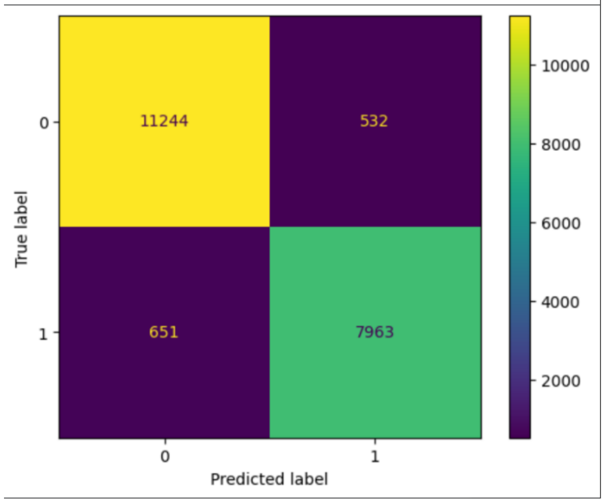


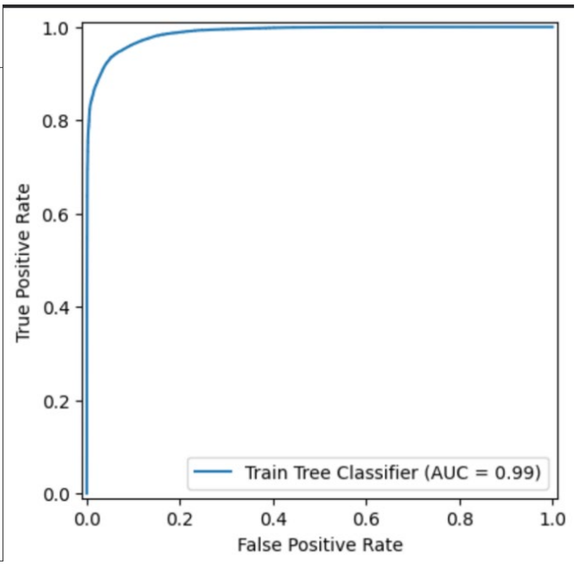


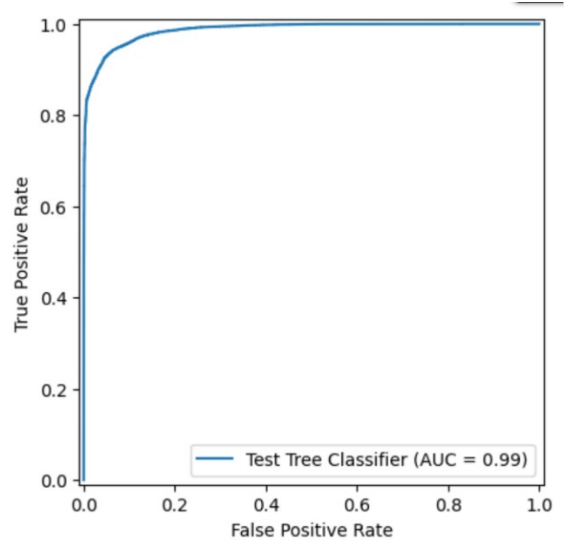
Project 3 Report
Christina Carbajal

Part B – Non-logistic Model

A gradient boosted classifier uses multiple decision trees to determine the class for each value. Some advantages of using the GBC model are that it can handle various data types, such as numerical or categorical and can handle missing values if needed. Some disadvantages include increased computational time and expense and could potentially overfit the data if the hyperparameters are tuned not carefully. Like the logistic model, GridSearchCV was used for cross-validation and hyperparameter tuning. The resulting model evaluation metrics, mentioned in the previous model, were around .92, which is better than the logistic model. The AUC values for both the train and test sets were at .99, which is much closer to 1 than the previous model. Again, the only challenge when building this model was time it took to find the hyperparameters.







Project 3 Report
Christina Carbajal

Discussion

The GBC model did considerably better than the logistic model by only about .10 in the model evaluation metrics and only .04 in the AUC values. This is most likely because of the GBC model being more complex and advanced than the logistic model. This may also mean that our data may not be linear, and thus the GBC model could handle the features better than the logistic model. However, the GBC model took a longer amount of time to run than the logistic model trying to fit the hyperparameters. If time and storage were in issue, then the logistic model may be a better fit in a situation. But overall, the GBC had higher accuracy, precision, recall, f1 score, and AUC value than the logistic model. To demonstrate my findings to a business parter, I would highly recommend the GBC model if time was not a constraint. I would explain that since the GBC model is a bit more complex than the logistic model, it can better categorize values correctly than the logistic model. Although the logistic model is more simplistic and is manageable with the data, it could be a backup model if time and storage was a problem.

	Logistic Regression	Gradient Boosting Classifier
AUC Training	.95	.99
AUC Test	.95	.99