Andre Archer
EGN5442

Project 3

Introduction and Overview

This project aims to clean, explore and apply basic modeling principles to a supplied blind dataset that is in a "dirty" state to begin with. The dataset originally consists of a binary target variable "y" and 28 columns of independent variables that need to be examined and cleaned up before analysis can be performed. Once the dataset is cleaned, some exploratory analysis will be performed on the dataset in terms of grouping data and looking for correlating variables. However, since this is a blind dataset, it is difficult to logically peruse through the data for variables that may be correlated or have trends of interest without any intuition as to where to begin. Binning of continuous variables and distribution analysis is helpful in guiding the exploration.

Cleaning will be done by taking the data and identifying which columns are already numerical data and which are strings. Then, the columns consisting of strings will be converted to numerical data to represent categorical variables when appropriate or by removing symbols such as '$' or '()' to isolate the numerical values. Finally, the dataset will be cleaned up by removing columns of constant values, empty columns, and by removing rows that have empty cells.

Once the data is cleaned and in a format that is usable for modeling, two different approaches will be taken to fit a model to the data. First, a logistic regression is fit to the data to predict the classifier y. To do this component, all the continuous data is scaled to be between 0 and 1 and some hyperparameter tuning is considered even though default values perform quite well. For comparison, a random forest classification scheme is applied to the dataset with some similar hyperparameter tuning considered. In both schemes, 80% of the data is reserved for training and 20% is used for the test set. The AUC values for both methods are compared as well as what independent variables each model identifies as the most important or having the most weight.

Data Cleaning and Preparation

The supplied dataset was analyzed and cleaned in the following order:

1) Columns Containing Strings
2) Columns of Constant Values
3) Empty Columns
4) Rows with Missing or NaN Values
5) Scaling and Setting Datatypes for Model Input

After cleaning the data, correlations between variables were found and visualized using a heatmap, where it was found that x5 and x6 are duplicates with a correlation value of 1 with each other, so it was dropped from the dataset (Figure 1)

Andre Archer
EGN5442

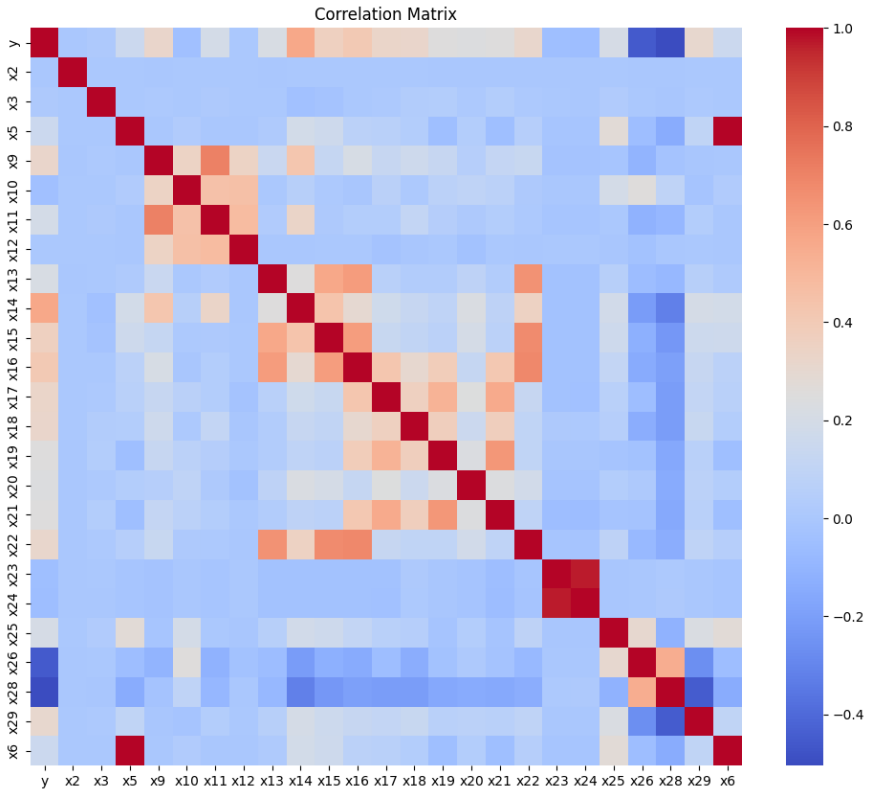

Figure 1: Correlation heatmap of cleaned dataset

Part A – Logistic Regression Model

Logistic regression is a binary classification method that fits the probability of being assigned to a class to the logistic sigmoid function. Using the input features, the model is able to output a probability between 0 and 1 of belonging to one of two classes (0 or 1). Based on the threshold assigned, this is how the model predicts one of the two classes. Logistic regression is simple to interpret and is fast to compute but is sensitive to collinearity and imbalanced data. The data here is balanced, so it should not over-assign one class from the other based on that. Also, logistic regression requires rescaling of all continuous data between 0 and 1.

Part B – Non- Logistic Model

For the non-logistic model component, a random forest method was taken. This classifier method uses a series of decision trees to compute a prediction based on random subsets of data and their features. This method is more general than logistic regression and is generally easy to find higher accuracy on blind datasets. However, this method takes longer to train and optimize, as was found in the case of hyperparameter tuning. Random forest classifiers also do not require rescaling of data between 0 and 1.

Discussions

In general, the random forest model outperformed the logistic regression. However, it took significantly longer to train and hyperparameter tuning took much longer. While I am not well versed in the parameter space for random forest classifiers, the hyperparameter tuning was not

very successful, showing a slightly reduced accuracy overall. In Figure 2, it can be seen that the AUC for the Random Forest is significantly higher than that of the Logistic Regression.

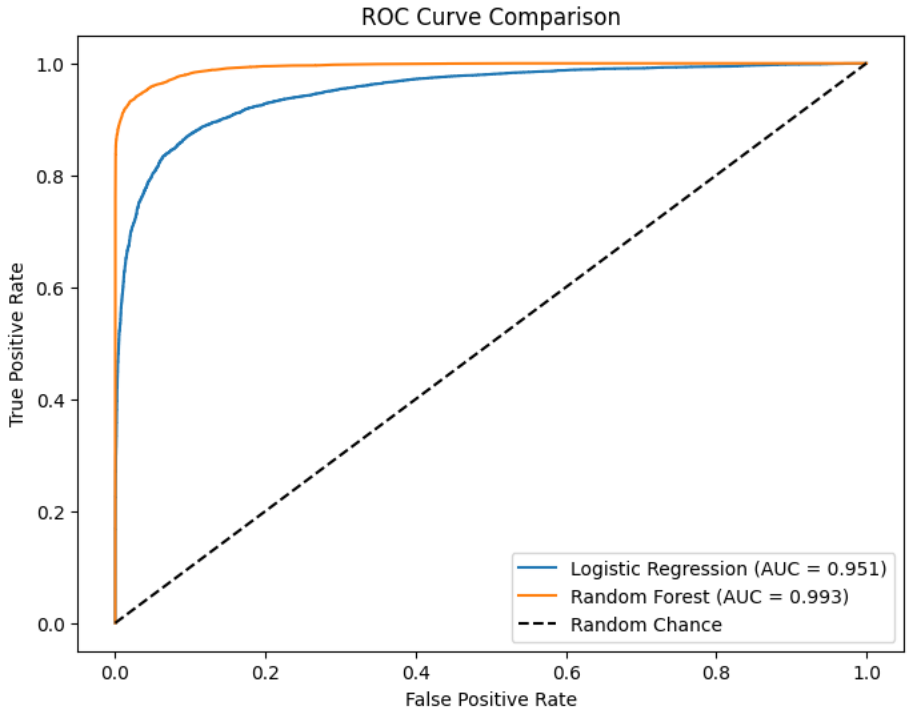

Figure 2: ROC Curves with Logistic Regression and Random Forest results overlayed

If the two models were to be discussed without using terms such as AUC, the confusion matrices in Figure 3 do a reasonable job of demonstrating the higher accuracy of the Random Forest classifier compared to Logistic Regression. Specifically, the Random Forest classifier is much less prone to false positives than the Logistic Regression.

Andre Archer
EGN5442

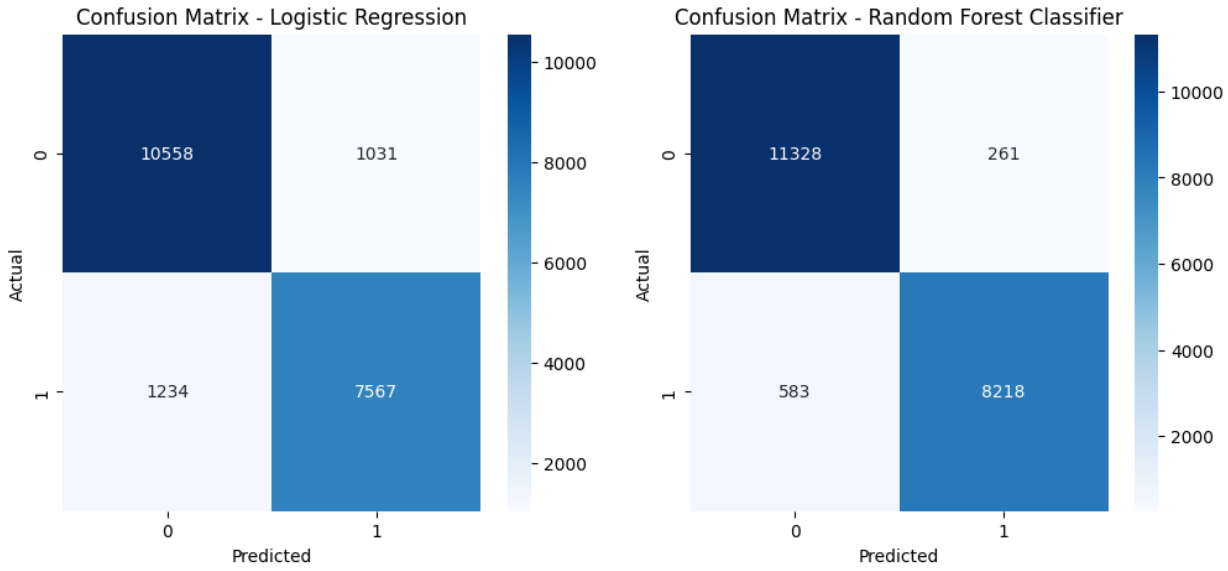

Figure 3: Confusion matrices for Logistic Regression and Random Forest classifier, showing that the Random Forest classifier outperforms logistic regression with significantly less false positives.

In terms of comparing what each model sees as important features from the input dataset, they generally agree well with one another on which features are assigned the most weight, as shown in Figure 4. The logistic regression is able to output coefficient for each parameter that is easy to interpret, identifying x14, x25, and x26 as the most important. It also essentially identifies x2, x3, and x5 as not important at all and x15 as having a coefficient of 0. For the random forest importance values, which are harder to interpret, the same top three features are identified as most important. However, there are none that are assigned a value of 0.

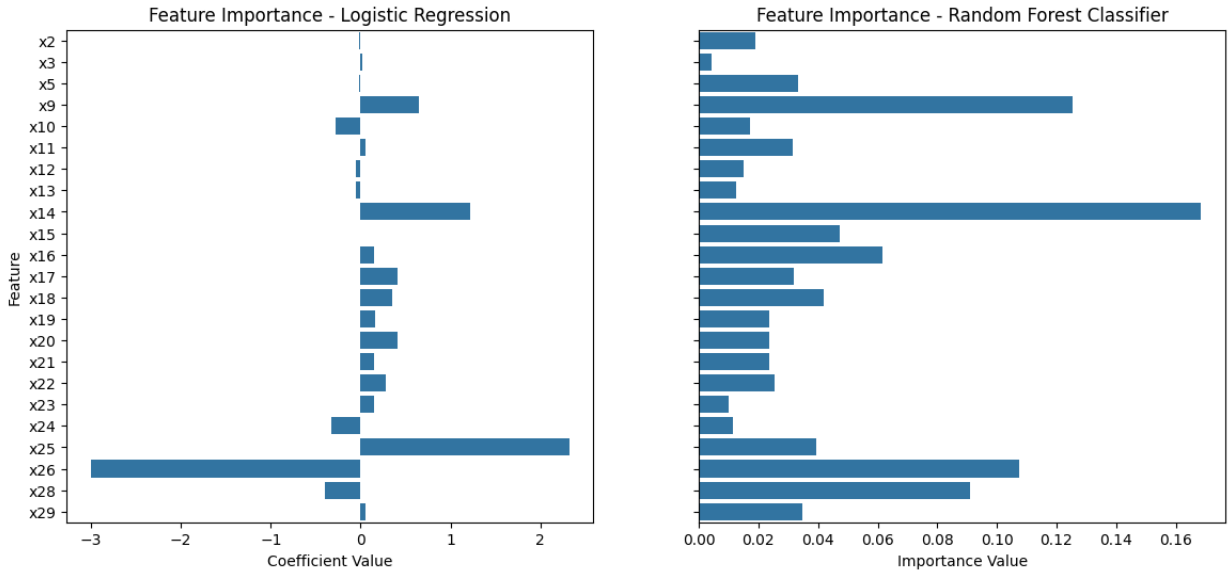

Figure 4: Comparison of feature importance for both models, where the two models generally agree on the most important features by comparison.

Table 1: Performance comparison between two models.

	Model1-logistic regression	**Model2-non-logistic**
AUC for training (80%)	0.950	1.000
AUC for testing (20%)	0.951	0.993

**validation set is recommended, it's okay if you did not reserve data for validation*