Assignment 6

Heman Kolla, kollah@rpi.edu

ITWS 4600-02: Data Analytics

Dr. Ahmed Eleish

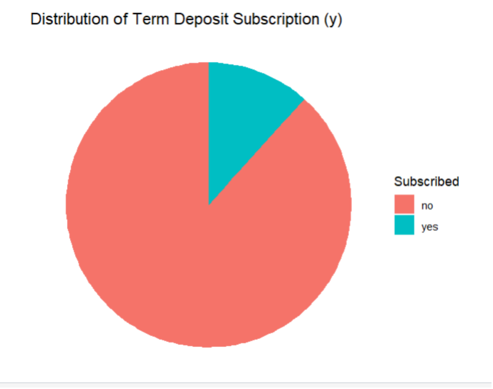
Exploratory Data Analysis

To start, I selected the “Bank Marketing" dataset of the 10 provided. I used the “bank-full.csv” for the data, which had 45211 observations over 17 mixed variables (numeric and categorical). After this preliminary analysis, I began my exploratory analysis,

The first step of EDA I did was data cleaning, where I removed rows with missing / NaN values (for numerical features).

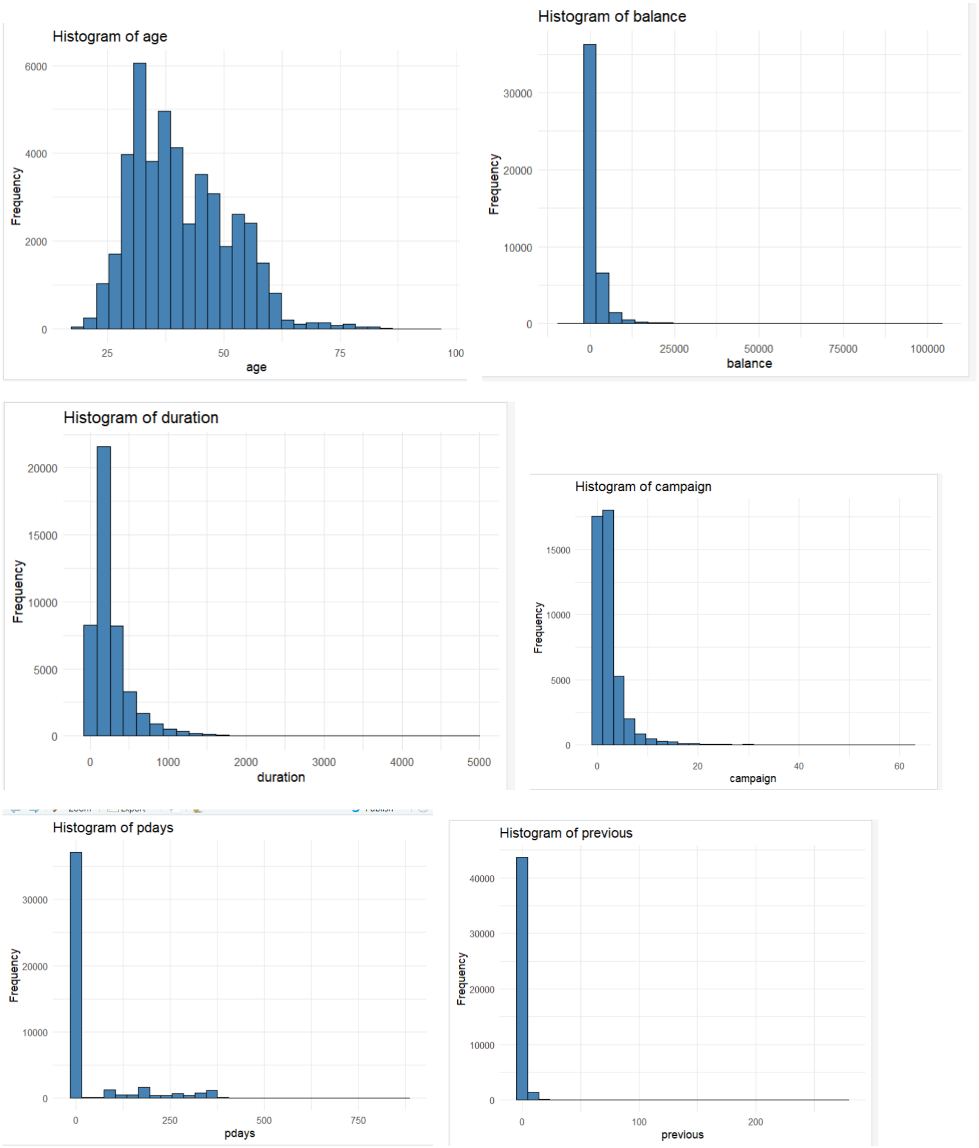
Secondly, I analyzed the target variable “y”, which referred to whether or not an individual had subscribed to the bank’s service. To do so, I plotted a pie chart, which you can see below. It demonstrated how only ~12.5% of people were subscribed to the term deposit.

Figure 1: Pie Chart of “Y”



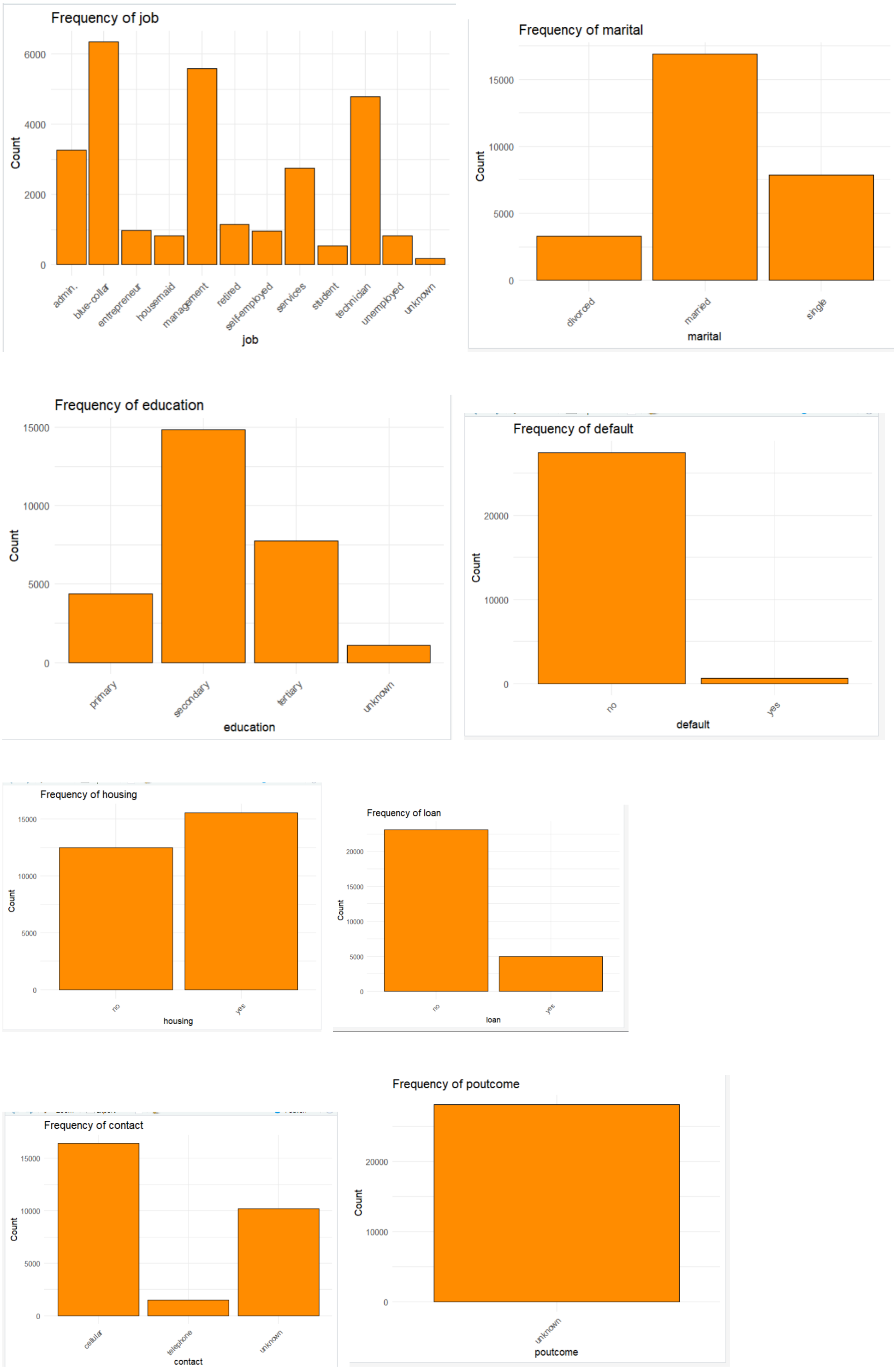
Next, I looked at the input variables, beginning with the numerical ones. I plotted frequency distributions for each through histograms. Looking at the x-axis scaling, there are no extreme ranges. As such, there is no visual evidence that suggests that I would need to perform logistic regression. However, as the scales have observations mainly on one side of an axis, this suggests outliers, which I remove as my next step according to the 1.5IQR rule.

Figure 2: Frequency Distribution Histograms of Numerical Input Variables



Now that I have taken a look at the numerical input variable, I want to now take a look at the categorical input variables. To do this, I plot frequency distributions for each as a bar chart. It is important to note that “unknown” is a completely valid observation, and trying to clean it would cut the number of observations in half (approximately). These charts display no evidence suggesting transformations are required.

Figure 3: Frequency Distribution Bar Chart ofCategorical Input Variables



With this, exploratory data analysis is concluded. The exploration has shown me that numerical variables need no transformation, and that there do exist many outliers, as seen with the reduced dataframe size post cleaning outliers. The categorical variables themselves have many “unknown” values in many fields, indicating a possible source of error in model development. However, it is still a valid observation to be reported.

The goal of the dataset was stated to be examining “y”, which is the subscriptions (YES/NO) to a bank’s term deposit service. As such, I want to try three classification-based models. Picking from the ones we covered in class that I know to be able to accomplish this, I will do: (1) kNN (2) Random Forest (3) Naive Bayes.

Model Development, Validation, & Optimization

To prepare for the rest of model development, I performed additional data preparation. This extends to cleaning and creating a proper training/testing split from the cleaned dataframe.

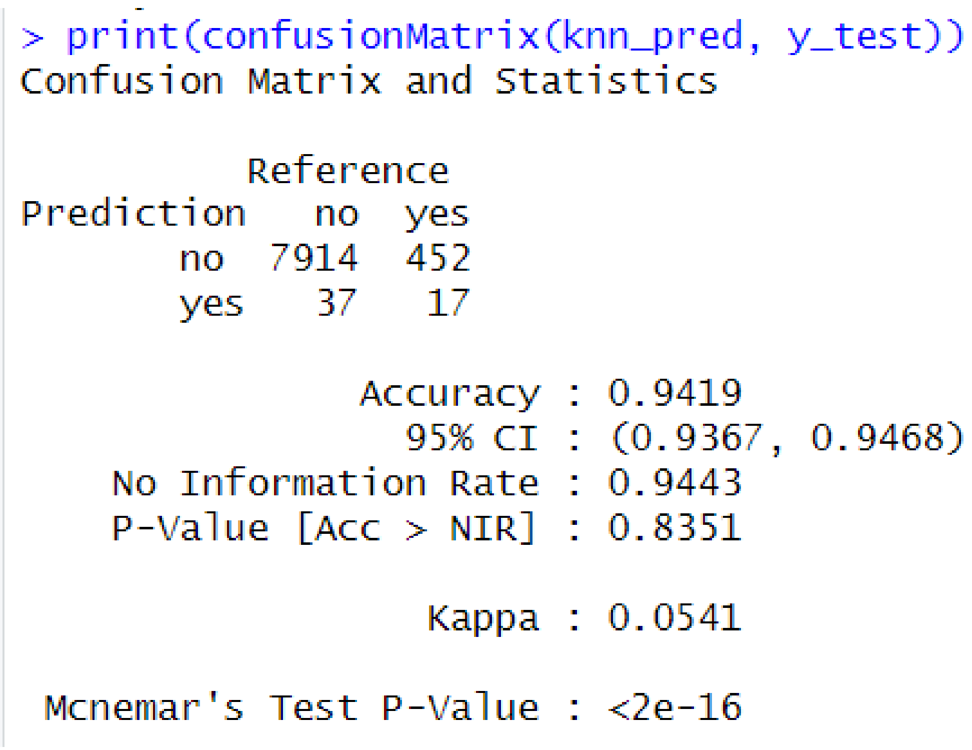
In terms of dimension reduction, PCA was considered. However, Random Forest and Naive Bayes both do not require scaling or orthogonality in the predictors. As such, PCA was not performed with the hope of preserving the original values better for training.

A. kNN

The kNN has one main difference from the subsequent models. As it is distance-based, it cannot work with categorical variables. As such, this model underwent minimal data prep to ensure that it is only trained on the numerical input variables that it is compatible with.

The workflow consisted of standardizing the data via the scale() function, then fitting the model to make a prediction. Then the outputs were displayed in a confusion matrix.

Figure 4: kNN Confusion Matrix & Statistics



```
> print(confusionMatrix(knn_pred, y_test))
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  7914  452
       yes   37   17
                                          
               Accuracy : 0.9419          
                 95% CI : (0.9367, 0.9468)
    No Information Rate : 0.9443          
    P-Value [Acc > NIR] : 0.8351          
                                          
                  Kappa : 0.0541          
                                          
 Mcnemar's Test P-Value : <2e-16          
```

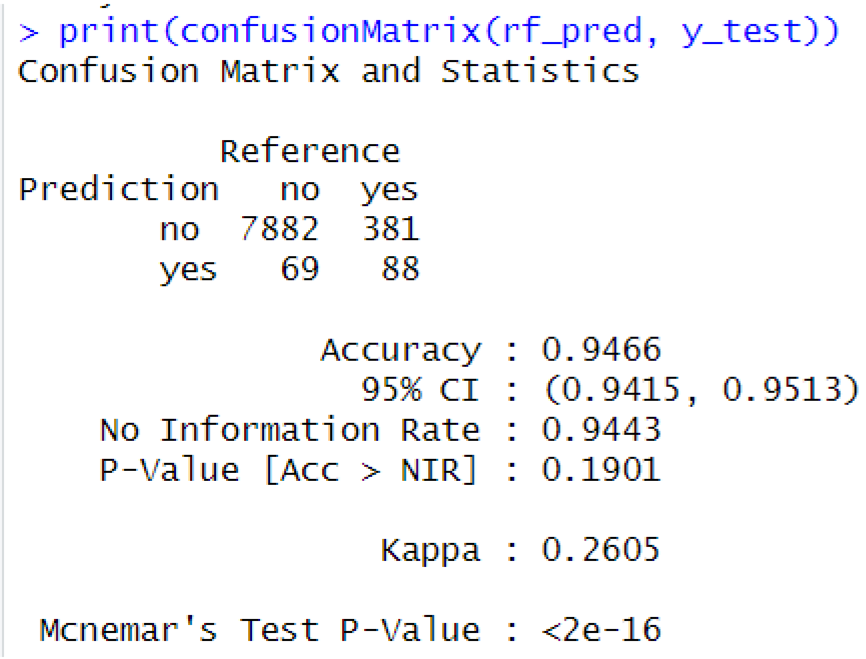
This model was validated on the test split, an out-of-sample measurement, and yielded an accuracy of 94.19%, which is very good performance with the kNN. Based on the confusion matrix, it appears that the model was significantly better at identifying those who did not subscribe to the bank service as opposed to those who did.

B. Random Forest

The Random Forest model did not have any numerical-based constraints. As such, it was trained on a mixed set of input variables—all of the numerical and categorical variables.

The workflow consisted of defining the relationship between the target feature “y” and the entire feature set. From there, the model was fitted, and used to predict. This was then displayed in a confusion matrix.

Figure 4: Random Forest Confusion Matrix & Statistics



```
> print(confusionMatrix(rf_pred, y_test))
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  7882  381
       yes   69   88

               Accuracy : 0.9466
                 95% CI : (0.9415, 0.9513)
    No Information Rate : 0.9443
    P-Value [Acc > NIR] : 0.1901

                  Kappa : 0.2605

 Mcnemar's Test P-Value : <2e-16
```

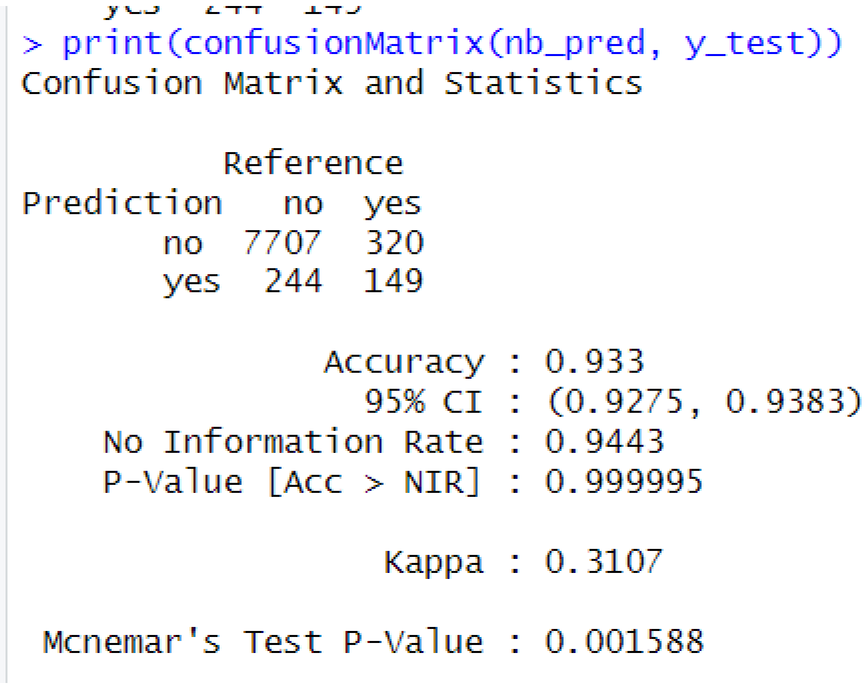
Similar to the kNN, this model was validated on the test split, an out-of-sample measurement, and yielded an accuracy of 94.66%. This is very good performance, which exceeds that of the kNN. Additionally, from the printed table of the confusion matrix, more people who selected to subscribe to the banks' service were correctly identified.

C. Naive Bayes

Similar to the Random Forest model, the Naive Bayes model does not have any numerical-based constraints. As such, it was also trained on a mixed set of input variables—all of the numerical and categorical variables.

Being trained with the same input variable set, the workflow was highly similar. The relationship between the target feature "y" and the entire feature set was defined. From there, the model was fitted, and used to predict. This was then displayed in a confusion matrix.

Figure $: Naive Bayes Confusion Matrix & Statistics



```
    yes   244   149
> print(confusionMatrix(nb_pred, y_test))
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  7707  320
       yes  244  149
                                          
               Accuracy : 0.933           
                 95% CI : (0.9275, 0.9383)
    No Information Rate : 0.9443          
    P-Value [Acc > NIR] : 0.999995        
                                          
                  Kappa : 0.3107          
                                          
 Mcnemar's Test P-Value : 0.001588        
```

Similar to both prior models, the Naive Bayes was validated on the test split. It had an accuracy of 93.3%, which is good model performance. This is, however, lower than the accuracy of both prior models. It is interesting to note though, that the confusion matrix reports that it is better at identifying those who did subscribe to the banks' service, but got worse at identifying those who did not.

Decisions

All 3 models developed in this assignment—the kNN, Random Forest, and Naive Bayes— displayed strong performance with accuracies >90%. This implies that the input features given by the dataset have strong predictive power regarding whether or not a client will subscribe to a term deposit service. Random Forest achieved the highest overall performance at 94.66%, while the Naive Bayes performed the worst. But, the Naive Bayes model predicted with the most accuracy who will subscribe to the service, even though its overall accuracy was lower.

As such, for practical use to inform banking decisions, I would conclude that the Random Forest be used as the model of choice. However, I would also suggest that further optimizations to the Naive Bayes should be done internally as it is better at identifying those who will subscribe to the service, which is important for a bank trying to maximize revenue via said subscriptions. Still, since each model has a >5% probability of being incorrect overall, I would hesitate to use these models as the sole reasoning behind a banking decision. They should be used as complementary until refined.