Final Data Analysis Report:
Predictive Modeling for Bank Deposit Subscriptions

(OCRUG Data Science Hackathon)

Won Second place for
Best Model & Best Visualization

Project done by:
Yuan Liu
Chuyan Zhang
Takako Suzuki
Dora Yuan
York Fang

Report written by:
Takako Suzuki

November 10, 2019

ABSTRACT
This report summarizes the predictive modeling and analysis results associated with the OCRUG 2019 Hackathon competition. The purpose of this report is to document the decision making phases of data preprocessing procedures and corresponding data modeling.

BACKGROUND
The original dataset is collected from a Portuguese marketing campaign related to bank deposit subscription. Our objective is to build a predictive model that explains the success of contacts and further increase efficiency by identifying the important attributes for success.

Our methodology for this project can be simply defined with four stages. See table 1.1

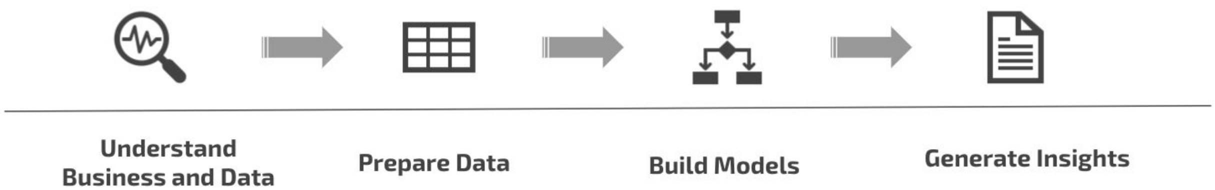
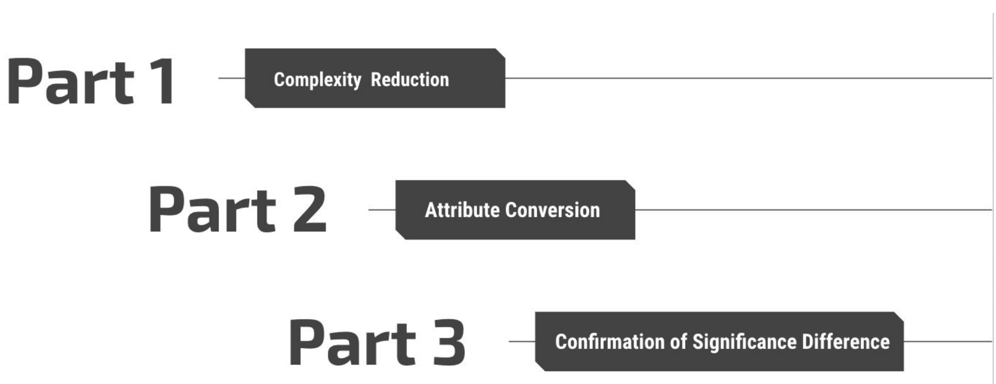


Table 1.1

A. UNDERSTAND BUSINESS / DATA

In this dataset, the marketing campaign was based on phone calls and success was measured by whether or not the client makes a deposit after the campaign. The attributes in this dataset are categorized into three groups: bank client data, related to the last contact of the current campaign, and other attributes. Generally speaking, every marketing campaign has one goal: conversion from leads to sales. Therefore, our team decided to start with the common inquiry of every marketing campaign - what contributes to a successful conversion through the phone campaign.

B. DATA PREPROCESSING



There is no missing data in this dataset so we started by converting all categorical attributes to numeric data and exploring the correlation of all independent attributes with dependent variable “y” (Whether the client makes a deposit or not) using linear regression. The result shows that all attributes have very low correlation values. Our initial assumption of this outcome is that all the other variables have up to 4 levels but in the “job” attribute, it has 12 levels and that factor might lead to desaturation of the correlation value.

Part 1. Complexity Reduction

Our next step is to perform a clustering method on the “job” attribute by assigning occupations into groups based on the similarity in personal information (age, marital, education, default, balance, housing, and loan). By doing so, we reduced 12 levels to 5 levels. Using K-means Nearest Neighbor we found the initial optimal number of clusters is 4 with SSE of 30,000 (see Table 1.2)

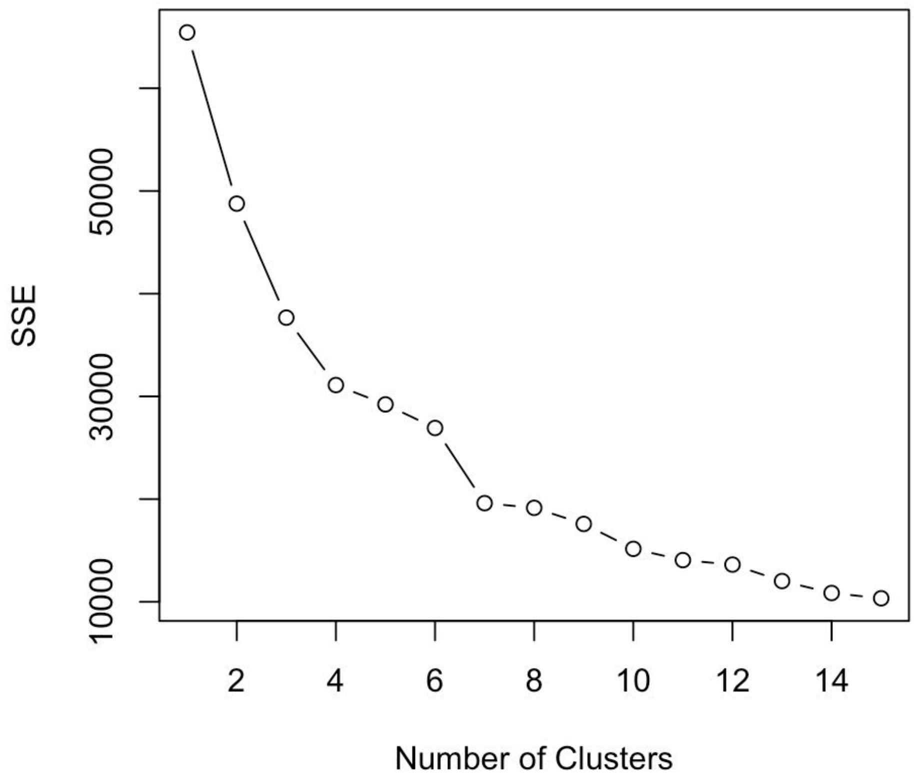


Table 1.2

We then assigned jobs into cluster 1, 2, 3, and 4 by selecting the highest count of each job in the clusters; however, due to the imbalance ratio of “yes” and “no” in y attribute, we were unable to assign any job with the highest count in cluster 3. Therefore, we decided to use a total of 5 clusters so that each cluster has at least one highest count of the job (See Table 1.3).

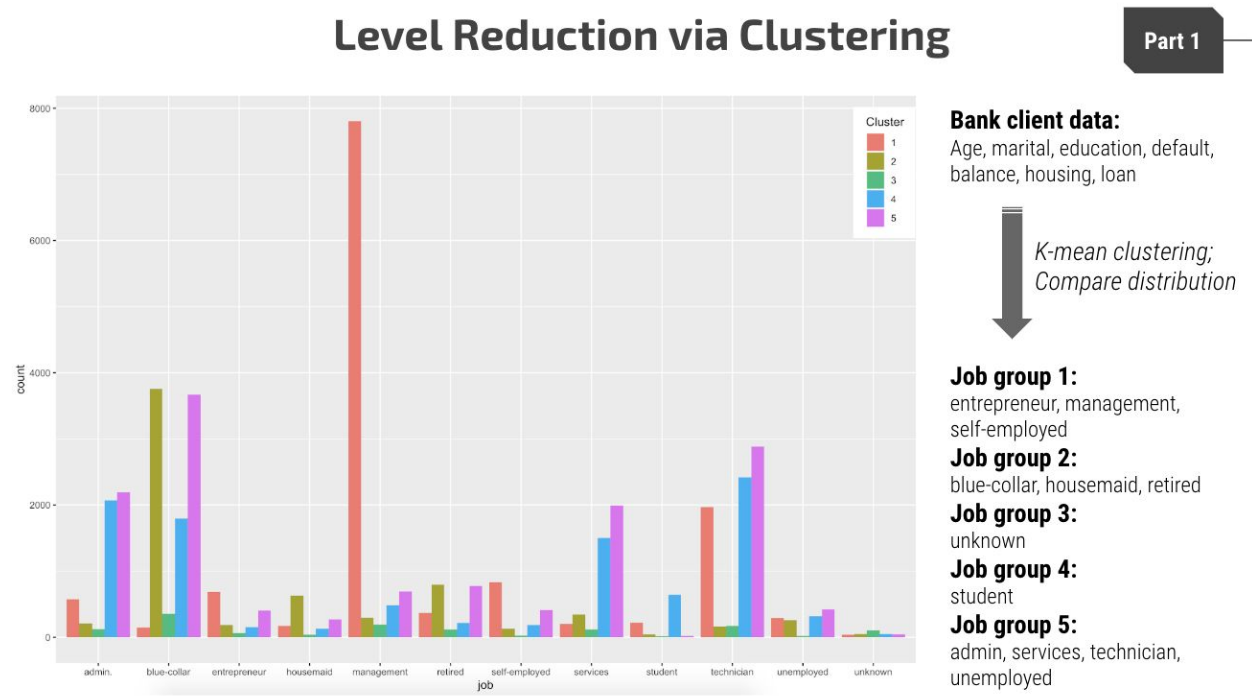


Table 1.3

Part 2. Attribute Conversion

With in-depth research, we finalized our two tentative models that are commonly used among marketing analysis: Naive Bayes and Decision Tree. Due to the limitation of Naive Bayes' model which works better with categorical variables, we converted numerical variables to categorical by binning using the Smbinning package. Optimal Binning analyzes the relationship with a binary target variable and finds the optimal cutpoints (See Table 1.4)

```
> result <- smbinning(df=df, y="y", x="balance")
> result$ivtable
  Cutpoint CntRec CntGood CntBad CntCumRec CntCumGood CntCumBad PctRec GoodRate BadRate
1   <= -47   3193     166   3027      3193        166      3027 0.0706   0.0520  0.9480
2    <= 60   7628     594   7034     10821        760     10061 0.1687   0.0779  0.9221
3   <= 798  17577    1963  15614     28398       2723     25675 0.3888   0.1117  0.8883
4    > 798  16813    2566  14247     45211       5289     39922 0.3719   0.1526  0.8474
5  Missing      0       0      0     45211       5289     39922 0.0000      NaN     NaN
6    Total  45211    5289  39922        NA         NA        NA 1.0000   0.1170  0.8830
    Odds  LnOdds     WoE     IV
1 0.0548 -2.9033 -0.8820 0.0392
2 0.0844 -2.4716 -0.4503 0.0288
3 0.1257 -2.0737 -0.0524 0.0010
4 0.1801 -1.7142  0.3071 0.0394
5    NaN     NaN     NaN    NaN
6 0.1325 -2.0213  0.0000 0.1084
```

Table 1.4

We were aware of the disadvantage of this binning method because combining a large number of levels in a variable will lose accuracy due to the difference in weight percentage. However, our purpose for using this method is to avoid overfitting of our data when performing the predictive model and to decrease processing time.

Part 3. Confirmation of Significance

Our last step of data preprocessing is to confirm once again that the attributes we dropped are proven to be insignificant. One way of approaching this is to compare the correlation between the levels of each variable. The "Default" attribute in Table 1.5 can be discarded because the rate of success for the two levels is very close. We noted that this graphical analysis may have flaws because we are disregarding the interaction of "Default" with other attributes. Hence, we run linear regressions to examine each attribute and compare the significance value.

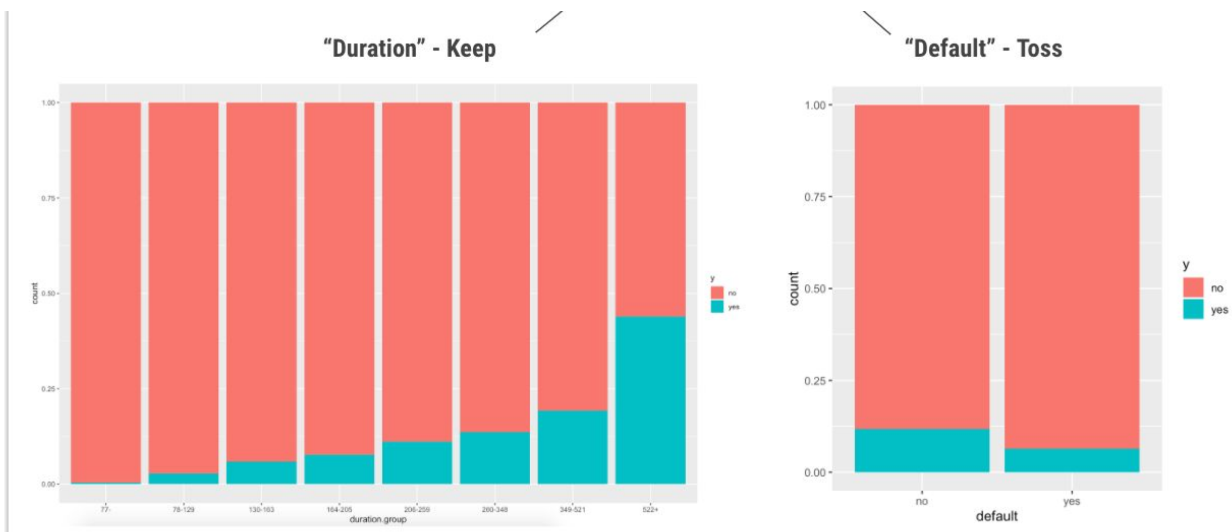
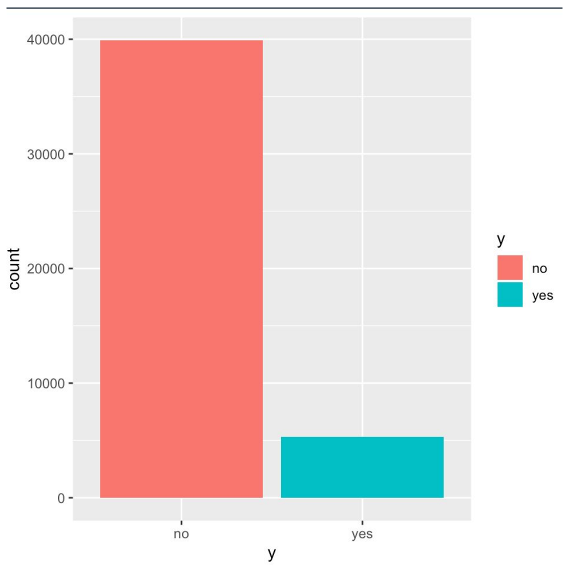


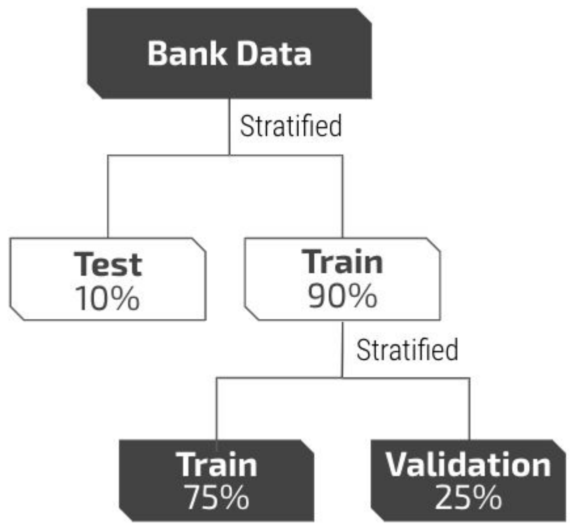
Table 1.5

Finally, we found out that "Pday", "Previous", "Default", and "Contacted" have the least correlation with our dependent variable and we dropped the four low impact variable.

C. DATA MODELING

The dependent variable (as shown in Table 2.1) is significantly skewed, so we used a stratified random sampling method to ensure the ratio of "yes" and "no" is identical.





Naive Bayes & Decision Tree Model

We explored the level of importance of each attribute using Naive Bayes and Decision Tree and found out that the most important attribute for both models is "duration". (See Table 2.2 & Table 2.3)

```
### Level of importance
x.nb <- varImp(nb)
impTab <- x.nb$importance
ggplot(impTab, aes(x= reorder(row.names(impTab), +yes), y=yes)) +
  geom_bar(stat = 'identity', aes(fill = row.names(impTab))) +
  labs(title = "Variable in predicting term deposit", x = "Variables", y = "Importance") +
  scale_fill_brewer(palette = "Set3") + coord_flip() +
  theme_classic() +
  theme(legend.position = "none")
```

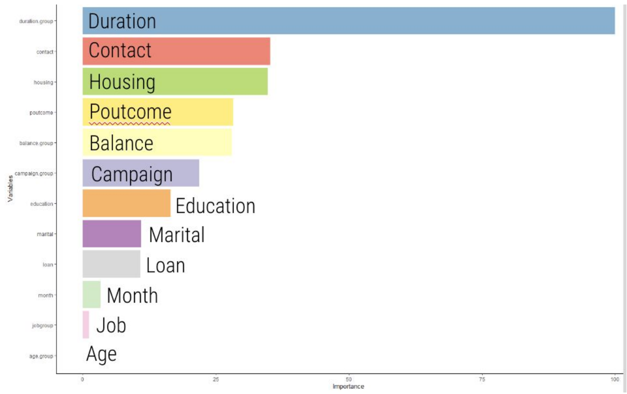


Table 2.2 Naive Bayes

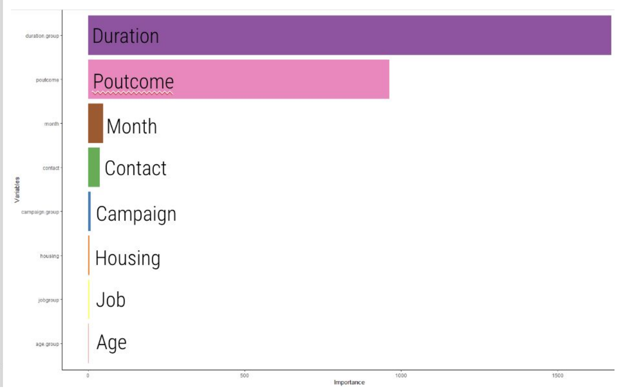


Table 2.3 Decision Tree

In the modeling phase, we successfully tested the two models and gathered the accuracy rate in each model. (see Table 2.4 & 2.5)

```
> confusionMatrix(nb.pred, validate.data$y)
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  8579  698
       yes  403  492

               Accuracy : 0.8918
                 95% CI : (0.8856, 0.8977)
    No Information Rate : 0.883
    P-Value [Acc > NIR] : 0.002921

                  Kappa : 0.413

 Mcnemar's Test P-Value : < 0.00000000000000022

            Sensitivity : 0.9551
            Specificity : 0.4134
         Pos Pred Value : 0.9248
         Neg Pred Value : 0.5497
             Prevalence : 0.8830
         Detection Rate : 0.8434
   Detection Prevalence : 0.9120
      Balanced Accuracy : 0.6843

       'Positive' Class : no
```

Table 2.4 - Naive Bayes

```
> confusionMatrix(dtree.pred, validate.data$y)
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  8784  825
       yes  198  365

               Accuracy : 0.8994
                 95% CI : (0.8934, 0.9052)
    No Information Rate : 0.883
    P-Value [Acc > NIR] : 0.00000007894

                  Kappa : 0.369

 Mcnemar's Test P-Value : < 0.00000000000000022

            Sensitivity : 0.9780
            Specificity : 0.3067
         Pos Pred Value : 0.9141
         Neg Pred Value : 0.6483
             Prevalence : 0.8830
         Detection Rate : 0.8635
   Detection Prevalence : 0.9447
      Balanced Accuracy : 0.6423

       'Positive' Class : no
```

Table 2.5 - Decision Tree

Using the validation dataset, we compared the two models' accuracy rate and concluded that Decision Tree is higher than Naive Bayes by 0.008. Although the two models show a comparably high accuracy, we chose Decision Tree as our final model because it takes care of various issues such as outliers and missing values and these issues are present in our dataset; therefore, we chose Decision Tree.

Decision Tree-Test

The following table on the right shows an accuracy of 0.8936 for our Decision Tree model with the testing dataset.

```
> confusionMatrix(dtree.test, test.data$y)
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  3902  391
       yes   90  137

               Accuracy : 0.8936
                 95% CI : (0.8842, 0.9024)
    No Information Rate : 0.8832
    P-Value [Acc > NIR] : 0.01478

                  Kappa : 0.3148

 Mcnemar's Test P-Value : < 0.0000000000000002

            Sensitivity : 0.9775
            Specificity : 0.2595
         Pos Pred Value : 0.9089
         Neg Pred Value : 0.6035
             Prevalence : 0.8832
         Detection Rate : 0.8633
   Detection Prevalence : 0.9498
      Balanced Accuracy : 0.6185

       'Positive' Class : no
```

Accuracy: 0.8936

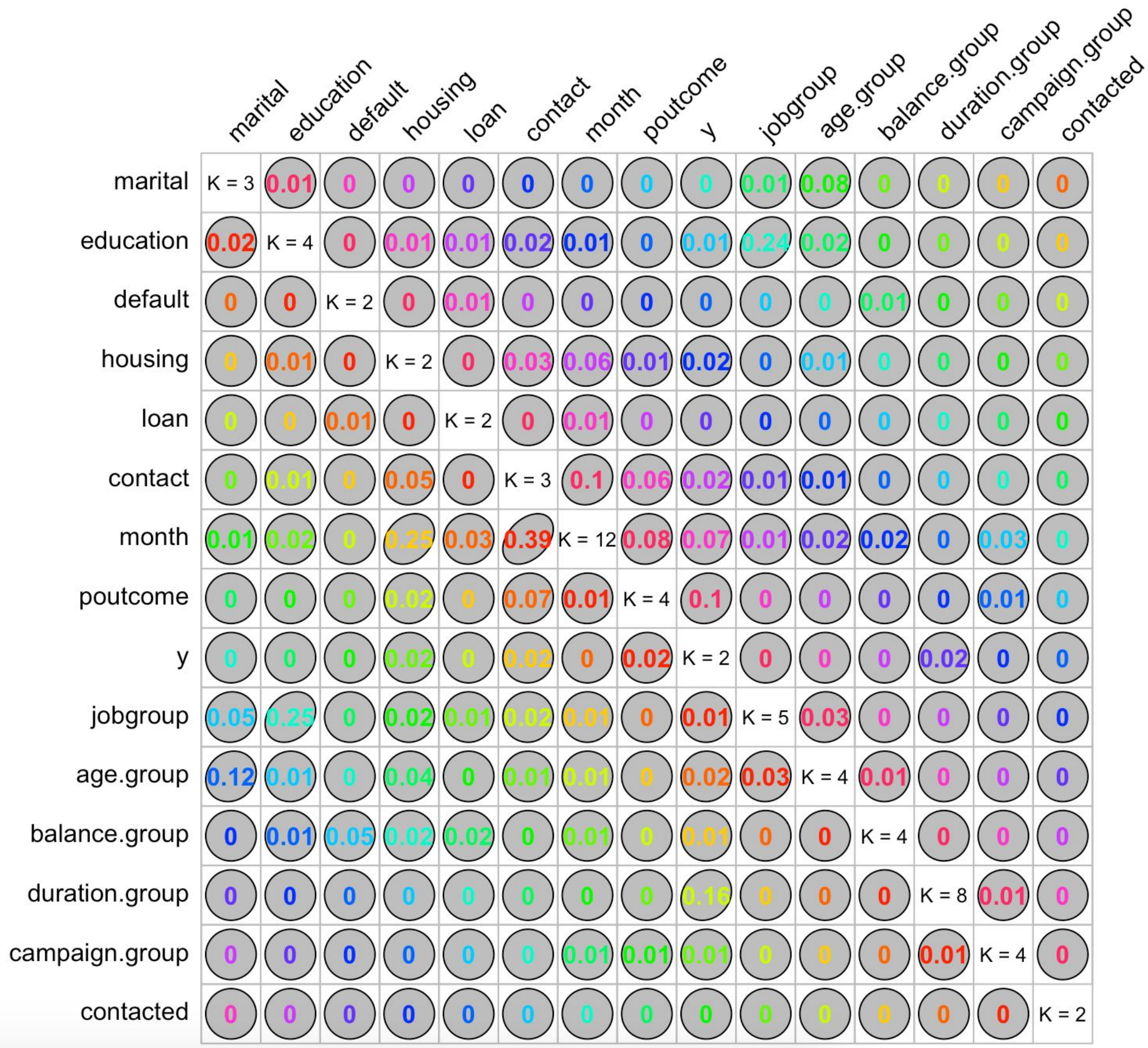
C. CONCLUSION

In this project, our team spent the most amount of time to decide how to convert categorical variables and what is the most effective approach to data preparation for modeling.

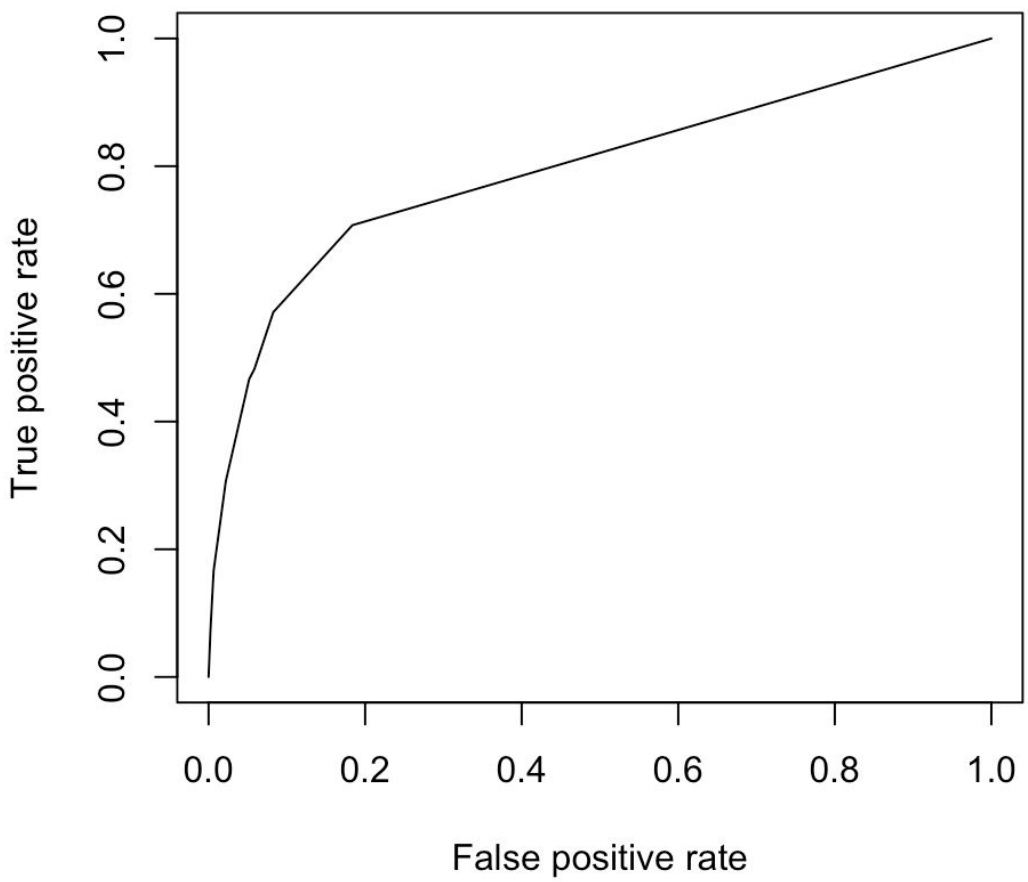
Duration is the most relevant feature, meaning that the longer the call representative spends with a customer, the higher the conversion rate. The second feature is Poutcome which indicates that customers who have deposited before having a higher chance to deposit again. This is common in marketing when a customer has already established customer loyalty with the company. A loyal customer remains loyal when offering lower prices and better discounts; therefore, a phone campaign with alluring deals will achieve a higher conversion rate. A new direction we wish to approach in the future is to segment customers into two groups: old customers and new customers. This is because the two groups present different purchasing behaviors and lead to different outcomes.

Appendix

Appendix A: Correlation Matrix after dropping low impact variables



Appendix B: ROC graph of Decision Tree Model (Test dataset)


True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate