Jaclyn Coate, Josh Eysenbach, Branum Stephan

Abstract

A basic logistic regression analysis to predict a bank customer’s subscription and additional competing complex models for best prediction results.

Subscriptions

Table of Contents

[1. Introduction 12](#_Toc38110119)

[2. Data description 12](#_Toc38110122)

[3. Exploratory Data Analysis (EDA) 13](#_Toc38110147)

[4. Objective 1: Simple Logistic Regression Model 14](#_Toc38110157)

[A. Problem Statement and Approach Review 14](#_Toc38110158)

[B. Model Selection 15](#_Toc38110160)

[C. Parameter Interpretation 15](#_Toc38110165)

[i. Interpretation 15](#_Toc38110166)

[ii. Confidence Intervals 16](#_Toc38110168)

[D. Conclusion 16](#_Toc38110170)

[5. Objective 2: Competing Models 16](#_Toc38110172)

[A. Model Selection 16](#_Toc38110173)

[i. Complex Logistic Regression 16](#_Toc38110175)

[ii. Linear Discriminant Analysis (LDA) / Quadratic Discriminant Analysis (QDA) 16](#_Toc38110177)

[i. K-Nearest Neighbor (KNN) 17](#_Toc38110183)

[ii. Random Forest 18](#_Toc38110188)

[iii. Comparing Engineered Models 19](#_Toc38110193)

[C. Main Analysis Content 19](#_Toc38110195)

[D. Conclusion / Discussion 20](#_Toc38110212)

[6. Appendix 21](#_Toc38110214)

[A. Figure 1 21](#_Toc38110215)

[B. Figure 2 21](#_Toc38110216)

[C. Figure 3 21](#_Toc38110217)

[D. Figure 4 22](#_Toc38110218)

[E. Figure 5 22](#_Toc38110219)

[F. Figure 6 22](#_Toc38110220)

[G. Figure 7 23](#_Toc38110221)

[H. Figure 8 23](#_Toc38110222)

[I. Figure 9 23](#_Toc38110223)

[J. Figure 10 24](#_Toc38110224)

[K. Figure 11 24](#_Toc38110225)

[L. Figure 12 24](#_Toc38110226)

[M. Figure 13 25](#_Toc38110227)

[N. Figure 14 25](#_Toc38110228)

[O. Figure 15 25](#_Toc38110229)

[P. Figure 16 27](#_Toc38110230)

[Q. Figure 17 27](#_Toc38110231)

[R. Figure 18 27](#_Toc38110232)

[S. Figure 19 28](#_Toc38110233)

[T. Figure 20 28](#_Toc38110234)

[U. Figure 21 28](#_Toc38110235)

[V. Figure 22 30](#_Toc38110236)

[W. Figure 23 30](#_Toc38110238)

[X. Figure 24 30](#_Toc38110239)

[Y. Figure 25 30](#_Toc38110240)

[Z. Figure 26 31](#_Toc38110241)

[AA. Figure 27 31](#_Toc38110242)

[BB. Figure 28 32](#_Toc38110243)

[CC. Figure 29 32](#_Toc38110244)

[DD. Figure 30 32](#_Toc38110245)

[EE. Figure 31 33](#_Toc38110246)

[FF. Figure 32 33](#_Toc38110247)

[GG. Figure 33 34](#_Toc38110248)

[HH. Figure 34 34](#_Toc38110250)

[II. Figure 35 35](#_Toc38110251)

[JJ. Figure 36 35](#_Toc38110252)

[KK. Figure 37 36](#_Toc38110253)

[LL. Figure 38 36](#_Toc38110254)

[MM. Figure 39 36](#_Toc38110255)

[NN. Figure 40 37](#_Toc38110256)

[OO. Figure 41 37](#_Toc38110258)

[PP. Figure 42 37](#_Toc38110259)

[QQ. Figure 43 38](#_Toc38110260)

[RR. Figure 44 38](#_Toc38110262)

[SS. Figure 45 38](#_Toc38110265)

[TT. Figure 46 38](#_Toc38110267)

[UU. Figure 47 39](#_Toc38110270)

[VV. Figure 48 39](#_Toc38110271)

[WW. Figure 49 39](#_Toc38110272)

[XX. Figure 50 39](#_Toc38110273)

[YY. Figure 51 39](#_Toc38110274)

[ZZ. Figure 52 40](#_Toc38110275)

[AAA. Figure 53 40](#_Toc38110276)

[BBB. Figure 54 40](#_Toc38110277)

[CCC. Figure 55 40](#_Toc38110278)

[DDD. Figure 56 41](#_Toc38110279)

[EEE. Figure 57 41](#_Toc38110280)

[FFF. Figure 58 41](#_Toc38110281)

[GGG. Figure 59 41](#_Toc38110282)

[HHH. Figure 60 42](#_Toc38110283)

[III. Figure 61 42](#_Toc38110284)

[JJJ. Figure 62 42](#_Toc38110285)

[KKK. Figure 63 42](#_Toc38110286)

[LLL. Figure 64 43](#_Toc38110287)

[MMM. Figure 65 43](#_Toc38110288)

[NNN. Figure 66 43](#_Toc38110289)

[OOO. Figure 67 44](#_Toc38110290)

[PPP. Figure 68 44](#_Toc38110291)

[QQQ. Figure 69 44](#_Toc38110292)

[RRR. Figure 70 45](#_Toc38110293)

[SSS. Figure 71 45](#_Toc38110294)

[TTT. Figure 72 45](#_Toc38110295)

[UUU. Figure 73 46](#_Toc38110296)

[VVV. Figure 74 46](#_Toc38110297)

[WWW. Figure 75 46](#_Toc38110298)

[XXX. Figure 76 47](#_Toc38110299)

[YYY. Figure 77 47](#_Toc38110300)

[ZZZ. Figure 78 47](#_Toc38110301)

Subscriptions

# Introduction

## Banking institutions are one of the largest in the world. They complete millions of transactions daily in person, digitally, and over the phone. Some of these transactions are efforts at engaging their customers in campaigns to sell additional banking products and offerings. Currently banking institutions are executing their best efforts to engage some of the youngest generations that are aging into the top salary brackets. The majority of their funding is going into digitizing their banking features. Additionally, [Everfi](https://everfi.com/insights/blog/retail-bank-marketing-trends/) has found that putting more resources into data is one of the continuing trends for banks today. While putting funds towards data in marketing for banks is important, so is the ability to properly analyze the data and predict desired outcomes based on a bank’s customer’s unique traits. The following paper works to achieve just this.

## Our chosen data set is from a Portuguese banking institution and is referencing  a campaign that is trying to achieve a subscription (yes or no outcome) of a banking deposit by their customers. In this report we analyze the given data set variables via an exploratory data analysis, simple logistic regression model, and complex logistic regression models to determine the best variables to predict the subscription outcome.

# Data description

## The data we utilized for this analysis was provided by the UCI Machine Learning group which can be found [here.](https://archive.ics.uci.edu/ml/datasets/Bank%20Marketing) The data is related to the direct marketing campaigns of a Portuguese banking institution. These marketing campaigns were based on phone calls. On many occasions there was more than one contact to the client, and it was required in order to determine if the product subscription (y - bank term deposit) would be a ‘yes’ or a ‘no’.

|  |  |
| --- | --- |
| **Variable Name** | **Variable Description** |
| *Age* | Age of customer |
| *Job* | type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown') |
| *Martial* | marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed) |
| *Education* | (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown') |
| *Default* | has credit in default? (categorical: 'no','yes','unknown') |
| *Housing* | has housing loan? (categorical: 'no','yes','unknown') |
| *Loan* | has personal loan? (categorical: 'no','yes','unknown') # related with the last contact of the current campaign: |
| *Contact* | contact communication type (categorical: 'cellular','telephone') |
| *Month* | last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec') |
| *Day\_of\_Week* | last contact day of the week (categorical: 'mon','tue','wed','thu','fri') |
| *Duration* | last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model. |
| *Campaign* | number of contacts performed before this campaign and for this client (numeric) |
| *Pdays* | number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted) |
| *Previous* | number of contacts performed before this campaign and for this client (numeric) |
| *Poutcome* | outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success') |
| *Emp.var.rate* | employment variation rate - quarterly indicator (numeric) |
| *Cons.price.idx* | consumer price index - monthly indicator (numeric) |
| *Cons.conf.idx* | consumer confidence index - monthly indicator (numeric) |
| *euribor3m* | euribor 3 month rate - daily indicator (numeric) |
| *Nr.employed* | number of employees - quarterly indicator (numeric) |
| *y* | Dependent Variable: Subscription completion (yes or no) |

# Exploratory Data Analysis (EDA)

## Upon first look we reviewed the raw data set we were given. We can see there is a full complement of both continuous and categorical variables independent variables with a single dependent variable - y [[Figure 1](#_Figure_1)]. Variable y is a dichotomous outcome of yes or no. It is referring to whether or not a bank customer signed up for a deposit subscription. In order to keep variable logically labeled, we changed the name of variable *y* to variable *Subscription*.

## The next step in our EDA consisted of dropping logically irrelevant variables. At first glance we did not feel comfortable eliminating any variables without further insight into if they may or may not affect the dependent variable - y. Further investigation led us to realize that the *duration* variable is actually a measurement taken post call completion. As this would not be a variable unique to a specific customer premarketing campaign phone call it would not be appropriate to use in predictive modeling. Therefore, we removed it from our list of viable variables [[Figure 2](#_Figure_2)].

## Our next round of investigation took us into looking for any missing values or NAs [[Figure 3](#_Figure_3)]. We used an R package called mice and were able to quickly determine that there was no need to complete any NA value extrapolation to move forward.

## The continuous variable evaluation was the next piece of the EDA. We accomplished this by creating individual boxplots of the continuous variables versus *Subscription*. When using this methodology, you must examine the graph for large differences in the box-plot ranges. These boxplot graphs can be found in [Figure 4](#_Figure_4) through [Figure 14](#_Figure_14). An additional way to evaluate the important continuous variables is through density plots. When peaks are separated by different colors it is an indicator that the variable could be a good one to determine the dependent variable - *Subscription*. Our function can be found in [Figure 15](#_Figure_15) along with every density plot supplied by that function. Using both the boxplot and density plot methods there does seem to be some continuous variables that would be good for predicting, these include: *pdays, campaign, previous, cons.price.indx, cons.conf.idx, emp.var.rate, nr.employed*. Leaving us only with *age* to remove for now.

### NOTE: We noticed that the *pdays* variable appears like customers fall into one of two groups: *a number* versus *999*. We think ideally this would be a categorical variable for better predicting. We address this in [Objective 2: Complex Logistic Model](#_Complex_Logistic_Regression).

## We have completed selecting those continuous variables that are significant for our evaluation. However, we need to now check on if these continuous variables contain any multicollinearity. In [Figure 16](#_Figure_16) you can see we performed a large pairs plots with the separation of yes and no responses from our dependent variable *Subscription*. We can see some strong linearity showing between *nr.employed, euribor3m, emp.var.rate,* and *previous.* However, we are not observing any large separation between the yes and no categorical responses from *Subscription*; therefore, we will not move forward with a Principal Component Analysis (PCA) for our continuous variable reduction. We are however, going to take a closer look at the level of linearity correlation matrix. In [Figure 17](#_Figure_17) you can see a large correlation matrix that gives us the correlation gradient from highly negatively correlated to highly positively correlated from dark orange to dark blue respectively. When reviewing this matrix, it became clear that multicollinearity would be an issue if some of these continuous variables were not removed. We moved forward with removing the following variables: *pdays, nr.employed, emp.var.rate, age* [[Figure 18](#_Figure_18)]. A secondary correlation matrix showing little to no multicollinearity can be found in [Figure 19.](#_Figure_19)

## After our continuous variable response analysis, we move forward to our categorical variable response analysis. In order to complete this we wrote a function that created a plot grid of all the bar graphs that were filled to percent of total for easy comparison across categories. However, the plot grid was difficult to read. So, we moved forward with leveraging that function for the individual analysis of each categorical variable towards our dependent variable *Subscription*. This function can be found in [Figure 20](#_Figure_20) and every corresponding graph for each categorical variable graphed against *Subscription* can be found in [Figure 21](#_Figure_21). In these bar graphs we can see where the yes *Subscription* response differs greatly for certain categories, this is our indication that the particular categorical variable graphed should be leveraged for our model. After completion of reviewing these graphs individually we have determined there are a group of variables we can leave behind [[Figure 22](#_Figure_22)] and move forward with our final collection of variables.

### NOTE: When reviewing the categorical variables, it stood out to us that many of them might benefit from slight manipulation and transformation in order to strengthen our model. However, this would make interpretation much more difficult. We address this in [Objective 2: Complex Logistic Model](#_Complex_Logistic_Regression) by creating a secondary logistic model that is more  with these altered variables. The actions we take for these manipulations will be outlined and is our secondary EDA for a logistic model completion.

## Our final step in our EDA process will be to do a quick statistical summary on our final selected variables to make sure that all variables look good to move forward with modeling. In [Figure 23](#_Figure_23) we have completed this step and are ready to move forward with our simple logistic model for [Objective 1](#_Objective_1:_Simple).

# Objective 1: Simple Logistic Regression Model

## Problem Statement and Approach Review

## This analysis consists of a simple logistic approach in order to determine the best variables in the given data set to predict the subscription (y) variable outcome of yes or no. Through our [EDA](#_Exploratory_Data_Analysis) we were able to reduce the variable options as well as develop intimate knowledge of our independent variables. Therefore, we felt it more valuable to utilize our intuition to build our simple logistic model. Validation of results was accomplished by splitting the data into testing and training sets. The training set was partitioned such that it had an equal number of “yes” and “no” outcomes by splitting the “yes” outcomes in half and randomly selecting an approximately equal number of accompanying “no” responses; the rest of the data was left to the test set. These training and testing partitions were exported to csv files for use with other models in R as well for consistency.

## Model Selection

### Type of Selection: Manual / Intuitive

### The initial regression model was built using variables that distinguished themselves through the EDA. These runs were accomplished through PROC LOGISTIC in SAS. In the first run, two of those variables (education, previous) did not show statistical significance in the Type 3 Analysis of Effects and were removed [[Figure 24](#_Figure_24)]. All remaining variables were significant in the successive run [[Figure 25](#_Figure_25)]. Comparisons of the ROC Curves between the training set [[Figure 26](#_Figure_26)] and the test set [[Figure 27](#_Figure_27)] did not reveal any issues between the train and test sets such as overfitting. The Area under the curve for the two were similar. The Fit statistics for the model were scored on the test set to get a realistic idea of performance [[Figure 28](#_Figure_28)]. This was important because the training set was made with an equal proportion of outcomes to prevent issues with running the models, but the test set was more representative of the original data where there were far more “no” responses than “yes”. [[Figure 29](#_Figure_29)] is a frequency table of the results on the test set predictions that shows an approximate sensitivity of 65% and specificity of 83%. These are based on the default prior probability of the model of .5.

#### Note: We will further discuss in our [Objective 2: Conclusion](#_Conclusion_/_Discussion) on the potential choices to make when tuning the prior probability.

### For model comparison, another model was built using LASSO in the PROC HPGENSELECT function to select variables from the entire set of original variables (main effects only). When the selected variables were re-run in PROC LOGISTIC and checked again for significance, the remaining variables were almost identical to those chosen through the [EDA](#_Exploratory_Data_Analysis). The job variable was the only one excluded using LASSO; this is not a surprising consequence as this variable had the highest p value in the Type 3 analysis of effects of our initial model. An examination of the results [[Figure 30](#_Figure_30)] showed no benefit to changing our initial model for the slight difference in variable selection so it was left as is.

### Assumptions

#### Lack of Fit Test

#### The Deviance and Pearson Goodness of Fit tests did not show any signs of poor fit [[Figure 31](#_Figure_31)].

#### Influential Point Analysis

#### [Figure 32](#_Figure_32) shows a leverage plot. While there appears to eb an outlier, it does not have enough leverage to cause any concern in our model.

## Parameter Interpretation

### Interpretation

### The binary logistic regression parameter estimates (log odds ratios) for the model and their standard errors are shown in [[Figure 33](#_Figure_33)]. These are the coefficients used when the model predicting the log odds of a “yes” response is defined by the equation *log[p/(1-p)] = 𝛃0 + 𝛃1X1 + 𝛃2X2 + 𝛃3X3 … 𝛃nXn* where *p* is the probability of a “yes” response. Our model can be represented using the equation log[*pyes*/(1-*pyes*)] = -35.9 -.495\**Jobadmin* - .605\**Jobblue-collar* - .196\**Jobentrepreneur* - .743\**Jobhousemaid* - …***{see*** [**Figure 33**](#_Figure_33)***}*** … - .520\**euribor3m* + .433\**cons\_price\_idx* + .047\**cons\_conf\_idx*

### Confidence Intervals

### [Figure 34](#__Figure_35_1) displays the predictor variables in terms of odds ratios and their confidence intervals. These represent the exponentiation of the model parameter estimates. These estimates can be used to interpret the effect of a single unit change of any predictor on the odds ratio of obtaining a “yes”, given the other variables in the model are held constant. For example, for a one unit increase in the consumer price index (*cons\_price\_idx*), the odds ratio for getting a “yes” response will increase by 1.54 (95% confidence interval of this change of [1.03,1.07]), holding all other predictors constant.

## Conclusion

## Based on the construction of our initial regression model and an extensive exploratory data analysis, we had some suspicions about how the nature of this dataset might limit the maximum effectiveness of any models trying to predict subscriptions using the variables provided. We will examine this further in [Objective 2](#_Objective_2:_Competing), but due to the lack of many true continuous variables (most of the numeric values are in groups by time period) there is not likely much improvement to be made using more complex models. The most influential variables in the model are *euribor3m* and *poutcome* (see high Chi-sq scores in [[Figure 35](#_Figure_35_2)]). Adding variables after these marginally increases predictive capability and this is most apparent when looking at ROC curves for models that exclude these variables or only include these two variables in comparison to our regression model [[Figure 36](#_Figure_36)]. We can still glean some confidence from a contrast of these ROC curves [[Figure 37](#_Figure_37)] that shows that we did produce statistically significant improvement in our model by adding additional variables. It could also be indicative of the limitations of this dataset and its maximum potential for modeling that using LASSO to select from all of the available variables produced an almost identical list of variables to those we had chosen.

# Objective 2: Competing Models

## Model Selection

## NOTE: Train and Test sets were formed to be leveraged by all the following models for consistency and comparable metrics.

### Complex Logistic Regression

### In [Objective 1](#_Objective_1:_Simple), we did an extensive [EDA](#_Exploratory_Data_Analysis) of the variables to determine which would be best for the logistic regression model and compared our model to one using LASSO to select variables. We also attempted using LASSO to select among all two-way interactions of variables we included in our initial model as well as only including interactions we thought might be significant. None of these models improved upon our initial regression model in any meaningful way, with ROC Curve analyses of the models all resulting in a maximum AUC of around .77-.79.  For the best area under the curve (AUC) results obtained from these more complex models see ROC Curves for the train [[Figure 38](#_Figure_38)] and test [[Figure 39](#_Figure_39)] sets. Although we sometimes saw slight improvements like this in the training set, the results on the test set were marginal or worse. For this reason, we decided to leave our original logistic regression model as is and see if any other modeling techniques could out-perform it

### Linear Discriminant Analysis (LDA) / Quadratic Discriminant Analysis (QDA)

### During the [EDA](#_Exploratory_Data_Analysis) we surfaced very little separation from a simple two-dimensional comparison view of the continuous variables. If you view [Figure 16](#_Figure_16) you can see it appears that all the variables very randomly overlap without any sort of solid separation. In further investigating the ability to run an LDA/QDA on this data, we completed the same pairs plot on the only continuous variables left [[Figure 40](#_Figure_40)]. Again, we don’t see any clear separation, instead we moved forward with forming an LDA and QDA using a three-dimensional model from all three continuous variables we had after our completed [EDA](#_Exploratory_Data_Analysis).

### We leveraged the same test and train sets as the rest of the models outlined in the [Objective 2: Note](#_NOTE:_Train_and). First, properly filtered our data set into the desired variables, this can be seen in [Figure 41](#_Figure_41). Then we did a *yes* and *no* results check.we created a three-dimensional plot on our test set to see if there was any evident separation between the two results for the dependent variable, Subscription [[Figure 42](#_Figure_42)].

### Once we had made sure the data was where it needed to be, we proceeded to complete both the LDA model. As you can see in [Figure 43](#_Figure_43), we have used the confusionMatrix function and pulled the accuracy, sensitivity, and specificity after we modeled our LDA on the *banktrain* set and completed the predictions on the *banktest* set. It resulted in the following percentages: 72.07%, 72.15%, and 70.84 respectively. The accompanying ROC curve for this matrix is in [Figure 44](#_Figure_44) with an Area Under the Curve (AUC) of .73. This is much higher than expected during the [EDA](#_Exploratory_Data_Analysis) evaluation of separation from [Figure 16](#_Figure_16) and [Figure 40](#_Figure_40). In order to see if the QDA would help our metrics we changed our code and performed a QDA. The resulting accuracy, sensitivity, and specificity metrics were: 89.62%, 92.46%, and 47.29% respectively [[Figure 45](#_Figure_45)]. The accompanying ROC curve can be found in [Figure 46](#_Figure_46) with its AUC of .77.

## NOTE: The following is an explanation of recycled functions utilized when comparing our [KNN](#_K-Nearest_Neighbor_(KNN)), [Random Forest](#_Random_Forest), and [MLR](#_Complex_MLR) models. This was completed in order to make cleaner, more readable code.

## First, we created a function to make labels for the models to easily compare in plots in an automated manner. We were able to achieve this by accessing the label property of the caret object. By updating this value, we are directly able to access the label [[Figure 47](#_Figure_47)]. Once we had our label setting function, we created a function to make predictions on a list of multiple models using the same test set. This is a great way to do a cross comparison against multiple models because we know we are testing them on the same data set. All referenceable code can be found in [Figure 48](#_Figure_48). Once we pass in our models, we output a cleaned data frame containing performance metrics such as accuracy, specificity, and sensitivity like in the example below (the results of which will be discussed in the subsequent sections) [[Figure 49](#_Figure_49)]. Based on that data frame output, we can save our output information and pass it in the function in order to generate bar plots of all the models in the data frame. By using the model labels, we defined with the *“set\_model\_labels”* function, we can easily identify which model is which. This is a great way to visually compare our outputs [[Figure 50](#_Figure_50)]. Finally, we generate a multi-model ROC curve based on the list of models we pass. Similar to the *“model\_bar\_plots”* function, using the labels we set for each model in the *“set\_model\_label”* function allows us to automatically classify the models in any way that we’d like to describe them. In the model building phase, all models were passed a controller to help tune the model and add additional features such as a 10-fold cross validation and the return of predictions for use with our ROC plots [[Figure 51](#_Figure_51)]. By using the aforementioned approach, we were able to quickly and cleanly compare multiple models in a reproducible way. Next we will review our [KNN](#_K-Nearest_Neighbor_(KNN)), [Random Forest](#_Random_Forest), and [MLR](#_Complex_MLR) models’ results.

### K-Nearest Neighbor (KNN)

### One of the other model options explored was the K-Nearest Neighbor model (KNN). Running on a full, reduced, and engineered model, we ran 10 different iterations with k from 1-10 with a 10-fold cross validation set to maximize the ROC value(caret parameter).  Using the base model with an optimal k-value of 9, we were able to achieve a pretty high starting accuracy from the raw, under sampled data set [[Figure 52](#_Figure_52)]. Based on the entire dataset, we can see the following model metrics below. Note that our chosen *k* value of 9 was that which maximized the ROC’s AUC [[Figure 53](#_Figure_53)]. By generating a confusion matrix for our model, we can check our performance metrics on the entire test data set. Notice here that the KNN model had a slightly lower sensitivity due to the unbalance of the data set itself. However, our attempt to under sample the data set boosted the sensitivity to a much more reasonable value [[Figure 54](#_Figure_54)].

### Even though we had a fairly good starting point, we wanted to focus on reducing the total features in the model to alleviate concern of overfitting the model. After a combination of taking only significant features from our original logistic regression model and performing some additional ad hoc EDA, we were able to reduce the total features in the model from 21 to 9 without a very noticeable reduction in specificity. You can see in [Figure 55](#_Figure_55) we have created a reduced data set upon which we select only specific features and use that to train our new KNN model. By checking our model output, we see that our optimal *k* value was once again 9. This value sought to maximize the ROC area under the curve based on the reduced training dataset [[Figure 56](#_Figure_56)]. Based on the confusion matrix of the reduced feature predictions [[Figure 57](#_Figure_57)], we can see that our specificity took a slight reduction, with our sensitivity increasing to lead to an overall increase of model accuracy to 78.7%. Although we took a slight reduction in specificity, we were able to further simplify our model.

### In order to further optimize the model, we sought to combine features to create a categorical interaction by utilizing a concatenation of each category, separated by an underscore character. By utilizing the new features of *“month\_marketing campaign”* [[Figure 72](#_Figure_73)] and *“job\_seniority*” (age over 60) [[Figure 73](#_Figure_73)] we were able to remove 4 more features. Additionally, checking correlation between features allowed us to select only one of three highly correlated features (*emp\_var\_rate*) [[Figure 74](#_Figure_74)]. The result was an extremely similar score with an even more reduced model comprising 5 features (4 if you don’t count the response). Although we were not able to achieve a much higher score, dropping 16 features has allowed for a much stronger model that is less prone to overfitting [[Figure 58](#_Figure_58)]. Therefore, we’d argue that we were successful in creating an improved model over the raw data set. Based on the model output, we can see that the engineered model utilized a k value of 5 to maximize the ROC’s AUC [[Figure 59](#_Figure_59)]. By using our model to generate a confusion matrix, we see that the total accuracy increased to 82.29% with a sensitivity of 83.61% and a specificity of 62.52% [[Figure 60](#_Figure_60)].

### Finally, to choose an optimal KNN model, we are comparing models based on the ROC’s AUC metric. By plotting the base, reduced, and engineered case models against one another in a ROC plot, we see that the engineered KNN model both outperformed the other KNN models as well as created a much more simple, robust model [[Figure 61](#_Figure_61)].

### Random Forest

### One of the options we explored was the Random Forest model due to its ease of interpretation and the ability to model complex behavior without adding extra complexity such as interactions. Additionally, the Random Forest model is often used as a first model by many data scientists to create a baseline upon which to compare other models to. In order to get an idea of what the baseline prediction ROC was for the random forest model, we started by training the model on all predictors of the under sampled data set. Using all of the features from [Figure 62](#_Figure_62), we can see that we were able to achieve an ROC of 0.79 which is approximately equal to the first version of our complex logistic regression model and is an improvement over the [KNN](#_K-Nearest_Neighbor_(KNN)) model. By examining the model output in [Figure 63](#_Figure_63), we see that our optimal mtry value was 2. This is the value that maximized ROC the greatest. By performing analysis on the test data set for the full model, we see that the model scored a sensitivity of 86.18% with a specificity of 63.31% for a combined accuracy of 84.75% [[Figure 64](#_Figure_64)].

### The next steps were to see which variables were not needed, to reduce the possibility of overfitting our model. By checking the significant variables from the first version of the full logistic regression base model (which used the same predictors), we narrowed down to only those features that bore significance to the base case model. In the process, we dropped from 21 features to 9 features [[Figure 65](#_Figure_65)]and were able to attain a very comparable model score yet reduce model complexity and boost interpretability. By checking the model output of our reduced model, we once again see an optimal *mtry* value of 2 [[Figure 66](#_Figure_66)]. Running our predictions on the reduced test set shows a sensitivity of 84.10% and a specificity of 64.49%. In all this model experience an increase in specificity and a decrease in sensitivity. For an instance in which we are more focused on accuracy predicting “yes” opposed to total accuracy, this would be a better option than the full model. The reduction of features also aids in making this model much simpler [[Figure 67](#_Figure_67)].

### Now that we were able to achieve a very similar score with a much leaner model, we utilized some exploratory analysis to create some “engineered” features that may improve score. Some of the things that stood out was that *job, month, age* (those over 60 correspond to the high number of “retiree” jobs), and success of the marketing campaign were all pretty crucial identifiers in the data. By implementing some “interactions” between the factor levels from combining multiple features and an underscore, we were able to see more defined separation. Our new interaction variables were *month\_marketingcampaign* displayed in [Figure 68](#_Figure_68). Another variable was *job\_age* which can be seen in [Figure 69](#_Figure_69) and the code to see our engineer variables can be found in [Figure 70](#_Figure_70). Additionally, we realized the high correlation between the remaining numeric features [[Figure 71](#_Figure_71)] and opted to keep *emp\_var\_rate* because it explained the highest correlation with the other numeric features, thus we believe it to better represent the trends that we sought out in the remaining three numeric features.

### By checking our engineered model output [[Figure 72](#_Figure_72)], we see that mtry of 2 is still the best parameter to maximize ROC [[Figure 73](#_Figure_73)] area under the curve. In creating a confusion matrix on our engineered random forest model [[Figure 74](#_Figure_74)], we see that the specificity is now 63.88% which is approximately equal to the specificity of our original full model. In contrast, the engineered model experiences a reduction in model sensitivity compared to the full model.

### Comparing Engineered Models

### In doing a final comparison between the models, we can see that among KNN, Random Forest, and MLR, the engineered multiple logistic regression had the highest specificity. The models’ sensitivity comparison can be found in [Figure 75](#_Figure_75), the models’ specificity comparison can be seen in [Figure 76](#_Figure_76), and the models’ accuracy comparison can be seen in [Figure 77](#_Figure_77). The comparative ROC curves between KNN, Random Forest, and MLR can be seen in [Figure 78](#_Figure_78).

## Main Analysis Content

### We used area under the curve (AUC) of receiving operating characteristic curves (ROC curves) as the performance metric for comparing models.

### Below are our listed results of the AUC for each of our models. The chart illustrates our early discussion points where we were unable to produce a more powerful regression model by adding more complexity. This led us to conclude that the simpler logistic regression model formed in [Objective 1](#_Objective_1:_Simple) would be the most effective choice.

|  |  |
| --- | --- |
| **Model** | **ROC AUC** |
| Simple Logistic | .79 |
| Complex Logistic | .79 |
| Linear Discriminant Analysis | .73 |
| Quadratic Discriminant Analysis | .77 |
| K-Nearest Neighbor | .77 (engineered model) |
| Random Forest | .78 (reduced model) |

## Conclusion / Discussion

## Upon reviewing all of our given models it seems near impossible to find a single one that stands out from the other. We have taken each model and revisited our EDA per that model’s parameters; this included creating interactions, feature reduction, and variable selection manipulation in an attempt to increase our model interpretability as well as improved Accuracy, Specificity, and Sensitivity performance metrics.

All our models produced relatively comparable performance metrics, but we decided to officially recommend the Simple Logistic model for both parameter interpretation and prediction, as it appeared to perform the best. We chose the Simple Logistic model for interpretation because these variables are not manipulated and will be easily translated into a real-world scenario, and still recommend the model for prediction because any attempt at adding complexity does not appear to produce improved predictive performance.

# Appendix

## Figure 1

A screenshot of a social media post

Description automatically generated



## Figure 2

A screenshot of a cell phone

Description automatically generated

## Figure 3

A screenshot of a cell phone

Description automatically generated

## Figure 4

A screenshot of a cell phone

Description automatically generated

## Figure 5

A screenshot of a social media post

Description automatically generated

## Figure 6

A screenshot of a cell phone

Description automatically generated

## Figure 7

A screenshot of a social media post

Description automatically generated

## Figure 8

A screenshot of a cell phone

Description automatically generated

## Figure 9

A screenshot of a cell phone

Description automatically generated

## Figure 10

A screenshot of a social media post

Description automatically generated

## Figure 11

A screenshot of a social media post

Description automatically generated

## Figure 12

A screenshot of a cell phone

Description automatically generated

## Figure 13

A picture containing screenshot

Description automatically generated

## Figure 14

A screenshot of a cell phone

Description automatically generated

## Figure 15

A screenshot of a cell phone

Description automatically generated

A close up of a logo

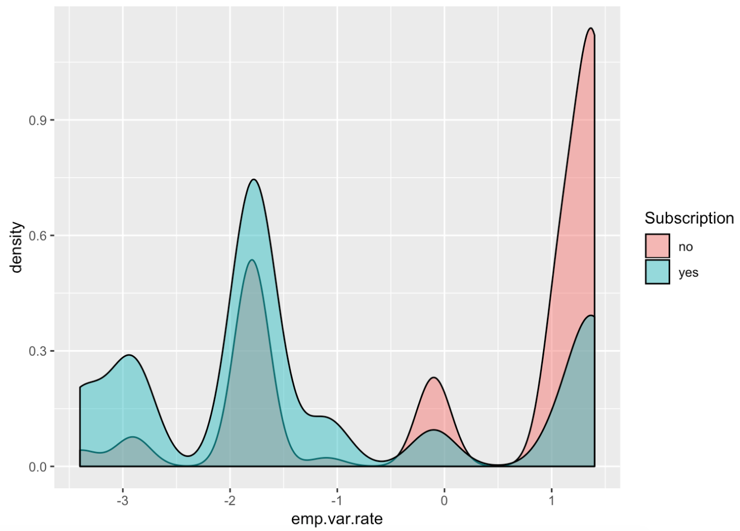
Description automatically generatedA close up of a white wall

Description automatically generated

A screenshot of a cell phone

Description automatically generatedA picture containing white, man, water

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A close up of a logo

Description automatically generatedA close up of a logo

Description automatically generated

A close up of a logo

Description automatically generated

## Figure 16

A screenshot of a cell phone

Description automatically generated

## Figure 17

A screenshot of a cell phone

Description automatically generated

## Figure 18



## Figure 19

A screenshot of a cell phone

Description automatically generated

## Figure 20

A screenshot of a social media post

Description automatically generated

## Figure 21

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

## Figure 22

## 

## Figure 23

A close up of text on a white background

Description automatically generated

## Figure 24

**proc** **logistic** data=banktrain plots=ROC;

class Subscription job education default contact month poutcome / param=ref;

model Subscription(event='yes') = job default education contact month campaign previous poutcome euribor3m cons\_price\_idx cons\_conf\_idx / scale=none aggregate lackfit ctable outroc=trainroc;

output out=trainpreds;

score data=banktest out=testpreds outroc=testroc fitstat;

**run**;

A screenshot of a cell phone

Description automatically generated

## Figure 25

**proc** **logistic** data=banktrain plots=ROC;

class Subscription job default contact month poutcome / param=ref;

model Subscription(event='yes') = job default contact month campaign poutcome euribor3m cons\_price\_idx cons\_conf\_idx / scale=none aggregate lackfit ctable outroc=trainroc;

output out=trainpreds;

score data=banktest out=testpreds outroc=testroc fitstat;

**run**;

A screenshot of a cell phone

Description automatically generated

## Figure 26

**proc** **logistic** data=banktrain\_raw plots=ROC;

class Subscription contact default month poutcome / param=ref;

model Subscription(event='ye') =  contact campaign default month poutcome cons\_price\_idx cons\_conf\_idx euribor3m / scale=none aggregate lackfit outroc=rtrainroc;

output out=rtrainpreds;

score data=banktest\_raw out=rtestpreds outroc=rtestroc fitstat;

**run**;

**proc** **freq** data=rtestpreds;

tables I\_Subscription\*Subscription;

**run**;

A close up of a map

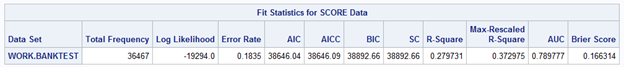
Description automatically generated

## Figure 27

A close up of a map

Description automatically generated

## Figure 28



## Figure 29

A screenshot of a cell phone

Description automatically generated

## Figure 30

**proc** **hpgenselect** data=banktrain\_raw;

class Subscription marital housing loan contact day\_of\_week job education default month poutcome;

model Subscription(event='ye') = marital previous contact campaign job education default day\_of\_week housing loan month poutcome age cons\_price\_idx cons\_conf\_idx euribor3m emp\_var\_rate/ dist=binary;

selection method=lasso details=all;

**run**;

\*Run 2: Run with variables selected by LASSO;

**proc** **logistic** data=banktrain\_raw plots=ROC;

class Subscription contact default month poutcome / param=ref;

model Subscription(event='ye') =  previous contact campaign default month poutcome age cons\_price\_idx cons\_conf\_idx euribor3m / scale=none aggregate lackfit outroc=rtrainroc;

output out=rtrainpreds;

score data=banktest\_raw out=rtestpreds outroc=rtestroc fitstat;

**run**;

\*Run 3: Remove insig variables and run again;

**proc** **logistic** data=banktrain\_raw plots=ROC;

class Subscription contact default month poutcome / param=ref;

model Subscription(event='ye') =  contact campaign default month poutcome cons\_price\_idx cons\_conf\_idx euribor3m / scale=none aggregate lackfit outroc=rtrainroc;

output out=rtrainpreds;

score data=banktest\_raw out=rtestpreds outroc=rtestroc fitstat;

**run**;

**proc** **freq** data=rtestpreds;

tables I\_Subscription\*Subscription;

**run**;

A close up of a map

Description automatically generated

## Figure 31

A screenshot of a cell phone

Description automatically generated

## Figure 32

A screenshot of a cell phone

Description automatically generated

## Figure 33

## A screenshot of a cell phone Description automatically generated

## Figure 34

A screenshot of a cell phone

Description automatically generated

## Figure 35

A screenshot of a cell phone

Description automatically generated

## Figure 36

**proc** **logistic** data=banktrain plots=ROC;

class Subscription job default contact month poutcome / param=ref;

model Subscription(event='yes') = job default contact month campaign poutcome euribor3m cons\_price\_idx cons\_conf\_idx / scale=none aggregate lackfit ctable outroc=trainroc;

output out=trainpreds;

score data=banktest out=testpreds outroc=testroc fitstat;

roc 'Excluding Euribor3m' job default contact month campaign poutcome cons\_price\_idx cons\_conf\_idx;

roc 'Excluding Euribor3m and poutcome' job default contact month campaign cons\_price\_idx cons\_conf\_idx;

roc 'Only Euribor3m and poutcome' poutcome euribor3m;

roccontrast reference('Excluding Euribor3m and poutcome') /estimate;

**run**;

A close up of a map

Description automatically generated

## Figure 37

A screenshot of a cell phone

Description automatically generated

## Figure 38

**proc** **logistic** data=banktrain\_raw plots=ROC;

class Subscription contact default month poutcome job / param=ref;

model Subscription(event='ye') = job contact campaign default month poutcome cons\_price\_idx cons\_conf\_idx euribor3m\*month job\*age month\*campaign/ scale=none aggregate lackfit outroc=rtrainroc;

output out=rtrainpreds;

score data=banktest\_raw out=rtestpreds outroc=rtestroc fitstat;

**run**;

**proc** **freq** data=rtestpreds;

tables I\_Subscription\*Subscription;

**run**;

A close up of a map

Description automatically generated

## Figure 39

A close up of a map

Description automatically generated

## Figure 40

## A screenshot of a cell phone Description automatically generated

## Figure 41

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

## Figure 42

A screenshot of a cell phone

Description automatically generated

A picture containing map

Description automatically generated

## Figure 43

## A screenshot of a social media post Description automatically generated A screenshot of a social media post Description automatically generated

## Figure 44

## 

## A close up of a map Description automatically generated

## Figure 45

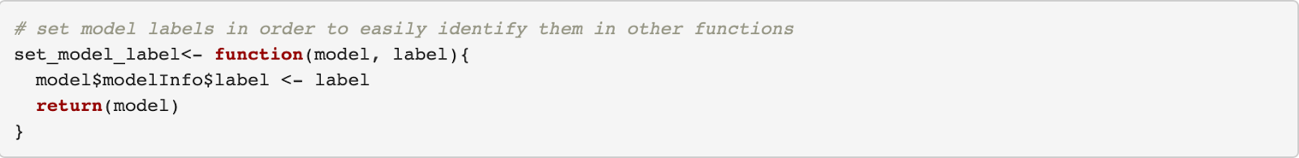
## A screenshot of a social media post Description automatically generatedA screenshot of a cell phone Description automatically generated

## Figure 46

## 

## A close up of a map Description automatically generated

## Figure 47



## Figure 48

A screenshot of a social media post

Description automatically generated

## Figure 49

A screenshot of a cell phone

Description automatically generated

## Figure 50

A screenshot of a social media post

Description automatically generated

## Figure 51

A close up of a logo

Description automatically generated

## Figure 52

A screenshot of a cell phone

Description automatically generated

## Figure 53

A screenshot of text

Description automatically generated

## Figure 54

A screenshot of a cell phone

Description automatically generated

## Figure 55

A screenshot of a cell phone

Description automatically generated



## Figure 56

A screenshot of text

Description automatically generated

## Figure 57

A screenshot of a cell phone

Description automatically generated

## Figure 58

A screenshot of a social media post

Description automatically generated



## Figure 59

A screenshot of text

Description automatically generated

## Figure 60

A screenshot of a cell phone

Description automatically generated

## Figure 61

A close up of a map

Description automatically generated

## Figure 62

A screenshot of a cell phone

Description automatically generated

## Figure 63

A screenshot of text

Description automatically generated

## Figure 64

A screenshot of a cell phone

Description automatically generated

## Figure 65

A screenshot of a cell phone

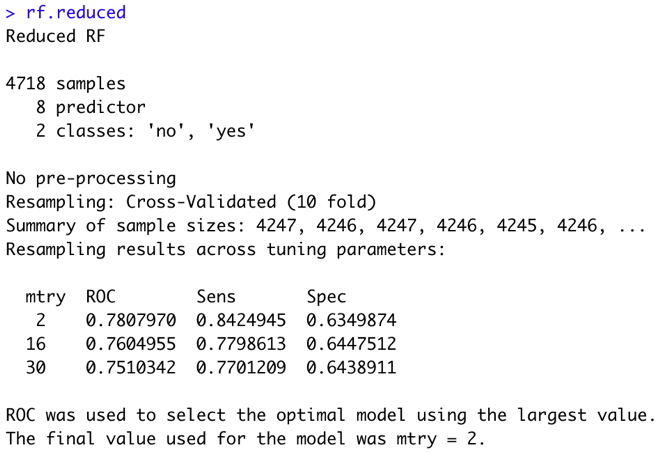
Description automatically generated



A screenshot of a cell phone

Description automatically generated

## Figure 66



## Figure 67

A screenshot of a cell phone

Description automatically generated

## Figure 68

A screenshot of a cell phone

Description automatically generated

## Figure 69

A screenshot of a cell phone

Description automatically generated

## Figure 70

A screenshot of a social media post

Description automatically generated



## Figure 71

A screenshot of a cell phone

Description automatically generated

## Figure 72

A screenshot of text

Description automatically generated

## Figure 73

A close up of a map

Description automatically generated

## Figure 74

A screenshot of a cell phone

Description automatically generated

## Figure 75

A screenshot of a cell phone

Description automatically generated

## Figure 76

A screenshot of a cell phone

Description automatically generated

## Figure 77

A screenshot of a cell phone

Description automatically generated

## Figure 78

A close up of a map

Description automatically generated