**Summer 2021 Project 2**

**Rashmi Patel, Taylor Bonar, Michael Burgess**

**Introduction:**

The goal of the following analysis seeks to identify and describe the key factors and interactions which affect client decision to subscribe for term deposit and predict if the customer will make a subscription to term deposit or not. The dataset is obtained from [UCI Machine Learning Repository](https://archive.ics.uci.edu/ml/datasets/Bank+Marketing) which related to the direct marketing campaign of a Portuguese banking institution.

**Data Description:**

The direct marketing campaign data includes of a multitude of variables including, age, job type, marital status, education, etc. and a categorical response variable. Several data points in multiple variables which was unknown were set to NA. This dataset is collected over time from May 2008 to November 2011. The dataset is delivered as delimiter separated values with 41,188 observations, 20 inputs and one response variable. Of the inputs, there were 10 continuous variables and 10 categorical string variables.

**Exploratory Data Analysis:**

**Chart, surface chart

Description automatically generated with medium confidence**When the data was examined with some basic plots, we found that some categorical variables have unknown group (1.5% of the data) in them which was contributing nothing to the analysis being done. So, moved forward replaced those unknown values with NA. After the replacement, we removed those NA values from the data. After getting the cleaned data, we started doing exploratory data analysis.

A picture containing graphical user interface

Description automatically generatedThe following results were found:

1. Out of 30488, 26629 clients (87.34%) did not choose subscription for term deposit. Only 3859 clients (12.66%) chose to subscribe for term deposit. Keeping this in mind, we’ll need to ensure either our sampling is weighted, or consider algorithms that can oversample or under sample when fitting our logistic regression model, otherwise we may introduce bias and lower our accuracy on predicting when a client successfully subscribes to a bank term deposit.
2. While examining the normality of numeric variables using histograms, we found that there are some potential normality issues. Age, Duration, Campaign, Previous campaigns are the variables that are highly right-skewed, and others seems to be distorted across different with some visual skewness.
3. While examining the categorical variables, we found the following results:
4. In job variable, the highest 3 categories were- admin with 25.3%, blue-collar with 22.47% and technician with 16.37%. Other job categories contribute only to 35.85% in which 0.8% were categorized as “unknown”.
5. In marital status, 60.52% of the clients were married with 28.09% single, 11.2% divorced and 0.19% were categorized as “unknown”.
6. Looking at the education of clients, the top 3 category have 29.54% of university graduates, 23.1% of high school education, and 14.68% of basic 9 year. Other education categories contribute up to 32.68% in which 4.2% were “unknown” and 0.04% were illiterate.
7. In housing variable which describing whether clients have housing loan or not, 52.38% of the clients have housing loan, 45.21% of the clients did not have housing loan and remaining 2.4% was categorized as “unknown”.
8. Looking at the loan variable which describe whether the client have personal loan or not, 82.43% of the clients have personal loan, 15.17% did not have personal loan and remaining 2.4% is categorized as “unknown”.
9. In preferred method of contact, 63.47% of the clients chose cellular communication type and 36.53% of the clients chose telephone to be preferred method of contact.
10. In last contact month of the year, 33.43% of the clients were last contacted in the month of May, 17.42% in the month of July and 15% in the month of August. No clients were contacted in the month of January and February.
11. Talking about the day of the week for last contact, 20.94% of the clients were last contacted on Thursday, 20.67% on Monday and the rest of them were last contacted in other 3 business days.
12. In poutcome variable which describes the outcome of previous marketing campaign, we found that only 3.33% of the marketing campaign were successful, 10.32% were failure and 86.34% were non-existent.
13. In default variable which describes whether a client has credit in default or not, we found that only 0.01% of the clients had credit in default. 79% of the client did not have credit in default and 20.87% were categorized as “unknown”.
14. A picture containing chart

    Description automatically generatedWhen we analyzed the multicollinearity of variables (correlation between numeric variables) using corrplot () function, we found the following results:
    1. emp.var.rate and euribor3m are 97% positively correlated.
    2. nr.employed and euribor3m are 94% positively correlated.
    3. nr.employed and emp.var.rate are 90% positively correlated.
    4. cons.price.idx and emp.var.rate are 77% positively correlated.
    5. cons.price.idx and euribor3m are 67% positively correlated.
    6. cons.price.idx and nr.employed are 49% positively correlated.
    7. pdays and previous are 59% negatively correlated.
    8. Nr.employed and previous are 49% negatively correlated.
    9. euribor3m and previous are 44% negatively correlated.
    10. emp.var.rate and previous were 40% negatively correlated.

**Linear Discriminant Analysis:**

In exploring possible models for predicting whether a client would subscribe to a term deposit or not, we applied a linear discriminant model to the data. The data were reduced to include only continuous predictors and were examined to see if the data met the assumptions of the linear discriminant model.

A picture containing chart

Description automatically generated

Although the data meets the assumptions of independence and no multicollinearity, it lacks any clear linearity or consistency in its variance. For these reasons, a linear discriminant model would not be recommended for making accurate predictions.

Although the model isn’t recommended, we applied the linear discriminant model for the sake of comparison. The data were split into an 80/20 randomly sampled train/test set consiting of only the continuous variables from the original data set. As expected, the accuracy is lacking at only 66.89% as shown in our confusion matrix or 68.4% as shown in the ROC curve. Because the data set is heavily unbalanced the linear discriminant model only achieved a sensitivity of 29.36% but had a specificity of 88.08%. A technique like under sampling may help to balance the sensitivity and specificity. Including categorical variables may also increase the accuracy and provide more insight into the model’s decision criteria, but as stated previously the linear discriminant model will never provide great accuracy for the given data.

|  |  |
| --- | --- |
| A picture containing text, receipt  Description automatically generated | Chart, line chart, scatter chart  Description automatically generated |

**LASSO Logistic Regression Model**

When utilizing a dataset with many predictors, it can be preferable to use an automatic variable selection method to help reduce unnecessary variables. In doing so for logistic regression, we opted to use LASSO as our method. Before creating our LASSO regression model, we needed to determine our ideal value for controlling the shrinkage of coefficients, lambda.

To pick our ideal lambda, we need to perform k-fold cross-validation to reduce our cross-validation prediction error rate. As can be observed from our k-fold cross validation plot, the vertical dashed line indicates that the log of our optimal value of lambda is approximately -5 (exact lambda is) to minimize our prediction error. Additionally, we can infer what coefficients are non-zero from our coefficient plot of our cross-validation method for our minimal lambda, in which we can observe that we have two coefficients exactly equal to zero (marital status of single and education level of high school).

|  |  |
| --- | --- |
| Graphical user interface  Description automatically generated | Graphical user interface, application, table  Description automatically generated |

However, we do have an additional option in our regularization method to balance accuracy and simplicity. As we elected to use a cross-validation technique, we also can determine a lambda to give us our most simple model that lies within one standard error of our optimal value of lambda. The following coefficient plot is much simpler and only 32 variables have a coefficient exactly equal to zero:

Graphical user interface

Description automatically generated with low confidence

When choosing which lambda value to use when fitting our model, we generally want a balance between accuracy but also simplicity. That way, we can easily interpret the model if need be. As we seek to understand how coefficients interact within the model and understanding how our data comes into play, we shall resort to the simpler model and use a lambda within one standard error of our optimal lambda for our LASSO model instead of focusing on accuracy.

Once we fit our model, we determined that our test data was roughly 90% accurate in predicting whether a client would or would not subscribe to a bank term deposit. With this in mind, we can conclude our coefficients from our simpler model may be most relevant and significant for this direct marketing campaign.

**Analysis of Models**

Lasso Logistic Regression Model:

PCA Model:

LDA Model:

**Appendix**

**Github:** <https://github.com/tbonar/MSDS-6372-Project2>

**Rmd file**

---

title: "Bank Analysis And Modeling"

author: "Taylor Bonar & Michael Burgess & Rashmi Patel"

date: "7/30/2021"

output: html\_document

---

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

library(tidyverse) # Data handling

library(naniar) # Viz on Missing Data

library(GGally) # Graphs!

library(DataExplorer) # More Graphs!

library(ggplot2) # More more Graphs!

library(cowplot) # ggplot2 defaults

library(funModeling) # Helpful Functions for EDA Function

library(Hmisc) # Helpful Functions for EDA Function

library(caret) # Data Partitioning

library(glmnet) # Modeling

library(coefplot) # Coefficient modeling of glmnet objects

library(corrplot) # Correlation Plotting

library(car) # vif function

library(pROC) # ROC Curves

setwd(".")

basic\_eda <- function(data) # Sample Function Source: https://blog.datascienceheroes.com/exploratory-data-analysis-in-r-intro/

{

glimpse(data)

print(status(data))

freq(data)

print(profiling\_num(data))

plot\_num(data)

describe(data)

}

# logit^{-1}: Can use to convert results of logistic regression to probability

invlogit <- function(x) {1 / (1 + exp(-x))}

# Pseudo R^2

psuedo\_rsq <- function(glmmodel) {

# Creating Psuedo R^2

full.ll.null <- glmmodel$null.deviance/-2

full.ll.proposed <- glmmodel$deviance/-2

full.ll.rsq <- (full.ll.null - full.ll.proposed) / full.ll.null

# Chi-square distribution p-value

pchisq <- 1 - pchisq(2\*(full.ll.proposed - full.ll.null), df = (length(full.logistic.fit$coefficients)-1))

psuedo\_rsq <- data.frame(

psuedo\_rsq=full.ll.rsq,

chisq\_pval=pchisq)

print(psuedo\_rsq)

}

s\_plot <- function(glmmodel,bank\_data){

predicted.data <- data.frame(

probability.of.subscribe=glmmodel$fitted.values,

bank.term.deposit=bank\_data$y)

predicted.data <- predicted.data[

order(predicted.data$probability.of.subscribe, decreasing=FALSE),]

predicted.data$rank <- 1:nrow(predicted.data)

## Lastly, we can plot the predicted probabilities for each sample subscribing

## and color by whether or not they actually subscribed

ggplot(data=predicted.data, aes(x=rank, y=probability.of.subscribe)) +

geom\_point(aes(color=bank.term.deposit), alpha=1, shape=4, stroke=2) +

xlab("Index") +

ylab("Predicted probability of Subscribing")

}

get\_logistic\_pred = function(mod, data, res = "y", pos = 1, neg = 0, cut = 0.5) {

probs = predict(mod, newdata = data, type = "response")

ifelse(probs > cut, pos, neg)

}

```

# Introduction

The goal of this paper is to investigate banking data via an exploratory data analysis. Once we have initially examined the data, we will then move forward with attempting a classification model via logistic regression for predicting whether or not a client will subscribe to a banking institution given a direct marketing campaign taking place.

# Exploratory Data Analysis

The data we'll be exploring is from the [UCI Machine Learning Repository](https://archive.ics.uci.edu/ml/datasets/Bank+Marketing) relating to the direct marketing campaigns of a Portuguese banking institution. The initial data set consists of 41,188 observations with 20 inputs, dating between May 2008 to November 2010.

As described by the UCI Machine Learning Repository, each of the variables/columns are described as:

>\*\*Input variables:\*\*

>\*\*bank client data:\*\*

>

>1 - age (numeric)

>

>2 - job : type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')

>

>3 - marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)

>

>4 - education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown')

>

>5 - default: has credit in default? (categorical: 'no','yes','unknown')

>

>6 - housing: has housing loan? (categorical: 'no','yes','unknown')

>

>7 - loan: has personal loan? (categorical: 'no','yes','unknown')

>

>\*\*related with the last contact of the current campaign:\*\*

>

>8 - contact: contact communication type (categorical: 'cellular','telephone')

>

>9 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')

>

>10 - day\_of\_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')

>

>11 - 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.

>

>\*\*other attributes:\*\*

>

>12 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)

>

>13 - 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)

>

>14 - previous: number of contacts performed before this campaign and for this client (numeric)

>

>15 - poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')

>

># social and economic context attributes

>

>16 - emp.var.rate: employment variation rate - quarterly indicator (numeric)

>

>17 - cons.price.idx: consumer price index - monthly indicator (numeric)

>

>18 - cons.conf.idx: consumer confidence index - monthly indicator (numeric)

>

>19 - euribor3m: euribor 3 month rate - daily indicator (numeric)

>

>20 - nr.employed: number of employees - quarterly indicator (numeric)

>

>\*\*Output variable (desired target):\*\*

>

>21 - y - has the client subscribed a term deposit? (binary: 'yes','no')

Another caution we will need to access is that within this dataset, there is potential for more than one contact to the same client. This repeat contact was necessary as it was required to assess the product (i.e. a bank term deposit) and whether the client would or would not be subscribed.

```{r EDA}

# Retrieve datasets zip

if(!file.exists("./data/bank.zip")) {

download.file("https://archive.ics.uci.edu/ml/machine-learning-databases/00222/bank-additional.zip", "./bank-additional.zip", mode="wb")

}

unzip("./bank-additional.zip", files = c("bank-additional/bank-additional.csv","bank-additional/bank-additional-full.csv","bank-additional/bank-additional-names.txt"))

# Read data into a data frame object

full\_bank <- read.csv("./bank-additional/bank-additional-full.csv", header = T, sep = ";")

# Use function to create initial data insights for bank data

basic\_eda(full\_bank)

# Create a bird's eye view of missing data using naniar library if missing data exists

if(sum(!complete.cases(full\_bank)) > 0)

{

vis\_miss(full\_bank, cluster = F) + # Without aggregating observations

labs(title = "NAs in Bank Data from May 2008 - Nov 2010") +

theme(axis.text.x = element\_text(angle=90))

}

```

As we can see in the initial EDA, we have a small percentage of data that is marked unknown. As we cannot accurately categorize this data into their appropriate categories, we'll convert this data into NAs and drop them since they are at most 1.5% of the total data.

```{r unknown-transformation}

full\_bank[full\_bank=="unknown"] <- NA

freq(full\_bank, plot = F)

# Create a bird's eye view of missing data using naniar library if missing data exists

vis\_miss(full\_bank, cluster = F) +

labs(title = "NAs in Bank Data from May 2008 - Nov 2010") +

theme(axis.text.x = element\_text(angle=90))

gg\_miss\_upset(full\_bank)

```

Some interesting notes on the missing data is that the majority of the data seems to be the unknown status of whether an individual has defaulted on their credit or not. If default becomes a major predictor for whether a client will subscribe to a bank term deposit or not, some research should be explored here to discover why we cannot accurately define whether an individual defaults on their credit or not.

```{r remove-NAs}

complete\_full\_bank <- full\_bank[complete.cases(full\_bank),]

# Transform all chr objects in data frame to a factor class as a secondary data frame object

complete\_full\_bank <- as.data.frame(unclass(complete\_full\_bank), stringsAsFactors = T)

vis\_miss(complete\_full\_bank, cluster = F) + # Without aggregating observations b/c takes too long to aggregate

labs(title = "Cleaned Observations in Bank Data from May 2008 - Nov 2010") +

theme(axis.text.x = element\_text(angle=90))

```

Now we have no missing data.

## Addressing Character Variable and Numeric Variable

```{r}

# Checking for number of columns with numeric type

numeric\_var\_who=sum(sapply(complete\_full\_bank[,1:21],is.numeric))

numeric\_var\_who

# Checking for number of columns with character type

char\_var\_who=sum(sapply(complete\_full\_bank[,1:21],is.character))

char\_var\_who

# Checking for column names with numeric type

numeric\_varname\_who=which(sapply(complete\_full\_bank[,1:21],is.numeric))

numeric\_varname\_who

# Checking for column names with character type

char\_varname\_who=which(sapply(complete\_full\_bank[,1:21],is.character))

char\_varname\_who

```

### Is our Response Variables Unbalanced?

```{r is-data-balanced}

prop.table(table(complete\_full\_bank$y))

freq(complete\_full\_bank, input="y")

plot\_bar(data=complete\_full\_bank, by = "y", nrow=2, ncol=2)

```

Looking closer at our response variable, we have a significant lower percentage of yes's versus no's (12.66% yes's to 87.34% no's). With this in mind, we'll need to ensure either our sampling is weighted, or consider algorithms that can oversample or undersample when fitting our logistic regression model, otherwise we may introduce bias and lower our accuracy on predicting when a client successfully subscribes to a bank term deposit.

Additionally, we do not have any representation of those who have defaulted on their credit and if they choose not to subscribe to a bank term deposit. This variable could pose some issues when analyzing how defaults affect a client's desire to subscribe. Later we will look further into this and see if this predictor more closely when creating our models.

### Examining Normality of Numeric Variables

```{r normality-numeric}

plot\_num(complete\_full\_bank)

```

Looking closer at the histograms of the scale variables, it appears there are some potential normality issues, particularly the pdays.

## Initial Full Logistic Regression Model

Before we go into automatic model variable selection tools, we should attempt to understand the full model and how our explanatory variables may come into play and assess whether the assumptions of logistic regression are met or not.

```{r full-model}

full.logistic.fit <- glm(y ~ ., data = complete\_full\_bank, family="binomial")

summary(full.logistic.fit) # Full Model Statistics

psuedo\_rsq(full.logistic.fit) # Psuedo R Statistic w/ Chi-Sq P-value

library(ROCit)

full.measures <- measureit(score = full.logistic.fit$fitted.values, class = full.logistic.fit$y,

measure = c("ACC", "SENS", "FSCR"))

plot(full.measures$ACC~full.measures$Cutoff, type = "l")

roc\_empirical <- rocit(score = full.logistic.fit$fitted.values, class = full.logistic.fit$y, negref = "0")

plot(roc\_empirical, values=F)

```

### Interpreting the coefficients

The following variables are not useful predictors as they contain moderate to large p-values depending on the confidence level we seek to achieve:

\* age w/ p-value of 0.42

\* Jobs:

+ entrepreneur -- p-value of 0.22

+ maid -- p-value of 0.81

+ management -- p-value of 0.82

+ employed -- p-value of 0.52

+ technician -- p-value of 0.63

+ unemployed -- p-value of 0.82

\* Martial statuses of married (p-value of .96) and single (p-value of .75)

\* Education Levels:

+ basic of 6 years -- p-value of .42

+ basic of 9 years -- p-value of .82

+ high school -- p-value of .40

+ illiterate -- p-value of .06

+ professional course -- p-value of 0.24

\* Defaulting on Credit (p-values of .94)

\* Housing Loan (p-value of .64)

\* A loan in general (p-value of .36)

\* Last Month of Contact During Campaign:

+ July -- p-value of .66

+ October -- p-value of .14

+ December -- p-value of .31

\* Day of the Week of Contact During Campaign:

+ Monday -- p-value of .35

+ Thursday -- p-value of.08

\* The Number of Previous Contacts Performed Before the Campaign -- p-value of .41

Additionally, we can observe that for our full model, our residual deviance is 13,902 with an AIC of 13,998. We can also see that our psuedo R^2 value is 0.39975 with an extremely small Chi-squared p-value that it is nonetheless 0.

Lastly, we can see how the full model predicted by the following graphic:

```{r full-model-s-line}

s\_plot(full.logistic.fit, complete\_full\_bank)

```

As you can see, there looks to be some scattered mis-classifications within the yes section. This is perhaps due to unbalanced data that we need to sort out in our refined model with sampling methods as previously mentioned.

## Assumptions Investigation

Now that we understand more about our data and logistic regression model, we should do some preliminary assumptions check for logistic regression. These metrics are important as we will use them to compare our later simpler logistic regression model for performance evaluation.

### Binary/Ordinal Response Variable

```{r response-variable-assumption}

nlevels(complete\_full\_bank$y)

levels(complete\_full\_bank$y)

```

As you can see above, our response variable, y, consists of two levels, resulting in a binary response. We will move forward with a binary logistic regression model instead of an ordinal logistic regression model.

### Influential Observations

```{r residuals}

plot(full.logistic.fit, which = 4, id.n = 3)

```

As we can see from our Cook's Distance visualization, it appears we may have 3 observations of influence between 0.08 and 0.012. However, despite these are small values, we will elect to remove them as an extra precaution.

```{r remove-influential}

complete\_full\_bank\_2 <- complete\_full\_bank[-c(18463,27544,27535)]

```

### Linearity of Continuous/Scale Variables vs. Log Odds of Response Variable

```{r linearity-assumption}

numeric\_pred <- complete\_full\_bank %>% select\_if(is.numeric)

numeric\_pred\_names <- colnames(numeric\_pred)

# binding logit and numeric predictors for scatterplots

linearity\_data <- numeric\_pred %>%

mutate(logit = log(full.logistic.fit$fitted.values/(1-full.logistic.fit$fitted.values))) %>%

gather(key = "predictors", value = "predictor.value", -logit)

ggplot(linearity\_data, aes(logit, predictor.value))+

geom\_point(size = 0.5, alpha = 0.5) +

geom\_smooth(method = "loess", formula = "y~x") +

theme\_bw() +

facet\_wrap(~predictors, scales = "free\_y")

```

We can observe from our plot that a number of our variables do not follow a linear trend. However, we do see that duration and potentially, campaign, have a linear trend associated with subscribing to a bank term deposit.

For all other variables, transformations may be beneficial for classifying, but for now we will leave them as is.

### Multicollinearity of Explanatory Variables

The below correlation table shows the correlation between the numerical variables:

\* nr.employed and emp.var.rate are 91% correlated.

\* nr.employed and euribor3m are 95% correlated.

\* emp.var.rate and euribor3m are 97% correlated.

\* cons.price.idx and emp.var.rate are 78% correlated.

\* cons.price.idx and euribor3m are 69% correlated.

\* cons.price.idx and nr.employed are 52% correlated.

Furthermore, we do have a moderate multicollinearity issue with some variables for previous and pdays. We may want to examine them closer.

```{r Correlation}

ggcorr(data=numeric\_pred, label = T, nbreaks=5, label\_size = 3, hjust = 0.9, size = 3, layout.exp = 4) +

labs(title = "Multicollinearity of Variables (Pairwise / Pearson's correlation)")

```

For variables with moderate to large Pearson's correlation, we'll need to examine the VIF to see which variables may be need to be removed due to a problematic amount of collinearity.

```{r VIF}

vif(full.logistic.fit)

```

As a rule of thumb, we should pay attention to variables with a VIF higher than 5 or 10, but we'll use 5 for logistic regression. The following variables are showing significant VIF values:

\* job

\* month

\* pdays

\* poutcome

\* emp.var.rate

\* cons.price.idx

\* cons.conf.idx

\* euribor3m

\* nr.employed

With our earlier analysis on the coefficients, we can see why some predictors were not as significant or useful (e.g. jobs, month, etc.). To refine our model further, we'll remove the above variables.

## PCA Examination

```{r PCA}

numeric\_preds\_bank <- complete\_full\_bank %>% keep(is.numeric)

my.cor <- cor(numeric\_preds\_bank)

library(gplots)

library(ggplot2)

heatmap.2(my.cor,

col=redgreen(75),

density.info="none",

trace="none",

dendrogram=c("row"),

symm=F,

symkey=T,

symbreaks=T,

scale="none",

srtCol = 35)

#Another option here would be to do PCA among the continous predictors to see

#if they seperate out. Or a heatmap.

pc.result<-prcomp(numeric\_preds\_bank %>% keep(is.numeric),scale.=TRUE)

pc.scores<-pc.result$x

pc.scores<-data.frame(pc.scores)

pc.scores$y<-complete\_full\_bank$y

#Use ggplot2 to plot the first few pc's

ggplot(data = pc.scores, aes(x = PC1, y = PC2)) +

geom\_point(aes(col=y), size=1)+

ggtitle("PCA of y")

ggplot(data = pc.scores, aes(x = PC2, y = PC3)) +

geom\_point(aes(col=y), size=1)+

ggtitle("PCA of y")

ggplot(data = pc.scores, aes(x = PC3, y = PC4)) +

geom\_point(aes(col=y), size=1)+

ggtitle("PCA of y")

```

There does not seem to be a large separation with PCA.

## Second Logistic Regression Model

```{r lr-model-2}

logistic.fit.2 <- glm(y ~ . -nr.employed -euribor3m -cons.conf.idx -cons.price.idx -emp.var.rate -poutcome -pdays -month -job, data = complete\_full\_bank, family="binomial")

summary(logistic.fit.2)

psuedo\_rsq(logistic.fit.2)

s\_plot(logistic.fit.2, complete\_full\_bank\_2)

```

After adjusting the model for multicollinearity problems, we can now see that our model has worsened in performance. However, it does seem that our s-plot is starting to form when compared with the initial full model.

## Model Selection

As we can see, manually modifying and tweaking a model can have adverse results. With this we'll introduced a third and final model to do a comparison with using automatic variable selection method, LASSO, for this, we'll need to create a sampling of test data to train the model and then a test set to predict with. We'll use a 80:20 split and balance our data.

```{r data-split}

set.seed(2008)

training\_samples <- complete\_full\_bank\_2$y %>% createDataPartition(p=0.8, list = F)

train.data <- complete\_full\_bank[training\_samples,]

test.data <- complete\_full\_bank[-training\_samples,]

# Create matrix of predictors & convert to categorical predictors to appropriate dummy values

## Dummy code categorical predictor variables

x <- model.matrix(y~., train.data)[,-1]

## Convert outcome/class to numerical variable

y <- ifelse(train.data$y == "no", 1, 0)

# Source on stepping through Penalized Logistic Regression: http://www.sthda.com/english/articles/36-classification-methods-essentials/149-penalized-logistic-regression-essentials-in-r-ridge-lasso-and-elastic-net/#compute-lasso-regression

```

### LASSO

```{r LASSO Lambda}

# Set seed and find ideal lambda for LASSO

set.seed(2008)

cv.l.model <- cv.glmnet(x, y, family = "binomial", alpha = 1) # Remeber alpha=1 means LASSO Regression

# Plot of ideal lambda for minimizing CV error

plot(cv.l.model)

# Comparing Regression Coefficients from CV of lambda

coefplot(cv.l.model, lambda=cv.l.model$lambda.min, family="binomial")

coef(cv.l.model, cv.l.model$lambda.min)

coef(cv.l.model, cv.l.model$lambda.1se)

coefplot(cv.l.model, lambda=cv.l.model$lambda.1se, family="binomial")

# Graphic to interact with lambda and coeficients

coefpath(cv.l.model)

```

As we do not have an analyst to help specify the lambda value to use in our LASSO model to control the coefficient shrinkage, we elected to use cross-validation error to find a suitable lambda for our data. As can be observed, when examining the cross-validation error according to the log of lambda, our left dashed vertical line indicates the optimal value of -6.

When choosing which lambda value to use when fitting our model, we generally want a balance between accuracy but also simplicity. That way, we can easily interpret the model if need be. Looking closer, we can examine from the coefficient tables, which lambda will provide a simple model. In this case, the within 1 standard error (1se) lambda has 25 variables that have non-zero coefficients, while our minimum lambda has 7 non-zero coefficients. For our initial model, we will use within 1 standard error lambda to produce a simpler model for understanding rather than for accuracy.

```{R LASSO-model}

set.seed(2008)

# Fit a model w/ ideal lambda from cross-validation

l.model <- glmnet(x, y, alpha = 1, family="binomial", lambda = cv.l.model$lambda.1se)

# Predict on test data

x.test <- model.matrix(y~., test.data)[,-1]

probabilities <- l.model %>% predict(newx = x.test, type="response")

predicted.classes <- ifelse(probabilities > 0.5, "no", "yes")

# Accuracy Rate

observed.classes <- test.data$y

mean(predicted.classes == observed.classes)

assess.glmnet(l.model, newx = x.test, newy = test.data$y)

confusion.glmnet(l.model, newx = x.test, newy = test.data$y)

lasso.roc <- roc.glmnet(l.model, newx = x.test, newy = test.data$y)

plot(lasso.roc)

```

|  |
| --- |
| library(tidyverse) |
|  |

|  |
| --- |
| library(MASS) |
|  |

|  |
| --- |
| library(caret) |
|  |

|  |
| --- |
| library(naniar) |
|  |

|  |
| --- |
| library(GGally) |
|  |

|  |
| --- |
| library(funModeling) |
|  |

|  |
| --- |
| library(Hmisc) |
|  |

|  |
| --- |
| library(ROCC) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #find numeric columns |
|  |

|  |
| --- |
| str(full\_bank\_2) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #create data frame with only numeric columns and response variable |
|  |

|  |
| --- |
| bank.num <- full\_bank\_2[,c(1,12:14,16:21)] |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #remove 999 from pdays column as this is an outlier representing customers not contacted in previous campaign |
|  |

|  |
| --- |
| bank.num <- bank.num %>% filter(pdays != 999) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #plot correlation between predictors |
|  |

|  |
| --- |
| ggpairs(bank.num, columns = c(1:9), aes(colour = y)) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #create training test split |
|  |

|  |
| --- |
| set.seed(123) |
|  |

|  |
| --- |
| training.samples <- bank.num$y %>% createDataPartition(p = 0.8, list = FALSE) |
|  |

|  |
| --- |
| train.data <- bank.num[training.samples,] |
|  |

|  |
| --- |
| test.data <- bank.num[-training.samples,] |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #center and scale data, produced nearly identical results so probably not needed |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #preproc.param <- train.data %>% preProcess(method= c("center", "scale")) |
|  |

|  |
| --- |
| #train.transformed <-preproc.param %>% predict(train.data) |
|  |

|  |
| --- |
| #test.transformed <-preproc.param %>% predict(test.data) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #create lda model |
|  |

|  |
| --- |
| mylda <- lda(y ~ ., data = train.data) |
|  |

|  |
| --- |
| mylda |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #make predictions and find accuracy |
|  |

|  |
| --- |
| predictions <- mylda %>% predict(test.data) |
|  |

|  |
| --- |
| mean(predictions$class==test.data$y) |
|  |

|  |
| --- |
| CM = confusionMatrix(table(predictions$class,test.data$y)) |
|  |

|  |
| --- |
| CM |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #plot ld1 |
|  |

|  |
| --- |
| plot(mylda) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #overlay ld1 plot for each category |
|  |

|  |
| --- |
| lda.data <- cbind(train.data, predict(mylda)$x) |
|  |

|  |
| --- |
| ggplot(lda.data, aes(x=LD1)) + geom\_histogram(aes(fill=y)) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #From https://towardsdatascience.com/linear-discriminant-analysis-lda-101-using-r-6a97217a55a6 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #grab the posterior from the predictions |
|  |

|  |
| --- |
| predictions.posteriors <- as.data.frame(predictions$posterior) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #create values for and plot ROC |
|  |

|  |
| --- |
| pred <- prediction(predictions.posteriors[,2], test.data$y) |
|  |

|  |
| --- |
| roc.perf = performance(pred, measure = "tpr", x.measure = "fpr") |
|  |

|  |
| --- |
| auc.train <- performance(pred, measure = "auc") |
|  |

|  |
| --- |
| auc.train <- auc.train@y.values |
|  |

|  |
| --- |
| # Plot |
|  |

|  |
| --- |
| plot(roc.perf) |
|  |

|  |
| --- |
| abline(a=0, b= 1) |
|  |

text(x = .25, y = .65 ,paste("AUC = ", round(auc.train[[1]],3), sep = ""))