|  |
| --- |
| If real wages aren't rising, how is household income going up? |
| DS6372 Project 2 Logistic Regression Adult Income  April 10, 2022 |
| |  | | --- | | Eric Laigaie, Rayon Morris & Douglas Yip | |

**1. Introduction**

This project will focus on logistic regression where we will be analyzing the response (pay\_reponse) that indicates if an individual is making either greater or less than $50,000.

The following report will contain a detailed analysis and conclusions of the following:

* Initial (Exploratory Data Analysis) EDA
* Building a Logistic Regression Model to predict the binary pay\_response
* Comparing and compiling different regression models, where at least one contains complex variables and at least one that is non-parametric.
* Conclusion and determination of our best model that can predict a binary outcome of in an individual make greater or less than 50,000.

**2. Data description**

For this project, we downloaded the training and testing sets from an online census data source. Our data exploration will mainly take place in the training set. Any transformations made to the training set will also be performed on the training set to keep them consistent.

The training data set contains 32,561 records with 16 different attributes (Table 2.1). Further changes of the data set will be addressed in our exploratory data analysis. Below is a summary of the original file.

***Table 2.1. R output of the car data set that contains the 16 different variables.***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Variable Name | Data Type | Levels | Min | Max | Mean | Median |
| Id | Int | 32,561 | 1 | 32,561 | 16,281 | 16,281 |
| Age | Int | 73 | 17 | 90 | 38.58 | 37 |
| Workclass | Factor | 9 | -- | -- | -- | -- |
| fnlwgt | Int | 21648 |  |  |  |  |
| Education | Factor | 16 | -- | -- | -- | -- |
| Education\_num | Factor | 16 | -- | -- | -- | -- |
| Martial\_status | Factor | 7 | -- | -- | -- | -- |
| Occupation | Factor | 15 | -- | -- | -- | -- |
| Relationship | Factor | 6 | -- | -- | -- | -- |
| Race | Factor | 5 | -- | -- | -- | -- |
| Sex | Factor | 72 | -- | -- | -- | -- |
| Capital\_gain | Int | 119 | 0 | 99999 | 1082 | 0 |
| Capital\_loss | Int | 92 | 0 | 3770 | 87.9 | 0 |
| Hours\_per\_week | Int | 94 | 1 | 99 | 40.39 | 40 |
| Native\_country | Factor | 42 | -- | -- | -- | -- |
| Pay\_Response | Factor | 2 | -- | -- | -- | -- |

Source: Appendix 2.1 and 2.2**3. Exploratory Data Analysis**

**Missing Values (Appendix 3.1)**

Identifying missing values was necessary to obtain accurate summary statistics. We first evaluated whether there were any NA variables or blanks within both data sets. Based on our results, there were no NA variables. However, based on the summary output, we noticed that workclass, occupation and native\_country contained “?” values. After checking for “?” counts in each of these variables, we found that these ‘missing’ values were contained to less than 2000 records (Appendix 3.1). Given that we have a training dataset of 32,561 and that there was no logical method to impute the data, in our analysis, *we removed all ‘?’ rows from both the training and test data to complete this study.*

**Unbalanced datasets (Appendix 3.3)**

As a result of this study being a logistic regression, we checked the response value to see if we had a balanced dataset for both training/test data sets. Based on our count (Appendix 3.3), we identified approximately 25% of the results showing a pay\_response greater than $50,000. This would suggest that we are dealing with an unbalanced data set. As such, *we will identify the optimal cut off to maximize accuracy in our logistic regression model and prediction*.

It is also important to note that there is no difference in severity between a false positive and false negative in this case. Therefore, overall accuracy may provide a better picture of model performance than specificity or sensitivity alone.

**Continuous variables collinearity check (Appendix 3.4)**

The correlation grid (Appendix 3.4) for the continuous variables provides no evidence that any of the variables are correlated. *No action was taken to the continuous variables as a result of the correlation grid*.

**Effects of continuous variables on pay response (Appendix 3.5)**

The cluster heat map (Appendix 3.5) of the response variable was evaluated and we see an effect of capital gain and capital loss to the response variables. All other variables were difficult to determine to see if there was any separation of the response. *No action was taken to the continuous variable as a result of the heat map.*

**Categorical Variable Exploration**

Using a bar chart of pay\_response proportions by factor level, we examined each categorical variable to determine if level consolidation could take place. Below you will find a short breakdown of each variable we examined along with the actions we took after observation.

***Education (Appendix 3.6-3.7)***

This variable initially had 16 levels, but the bar chart in appendix 3.6 indicates that many of these education levels display similar pay\_response proportions. Therefore, *we grouped education based on the groups below, reducing this variable to seven levels.*

* Preschool
* Grade School (grade 1-12 of original data)
* HS Grads (HS Grads + Some college of original data)
* Assocs (Assoc-voc + Assoc-acdm of original data)
* Bachelors
* Masters
* Docs/Profs (Prof-school + Doctorate of original data)

***Workclass (Appendix 3.8-3.9)***

The pay\_response for the government classified workers was similar in both data sets and *we grouped government (Local, State and Federal) workclass to reduce the levels from 7 to 5 level*s.

***Occupation (Appendix 3.10)***

A bar chart of pay\_response was performed on the occupation variable as we identified 15 levels in the data. While we could understand grouping education because the difference between 10th and 11th grade is negligible, occupations are much more different. Therefore, *we left the variable* as is.

***Marital\_Status (Appendix 3.11-3.12)***

After observing the initial distributions, we noticed the pay\_response proportions for the married and formerly married responses were similar in both data sets. Therefore, *we grouped and created “married” and “single was married” marital statuses to reduce the levels from 7 to 4.*

***Native\_Country (Appendix 3.13-3.14)***

This variable had many levels that did not seem to follow an exact pattern. In an effort to find some sort of logical grouping, we *grouped countries into the continents and reduced the levels from 42 to 7 levels.* One important distinction is the ‘South’ level. Since this country could be South Korea or South Africa, we decided to leave it as its own level.

**Redundancy in education and education \_num (Appendix 3.15)**

A box plot graph was made between the variable education and education\_num. As expected, we observed that these metrics were perfect matches and were therefore redundant. (Appendix 3.15) *Education is best viewed through factors (there isn't a numerical relationship in education levels), so we kept education in our final dataset.*

**Capital Gain and Loss**

After observing the variables capital\_gain and capital\_loss, we noticed that one value was always zero. Therefore, we decided to *consolidate these using the formula capital\_gain – capital\_loss.* This resulted in a new column, capital\_net.

**Mosaic plots to check multicollinearity for categorical variable (Appendix 3.16-3.17)**

Mosaic plots were made to check for multicollinearity for categorical variables. Four of the graphs (appendix 3.16) workclass & education; marital status & education; race & education; and race & marital status exhibit little to no evidence of correlation. However, relationship & marital status (appendix 3.17) showed strong evidence of correlation. As a result, *the relationship column was removed in favor of marital status*.

**Removal of categorical ID and fnlgwt variables**

Two variables were removed from dataset prior to any modeling, as the variables provided did not support the prediction of income based on our research and knowledge. ‘ID’ is an arbitrary identifier column and ‘fnlwgt’ is used by the US Census to signify how many people are represented by that record. *We removed the two variables from the data*.

**4. Objective 1**

Build a logistic regression model to complete the analysis: 1) hypothesis test to see if we have any significant variables that could predict an individual having income greater or less than $50,000; 2) Determine the model and variables used for the analysis, which include the interpretation of each variable and confidence intervals for each parameter of the model.

**Model Selection Methodology**

Logistic regression was selected for our model given that the response is a binary variable of <$50K and =>$50K. After our EDA, a total of 9 variables were used in stepwise logistic regression model. The following explanatory variables exists in the model; age, workclass, education, marital\_status, occupation, race, sex, hours\_per\_week, native\_country, and capital\_net. Additionally, a check for influential points was completed with the cooksd() function. We found that no points qualified as an influential outlier, so our full dataset was fine to put into the model.

The resulting model provided a large number of predictors with varying levels of statistical significance. With more than 20 predictors, interpretability was in danger. A majority of the least-significant predictors came from the occupation variable, so we decided to run the stepwise process again without it. After this new model was created, we noticed that many insignificant predictors were weeded out without sacrificing a practically significant amount of accuracy (Table 4.1).

**Table 4.1 Interpretation of Parameters for Final Simple Model**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **# of betas at P=<0.01** | **Accuracy** | **Specificity** | **Sensitivity** |
| ***All predictor*** | ***20*** | ***83.9%*** | ***92.5%*** | ***57.6%*** |
| ***Occupation Removed*** | ***9*** | ***83.1%*** | ***92.9%*** | ***53.2%*** |

*Source: Appendix (4.1 and 4.2)*

**Test for Fit**

Given the training sample for our train is greater than +30,000 observation, we did not perform the Hosmer-Lemeshow tests to determine goodness of fit as it is not robust for large datasets. The combination of the training accuracy and the cross validation with the test set and the plot for the ROC curve (Appendix 4.3), we determine that our model is a good fit. In addition, a sensitivity of the cutoff was reviewed (Appendix 4.4) and we determine that the default 0.5 was a suitable cutoff to use to maximize our model’s accuracy.

**Final Model**

|  |
| --- |
| **Final Simple Model** |
|  |

**Table 4.1 Interpretation of Parameters for Final Simple Model**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Odds Ratio** | | |  |
| **Variable (P<0.01)** | **Parameter** | **Value** | **Odds** | **CI 2.5%** | **CI**  **97.5%** | **Interpretation with CI** |
| Intercept |  | -24.43 | 0.00 | 0.00 | 0.00 | Odds of a person pay >50k are even when all is zero and after accounting for all categorical variables. |
| Age |  | 0.0268 | 1.027 | 1.024 | 1.03 | The odds of a person having >50k is 1.027 times higher than a person 1 year younger holding all other variables fixed. 95% CI (1.024,1.031). |
| WorkClass Self-emp-inc |  | 0.3395 | 1.404 | 1.175 | 1.679 | The odds of Self-employed with income workers having >50k is 1.404 times more than government worker holding all other variables fixed. 95% CI (1.17,1.68) |
| WorkClass Self-emp-not-inc |  | -0.621 | 0.537 | 0.467 | 0.618 | The odds of Self-employed with no income workers having >50k is 1.89 (1/0.5237) times less than a government worker holding all other variables fixed. 95% CI (1.61,2.14) |
| Marital Status Married |  | 2.136 | 8.461 | 7.533 | 9.505 | The odds of Married individual having >50k is 8.46 times more than a previously married individual holding all other variables fixed. 95% CI (7.53,9.505) |
| Marital Status  Never Married |  | -0.531 | 0.587 | 0.506 | 0.682 | The odds of Never married individual having >50k is 1.7 times less than a previously married individual holding all other variables fixed. 95% CI (1.97,1.47) |
| Race White |  | 0.0642 | 1.900 | 1.240 | 2.912 | The odds of White individual having >50k is 2.0 times more than Indian American individual holding all other variables fixed. 95% CI (1.24,2.912) |
| Sex Male |  | 0.0135 | 1.144 | 1.039 | 1.261 | The odds of males having >50k is 1.1 times more than females holding all other variables fixed. 95% CI (1.039,1.261) |
| Hours per week |  | 0.03 | 1.030 | 1.027 | 1.033 | The odds of a person having >50k is 1.030 times higher than a person working an additional hour holding all other variables fixed. (95% CI (1.027,1.033) |
| Capital Net |  | 0.0003 | 1.000 | 1.000 | 1.000 | The odds of a person having >50k is even to a person making $1 younger holding all other variables fixed. (95% CI (1.000,1.0000)) |

**Conclusion**

Unfortunately, this model, while interpretable, does not provide a high level of accuracy (90%+). Although, some important relationships can be pulled out of this model. We found that white males are much more likely to classify as >50K as opposed to other racial and gender demographics. Additionally, marriage plays a large part in this classification model. Therefore, a new question is raised: in this data, is the pay\_response variable representing the income of one person, or their entire household?

**5. Objective 2**

**Problem Statement**

In objective 1, our goal was to create a simple, highly interpretable model. For this objective, we will be using that model as a baseline and use various methods in an attempt to improve model performance. First, we well keep the variables produced in the stepwise model but add complexity through interaction terms. Then, we will use LDA and QDA to see if performs better that our 2 models. We have also included a Random Forest model, a non-parametric test to complete this objective. Lastly, these models will be contrasted with the stepwise model to compare performance and determine the most optimal solution.

**Complex Model**

Our method for creating this model was to run through many prospective models with varying terms and viewing how these interactions turned out in the model summary. If we found certain pairings of variables were often insignificant, we moved away from them and explored new interactions. To better understand any roadblocks for this model, we used the cooksd() function to provide us with any outlying and highly influential points. Similar to objective 1, we did not find any points returned here.

After reaching a final model, we investigated every cutoff value between 10 and 90 percent with an interval of .1%. Our code found that a classification cutoff of 48% provided the maximum accuracy of 82.71%. In Appendices 5.1-5.4, you can find this model’s summary, confusion matrix, accuracy vs. cutoff chart, and roc curve.

**LDA/QDA**

To give our LDA a chance to compete against the other models, we will start with the original dataset to see how well LDA performs. The assumptions for LDA are Normality, and equal covariance matrices. If the equal covariance is not valid, we will need to utilize QDA. For our initial LDA model, we see that it fails the normality and the equal covariance assumptions. Before proceeding with QDA, we log transformed the explanatory variables of the LDA model. Here we saw better separation of the data. However, both assumptions were still violated. For that reason, we ran a QDA model (See Appendix 5.5 -5.12). Our conclusion is that the LDA, though it failed its assumptions, gave a better accuracy that the QDA model. This is because the LDA model is robust to the equal covariance assumption.

**Random Forest**

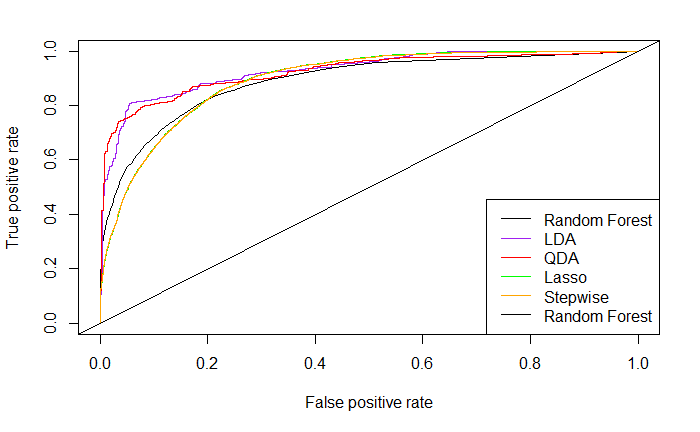
The random forest is a classification algorithm consisting of many decisions trees. The model builds decision trees on different samples and takes their majority vote for classification and average in case of regression. A MTRY model was run to identify the optimal MTRY to use for the Random Forest Model. We identified 9, from our training model, to optimize accuracy for the prediction model (Appendix 5.13). Utilizing Random Forest model and the optimal mtry of 9, the achieved prediction accuracy for the test set was 84.06%. (Appendix 5.14)

**Model Results**

Table 5.1 Summary of statics for the models predicting test and validation data.

|  |  |  |  |
| --- | --- | --- | --- |
| Predictive Models Test Statistics | Accuracy | Sensitivity | Specificity |
| Objective 1 Model | 83.12% | 92.88% | 53.16% |
| Complex Model | 82.71% | 90.87% | 57.65% |
| LDA | 83.87% | 74.15% | 89.74% |
| QDA | 81.87% | 89.87% | 68.64% |
| Random Forest | 84.06% | 92.29% | 62.67% |

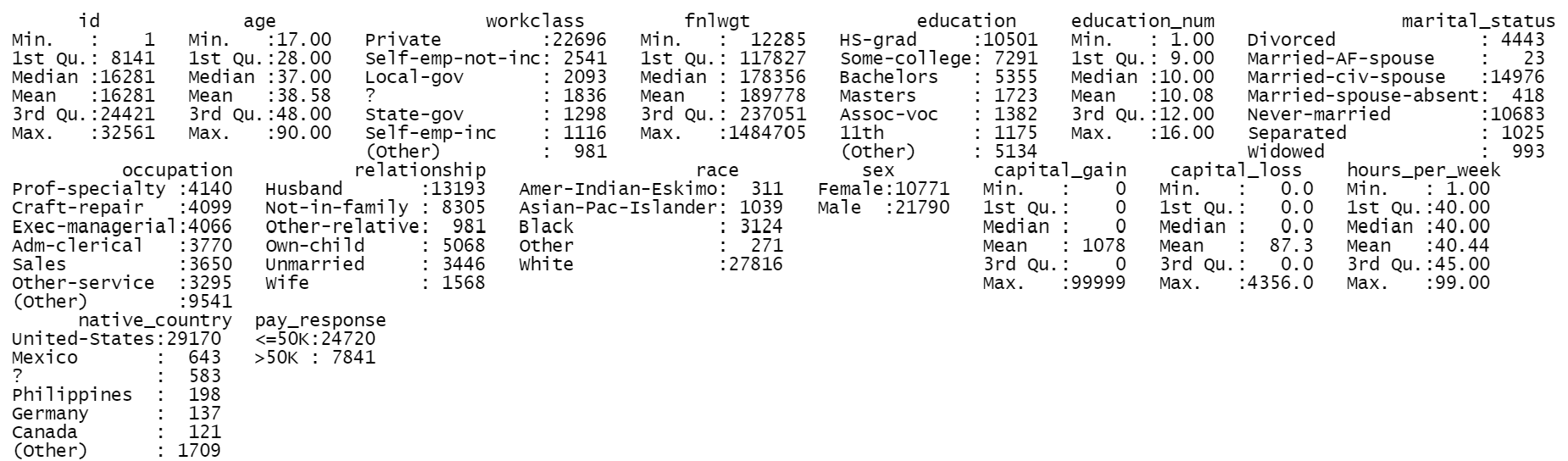
**6. Final Summary**

Through both objectives, we built a menu of models to be evaluated both alone and together. Although we attempted to build high-accuracy models, our results led us to conclude that this dataset leaves a lot to be desired (Figure 5.1). With only three numeric features, the regression models we built have struggled to achieve a strong accuracy. Our marginally best model, Random Forest model, achieved an 84.06% test accuracy. However, this research and model development was beneficial because it can be used to point future modelers towards success. A few suggestions for next steps include deeper exploration into related datasets that could bolster the census data and provide more continuous variables to use as predictors. Additionally, more complicated models such as k-nearest neighbors and neural networks could be used in an effort to drive up complexity and, hopefully, accuracy.

**Figure 5.1 Summary of ROC plots**

**7. Appendix**

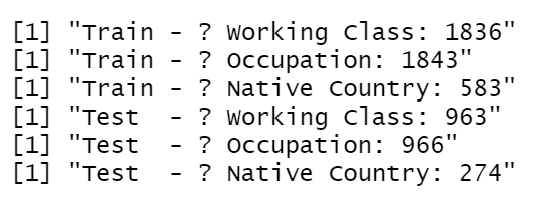
**Appendix 2.1 – Initial Summary Statistics**

****

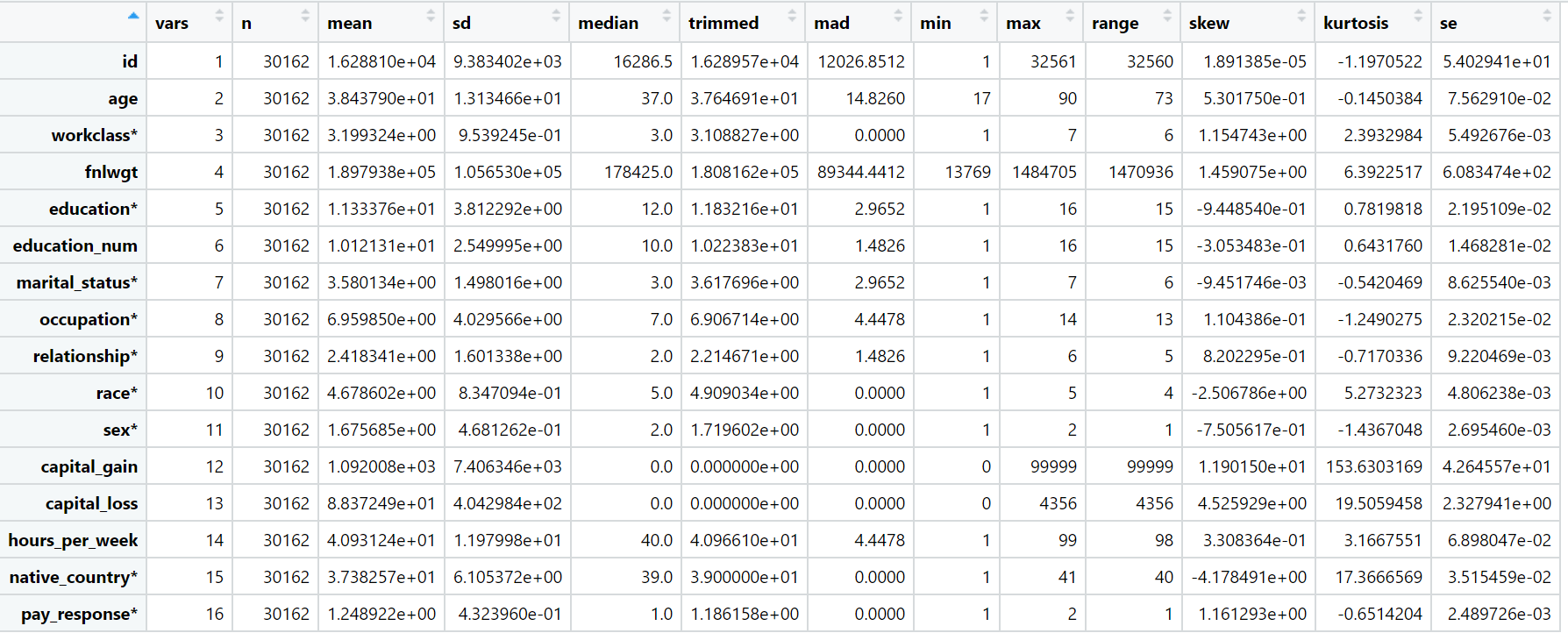
**Appendix 2.2 – Number of levels by Variable**

****

**Appendix 3.1 R output of number of unknown variables coded “?”**

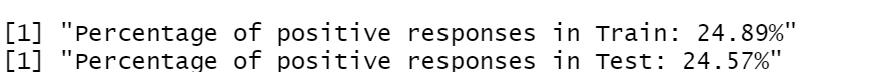
****

**Appendix 3.2 Summary of data set after removal of “?”**

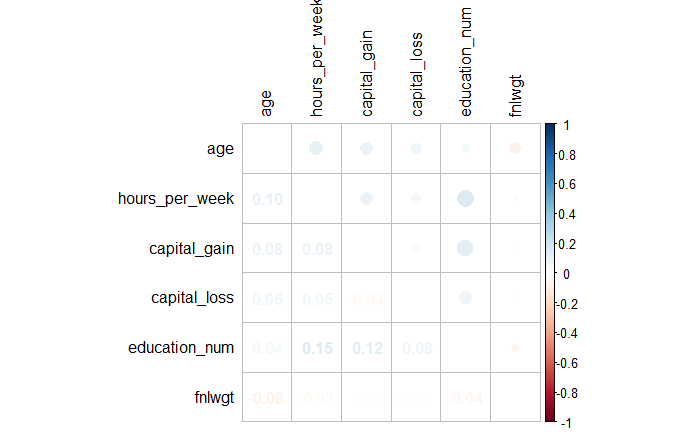
****

**Appendix 3.3 R output of Unbalanced response for both the train and test data sets**

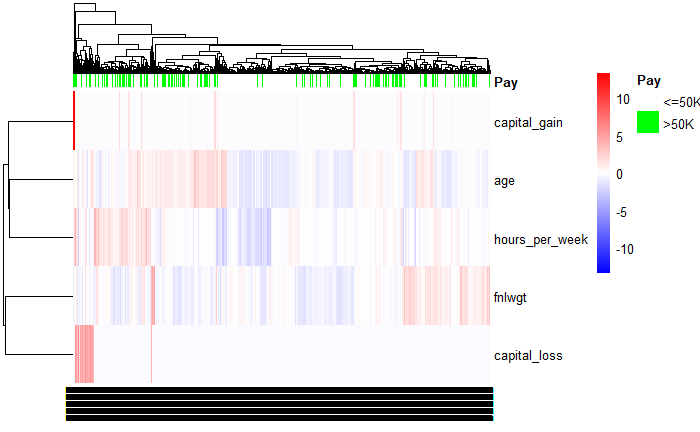
****



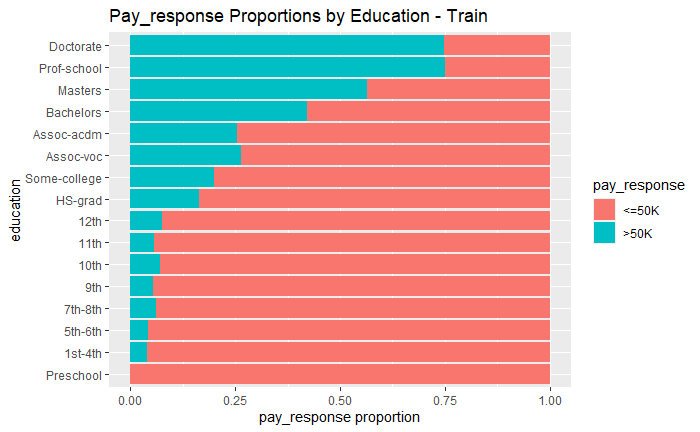
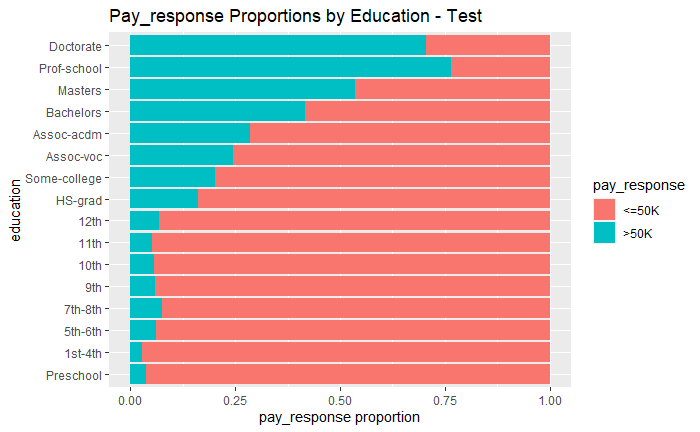
**Appendix 3.4 Correlation plot of continous variables.**



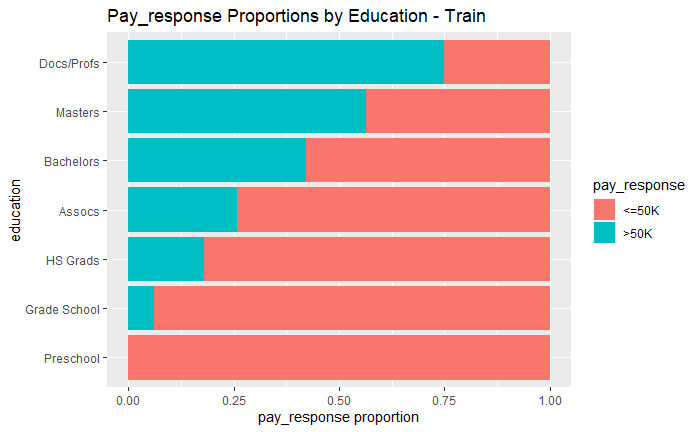
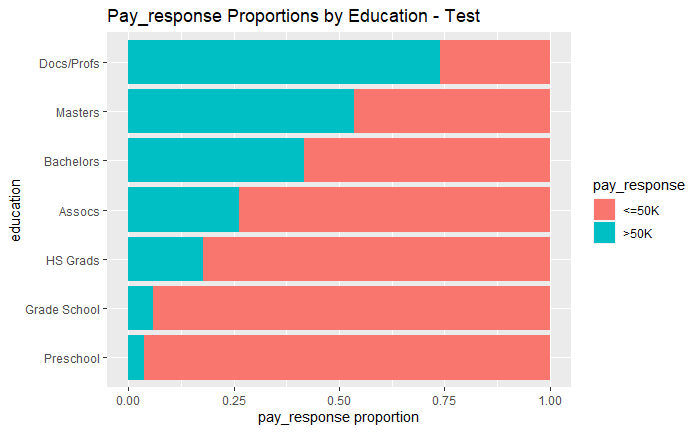
**Appendix 3.5 Cluster Heat map to determine the impact to pay response**



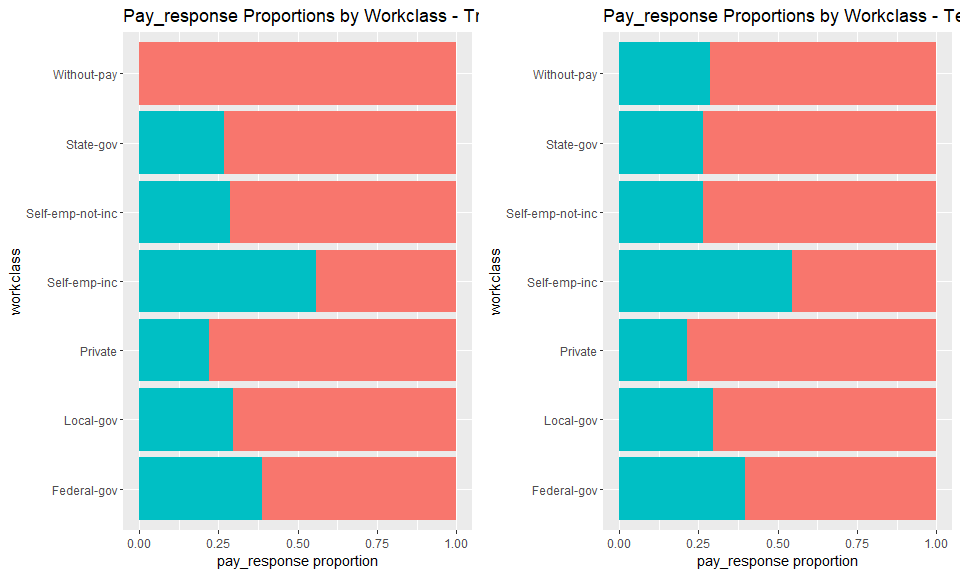
**Appendix 3.6 Pay response for Education pre transformation (Training and Test Sets)**

**Appendix 3.7 Pay response for Education post transformation (Training and Test Sets)**

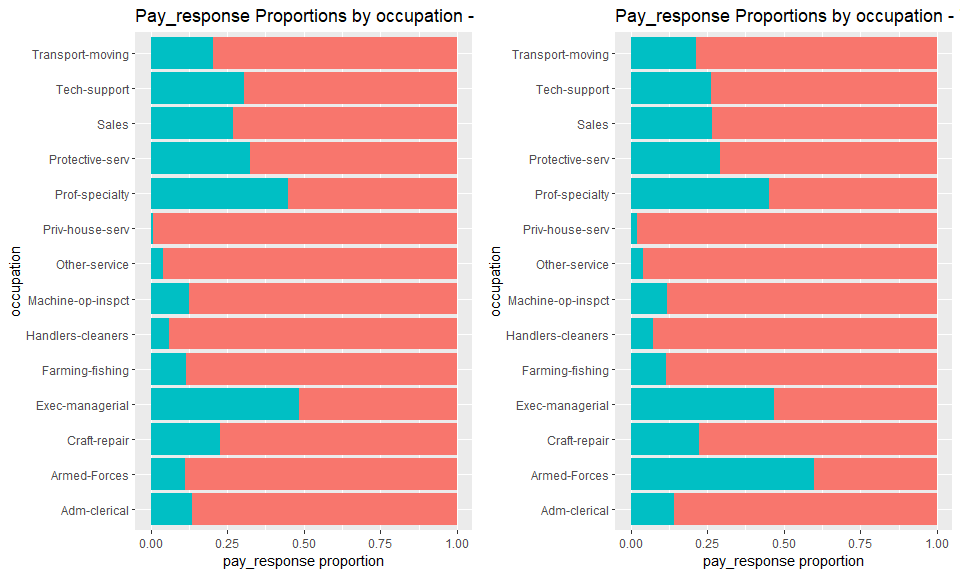
**Appendix 3.8 Pay response for Workclass pre transformation (Training and Test Sets)**



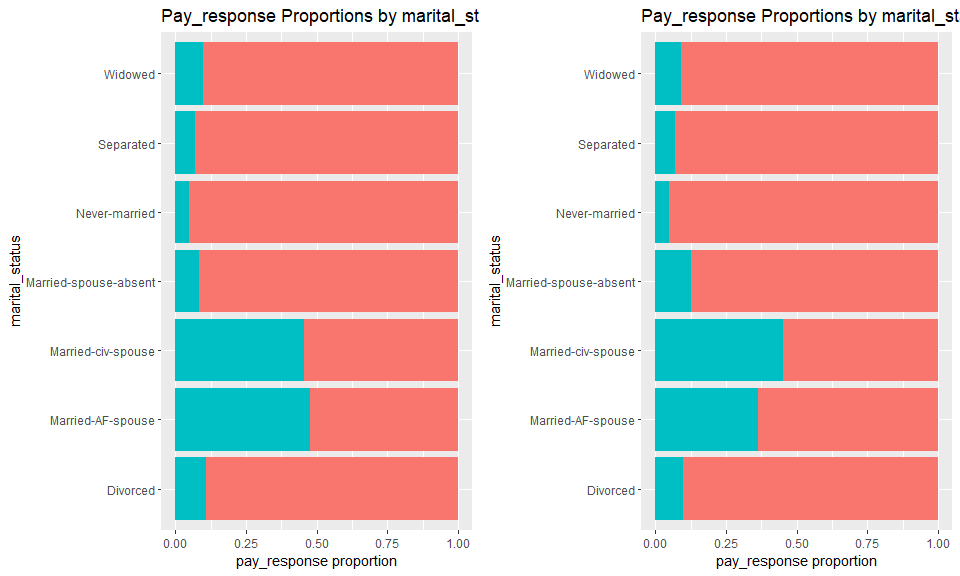
**Appendix 3.9 Pay response for Workclass post transformation (Training and Test Sets)**



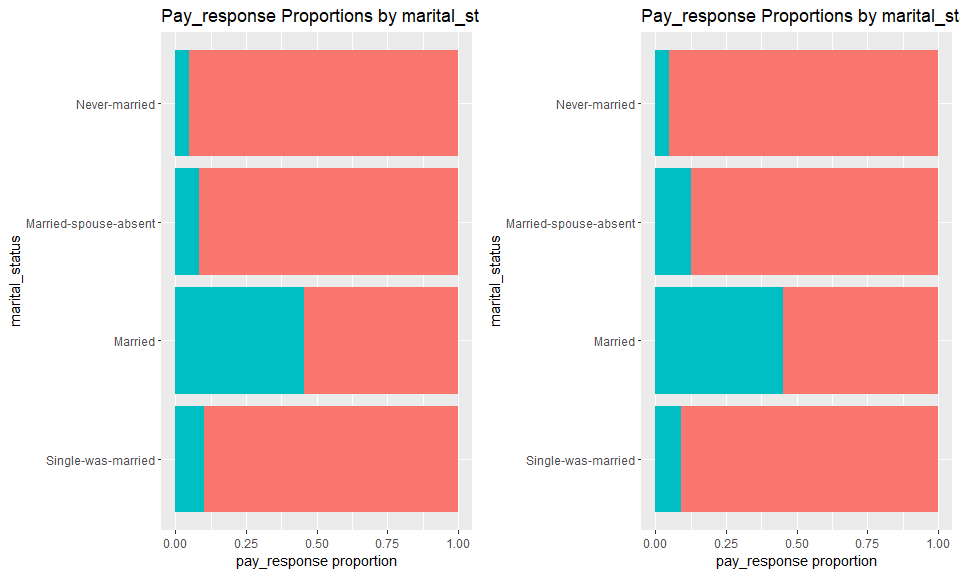
**Appendix 3.10 Pay response for Occupation**



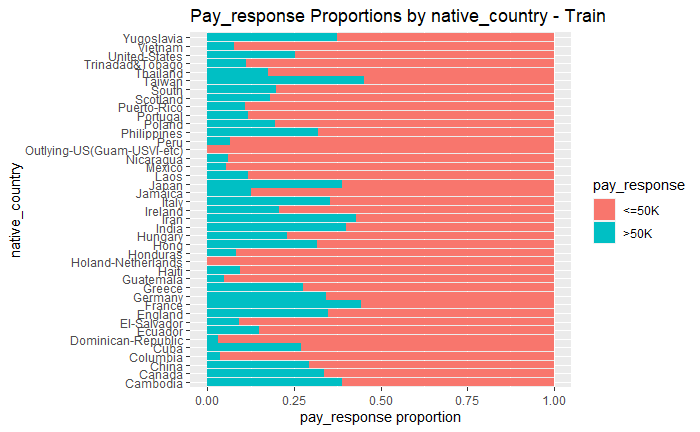
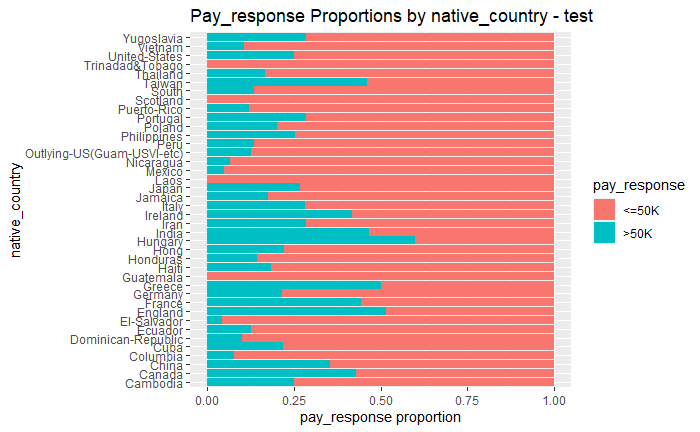
**Appendix 3.11 Pay response for Marital status pre transformation (Training and Test Sets)**



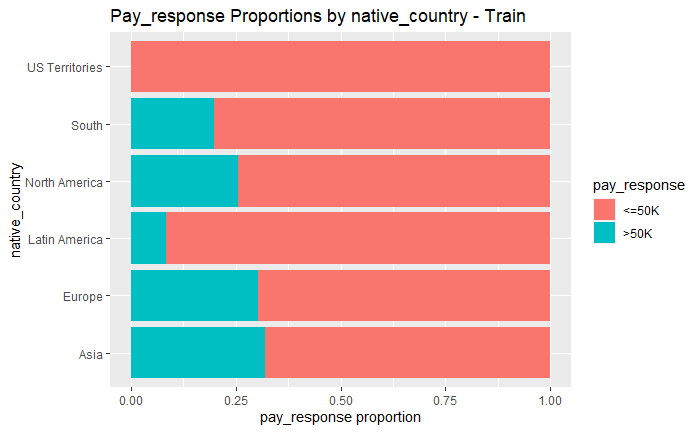
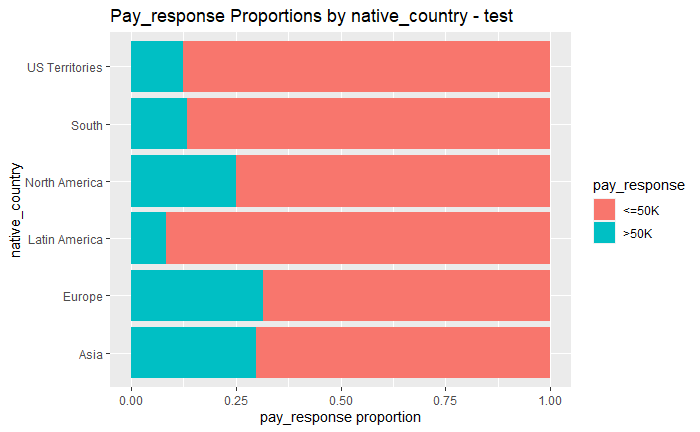
**Appendix 3.12 Pay response for Marital status post transformation (Training and Test Sets)**



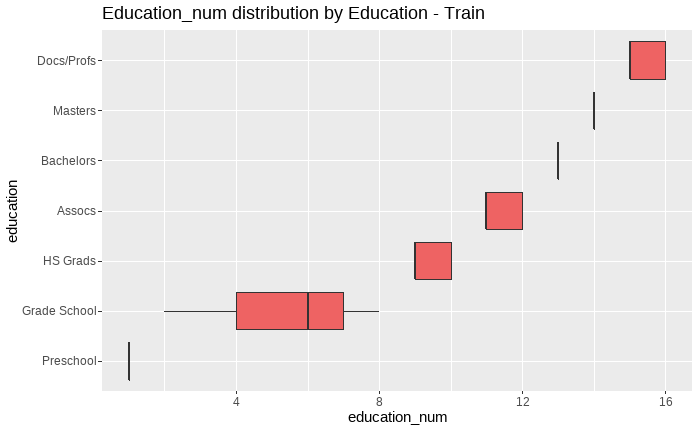
**Appendix 3.13 Pay response for Native Country pre transformation (Training and Test Sets)**

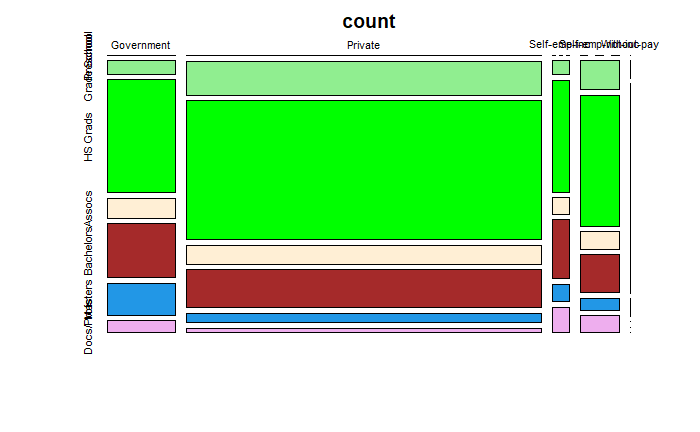
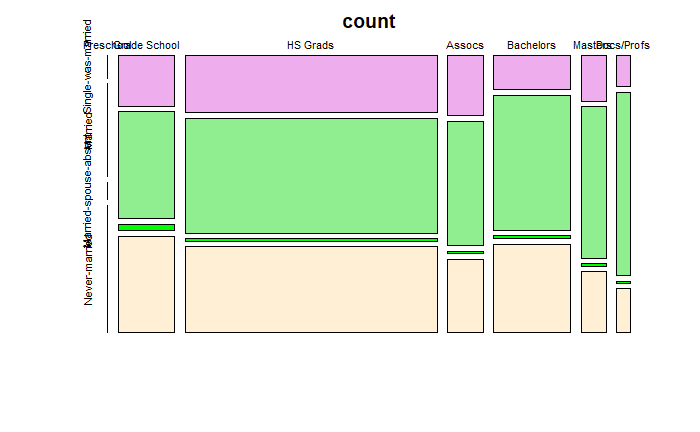
**Appendix 3.14 Pay response for Native Country post transformation (Training and Test Sets)**

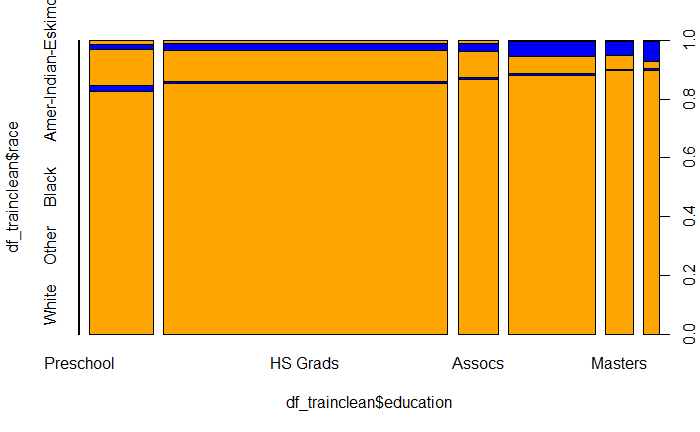
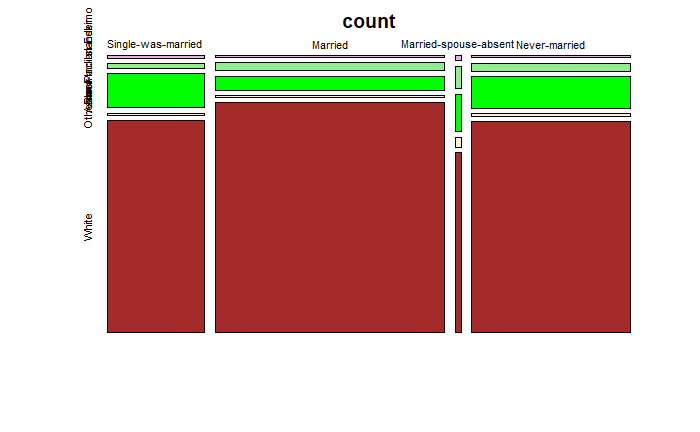
 

**Appendix 3.15 Box plot graphs of the redundancy of education and education\_num variables**

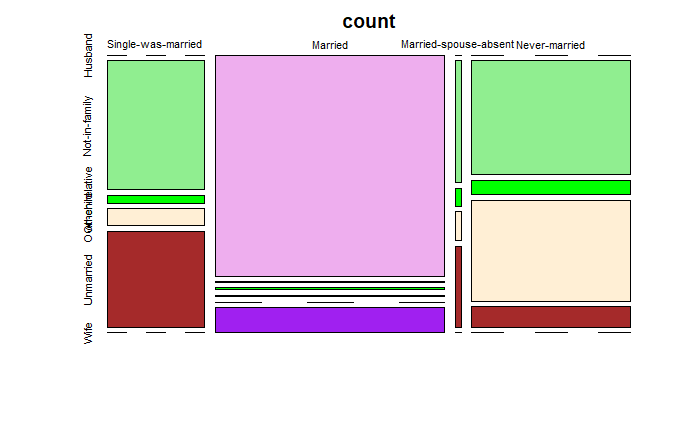


**Appendix 3.16 Mosaic plots to check for collinearity among categorial variables (minimal to no evidence)**

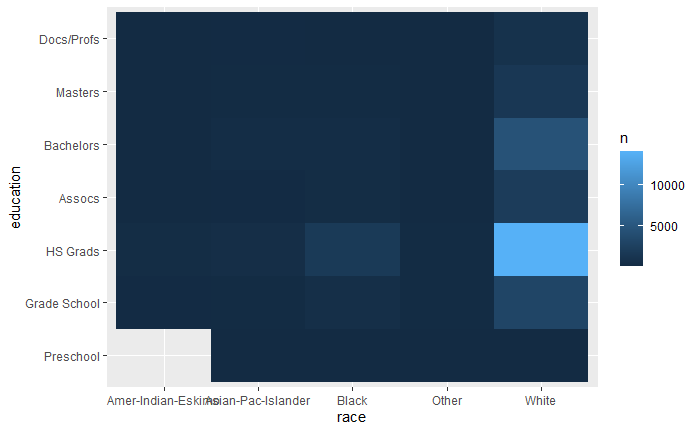
 

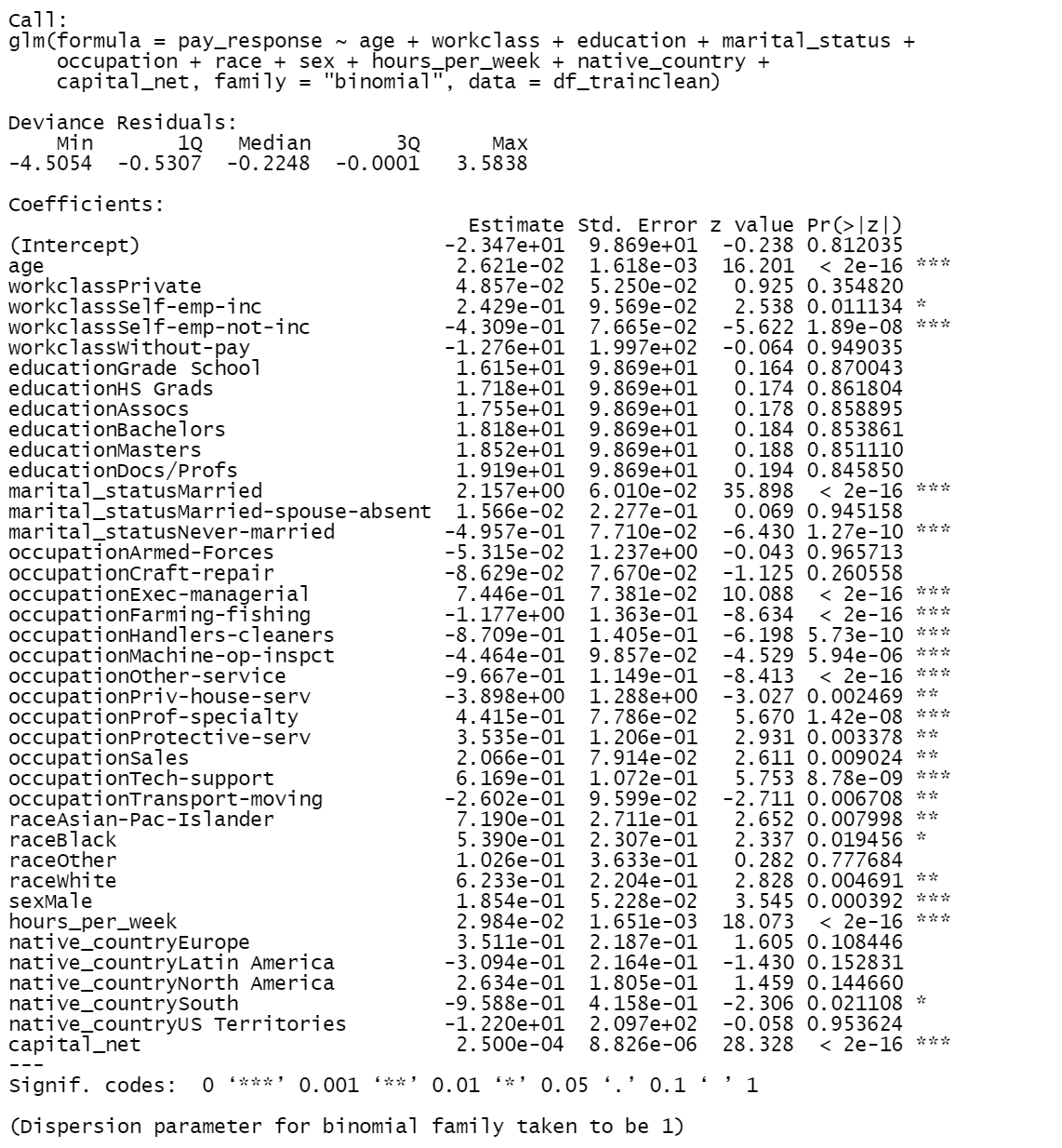
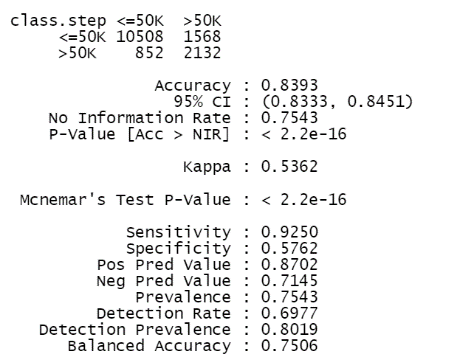
**Appendix 3.17 Mosaic plots to check for collinearity among categorial variables (Strong evidence)**



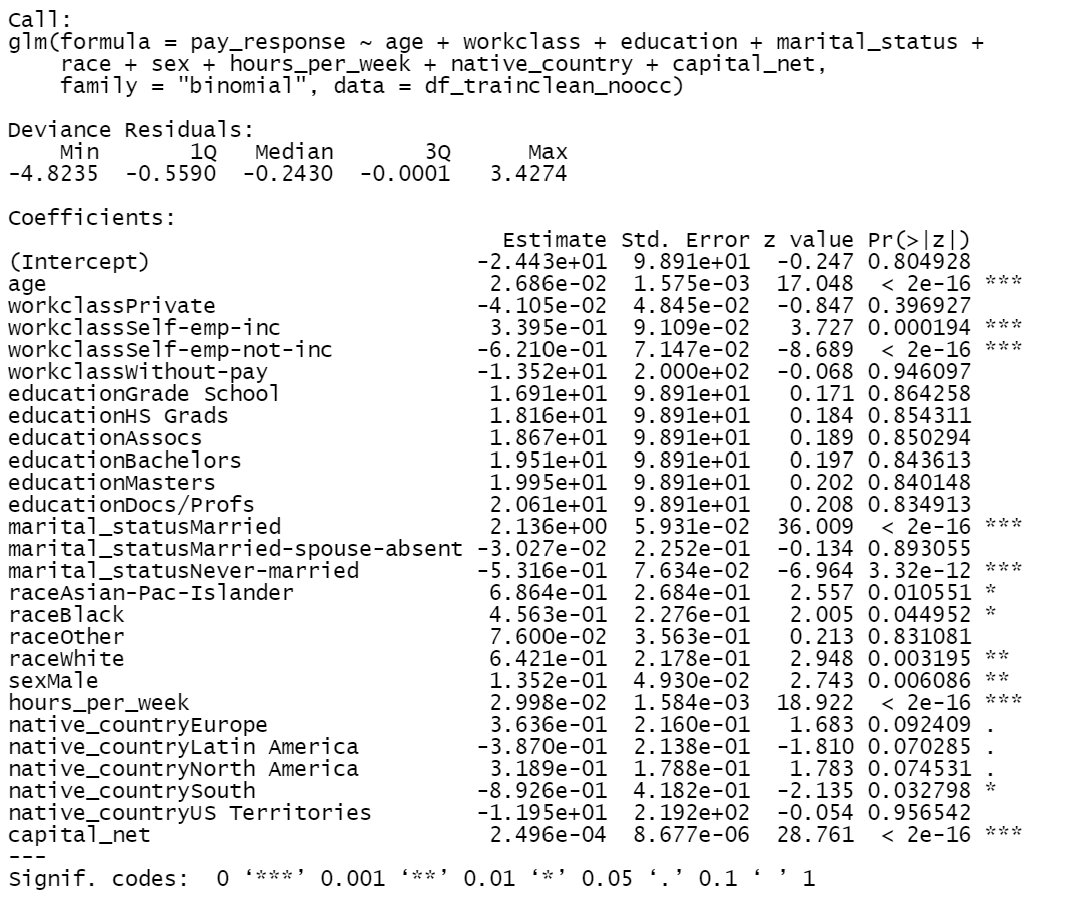
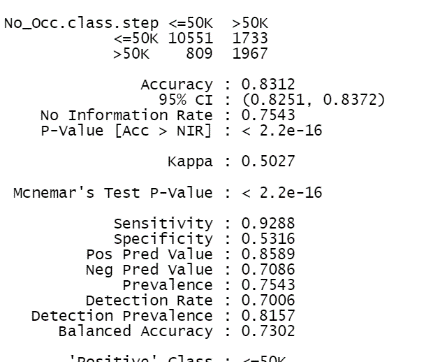
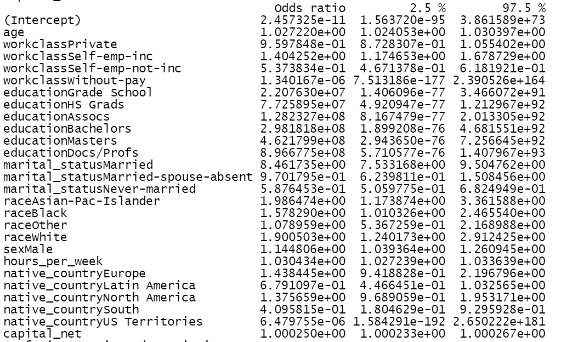
**Appendix 3.18 Race by education heat map count**



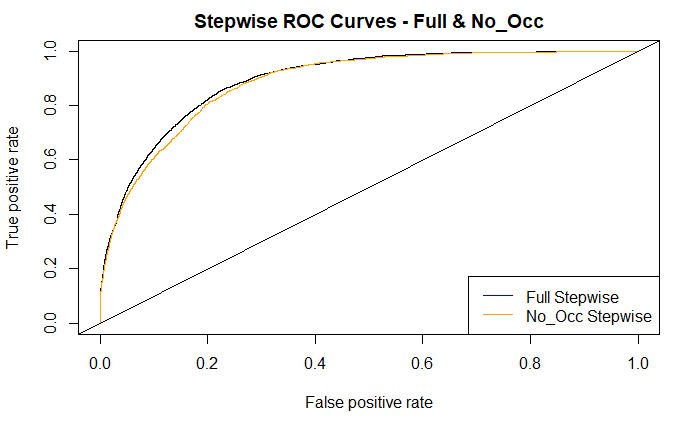
**Appendix 4.1 Step Wise Logistic Regression Output**

****

**Appendix 4.2 Step Wise Logistic Regression Output without occupation**

****

**Appendix 4.3 ROC for both stepwise test (one with occupation and one without)**



**Appendix 4.4 Cut off value of stepwise model excluding occupation**



**Appendix 5.1 Interaction Regression Output**

**Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated**

**Appendix 5.2 Interaction Regression Confusion Matrix**

Graphical user interface, text

Description automatically generated

**Appendix 5.3 Interaction Regression Accuracy / Cutoff Chart**

Chart, line chart

Description automatically generated

**Appendix 5.4 Interaction Regression ROC Curve**

Chart, scatter chart

Description automatically generated

**Appendix 5.5 shows the qqplot of the distribution with a response variable >50K.**

Graphical user interface, line chart

Description automatically generated with medium confidence

**Appendix 5.6 shows the qqplot of the distribution with a response variable <=50K.**

Chart, line chart

Description automatically generated

**Appendix 5.7 shows the scatter plot of the data indicating very little separation.**

**Chart

Description automatically generated**

**Appendix 5.8 shows the qqplot of the distribution with a response variable >50K after performing a log transformation.**

Graphical user interface, chart, line chart

Description automatically generated

**Appendix 5.9** shows the scatter plot of the data with better separation. However, the equal covariance assumption is violated.

Chart, scatter chart

Description automatically generated

**Appendix 5.10 shows the ROC plot of the LDA, QDA, and Stepwise models**

Chart, line chart

Description automatically generated

**Appendix 5.11 shows the output of the LDA model**

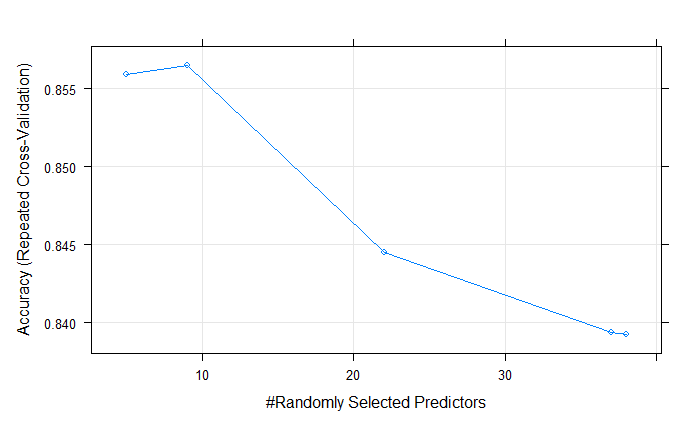
A screenshot of a computer

Description automatically generated with low confidence

**Appendix 5.12 shows the output of the QDA model**

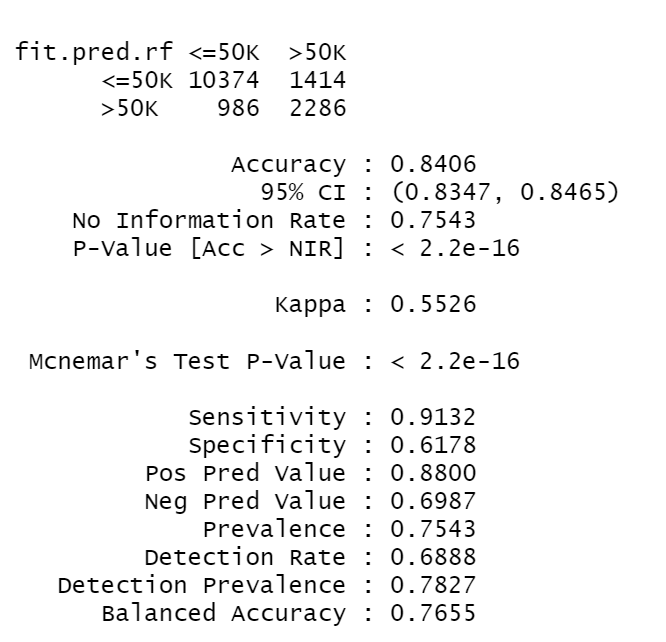
A picture containing text, receipt

Description automatically generated

**Appendix 5.13 MTRY generator output and Graph**

****

**Appendix 5.14 Random Forest Confusion Matrix for test results**

****

**R-Mark Down Code**

---

title: "SP2\_Master"

author: "Eric Laigaie"

date: "3/28/2022"

output: html\_document

---

Loading in packages

```{r setup, include=FALSE}

rm(list = ls())

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

library(readr)

library(tidyverse)

library(ggplot2)

library(GGally)

library(ggpubr)

library(MASS)

library(car)

library(caret)

library(ROCR)

library(glmnet)

library(corrplot)

library(RColorBrewer)

library(pheatmap)

library(psych)

library(tree)

library(ISLR)

library(randomForest)

```

### Data is stored in a .data and .test file. Let's read them as .csvs & define the column names.

```{r}

df\_train <- read.csv("https://raw.githubusercontent.com/ericlaigaie/StatsProject\_2/main/train.csv")

df\_test <- read.csv("https://raw.githubusercontent.com/ericlaigaie/StatsProject\_2/main/test.csv")

```

### A) Clean table and check for typos and NAs

### A.1) Fix Response Typo. Only in test.

```{r}

# Just realized the pay\_response variable is stored as '<=50K.' and '>50K.'. Let's get rid of those periods

df\_test <- df\_test %>% mutate(pay\_response = ifelse(pay\_response == '<=50K.', '<=50K', '>50K'))

```

### A.2) Check for NA's

```{r}

#check for number of null values in data

sapply(df\_train, function(x) sum(is.na(x)))

sapply(df\_test, function(x) sum(is.na(x)))

```

### A.3)summary tables

```{r}

#create summary

summary(df\_train)

str(df\_train)

summary(df\_test)

str(df\_test) # Update this from train to test.

#count number of levels

levelcount <- as.data.frame(t(df\_train%>% summarise\_all(n\_distinct)))

colnames(levelcount) <- c('Level Count')

print(levelcount)

#added code to count the levels for test.

levelcount\_tst <- as.data.frame(t(df\_test%>% summarise\_all(n\_distinct)))

colnames(levelcount\_tst) <- c('Level Count')

print(levelcount\_tst)

```

### A.4) Check ? Counts

```{r}

# No ?'s in education, marital\_status, relationship, race, sex, pay\_response

print(paste('Train - ? Working Class:', nrow(df\_train %>% filter(workclass == '?')), sep=' '))

print(paste('Train - ? Occupation:', nrow(df\_train %>% filter(occupation == '?')), sep=' '))

print(paste('Train - ? Native Country:', nrow(df\_train %>% filter(native\_country == '?')), sep=' '))

print(paste('Test - ? Working Class:', nrow(df\_test %>% filter(workclass == '?')), sep=' '))

print(paste('Test - ? Occupation:', nrow(df\_test %>% filter(occupation == '?')), sep=' '))

print(paste('Test - ? Native Country:', nrow(df\_test %>% filter(native\_country == '?')), sep=' '))

```

### A.5) Removing ? from Train and Test

```{r}

# Let's make a version 2 of train and test that has no ?'s

df\_train2 <- df\_train %>% filter(workclass != '?')

df\_train2 <- df\_train2 %>% filter(occupation != '?')

df\_train2 <- df\_train2 %>% filter(native\_country != '?')

df\_test2 <- df\_test %>% filter(workclass != '?')

df\_test2 <- df\_test2 %>% filter(occupation != '?')

df\_test2 <- df\_test2 %>% filter(native\_country != '?')

#Created 2 new dataframes: df\_train2 and df\_test2

```

### A.6) Change all strings to factors

```{r}

df\_train2 <- as.data.frame(unclass(df\_train2), stringsAsFactors = TRUE)

df\_test2 <- as.data.frame(unclass(df\_test2), stringsAsFactors = TRUE)

```

```{r}

#Gather the counts of the response variables in each dataset

describe(df\_train2)

# Count of <=50K and >50K

sum(with(df\_train2,pay\_response == "<=50K")) # 22654

sum(with(df\_train2,pay\_response == ">50K")) #7508

sum(with(df\_test2,pay\_response == "<=50K")) #11360

sum(with(df\_test2,pay\_response == ">50K")) #3700

```

###EDA Exploratory Data Analysis

###Will look for multicollinearity, ways to reduce levels in our categorical responses and removing columns in our clean output.

###This section check correlation of continous values

```{r}

# Find all numeric variables in cleaned dataset

numericVars <- which(sapply(df\_train2, is.numeric))

# Create df of numeric categories only

both\_numVar <- df\_train2[, numericVars]

#correlations of numeric variables

cor\_numVar <- cor(both\_numVar, use="pairwise.complete.obs")

#Sort by decreasing correlations with MSRP

cor\_sorted <- as.matrix(sort(cor\_numVar[,'age'], decreasing = TRUE))

# Select only high correlations

CorHigh <- names(which(apply(cor\_sorted, 1, function(x) abs(x)>0.01)))

cor\_numVar <- cor\_numVar[CorHigh, CorHigh]

# Plot correlations

corrplot.mixed(cor\_numVar, tl.col="black", tl.pos = "lt")

#We observe no correlation among the continuous variables

```

###Heat Map to check clusters

```{r}

#cluster.train.x <- df\_train2[,c(2,4,6,12,13,14)]

#cluster.train.y <- df\_train2$pay\_response

#cluster.train.y <- as.factor(as.character(cluster.train.y))

df\_train\_phmap <- as.data.frame(unclass(df\_train2), stringsAsFactors = TRUE)

cluster.train.x <- df\_train\_phmap[,c(2,4,12,13,14)]

cluster.train.y <- df\_train2$pay\_response

cols <- colorRampPalette(brewer.pal(9, "Set1"))

x<-t(cluster.train.x)

pos\_df = data.frame("Pay" = cluster.train.y)

colnames(x) <- rownames(pos\_df)

#pheatmap(x,cluster\_rows = F, annotation\_col=data.frame("Pay"=cluster.train.y),annotation\_colors=list("Pay"=c("<=50K"="white",">50K"="green")),scale="row",legend=T,color=colorRampPalette(c("blue","white", "red"), space = "rgb")(30000))

pheatmap(x,cluster\_rows = T, annotation\_col=pos\_df,annotation\_colors=list("Pay"=c("<=50K"="white",">50K"="green")),scale="row",legend=T,color=colorRampPalette(c("blue","white", "red"), space = "rgb")(30000))

```

### Check large factors - education

```{r}

# Let's try and see if we can collapse some levels of the education factor

# First, look at train ----------------------------------------------------------------------------------------------------

# Establish factor order

edu\_levels <- c('Preschool','1st-4th','5th-6th','7th-8th','9th','10th','11th','12th','HS-grad','Some-college','Assoc-voc','Assoc-acdm','Bachelors','Masters','Prof-school','Doctorate')

df\_train2 <- df\_train2 %>%

mutate(education = fct\_relevel(education, edu\_levels))

# Plot pay\_response proportion by level

ggplot(df\_train2, aes(y=education, fill=pay\_response)) + geom\_bar(position='fill') + labs(x='pay\_response proportion', title='Pay\_response Proportions by Education - Train')

# New levels of factors (any unlisted here are staying single)

grades <- c('1st-4th','5th-6th','7th-8th','9th','10th','11th','12th')

grads <- c('HS-grad','Some-college')

assocs <- c('Assoc-voc','Assoc-acdm')

docs <- c('Prof-school','Doctorate')

# Mutate education factor into new levels

df\_trainclean <- df\_train2 %>% mutate(

education = case\_when(

education == 'Preschool' ~ 'Preschool',

education %in% grades ~ 'Grade School',

education %in% grads ~ 'HS Grads',

education %in% assocs ~ 'Assocs',

education == 'Bachelors' ~ 'Bachelors',

education == 'Masters' ~ 'Masters',

education %in% docs ~ 'Docs/Profs'

)

)

# Establish new factor order

edu\_levels <- c('Preschool', 'Grade School', 'HS Grads', 'Assocs', 'Bachelors', 'Masters', 'Docs/Profs')

df\_trainclean <- df\_trainclean %>% mutate(education = factor(education), education = fct\_relevel(education, edu\_levels))

# Plot final levels

ggplot(df\_trainclean, aes(y=education, fill=pay\_response)) + geom\_bar(position='fill') + labs(x='pay\_response proportion', title='Pay\_response Proportions by Education - Train')

# Repeat process with test ----------------------------------------------------------------------------------------------

edu\_levels <- c('Preschool','1st-4th','5th-6th','7th-8th','9th','10th','11th','12th','HS-grad','Some-college','Assoc-voc','Assoc-acdm','Bachelors','Masters','Prof-school','Doctorate')

df\_test2 <- df\_test2 %>%

mutate(education = fct\_relevel(education, edu\_levels))

# Plot pay\_response proportion by level

ggplot(df\_test2, aes(y=education, fill=pay\_response)) + geom\_bar(position='fill') + labs(x='pay\_response proportion', title='Pay\_response Proportions by Education - Test')

# While the distributions here are slightly different, let's keep the same groups

grades <- c('1st-4th','5th-6th','7th-8th','9th','10th','11th','12th')

grads <- c('HS-grad','Some-college')

assocs <- c('Assoc-voc','Assoc-acdm')

docs <- c('Prof-school','Doctorate')

# Mutate education factor into new levels

df\_testclean <- df\_test2 %>% mutate(

education = case\_when(

education == 'Preschool' ~ 'Preschool',

education %in% grades ~ 'Grade School',

education %in% grads ~ 'HS Grads',

education %in% assocs ~ 'Assocs',

education == 'Bachelors' ~ 'Bachelors',

education == 'Masters' ~ 'Masters',

education %in% docs ~ 'Docs/Profs'

)

)

# Establish new factor order

edu\_levels <- c('Preschool', 'Grade School', 'HS Grads', 'Assocs', 'Bachelors', 'Masters', 'Docs/Profs')

df\_testclean <- df\_testclean %>% mutate(education = factor(education), education = fct\_relevel(education, edu\_levels))

# Plot final levels

ggplot(df\_testclean, aes(y=education, fill=pay\_response)) + geom\_bar(position='fill') + labs(x='pay\_response proportion', title='Pay\_response Proportions by Education - Test')

#Response variable (>50K) increases as the level of school increases!

```

### Check large factors - workclass

```{r workclass, fig.width=10, fig.height=6}

ggarrange(

ggplot(df\_train2, aes(y=workclass, fill=pay\_response)) + geom\_bar(position='fill') + labs(x='pay\_response proportion', title='Pay\_response Proportions by Workclass - Train') + theme(legend.position='none'),

ggplot(df\_test2, aes(y=workclass, fill=pay\_response)) + geom\_bar(position='fill') + labs(x='pay\_response proportion', title='Pay\_response Proportions by Workclass - Test') + theme(legend.position='none'),

nrow=1)

# There aren't too many levels here, and I can't find any pair I would really like to collapse together.

#Rayon - I agree with this. no clear way of collapsing it.All Workclass included our variable of interest. Exception is Without Pay in Training

#based on the original response for government workclass, we will group local, state and Fed into Government

df\_trainclean$workclass <- recode\_factor(df\_trainclean$workclass,

`State-gov` = "Government",

`Local-gov` = "Government",

`Federal-gov` = "Government")

df\_testclean$workclass <- recode\_factor(df\_testclean$workclass,

`State-gov` = "Government",

`Local-gov` = "Government",

`Federal-gov` = "Government")

#Visual of new workclass plot

ggarrange(

ggplot(df\_trainclean, aes(y=workclass, fill=pay\_response)) + geom\_bar(position='fill') + labs(x='pay\_response proportion', title='Pay\_response Proportions by Workclass - Train') + theme(legend.position='none'),

ggplot(df\_testclean, aes(y=workclass, fill=pay\_response)) + geom\_bar(position='fill') + labs(x='pay\_response proportion', title='Pay\_response Proportions by Workclass - Test') + theme(legend.position='none'),

nrow=1)

```

### Check large factors - occupation

```{r occupation, fig.width=10, fig.height=6}

ggarrange(

ggplot(df\_train2, aes(y=occupation, fill=pay\_response)) + geom\_bar(position='fill') + labs(x='pay\_response proportion', title='Pay\_response Proportions by occupation - Train') + theme(legend.position='none'),

ggplot(df\_test2, aes(y=occupation, fill=pay\_response)) + geom\_bar(position='fill') + labs(x='pay\_response proportion', title='Pay\_response Proportions by occupation - Test') + theme(legend.position='none'),

nrow=1)

# While there are a lot of levels here, I don't see any groups that would make sense to collapse together.

```

### Check large factors - marital\_status

```{r marital\_status, fig.width=10, fig.height=6}

ggarrange(

ggplot(df\_train2, aes(y=marital\_status, fill=pay\_response)) + geom\_bar(position='fill') + labs(x='pay\_response proportion', title='Pay\_response Proportions by marital\_status - Train') + theme(legend.position='none'),

ggplot(df\_test2, aes(y=marital\_status, fill=pay\_response)) + geom\_bar(position='fill') + labs(x='pay\_response proportion', title='Pay\_response Proportions by marital\_status - Test') + theme(legend.position='none'),

nrow=1)

# There's not a ton of levels and I don't see any groups that would make sense to collapse together.

# Rayon I agree

#based on the original responses we observed similar response portions for 1) married with spouse and 2) single who was once married

#we will group the following 1) Married-spouse = Married-civ-spouse + Married-af-spouse 2) Single\_was-married = Widowed, separated, divorced

df\_trainclean$marital\_status <- recode\_factor(df\_trainclean$marital\_status,

`Widowed` = "Single-was-married",

`Separated` = "Single-was-married",

`Divorced` = "Single-was-married",

`Married-AF-spouse` = "Married",

`Married-civ-spouse` = "Married")

df\_testclean$marital\_status <- recode\_factor(df\_testclean$marital\_status,

`Widowed` = "Single-was-married",

`Separated` = "Single-was-married",

`Divorced` = "Single-was-married",

`Married-AF-spouse` = "Married",

`Married-civ-spouse` = "Married")

#View new marital status levels responses

ggarrange(

ggplot(df\_trainclean, aes(y=marital\_status, fill=pay\_response)) + geom\_bar(position='fill') + labs(x='pay\_response proportion', title='Pay\_response Proportions by marital\_status - Train') + theme(legend.position='none'),

ggplot(df\_testclean, aes(y=marital\_status, fill=pay\_response)) + geom\_bar(position='fill') + labs(x='pay\_response proportion', title='Pay\_response Proportions by marital\_status - Test') + theme(legend.position='none'),

nrow=1)

```

### Check large factors - native\_country

```{r}

# First, let's look at train ---------------------------------------------------------------------------------------------

ggplot(df\_train2, aes(y=native\_country, fill=pay\_response)) + geom\_bar(position='fill') + labs(x='pay\_response proportion', title='Pay\_response Proportions by native\_country - Train')

# There are way too many levels here to keep. The most logical way to collapse these would be by continent

# Grouping by asia, europe, north america, and latin america, we are left with Outlying-US territories and 'South' -- this could be South Africa or South Korea...it is also weird that there are no African countries.

asia <- c('Cambodia','China','Hong','India','Iran','Japan','Laos','Philippines','Taiwan','Thailand','Vietnam')

europe <- c('England','France','Germany','Greece','Holand-Netherlands','Hungary','Ireland','Italy','Poland','Portugal','Scotland','Yugoslavia')

namerica <- c('Canada','United-States')

lamerica <- c('Cuba','Dominican-Republic','El-Salvador','Guatemala','Haiti','Honduras','Jamaica','Mexico','Nicaragua','Puerto-Rico','Trinadad&Tobago','Columbia','Ecuador','Peru')

islands <- c('Outlying-US(Guam-USVI-etc)')

df\_trainclean <- df\_trainclean %>% mutate(

native\_country = case\_when(

native\_country %in% asia ~ 'Asia',

native\_country %in% europe ~ 'Europe',

native\_country %in% namerica ~ 'North America',

native\_country %in% lamerica ~ 'Latin America',

native\_country %in% islands ~ 'US Territories',

TRUE ~ 'South'

)

)

ggplot(df\_trainclean, aes(y=native\_country, fill=pay\_response)) + geom\_bar(position='fill') + labs(x='pay\_response proportion', title='Pay\_response Proportions by native\_country - Train')

# Next, let's look at test -----------------------------------------------------------------------------------------------

ggplot(df\_test2, aes(y=native\_country, fill=pay\_response)) + geom\_bar(position='fill') + labs(x='pay\_response proportion', title='Pay\_response Proportions by native\_country - test')

# There are way too many levels here to keep. The most logical way to collapse these would be by continent

# Grouping by asia, europe, north america, and latin america, we are left with Outlying-US territories and 'South' -- this could be South Africa or South Korea...it is also weird that there are no African countries.

asia <- c('Cambodia','China','Hong','India','Iran','Japan','Laos','Philippines','Taiwan','Thailand','Vietnam')

europe <- c('England','France','Germany','Greece','Holand-Netherlands','Hungary','Ireland','Italy','Poland','Portugal','Scotland','Yugoslavia')

namerica <- c('Canada','United-States')

lamerica <- c('Cuba','Dominican-Republic','El-Salvador','Guatemala','Haiti','Honduras','Jamaica','Mexico','Nicaragua','Puerto-Rico','Trinadad&Tobago','Columbia','Ecuador','Peru')

islands <- c('Outlying-US(Guam-USVI-etc)')

df\_testclean <- df\_testclean %>% mutate(

native\_country = case\_when(

native\_country %in% asia ~ 'Asia',

native\_country %in% europe ~ 'Europe',

native\_country %in% namerica ~ 'North America',

native\_country %in% lamerica ~ 'Latin America',

native\_country %in% islands ~ 'US Territories',

TRUE ~ 'South'

)

)

ggplot(df\_testclean, aes(y=native\_country, fill=pay\_response)) + geom\_bar(position='fill') + labs(x='pay\_response proportion', title='Pay\_response Proportions by native\_country - test')

```

### Capital Gain & Capital Loss

```{r}

# We found that no person has nonzero gain and loss (figured as much), so how can we calculate cap\_net?

df\_train2 %>% filter(capital\_gain != 0 & capital\_loss != 0)

df\_test2 %>% filter(capital\_gain != 0 & capital\_loss != 0)

# Since we know that cap\_gain and cap\_loss cannot exist together, simply calculating cap\_gain - cap\_loss would give us what we want...two possible cases:

# Capital\_gain exists: Capital\_net = X - 0

# Capital\_loss exists: Capital\_net = 0 - X

df\_trainclean$capital\_net = df\_trainclean$capital\_gain - df\_trainclean$capital\_loss

df\_testclean$capital\_net = df\_testclean$capital\_gain - df\_testclean$capital\_loss

df\_trainclean <- dplyr::select(df\_trainclean,-c(capital\_gain, capital\_loss))

df\_testclean <- dplyr::select(df\_testclean,-c(capital\_gain, capital\_loss))

```

### Education & Education\_Num

```{r}

# Let's check for multicollinearity

ggplot(df\_trainclean, aes(y=education, x=education\_num)) + geom\_boxplot(fill='indianred2') + labs(title='Education\_num distribution by Education - Train') + scale\_y\_discrete("education", limits = c("Preschool" , "Grade School", "HS Grads", "Assocs", "Bachelors" , "Masters" , "Docs/Profs"))

ggplot(df\_testclean, aes(y=education, x=education\_num)) + geom\_boxplot(fill='indianred2') + labs(title='Education\_num distribution by Education - Test') + scale\_y\_discrete("education", limits = c("Preschool" , "Grade School", "HS Grads", "Assocs", "Bachelors" , "Masters" , "Docs/Profs"))

# Here, you can see that these two variables are directly related, as expected...so we can only include one. Since education is best viewed through factors (there isn't a numerical relationship between education levels), we'll keep education.

df\_trainclean <- dplyr::select(df\_trainclean,-c(education\_num))

df\_testclean <- dplyr::select(df\_testclean,-c(education\_num))

```

### Plot Matrix

```{r}

ggpairs(df\_trainclean, columns = c(2,4,11,14), ggplot2::aes(colour=pay\_response))

ggpairs(df\_testclean, columns = c(2,4,11,14), ggplot2::aes(colour=pay\_response))

#DY - what is this used for?

#Answer(Rayon) - Looking for Multicollinearity among the continuous variables.

```

### Response Variable Balance

```{r}

print(paste('Percentage of positive responses in Train: ',

round((nrow(df\_train2 %>% filter(pay\_response == '>50K')) / nrow(df\_train2)) \* 100,2),

'%', sep=''))

print(paste('Percentage of positive responses in Test: ',

round((nrow(df\_test2 %>% filter(pay\_response == '>50K')) / nrow(df\_test2)) \* 100,2),

'%', sep=''))

```

### What's Next?

1. We should look through the factor variables for any evidence of collinearity.

```{r workclass vs. education}

t(aggregate(workclass~education,data=df\_trainclean,summary))

count <- table(df\_trainclean$workclass, df\_trainclean$education)

plot(count, col=c("plum2","light green", "green", "papayawhip", "brown", off = 20))

#Observed a slight collinearity among the workclass and education variables.

```

```{r maritalstatus vs education}

t(aggregate(marital\_status~education,data=df\_trainclean,summary))

count <- t(table(df\_trainclean$marital\_status, df\_trainclean$education))

plot(count, col=c("plum2","light green", "green", "papayawhip", off = 20))

#Didn't observe any collinearity among the maritalstatus and education variables.

```

```{r Race vs education}

t(aggregate(race~education,data=df\_trainclean,summary))

count <- t(table(df\_trainclean$race, df\_trainclean$education))

plot(count, col=c("plum2","light green", "green", "papayawhip", "brown","purple", off = 20))

# Observed no multicollinearity among the variables race and education

# We observe race is heavily skewed to race=White

```

```{r race vs Marital status}

count <- t(table(df\_trainclean$race, df\_trainclean$marital\_status))

plot(count, col=c("plum2","light green", "green", "papayawhip", "brown", off = 20))

#Slight evidence of collinearity among race and marital status. However, there were more observations for White (85%) than other races in the data.

df\_trainclean %>% group\_by(race) %>% summarise(count = n())

```

```{r relationship and Marital status}

#Count of relationship vs. Marital Status

count <- t(table(df\_trainclean$relationship, df\_trainclean$marital\_status))

plot(count, col=c("plum2","light green", "green", "papayawhip", "brown","purple", off = 20))

# There is strong evidence of collinearity between race and marital status.

```

```{r}

df\_trainclean %>% group\_by(race, occupation) %>% summarise(count = n())

```

```{r}

#creating two new variable for LDA

#df\_trainclean\_lda <- df\_trainclean

#df\_testclean\_lda <- df\_testclean

# Remove fnlwgt --- Doesn't make sense logically and Turner said it was okay - Eric

df\_trainclean <- df\_trainclean %>% dplyr::select(-fnlwgt)

df\_testclean <- df\_testclean %>% dplyr::select(-fnlwgt)

df\_trainclean <- df\_trainclean %>% dplyr::select(-id)

df\_testclean <- df\_testclean %>% dplyr::select(-id)

df\_trainclean <- df\_trainclean %>% dplyr::select(-relationship)

df\_testclean <- df\_testclean %>% dplyr::select(-relationship)

```

2. We should consider if variables make sense to include in the model (fnlwgt).

### Objective 1 - Stepwise & LASSO Models

```{r}

# Full Dataset - Stepwise Model

full.log<-glm(pay\_response~.,family="binomial",data=df\_trainclean)

step.log<-full.log %>% stepAIC(trace=FALSE)

# Check Cook's Distance

Step.cd <- cooks.distance(step.log)

idx <- which(Step.cd >1) # row numbers

Step.cd[idx]

# All good here

# Summary of model

summary(step.log)

coef(step.log)

vif(step.log)

# Odds ratios of model

exp(cbind("Odds ratio" = coef(step.log), confint.default(step.log, level = 0.95)))

# Make predictions on model

fit.pred.step<-predict(step.log,newdata=df\_testclean,type="response")

# Set cutoff and make classifications from predictions

cutoff<- 0.5

class.step<-factor(ifelse(fit.pred.step>cutoff,">50K","<=50K"),levels=c("<=50K",">50K"))

# Print confusion matrix

conf.step<-table(class.step,df\_testclean$pay\_response)

cm.step <- confusionMatrix(conf.step)

cm.step

# Test different cutoffs - create vectors to store cutoffs and accuracies

step.cutoffs <- seq(.1,.9,.01)

step.accuracies <- c()

# Loop through cutoffs and find confusion matrix and accuracy of new classifications

for (i in step.cutoffs) {

curr.class.step <- factor(ifelse(fit.pred.step>i,">50K","<=50K"),levels=c("<=50K",">50K"))

curr.conf.step <- table(curr.class.step,df\_testclean$pay\_response)

curr.cm.step <- confusionMatrix(curr.conf.step)

step.accuracies <- append(step.accuracies, 100\*curr.cm.step$overall[1])

}

# Create dataframe from vectors and establish max value for each column

step.cutoff\_acc\_loop <- data.frame('cutoffs' = step.cutoffs, 'accuracies' = step.accuracies)

step.maxacc <- max(step.cutoff\_acc\_loop$accuracies)

step.maxcut <- step.cutoff\_acc\_loop %>% filter(accuracies == step.maxacc) %>% pull(cutoffs)

step.maxcut <- step.maxcut[1]

# Create accuracy geom\_line chart and mark maximum accuracy

ggplot(step.cutoff\_acc\_loop, aes(x=cutoffs, y=accuracies, label=accuracies)) +

geom\_line(size=2, color='indianred2') +

labs(x='Classiciation Cutoff', y='Accuracy', title='Accuracy by Cutoff Value - Stepwise') +

geom\_point(aes(y=step.maxacc, x=step.maxcut), size=5, color='black') +

geom\_text(

aes(label=

ifelse(cutoffs == step.maxcut,

paste(round(accuracies,2),'%, cutoff = ',step.maxcut,sep='')

,'')

),hjust=-.2, vjust=0

)

```

```{r}

# We have an idea to cut out occupation because it's similar to workclass and has many levels that decreases interpretability

# We feel like Occupation and Workclass are similar, and one could be removed....let's try Occupation

df\_trainclean\_noocc <- df\_trainclean %>% dplyr::select(-occupation)

df\_testclean\_noocc <- df\_testclean %>% dplyr::select(-occupation)

# No\_Occ Dataset - Stepwise Model

No\_Occ.log<-glm(pay\_response~.,family="binomial",data=df\_trainclean\_noocc)

No\_Occ.step.log<-No\_Occ.log %>% stepAIC(trace=FALSE)

# Check Cook's Distance

No\_Occ.cd <- cooks.distance(No\_Occ.step.log)

idx <- which(No\_Occ.cd >1)

No\_Occ.cd[idx]

# All good here

# Summary of model

summary(No\_Occ.step.log)

coef(No\_Occ.step.log)

vif(No\_Occ.step.log)

# Odds ratios of model

exp(cbind("Odds ratio" = coef(No\_Occ.step.log), confint.default(No\_Occ.step.log, level = 0.95)))

# Make predictions on model

No\_Occ.fit.pred.step<-predict(No\_Occ.step.log,newdata=df\_testclean\_noocc,type="response")

# Set cutoff and make classifications from predictions

cutoff<- 0.5

No\_Occ.class.step<-factor(ifelse(No\_Occ.fit.pred.step>cutoff,">50K","<=50K"),levels=c("<=50K",">50K"))

# Print confusion matrix

No\_Occ.conf.step<-table(No\_Occ.class.step,df\_testclean\_noocc$pay\_response)

No\_Occ.cm.step <- confusionMatrix(No\_Occ.conf.step)

No\_Occ.cm.step

# Test different cutoffs - create vectors to store cutoffs and accuracies

step.cutoffs <- seq(.1,.9,.01)

No\_Occ.step.accuracies <- c()

# Loop through cutoffs and find confusion matrix and accuracy of new classifications

for (i in step.cutoffs) {

curr.class.step <- factor(ifelse(No\_Occ.fit.pred.step>i,">50K","<=50K"),levels=c("<=50K",">50K"))

curr.conf.step <- table(curr.class.step,df\_testclean$pay\_response)

curr.cm.step <- confusionMatrix(curr.conf.step)

No\_Occ.step.accuracies <- append(No\_Occ.step.accuracies, 100\*curr.cm.step$overall[1])

}

# Create dataframe from vectors and establish max value for each column

No\_Occ.step.cutoff\_acc\_loop <- data.frame('cutoffs' = step.cutoffs, 'accuracies' = No\_Occ.step.accuracies)

No\_Occ.step.maxacc <- max(No\_Occ.step.cutoff\_acc\_loop$accuracies)

No\_Occ.step.maxcut <- No\_Occ.step.cutoff\_acc\_loop %>% filter(accuracies == No\_Occ.step.maxacc) %>% pull(cutoffs)

No\_Occ.step.maxcut <- No\_Occ.step.maxcut[1]

# Create accuracy geom\_line chart and mark maximum accuracy

ggplot(No\_Occ.step.cutoff\_acc\_loop, aes(x=cutoffs, y=accuracies, label=accuracies)) +

geom\_line(size=2, color='indianred2') +

labs(x='Classiciation Cutoff', y='Accuracy', title='Accuracy by Cutoff Value - Stepwise') +

geom\_point(aes(y=No\_Occ.step.maxacc, x=No\_Occ.step.maxcut), size=5, color='black') +

geom\_text(

aes(label=

ifelse(cutoffs == No\_Occ.step.maxcut,

paste(round(accuracies,2),'%, cutoff = ',No\_Occ.step.maxcut,sep='')

,'')

),hjust=-.2, vjust=0

)

```

```{r}

# Here, we can see that removing occupation does little to no damage on the model, but increases interpretability by a lot. One important variable to note is education...no levels are significant.

# Let's prepare a ROC curve for both models

# Prepare Step ROC Curve

full.results.step<-prediction(fit.pred.step, df\_testclean$pay\_response,label.ordering=c("<=50K",">50K"))

full.roc.step = performance(full.results.step, measure = "tpr", x.measure = "fpr")

# Prepare NoOcc Step ROC Curve

No\_Occ.results.step<-prediction(No\_Occ.fit.pred.step, df\_testclean\_noocc$pay\_response,label.ordering=c("<=50K",">50K"))

No\_Occ.roc.step = performance(No\_Occ.results.step, measure = "tpr", x.measure = "fpr")

plot(full.roc.step, main='Stepwise ROC Curves - Full & No\_Occ')

plot(No\_Occ.roc.step,col="orange", add = TRUE)

legend("bottomright",legend=c("Full Stepwise","No\_Occ Stepwise"),col=c("blue", "orange"),lty=1,lwd=1)

abline(a=0, b= 1)

```

```{r}

# Prepare matrices

dat.train.x <- model.matrix(pay\_response~.-1,df\_trainclean)

dat.train.y<-df\_trainclean[,10]

# Do cross-validation and plot

cvfit <- cv.glmnet(dat.train.x, dat.train.y, family = "binomial", type.measure = "class", nlambda = 1000)

plot(cvfit)

coef(cvfit, s = "lambda.min")

print("CV Error Rate:")

cvfit$cvm[which(cvfit$lambda==cvfit$lambda.min)]

#Optimal penalty

print("Penalty Value:")

cvfit$lambda.min

#For final model predictions go ahead and refit lasso using entire

#data set

finalmodel<-glmnet(dat.train.x, dat.train.y, family = "binomial",lambda=cvfit$lambda.min)

finalmodel$beta

#Test set predictions & confusion matrix

dat.test.x<-model.matrix(pay\_response~.-1,df\_testclean)

fit.pred.lasso <- predict(finalmodel, newx = dat.test.x, type = "response")

# Set cutoff and make classifications off of predictions

cutoff<-0.5

class.lasso<-factor(ifelse(fit.pred.lasso>cutoff,">50K","<=50K"),levels=c("<=50K",">50K"))

#Confusion Matrix for Lasso

conf.lasso<-table(class.lasso,df\_testclean$pay\_response)

confusionMatrix(conf.lasso)

# Test different cutoffs - create vectors to store cutoffs and accuracies

lasso.cutoffs <- seq(.1,.9,.01)

lasso.accuracies <- c()

# Loop through cutoffs and find confusion matrix and accuracy of new classifications

for (i in lasso.cutoffs) {

curr.class.lasso <- factor(ifelse(fit.pred.lasso>i,">50K","<=50K"),levels=c("<=50K",">50K"))

curr.conf.lasso <- table(curr.class.lasso,df\_testclean$pay\_response)

curr.cm.lasso <- confusionMatrix(curr.conf.lasso)

lasso.accuracies <- append(lasso.accuracies, 100\*curr.cm.lasso$overall[1])

}

# Create dataframe from vectors and establish max value for each column

lasso.cutoff\_acc\_loop <- data.frame('cutoffs' = lasso.cutoffs, 'accuracies' = lasso.accuracies)

lasso.maxacc <- max(lasso.cutoff\_acc\_loop$accuracies)

lasso.maxcut <- lasso.cutoff\_acc\_loop %>% filter(accuracies == lasso.maxacc) %>% pull(cutoffs)

# Create accuracy geom\_line chart and mark maximum accuracy

ggplot(lasso.cutoff\_acc\_loop, aes(x=cutoffs, y=accuracies, label=accuracies)) +

geom\_line(size=2, color='indianred2') +

labs(x='Classiciation Cutoff', y='Accuracy', title='Accuracy by Cutoff Value - lassowise') +

geom\_point(aes(y=lasso.maxacc, x=lasso.maxcut), size=5, color='black') +

geom\_text(

aes(label=

ifelse(accuracies == lasso.maxacc,

paste(round(accuracies,2),'%, cutoff = ',lasso.maxcut,sep='')

,'')

),hjust=-.2, vjust=0

)

```

```{r}

# Prepare LASSO ROC Curve

results.lasso<-prediction(fit.pred.lasso, df\_testclean$pay\_response,label.ordering=c("<=50K",">50K"))

roc.lasso = performance(results.lasso, measure = "tpr", x.measure = "fpr")

# Prepare Step ROC Curve

results.step<-prediction(fit.pred.step, df\_testclean$pay\_response,label.ordering=c("<=50K",">50K"))

roc.step = performance(results.step, measure = "tpr", x.measure = "fpr")

plot(roc.lasso)

plot(roc.step,col="orange", add = TRUE)

legend("bottomright",legend=c("Lasso","Stepwise"),col=c("blue", "orange"),lty=1,lwd=1)

abline(a=0, b= 1)

```

#Objective 2

#Complex variables to see if model improves

# Interaction Model

```{r}

# Build Interaction Model

fit.int <- glm(pay\_response ~ hours\_per\_week + capital\_net + age\*capital\_net + age:sex + race:age + age:marital\_status + native\_country:age + marital\_status:hours\_per\_week + occupation:sex + age:occupation, family='binomial', data=df\_trainclean)

# Check Cook's Distance

fit.int.cd <- cooks.distance(fit.int)

idx <- which(fit.int.cd >1) # row numbers

fit.int.cd[idx]

# All good here

# Model Summary

summary(fit.int)

#confint(fit.int)

# Make predictions on model

fit.pred.int<-predict(fit.int,newdata=df\_testclean,type="response")

# Set cutoff and make classifications from predictions

cutoff<- 0.48

class.int<-factor(ifelse(fit.pred.int>cutoff,">50K","<=50K"),levels=c("<=50K",">50K"))

# Print confusion matrix

conf.int<-table(class.int,df\_testclean$pay\_response)

cm.int <- confusionMatrix(conf.int)

cm.int

# Test different cutoffs - create vectors to store cutoffs and accuracies

int.cutoffs <- seq(.1,.9,.01)

int.accuracies <- c()

# Loop through cutoffs and find confusion matrix and accuracy of new classifications

for (i in int.cutoffs) {

curr.class.int <- factor(ifelse(fit.pred.int>i,">50K","<=50K"),levels=c("<=50K",">50K"))

curr.conf.int <- table(curr.class.int,df\_testclean$pay\_response)

curr.cm.int <- confusionMatrix(curr.conf.int)

int.accuracies <- append(int.accuracies, 100\*curr.cm.int$overall[1])

}

# Create dataframe from vectors and establish max value for each column

int.cutoff\_acc\_loop <- data.frame('cutoffs' = int.cutoffs, 'accuracies' = int.accuracies)

int.maxacc <- max(int.cutoff\_acc\_loop$accuracies)

int.maxcut <- int.cutoff\_acc\_loop %>% filter(accuracies == int.maxacc) %>% pull(cutoffs)

int.maxcut <- int.maxcut[1]

# Create accuracy geom\_line chart and mark maximum accuracy

ggplot(int.cutoff\_acc\_loop, aes(x=cutoffs, y=accuracies, label=accuracies)) +

geom\_line(size=2, color='indianred2') +

labs(x='Classiciation Cutoff', y='Accuracy', title='Accuracy by Cutoff Value - Interaction Based') +

geom\_point(aes(y=int.maxacc, x=int.maxcut), size=5, color='black') +

geom\_text(

aes(label=

ifelse(cutoffs == int.maxcut,

paste(round(accuracies,2),'%, cutoff = ',int.maxcut,sep='')

,'')

),hjust=-.2, vjust=0

)

# Prepare Step ROC Curve

results.int<-prediction(fit.pred.int, df\_testclean$pay\_response,label.ordering=c("<=50K",">50K"))

roc.int = performance(results.int, measure = "tpr", x.measure = "fpr")

plot(roc.int, col='Orange', main='ROC Model - Interaction Model')

legend("bottomright",legend=c('Interaction Model'),col=c("orange"),lty=1,lwd=1)

abline(a=0, b= 1)

```

# Create another model using just the continuouspredictors and use LDA or QDA.

# LDA/QDA CODE

## LDA Assumptions

###Assumption for equal variances between groups for each variable

```{r LDAassumptions1, fig.height=6, fig.width=10}

lda\_eq\_var <- function(myVar) {

ggplot(df\_trainclean, aes\_string(x='pay\_response', y={{myVar}}, fill='pay\_response')) +

geom\_boxplot() +

theme(legend.position='none') +

labs(title=paste({{myVar}}, "by pay\_response", sep=' '))

}

ggarrange(lda\_eq\_var("age"),lda\_eq\_var("hours\_per\_week"),lda\_eq\_var("capital\_net"),nrow=2,ncol=2)

lda\_eq\_var\_cov <- function(myVarX, myVarY) {

ggplot(df\_trainclean, aes\_string(x = {{myVarX}}, y = {{myVarY}}, col = 'pay\_response')) +

geom\_point() +

stat\_ellipse() +

labs(title=paste({{myVarX}},'vs.',{{myVarY}},'by outcome', sep=' ')) +

theme(legend.position='none')

}

ggarrange(lda\_eq\_var\_cov("age","hours\_per\_week"),

lda\_eq\_var\_cov("age","capital\_net"),

lda\_eq\_var\_cov("hours\_per\_week","capital\_net"),

nrow=2, ncol=3

)

```

###Normality assumption

```{r Normality Assumption}

pay.yes <- subset(df\_trainclean, pay\_response == ">50K")

pay.no <- subset(df\_trainclean, pay\_response == "<=50K")

variable\_1 <- c("age","hours\_per\_week", "capital\_net")

par(mfrow = c(2, 2))

for(i in variable\_1) {

qqnorm(pay.yes[[i]]); qqline(pay.yes[[i]], col = 2)

}

par(mfrow = c(2, 2))

for(i in variable\_1) {

qqnorm(pay.no[[i]]); qqline(pay.no[[i]], col = 2)

}

```

# LOG Transformation since assumptions failed for LDA with the continuous explanatory variables

### LDA LOGIC

```{r}

df\_traincleanlog <- df\_trainclean[, c(1, 8, 10, 11)]

df\_testcleanlog <- df\_testclean[, c(1, 8, 10, 11)]

df\_traincleanlog$logage <- log(df\_traincleanlog$age)

df\_traincleanlog$loghours\_per\_week <- log(df\_traincleanlog$hours\_per\_week)

df\_traincleanlog$logcapital\_net <- log(df\_traincleanlog$capital\_net)

df\_testcleanlog$logage <- log(df\_testcleanlog$age)

df\_testcleanlog$loghours\_per\_week <- log(df\_testcleanlog$hours\_per\_week)

df\_testcleanlog$logcapital\_net <- log(df\_testcleanlog$capital\_net)

lda\_eq\_var1 <- function(myVar) {

ggplot(df\_traincleanlog, aes\_string(x='pay\_response', y={{myVar}}, fill='pay\_response')) +

geom\_boxplot() +

theme(legend.position='none') +

labs(title=paste({{myVar}}, "by pay\_response", sep=' '))

}

ggarrange(lda\_eq\_var1("logage"),lda\_eq\_var1("loghours\_per\_week"),lda\_eq\_var1("logcapital\_net"),nrow=2,ncol=2)

lda\_eq\_var\_cov1 <- function(myVarX, myVarY) {

ggplot(df\_traincleanlog, aes\_string(x = {{myVarX}}, y = {{myVarY}}, col = 'pay\_response')) +

geom\_point() +

stat\_ellipse() +

labs(title=paste({{myVarX}},'vs.',{{myVarY}},'by outcome', sep=' ')) +

theme(legend.position='none')

}

ggarrange(lda\_eq\_var\_cov1("logage","loghours\_per\_week"),

lda\_eq\_var\_cov1("logage","logcapital\_net"),

lda\_eq\_var\_cov1("loghours\_per\_week","logcapital\_net"),

nrow=2, ncol=3

)

pay.yes <- subset(df\_traincleanlog, pay\_response == ">50K")

pay.no <- subset(df\_traincleanlog, pay\_response == "<=50K")

pay.tyes <- subset(df\_testcleanlog, pay\_response == ">50K")

pay.tno <- subset(df\_testcleanlog, pay\_response == "<=50K")

pay.yes <- subset(pay.yes, logcapital\_net != "-Inf" )

pay.yes <- subset(pay.yes, logcapital\_net != "NaN" )

pay.tyes <- subset(pay.tyes, logcapital\_net != "-Inf" )

pay.tyes <- subset(pay.tyes, logcapital\_net != "NaN" )

pay.no <- subset(pay.no, logcapital\_net != "-Inf" )

pay.no <- subset(pay.no, logcapital\_net != "NaN" )

pay.tno <- subset(pay.tno, logcapital\_net != "-Inf" )

pay.tno <- subset(pay.tno, logcapital\_net != "NaN" )

variable\_2 <- c("logage", "loghours\_per\_week", "logcapital\_net")

par(mfrow = c(2, 2))

for(i in variable\_2) {

qqnorm(pay.yes[[i]]); qqline(pay.yes[[i]], col = 2)

}

par(mfrow = c(2, 2))

for(i in variable\_1) {

qqnorm(pay.no[[i]]); qqline(pay.no[[i]], col = 2)

}

# what we need for LDA is that the point clouds for the two response categories should have similar elliptical shapes. Gross departures from this would require the use of QDA instead.

#Conclusion - we'll run QDA for our project.

```

# LDA Code for Logs

```{r}

df\_traincleanlogx <- rbind(pay.yes, pay.no)

df\_testcleanlogx <- rbind(pay.tyes, pay.tno)

#view(df\_traincleanlogx)

lda.fit = lda(pay\_response~., data = df\_traincleanlogx[,c(3,5,6,7)])

lda.fit

##Predict Training Results

predmodel.train.ldax <- predict(lda.fit, data=df\_traincleanlogx)

predmodel.train.ldax

# Create table to get prediction accuracy

table(Predicted <- predmodel.train.ldax$class, pay\_response = df\_traincleanlogx$pay\_response)

conf\_matrix\_train <- table(Predicted <- predmodel.train.ldax$class, pay\_response = df\_traincleanlogx$pay\_response)

sensitivity(conf\_matrix\_train)

specificity(conf\_matrix\_train)

# print out histogram to show split

ldahist(predmodel.train.ldax$x[,1], g= predmodel.train.ldax$class)

#Test

predmodel.test.ldax = predict(lda.fit, newdata=df\_testcleanlogx)

table(Predicted=predmodel.test.ldax$class, pay\_response=df\_testcleanlogx$pay\_response)

conf\_matrix\_test <- table(Predicted=predmodel.test.ldax$class, pay\_response=df\_testcleanlogx$pay\_response)

cm\_test <- confusionMatrix(conf\_matrix\_test, positive = "<=50K")

cm\_test

#ROC for LDA

preds <- predmodel.test.ldax$posterior

preds <- as.data.frame(preds)

pred <- prediction(preds[,2],df\_testcleanlogx$pay\_response)

roc.lda = performance(pred, measure = "tpr", x.measure = "fpr")

plot(roc.lda)

abline(a=0, b= 1)

```

#QDA

```{r}

qda.fit = qda(pay\_response~., data = df\_traincleanlogx[,c(3,5,6,7)])

qda.fit

##Predict Training Results

predmodel.train.qdax <- predict(qda.fit, data=df\_traincleanlogx)

predmodel.train.qdax

# Create table to get prediction accuracy

table(Predicted <- predmodel.train.qdax$class, pay\_response = df\_traincleanlogx$pay\_response)

conf\_matrix\_train <- table(Predicted <- predmodel.train.qdax$class, pay\_response = df\_traincleanlogx$pay\_response)

sensitivity(conf\_matrix\_train)

specificity(conf\_matrix\_train)

# print out histogram to show split

#ldahist(predmodel.train.qdax$x[,1], g= predmodel.train.qdax$class)

#Test

predmodel.test.qdax = predict(qda.fit, newdata=df\_testcleanlogx)

table(Predicted=predmodel.test.qdax$class, pay\_response=df\_testcleanlogx$pay\_response)

conf\_matrix\_test <- table(Predicted=predmodel.test.qdax$class, pay\_response=df\_testcleanlogx$pay\_response)

cm\_test <- confusionMatrix(conf\_matrix\_test, positive = ">50K")

cm\_test

#ROC for QDA

preds <- predmodel.test.qdax$posterior

preds <- as.data.frame(preds)

pred <- prediction(preds[,2],df\_testcleanlogx$pay\_response)

roc.qda = performance(pred, measure = "tpr", x.measure = "fpr")

plot(roc.qda)

abline(a=0, b= 1)

```

#Random Forest

```{r}

# Random Search for mtyr for Random Forest

#control <- trainControl(method="repeatedcv", number=2, repeats=2, search="random")

#set.seed(1234)

#mtry <- sqrt(ncol(x))

#rf\_random <- train(pay\_response~., data=df\_trainclean, method="rf", metric="Accuracy", tuneLength=5, trControl=control)

#print(rf\_random)

#plot(rf\_random)

#the optimal mtry is 9 based on our randome serach.

#Train Model with RF

train.rf<-randomForest(pay\_response~.,data=df\_trainclean,mtry=9,ntree=50,importance=T)

#predict test with RF model

#Print RF confusion matrix

fit.pred.rf<-predict(train.rf,newdata=df\_testclean,type="response")

conf.rf <- table(fit.pred.rf, df\_testclean$pay\_response)

cm.rf <- confusionMatrix(conf.rf)

cm.rf

#Create RF ROC table

pred.rf<-predict(train.rf,newdata=df\_testclean,type="prob")

pred.table.rf <- prediction(pred.rf[,2], df\_testclean$pay\_response)

roc.perf.rf = performance(pred.table.rf, measure = "tpr", x.measure = "fpr")

plot(roc.perf.rf)

plot(roc.qda, col ="red", add = TRUE)

abline(a=0, b= 1)

```

###Final ROC for outputs

```{r}

plot(roc.perf.rf)

plot(roc.lda, col = "purple", add = TRUE)

plot(roc.qda, col ="red", add = TRUE)

plot(roc.lasso,col="green", add = TRUE)

plot(roc.step,col="orange", add = TRUE)

legend("bottomright",legend=c("Random Forest", "LDA","QDA","Lasso","Stepwise","Random Forest"),col=c("black","purple","red","green", "orange"),lty=1,lwd=1)

abline(a=0, b= 1)

```