Using Data Mining Techniques to Analyze Tax Returns

Shadi Chamseddine - 100937807
Christopher Lee - 100937241

STAT 5703 W

Carleton University

Shirley Mills

Due Date: April 26, 2020

Introduction

It is important for any government agency to ensure its citizens are paying the proper amount of taxes. This however is not always the case as citizens may accidentally file their tax returns incorrectly, or in some cases deliberately find loopholes to avoid paying their taxes. It is crucial for the government to properly identify incorrectly filed tax returns and make adjustments to these client's financial statements when necessary. Proper auditing of client's financial statements will help ensure these statements are properly completed in the future and penalties and fines are handed out to individuals who are purposely evading their taxes. Prior to the significant improvement of computing power, government agencies would have to manually investigate a client's financial statements to determine if an audit was required. Given most countries have growing populations in the millions, it would be neither efficient nor practical to manually investigate each individual financial statement one by one to determine whether an audit is necessary. Today however, we can develop and make use of computer models which can quickly and systematically weed out suspicious or uncharacteristic tax returns and flag them to be audited. This saves significant time for auditors to conduct assessments on files requiring audits and also improves the likelihood that the government is collecting the proper amount of taxes annually. Using data mining techniques, we will investigate the effectiveness of various models to both properly flag tax returns which require audits, and to try and determine the dollar value of the adjustment that results from the audit.

Objective

Given a small sample of a client's financial statements, we aim to predict the binary variable (TARGET_Adjusted) which identifies whether a client's financial statement is a target to get audited and have potential adjustments applied to it. In addition we aim to predict the continuous variable (RISK_Adjustment) which records the monetary amount of any adjustment to the person's financial status as a result of a productive audit. This variable is a measure of the size of the risk associated with the person, where a productive audit refers to that which results in an adjustment being made to a client's financial statement. Given these goals, we would prefer to over-classifying a client's financial statement as requiring an audit as opposed to under-classifying as not requiring an audit. The reason for this is because it is easier to assess a client's financial statement and come to the conclusion that no adjustment is necessary than to completely miss a client's financial statement that requires an adjustment.

Methodology

We begin by loading the dataset into R and viewing it's properties which has 2,000 observations and 12 variables. The variables are:

- ID - unique identifier for each person,
- Age - age of person,
- Employment - type of employment
- Education - highest level of education
- Marital - current marital status
- Occupation - type of occupation
- Outcome - amount of income declared
- Gender - gender of person
- Deductions - total amount of expenses that a person claims in their financial statement
- Hours - average hours worked on a weekly basis

- RISK_Adjustment - the monetary amount of any adjustment to the person's financial status as a result of a productive audit, and
- TARGET_Adjusted - binary target variable indicating unproductive and productive audits

For our analysis, the ID variable can be dropped, as it is just a unique identifier for each person and holds no significant importance in predicting whether a client's financial statement should be audited and the amount of adjustment. Next we are interested in browsing through the data to determine any issues, inconsistencies, or missing values that we need to account for. We find there are missing values in two variables, 100 occurrences in the Employment variable and 101 occurences in the Occupation variable. Of these missing values, 100 occurrences are overlapping on the same observation, meaning that we are missing a value for both the Employment and the Occupation variable for those 100 observations. The single leftover missing value in the Occupation variable is associated with a value of "Unemployed", so for this observation we set the value of the associated Employment variable to "Unemployed". This leaves us with 100 rows of data that are missing a data point in both the Employment variable and the Occupation variable. One option to deal with these missing values would be to delete the 100 rows with missing values. However, we figured that deleting 100 rows of our relatively small dataset of 2,000 observations would cost us the explanatory information that we would hope to draw out of the remaining values for those observations. Therefore, we decided that we would deal with these missing values by imputing them using the missForest package. As this package allows for imputing missing values in datasets with various data types. It does this by training a random forest on the existing data in the dataset to predict the missing values.

After dropping out the ID column and imputing our missing values, our dataset is now left with 11 variables and 2,000 observations. Our next objective is to explore the possibility of data reduction on the dataset. While the current dataset is fairly small and would not be significantly taxing on computer resources when running analysis on it, we believe this dataset represents only a subset of client financial statements that one could expect to encounter in a government setting for instance. As such, we want to make use of data reduction to account for the possibility of datasets that could be many times larger, and for which data reduction could have significant implications on model performance. To determine which variables are important to the prediction process, we run a Random Forest algorithm on the dataset. There are two steps to the audit process, the first step is identifying whether or not a client's financial statement needs to be audited, then the next step is to apply adjustments if necessary. In our Random Forest algorithm we predict the TARGET_Adjusted using the remaining variables except RISK_Adjustment because it is determined after the value of TARGET_Adjusted is determined. From the results of this method we sort the variables from least to most importance using the Mean Decrease in Accuracy (MDA). The resulting values for each variable in the MDA is computed from the Out-Of-Bag (OOB) samples in the Random Forest algorithm. As each decision tree in the Random Forest is independent of one another, we can find the MDA or permutation importance each of these variables has on the overall model. Variables which contribute to less than 10% of the overall cumulative MDA will be dropped from our dataset.

The next step is to see if we can reduce our dataset even further, this time by the means of dimension reduction. The goal of dimension reduction is to be able to reduce the number of features in our analysis

without losing the key information that would negatively impact the results. At a high level our dataset now has two different types of data, qualitative variables and quantitative variables. We will employ three different methods of dimension reduction on the dataset, Principal Component Analysis (PCA), Multiple Correspondence Analysis (MCA), and a joint PCA and MCA on the dataset. PCA can only work on quantitative variables because it is based on the Euclidean distance between the different variables. To do this we convert all the qualitative variables to dummy variables using one-hot encoding and normalize all variables. One-hot encoding converts a qualitative variable into one or more quantitative variables depending on how many categories are present in the qualitative variable. Each of these newly created quantitative variables are binary representing whether they are a part of said category or not. Since the method works based on the Euclidean distance between the different variables we will need to normalize all the variables in order for variables with high quantitative values (eg. income) to not incorrectly dominate the calculations. The next method is doing a MCA on the qualitative variables. MCA is a form of correspondence analysis technique for qualitative variables. It provides the analyst with a general understanding of the relationship between the qualitative variables in the dataset. The final method is called Factor Analysis of Mixed Data (FAMD). FAMD works by using PCA for quantitative variables and MCA for qualitative variables. We employ a variety of dimension reduction methods to ensure the results are robust because of the different types of variables present in the dataset.

Once our dataset has been cleaned up and reduced into a smaller dataset without eliminating too much of the information we can proceed to predicting which client's financial statements to flag for an audit and what the audit adjustment would be if necessary. We can look at this prediction process in two ways, with unsupervised learning algorithms and with supervised learning algorithms. The reason we will try two different broad approaches is because we can look at the given dataset in two ways. We can either treat it as our full dataset or as a subset of the whole dataset. In the case we treat it as our full dataset, we will employ unsupervised learning algorithms and will drop the TARGET_Adjusted and RISK_Adjustment variables. We use the remaining variables to predict whether an audit (TARGET_Adjusted) on a client's financial statement will be necessary. The unsupervised learning algorithms we will use are K-Means clustering and hierarchical clustering to identify the two clusters (audit or no audit) in the data. In the case we treat it as a subset of the whole dataset we will employ supervised learning algorithms. There are two different values we wish to predict for each client's financial statement, their TARGET_Adjusted value and their RISK_Adjustment value. In the prediction of a client's TARGET_Adjusted value the RISK_Adjustment variable will be dropped from the models because it is determined after TARGET_Adjusted is determined. We will be using four different models to predict a client's TARGET_Adjusted value, these are Random Forest, K-Nearest Neighbour, Naïve Bayes, and Neural Networks. In the prediction of a client's RISK_Adjustment all variables will be used in the models because it is determined after TARGET_Adjusted is determined. We will be using two different models to predict a client's RISK_Adjustment, these are Random Forest, and a regression.

Results

Upon loading our dataset into R we view it's properties and variables to get a rough understanding of the values. Given there are so many variables in the dataset we visualize the data in Parallel Coordinates Plot as shown below in Figure 1. A scatterplot matrix illustrating the relationship between each of the variables is left in the appendix as Figure 15. The values are coloured by their TARGET_Adjusted value,

with red representing a TARGET_Adjusted value of 1 (audited) and yellow representing a TARGET_Adjusted value of 0 (not audited). We see mostly low positive values of RISK_Adjustment when TARGET_Adjusted is 1, meaning that most audits result in only small adjustments. We also see that RISK_Adjustment is 0 when TARGET_Adjustment is 0, which makes sense logically, as no audit, means no adjustments to a client's financial statement. TARGET_Adjusted values of 1 are usually associated with low income levels, select levels of education achievements (potential lower levels). Some marital statuses have higher proportions of being associated with having a TARGET_Adjusted value of 1. The other variables in the dataset do not show a fairly obvious distinction of clustering the TARGET_Adjusted value into subgroups.

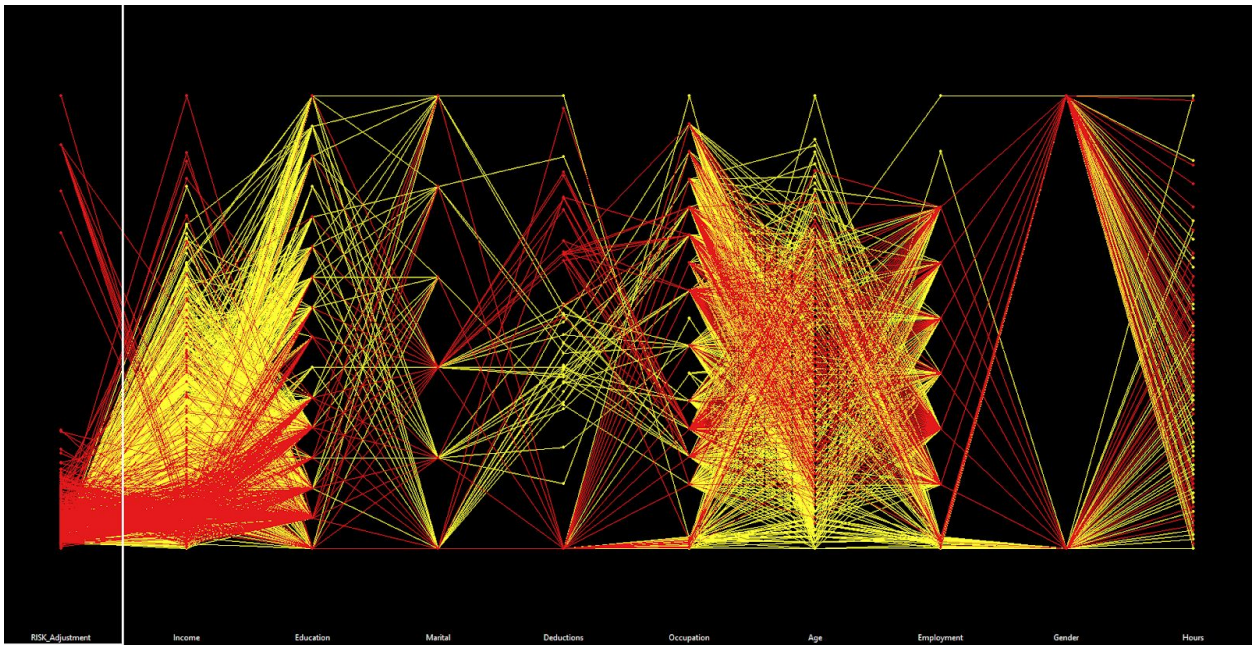

Figure 1

To the naked eye some of the variables do not seem to have an impact in determining the value of TARGET_Adjusted, using a Random Forest we determine the statistical significance of each of the variables in the model. This Random Forest was run with 2,000 decision trees to ensure convergence and had an Out Of Bag (OOB) estimate of error rate of 16.0%. The importance of each of the variables in this Random Forest model are shown below in Figure 2. Using the MDA measure we can see that Hours, Employment, and Gender variables have low impact on determining the value of TARGET_Adjusted. These three variables can be seen in Figure 1 above as having little to no predictive power on TARGET_Adjusted to the naked eye. Given our criteria of removing variables which contribute to less than 10% of the overall cumulative MDA, the Hours columns and Employment column will be dropped from our dataset.

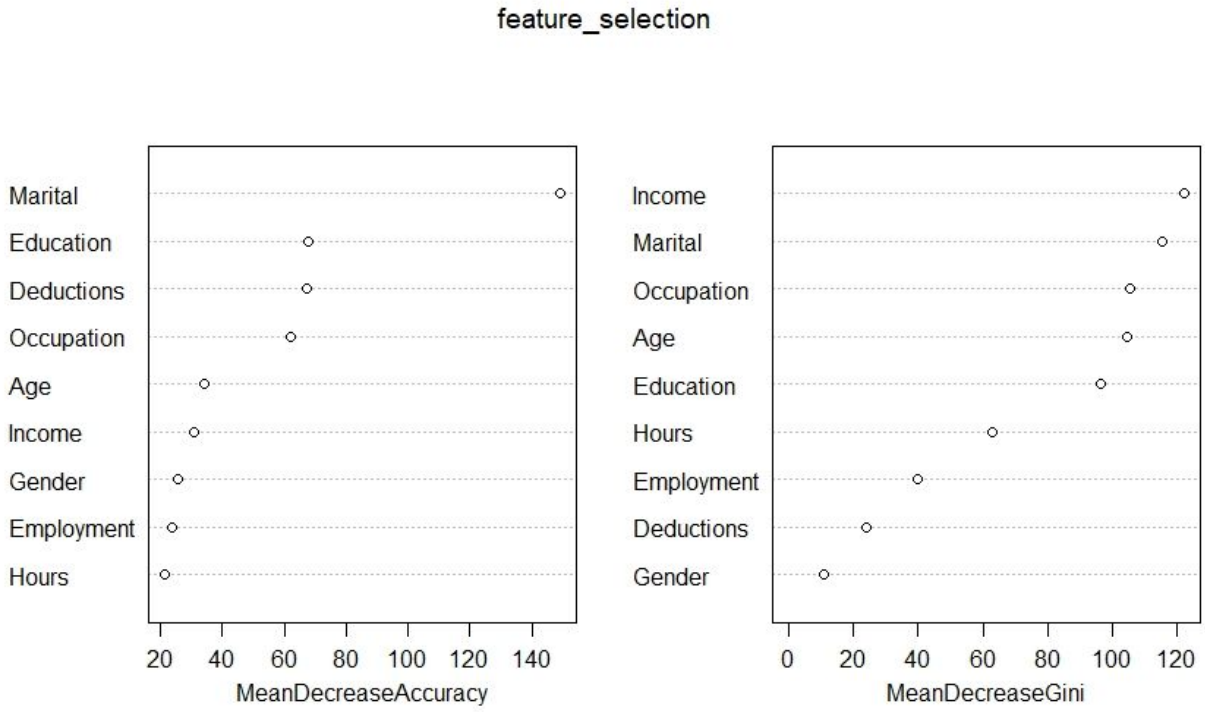

Figure 2

With fewer variables in our dataset, we seek to conduct dimension reduction on the dataset in aims of reducing the number of features in our analysis without losing key information. We begin by conducting a PCA on all the variables in the dataset. The result of this method is shown below in Figure 3. One of the main reasons to conduct dimension reduction is to allow for easier visualization of the dataset. With our results, if we would like to represent our data in 2D, we would require just the first two principal components. The cumulative variance explained by the first two principal components is 27.8% which is too low to explain the variance in the dataset. If we decide to represent our data in 3D, the first three principal components would be required. The cumulative variance explained by the first three principal components is 36.5% which is still too low to explain the variance in the dataset. Adding more principal components would be counterintuitive of dimension reduction so we will not go forward with this method.

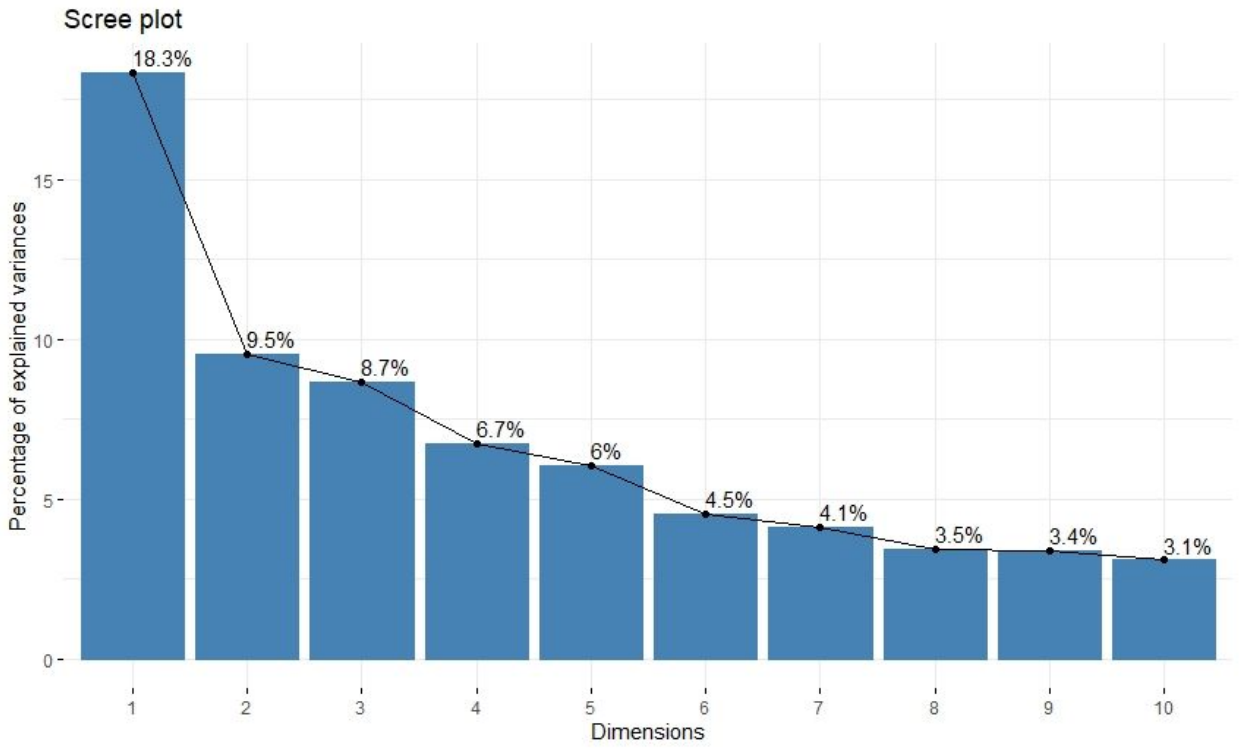

Figure 3

We figure a potential issue with the PCA approach comes from the conversion of the qualitative variables into quantitative variables using a one-hot encoding method. As such, the PCA analysis was not fully accurate in explaining the differences in the qualitative variables, resulting in suboptimal results. Given this outcome, we try a different approach by applying an MCA on just the qualitative variables to see if the variance is maintained. The result of this analysis is shown below in Figure 4. Once again if we look at just the first two principal components, the cumulative variance is 9.5% which is too low to explain the variance in the dataset. The first three principal components explains 13.7% of the cumulative variance, which is still too low to explain the variance in the dataset. We see that even focusing solely on the qualitative variables we still cannot extract a significant amount of the variance in the dataset.

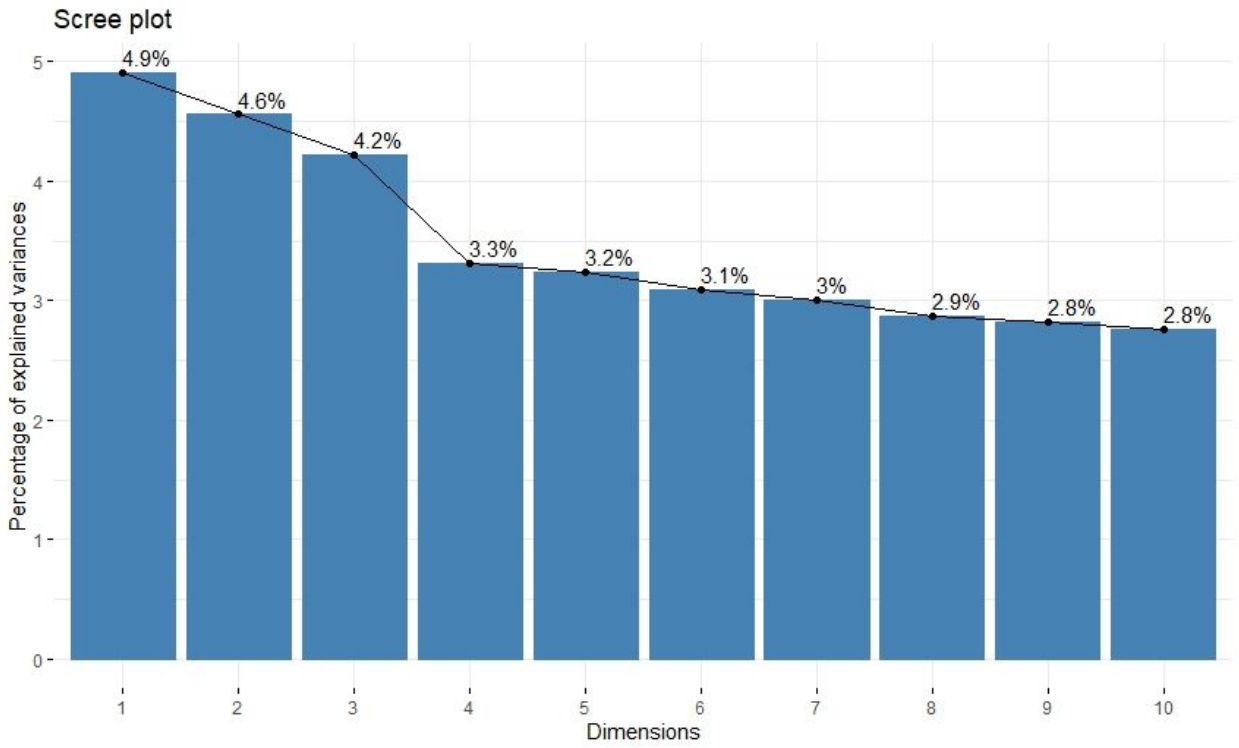

Figure 4

Given that neither the quantitative PCA approach nor the qualitative MCA approach were able to explain a significant amount of the variation in the dataset with a small number of dimensions, we attempt one final approach in which we combine the two methods. We combine both of these methods above using FAMD to see if a higher amount of the dataset's variance can be retained, this is shown below in Figure 5. Once again looking at the first two principal components, the cumulative variance is 10.5% which is too low to explain the variance in the dataset.

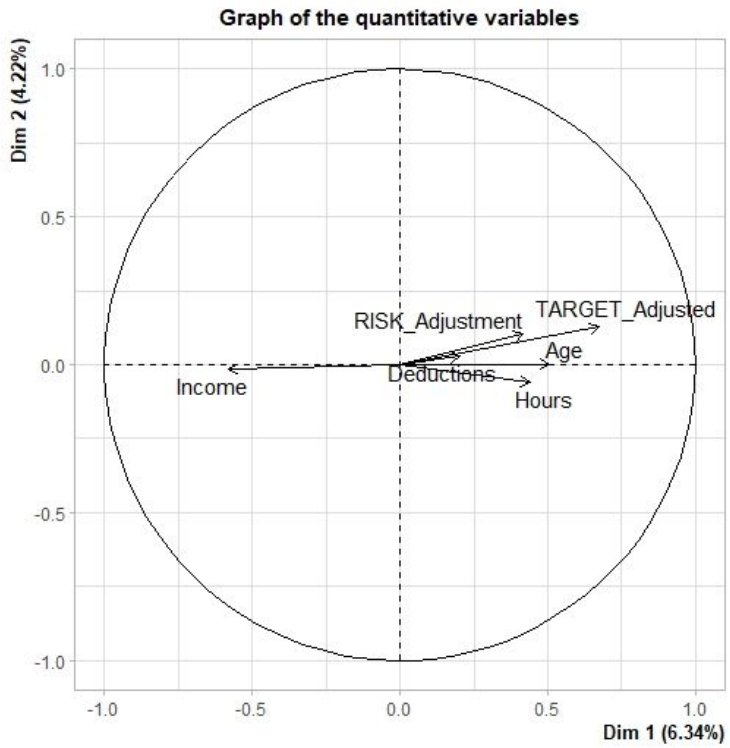

Figure 5

Given the results of the aforementioned dimension reduction methods, we have yet to yield an effective solution to dimension reduction for this particular dataset. Thus, we proceed with our analysis using all of the variables we had prior to attempting dimension reduction.

We move on to using unsupervised learning algorithms to try and predict whether an audit (TARGET_Adjusted) on a client's financial statement will be necessary. We begin by treating the existing dataset as the population and employ a variety of unsupervised learning algorithms such as K-Means cluster and hierarchical cluster to identify the two clusters (audit or no audit) in the data. The results from the K-Means cluster are shown below in Figure 6. The clustering routine does not generate a consistent labeling of the clusters. i.e. the 1 and 2 labels do not correspond with the 0 and 1 labels in our dataset. Upon further investigations of the results, we find a predicted value of 1 is associated with an actual value of 1 and a predicted value of 2 is associated with an actual value of 0. We can see that this clustering algorithm has an accuracy rate of 65.4%. The accuracy rate shows the K-Means clustering did a decent job of identifying the TARGET_Adjusted value and thus, whether a client's financial statement will need to be audited.

	Actual	
Predicted	0	1
1	603	373
2	934	90

Figure 6

Following the results from the K-Means clustering method we seek to analyze whether a hierarchical clustering method may better identify the clusters in the dataset. Hierarchical clustering assumes we do not know how many clusters there are in the dataset and will let the algorithm determine the optimal amounts of clusters. Three different methods will be used to join points together into a cluster:

- Single linkage which joins those with the closest points in clusters together
- Complete linkage which joins those with the farther points in clusters together
- Average linkage which joins those with the average points in clusters together

As we know there are two clusters in the dataset we will also create two cluster borders around the data. Hierarchical clustering using any of the three methods did not do a good job of identifying the clusters in the dataset, however the complete linkage method did the best out of the three. The results from the complete linkage method is shown below in Figure 7. The results from the single linkage (Figure 16) and average linkage (Figure 17) method are left in the appendix. We believe unsupervised learning did not do well in identifying the two clusters in any of the methods because of two reasons, the variable types and the complexity of the tax system. Firstly, as our unsupervised learning algorithms are based on Euclidean distance, the qualitative variables will need to be changed to quantitative variables. This conversion may distort some of the meaning between the variables. Secondly, the tax system is complex, it is not as simple as if A then yes, but more like if A and B then yes, but if A and C then no. The intricate relationships between the variables in the dataset and the potentially small sample size might make it difficult for the unsupervised learning algorithms to correctly identify the two clusters in the dataset.

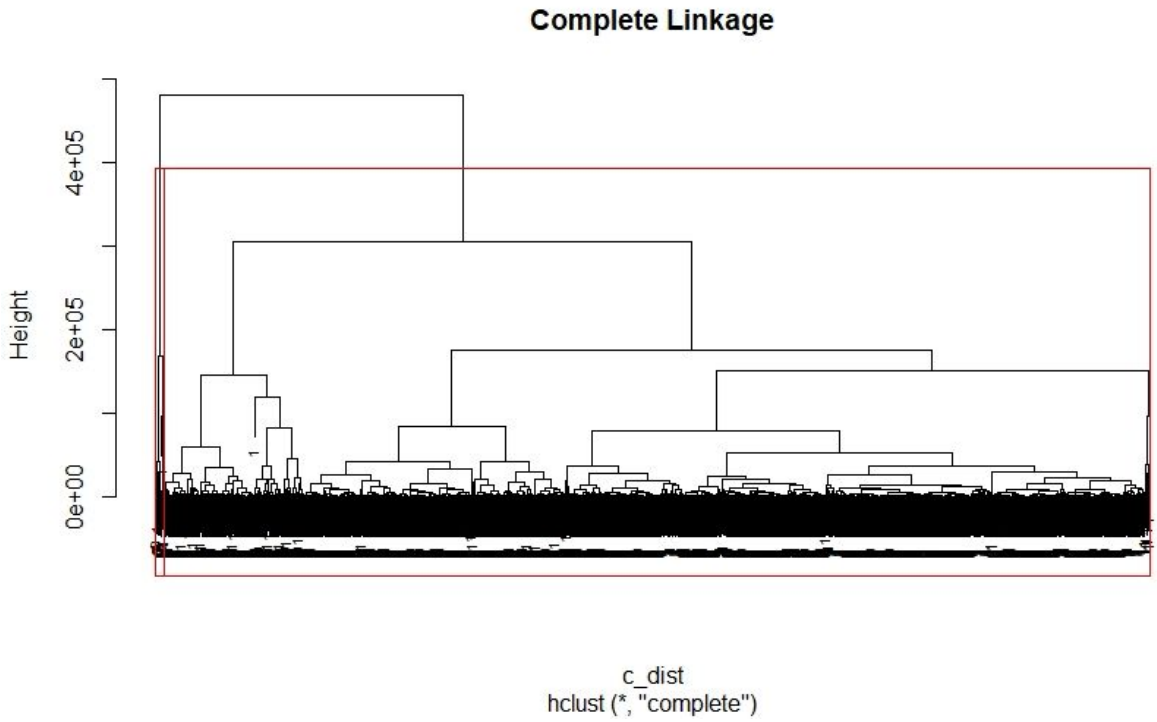

Figure 7

We will now treat our data as a subset of the whole dataset and we will employ supervised learning algorithms. Firstly, we will predict a client's TARGET_Adjusted value. The results using a Random Forest approach is shown below in Figure 8. This approach has an overall accuracy rate of 82.6%. It is important to capture the majority of client's financial statements that are flagged for audits than to miss any because it is easier to audit a client's financial statement and find out no adjustment is necessary compared to missing and not auditing a client's financial statement which requires an audit. These two scenarios are referred to as the sensitivity rate and the specificity rate. Sensitivity which is True Positives/(True Positives + False Negatives), it's the proportion of samples that are genuinely positive that give a positive result using the test in question. Specificity is True Negatives/(True Negatives + False Positives), this is the proportion of samples that test negative using the test in question that are genuinely negative. The sensitivity rate is 56.3% and the specificity rate is 91.4%.

	Actual	
Predicted	0	1
0	615	98
1	58	126

Figure 8

The results using a K-Nearest Neighbour approach is shown below in Figure 9. This approach has an overall accuracy rate of 69.1%. The sensitivity rate is 23.2% and the specificity rate is 84.4%.

	Actual	
Predicted	0	1
0	568	172
1	105	52

Figure 9

The results using a Naïve Bayes approach is shown below in Figure 10. This approach has an overall accuracy rate of 82.3%. The sensitivity rate is 56.7% and the specificity rate is 90.8%.

	Actual	
Predicted	0	1
0	611	97
1	62	127

Figure 10

The results using a Neural Network approach is shown below in Figure 11. This approach has an overall accuracy rate of 75.4%. The sensitivity rate is 1.3% and the specificity rate is 100.0%.

	Actual	
Predicted	0	1
0	673	221
1	0	3

Figure 11

Now that we have explored the various methods to try and predict whether or not an audit is needed, we can move on to try and predict the size of the adjustment given that an audit is necessary. In the prediction of a client's RISK_Adjustment all variables will be used in the models because it is determined after TARGET_Adjusted is determined. We will be using two different models to predict a client's RISK_Adjustment, these are Random Forest, and a regression model. Using a Random Forest we are able to explain 55.4% of the variance in the dataset. The residuals from this model are shown below in Figure 12.

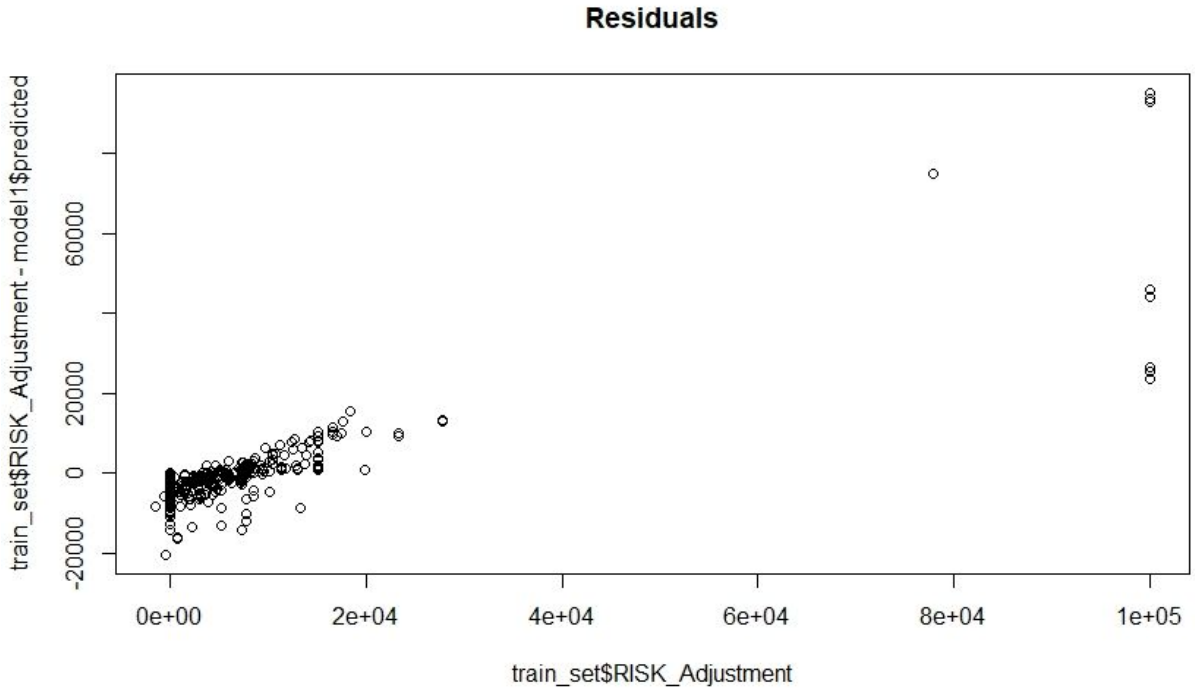

Figure 12

Next, regressing our independent variables; Age, Employment, Education, Marital, Occupation, Income, Gender, Deductions, Hours , and TARGET_Adjusted on our dependent variable RISK_Adjustment, provides us with more details on the magnitude and statistical significance in Figure 13 below. We can see that only five variables are statistically significant at the 5% level or greater, these are EmploymentPrivate, EmploymentFederal, EducationProfessional, Target_Adjusted, and the intercept which is an individual with the following specifications:

Employed as a consultant (EmploymentConsultant), with an education of an associate's degree (EducationAssociate), who is absent in their marital status (MaritalAbsent), is a cleaner by occupation (OccupationCleaner), and is female (GenderFemale).

```
Residuals:
   Min     1Q Median     3Q    Max
-24567  -1104    120    914  89728

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  3.916e+03  1.834e+03   2.135  0.03294 *
Age                          5.494e+00  1.712e+01   0.321  0.74836
EmploymentPrivate           -2.128e+03  7.490e+02  -2.841  0.00456 **
EmploymentPSFederal         -2.340e+03  1.154e+03  -2.028  0.04278 *
EmploymentPSLocal            1.749e+03  1.010e+03   1.732  0.08348 .
EmploymentPSState           -1.470e+03  1.139e+03  -1.291  0.19689
EmploymentSelfEmp           -2.046e+03  1.096e+03  -1.867  0.06211 .
EmploymentVolunteer         -2.131e+03  5.207e+03  -0.409  0.68246
EducationBachelor           -8.000e+02  1.191e+03  -0.672  0.50190
EducationCollege            -1.887e+03  1.155e+03  -1.634  0.10239
EducationDoctorate          -1.971e+02  2.257e+03  -0.087  0.93042
EducationHSgrad             -1.541e+03  1.142e+03  -1.349  0.17742
EducationMaster             -1.183e+03  1.420e+03  -0.833  0.40474
EducationPreschool          -2.565e+03  3.398e+03  -0.755  0.45047
EducationProfessional        1.714e+04  2.283e+03   7.507 1.01e-13 ***
EducationVocational         -1.308e+03  1.376e+03  -0.951  0.34193
EducationYr10               -1.306e+03  1.459e+03  -0.895  0.37091
EducationYr11               -1.340e+03  1.434e+03  -0.934  0.35027
EducationYr12               -1.783e+03  2.183e+03  -0.817  0.41411
EducationYr1t4              -6.790e+02  3.358e+03  -0.202  0.83978
EducationYr5t6              -1.774e+03  2.068e+03  -0.858  0.39092
EducationYr7t8              -1.479e+03  1.888e+03  -0.783  0.43357
EducationYr9                -1.053e+03  1.727e+03  -0.610  0.54193
MaritalDivorced             -1.077e+03  6.347e+02  -1.697  0.08983 .
MaritalMarried              -9.345e+02  5.311e+02  -1.760  0.07866 .
MaritalMarried-spouse-absent -1.522e+03 1.600e+03  -0.951  0.34185
MaritalUnmarried            -6.192e+02  9.934e+02  -0.623  0.53312
MaritalWidowed              -1.706e+03  1.340e+03  -1.273  0.20328
OccupationClerical           1.242e+03  9.689e+02   1.281  0.20022
OccupationExecutive         -7.646e+02  9.720e+02  -0.787  0.43163
OccupationFarming           -8.910e+02  1.387e+03  -0.643  0.52057
OccupationHome               2.794e+01  5.005e+03   0.006  0.99555
OccupationMachinist          4.415e+02  9.962e+02   0.443  0.65773
OccupationProfessional      -1.151e+03  1.047e+03  -1.100  0.27151
OccupationProtective        -5.872e+02  1.443e+03  -0.407  0.68419
OccupationRepair             4.246e+02  9.290e+02   0.457  0.64768
OccupationSales             -5.531e+02  9.584e+02  -0.577  0.56393
OccupationService            2.292e+02  9.123e+02   0.251  0.80165
OccupationSupport           -8.475e+02  1.349e+03  -0.628  0.52981
OccupationTransport         -4.637e+02  1.067e+03  -0.434  0.66403
Income                      -3.793e-03  3.017e-03  -1.257  0.20890
GenderMale                  -7.533e+02  4.948e+02  -1.522  0.12813
Deductions                  -8.799e-01  5.538e-01  -1.589  0.11230
Hours                        1.295e+01  1.629e+01   0.795  0.42680
TARGET_Adjusted              8.449e+03  5.261e+02  16.062  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6955 on 1555 degrees of freedom
Multiple R-squared:  0.2727,    Adjusted R-squared:  0.2521
F-statistic: 13.25 on 44 and 1555 DF,  p-value: < 2.2e-16
```

Figure 13

From the regression model we also obtain a QQ plot shown below in Figure 14. As the data values do not fall on a straight line our residuals are not normally distributed. This is because while most client's will

not have adjustments made to their financial statements, the few that do mostly have small positive adjustments. There are a few outliers with negative adjustments or large positive adjustments to their financial statements.

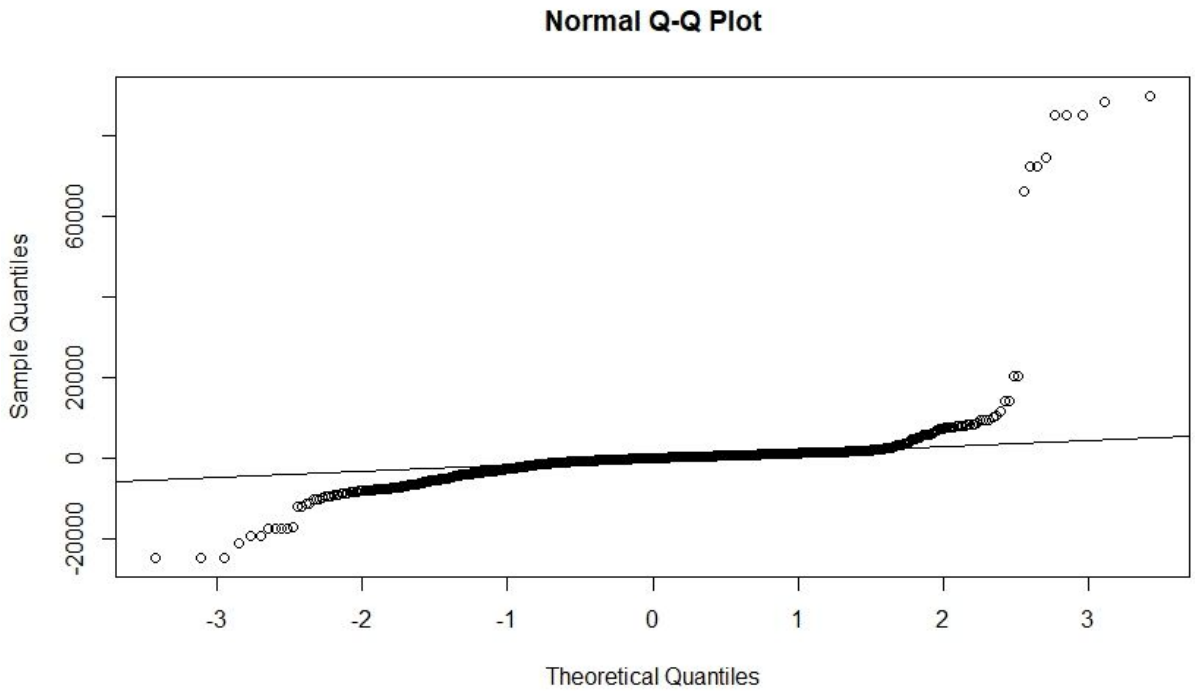

Figure 14

Conclusion

Our goal was to predict the binary variable (TARGET_Adjusted) to determine if a client's financial statement required an audit and the associated continuous variable (RISK_Adjustment) which determines the amount of adjustment required if a client is audited. We used a variety of data mining techniques to process, manipulate and visualize the data to prepare for our analysis. Overall, unsupervised learning did not do a good job overall in identifying the two clusters (TARGET_Adjusted) in the dataset across the methods that were attempted. On the other hand, many of our supervised learning methods did a relatively good job at identifying the two clusters in the dataset. However, the supervising learning algorithms did not do a good job at predicting the monetary amount of any adjustment to the person's financial status (RISK_Adjustment). Given these results, we figure that the feature columns provided in the dataset have more explanatory power about whether or not an audit is required, but once the audit is required, these variables do not provide as much insight into how large the adjustment will be.

Responsibility

Shadi Chamseddine

- Writing the report
- Data reduction
- Data visualization
- Unsupervised learning
- Supervised learning (RISK_Adjustment)

Christopher Lee

- Writing the report
- Dimension reduction
- Data visualization
- Supervised learning (TARGET_Adjusted)
- Supervised learning (RISK_Adjustment)

References

Hoare, J. (2018, July 30). How is Variable Importance Calculated for a Random Forest? Retrieved from https://www.displayr.com/how-is-variable-importance-calculated-for-a-random-forest/

Husson, F., Josse, J., Le, S., & Mazet, J. (2020, February 29). Package 'FactoMineR'. Retrieved from https://cran.r-project.org/web/packages/FactoMineR/FactoMineR.pdf

Stekhoven, D. (2016, August 29). Package 'missForest'. Retrieved from https://cran.r-project.org/web/packages/missForest/missForest.pdf

Appendix

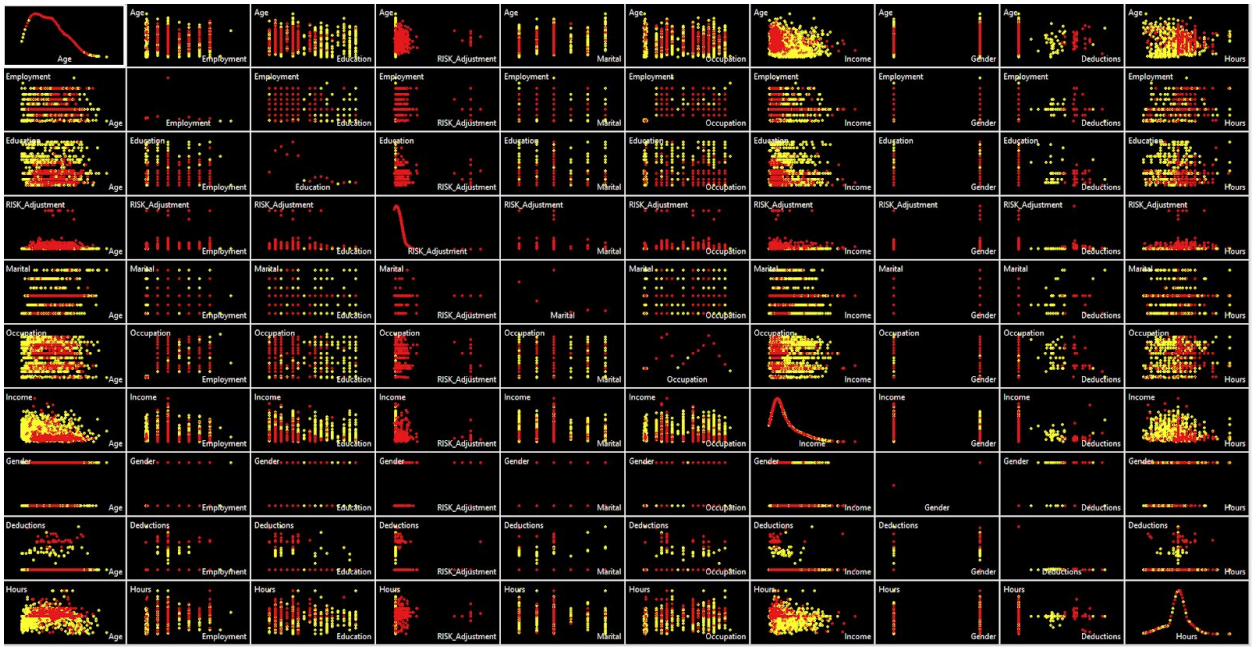

Figure 15

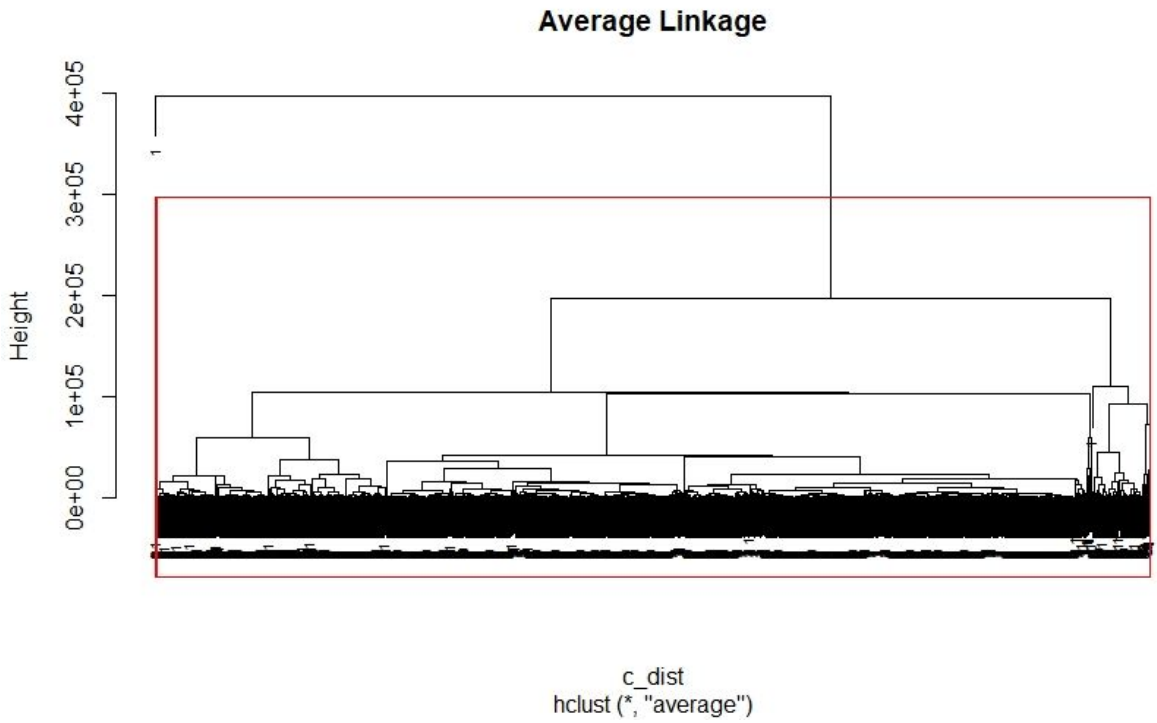

Figure 16

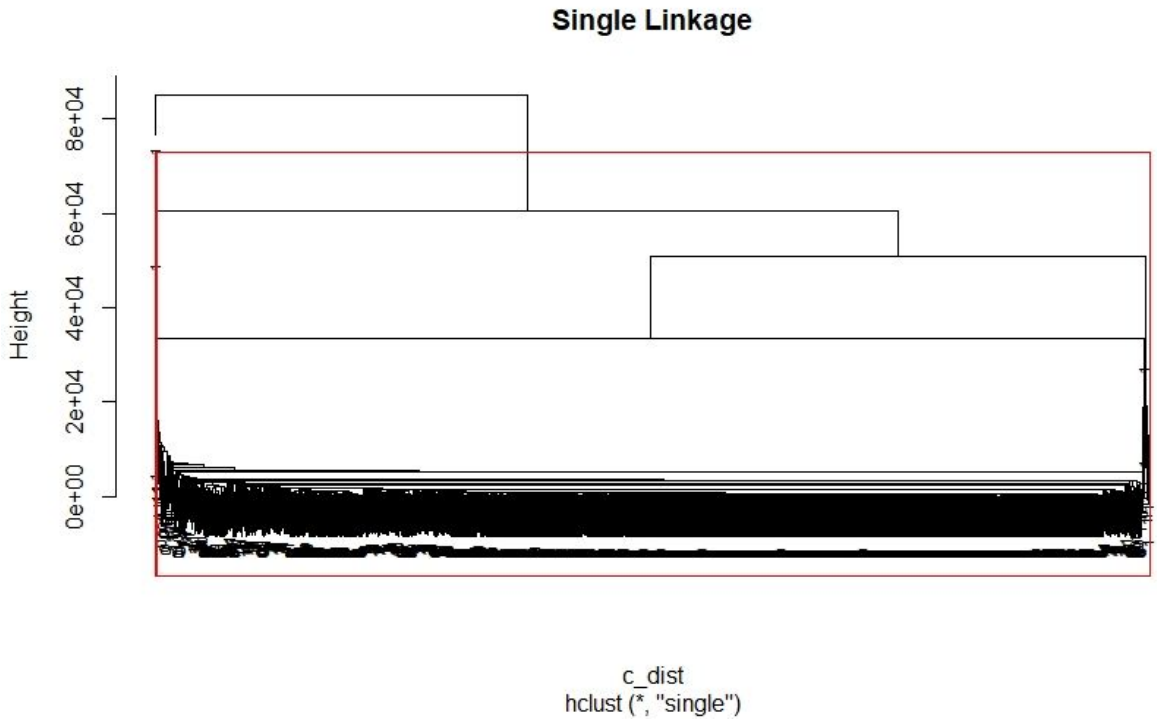

Figure 17

Code for the analysis

```
# Install packages if not already installed
if(!require(caret)) install.packages("caret") # Dummy variables
if(!require(class)) install.packages("class") # K-Nearest Neighbour
if(!require(e1071)) install.packages("e1071") # Naive Bayes
if(!require(FactoMineR)) install.packages("FactoMineR") # MCA
if(!require(factoextra)) install.packages("factoextra") # Graph MCA
if(!require(missForest)) install.packages("missForest") # Impute missing values
if(!require(nnet)) install.packages("nnet") # Neural networks
if(!require(randomForest)) install.packages("randomForest") # Random Forest
if(!require(rggobi)) install.packages("rggobi") # ggobi
if(!require(stats)) install.packages("stats") # K-Mmeans

# Load dataset into R
file_dir = paste0("C:/Users/Chris/Google Drive/University/Graduate Studies/Year 2 (Fall 2019 -
Winter 2020)/Term 2 (Winter 2019)/STAT 5703 - Data Mining I/Final Project/audit.csv")
audit = read.csv(file = file_dir)

# View dataset properties before data manipulation
dim(audit)
sapply(audit,class)
(sapply(audit, function(x) sum(length(which(is.na(x))))))
summary(audit)

# DATA REDUCTION

# Drop the ID variable
audit = subset(audit, select = -ID)

# Set the one record unemployed for both occupation and unemployment
audit$Occupation = as.character(audit$Occupation)
audit$Occupation[audit$Employment == "Unemployed"] = "Unemployed"
audit$Occupation = as.factor(audit$Occupation)

# Impute missing values, using all parameters as default values
imp = missForest(audit, maxiter = 10)
audit_data = imp$ximp

# Visualize data in ggobi
audit_vis = ggobi(audit)
display(audit_vis[1], "Scatterplot Matrix")
display(audit_vis[1], "Parallel Coordinates Display")
close(audit_vis)

# Random Forest to determine feature selection
feature_selection = randomForest(as.factor(TARGET_Adjusted) ~ Age + Employment + Education +
Marital + Occupation + Income +
                      Gender + Deductions + Hours, data = audit_data, ntree = 2000, replace =
TRUE, importance = TRUE)
feature_selection
varImpPlot(feature_selection)
imp_vars = as.data.frame(data.frame(importance(feature_selection)))
imp_vars = subset(imp_vars, select = MeanDecreaseAccuracy)
imp_vars$perc = apply(imp_vars, 2, function(x) x/sum(x))
imp_vars = imp_vars[order(imp_vars$perc),]
imp_vars$cum_perc = cumsum(imp_vars$perc)
# Drop variables that make up less than 10% of the variance
```

```
imp_vars = imp_vars[which(imp_vars$cum_perc >= 0.1),]
sel_vars = append(rownames(imp_vars), c("RISK_Adjustment", "TARGET_Adjusted"))

# Save new reduced dataset
audit_data = subset(audit_data, select = sel_vars)

# VISUALIZATION

# Visualize variable stats
for (i in 1:ncol(audit_data)) {
  print(table(audit_data[i]))
  plot(audit_data[,i], main=colnames(audit_data)[i], ylab = "Count")
  readline("Hit Enter for next column: ")
}

# View dataset properties after data manipulation
dim(audit_data)
sapply(audit_data,class)
(sapply(audit_data, function(x) sum(length(which(is.na(x))))))
summary(audit_data)

# DIMENSION REDUCTION

# Create dummy variables
dmy = dummyVars("~ .", data = audit_data)
audit_data_dum = data.frame(predict(dmy, newdata = audit_data))

#Normalize data
audit_data_dum_norm = as.data.frame(apply(audit_data_dum[,], 2, function(x) (x -
min(x))/(max(x)-min(x))))

# PCA on normalized data
audit_pca_dum_norm = prcomp(audit_data_dum_norm)
summary(audit_pca_dum_norm)
audit_pca_dum_norm
fviz_screeplot(audit_pca_dum_norm, addlabels = TRUE)

# MCA
audit_data_factor = audit_data[,sapply(audit_data, is.factor)]
audit_mca = MCA(audit_data_factor, graph = FALSE)
eig_val = get_eigenvalue(audit_mca)
fviz_screeplot(audit_mca, addlabels = TRUE)

# PCA and MCA
together = FAMD(audit_data, ncp = ncol(audit_data))
summary(together)

# UNSUPERVISED LEARNING (CLUSTERING)

# K-Means cluster
KM = kmeans(audit_data_dum_norm[1:(ncol(audit_data_dum_norm) - 2)], 2, 10)
table(Predicted = KM$cluster, Actual = audit_data$TARGET_Adjusted)

# Hierarchical clustering
c_dist = dist(audit_data_dum)

# Single linkage hierarchical clustering
```

```
c_hclust = hclust(c_dist, method = "single" )
c_hclust$labels = audit_data_dum$TARGET_Adjusted
plot(c_hclust, main = "Single Linkage", cex = 0.6)
rect.hclust(c_hclust, k = 2, border = "red")

# Complete linkage hierarchical clustering
c_hclust = hclust(c_dist, method = "complete" )
c_hclust$labels = audit_data_dum$TARGET_Adjusted
plot(c_hclust, main = "Complete Linkage", cex = 0.6)
rect.hclust(c_hclust, k = 2, border = "red")

# Average linkage hierarchical clustering
c_hclust=hclust(c_dist, method = "ave" )
c_hclust$labels = audit_data_dum$TARGET_Adjusted
plot(c_hclust, main = "Average Linkage", cex = 0.6)
rect.hclust(c_hclust, k = 2, border = "red")

# SUPERVISED LEARNING (CLASSIFICATION)

# Create training and testing sets
set.seed(123)
train = sample(nrow(audit_data), 0.8 * nrow(audit_data), replace = TRUE)
# Train and test sets for prediction of RISK_Adjustment
train_set = audit_data[train,]
test_set = audit_data[-train,]
# Train and test sets for prediction of TARGET_Adjusted
train_set_no_risk = subset(train_set, select = - RISK_Adjustment)
test_set_no_risk = subset(test_set, select = - RISK_Adjustment)

# Random Forest on TARGET_Adjusted
rf = randomForest(as.factor(TARGET_Adjusted) ~ ., data = train_set_no_risk, replace = TRUE,
importance = TRUE)
rf
plot(rf, main = "Random Forest")
# Prediction on the training set
pred_train_rf = predict(rf, train_set_no_risk, type = "class")
table(Predicted = pred_train_rf, Actual = train_set_no_risk$TARGET_Adjusted)
# Prediction on the test set
pred_test_rf = predict(rf, test_set_no_risk, type = "class")
table(Predicted = pred_test_rf, Actual = test_set_no_risk$TARGET_Adjusted)
mean(pred_test_rf == test_set_no_risk$TARGET_Adjusted)

# K-NN on TARGET_Adjusted
train_x_knn = model.matrix(~ 0 + ., data = train_set_no_risk)
test_x_knn = model.matrix(~ 0 + ., data = test_set_no_risk)
train_y_knn = train_set_no_risk$TARGET_Adjusted
# Prediction on the test set
test_y_knn = knn(train_x_knn, test_x_knn, train_y_knn, k=5)
table(Predicted = test_y_knn, Actual = test_set_no_risk$TARGET_Adjusted)
mean(test_y_knn == test_set_no_risk$TARGET_Adjusted)

# Naive Bayes on TARGET_Adjusted
nb = naiveBayes(as.factor(TARGET_Adjusted) ~ ., data=train_set_no_risk)
# Prediction on the training set
pred_train_nb = predict(nb, train_set_no_risk)
table(Predicted = pred_train_nb, Actual = train_set_no_risk$TARGET_Adjusted)
# Prediction on the test set
```

```
pred_test_nb = predict(nb, test_set_no_risk)
table(Predicted = pred_test_nb, Actual = test_set_no_risk$TARGET_Adjusted)
mean(pred_test_nb == test_set_no_risk$TARGET_Adjusted)

# Neural Networks on TARGET_Adjusted
nn = nnet(as.factor(TARGET_Adjusted) ~ ., data = train_set_no_risk, size = 5)
summary(nn)
# Prediction on the training set
pred_train_nn = predict(nn, newdata = train_set_no_risk, type = "class")
table(Predicted = pred_train_nn, Actual = train_set_no_risk$TARGET_Adjusted)
# Prediction on the test set
pred_test_nn = predict(nn, newdata = test_set_no_risk, type = "class")
table(Predicted = pred_test_nn, Actual = test_set_no_risk$TARGET_Adjusted)
mean(pred_test_nn == test_set_no_risk$TARGET_Adjusted)

# Attempt to do model averaging ensembles but for some reason the averaging gives wrong
results mathematically
test_results = subset(test_set, select = TARGET_Adjusted)
test_results$pred_rf_prob = predict(rf, test_set_no_risk, type = "class")
test_results$pred_knn_prob = knn(train_x_knn, test_x_knn, train_y_knn, k=5)
test_results$pred_nb_prob = predict(nb, test_set_no_risk)
test_results$pred_nn_prob = predict(nn, newdata = test_set_no_risk, type = "class")
test_results$pred_avg = round((as.numeric(test_results$pred_rf_prob) +
as.numeric(test_results$pred_knn_prob) +
                       as.numeric(test_results$pred_nb_prob) +
as.numeric(test_results$pred_nn_prob))/4,0)
table(Predicted = test_results$pred_avg, Actual = test_results$TARGET_Adjusted)

# Random Forest on RISK_Adjustment
model1 = randomForest(RISK_Adjustment ~ ., data = train_set, replace = TRUE, importance =
TRUE)
model1
plot(model1)
plot(train_set$RISK_Adjustment, train_set$RISK_Adjustment - model1$predicted, main =
"Residuals")

# Regression on RISK_Adjustment
reg = lm(RISK_Adjustment ~ ., data = train_set)
summary(reg)
pred = predict(reg, train_set)
pred = as.data.frame(pred)
x = as.data.frame(resid(reg))
plot(density(resid(reg)))
plot(train_set$RISK_Adjustment, resid(reg), main = "Residuals")
qqnorm(resid(reg))
qqline(resid(reg))

graphics.off()
```