Predicting Customer Spending Scores for Purchasing Cars

Ryan Cramer

Shivani Dandir

Jianbo Gong

Abhilash Halappanavar

MIS 545 Final Report

November 29, 2021

Abstract

This paper looks at predicting a customer's spending scores for cars. Accurate predictions of a given customer's spending score will lead to effective marketing campaigns and customer segmentation. To predict a customer's spending score, we created 5 different models. Each model had a binary output with a customer either having a "High" spending score or a "Low" spending score. Based on accuracy and reducing false negatives, we deemed the Neural Network model as the best model for our predictions.

Problem

-A statement of your problem. Justify why this problem is important (perhaps share an anecdote or some statistics) and how these data mining outcomes could potentially solve it. (Jianbo)

Customer segmentation is an important strategy in marketing. The core value of customer segmentation is to divide customers into different groups based on their attributes(such as: personalities, living areas, income…), so that companies can adopt and use the correct marketing strategies to achieve a better result in their sales. So, if a company could utilize a proper segmentation method in their potential customer groups, it can identify their real needs and provide the correct services(or adjust their current services to meet the needs).

In our project, we have customer data containing information of spending score, gender, profession, family size, years of work experience, age that was collected by an automobile company. We try to predict customers' spending scores (dependent variable), by using the rest of the features in order to market and recommend different models of automobiles to customers with different spending scores specifically. As a result, if we can successfully predict/classify the customers, our marketing operations could be more efficient and more effective. In this case, the automobile company could increase their sales and reduce these advertisement costs.

- Describe your dataset. Where/how you procured it, some interesting summary statistics, what variables are included along with their data types and description (present this in a table with the same format used in all of your lab assignment instructions), and how many records you have.

Dataset

Our dataset is based on automotive customer data and was acquired from Kaggle.com. The goal of the original data was to segment customers into 4 categories of buyers based on their acquired attributes. We were provided with a train-set.csv that had segment labels for customers and a test-set.csv that did not. For our project, we only used the train-set.csv which has 8,068 records before pre-processing.

There was little information about what segments or categories meant, so we decided to change our predicted variable to spending score. Spending score originally had 3 factors: Low, Average, and High. To get a binary output, all “Average” scores were turned into “High” spending scores. This left 60% of the customers having a low spending score and 40% having a high spending score. A major drawback to our data is not fully understanding what “spending score” means or its numeric value. Our assumption is that spending score refers to a customer's willingness to spend a small or large amount of money on an automobile.

With spending score as our dependent variable, the independent variables we kept in were gender, age, graduate status, profession, work experience in years, and family size. 55% of the customers were male, 58% were married, 62% had graduated, and the most popular profession was an artist (31% of customers). The attribute “Graduated” was not defined, so we assume it means a customer has graduated from high school. After getting rid of null values, there were 6,821 records remaining.

Data Type Table:

Column	R Tibble Data Type	Description
Male	logical	0 = Customer is a female 1 = Customer is a male
Age	integer	Age of customer
Graduate	logical	0 = Customer is not a graduate 1 = Customer is a graduate
Profession	factor	Profession of the customer
WorkExperience	integer	Work experience of the customer in years
FamilySize	integer	# of family members of the customer (including the customer)
SpendingScore	logical	0 = Low spending score 1 = High spending score

- List justified hypotheses for the impact each of your independent variables will have on the dependent variable (direct relationship, indirect relationship, or no relationship). It's okay if you end up wrong, but still provide your justification. (Jianbo)

After cleaning our dataset, there are six independent variables and one dependent variable that we are trying to predict.

The six independent variables are :
Gender
Age

Graduated (From at least high school)
Profession
Work experience
Family size
And the dependent variable is Spending score.

Based on the dataset and what we are trying to predict, we think age, work experience, and the graduated attribute will have a direct relationship on a customer's spending score. We think that an older individual, with more work experience and has graduated has a greater chance of having a "High" spending score over someone who is younger, less work experience, and has not graduated. This is because we think the first group has had more of an opportunity to make more money and will therefore spend more on a car.

We predict family size will have an indirect relationship with spending score. As family size increases, customers will have to spend their money on other expenses like college and more living expenses rather than an expensive car. We predict that gender will not have an impact on our model. We do not think a male or female should necessarily have a higher spending score and we hope our model is not bias towards one gender. Finally, profession we expect half to have a direct relationship and half to have an indirect relationship with our independent variable. We think being in a certain profession will either help or hurt your chances of getting a "High" spending score.

- Describe what you learned from your 3 interesting data queries. (Shivani)

```
print(spendingScoreTraining %>%
      group_by(Profession) %>%
      summarize(mean(Age)))
```

```
# A tibble: 9 x 2
  Profession    `mean(Age)`
  <fct>               <dbl>
1 Marketing            38.0
2 Lawyer               75.0
3 Homemaker            36.9
4 Healthcare           26.8
5 Executive            51.5
6 Entertainment        43.0
7 Engineer             42.1
8 Doctor               37.4
9 Artist               46.4
```

We have oldest Lawyer and youngest as Healthcare this was the insight gathered from our dataset

```
# 2/3 show work experience status for different gender groups
# (True for male, False for female; True for graduated, False for not graduated)
print(spendingScoreTraining %>%
      group_by(Male) %>%
      summarize(mean(Graduated)))
```

```
# A tibble: 2 x 2
  Male  `mean(Graduated)`
  <lgl>             <dbl>
1 FALSE             0.653
2 TRUE              0.616
```

We found out that Females are more educated than males

```
# 3/3 average group high spending score numbers
print(spendingScoreTraining %>%
      group_by(Profession) %>%
      filter(SpendingScore == TRUE) %>%
      summarize(mean(WorkExperience)))
```

```
> # 3/3 average group high spending score numbers
> print(spendingScoreTraining %>%
+         group_by(Profession) %>%
+         filter(SpendingScore == TRUE) %>%
+         summarize(mean(WorkExperience)))
# A tibble: 9 x 2
  Profession    `mean(WorkExperience)`
  <fct>                          <dbl>
1 Marketing                       2.41
2 Lawyer                          1.17
3 Homemaker                       6.33
4 Healthcare                      3.53
5 Executive                       2.14
6 Entertainment                   2.30
7 Engineer                        2.75
8 Doctor                          2.45
9 Artist                          2.19
```

There is an anomaly here: we have the oldest age group in Lawyer and that have the lowest years of experience which is odd for a dataset. It could be possible there is some issue in the data gathering process.

Few more data queries:-

```
#filter
meta_Spendingscore <- Spendingscore %>%
filter(Profession == "Engineer")
#there are only 602 records for engineer and by filtering we can get those records
```

```
#select
Spendingscore1 <- Spendingscore %>% select(-Age)
# we can remove data from the tibble by using select, or customize the view
#group_by and summarize
metadata <-Spendingscore %>% group_by(WorkExperience)%>%summarize(n())
print(metadata)
#we get the total number of records grouped by workexperience, we found out that there are more people with 0 and 1 year of workex compared to others
```

- Describe what steps you did to preprocess your data (Ryan)

Most of our pre-processing happened in excel. Excel's user interface allows us to quickly and easily manipulate the data to make training our model in R easier. The first step was to get a binary dependent variable. This was achieved by changing "Average" in the spending score column to "High". Now the customer spending score was "Low" 60% of the time and "High" 40% of the time.

Next we removed any rows that had null or blank values. This was done by sorting the columns in ascending order and deleting blank rows that would appear at the bottom of the dataset. The next step was transforming our logical columns into 0s and 1s to simplify the dataset. Gender was turned into a Male column with 1 being a male and 0 being a female. The graduate column was updated to have a 1 if the customer graduated and a 0 if they did not. The spending score was updated to be a 1 if the score was "High" and a 0 if the score was "Low". Now the data was ready to be read into a tibble in R.

Using RStudio, we made our "Profession" variable into a dummy variable. This created 9 new columns with a 1 meaning the customer was in that profession and a 0 meaning they were not. The data was then split into training and testing datasets, 75% and 25% respectively. For the neural network model age, family size, and work experience were scaled to be a number between 0 and 1. This helps the neural network converge to a solution quicker.

Correlation - no variables to remove

- Briefly describe all 5 algorithms and how you will use them to solve your problem. (abhilash)

-> Here the 5 algorithms are designed in a way to predict the Spending Score of the consumers listed in the dataset collected by the automobile company. With the help of the spending score the company would segment the market which would help them design their products and price them in accordance to the targeted demographic.

The following are the models/algorithms used in predicting the spending score.

1)Logistic Regression:

This model uses 1 or 0 indicator in the consumer dataset, which indicates what the spending score of the customer could be either high or low.

In our dataset we have 6822 rows (each representing a unique customer) with 7 columns: 6 features, 1 target feature (Spending Score). The data is composed of both numerical and categorical features, so we have addressed each of the data types respectively.

Numeric Features: Age, Work Experience, Family Size

Boolean/Logical Features: Graduation, Spending Score

Categorical: Gender, Profession

We're trying to predict whether the spending score for a given customer would be high or low, we take into consideration the difference between training data and predicted training data, along with the actual test data and predicted test data sample.

A confusion matrix plays an important role in the logistic regression model; this is an extremely strong method of evaluating the performance of our classifier. A confusion matrix is a visual representation which tells us the degree of four important classification metrics the following determine the confusion matrix.

1) True Positives : The number of records where the algorithm predicted the customer would have high spending score (1), and they did have a high spending score (1)

2) True Negatives : The number of records where the algorithm predicted the customer would have a low spending score (0), and they did have a low spending score (0).

3) **False Positives :** The number of records where the algorithm predicted the customer will have a high spending score (1), but in real life they had a low spending score (0).

4) **False Negatives:** The number of records where the algorithm predicted the customer will have a low spending score (0), but in real life they had a high spending score (1).

The below is the Confusion Matrix we observed for the Logistic Regression executed on the dataset of the consumers:

	FALSE	TRUE
FALSE	**865**	**188**
TRUE	**219**	**433**

One axis of the above confusion matrix represents the actual value, while the other will represent the predicted values. A high false negative means that many customers in the dataset would actually be having a low spending score (0) and predicted to be having a high spending score (1). This means that our False Negatives are more important which need major attention. Which in this case is approximately 34%.

2) <u>K-NN:</u>

K nearest neighbors is a machine learning model that stores all available cases in the dataset and classifies them based on distance functions or similarity measures.The K-NN algorithms assigns a class to an unlabeled data point based upon the most common class of existing similar data points present in the dataset.

Our task here is to use this dataset to examine records of customers in the training set and use that information to predict whether the customers in the evaluation set are likely to have a high or low spending score. The k-nearest neighbors method is designed on the basic idea that entities that are similar are likely to have properties that are similar. Hence, assigning a class to new data, we first find k instances of existing data that are as similar as possible (nearest neighbors) to the new data. We use these labels of those nearest neighbors to predict the label of the new data.

The value of k has a significant impact on how data pointers/attributes are classified and the best predictive accuracy can be achieved.To quantify the distance between two data points, the k-nearest neighbors algorithm uses a distance function that works for data with more than two dimensions. This measure is known as Euclidean distance. Depending upon this Euclidean

distance a parameter called “K” determines the number of neighbors to consider for a particular classification.

The value of K is determined as square root of total number of samples present in the data which turned out to be 71, but the accuracy was 76%. So, we tried finding another value of K which will be optimal, such that it will give us highest accuracy. The value of K is determinant in getting the best predictive accuracy values, for our dataset the K value of 5 gave the best predictive accuracy of 80.46% in the determination of the Spending score.

3) Naive Bayes:

Naive Bayes algorithm uses Bayes theorem of probability to predict the class of unknown data sets.This methodology is based on the assumption of independence among predictors. The Naive Bayes classification is based on the assumption that the presence of a particular feature in a given class is unrelated to the presence of any other feature in that class. We know that in reality unrelated features are not always realistic, but the assumption helps simplify our model.

4) Decision Trees:

The decision tree algorithm is used to visually and explicitly represent decisions and decision making by branching out various parameters of the dataset. It is a methodology that breaks down the given data types to the granular levels in determining the accuracy of the target/dependent feature.

The complexity parameter or cp decides the size of the tree, it basically decides the splits in the decision tree. We are going to test 5 different complexity measures to find the highest accuracy, 0.01, 0.007, 0.002, 0.0015, 0.001.

For our decision tree model, the target feature of dataset which is the spending score is determined by branching the tree initially through the Age attribute of the dataset wherein, customers with an age greater than or equal to that of 34 are considered and for customers who fulfill this conditionality are classified as the ones with high spending score. This branch is further split in accordance to their family size with a condition of the family size being greater than or equal to two members. For such customers fulfilling the condition the spending score is assigned to be as high and for the one’s not satisfying the split condition are assigned a low spending score. The complexity parameter determines the split and it can be tested for each of the independent attributes determining the target feature.

Through this model a predictive accuracy regarding the spending score of the customers comes around 81%. The complexity parameters used in determining the accuracy are 0.01, 0.007 & 0.002.

5) Neural Networks:

The Neural Network algorithm makes use of training and testing data samples of the original customer dataset in determining the spending score of the customers. The input variables are scaled here the age, work-experience and family size are scaled from 0 TO 1. The neural network model is generated using the logistic function to smoothen the results with SpendingScore as the output.

Neural networks work on the basis of hidden layers and weights. With every iteration in the hidden layer we allocate weights on the variable based on its significance. Now these weights are carried on and used in another hidden layer which will define its own set of weights, the process carries on and the cumulative weights are used in final result calculation. In our current dataset we have used 3 hidden layers which are giving the accuracy of 82.17%.

● Compare the results from all 5 models. Comment on the impact of each independent variable on the dependent variable and how it compared with your hypothesis.

Logistic regression

```
Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)             -5.425735   0.283109 -19.165  < 2e-16 ***
MaleTRUE                 0.160173   0.071714   2.233 0.025517 *
Age                      0.052270   0.002987  17.498  < 2e-16 ***
GraduatedTRUE            0.416162   0.079657   5.224 1.75e-07 ***
ProfessionLawyer         0.560922   0.254357   2.205 0.027436 *
ProfessionHomemaker      1.578093   0.291259   5.418 6.02e-08 ***
ProfessionHealthcare    -0.711404   0.254184  -2.799 0.005130 **
ProfessionExecutive      2.532464   0.254992   9.932  < 2e-16 ***
ProfessionEntertainment  0.929499   0.231056   4.023 5.75e-05 ***
ProfessionEngineer       1.159094   0.236292   4.905 9.33e-07 ***
ProfessionDoctor         0.847979   0.240119   3.531 0.000413 ***
ProfessionArtist         1.472737   0.221055   6.662 2.70e-11 ***
WorkExperience           0.010335   0.010477   0.986 0.323920
FamilySize               0.453535   0.025831  17.558  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The Logistic regression is the first model we used on the consumer dataset. The accuracy of the model was 76.12%. The model has a false positive rate of 18% and a false negative rate of 34%.

A benefit to logistic regression is the ability to evaluate the coefficients of each of out independent variables to our dependent variable. Looking at the coefficients, we found all our independent variables to be significant except for work experience (Appendix). This means work experience did not have a significant impact on our model, which is the opposite of what we predicted in our hypothesis. We think the very low averages for work experience across the industries that we found in our queries explains this phenomenon. If we had accurate data on

work experience we think there would have been a significant and direct relationship with spending score.

All of our other independent variables have a positive coefficient except the Healthcare Profession. This suggests that all these variables have a direct relationship with spending score. Most of these align with what we predicted in our hypothesis except for gender, family size and healthcare. We thought gender would have no impact, but were shocked to find that being a male increases the chances of having a “High” spending score. For family size, our original prediction was an indirect relationship with spending score. The direct relationship could be explained if customers have to buy bigger, more expensive cars for their family. We were surprised to find healthcare the only factor that has an indirect relationship with spending score, but we still see different professions have a different size impact on determining spending score.

K-Nearest Neighbor

The data set is initially split into training and testing types, the K-Nearest Neighbor model is generated through the training data set, with the value of K = 71. 71 is selected because it is the nearest odd number to the square root of the records in the training dataset. This gave us an accuracy of 77.36% for the model.

After running a “for loop” to maximize accuracy, we found the optimal k value to be 5 clusters. 5 clusters is way smaller than our original 71, but we think this is explained by how noisy our data is. In addition, the data had very similar records. This made finding the nearest nerighbot more difficult with such close values as there were many ties between the classifications.

A tie can occur when two or more points are equidistant from an unclassified observation, thereby making it difficult to choose which neighbors are included. We had to use the odd numbers of k to classify the dataset.

One advantage of KNN is that it needs no training period. KNN is called a Lazy Learner, instance based learning. That means the model only stores the dataset and learns from it at the time of making real time predictions. This makes the algorithm extremely fast when it comes to predicting new data.

Naive Bayes

The accuracy of this model is the lowest at 73.55%. The false positive rate is 24% and the false negative rate is 30%. When looking at the individual predicted spending scores compared to the actual testing spending scores it is obvious that profession had the largest impact on deciding if a customer had a high or low spending score. It is hard to tell the relationships between the other independent variables to the spending score.

Decision Tree

In the Decision Tree model, with 5 different levels of complexity parameters(0.01 to 0.001), the trees all provide very similar results and accuracies(0.806 to 0.82). In all of the models, the trees think age(age >=34) is the most important predictor in all cases. Family size (if family size >=2) is the second important factor. The sequence for the rest of the predictor variables is profession, gender, work experience and graduated(not shown most of the time).

Compared with our hypothesis on the relationships, age is directly related to the spending score, and we were correct on this. But profession seems less important than those two, which only shows when we move the cp value from 0.01 to smaller. For graduates, we thought it would have a positive relationship with the dependent variable, but we were wrong on this, the model thinks this is an unnecessary variable(only never appears in any models). For work experience, it is positively related to the spending score, but only appears a few times in models with cp values less than 0.002. For gender, our hypothesis was correct that gender does not have a strong relationship with the dependent variable.

Neural network

The accuracy for neural networks was the highest between all the models at 82.17. Since neural networks often outperform the traditional models it is expected to have high accuracy for this particular model. Because they have the advantages of non-linearity, variable interactions, and customization.

Weights are assigned to all the attributes present in our dataset. According to it we generate hidden layers and can classify the final output. Also, the model trains itself from the data, which has a known outcome and then further optimizes the weights in the second layer. There is a limit to make layers to prevent overfitting by the model. In this scenario we have used only 3 hidden layers and a logistic model for classification as we are dealing with binary outcomes.

- Given your understanding of these models, which one would be the best choice for your use case (understanding that the model with the highest predictive accuracy doesn't necessarily mean it is going to be the best choice)?

->

The best model of our choice for this use case is the Neural Network model. We base this on the fact that this model gave the highest accuracy in comparison to the other four models. Also, the confusion matrix of this algorithm had the lowest false negative rate. In our case the automobile company would not like to miss out on a high spending customer.

However, we would like to give a disclaimer of the underlying conditions of the model's behavior in the hidden layers, this has a set of complexity which may be hard to fully comprehend.

Given your model results, provide 3 recommendations to the company/organization/entity in your context. (Ryan)

Being able to predict the spending score of customers can really benefit automobile companies. We recommend companies market different lines of cars based on a customer's spending score. This targeted marketing campaign will increase the company's effectiveness of advertising per dollar spent by promoting sales. Marketing could include exclusive discounts or financing options for a customer.

With customer spending scores, an automobile company will be able to do even more data analysis. By collecting additional data, such as where a customer lives, will allow the company to create customer segmentations.The last recommendation is to improve the accuracy of the model by collecting more granular data. For example, the company should collect a customer's specific role instead of their profession.

-add sentence

- Address data mining ethics based on the issues that were introduced in the Week 13 video set. Are there any unintended consequences of any decisions you would recommend to the company/organization/entity? (Abhilash)

-> **Data Privacy:**

Data Privacy is sacrosanct to any consumer-based company, especially when it makes decisions using algorithms and predictive analytics. Having said that, in our case the dataset collected by the automobile company contains several attributes that fall under the ambit of the personal data space of the consumer's/user's. For instance, the dataset contains information related to the profession, years of experience, and family size of the probable consumers/users. These are sensitive information that need to be protected and secured by the company making use of this data in determining the range of products/services it proposes to design in accordance with the market segmentation it wants to achieve.

Methodology:

Anonymous personalization of user data is the best way to handle the privacy challenges, this helps segmenting potential customer base without first knowing/identifying who they are. For algorithms designed for customer segmentation or to improve product profitability, organizations need to analyze large volumes of data to establish trends, but they do not need to be able to identify an individual within this analysis. Hence, anonymizing data can ensure privacy protection while still providing valuable results. Achieving equilibrium between the desire to maximizing predictive accuracy through algorithms and the demand for protecting privacy will often require an iterative process as organizations work toward a culture of privacy and espouse the essence of privacy regulations, that is, putting consumers first.

However, it is important to exercise caution so that companies don't turn data privacy and integrity compliance into a competitive advantage by focusing on data quality not quantity, therefore building consumer trust, by the virtue of utilizing privacy as a differentiator can establish brand loyalty and preference among the market segment that the automobile company wants to onboard for delivering its products and services.

Accurateness/Efficiency of the research and removal of bias:

The spending score which is being determined that intends to classify the potential customers into the category of wealthy and non-wealthy cannot be justified through a binary classification. Rather the spending score can be best justified based on a range. Especially, when the automobile company which wants to manufacture and sell vehicles should be able to manufacture them for each class of the market segment. Else it may end up manufacturing vehicles catering to a certain section/class of the market segment which may or may not end up buying the product. Likewise, spending patterns may or may not predict the wealth or likelihood of consumers buying an automobile. There are various random factors that may come into play. The model or algorithms may end up predicting a low spending score for a set of customers who in turn may be spending more and vice versa.

Thereby algorithms can have many inconsistencies, for instance we have chosen neural networks as our primary model, however the algorithms transparency regarding the customer classification is not clear. This anomaly creates a doubt on the correctness of the algorithm and its training which may or may not be biased. In the dataset we use attributes like gender and age so the possibility of an existing bias in the training set cannot be completely ruled out.

Keeping this in mind the algorithms need to be trained well by going into the granular level of datasets and collection of patterns and attributes that may even remotely have any impact on the nature of spending of any given consumer.

Given the analysis carried out through the running of various algorithmic models on the dataset collected by the automobile company, attributes based on gender and age can lead to bias if the initial training model is itself biased. This is clearly visible in the Decision Tree model's main split based on age.

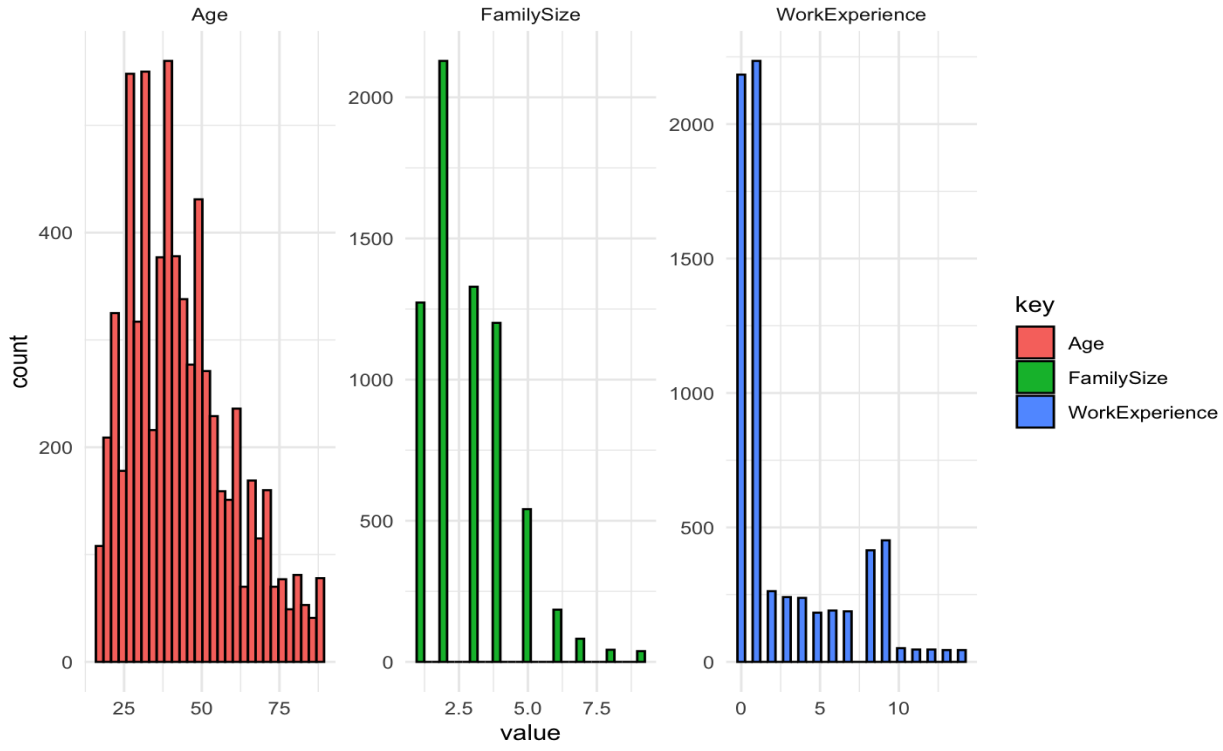
- A bulleted list of your team members along with (1) what contributions they made to the group project and (2) the percentage of the work they completed. The total percentage of all group members should add up to 100%. For example: Jolene Switzer found the dataset, wrote the code for the data preprocessing, and built the slide presentation. Her contribution was 30%....

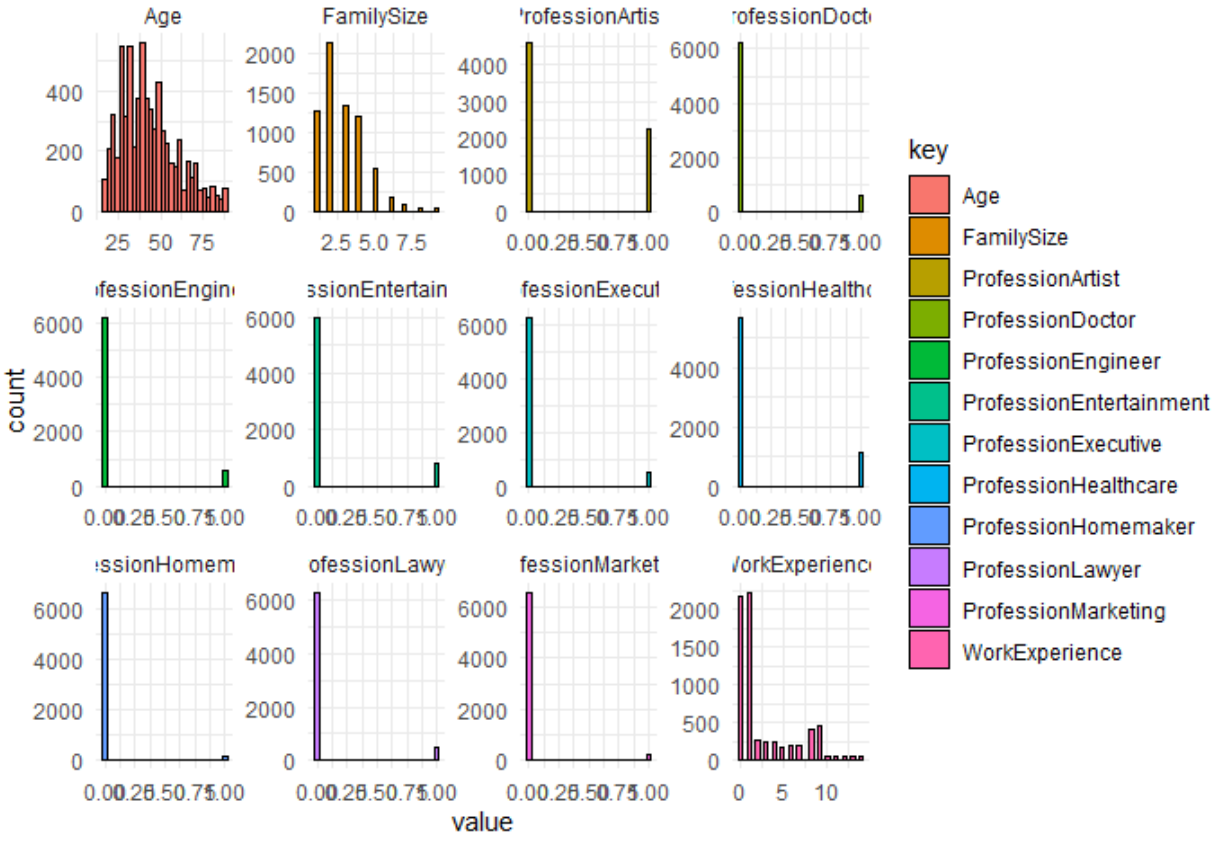
Contributions:-

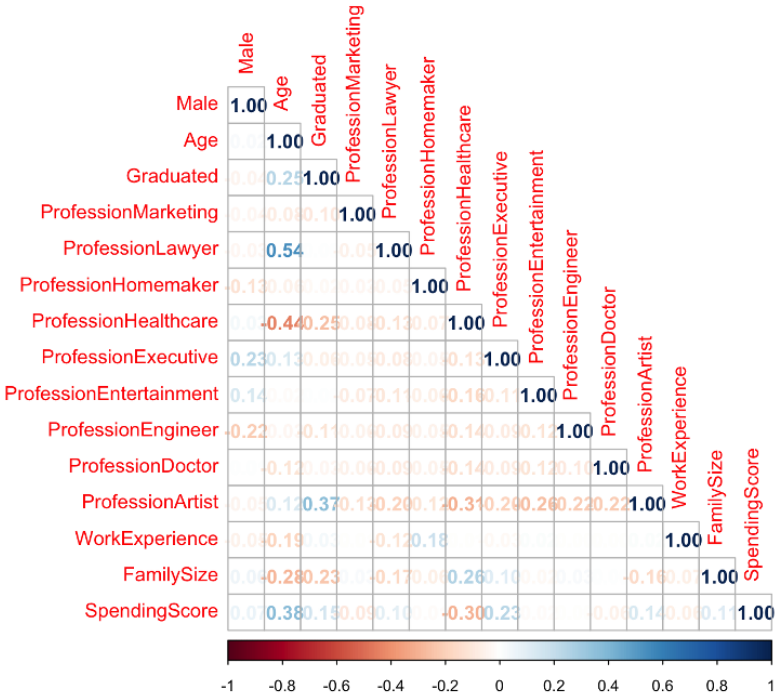
- Ryan Cramer: Worked on paper, cleaned train-set.csv in excel. Did Naive bayes algorithm in R. Total contribution was 25%.
- Shivani Dandir : Worked on paper, scrapped data from Indeed.com and cleaned it, did KNN and Neural Network. Total Contribution was 25 %

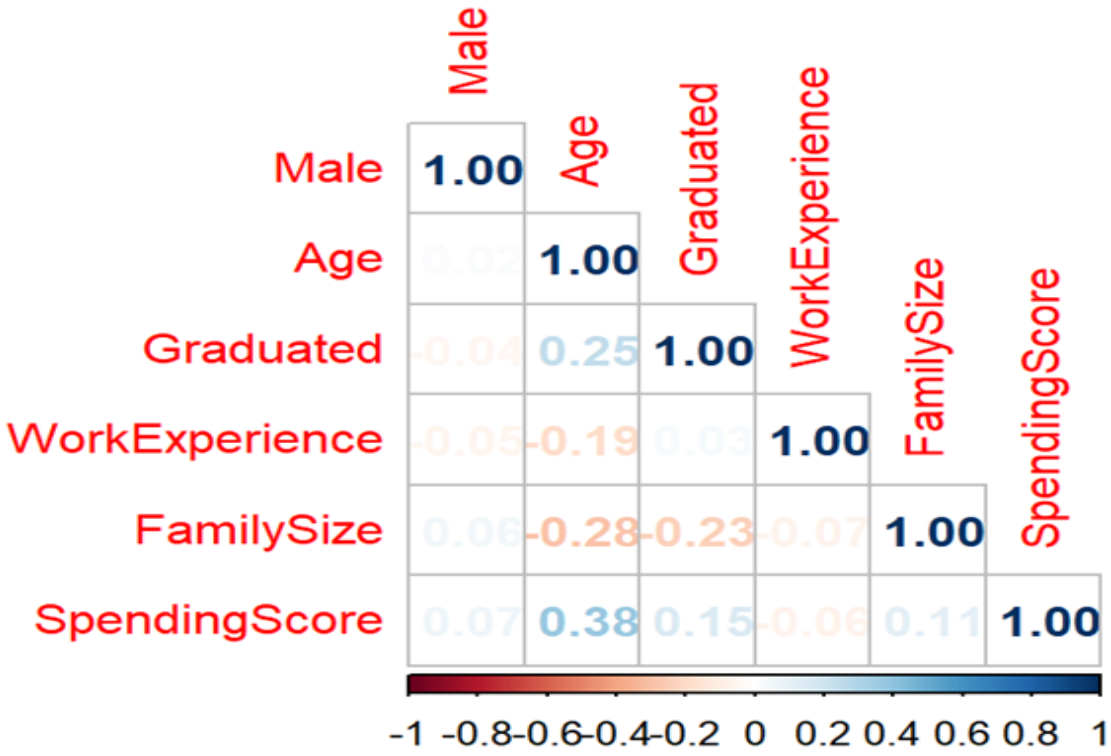
- Jianbo Gong: Found the dataset, developed the decision tree model, worked on paper, Total contribution was 25%.
- Abhilash Annasaheb Halappanavar: Worked on the compilation of the project report and ran the logistic regression model on the dataset. Total contribution was 25%.
- An appendix containing (there is no page limit for the appendix):
  - All R source code, presented in a fixed-width font (resized to ensure no line extends down to the next line)
  - Model results screenshots from R for each of the 5 models
  - The following visualizations:
    - Histograms of all continuous variables
    - Any appropriate scatterplots or boxplots
    - Correlation plot
    - Decision tree plots
    - Neural network plot
    - Confusion matrix for each algorithm
    - Any other visualizations that you deem appropriate

SpendingScore




Age
FamilySize
rofessionArtis
ofessionDoct
fessionEngin
ssionEntertain
fessionExecut
essionHealth
essionHomem
ofessionLawy
fessionMarket
/orkExperienc
count
value
key
Age
FamilySize
ProfessionArtist
ProfessionDoctor
ProfessionEngineer
ProfessionEntertainment
ProfessionExecutive
ProfessionHealthcare
ProfessionHomemaker
ProfessionLawyer
ProfessionMarketing
WorkExperience


Male
Age
Graduated
ProfessionMarketing
ProfessionLawyer
ProfessionHomemaker
ProfessionHealthcare
ProfessionExecutive
ProfessionEntertainment
ProfessionEngineer
ProfessionDoctor
ProfessionArtist
WorkExperience
FamilySize
SpendingScore
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1



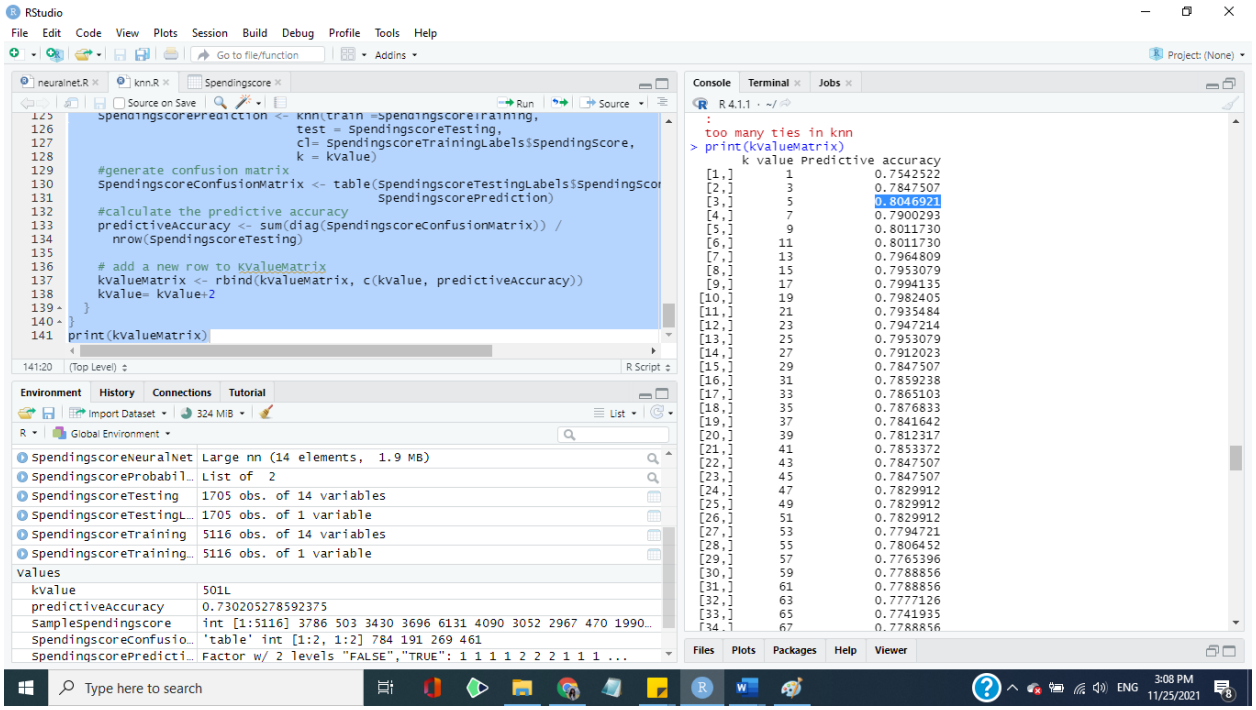

```
# KNN model
library(tidyverse)#loading tidyverse
library(class)
library(dummies)
library(e1071)#loading tidyverse
library(corrplot)

Spendingscore1 <-
read_csv("C:/Users/INDIA02/Documents/University_documents/545/Project/ProjectDa
taClean.csv",col_types="lilfiil")# reading csv

print(Spendingscore1) #displaying tibble
head(Spendingscore1, 20) #displaying 20 rows
str(Spendingscore1) #structure of tibble
summary(Spendingscore1) #summary of tibble

#Separate the tibble into two. One with just the label and one with the other
#variables. Note: after this step, you should have two tibbles:
SpendingscoreDF <- data.frame(Spendingscore1)
Spendingscore <- as_tibble(dummy.data.frame(data =SpendingscoreDF,
names="Profession"))
view(Spendingscore)
```

```
# Displays a correlation matrix rounded to 2 decimal places
round(cor(Spendingscore), 2)

# Display a correlation plot  with the 'number' method and only
# displaying the bottom left section
corrplot(cor(Spendingscore), method = "number",type = "lower")
#this is the class we are looking to predict - y variable
SpendingscoreLabels <- Spendingscore %>% select(SpendingScore)

#removing Sedan Size from x variables
Spendingscore <- Spendingscore %>% select(-SpendingScore)
#seed variable is set to ensure we get the same result every time we run a
#sampling process
set.seed(100)
SampleSpendingscore<- sample(nrow(Spendingscore),
round(nrow(Spendingscore) * 0.75), replace = FALSE)

#Creating training and testing data
SpendingscoreTraining<- Spendingscore[SampleSpendingscore, ]
SpendingscoreTrainingLabels<-SpendingscoreLabels[SampleSpendingscore, ]

SpendingscoreTesting<- Spendingscore[-SampleSpendingscore, ]
SpendingscoreTestingLabels<- SpendingscoreLabels[-SampleSpendingscore, ]

#generate the K-nn neighbor model
#We are using 71 as it is the nearest odd number to the square root of records
#in the training dataset.
SpendingscorePrediction <- knn(train =SpendingscoreTraining,
test = SpendingscoreTesting,cl= SpendingscoreTrainingLabels$SpendingScore,k
=71)

#print the predicted data
print(SpendingscorePrediction)
print(summary(SpendingscorePrediction))

#creating confusion matrix
SpendingscoreConfusionMatrix <- table
(SpendingscoreTestingLabels$SpendingScore,SpendingscorePrediction)
print(SpendingscoreConfusionMatrix)

#false positive rate
SpendingscoreConfusionMatrix[1,2] /(SpendingscoreConfusionMatrix[1,2]+
SpendingscoreConfusionMatrix[1,1] )

#false negative
SpendingscoreConfusionMatrix[2,1] /(SpendingscoreConfusionMatrix[2,1]+
SpendingscoreConfusionMatrix[2,2] )

#accuracy
```

```
predictiveAccuracy <- sum(diag(SpendingscoreConfusionMatrix)) /
nrow(SpendingscoreTesting)
print(predictiveAccuracy)

#Create a matrix of k-values with their predictive accuracy
kValueMatrix <- matrix(data= NA,nrow=0,ncol=2)
colnames(kValueMatrix) <- c("k value","Predictive accuracy")

for (kValue in 1:nrow(SpendingscoreTraining)){
  if(kValue %% 2 !=0){
      #generate the model
    SpendingscorePrediction <- knn(train =SpendingscoreTraining,
test=SpendingscoreTesting,cl=SpendingscoreTrainingLabels$SpendingScore,k =
kValue)
      #generate confusion matrix
SpendingscoreConfusionMatrix <- table
(SpendingscoreTestingLabels$SpendingScore,SpendingscorePrediction)
      #calculate the predictive accuracy
      predictiveAccuracy <- sum(diag(SpendingscoreConfusionMatrix)) /
      nrow(SpendingscoreTesting)

      # add a new row to KValueMatrix
      kValueMatrix <- rbind(kValueMatrix, c(kValue,predictiveAccuracy))
      kValue= kValue+2
  }
}
print(kValueMatrix)
```

Accuracy -0.773607 when k=71

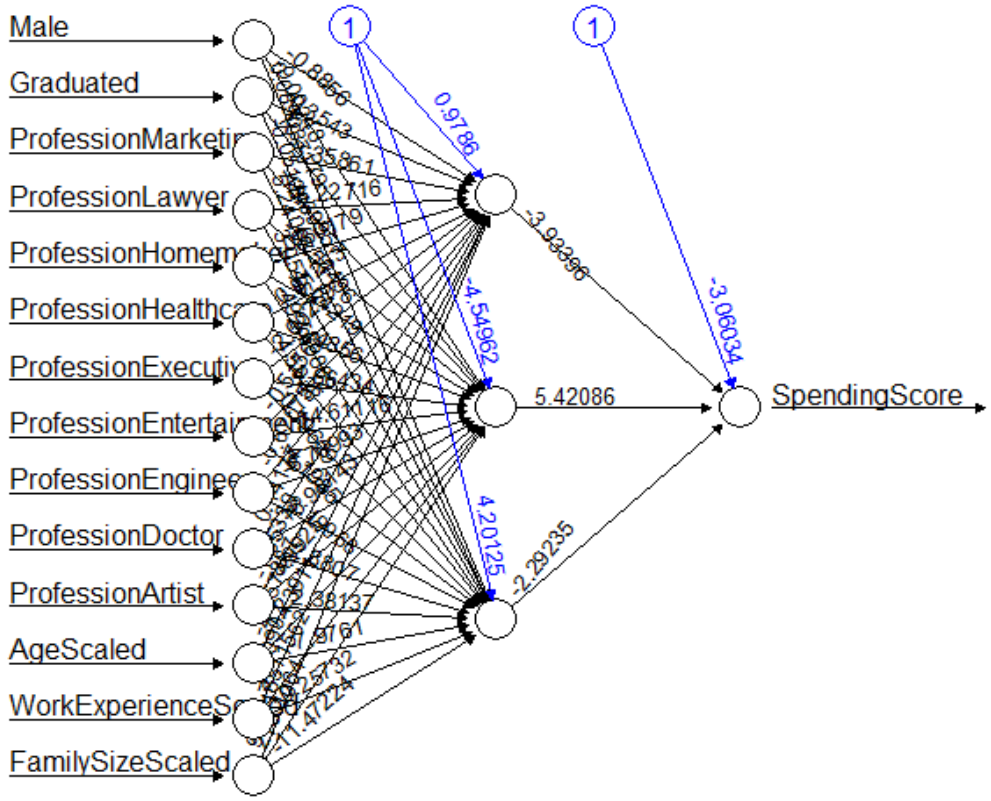
Optimal value of k =5 with accuracy 0.8046921

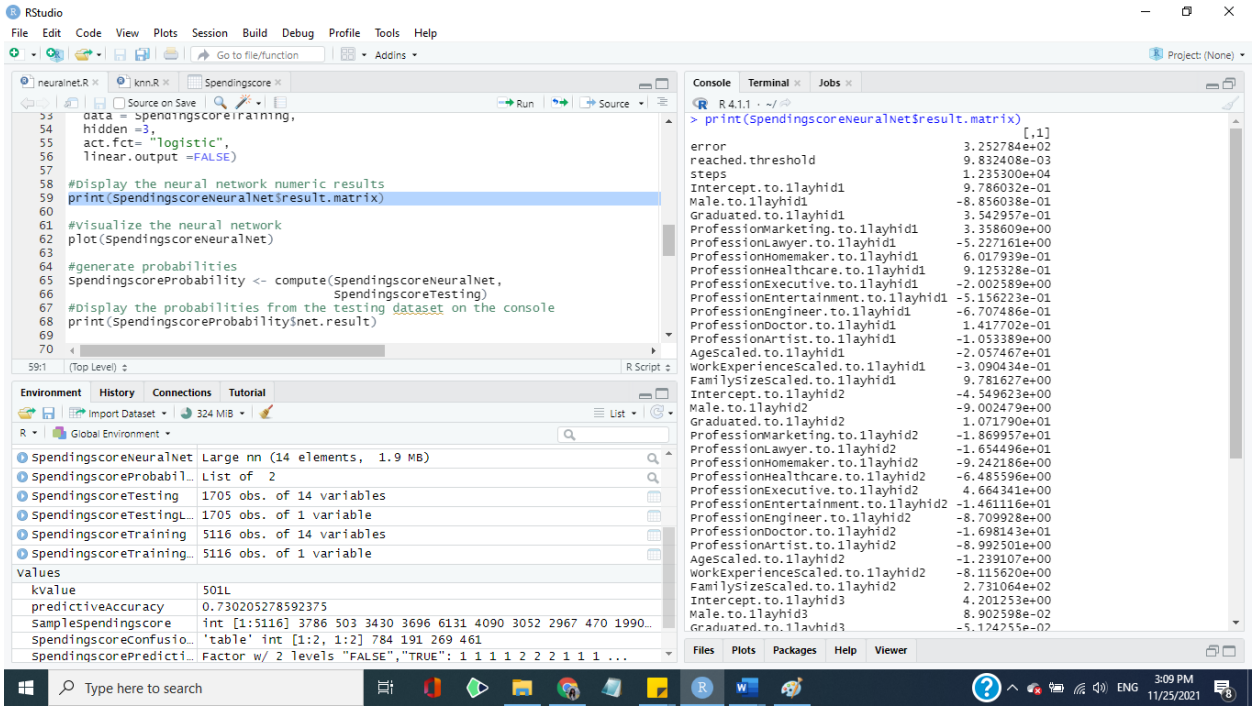



```
#generate neural network model

#install.packages("tidyverse")
#install.packages("neuralnet")
library(tidyverse)#loading tidyverse
library(neuralnet)#loading neuralnet
library(dummies)

Spendingscore <-
read_csv("C:/Users/INDIA02/Documents/University_documents/545/Project/ProjectDa
taClean.csv",col_types="lilfiil")# reading csv
print(Spendingscore) #displaying tibble
head(Spendingscore, 20) #displaying 20 rows
str(Spendingscore) #structure of tibble
summary(Spendingscore) #summary of tibble
SpendingscoreDF <- data.frame(Spendingscore)
Spendingscore <- as_tibble(dummy.data.frame(data =SpendingscoreDF,
names="Profession"))
view(Spendingscore)

#Scaling the variables from 0 to 1
Spendingscore <- Spendingscore %>%
mutate(AgeScaled = (Age - min(Age))/(max(Age)-min(Age)))

Spendingscore <- Spendingscore %>%
mutate(WorkExperienceScaled = (WorkExperience - min(WorkExperience))/
```

```
(max(WorkExperience)-min(WorkExperience)))

Spendingscore <- Spendingscore %>%
mutate(FamilySizeScaled = (FamilySize - min(FamilySize))/
(max(FamilySize)-min(FamilySize)))

Spendingscore <- Spendingscore %>% select(-Age)
Spendingscore <- Spendingscore %>% select(-WorkExperience)
Spendingscore <- Spendingscore %>% select(-FamilySize)

#seed variable is set to ensure we get the same result every time we run a
#sampling process
set.seed(100)
SampleSpendingscore<- sample(nrow(Spendingscore),
round(nrow(Spendingscore) * 0.75),replace = FALSE)
#Creating training and testing data
SpendingscoreTraining<- Spendingscore[SampleSpendingscore, ]
SpendingscoreTesting<- Spendingscore[-SampleSpendingscore, ]

#Generate the neural network model to predict
SpendingscoreNeuralNet <- neuralnet(formula = SpendingScore ~ .,
data = SpendingscoreTraining,hidden =3,act.fct= "logistic",
linear.output =FALSE)

#Display the neural network numeric results
print(SpendingscoreNeuralNet$result.matrix)
#Visualize the neural network
plot(SpendingscoreNeuralNet)
#generate probabilities
SpendingscoreProbability <- compute(SpendingscoreNeuralNet,
SpendingscoreTesting)
#Display the probabilities from the testing dataset on the console
print(SpendingscoreProbability$net.result)

#Convert probability predictions into 0/1 predictions
SpendingscorePrediction <-ifelse(SpendingscoreProbability$net.result> 0.5,1,0)
#Display the 0/1 predictions on the console
print(SpendingscorePrediction)

#Evaluate the model by forming a confusion matrix
SpendingscoreConfusionMatrix<-  table(SpendingscoreTesting$SpendingScore,
SpendingscorePrediction )

#Display the confusion matrix on the console
print(SpendingscoreConfusionMatrix)
#Calculate the model predictive accuracy
predictiveAccuracy <- sum(diag(SpendingscoreConfusionMatrix))/
nrow(SpendingscoreTesting)
#Display the predictive accuracy on the console
```

```
print(predictiveAccuracy)

Confusion matrix
        SpendingscorePrediction
          0   1
  FALSE 825 228
  TRUE   76 576

0.8217009 - accuracy

Neural Network Graph
```





Decision Tree:

```
# Jianbo Gong
# MIS545 Final Project
# Decision tree model on the vehicle customer dataset
# use decision tree to predict customer's spending score(1 for high, 0 for low)
# so that we can advertise different models to the corresponding customers.

library(tidyverse)
library(rpart)
library(rpart.plot)

# set working directory to the project folder
setwd("/Users/jianbo/Documents/UA/MIS Master/2021 Fall/MIS545/Project")

# read in training and testing data
customerData <- read_csv(file = "ProjectDataClean.csv",
                 col_types = "lilfiil",
                 col_names = TRUE)
customerData <- customerData %>%
        mutate(SpendingScore = ifelse(SpendingScore == TRUE, "High", "Low"))

set.seed(100)
sampleSet <- sample(nrow(customerData),
                    nrow(customerData) * 0.75,
                    replace = FALSE)
```

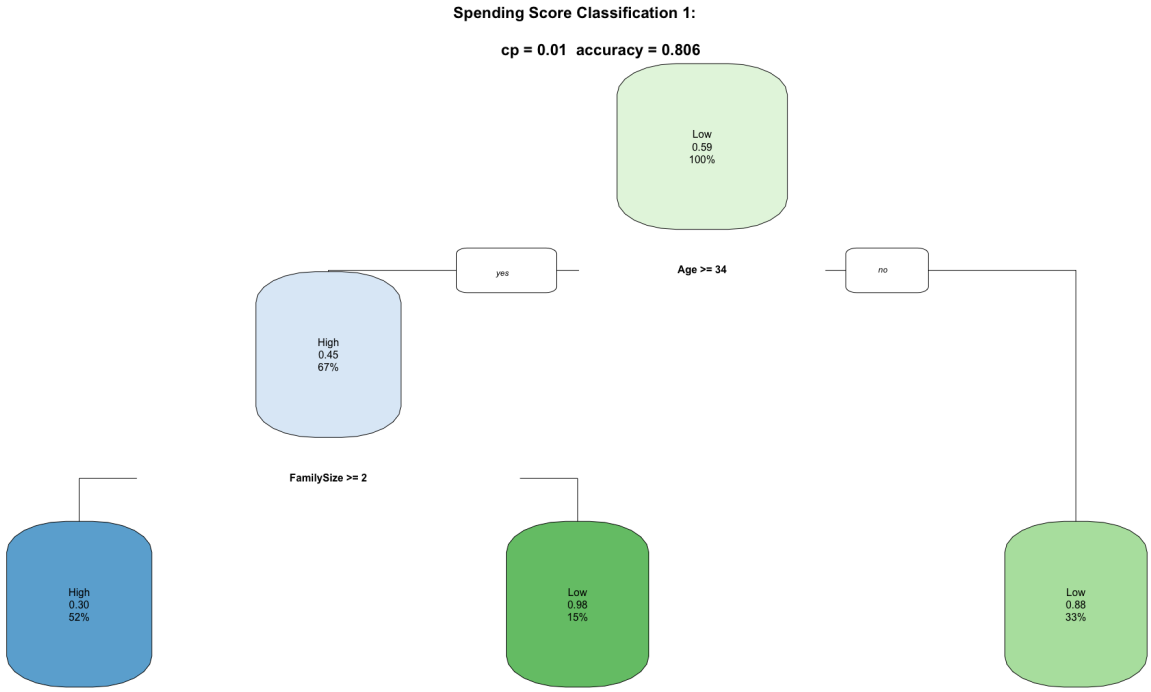
```
customerTraining <- customerData[sampleSet,]
customerTesting <- customerData[-sampleSet,]

# show summaries of the data
summary(customerTraining)
summary(customerTesting)

# run three interesting queries on the datasets
# 1/3 show mean age of different profession groups
print(customerTraining %>%
        group_by(Profession) %>%
        summarize(mean(Age)))

# 2/3 show work experience status for different gender groups
# (True for male, False for female; True for graduated, False for not graduated)
print(customerTraining %>%
        group_by(Male) %>%
        count(Graduated))

# 3/3 count group high spending score numbers
print(customerTraining %>%
        group_by(Profession) %>%
        filter(SpendingScore == "High")%>%
        count())

# build the first tree model using cp 0.01
tree1 <- rpart(data = customerTraining,
               formula = SpendingScore ~.,
               cp = 0.01,
               method = "class")
rpart.plot(tree1,main = "Spending Score Classification 1: \n
           cp = 0.01  accuracy = 0.806",cex.main = 1.5)

tree2 <- rpart(data = customerTraining,
               formula = SpendingScore ~.,
               cp = 0.007,
               method = "class")

rpart.plot(tree2,main = "Spending Score Classification 2: \n
           cp = 0.007  accuracy = 0.819",cex.main = 1.6)

tree3 <- rpart(data = customerTraining,
                        formula = SpendingScore ~.,
                        cp = 0.002,
                        method = "class")

rpart.plot(tree3,main = "Spending Score Classification 3: \n
           cp = 0.002  accuracy = 0.815",cex.main = 1.6)

tree4 <- rpart(data = customerTraining,
               formula = SpendingScore ~.,
               cp = 0.0015,
               method = "class")
```

```
rpart.plot(tree4,main = "Spending Score Classification 4: \n
           cp = 0.0015  accuracy = 0.815",cex.main = 1.6)


tree5 <- rpart(data = customerTraining,
               formula = SpendingScore ~.,
               cp = 0.001,
               method = "class")
vit
rpart.plot(tree5,main = "Spending Score Classification 5: \n
           cp = 0.001  accuracy = 0.819",cex.main = 1.6)

# prediction using tree1 model and show accuracy
prediction1 <- predict(tree1,customerTesting,type = "class")
print(prediction1)
tree1ConfusionMatrix <- table(customerTesting$SpendingScore,
                              prediction1)
print(tree1ConfusionMatrix)
accuracy1 <- sum(diag(tree1ConfusionMatrix))/nrow(customerTesting)
print(accuracy1)

# prediction using tree2 model and show accuracy
prediction2 <- predict(tree2,customerTesting,type = "class")
print(prediction2)
tree2ConfusionMatrix <- table(customerTesting$SpendingScore,
                              prediction2)
print(tree2ConfusionMatrix)
accuracy2 <- sum(diag(tree2ConfusionMatrix))/nrow(customerTesting)
print(accuracy2)

# prediction using tree3 model and show accuracy
prediction3 <- predict(tree3,customerTesting,type = "class")
print(prediction3)
tree3ConfusionMatrix <- table(customerTesting$SpendingScore,
                              prediction3)
print(tree3ConfusionMatrix)
accuracy3 <- sum(diag(tree3ConfusionMatrix))/nrow(customerTesting)
print(accuracy3)

# prediction using tree4 model and show accuracy
prediction4 <- predict(tree4,customerTesting,type = "class")
print(prediction4)
tree4ConfusionMatrix <- table(customerTesting$SpendingScore,
                              prediction4)
print(tree4ConfusionMatrix)
accuracy4 <- sum(diag(tree4ConfusionMatrix))/nrow(customerTesting)
print(accuracy4)

# prediction using tree5 model and show accuracy
prediction5 <- predict(tree5,customerTesting,type = "class")
print(prediction5)
tree5ConfusionMatrix <- table(customerTesting$SpendingScore,
                              prediction5)
print(tree5ConfusionMatrix)
accuracy5 <- sum(diag(tree5ConfusionMatrix))/nrow(customerTesting)
```

```
print(accuracy5)
```

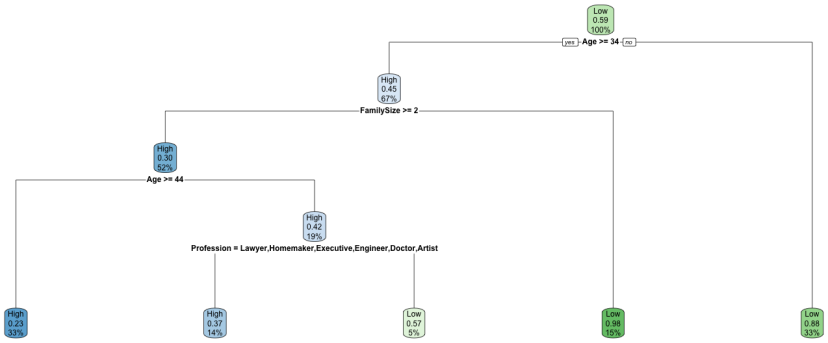
```
R 4.1.1 · ~/Documents/UA/MIS Master/2021 Fall/MIS545/Project/
> # run three interesting queries on the datasets
> # 1/3 show mean age of different profession groups
> print(customerTraining %>%
+         group_by(Profession) %>%
+         summarize(mean(Age)))
# A tibble: 9 × 2
  Profession    `mean(Age)`
  <fct>               <dbl>
1 Marketing            38.0
2 Lawyer               75.0
3 Homemaker            36.9
4 Healthcare           26.8
5 Executive            51.5
6 Entertainment        43.0
7 Engineer             42.1
8 Doctor               37.4
9 Artist               46.4
> # 2/3 show work experience status for different gender groups
> # (True for male, False for female; True for graduated, False for not graduated)
> print(customerTraining %>%
+         group_by(Male) %>%
+         count(Graduated))
# A tibble: 4 × 3
# Groups:   Male [2]
  Male  Graduated     n
  <lgl> <lgl>     <int>
1 FALSE FALSE       813
2 FALSE TRUE       1532
3 TRUE  FALSE      1065
4 TRUE  TRUE       1705
> # 3/3 count group high spending score numbers
> print(customerTraining %>%
+         group_by(Profession) %>%
+         filter(SpendingScore == "High")%>%
+         count())
# A tibble: 9 × 2
# Groups:   Profession [9]
  Profession        n
  <fct>         <int>
1 Marketing        32
2 Lawyer          213
3 Homemaker        52
4 Healthcare       64
5 Executive       318
```

```
> # 3/3 count group high spending score numbers
> print(customerTraining %>%
+         group_by(Profession) %>%
+         filter(SpendingScore == "High")%>%
+         count())
# A tibble: 9 x 2
# Groups:   Profession [9]
  Profession       n
  <fct>        <int>
1 Marketing       32
2 Lawyer         213
3 Homemaker       52
4 Healthcare      64
5 Executive      318
6 Entertainment  230
7 Engineer       183
8 Doctor         131
9 Artist         864
>
```



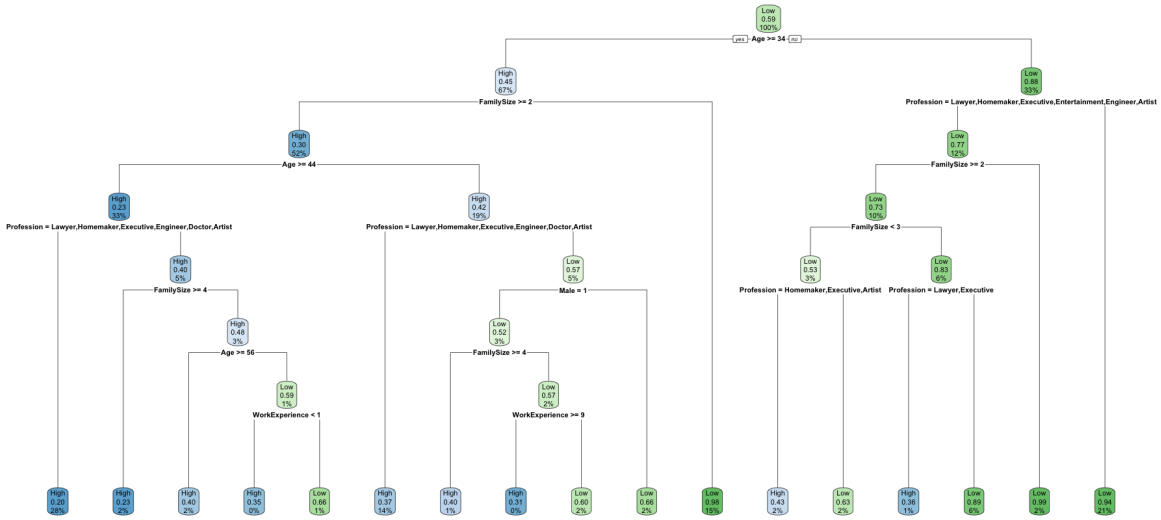
Spending Score Classification 2:

cp = 0.007 accuracy = 0.819



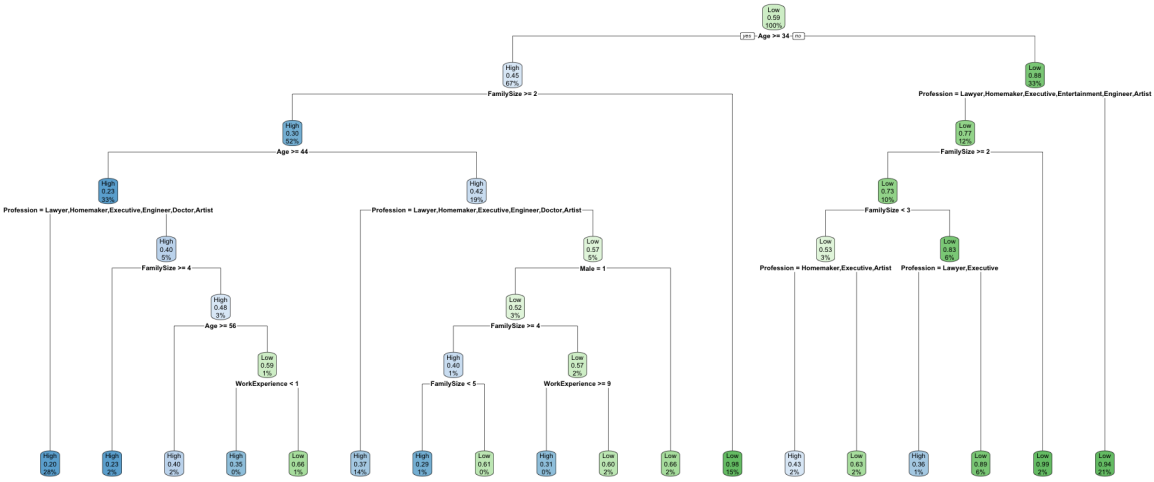
Spending Score Classification 3:

cp = 0.002 accuracy = 0.815



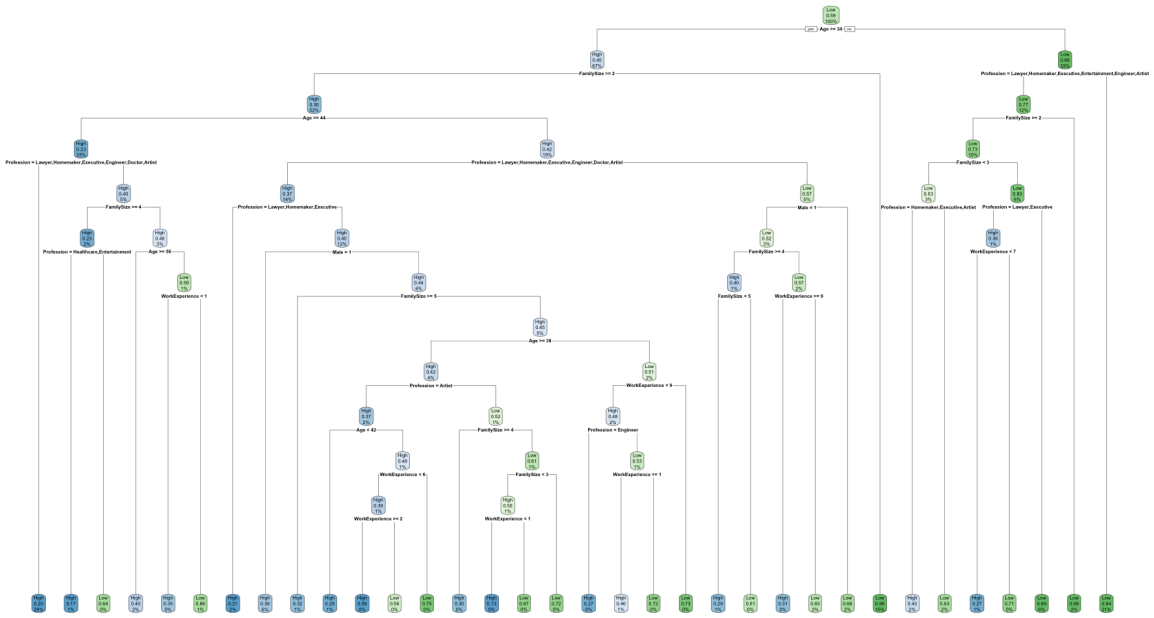
Spending Score Classification 4:

cp = 0.0015 accuracy = 0.815



Spending Score Classification 5:

cp = 0.001 accuracy = 0.819



Logistic Regression:

```
#Abhilash Annasaheb Halappanavar
# MIS545 Final Project
#Installing Various packages as necessary in the generation of #the regression
model
install.packages("tidyverse")
install.packages("dummies")
install.packages("corrplot")
install.packages("olsrr")
install.packages("smotefamily")

# Loading the tidyverse, corrplot, olsrr libraries
library(tidyverse)
library(dummies)
library(scales)
library("corrplot")
library("olsrr")

#Setting up the working directory
setwd("C:/Users/UAL-Laptop/Desktop/MIS545/Project")

# creating and reading the ProjectDataClean csv into a tibble Project
Project <- read_csv("ProjectDataClean.csv",
                        col_types = "lilfiil",
                        col_names = TRUE)
#Displaying Project in the console
print (Project)

#Displaying the structure of Project on the console
str(Project)



#Displaying the summary of Project in the console
summary(Project)

#Creating a displayAllHistograms() function that will take in a tibble
demonstration parameter that will display a histogram for all numeric features
displayAllHistograms <- function(tibbleDataset) {
  tibbleDataset %>%
    keep(is.numeric) %>%
    gather() %>%
    ggplot() +geom_histogram(mapping = aes(x=value, fill=key),
                             color = "black") +
    facet_wrap (~key, scales = "free") +
    theme_minimal ()
```

```
}

#Calling the displayAllHistograms() function, passing in Project as an argument

displayAllHistograms(Project)

# Dummy coding the Profession variable using dummy.data.frame()
Project1 <- data.frame(Project)

Project2 <- as_tibble(dummy.data.frame(data = Project1, names = "Profession"))

#Display the correlation matrix of Project2 rounded to two decimal places
round(cor(Project2), 2)

#Displaying a correlation plot using the "number" method and limit output to
the bottom left
corrplot(cor(Project2),
         method = "number",
         type = "lower")

cor(Project1[, c('Male', 'Age', 'Graduated','WorkExperience','SpendingScore')])
Project2 <- select(Project1, -Profession)

#Randomly splitting the dataset into Project2Training (75% of records) and
Project2Testing (25% of records) using 100 as the random seed

set.seed(100)

sampleSet <- sample(nrow(Project1),
                    round(nrow(Project1) * 0.75),
                    replace = FALSE)
#Putting the records from the 75% sample into Project2Training
Project2Training <- Project1[sampleSet, ]

#Putting the records from the 25% sample into Project2Testing
Project2Testing <- Project1[-sampleSet, ]




# Generating the logistic regression model (using SpendingScore as the binary
dependent variable) and saving it in an object called SpendingScoreModel

SpendingScoreModel <- glm(data = Project2Training,
                          family = binomial,
                          formula = SpendingScore ~.)
```

```
#Displaying the logistic regression model results using the summary() function
summary(SpendingScoreModel)

#Using the model to predict outcomes in the testing dataset
SpendingScorePrediction <-  predict(SpendingScoreModel,
               Project2Testing,type = "response")

#Displaying the prediction model
print(SpendingScorePrediction)

# Treating anything below or equal to 0.5 as TRUE, anything above 0.5 as FALSE

SpendingScorePrediction <- ifelse(SpendingScorePrediction >= 0.5,TRUE,FALSE)
print(SpendingScorePrediction)

#Generating a confusion matrix of predictions

SpendingScoreConfusionMatrix <-
  table(Project2Testing$SpendingScore,
        SpendingScorePrediction)
#Displaying Confusion Matrix
print(SpendingScoreConfusionMatrix)

#Calculating false positive rate
SpendingScoreConfusionMatrix[1,2] /
  (SpendingScoreConfusionMatrix[1,2] +
     SpendingScoreConfusionMatrix[1,1])

#Calculating the false negative rate
SpendingScoreConfusionMatrix[2,1] /
  (SpendingScoreConfusionMatrix[2,1] +
     SpendingScoreConfusionMatrix[2,2])

#Calculating the model prediction accuracy
sum(diag(SpendingScoreConfusionMatrix)) / nrow(Project2Testing)
```

Model Summary:

```
Coefficients:
                          Estimate Std. Error z value
(Intercept)             -3.952997   0.197287 -20.037
MaleTRUE                 0.160173   0.071714   2.233
Age                      0.052270   0.002987  17.498
GraduatedTRUE            0.416162   0.079657   5.224
ProfessionDoctor        -0.624759   0.127147  -4.914
ProfessionEngineer      -0.313644   0.123747  -2.535
ProfessionEntertainment -0.543238   0.107643  -5.047
ProfessionExecutive      1.059727   0.155618   6.810
ProfessionHealthcare    -2.184141   0.159876 -13.661
ProfessionHomemaker      0.105355   0.207239   0.508
ProfessionLawyer        -0.911815   0.149092  -6.116
ProfessionMarketing     -1.472737   0.221055  -6.662
WorkExperience           0.010335   0.010477   0.986
FamilySize               0.453535   0.025831  17.558
                        Pr(>|z|)
```

	Pr(>|z|)	
(Intercept)	< 2e-16	***
MaleTRUE	0.0255	*
Age	< 2e-16	***
GraduatedTRUE	1.75e-07	***
ProfessionDoctor	8.94e-07	***
ProfessionEngineer	0.0113	*
ProfessionEntertainment	4.50e-07	***
ProfessionExecutive	9.77e-12	***
ProfessionHealthcare	< 2e-16	***
ProfessionHomemaker	0.6112	
ProfessionLawyer	9.61e-10	***
ProfessionMarketing	2.70e-11	***
WorkExperience	0.3239	
FamilySize	< 2e-16	***

```
> sum(diag(SpendingScoreConfusionMatrix)) / nrow(Project2Testing)
[1] 0.7612903
```

Naive Bayes

```
# Used to install tidyvere and e1071 packages. Commented out after first use
# install.packages("tidyverse")
# install.packages("e1071")

# Loads in the tidyverse and e1071 libraries
library(tidyverse)
library(e1071)
library(dummies)

# Sets the working directory to the Lab08 folder
setwd("~/Desktop/MIS 545/Project")

# Reads ProjectDataClean.csv into a tibble called SpendingScore
SpendingScore <- read_csv(file = "ProjectDataClean.csv",
                          col_types = "lilfiil",
                          col_names = TRUE)

# Displays dwellingType in the SpendingScore
print(SpendingScore)

# Displays the structure of SpendingScore in the console
str(SpendingScore)

# Display the summary of SpendingScore in the console
summary(SpendingScore)

# dummy
SpendingscoreDF <- data.frame(SpendingScore)
SpendingScore <- as_tibble(dummy.data.frame(data =SpendingscoreDF,
names="Profession"))


view(SpendingScore)

# Creates a function called DisplayAllHistograms that takes in a tibble
paramter
# that will display a histogram for all numeric features in a tibble
displayAllHistograms <- function(SpendingScore) {
  SpendingScore %>%
    keep(is.numeric) %>%
    gather() %>%
    ggplot() + geom_histogram(mapping = aes(x=value,fill=key),
                              color = "black") +
    facet_wrap (~ key, scales = "free") +
    theme_minimal ()
}
```

```
# Call the displayAllHistograms() function, passing in SpendingScore as an
# argument
displayAllHistograms(SpendingScore)

# Displays a correlation matrix of SpendingScore rounded to 2 decimal places
round(cor(SpendingScore), 2)

# Display a correlation plot of SpendingScore with the 'number' method and only
# displaying the bottom left section
corrplot(cor(SpendingScore),
         method = "number",
         type = "lower")


# Sets our random seed to 100 so our result stays the same
set.seed(100)

# Creates a vector of 75% randomly sampled rows from the original dataset.
sampleSet <- sample(nrow(SpendingScore),
                    round(nrow(SpendingScore) * 0.75),
                    replace = FALSE)

# Puts the records from the 75% sample into SpendingScoreTraining
SpendingScoreTraining <- SpendingScore[sampleSet, ]

# Puts all other records, 25%, into SpendingScoreTesting
SpendingScoreTesting <- SpendingScore[-sampleSet, ]

# Generate the Naive Bayes model to predict SpendingScore based on the other
# variables in the dataset
spendingModel <- naiveBayes(formula = SpendingScore ~ .,
                            data = SpendingScoreTraining,
                            laplace = 1)

# Build probabilities for each record in the testing dataset and store them in
# SpendingScoreProbability
SpendingScoreProbability <- predict(spendingModel,
                                    SpendingScoreTesting,
                                    type = "raw")

# Display SpendingScoreProbability on the console
print(SpendingScoreProbability)

# Predict classes for each record in the testing dataset and store them in
# SpendingScorePrediction
SpendingScorePrediction <- predict(spendingModel,
                                   SpendingScoreTesting,
                                   type = "class")
```

```
# Display SpendingScorePrediction on the console
print(SpendingScorePrediction)

# Evaluate the model by forming a confusion matrix
SpendingScoreConfusionMatrix <- table(SpendingScoreTesting$SpendingScore,
                                      SpendingScorePrediction)

# Display the confusion matrix on the console
print(SpendingScoreConfusionMatrix)

# Calculate the model predictive accuracy
predictiveAccuracy <- sum(diag(SpendingScoreConfusionMatrix)) /
  nrow(SpendingScoreTesting)

# Display the predictive accuracy on the console
print(predictiveAccuracy)

#Calculating false positive rate
SpendingScoreConfusionMatrix[1,2] /
  (SpendingScoreConfusionMatrix[1,2] +
     SpendingScoreConfusionMatrix[1,1])

#Calculating the false negative rate
SpendingScoreConfusionMatrix[2,1] /
  (SpendingScoreConfusionMatrix[2,1] +
     SpendingScoreConfusionMatrix[2,2])
```

Confusion Matrix

Logistic Regression Confusion Matrix		
	Score Prediction	
	FALSE	TRUE
FALSE	865	188
TRUE	219	433
Accuracy	76.13%%	
False Positive	17.85%	
False Negative	33.59%	

K-NN Confusion Matrix (k=5)		
	Score Prediction	
	FALSE	TRUE
FALSE	863	190
TRUE	144	508
Accuracy	80.41%	
False Positive	18.04%	
False Negative	22.09%	

Naïve Bayes Confusion Matrix		
	Score Prediction	
	FALSE	TRUE
FALSE	799	254
TRUE	197	455
Accuracy	73.54%	
False Positive	24.12%	
False Negative	30.21%	

Decision Tree Confusion Matrix (cp = 0.007)		
	Score Prediction	
	FALSE	TRUE
FALSE	832	222
TRUE	87	565
Accuracy	81.94%	
False Positive	21.06%	
False Negative	13.34%	

Neural Network Confusion Matrix		
	Score Prediction	
	FALSE	TRUE
FALSE	825	228
TRUE	76	576
Accuracy	82.17%%	
False Positive	21.65%	
False Negative	11.66%	