Sushmita Upreti – 00810955

FA 24 - BANL 6625 - 03

Final Examination

Dataset Used: “BANL 6625_Final_Exam_Dataset.csv”

# Loading packages

```
library(ggplot2)

library(readr)

library(caret)

library(class)

library(rpart)

library(rpart.plot)
```

1. Loading the dataset. Displaying its structure and identifying the types of variables (e.g.numerical or categorical).

# Loading the dataset

```
data <- read.csv("BANL 6625_Final_Exam_Dataset.csv")
```

# Displaying it’s structure

```
str(data)
```

```
> str(data)

'data.frame':    100 obs. of  6 variables:
 $ Age           : int  56 69 46 32 60 25 38 56 36 40 ...
 $ Income          : int  52773 60996 38427 35398 84386 28244 41901 76755 49841 60947 ...
 $ Spending_Score  : int  59 32 96 88 52 62 58 52 12 39 ...
 $ City_Type       : chr  "Rural" "Suburban" "Suburban" "Suburban" ...
 $ Education_Level : chr  "Bachelor's" "Bachelor's" "High School" "High School" ...
 $ Product_Purchase: int  0 0 0 0 0 0 1 1 0 1 ...
```

Identifying the types of variables:

The total number of observations in the dataset are 100 and variables 6.

Numerical Variables:

- **Age:** It is an integer representing the age of individuals, it is considered a **numerical variable** in broader terms.
- **Income**: It represents the individuals income amount in USD and is a **numerical variable.**
- **Spending_Score:** It is a **numerical variable** reflecting the spending score of individuals. Higher scores is equal to higher spendings.

Categorical Variables:

- **City_Type**: It is a **categorical variable** that shows the areas where the individuals reside (e.g: Rural, Urban, Suburban).
- **Education_Level**: It is also a **categorical variable** representing the level of education everyone has (e.g: Bachelor's, high school, Master's, PhD)
- **Product_Purchase**: Although it contains binary numbers, it is considered as a **categorical variable** because they represent categories in which the individuals lie (e.g: 1= individuals have purchased products, 0= individuals have not purchased products)

Generating summary statistics:

# Displaying summary of the data

```
summary(data)
```

```
Age              Income           Spending_Score     City_Type          Education_Level
Min.   :19.00    Min.   : 15529   Min.   : 1.00      Length:100         Length:100
1st Qu.:31.75    1st Qu.: 43607   1st Qu.:20.00      Class :character   Class :character
Median :42.00    Median : 54966   Median :52.00      Mode  :character   Mode
:character
Mean   :43.35    Mean   : 55276   Mean   :48.76
3rd Qu.:57.00    3rd Qu.: 62895   3rd Qu.:73.50
Max.   :69.00    Max.   :101696          Max.   :99.00

Product_Purchase
Min.   :0.00
1st Qu.:0.00
Median :0.00
Mean   :0.39
3rd Qu.:1.00
Max.   :1.00
```

Key Insights:

- The sample includes individuals ranging from 19 to 69 years old, indicating a diverse range of ages. The median age is 42, which appears balanced and shows that half of the population are youths and half are older. Furthermore, the mean is extremely close to the median, indicating a symmetric distribution.

- Personal income ranges from $15,529 to $101,696. It appears that 25% of the population make $43,607 or less, while 75% earn $62,895 or less.

- Spending scores range from 1 to 99, indicating that individuals have a score as low as 1 or as high as 99. The mean (48.76) and median (52) appear to be close, indicating a symmetric distribution.

- The city type and education levels are both categorical variables so they are represented in characters and will need further analysis.

- In terms of product purchase, the mean is 39% which indicates that on average 39% of the population purchased the product.

Preprocessing data & handling missing values:

# Looking for missing values

```
sum(is.na(data))
```

```
> sum(is.na(data))

[1] 0
```

There seems to be no missing values in the entire data set.

# Transforming categorical variables to factors

```
data$City_Type <- as.factor(data$City_Type)

data$Education_Level <- as.factor(data$Education_Level)

data$Product_Purchase <- factor(data$Product_Purchase, levels = c(0,1), labels = c("No","Yes"))
```

# Displaying the structure for confirmation

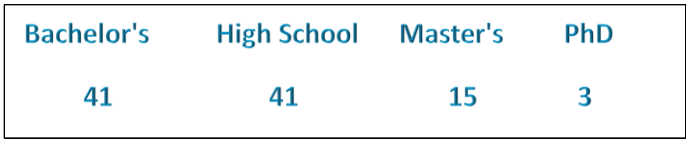
```
str(data)
```

```
> str(data)
'data.frame':    100 obs. of  6 variables:
 $ Age          : int  56 69 46 32 60 25 38 56 36 40 ...
 $ Income        : int  52773 60996 38427 35398 84386 28244 41901 76755 49841 60947 ...
 $ Spending_Score  : int  59 32 96 88 52 62 58 52 12 39 ...
 $ City_Type      : Factor w/ 3 levels "Rural","Suburban",..: 1 2 2 2 3 1 1 3 3 3 ...
 $ Education_Level : Factor w/ 4 levels "Bachelor's","High School",..: 1 1 2 2 4 1 2 2 1 1 ...
 $ Product_Purchase: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 2 1 2 ...
```

The categorical variables have successfully been transformed into factors, the "Education_Level" has four levels ("Bachelor's, "High School", "Master's", "PhD") and the "City_Type" variable has three levels ("Rural", "Suburban", "Urban"). The "Product_Purchase" variable has also been changed into factor variables with two levels ("No", "Yes").

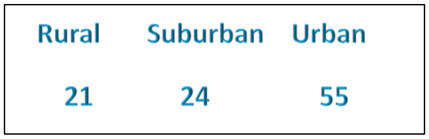
# Creating frequency table for overview

```
table(data$Education_Level)
```



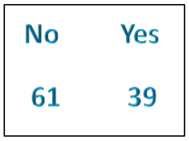
```
Bachelor's    High School    Master's    PhD

    41             41           15        3
```

```
table(data$City_Type)
```



```
Rural    Suburban    Urban

  21        24         55
```

```
table(data$Product_Purchase)
```



```
No     Yes

61      39
```

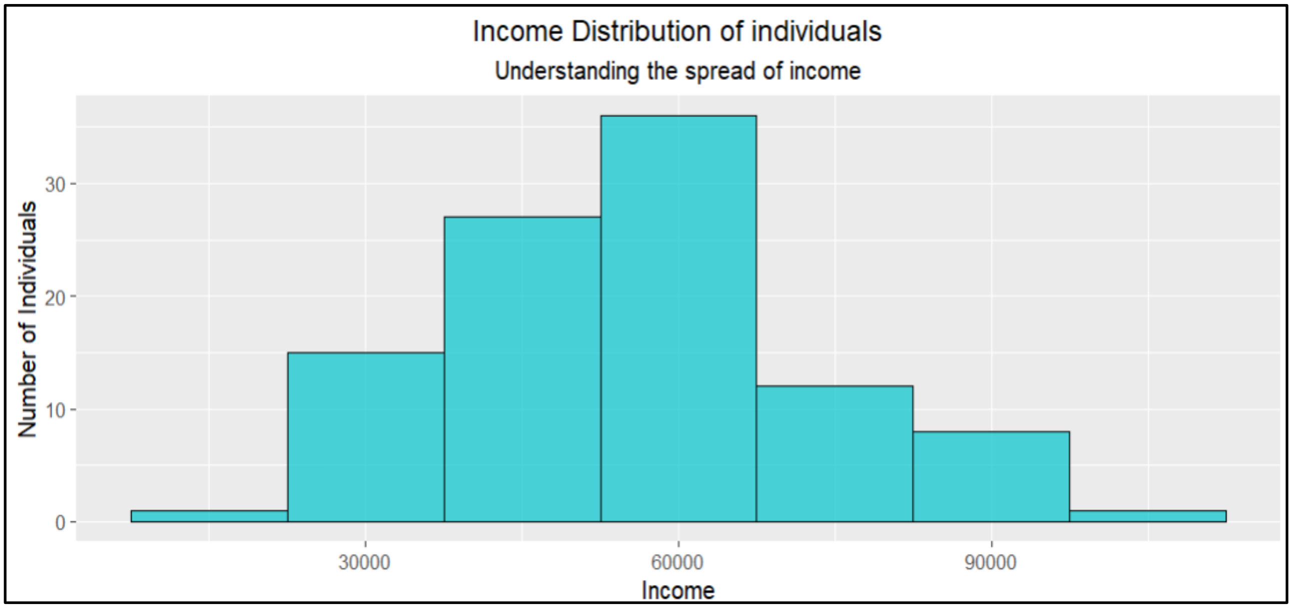
In terms of educational level, PhD is the least studied degree, followed by Master's. The urban area appears to have the highest population density. Finally, there appears to be more people who have not purchased any products (61) than those who have (39).

2. Creating at least two data visualizations (e.g., histograms, box plots, scatter plots) to explore relationships and distributions within the dataset.

```
options(scipen = 999)
```

# Histogram for income distribution

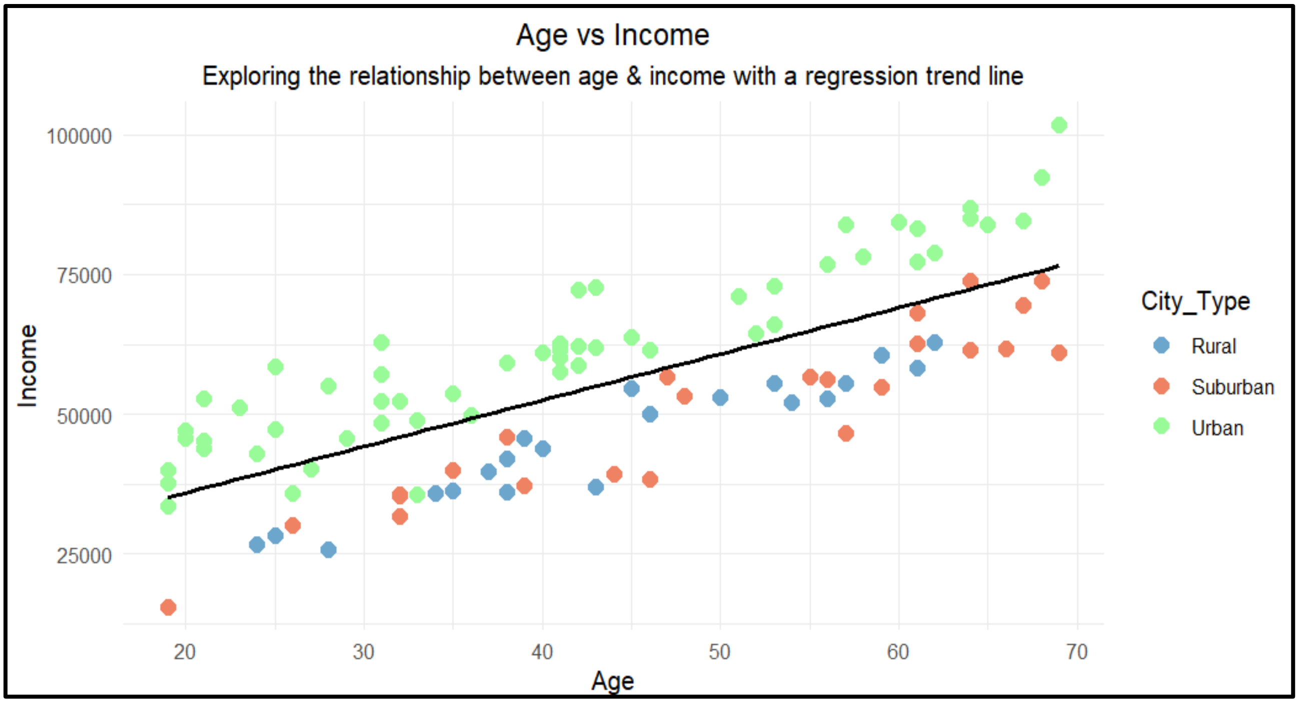
```
ggplot(data, aes(x = Income)) +
          geom_histogram(binwidth = 15000, color = "black", fill = "turquoise3", alpha = 0.7) +
          labs(
                  title = "Income Distribution of individuals",
                  subtitle = "Understanding the spread of income",
             x = "Income",
                  y = "Number of Individuals"
              )+
          theme(
                  plot.title = element_text(hjust = 0.5),
                  plot.subtitle = element_text(hjust = 0.5)
          )
```



The histogram does not appear to be excessively skewed to the left or right; instead, it appears to be roughly balanced. The majority of people make between $40,000 and $70,000 each year, with the highest earning $60,000. There appear to be fewer people in the higher income bracket (above $80.000).

Scatterplot for Age vs. Income with a trend line

```
ggplot(data, aes(x = Age, y = Income))+
          geom_point(aes(color = City_Type), size = 3)+
          geom_smooth(method = "lm", se = FALSE, color = "black", linetype = "solid")+
          theme_minimal()+
           labs(
                 title = "Age vs Income",
                 subtitle = "Exploring the relationship between age & income with a
regression trend line",
                 x = "Age",
                 y = "Income"
              )+
      scale_color_manual(values = c("Urban" = "palegreen", "Suburban" = "salmon2",
"Rural" = "skyblue3"))+
         theme(
                 plot.title = element_text(hjust = 0.5),
                 plot.subtitle = element_text(hjust = 0.5)
          )
```



Key Insights:

- The upward sloping trend line indicates a positive correlation between age and income. This means that elderly people typically have larger incomes.

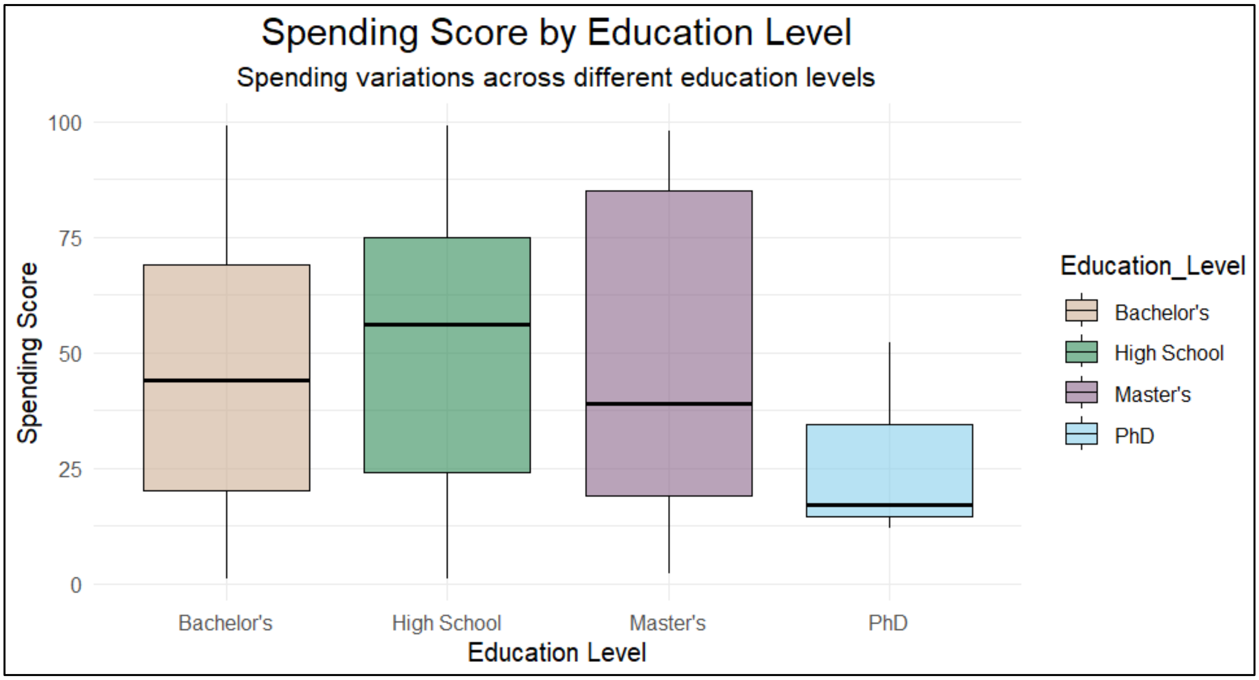
- The individuals residing in the urban city type (green) have the highest income (range = $25,000-$100,000) on average compared to the rural and suburban city type showing the strongest correlation in the plot.

- The maximum income earned by the rural and suburban population appears to be around $75000, almost all being below the trend line.

- Younger individuals who are below 30 years seem to earn less compared to the older individuals who are 50 years and above.

# Box plot showing spending scores by education level

```
ggplot(data, aes(x = Education_Level, y = Spending_Score, fill = Education_Level))+
         geom_boxplot(alpha = 0.6, color = "black")+
             labs(
                     title = "Spending Score by Education Level",
                     subtitle = "Spending variations across different education levels",
                     x = "Education Level",
                     y = "Spending Score"
                 )+
         theme_minimal()+
         theme(
                 plot.title = element_text(hjust = 0.5, size = 16),
                 plot.subtitle = element_text(hjust = 0.5, size = 11.5),
                 axis.text.x = element_text(angle = 0, hjust = 0.5),
                 legend.position = "right"
             )+
         scale_fill_manual(values = c(
                 "Bachelor's" = "peachpuff3",
                 "High School" = "seagreen",
                 "Master's" = "plum4",
                 "PhD" = "skyblue"
))
```



The individuals with a PhD appear to have lower spending scores compared to other education levels with its median being around 25. This may suggest that the people who have a PhD tend to be more conservative in purchasing habits. The population with a high school degree appears to have the most spending score with its median being slightly above 50. The median expenditure scores of Master's and high school groups are similar, however those with a master's degree exhibit higher variability in their purchasing habits.

3. Implementing a K-Nearest Neighbors model to predict the Product_Purchase column using the features Age, Income, Spending_Score, and City_Type. Evaluating the model’s performance using appropriate metrics (e.g., accuracy, confusion matrix) and report the accuracy when k=5.

I will be preprocessing the data, converting categorical variables to numeric.

```
set.seed(2024)
```

To ensure consistency, I used the set seed function to specify the random number generator's starting point. This ensures that the data is divided into training and testing sets in the same manner each time the code is called.

Splitting data into training and testing sets (70% training, 30% testing)

```
 trainIndex <- createDataPartition(data$Product_Purchase, p = 0.7,
                                  list = FALSE,

                                  times = 1)

 train_data <- data[trainIndex, ]

 test_data <- data[-trainIndex, ]
```

The dataset was separated into two subsets: 70% for model training and 30% for performance assessment. For the training dataset, row indices were chosen at random, with a 70/30 split. This strategy trains the model on 70% of the data, while the remaining 30% is used to evaluate the model's generalizability.

```
dim(train_data)

[1] 71  6

dim(test_data)

[1] 29  6
```

The training data now has 71 observations with 6 variables, and the test data has 29 observations and 6 variables. This confirms that the 70/30 split was successful.

Scaling the data

```
 preprocess <- preProcess(train_data[, c("Age", "Income", "Spending_Score")], method
= "scale")

 train_data_scaled <- predict(preprocess, train_data)

 test_data_scaled <- predict(preprocess, test_data)
```

The following code uses the preProcess() method to scale the numerical features (Age, Income, Spending_Score) in the training and testing datasets.

```
> str(train_data_scaled)
'data.frame':    71 obs. of  6 variables:
$ Age           : num  3.66 4.51 3.01 2.09 2.48 ...
$ Income         : num  3.2 3.7 2.33 2.15 2.54 ...
$ Spending_Score  : num  1.93 1.05 3.14 2.88 1.9 ...
$ City_Type       : Factor w/ 3 levels "Rural","Suburban",..: 1 2 2 2 1 3 3 3 3 3 ...
$ Education_Level : Factor w/ 4 levels "Bachelor's","High School",..: 1 1 2 2 2 2 1 1 1 1
...
$ Product_Purchase: Factor w/ 2 levels "No","Yes": 1 1 1 1 2 2 1 2 1 1 ...
```

Upon scaling, the numerical variables such as Age, Income, spending score have now been standardized. Their values have been transformed to a mean of 0 and standard deviation of 1.This ensures that all the variables contribute equally to the model despite their differences in units.

Training with KNN model with k=5

```
knn_model <- train(Product_Purchase ~ Age + Income + Spending_Score + City_Type,
                   data = train_data_scaled,
                   method = "knn",
                   tuneGrid = data.frame(k = 5))
```

The code will use the features such as Age, Income, Spending_Score and City_Type to predict the target variable which in this case is "Product_Purchase".

Making predictions on the test data using KNN model

```
predictions <- predict(knn_model, newdata = test_data_scaled)
```

Comparing these predictions to the actual values using a confusion matrix

```
conf_matrix <- confusionMatrix(predictions, test_data_scaled$Product_Purchase)
print(conf_matrix)
```

```
> print(conf_matrix)
Confusion Matrix and Statistics
          Reference
Prediction  No Yes
      No    14  8
      Yes    4  3

          Accuracy : 0.5862
            95% CI : (0.3894, 0.7648)
  No Information Rate : 0.6207
  P-Value [Acc > NIR] : 0.7202
             Kappa : 0.0543
  Mcnemar's Test P-Value : 0.3865
```

```
          Sensitivity : 0.7778
          Specificity : 0.2727
       Pos Pred Value : 0.6364
       Neg Pred Value : 0.4286
           Prevalence : 0.6207
       Detection Rate : 0.4828
 Detection Prevalence : 0.7586
    Balanced Accuracy : 0.5253

     'Positive' Class : No
```

Reflecting on the strengths and limitations of the model:

1. Confusion matrix:

There were 14 cases where the model properly predicted "did not purchase"; these were the true negatives. There were three cases where the model properly predicted "Yes" (purchased), which were true positives.

Regarding the false aspects, there were four instances in which the model predicted "Yes" instead of "No". There were eight instances in which the model predicted "did not purchase" instead of "purchased".

2. Accuracy:

The model's accuracy is 58.62%, indicating that it performs better than random guessing. However, there is still chances of improvement.

3. Sensitivity:

The model's sensitivity is 77.78%, indicating that it properly identifies true positives by 77.78%, which is fairly good. However, the model's weakness is that it generates a large number of false negatives, implying that it is still missing purchases.

4. Specificity:

The model's specificity is extremely low at 27.27%, indicating that it has difficulty predicting false positives.

5. P-value:

The p-value of 0.7202 suggests that the model's predictions are just slightly better than guessing the majority class.

4. Building a linear regression model to predict Income using the features Age, Spending_Score, and City_Type.

```
# Building a regression model to predict Income
 lm_model <- lm(Income ~ Age + Spending_Score + City_Type, data = train_data)
 View(lm_model)
 summary(lm_model)
```

```
Call:

lm(formula = Income ~ Age + Spending_Score + City_Type, data = train_data)

Residuals:

    Min       1Q       Median    3Q        Max

-11028.3  -3388.1  -178.3      3012.2   14411.4

Coefficients:

                       Estimate       Std. Error       t value        Pr(>|t|)

(Intercept)            125.11         2695.54          0.046          0.9631

Age                    980.03         41.78            23.456        <2e-16 ***

Spending_Score         40.40          20.39            1.981          0.0517 .

City_TypeSuburban      -1172.41       1820.27          -0.644         0.5218

City_TypeUrban         18892.00       1534.72          12.310         <2e-16 ***

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5102 on 66 degrees of freedom

Multiple R-squared:  0.9097,     Adjusted R-squared:  0.9042

F-statistic: 166.2 on 4 and 66 DF,  p-value: < 2.2e-16
```

The R-squared score is 0.9097, indicating that the model's predictors explain approximately 90.97% of the variation in Income. Similarly, the Adjusted R-squared takes into account the number of predictors and determines overfitting. Compared to the R-squared number, it provides a more conservative approach, which in this case is quite high at 90.42%. This shows that the model works properly.

The predictors "Age" and "City_TypeUrban" have three stars, suggesting that they are highly statistically significant and have a strong relationship with Income. Spending_Score has a dot, indicating that it is only slightly significant and has a weaker association with Income.

Identifying any additional metrics (e.g., Mean Squared Error) that I would use to evaluate the model's performance.

Predicting values by using linear regression model

```
predictions <- predict(lm_model, newdata = test_data)
```

Calculating the MSE

```
mse <- mean((test_data$Income - predictions)^2)

print(paste("Mean Squared Error (MSE):", round(mse, 2)))
```

```
"Mean Squared Error (MSE): 38799739.09"
```

The MSE for my linear regression model is 38799739.09. This number is the average of the squared discrepancies between the expected and actual values of Income. In this case, the high value of MSE indicates that the average squared error between the actual and predicted values of Income is quite high.

I will also be calculating the Root Mean Squared Error for further analysis.

```
rmse <- sqrt(mse)

print(paste("Root Mean Squared Error (RMSE):", round(rmse, 2)))
```

```
"Root Mean Squared Error (RMSE): 6228.94"
```

RMSE is the average distance between anticipated and actual values, calculated in the same units as the target variable, in this case Income. The RMSE is 6,228.94, indicating that the model's forecasts are often wrong by approximately 6,228.94 units of income.

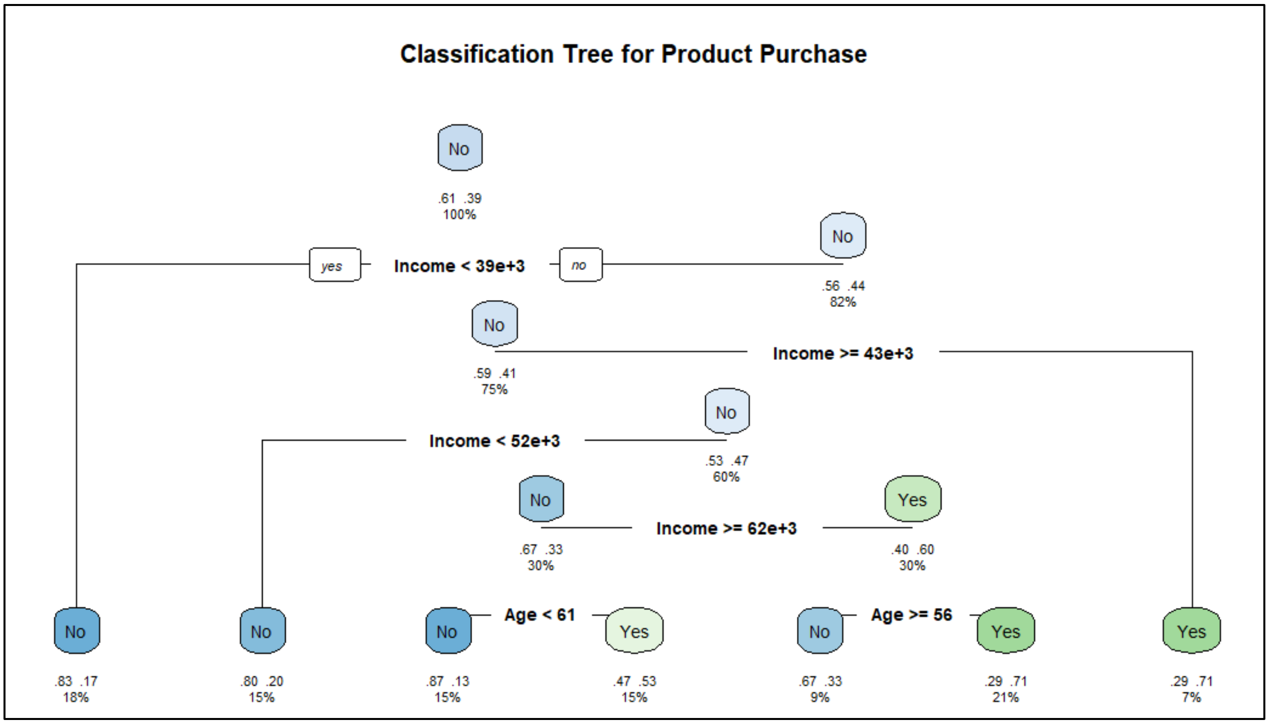
5. Creating a classification tree to predict Product_Purchase using all other variables in the dataset as predictors.

Building the classification tree

```
 tree_model <- rpart(Product_Purchase ~ Age + Income + Education_Level + City_Type +
Spending_Score ,

                  data = data,

                  method = "class",

                  control = rpart.control(cp = 0.01)) # Initial cp value
```

Visualizing the tree model

```
rpart.plot(tree_model, type = 2, extra = 104, under = TRUE, tweak = 1.2,

            main = "Classification Tree for Product Purchase")
```



The classification tree reveals that income and age are major determinants of Product_Purchase. The top primary split is Income < 39000, indicating that if an individual's income is less than $39000, the decision to purchase a product is "No". In terms of significant splits, if the income is $62,000 or higher, the tree splits by age. At this point, if the person is under the age of 61 and earns $62,000 or more, they are likely to purchase the product, with the prediction being "Yes".

Evaluate the classification trees performance using appropriate metrics:

I will be predicting on the test dataset,

```
test_predictions <- predict(tree_model, test_data, type = "class")
```

Next, I will be creating a confusion matrix

```
conf_matrix <- table(Predicted = test_predictions, Actual =
test_data$Product_Purchase)

print(conf_matrix)
```

	Actual	
Predicted	*No*	*Yes*
No	*16*	*4*
Yes	*2*	*7*

The **true negatives** where model predicted “No” and the outcome was also no (did not purchase) is **16**. Whereas, the **false negatives** where the outcome was yes (purchased) but is predicted as “No” is **4.** In terms of **false positive**, where the person did not purchase but is predicted as “Yes” is **2.** Finally, for **true positive** where the person did purchase products and is rightfully predicted as “Yes” is **7**.

# Calculating the accuracy

```
 accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
cat("Accuracy:", round(accuracy, 2), "\n")
```

```
Accuracy: 0.79
```

# Calculating precision, recall and F1 score

```
 TP <- conf_matrix["Yes", "Yes"]
 TN <- conf_matrix["No", "No"]
 FP <- conf_matrix["Yes", "No"]
 FN <- conf_matrix["No", "Yes"]
precision <- TP/ (TP + FP)
cat("Precision:", round(precision, 2), "\n")
```

```
Precision: 0.78
```

```
recall <- TP / (TP + FN)
cat("Recall:", round(recall, 2), "\n")
```

```
Precision: 0.64
```

```
f1_score <- 2 * (precision * recall) / (precision + recall)
cat("F1 Score:", round(f1_score, 2), "\n")
```

```
F1 Score: 0.7
```

The model obtained 79% accuracy, which means that 79% of its predictions were correct for both positive and negative classifications. I also calculated the Precision, Recall, and F1 scores to evaluate the effectiveness of the model. Recall measures the proportion of actual positives, which is 64%. This suggests that 64% of the "Yes" cases were accurately predicted, indicating chances for improvement. The F1 score of 0.7 indicates good performance in terms of both false positives and negatives.

6. Summary of my overall workflow, including data preparation, model development, and key findings:

1. Data Preparation & Overall Workflow:

The data set “BANL 6625_Final_Exam_Dataset.csv” had a wide range of individuals in terms of Age, Income and Spending scores. The sample data had no missing values so I moved onto data transformation where the categorical variables were transformed into factors for analysis. A total of 3 data visualizations were made; a histogram to analyze the distribution of Age, a scatter plot showing the relationship between Age, Income and City_Type, and the third visualization being a box plot for analyzing the Spending_Scores by Education_Level. Next, the data was split into training (70%) and testing sets (30%) to implement the K-Nearest Neighbors model to predict the Product_Purchase column. The next step was evaluating the model’s performance using the confusion matrix. Once the strengths and limitations were discussed I built a logistic regression model to predict the Income using other variables. Additional metrics such as MSE and RMSE were used to evaluate the model’s performance.Lastly, I created and visualized a classification tree to predict Product_Purchase using all othervariables in the dataset as predictors. I used metrics like Accuracy, Precision, Recall and F1 score to further evaluate my outcome.

2. Key Findings & Recommendations:

Based on the visualizations, the income distribution appeared to be reasonably balanced, with fewer people earning more than $80,000. There is a significant association between age and income, with older people earning more, particularly in urban areas. Spending Scores vary according to education level, with high school graduates spending the most and PhD holders spending the least.

While implementing the K-nearest Neighbors Model (k = 5) the achieved accuracy was 58.62 which was not bad but there is room for improvement. Similarly, the sensitivity was also very low at 27.27% indicating that the model’s predictions were only marginally better than random guessing. The predictors "Age" and "City_TypeUrban" had three stars, suggesting that they are highly statistically significant and have a strong relationship with Income.

Both the MSE and RMSE were very high with values 38,799,739.09 and 6,228.94 indicating quite high predictor errors. Lastly, the decision tree revealed income and age as the major predictors of product purchase, the population with income less than $39000 are highly unlikely to purchase whereas, the combination of younger individuals and higher income are most likely to buy.

Finally, both the KNN model and the classification tree model struggled to reliably predict product purchases, indicating possible areas for development.

Reference:

OpenAI. (2024). ChatGPT (Mar 14 version) [Large language model]

https://chatgpt.com/c/67573622-0298-8009-a107-ca01ce06c37f