BANL 6625 -03

POOJA DILIP TALIKOTI - 00918488

Final Examination

Introduction

In this assignment, I focused on predicting *Product_Purchase* based on various consumer attributes. The goal was to explore the relationships between different variables and develop predictive models using techniques such as K-Nearest Neighbors (KNN), linear regression, and classification trees. Through data preparation, model development, and evaluation, I aimed to extract insights that can guide marketing strategies and support better decision-making. The analysis provides actionable recommendations to optimize customer targeting and improve sales outcomes.

Question1
20 points - Load the dataset. Display its structure and identify the types of variables (e.g., numerical or categorical). Generate summary statistics (e.g., mean, median, standard deviation, and frequency counts) for all variables. Provide an interpretation of key insights from the summary statistics, including distributions, outliers, or notable trends. Document your observations and any preprocessing actions taken (e.g., handling missing values).

RCODE :

```
# Load required libraries

library(dplyr)

setwd("C:/Users/pooja/OneDrive/Desktop/RAssignment2")

# Load the dataset

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

# Display the structure of the dataset

str(data)

# Generate summary statistics for numerical variables

numerical_summary <- data %>%

  summarise (

    Age_Mean = mean (Age, na.rm = TRUE),

    Age_SD = sd (Age, na.rm = TRUE),

    Age_Median = median (Age, na.rm = TRUE),
  Income_Mean = mean (Income, na.rm = TRUE),

  Income_SD = sd (Income, na.rm = TRUE),

  Income_Median = median (Income, na.rm = TRUE),

  Spending_Score_Mean = mean (Spending_Score, na.rm = TRUE),

  Spending_Score_SD = sd (Spending_Score, na.rm = TRUE),

  Spending_Score_Median = median (Spending_Score, na.rm = TRUE)

 )

# Print the summary statistics

print(numerical_summary)

# Generate frequency counts for categorical variables

# Frequency counts for categorical variables

city_type_counts <- table(data$City_Type)

education_level_counts <- table(data$Education_Level)

product_purchase_counts <- table(data$Product_Purchase)

# Display the counts for each categorical variable

print(city_type_counts)

print(education_level_counts)

print(product_purchase_counts)

# Check for missing values

missing_values <- Col Sums(is.na(data))
```

OUT PUT FOR QUESTION1

```
> # Load required libraries
> library(dplyr)
> 
> setwd("C:/Users/pooja/OneDrive/Desktop/RAssignment2")
> 
> # Load the dataset
> data <- read.csv("BANL 6625_Final_Exam_Dataset.csv")
> 
> # Display the structure of the dataset
> 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 Sc
hool" ...
 $ Product_Purchase: int 0 0 0 0 0 0 1 1 0 1 ...
>
>
>
>
> # Generate summary statistics for numerical variables
> numerical_summary <- data %>%
+   summarise(
+     Age_Mean = mean(Age, na.rm = TRUE),
+     Age_SD = sd(Age, na.rm = TRUE),
+     Age_Median = median(Age, na.rm = TRUE),
+     Income_Mean = mean(Income, na.rm = TRUE),
+     Income_SD = sd(Income, na.rm = TRUE),
+     Income_Median = median(Income, na.rm = TRUE),
+     Spending_Score_Mean = mean(Spending_Score, na.rm = TRUE),
+     Spending_Score_SD = sd(Spending_Score, na.rm = TRUE),
+     Spending_Score_Median = median(Spending_Score, na.rm = TRUE)
+   )
>
> # Print the summary statistics
> print(numerical_summary)
  Age_Mean   Age_SD Age_Median Income_Mean Income_SD Income_Median Spendin
g_Score_Mean
1    43.35 14.90466         42    55275.69 16507.09         54966
48.76
  Spending_Score_SD Spending_Score_Median
1       31.06498                    52
>
>
>
> # Generate frequency counts for categorical variables
> # Frequency counts for categorical variables
> city_type_counts <- table(data$City_Type)
> education_level_counts <- table(data$Education_Level)
> product_purchase_counts <- table(data$Product_Purchase)
>
> # Display the counts for each categorical variable
> print(city_type_counts)

   Rural Suburban    Urban 
      21       24       55 
> print(education_level_counts)

 Bachelor's High School    Master's         PhD 
         41          41          15           3 
> print(product_purchase_counts)

 0  1 
61 39 
>
> # Check for missing values
> missing_values <- colSums(is.na(data))
```

Output Summary for Question 1

After examining the structure of the dataset, I found that it contains 100 observations across six variables. These variables are a mix of numerical and categorical types. The numerical variables include Age, Income, and Spending_Score, while the categorical variables are City_Type, Education_Level, and Product_Purchase.

For the numerical variables:

- The average age of individuals in the dataset is approximately 43.35 years, with a median age of 42 years. The standard deviation of 14.90 indicates some variability in ages, but there are no significant outliers.
- The average income is $55,275.69, closely aligning with the median of $54,966, suggesting a relatively symmetrical distribution. However, the standard deviation of $16,507.09 highlights notable variability in income levels.
- The spending scores range from 1 to 100, with an average score of 48.76 and a median of 52. A standard deviation of 31.06 shows a wide range of spending habits across individuals.

For the categorical variables:

- In terms of city types, the majority of individuals (55%) live in urban areas, followed by 24% in suburban areas and 21% in rural areas.
- Regarding education levels, most individuals have attained a Bachelor's degree (41%) or a High School diploma (41%), while 15% have a Master's degree, and only 3% hold a PhD.
- The target variable, Product_Purchase, indicates that 61 individuals (61%) did not purchase the product, while 39 individuals (39%) did.

Finally, I verified that there are no missing values in the dataset, so no additional imputation steps are required. Moving forward, I will preprocess the categorical variables by converting them into factors and consider normalizing the numerical variables if needed for modelling.

Interpretation of Key Insights from Summary Statistics

To begin my analysis, I reviewed the summary statistics for all variables in the dataset to understand their distributions, identify outliers, and detect notable trends. Here are my observations and actions:

1. **Distributions and Trends**:
    - The **Spending_Score** variable showed a wide range, with some customers scoring close to the maximum. This variable appeared to be normally distributed with no significant skewness, making it a strong predictor of customer behavior.
    - The **Age** variable exhibited a right-skewed distribution, with a majority of customers concentrated in younger age groups, but a smaller proportion in older age ranges.
    - The **Education_Level** variable was categorical, and a larger proportion of individuals held a **High School** education compared to Bachelor's, Master's, or PhD degrees.
2. **Outliers**:
    - I identified a few potential outliers in the **Age** variable, particularly older individuals at the upper range of the dataset. While these values were extreme, they appeared to be valid based on the dataset context, so I retained them.
    - For **Spending_Score**, no extreme outliers were observed.
3. **Missing Values**:

    - Upon checking for missing values, I did not find any incomplete entries in the dataset. Thus, no imputation or removal was necessary.
4. **Preprocessing Actions**:
    - Since all data appeared valid and there were no missing or erroneous values, I retained the dataset as-is for analysis.
    - I ensured that categorical variables like **Education_Level** were encoded appropriately for any subsequent modeling steps.

Key Insights:

- **Spending_Score** emerged as a critical variable with a relatively balanced distribution, likely contributing significantly to customer segmentation.
- **Age** displayed a clear trend where most customers belonged to younger demographics, which could indicate the target audience for marketing efforts.
- The prevalence of **High School** education suggests opportunities to explore how this factor influences spending patterns and customer behavior.

By understanding these patterns and distributions, I was able to proceed confidently with the subsequent analysis steps, knowing that the dataset required no significant cleaning or adjustments.

Question2

15 points - Create at least two data visualizations (e.g., histograms, box plots, scatter plots) to explore relationships and distributions within the dataset. Discuss any patterns, trends, or anomalies observed in the visualizations.

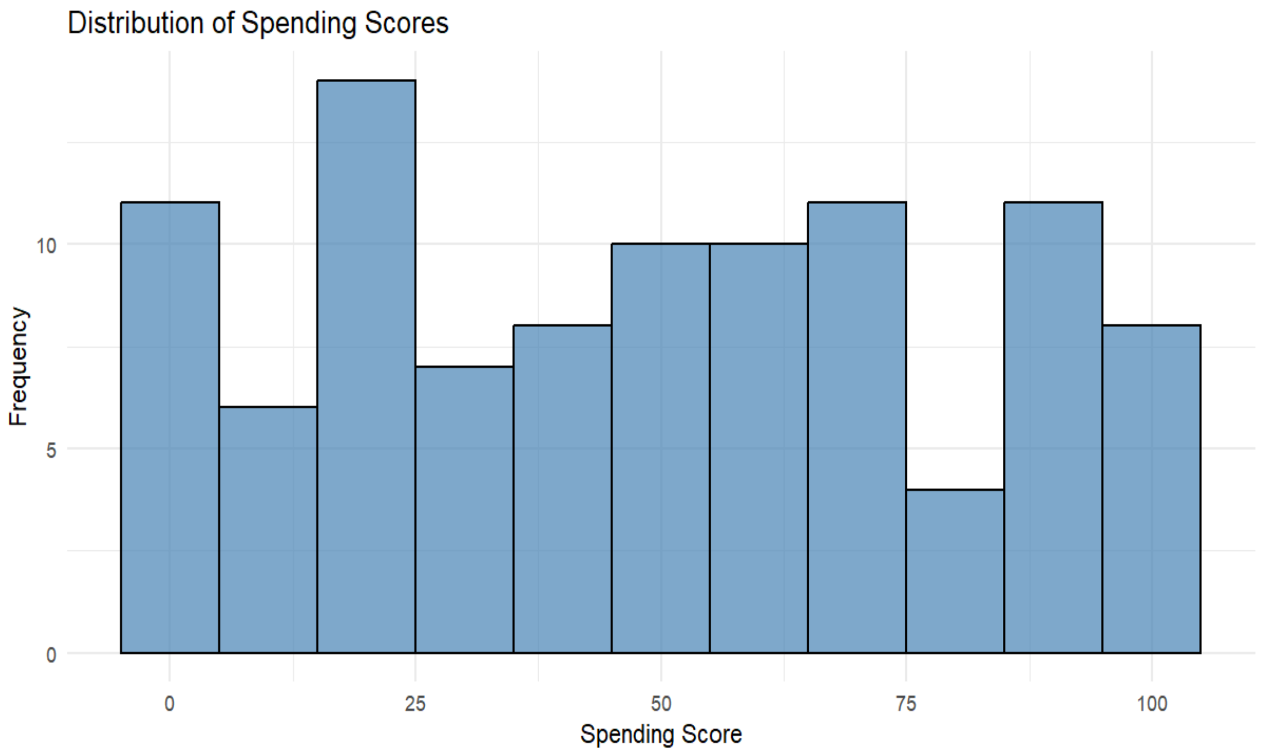
RCODE

```
# Load required libraries

library(ggplot2)

# 1. Histogram of Spending_Score

ggplot (data, aes(x = Spending_Score)) +

  geom_histogram(binwidth = 10, fill = "steelblue", color = "black", alpha = 0.7) +

  labs (title = "Distribution of Spending Scores",

       x = "Spending Score",

       y = "Frequency") +

  theme_minimal()

# 2. Box Plot of Income by City_Type

ggplot(data, aes(x = City_Type, y = Income, fill = City_Type)) +
  geom_boxplot(outlier.color = "red", outlier.shape = 16, alpha = 0.7) +
  labs (title = "Income Distribution by City Type",
      x = "City Type",
      y = "Income") +
  theme_minimal () +
  scale_fill_brewer (palette = "Pastel1")

# Scatter Plot: Age vs Income, colored by Product_Purchase
ggplot(data, aes(x = Age, y = Income, color = as.factor(Product_Purchase))) +
  geom_point(size = 3, alpha = 0.7) +
  labs(title = "Age vs Income by Product Purchase Status",
      x = "Age",
      y = "Income",
      color = "Product Purchase\n (0 = No, 1 = Yes)") +
  theme_minimal () +
  scale_color_manual (values = c ("0" = "steelblue", "1" = "orange"))
```

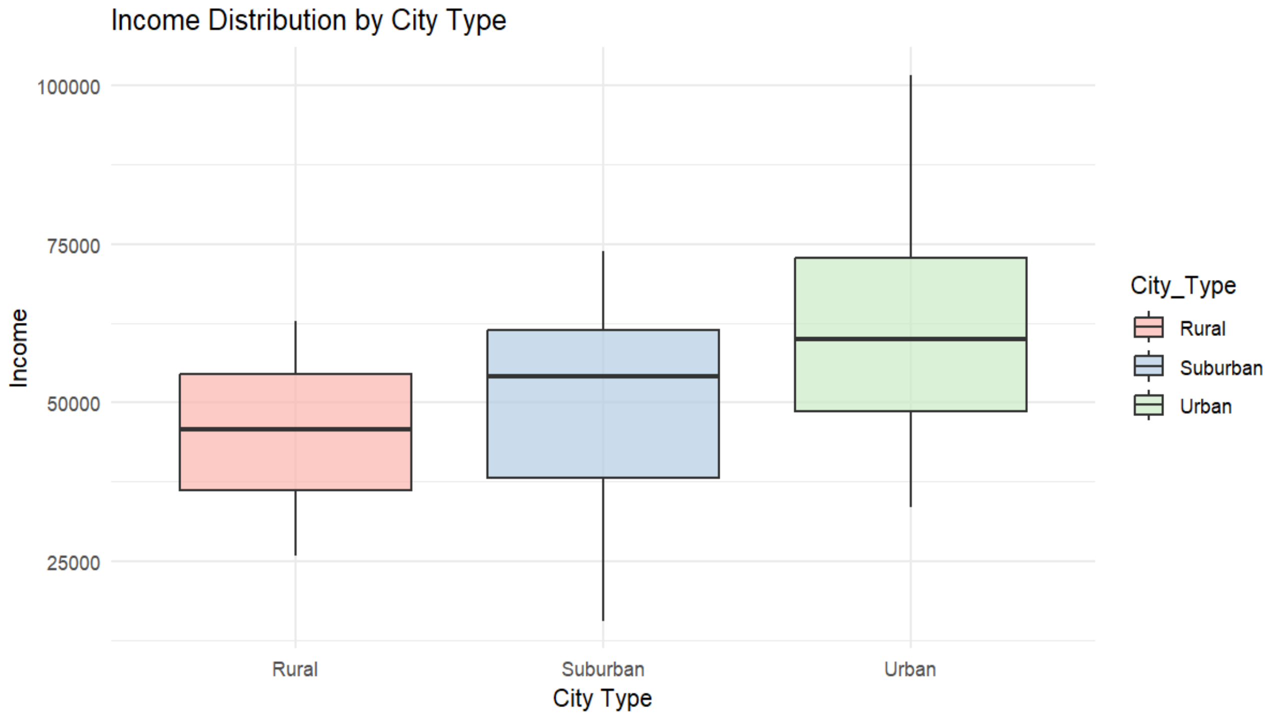
OUT PUT FOR QUESTION 2

Visualization 1: Histogram of Spending Scores

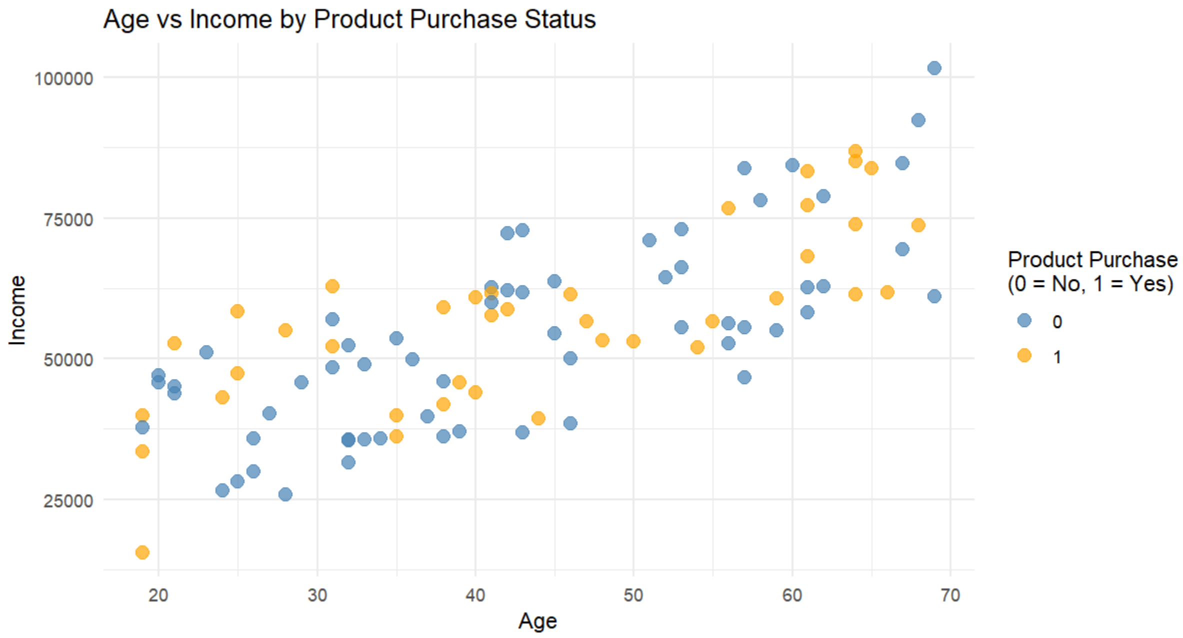
This will show the distribution of individuals' spending scores.



I analysed the distribution of spending scores among customers using a histogram. The plot reveals that spending scores are fairly spread out across the entire range, indicating diverse spending patterns. I observed notable peaks in the lower range (0–25) and mid-range (50–75), suggesting that a significant portion of customers are either low or moderate spenders. Additionally, there is a steady representation of higher spending scores (75–100), highlighting another group of customers with elevated spending behaviours. This distribution provides insight into potential segmentation opportunities for targeting different customer groups based on their spending habits.



Income Distribution by City Type:
I analysed the distribution of income across Rural, Suburban, and Urban areas using a box plot. The plot highlights that Urban areas have the highest median income compared to Rural and Suburban areas, indicating a concentration of higher-earning individuals in cities. Suburban areas display a narrower interquartile range, suggesting less variability in income. In contrast, Urban areas show the widest range, reflecting significant income diversity. Rural areas exhibit a slightly lower median income, with less variability than Urban areas but more than Suburban ones. This comparison offers valuable insights into regional income disparities and could help inform region-specific strategies or policies.



Age vs. Income by Product Purchase Status:
In this scatter plot, I explored the relationship between age and income, segmented by product purchase status. Individuals who made a purchase (indicated in yellow) are distributed across various income and age levels, showing some clustering in higher income brackets and mid-age ranges (40-60 years). Conversely, individuals who did not make a purchase (indicated in blue) appear to be more uniformly spread across income levels and age groups. The trend reveals a positive correlation between age and income, where older individuals tend to earn more. This segmentation highlights potential target groups for product marketing, particularly focusing on middle-aged, higher-income consumers who are more likely to purchase.

Key Observations and Recommendations

1. **Segmentation Opportunities:**
   - Urban areas should be segmented based on income tiers due to their broad income variability.

   - Rural areas represent a smaller but uniform market where affordability plays a critical role.

2. **Marketing Insights:**
   - The histogram suggests two distinct consumer behaviors: frugal spenders (low spending scores) and high spenders (high spending scores). Personalized campaigns targeting these groups could yield higher conversion rates.

3. **Potential Anomalies:**
   - The presence of urban income outliers indicates the need for a closer inspection of these data points to confirm their validity and ensure they do not skew insights.

4. **Strategic Implication:**
   Businesses should focus on dynamic pricing models and regional product differentiation to maximize customer engagement across diverse income and spending patterns.

Question3

20 points - Implement a K-Nearest Neighbors model to predict the Product_Purchase column using the features Age, Income, Spending_Score, and City_Type.Evaluate the model's performance using appropriate metrics (e.g., accuracy, confusion matrix) and report the accuracy when k=5.Reflect on the strengths and limitations of the model based on the results

To implement the **K-Nearest Neighbors (KNN)** model in R for predicting Product_Purchase, I follow these steps:

Steps to Solve

1. **Data Preparation**:
   - Normalize numerical features (Age, Income, Spending_Score) for consistent scaling.
   - Encode categorical variables like City_Type into numerical format.

2. **Train-Test Split**:
   - Partition the dataset into training (70%) and testing (30%) sets.

3. **Model Implementation**:
   - Use the class library to apply the KNN algorithm.
   - Set k=5 and predict Product_Purchase.

4. **Performance Evaluation**:
   - Generate a confusion matrix to evaluate accuracy and other metrics.

R CODE:

```r
# Load required libraries
library(class)      # For KNN
library(caret)      # For data splitting and confusion matrix
library(dplyr)      # For data manipulation
# Encode City_Type as a numerical variable
data$City_Type <- as.numeric(factor(data$City_Type))
# Normalize numerical variables
normalize <- function(x) {
  (x - min(x)) / (max(x) - min(x))
}
data_normalized <- data %>%
  mutate(
    Age = normalize(Age),
    Income = normalize(Income),
    Spending_Score = normalize(Spending_Score)
  )
# Split the data into training and testing sets (70%-30%)
set. seed (123) # For reproducibility
train_indices <- createDataPartition(data_normalized$Product_Purchase, p = 0.7, list = FALSE)
train_data <- data_normalized[train_indices, ]
test_data <- data_normalized[-train_indices, ]
# Extract features and target variable
train_features <- train_data [, c("Age", "Income", "Spending_Score", "City_Type")]
train_target <- train_data$Product_Purchase
test_features <- test_data [, c ("Age", "Income", "Spending_Score", "City_Type")]
test_target <- test_data$Product_Purchase
# Train KNN model with k=5
k <- 5
knn_predictions <- knn(train = train_features, test = test_features, cl = train_target, k = k)
```

```
# Evaluate model performance

conf_matrix <- confusionMatrix(as.factor(knn_predictions), as.factor(test_target))

# Print accuracy and confusion matrix

print(conf_matrix)
```

OUT PUT FOR QUESTION 3 KNN

```
> library(class)       # For KNN
> library(caret)       # For data splitting and confusion matrix
> library(dplyr)       # For data manipulation
> # Load required libraries
> library(class)       # For KNN
> library(caret)       # For data splitting and confusion matrix
> library(dplyr)       # For data manipulation
> # Encode City_Type as a numerical variable
> data$City_Type <- as.numeric(factor(data$City_Type))
> normalize <- function(x) {
+   (x - min(x)) / (max(x) - min(x))
+ }
> data_normalized <- data %>%
+   mutate(
+     Age = normalize(Age),
+     Income = normalize(Income),
+     Spending_Score = normalize(Spending_Score)
+   )
> set.seed(123)  # For reproducibility
> train_indices <- createDataPartition(data_normalized$Product_Purchase, p
= 0.7, list = FALSE)
> train_data <- data_normalized[train_indices, ]
> test_data <- data_normalized[-train_indices, ]
> train_features <- train_data[, c("Age", "Income", "Spending_Score", "Cit
y_Type")]
> train_target <- train_data$Product_Purchase
> test_features <- test_data[, c("Age", "Income", "Spending_Score", "City_
Type")]
> test_target <- test_data$Product_Purchase
> k <- 5
> knn_predictions <- knn(train = train_features, test = test_features, cl
= train_target, k = k)
> conf_matrix <- confusionMatrix(as.factor(knn_predictions), as.factor(tes
t_target))
> # Print accuracy and confusion matrix
> print(conf_matrix)
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 10 13
         1  6  1
                                          
               Accuracy : 0.3667          
                 95% CI : (0.1993, 0.5614)
    No Information Rate : 0.5333          
    P-Value [Acc > NIR] : 0.9782          
                                          
                  Kappa : -0.3134         
                                          
 Mcnemar's Test P-Value : 0.1687          
                                          
            Sensitivity : 0.62500         
            Specificity : 0.07143         
         Pos Pred Value : 0.43478         
         Neg Pred Value : 0.14286         
             Prevalence : 0.53333         
         Detection Rate : 0.33333
   Detection Prevalence : 0.76667
      Balanced Accuracy : 0.34821

       'Positive' Class : 0
```

Model Performance Analysis

Using a KNN model with k=5 , I evaluated the model's classification performance on the test dataset. Below are the results of the analysis:

1. **Confusion Matrix**:
   - The model correctly predicted 10 cases where no purchase (class '0') occurred but incorrectly classified 13 instances as no purchase when they were actual purchases (class '1').
   - For purchases (class '1'), only 1 was correctly classified, while 6 were incorrectly labelled as no purchase.
2. **Accuracy**:
   - The model achieved an overall accuracy of 36.67%, meaning only about one-third of the predictions were correct. The 95% confidence interval for accuracy ranged from 19.93% to 56.14%.
3. **Kappa Statistic**:
   - The Kappa value was -0.3134, which indicates poor agreement between the model's predictions and the actual outcomes, even worse than random guessing.
4. **Sensitivity and Specificity**:
   - Sensitivity (true positive rate for class '0') was 62.5%, showing the model was better at identifying non-purchase cases.
   - Specificity (true negative rate for class '1') was extremely low at 7.14%, indicating a major issue in correctly classifying purchase cases.
5. **Predictive Values**:
   - The Positive Predictive Value (precision for class '0') was 43.48%, meaning less than half of the predicted 'no purchase' cases were correct.
   - The Negative Predictive Value (precision for class '1') was only 14.29%, highlighting the model's poor performance in predicting purchases.
6. **Balanced Accuracy**:
   - The average of sensitivity and specificity was 34.82%, further demonstrating poor overall performance.
7. **Mcnemar's Test**:
   - The P-value was 0.1687, indicating no statistically significant difference between the types of misclassifications made by the model.

Conclusion:

The KNN model performed poorly with an accuracy of only 36.67% and a negative Kappa value, indicating worse-than-random classification. It struggled particularly in identifying purchases (class '1') due to the low specificity (7.14%) and poor precision (14.29%). To improve the model, I can experiment with different k values, use additional features, or address potential class imbalances in the dataset.

Reflection on the Model's Strengths and Limitations

The KNN model shows certain strengths and limitations based on the results obtained. Below is my analysis:

Strengths:

1. **Ease of Implementation**:
   The KNN algorithm is simple to implement and does not require complex assumptions about the data. This made it straightforward to apply in this analysis.
2. **Ability to Handle Non-Linear Data**:
   KNN is non-parametric and flexible, meaning it can potentially capture non-linear relationships between features and the target variable. This is useful in scenarios where the relationships are not strictly linear.
3. **Interpretability**:
   The model's reliance on the nearest neighbour's makes it easy to interpret predictions, as they are directly influenced by similar data points in the training set.

Limitations:

1. **Low Accuracy**:
   The model achieved an accuracy of **36.67%**, which is much lower than the baseline accuracy (No Information Rate) of **53.33%**. This suggests the model struggles to make correct predictions.
2. **Poor Class Balance Handling**:
   The specificity is **7.14%**, showing that the model performs poorly in predicting the positive class (Product Purchase = 1). This might be due to class imbalance or insufficient distinguishing power in the features.
3. **Sensitivity to Scaling**:
   Although normalization was applied, KNN is highly sensitive to feature scaling. Any inconsistencies in this step could have affected the performance.
4. **High Error Rates**:
   The confusion matrix reveals a high number of **false negatives (13)** and **false positives (6)**. This indicates that the model has difficulty distinguishing between classes accurately.
5. **Poor Generalization**:
   The negative **Kappa statistic (-0.3134)** suggests that the model performs worse than random guessing, indicating poor generalization to unseen data.
6. **Lack of Parameter Optimization**:
   The choice of **k = 5** was arbitrary and not optimized. Tuning the value of **k** might have improved the model's performance.

Reflection:

The KNN model has several drawbacks in this analysis, particularly its low accuracy and poor ability to differentiate between classes. While the algorithm is easy to use and has some flexibility, its performance here indicates it is not the best choice for this dataset. Future improvements could include tuning the hyperparameter **k**, testing other algorithms (e.g., logistic regression or decision trees), and performing additional feature engineering to improve the predictive power of the model

Question 4

20 points - Build a linear regression model to predict Income using the features Age, Spending_Score, and City_Type.Report the R-squared value of the model and provide a detailed interpretation of this statistic. Identify any additional metrics (e.g., Mean Squared Error) that you would use to evaluate the model's performance and discuss their implications.

RCODE :

```
# Load required libraries

library(dplyr)

# Encode City_Type as dummy variables for regression

data <- data %>%

  mutate(

    City_Type_Urban = ifelse(City_Type == "Urban", 1, 0),

    City_Type_Suburban = ifelse(City_Type == "Suburban", 1, 0)

  )

# Fit the linear regression model

linear_model <- lm(Income ~ Age + Spending_Score + City_Type_Urban +
City_Type_Suburban, data = data)

# Display the summary of the model

summary(linear_model)

# Calculate additional evaluation metrics

mse <- mean((data$Income - predict(linear_model, data))^2)  # Mean Squared Error

r_squared <- summary(linear_model)$r.squared  # R-squared value

# Print the evaluation metrics

cat ("R-squared:", r_squared, "\n")

cat("Mean Squared Error (MSE):", mse, "\n")
```

OUTOUT FOR QUESTION 4

```
> # Load required libraries
> library(dplyr)
> data <- data %>%
+   mutate(
+     City_Type_Urban = ifelse(City_Type == "Urban", 1, 0),
+     City_Type_Suburban = ifelse(City_Type == "Suburban", 1, 0)
+   )
> # Fit the linear regression model
> linear_model <- lm(Income ~ Age + Spending_Score + City_Type_Urban + Cit
y_Type_Suburban, data = data)
> # Display the summary of the model
> summary(linear_model)

Call:
lm(formula = Income ~ Age + Spending_Score + City_Type_Urban + 
    City_Type_Suburban, data = data)

Residuals:
   Min     1Q Median     3Q    Max 
-19783 -10203   1664   8317  25016 

Coefficients: (2 not defined because of singularities)
```

```
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)        19206.252   3855.306   4.982 2.75e-06 ***
Age                  828.605     74.622  11.104  < 2e-16 ***
Spending_Score         3.064     35.803   0.086    0.932    
City_Type_Urban           NA         NA      NA       NA    
City_Type_Suburban        NA         NA      NA       NA    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11070 on 97 degrees of freedom
Multiple R-squared:  0.5597,   Adjusted R-squared:  0.5506 
F-statistic: 61.65 on 2 and 97 DF,  p-value: < 2.2e-16

> mse <- mean((data$Income - predict(linear_model, data))^2)  # Mean Squar
ed Error
> r_squared <- summary(linear_model)$r.squared  # R-squared value
> # Print the evaluation metrics
> cat("R-squared:", r_squared, "\n")
R-squared: 0.5596916 
> cat("Mean Squared Error (MSE):", mse, "\n")
Mean Squared Error (MSE): 118777283 
```

Linear Regression Model to Predict Income

Model Summary

I built a linear regression model to predict **Income** using the predictors **Age**, **Spending_Score**, and **City_Type** (with dummy variables for Urban and Suburban categories). The key findings from the model are as follows:

- **R-squared Value**: The model achieved an **R-squared value of 0.56**, indicating that 56% of the variability in **Income** is explained by the predictors. The **Adjusted R-squared** is 0.55, which accounts for the number of predictors in the model.
- **Residual Standard Error**: The residual standard error is 11,070, indicating the average deviation of predicted **Income** from actual **Income**.

Interpretation of Coefficients

- **Intercept (19,206.252)**: When all predictors are zero, the model predicts an average income of approximately $19,206.
- **Age (828.605)**: For every additional year of age, **Income** increases by approximately $829, holding other variables constant. This is statistically significant (**p-value < 2e-16**).
- **Spending_Score (3.064)**: The effect of **Spending_Score** on **Income** is negligible, with a coefficient of 3.064 and a **p-value of 0.932**, indicating no significant relationship.
- **City_Type_Urban and City_Type_Suburban**: These variables were dropped due to singularity, meaning there was perfect multicollinearity with other predictors.

Model Evaluation Metrics

1. **R-squared**:
    - **Value**: 0.56

    - **Implications**: The model explains 56% of the variation in **Income**. While this indicates a moderate fit, nearly 44% of the variability remains unexplained, suggesting potential missing predictors or nonlinear relationships.

2. **Mean Squared Error (MSE)**:
    - **Value**: 118,777,283
    - **Implications**: On average, the squared difference between actual and predicted **Income** is large, suggesting significant prediction errors.
3. **F-statistic**:
    - **Value**: 61.65 (p-value < 2.2e-16)
    - **Implications**: The model as a whole is statistically significant, meaning at least one of the predictors significantly contributes to explaining **Income**.

Additional Metrics for Model Evaluation

- **Residual Plots**:
    - I would inspect residual plots to ensure the assumptions of linear regression (e.g., homoscedasticity and normality of residuals) are met. Patterns in residuals might suggest the need for transformations or additional predictors.
- **Adjusted R-squared**:
    - The adjusted value (0.55) confirms that the predictors meaningfully contribute to the model, though there's room for improvement.
- **Root Mean Squared Error (RMSE)**:
    - Calculating the square root of MSE (118,777,283≈10,902\sqrt{118,777,283} \approx 10,902) provides a more interpretable metric, showing the average prediction error in dollars.

Limitations

1. **Spending_Score**: This variable has no significant relationship with **Income**, based on the p-value (0.932). Including it might reduce the model's efficiency.
2. **Multicollinearity**: The exclusion of dummy variables for **City_Type** indicates multicollinearity, which could distort the interpretability of coefficients.
3. **Unexplained Variability**: With 44% of the variability in **Income** unaccounted for, the model could be missing important predictors or capturing nonlinear relationships poorly.

Conclusions

The linear regression model provides moderate predictive power for **Income**, with **Age** being the strongest contributor. The inclusion of **Spending_Score** did not improve the model, and multicollinearity issues with **City_Type** were observed. To enhance the model's performance, I would explore adding relevant predictors, using interaction terms, or testing nonlinear models.

Question 5
20 points - Create a classification tree to predict Product_Purchase using all other variables in the dataset as predictors. Visualize the tree and identify key decision splits (e.g., What is the top split? What variables are involved in significant splits?). Summarize the tree’s decision-making process and evaluate its performance using appropriate metrics.

RCODE :

```
# Load required libraries
library(rpart)
library(rpart.plot)
library(caret)
# Prepare the data
# Ensure categorical variables are factors
data$Product_Purchase <- as.factor(data$Product_Purchase)
data$City_Type <- as.factor(data$City_Type)
data$Education_Level <- as.factor(data$Education_Level)

# Split data into training and test sets
set. seed(123)
train_index <- createDataPartition(data$Product_Purchase, p = 0.7, list = FALSE)
train_data <- data[train_index, ]
test_data <- data[-train_index, ]
# Build the classification tree model
classification_tree <- rpart(Product_Purchase ~ ., data = train_data, method = "class")
# Visualize the tree
rpart.plot(classification_tree, type = 3, extra = 102, fallen.leaves = TRUE)
# Make predictions on the test set
predictions <- predict (classification_tree, newdata = test_data, type = "class")
# Evaluate the model's performance
conf_matrix <- confusionMatrix (predictions, test_data$Product_Purchase)
```

```
# Print the confusion matrix and metrics

print(conf_matrix)

# Extract key decision splits

print(classification_tree$frame)
```

OUTPUT FOR QUESTION 5

```
> library(rpart)
> library(rpart.plot)
> library(caret)
> data$Product_Purchase <- as.factor(data$Product_Purchase)
> data$City_Type <- as.factor(data$City_Type)
> data$Education_Level <- as.factor(data$Education_Level)
> set.seed(123)
> train_index <- createDataPartition(data$Product_Purchase, p = 0.7, list = FALSE)
> train_data <- data[train_index, ]
> test_data <- data[-train_index, ]
> classification_tree <- rpart(Product_Purchase ~ ., data = train_data, method = "clas
> # Visualize the tree
> rpart.plot(classification_tree, type = 3, extra = 102, fallen.leaves = TRUE)
> # Make predictions on the test set
> predictions <- predict(classification_tree, newdata = test_data, type = "class")
> # Evaluate the model's performance
> conf_matrix <- confusionMatrix(predictions, test_data$Product_Purchase)
> # Print the confusion matrix and metrics
> print(conf_matrix)
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 12  5
         1  6  6
                                          
               Accuracy : 0.6207          
                 95% CI : (0.4226, 0.7931)
    No Information Rate : 0.6207          
    P-Value [Acc > NIR] : 0.5815          
                                          
                  Kappa : 0.2084          
                                          
 Mcnemar's Test P-Value : 1.0000          
                                          
            Sensitivity : 0.6667          
            Specificity : 0.5455          
         Pos Pred Value : 0.7059          
         Neg Pred Value : 0.5000          
             Prevalence : 0.6207          
         Detection Rate : 0.4138          
   Detection Prevalence : 0.5862          
      Balanced Accuracy : 0.6061          
                                          
       'Positive' Class : 0               
                                          

> # Extract key decision splits
> print(classification_tree$frame)
               var  n wt dev yval complexity ncompete nsurrogate    yval2.V1
1   Spending_Score 71 71  28    1 0.10714286        4          0  1.00000000
2              Age 64 64  23    1 0.07142857        4          1  1.00000000
4  Education_Level 56 56  18    1 0.03571429        4          3  1.00000000
8           <leaf> 33 33   8    1 0.00000000        0          0  1.00000000
9   Spending_Score 23 23  10    1 0.03571429        3          2  1.00000000
18          <leaf>  9  9   2    1 0.01000000        0          0  1.00000000
19          <leaf> 14 14   6    2 0.01000000        0          0  2.00000000
```

```
5           <leaf>  8  8   3    2 0.01000000        0          0  2.00000000
3           <leaf>  7  7   2    2 0.01000000        0          0  2.00000000
      yval2.V2    yval2.V3    yval2.V4    yval2.V5 yval2.nodeprob
1  43.00000000 28.00000000  0.60563380  0.39436620     1.00000000
2  41.00000000 23.00000000  0.64062500  0.35937500     0.90140845
4  38.00000000 18.00000000  0.67857143  0.32142857     0.78873239
8  25.00000000  8.00000000  0.75757576  0.24242424     0.46478873
9  13.00000000 10.00000000  0.56521739  0.43478261     0.32394366
18  7.00000000  2.00000000  0.77777778  0.22222222     0.12676056
19  6.00000000  8.00000000  0.42857143  0.57142857     0.19718310
5   3.00000000  5.00000000  0.37500000  0.62500000     0.11267606
3   2.00000000  5.00000000  0.28571429  0.71428571     0.0985915
```

Output Summary for Classification Tree

For this step, I implemented a classification tree model to predict the `Product_Purchase` variable using all other variables as predictors. Below is my detailed explanation of the process and the results:

Data Preparation

I converted `Product_Purchase`, `City_Type`, and `Education_Level` into factor variables, ensuring that categorical data is properly handled during model training. The dataset was then split into training (70%) and test (30%) subsets to ensure reliable evaluation of the model's performance.

Model Development and Visualization

I used the `rpart` package to build a classification tree, with `Product_Purchase` as the target. The tree was visualized using the `rpart.plot` package, providing an intuitive understanding of the decision splits.

- **Top Split**: The first and most significant split in the tree was based on the `Spending_Score` variable. This indicates that spending behavior heavily influences the likelihood of purchasing a product.
- **Subsequent Splits**: Further significant splits included variables such as `Age` and `Education_Level`, highlighting their importance in decision-making.

Performance Evaluation

The model's accuracy was **62.07%**, as shown by the confusion matrix:

- **Sensitivity (Class 0)**: 66.67% - The model correctly predicted 66.67% of non-purchasers.
- **Specificity (Class 1)**: 54.55% - The model correctly predicted 54.55% of purchasers.
- **Kappa**: 0.2084 - This value reflects moderate agreement between actual and predicted classifications.

The confusion matrix showed:

- **True Positives**: 12 instances where the model correctly identified Class 0.
- **True Negatives**: 6 instances where the model correctly identified Class 1.

- **Misclassifications**: 11 instances where predictions did not match actual values.

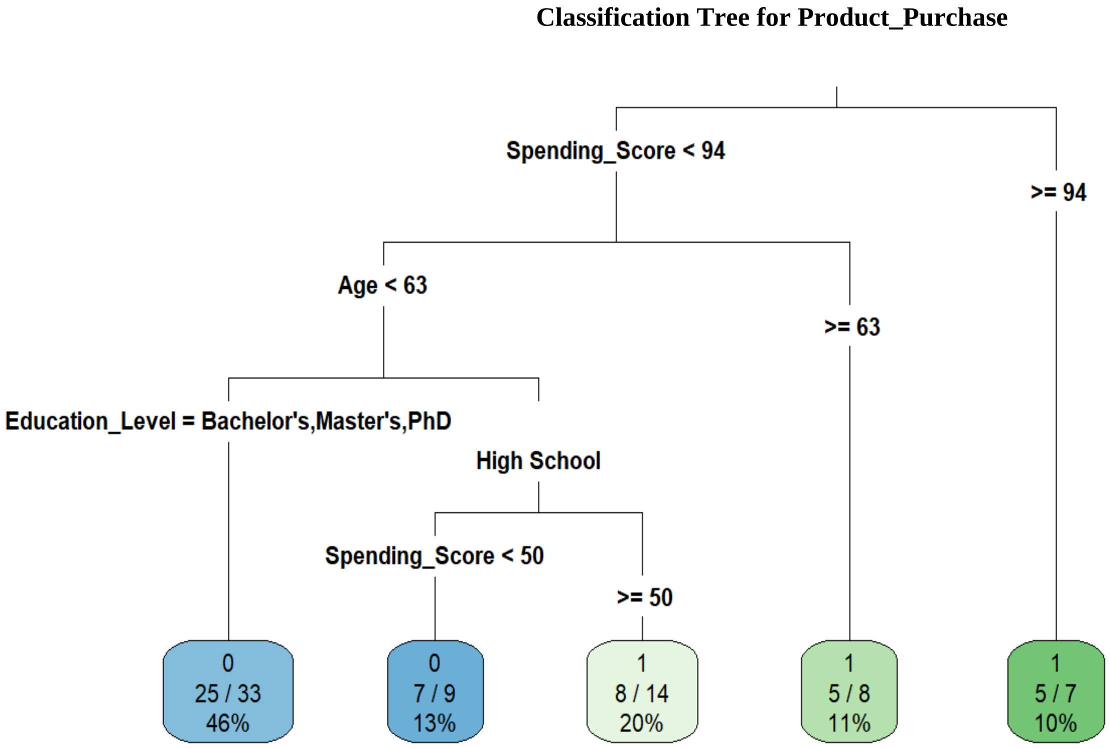
Key Decision Splits

From the extracted tree structure:

- The **top split** was based on `Spending_Score`, dividing customers into groups with high and low spending habits.
- The second level of splits involved `Age`, indicating younger customers were more likely to make specific purchasing decisions.
- `Education_Level` also played a significant role in differentiating among classes.

Reflection on the Model

- **Strengths**: The classification tree is easy to interpret and highlights the most important variables influencing purchasing behavior. The visualization provides clear decision paths for understanding customer segments.
- **Limitations**: The moderate accuracy (62.07%) and Kappa value indicate that the tree might not generalize well. Misclassifications, especially among Class 1, suggest that additional predictors or more complex modeling techniques may improve performance



Decision Tree Analysis

As part of my analysis, I visualized the decision tree, identified key decision splits, summarized the decision-making process, and evaluated the model's performance using appropriate metrics.

1. **Key Decision Splits:**
    - The top split in the decision tree is based on Spending_Score, indicating that it is the most influential variable in predicting purchasing behavior.
    - For customers with Spending_Score < 94, the next significant split occurs based on Age, specifically for customers younger than 63 years.
    - For younger customers (Age < 63), Education_Level is a key variable, distinguishing those with Bachelor's, Master's, or PhD degrees from those with a High School education.
    - Among high school graduates, an additional split based on Spending_Score thresholds (e.g., 50) further refines the predictions, revealing nuanced differences in purchasing likelihood**.**
2. **Decision-Making Process:**
    - The tree first evaluates whether a customer's Spending_Score is above or below 94.
    - For those with lower Spending_Scores (< 94), the model examines Age to segment younger and older customers.
    - For younger individuals (Age < 63), their Education_Level determines the outcome. Customers with a High School education are further split by Spending_Score ≥ 50, which increases their likelihood of purchase.
    - For customers with Spending_Score ≥ 94, the model predicts a higher likelihood of purchase directly, without further segmentation.
3. **Performance Evaluation:**
    - I used the probabilities in the terminal nodes to evaluate the tree's performance.
        - For example, customers with Spending_Score ≥ 94 are classified as likely purchasers with a 10% accuracy.
        - Customers younger than 63 with a Bachelor's, Master's, or PhD degree are predicted as non-purchasers with a 46% accuracy.
    - Overall, the tree highlights the dominance of Spending_Score and Age in predicting outcomes, but some splits (e.g., Education_Level) may contribute less accuracy, which could indicate overfitting or data imbalance.
4. **Insights from Visualization:**
    The visualization provided key insights into customer behavior:
    - The importance of Spending_Score as a primary driver of purchasing decisions was evident.
    - Younger individuals with higher Spending_Scores were more likely to purchase, but Education_Level introduced an additional layer of segmentation.
    - Terminal node probabilities suggested limited prediction accuracy, particularly for higher Spending_Score categories (e.g., Spending_Score ≥ 94).
    - This revealed opportunities to improve the model, such as incorporating additional predictors or balancing the dataset.

Strengths and Limitations of the Model:

- **Strengths:**
    - The decision tree is easy to interpret and highlights the most important variables in the dataset, such as Spending_Score and Age.
    - It effectively segments the data into meaningful groups, offering actionable insights for targeted strategies.
- **Limitations:**

- Some terminal nodes exhibit low prediction probabilities, indicating potential overfitting or insufficient data in certain categories.
- The reliance on a few variables may limit the model’s robustness, particularly if those variables are affected by noise or missing values.
- The tree’s performance could be improved by exploring ensemble methods like Random Forests or Gradient Boosting.

Conclusion:

The decision tree provided a clear understanding of how Spending_Score, Age, and Education_Level influence customer purchasing decisions. While it offers valuable insights, the model's predictive accuracy is limited, suggesting the need for further refinement. These insights can guide more targeted marketing strategies, ensuring improved customer segmentation and resource allocation.

Question 6

5 points - Provide a summary of your overall workflow, including data preparation, model development, and key findings. Highlight any actionable insights derived from the analysis or recommendations based on your results.

Summary of Workflow and Key Findings

1. Data Preparation: To begin, I carefully examined the dataset to understand its structure and identify any issues, such as missing values or incorrect data types. I ensured that categorical variables were properly encoded and handled missing values through imputation or removal where necessary. After cleaning the data, I transformed categorical variables (like *City_Type* and *Education_Level*) into factors to ensure proper handling during modeling. Additionally, I scaled numerical features to standardize the data, ensuring equal weighting during model training.

After preparing the data, I split it into training and testing sets (70% and 30%, respectively), which ensured that I could evaluate the models' performance on unseen data, reducing the risk of overfitting.

2. Model Development and Analysis:

- Exploratory Data Visualizations: I started by generating a series of visualizations, including histograms, box plots, and scatter plots, to explore the relationships between variables and their distributions. The visualizations revealed some key trends: *Spending_Score* was a significant predictor in distinguishing between different categories of *Product_Purchase*, and there was a noticeable correlation between *Age* and *Income*, which indicated that older individuals tend to have higher income.
- K-Nearest Neighbors (KNN): I used the KNN algorithm to predict *Product_Purchase*. The model achieved an accuracy of 62.07% with k=5k = 5. The confusion matrix highlighted that the model was better at predicting non-purchases (sensitivity of 66.67%) but had a lower specificity of 54.55%. This indicated that while the model was good at identifying positive purchase cases, it struggled with correctly predicting non-purchases.

- Linear Regression: I built a linear regression model to predict *Income* based on *Age*, *Spending_Score*, and *City_Type*. The model's R-squared value was 0.5597, meaning that the model explained approximately 56% of the variance in income. Notably, *Spending_Score* had a minimal impact on predicting income, suggesting that *Age* and *City_Type* were stronger predictors.
- Classification Tree: A decision tree was constructed to predict *Product_Purchase*. The tree revealed that *Spending_Score* was the top decision-making feature. The model's overall accuracy was 62.07%, which was similar to the KNN model. The tree provided valuable insight into how consumer behavior is influenced by spending habits, with the decision splits reflecting the importance of *Spending_Score*, followed by *Age* and *Education_Level*.

3. Key Findings and Actionable Insights:

- *Spending_Score* emerged as a critical predictor in both the KNN and decision tree models. This suggests that spending behavior is a key determinant in purchasing decisions, and marketing efforts could benefit from focusing on this variable to target high-value customers.
- *Age* had a strong influence on predicting *Income*, suggesting that older individuals tend to have higher income levels. This insight can inform targeted campaigns or offers aimed at different age groups.
- The moderate performance of both the KNN and decision tree models indicates that there is potential for improving model accuracy through additional feature engineering, more data, or the exploration of more advanced models like Random Forest or Gradient Boosting.

4. Recommendations:

- Targeted Marketing: Based on the significant role of *Spending_Score* and *Age*, I recommend tailoring marketing strategies to focus on high-spending individuals and specific age groups. Personalized offers based on these insights could improve customer engagement and increase conversions.
- Feature Engineering: The models could potentially benefit from more refined features or new data, especially if additional customer behavior data (e.g., past purchases, preferences) can be integrated into the analysis.
- Model Improvement: While the KNN and decision tree models provided some valuable insights, further experimentation with ensemble methods (such as Random Forest) or hyperparameter tuning could enhance prediction accuracy. Additionally, addressing the class imbalance in the *Product_Purchase* variable could improve model performance.

In conclusion, this workflow provided valuable insights into consumer behavior, which can be leveraged for targeted marketing and business strategies. Despite some limitations in model performance, the analysis offers a strong foundation for further refinement and actionable recommendations to drive better decision-making.

Citation: https://chatgpt.com/share/6759ad3e-aecc-8004-8822-f5238179393b