Marketing Campaign Prediction

2024-05-14

Table of contents

- **1.7 Recommendation**
- **References**

1.1 Introduction

In order to maintain a competitive edge and sustain their progress, many companies employ marketing campaigns as a strategy to engage with their customers, thereby enhancing their sales landscape and customer retention. Ascarza et al., (2018) delineated that customer retention refers to the continual interactions between customers and firms, a critical aspect considering that acquiring new customers may entail costs that are significantly higher, approximately five to six times, than retaining existing ones (Colgate & Danaher, 2000). To reduce expenses and enhance operational efficiency, numerous organizations are currently leveraging machine learning to forecast customer behavior. The integration of Artificial Intelligence (AI) into the corporate environment is of paramount importance in contemporary business settings. A multitude of enterprises are integrating AI technologies into their operations to augment their revenue streams. Essentially, this particular endeavor exemplifies the profound impact of machine learning, offering a comprehensive and invaluable perspective on the evolution of machine learing in accurately predicting customer responses to promotional offers based on individual customer traits.

The dataset used was retrieved from a Mysql database and underwent various stages of processing including preprocessing, cleansing, exploratory data analysis, feature manipulation, and model development in readiness for predictive analytics.

Load the required libraries

```
library(rpart)

library(rpart.plot)

library(DBI)
library(RMySQL)

library(caret)

library(pROC)

library(class)

library(ggplot2)
library(gridExtra)

library(dplyr)

library(cluster)

library(readr)

library(corrplot)

library(factoextra)
```

Setting the database user

```
DBUser <- 'root'
UserPassword <- 'harkinkunmsey'
HostName <- 'localhost'
DatabaseName <- 'world'
```

Extract customer campaign dataset from Mysql

```
database_con <- dbConnect(MySQL(), user = DBUser, password = UserPassword,
                host = HostName, dbname = DatabaseName, port=3306)

data <- dbGetQuery(database_con, statement = "Select * from
world.marketingcampaign")

dbDisconnect(database_con)

## [1] TRUE
```

1.2 Data Preprocessing and cleaning

The dataset comprises 22141 rows and 10 columns, with no missing data. It was uncovered that the majority of the participants are individuals of old age, with an average birth year of 1969 and an average annual income of 52514, ranging from a minimum of 1730 to a maximum of around 670000. The mean duration since the customer's most recent purchase is approximately 49 days, with an average of 5 web visits recorded last month. Additionally, the average customer has carried out 4 online purchases and roughly 6 in-store purchases.

Get first six rows of the data

```
head(data)

##      ID Year_Birth  Education MaritalStatus Income Recency NumWebPurchases
## 1 11000       1969 Graduation      Together  23228      71               2
## 2 11001       1963        PhD        Single  48918      21               1
## 3 11002       1951     Master       Married  67381      67               2
## 4 11003       1979 Graduation        Single  61825      56               4
## 5 11004       1969 Graduation       Married  44078      17               2
## 6 11005       1981 Graduation        Single  41967      66               1
##   NumStorePurchases NumWebVisitsMonth Response
## 1                 3                 8        0
## 2                 4                 4        0
## 3                 9                 7        0
## 4                 8                 4        0
## 5                 3                 5        0
## 6                 3                 4        0
```

data shape

```
dim(data)
```

```
## [1] 22141    10
```

summarise data

```
summary(data)
```

```
##        ID          Year_Birth    Education         MaritalStatus
##  Min.   :11000   Min.   :1893   Length:22141       Length:22141
##  1st Qu.:16592   1st Qu.:1959   Class :character   Class :character
##  Median :22197   Median :1970   Mode  :character   Mode  :character
##  Mean   :22198   Mean   :1969
##  3rd Qu.:27799   3rd Qu.:1978
##  Max.   :33399   Max.   :1996
##      Income          Recency      NumWebPurchases  NumStorePurchases
##  Min.   :  1730   Min.   : 0.00   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 35441   1st Qu.:24.00   1st Qu.: 2.000   1st Qu.: 3.000
##  Median : 51529   Median :49.00   Median : 4.000   Median : 5.000
##  Mean   : 52514   Mean   :48.78   Mean   : 4.103   Mean   : 5.801
##  3rd Qu.: 68682   3rd Qu.:73.00   3rd Qu.: 6.000   3rd Qu.: 8.000
##  Max.   :666666   Max.   :99.00   Max.   :27.000   Max.   :13.000
##  NumWebVisitsMonth    Response
##  Min.   : 0.000    Min.   :0.0000
##  1st Qu.: 3.000    1st Qu.:0.0000
##  Median : 6.000    Median :0.0000
##  Mean   : 5.317    Mean   :0.1532
##  3rd Qu.: 7.000    3rd Qu.:0.0000
##  Max.   :20.000    Max.   :1.0000
```

find missing observation

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

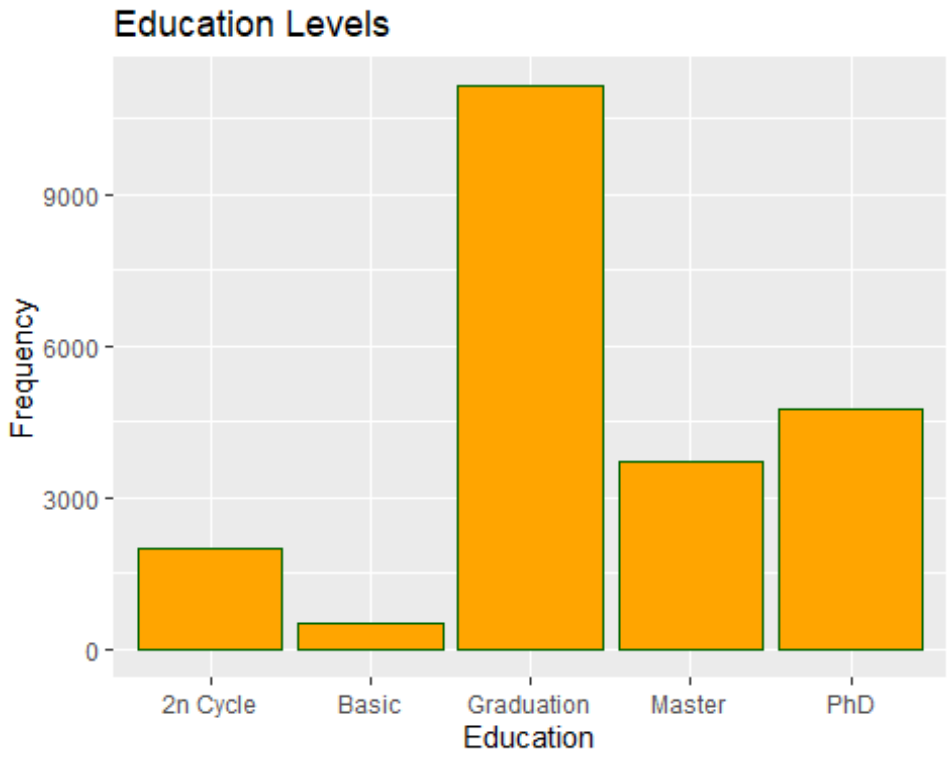
```
## [1] 0
```

Section 1.3 Project EDA

In as much as gaining a brief overview of the data is important, it has highly ensential that exploratory analysis is performed to gain in-depth insight about the intrinsic characteristics of the data at hand.

Level of Respondent Education Visualization

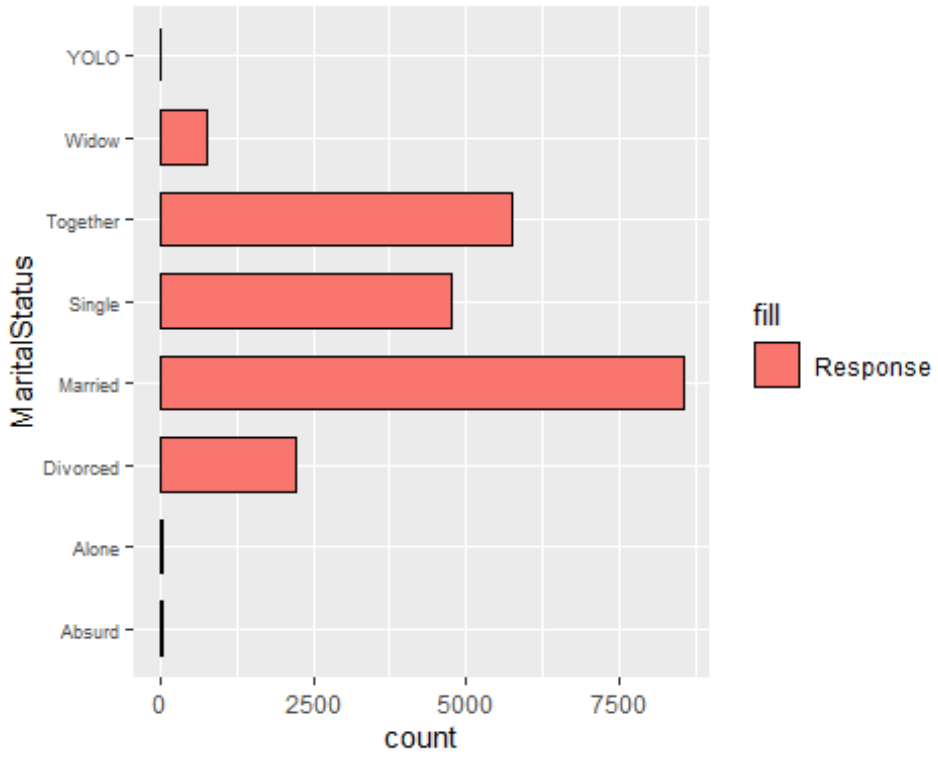
```
ggplot(data, aes(x = Education)) +
  geom_bar(color = "#006400", fill = "#FFA500")+
  labs(title = "Education Levels",
       x = "Education",
       y = "Frequency")
```



The findings of this analysis reveal that 9558 customers who have completed their education declined the campaign offer, in contrast to 1592 customers who did not. Additionally, 3751 customers with a Ph.D. declined the campaign offer, whereas only 3145 customers holding a Master's degree declined it. Interestingly, a higher rate of campaign offer decline was observed among graduate customers compared to those at other educational levels.

Marital Status to Campaign Response Visualization

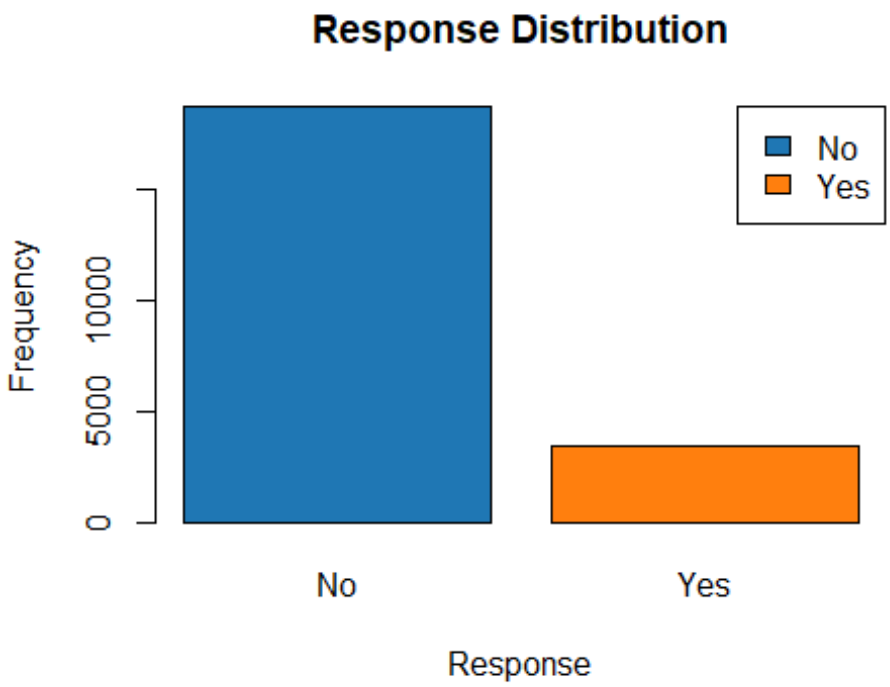
```
ggplot(data) + geom_bar(aes(x=MaritalStatus, fill = "Response"), color
="black", width = 0.65)+coord_flip() +
  theme(axis.text.y=element_text(size=rel(0.8)))
```



In relation to marital status, findings indicate a higher rate of refusal of the campaign proposal (7531) among married customers compared to other categories. Among those customers who are in relationships, 5164 declined the campaign offer, while 3653 of single customers also rejected it; in contrast, only 584 and 1111 of them, respectively, accepted the offer. However, the acceptance rate of the campaign offer is notably lower for customers with unconventional relationship statuses.

Campaign Response visualization

```
barplot(table(data$Response),
        col = c("#1F77B4", "#FF7F0E"),
        main = "Response Distribution",
        xlab = "Response",
        ylab = "Frequency",
        names.arg = c("No", "Yes"),
        border = "black",
        legend.text = c("No", "Yes"),
        args.legend = list(x = "topright"))
```



percentage of those that responded yes/no

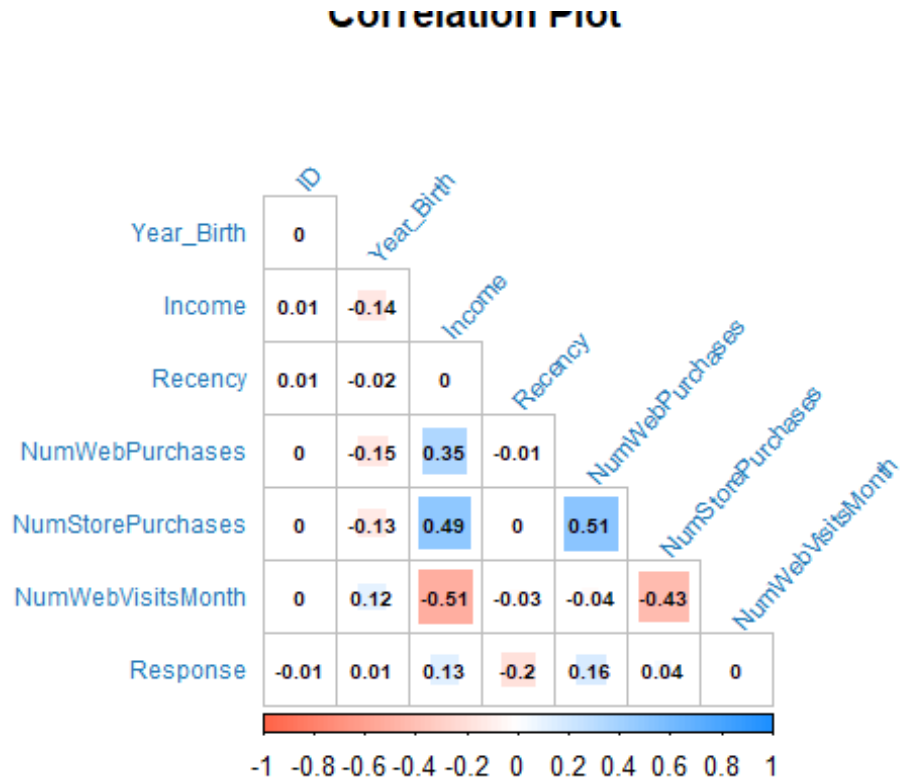
```
cbind(frequency = table(data$Response),
percent=prop.table(table(data$Response))*100)

##   frequency  percent
## 0     18748 84.67549
## 1      3393 15.32451
```

According to the analysis result, it was observed (18748)84.68% of the customers declined the campaign offer while on it was only (3393)15.32% of the customers that didn't which implies that there is class imbalance in the data set. However, a larger percentage of the customers rejected the campaign.

Corr heatmap

```
corr <- cor(data[,sapply(data, is.numeric)])
corrplot(corr,
         method = "square",
         type = "lower",
         tl.col = "#1F77B4",
         tl.srt = 47,
         diag = FALSE,
         addCoef.col = "black",
         col = colorRampPalette(c("#FF6347", "white", "#1E90FF"))(100),
         tl.cex = 0.75,
         number.cex = 0.6,
         tl.offset = 0.4,
```

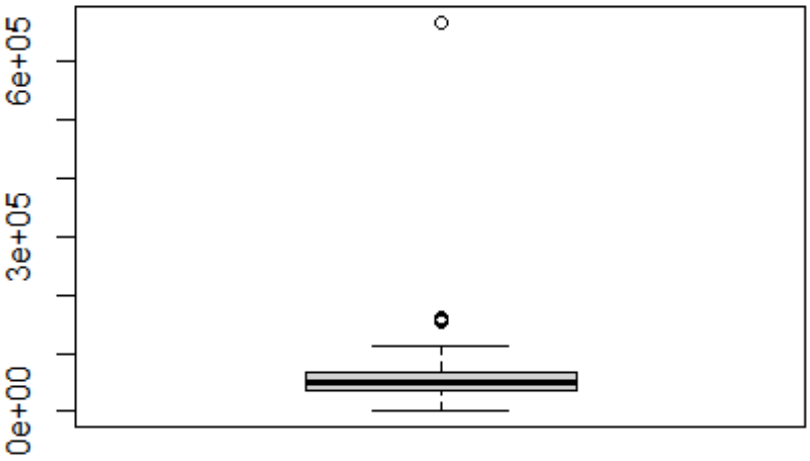
```
        addshade = "positive",
        title = "Correlation Plot"
)
```



Moreover, it was inferred that the duration since the last purchase made by the consumer, their birth year, and the frequency of visits to the organization's website in the preceding month do not affect the consumer's choice regarding an offer. Conversely, income, the number of online purchases, and in-store purchases are the variables that influence the consumer's reaction to a promotional offer, with online purchases being the most significant determinant. It was noted that some participants exhibited notably higher incomes than others, which could potentially negatively impact their response to the marketing campaign. Likewise, discrepancies were identified in the indicated birth years, as certain respondents erroneously recorded birth years before the year 1900.

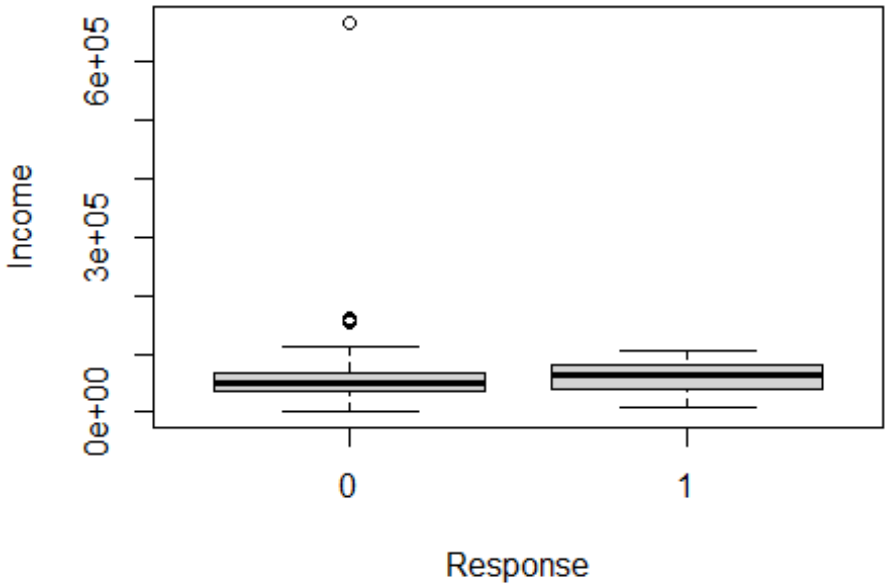
Check if there is outlier in income

```
boxplot(data$Income)
```



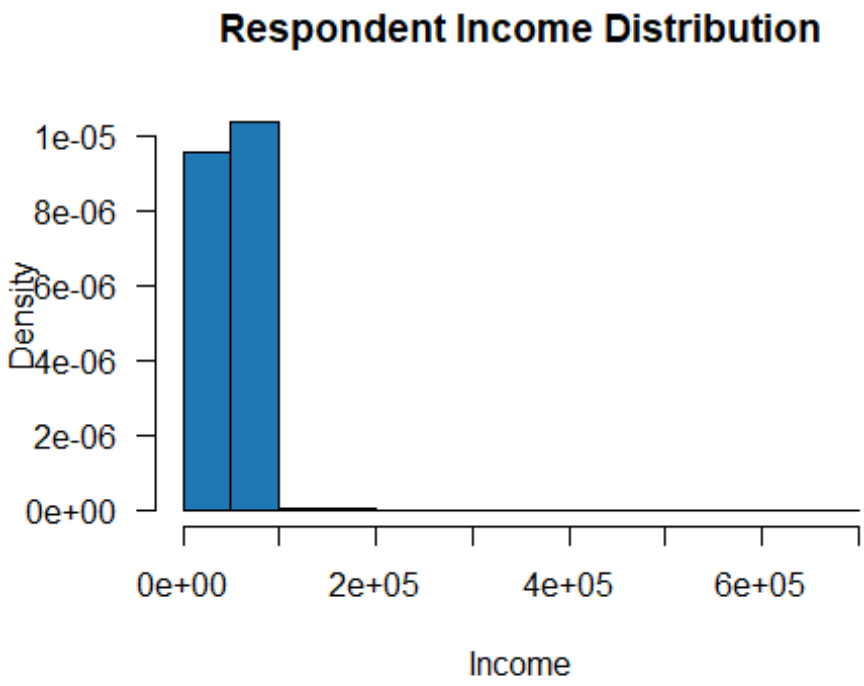
Inspect the response class with income outlier

```
boxplot(Income ~ Response, data = data)
```



Income distribution visual to double check the outlier

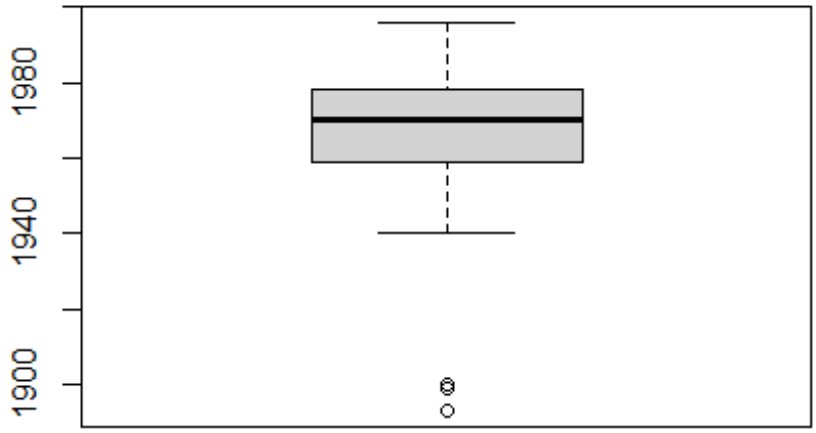
```
hist(data$Income,
     main = "Respondent Income Distribution",
     xlab = "Income",
     border = "black",
     col = "#1F77B4",
     prob = TRUE,
     las = 1
)
```



The histogram have confirmed the outlier in level of income, hence, the need for getting rid of this outlier.

Checking outlier in year of birth

```
boxplot(data$Year_Birth)
```



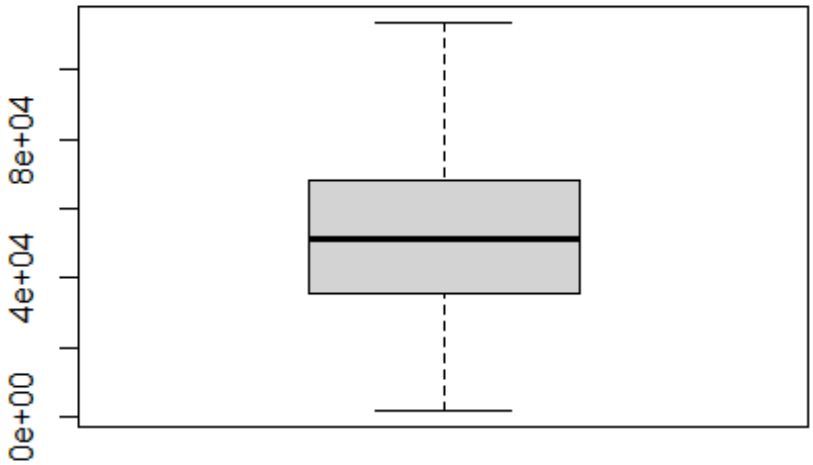
Removing outliers for both income and Age

Fixing outlier from the income column

```
qaunt <- quantile(data$Income, probs = c(0.25, 0.75))
# using the IQR function
inter_qt <- IQR(data$Income)

lb <- qaunt[1] - 1.5 * inter_qt
ub <- qaunt[2] + 1.5 * inter_qt

#create logical vector income_outliers where elements of the vector income
are outliers.
income_outliers <- data$Income < lb | data$Income > ub
data$Income[income_outliers] <- median(data$Income, na.rm = TRUE)
```

Inspect if the outlier has been removed

```
boxplot(data$Income)
```



Remove Outliers for BirthYear

```
data <- data[data$Year_Birth > 1900, ]
```

1.4 Data Feature Engineering

Convert the levels of the response variable to factor

```
data$Response <- ifelse(data$Response == 1, "Yes", "No")
# then make it a factor variable
data$Response <- factor(data$Response)
```

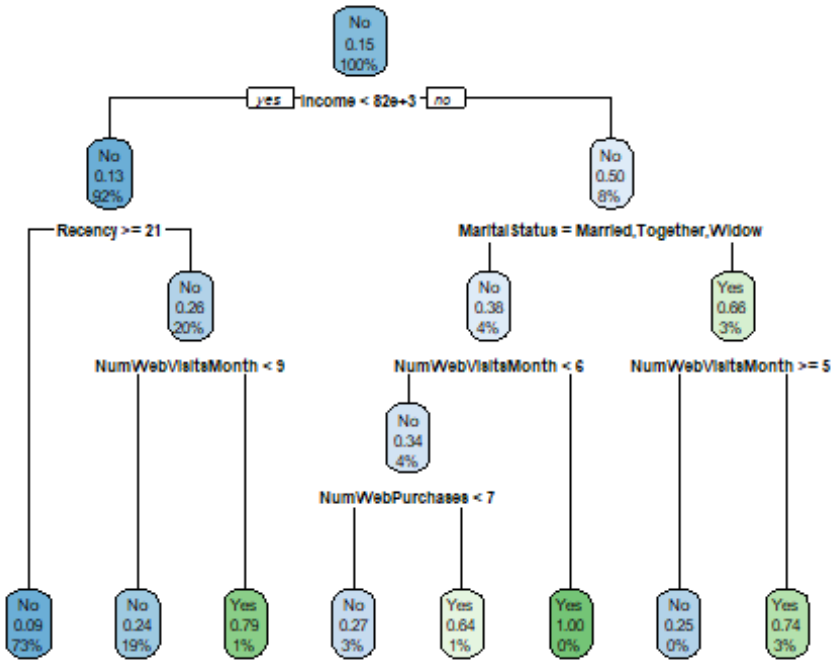
Drop ID Column

```
data <- dplyr::select(.data = data, -ID)
```

1.5 Project Model Building and Prediction

2.0 Model for Decision Tree

```
set.seed(123)
train_index <- sample(1:nrow(data), 0.7 * nrow(data))
train_data <- data[train_index, ]
test_data <- data[-train_index, ]

# Model Fitting
tree_model <- rpart(Response ~ ., data = train_data, method = "class")

# Model Plot
rpart.plot(tree_model)
```



• The root node of the decision tree indicates that the majority class (0) represents customers who did not accept the offer in the last campaign.

• The first split is based on Income, with customers having an Income less than $81,677.5 being more likely to accept the offer.

• Among customers with Income less than $81,677.5, the next split is based on MaritalStatus. Married, Together, and Widow customers are further split based on NumWebVisitsMonth, indicating higher acceptance rates for those with more web visits.

• Among customers with Income greater than or equal to $81,677.5, the split is based on MaritalStatus as well. Divorced and Single customers are further split based on Year_Birth, with younger customers being more likely to accept the offer.

Based on the decision tree analysis, the sales team can strategically decide on advertising targets by focusing efforts on specific customer segments. Customers with incomes below $81,677.5 demonstrate a higher likelihood of accepting offers, suggesting a priority focus on this demographic. Within this group, individuals who are Married, Together, or Widowed, particularly those engaging more frequently with the company's website, are more receptive. Moreover, targeting younger individuals among those with higher incomes (<1960.5 birth year) may yield favorable responses. Understanding customer behavior, particularly web engagement, is pivotal; customers visiting the company's website more frequently tend to be more receptive. Therefore, by tailoring advertising efforts towards these identified segments, the sales team can optimize campaign performance, maximizing success rates while minimizing costs.

Model predictions

```
predictions <- predict(tree_model, test_data, type = "class")
accuracy <- mean(predictions == test_data$Response)
print(paste("Accuracy:", accuracy))
```

```
## [1] "Accuracy: 0.864485981308411"
```

Decision Tree CF Matrix

```
test_y <- test_data$Response
caret::confusionMatrix(predictions, test_y, positive = "Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  5522  811
##        Yes   88  213
##
##                Accuracy : 0.8645
##                  95% CI : (0.856, 0.8726)
##     No Information Rate : 0.8456
##     P-Value [Acc > NIR] : 8.501e-06
##
##                   Kappa : 0.2703
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.20801
##             Specificity : 0.98431
##          Pos Pred Value : 0.70764
##          Neg Pred Value : 0.87194
##              Prevalence : 0.15436
##          Detection Rate : 0.03211
```

```
##    Detection Prevalence : 0.04537
##       Balanced Accuracy : 0.59616
##
##        'Positive' Class : Yes
##
```

The decision tree model attained an accuracy of around 86.46% on the test dataset, demonstrating its efficacy in forecasting customer response within the marketing campaign. Nonetheless, its efficiency and dependability in this particular study are questionable due to potential bias resulting from class imbalance. This assertion is supported by Ling et al., (2003), who argued that assessment metrics like accuracy, precision, recall, and F-score may not be suitable for analyzing imbalanced datasets. Furthermore, one could employ metrics like sensitivity, specificity, and precision. The outcomes from the confusion matrix reveal that sensitivity is relatively low at 20.80%, indicating difficulty in accurately identifying customers likely to accept the offer. Conversely, specificity is notably high at 98.43%, showing the model's ability to recognize customers who would decline the offer. Moreover, precision is at a moderate level of 70.76%, suggesting the model's confidence in predicting customer acceptance of the campaign offer. The low sensitivity implies a potential oversight of numerous acceptance cases, while the high specificity and positive predictive value underscore the model's capability in identifying customers likely to reject the offer.

Thus, the model's effectiveness resides in forecasting a customer's absence of active response to an offer.

Decision Tree ROC and AUC

```
set.seed(123)
tree_Control <- trainControl(method="cv", number=2, classProbs=TRUE,
                             summaryFunction=twoClassSummary)

treeTrain <- train(Response~., data = train_data, method="rpart",
metric="ROC",
             trControl=tree_Control)

treePred <- predict(treeTrain, newdata = test_data, type="prob")

treePred_probabilities <- treePred[, 2]

treeLabels <- ifelse(test_y == "Yes", 1, 0)

Roc <- roc(treeLabels, treePred_probabilities)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

aucValue <- auc(Roc)
print(aucValue)
```

```
## Area under the curve: 0.691
```

As it was mentioned earlier that there is class imbalance in the data with decision not based solely on accuracy metric, using the roc_auc measure in decision-making is one of the greatest approaches to solve this imbalance.

With an AUC of 59.8%, the decision tree model outperforms random chance in its ability to distinguish between customers who are likely to respond favorably to a campaign offer and those who are not. There isn't much of a difference between these two groups, though. This indicates that the 59.8% prediction accuracy is not very high, and hence, this statistic should not be the only one used to make business decisions.

3.0 Model for Logistic Model

logistic model

```
logit_mod <- glm(Response ~ ., data = train_data, family = "binomial")
```

A logistic regression model was applied to the training dataset utilizing the glm function in R, which is well-suited for addressing binary classification tasks. The model was trained to predict the probability of a customer accepting the offer based on predictor variables such as Education, MaritalStatus, Income, Recency, NumWebPurchases, NumStorePurchases, and NumWebVisitsMonth.

Get exponential of the coeficient for better interpretation

```
exp(coef(logit_mod))
```

```
##           (Intercept)            Year_Birth        EducationBasic
##          2.474891e-16          1.016375e+00          1.054308e+00
##   EducationGraduation       EducationMaster          EducationPhD
##          1.611234e+00          1.887455e+00          2.720334e+00
##    MaritalStatusAlone MaritalStatusDivorced  MaritalStatusMarried
##          7.547188e-01          3.700029e-01          1.933461e-01
##   MaritalStatusSingle MaritalStatusTogether    MaritalStatusWidow
##          5.009558e-01          1.692755e-01          5.815501e-01
##     MaritalStatusYOLO                Income               Recency
##          9.023167e-01          1.000062e+00          9.773349e-01
##       NumWebPurchases     NumStorePurchases     NumWebVisitsMonth
##          1.089891e+00          8.286788e-01          1.247139e+00
```

These coefficients offer insightful information about the relative contributions of each predictor variable to the model's ability to forecast how a client will react to a marketing offer. According to the model, a person's likelihood of accepting a campaign offer is greatly raised by a number of criteria, including having more recent interactions, being unmarried or young at heart, having a greater income, and having higher education levels—especially a PhD or master's degree.

On the other hand, characteristics like older age, low educational attainment, and less in-store transactions are linked to a decreased chance of accepting the campaign offer.

Model Predictions

```
test_data$Probs <- predict(logit_mod, test_data, type = "response")
test_data$Pred <- ifelse(test_data$Probs > 0.5, "Yes", "No")
test_data$Pred<- factor(test_data$Pred, levels = c("No", "Yes"))
```

The outcome of the model demonstrates an 85.3% probability of accurately determining if a customer will agree or reject a campaign offer. It demonstrates the model's strength in predicting customer responses. It is advisable to consider other metrics as decisions should not rely solely on accuracy due to class imbalance. The model exhibits a relatively low sensitivity (15.72%), meaning it may not accurately identify many customers who would accept the offer. However, it excels in specificity (98%) by recognizing customers likely to decline the offer. The precision is moderate (58.97%), signifying a 58.97% chance of a customer accepting the offer.

Simply put, the model needs improvement in sensitivity, possibly at the expense of specificity, to capture more potential acceptances. The AUC result (79%) indicates efficient differentiation between positive and negative classes, showcasing good discriminatory capability. This suggests the model outperforms random chance in distinguishing customers who will accept the campaign offer, instilling 79% confidence in making effective distinctions. Additionally, this logistic regression model is suitable for decisions concerning customer campaign offers.

Logistic Regression Confusion Matrix

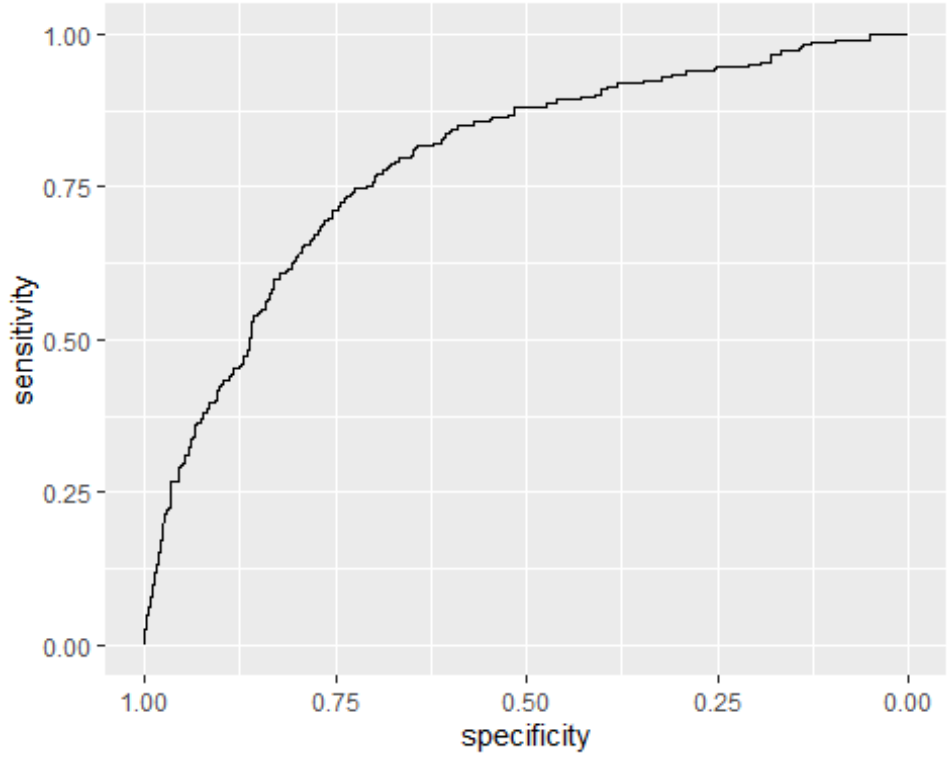
```
caret::confusionMatrix(test_data$Pred, test_y, positive = "Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  5498  863
##        Yes  112  161
##
##                Accuracy : 0.853
##                  95% CI : (0.8443, 0.8615)
##     No Information Rate : 0.8456
##     P-Value [Acc > NIR] : 0.04896
##
##                   Kappa : 0.196
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.15723
##             Specificity : 0.98004
##          Pos Pred Value : 0.58974
##          Neg Pred Value : 0.86433
##              Prevalence : 0.15436
```

```
##          Detection Rate : 0.02427
##    Detection Prevalence : 0.04115
##       Balanced Accuracy : 0.56863
##
##        'Positive' Class : Yes
##
```

Logistic Regression ROC and AUC

```
set.seed(123)
probabilities <- predict(logit_mod, test_data, type = "response")

log_roc <- roc(test_y, probabilities)

## Setting levels: control = No, case = Yes

## Setting direction: controls < cases

auc <- auc(log_roc)
ggroc(log_roc)
```




```
print(paste("AUC:", auc))

## [1] "AUC: 0.789765154996658"
```

The logistic regression model did a fantastic job in achieving the objective of predicting the likelihood of a specific customer accepting a marketing campaign offer. With a notable AUC

of approximately 0.789 observed in the test dataset, the model demonstrated its capacity to differentiate between customers inclined to accept the offer and those less inclined to do so.

This shows that the model can efficiently prioritize customers based on their chances of accepting the offer, offering valuable insights for targeted marketing strategies. As a result, the model's performance is in line with the aim of predicting customer acceptance probabilities directly from the data.

4.0 Model for K-Nearest Neighbor

Model Preparation for K-Nearest Neigbor(KNN)

```
set.seed(123)
train_indices <- sample(1:nrow(data), 0.8*nrow(data))
train_data <- data[train_indices, ]
test_data <- data[-train_indices, ]

# convert the levels of education and marital status to a dummy variable
train_data <- data.frame(model.matrix(~ Education + MaritalStatus + 0, data =
train_data), train_data)
test_data <- data.frame(model.matrix(~ Education + MaritalStatus + 0, data =
test_data), test_data)

# select all clean data
train_data <- train_data[, -c(which(names(train_data) %in% c("Education",
"MaritalStatus")))]
test_data <- test_data[, -c(which(names(test_data) %in%
c("Education","MaritalStatus")))]
num_cols <-
c("Year_Birth","Income","Recency","NumWebPurchases","NumStorePurchases","NumW
ebVisitsMonth")

# select all clean data
train_data <- train_data[complete.cases(train_data), ]
test_data <- test_data[complete.cases(test_data), ]

# numerical column scaling
train_data[, num_cols] <- scale(train_data[, num_cols])
test_data[, num_cols] <- scale(test_data[, num_cols])
```

The data has been prepared to get it all set for the KNN model. This preparation involved dividing the data into training and testing sets, transforming categorical variables (Education and Marital Status) using one-hot encoding, and standardizing numerical characteristics to guarantee consistency in feature scales across training and testing data.

For the training data, all features were considered except for the target variable (Response), which was utilized as the class variable. The model was set up with k=5, which means that predictions were made based on the five closest neighbors.

KNN Model fitting and Prediction

```
set.seed(12)
Model <- knn(train = train_data[, -which(names(train_data) == "Response")],
                 test = test_data[, -which(names(test_data) == "Response")],
                 cl = train_data$Response, k = 5)

accuracy <- sum(Model == test_data$Response) / length(test_data$Response)
print(paste("Accuracy: ", accuracy))

## [1] "Accuracy:  0.978747456477504"
```

K-nearest neighbor level of accuracy at 97.9% implies that the model's predictions are extremely precise. It signifies that the model accurately predicts customer reactions to the campaign offer with a great degree of certainty. Moreover, the confusion matrix reveals that both specificity (95.06%) and sensitivity (98.40%) are remarkably high, suggesting that the model excels at recognizing both positive and negative responses. Additionally, the model effectively anticipates a 99.% (precision) decrease in offer acceptance, indicating its confidence in predicting offer rejections.

KNN confusion matrix

```
confusionMatrix(data = Model, reference = test_data$Response)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  3695   34
##        Yes   60  634
##
##                Accuracy : 0.9787
##                  95% CI : (0.9741, 0.9828)
##     No Information Rate : 0.849
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9184
##
##  Mcnemar's Test P-Value : 0.009922
##
##             Sensitivity : 0.9840
##             Specificity : 0.9491
##          Pos Pred Value : 0.9909
##          Neg Pred Value : 0.9135
##              Prevalence : 0.8490
##          Detection Rate : 0.8354
##    Detection Prevalence : 0.8431
##       Balanced Accuracy : 0.9666
```

```
##
##        'Positive' Class : No
##
```

KNN ROC and AUC

```
set.seed(123)
knn_Rocmodel <- train(form = Response ~.,
                      data = train_data,
                      method = 'knn')

knn_predictions <- predict(object = knn_Rocmodel, newdata = test_data, type =
"prob")
knn_probabilities<- knn_predictions[,2]
KnnROC <- roc(response = test_data$Response, predictor = knn_probabilities)

## Setting levels: control = No, case = Yes

## Setting direction: controls < cases

knnAuc <- auc(KnnROC)
print(knnAuc)

## Area under the curve: 0.9986
```

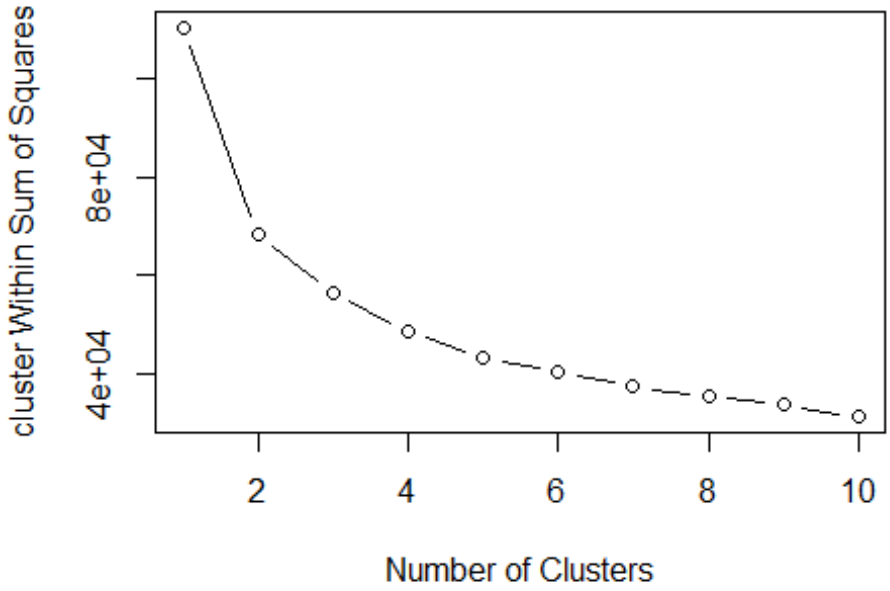
The AUC value of the model, which is an impressive 99.86%, showcases its exceptional ability to differentiate effectively. This indicates that the model is very good at distinguishing between positive and negative classes, with a high level of accuracy. It ranks a randomly selected positive instance higher than a randomly chosen negative instance 99.86% of the time. The AUC of the KNN model also demonstrates its excellent predictive capabilities, showing that it can reliably generate forecasts. With such a high AUC value, it is clear that the model is accurate, reliable, and effective in identifying customers who are likely to either accept or reject the campaign proposal.

5.0 Project Cluster Analysis

```
data_variables <- c("Income", "Recency", "NumWebPurchases",
"NumStorePurchases", "NumWebVisitsMonth")


# data scaling
s_data <- scale(data[, data_variables])

#  total within-cluster sum of square (1 to 10) clusters for optimal number
of cluster
WSS <- numeric(10)
for ( i in 1:10) {
  WSS[i] <- sum(kmeans(s_data, centers = i)$withinss)
}
```

Using the kmeans algorithm to determine relevant number of k

```
plot(1:10, WSS, type = "b", xlab = "Number of Clusters", ylab = "cluster
Within Sum of Squares")
```



In anticipation of conducting clustering analysis, a set of five critical attributes comprising Income, Recency, NumWebPurchases, NumStorePurchases, and NumWebVisitsMonth were specifically chosen for the clustering process.

To ensure an equitable contribution of each attribute to the clustering procedure, the data underwent a scaling process. The determination of the optimal number of clusters was carried out by assessing the within-cluster sum of squares (WSS) across a range of cluster quantities using the elbow method.

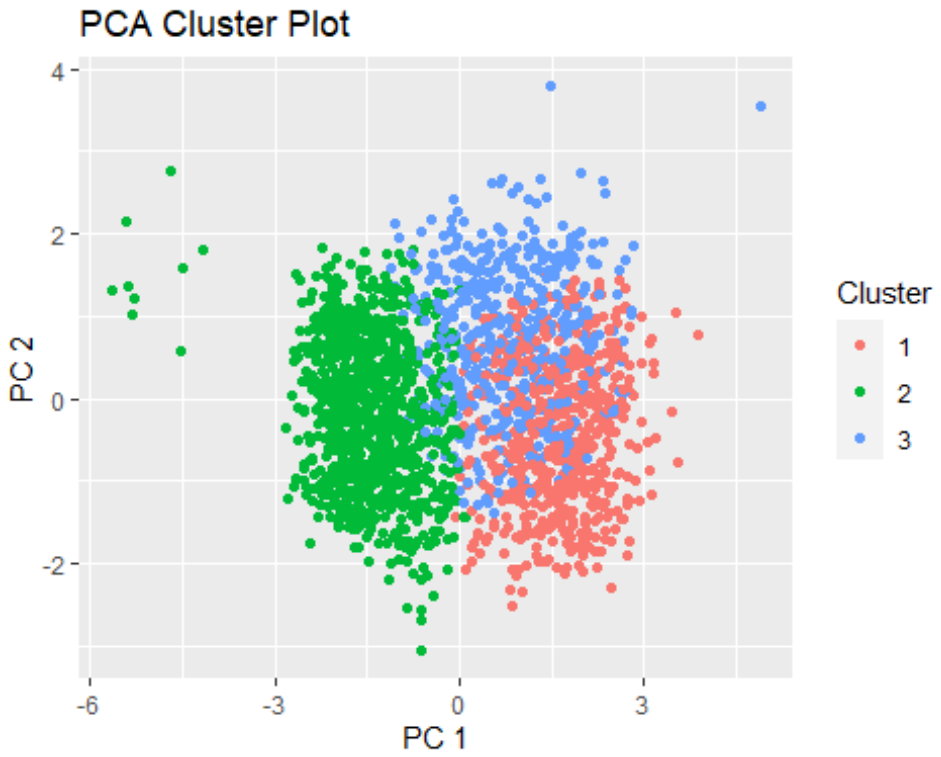
The graphical representation of WSS plotted against the cluster count unveiled a distinctive bend at k=4, indicating that the most suitable configuration for categorizing the data would entail four clusters.

Subsequently, the k-means clustering technique was executed on the standardized data with k=4. This method facilitated the identification of four cluster centers, serving as the focal points within the multi-dimensional feature space. Allocation of each data point to the closest cluster was based on the proximity to these cluster centers. The visualization aspect was addressed through the utilization of PCA.

```
KmeansModel <- kmeans(s_data, centers = 3)
data$cluster <- KmeansModel$cluster
KmeansModel$centers
```

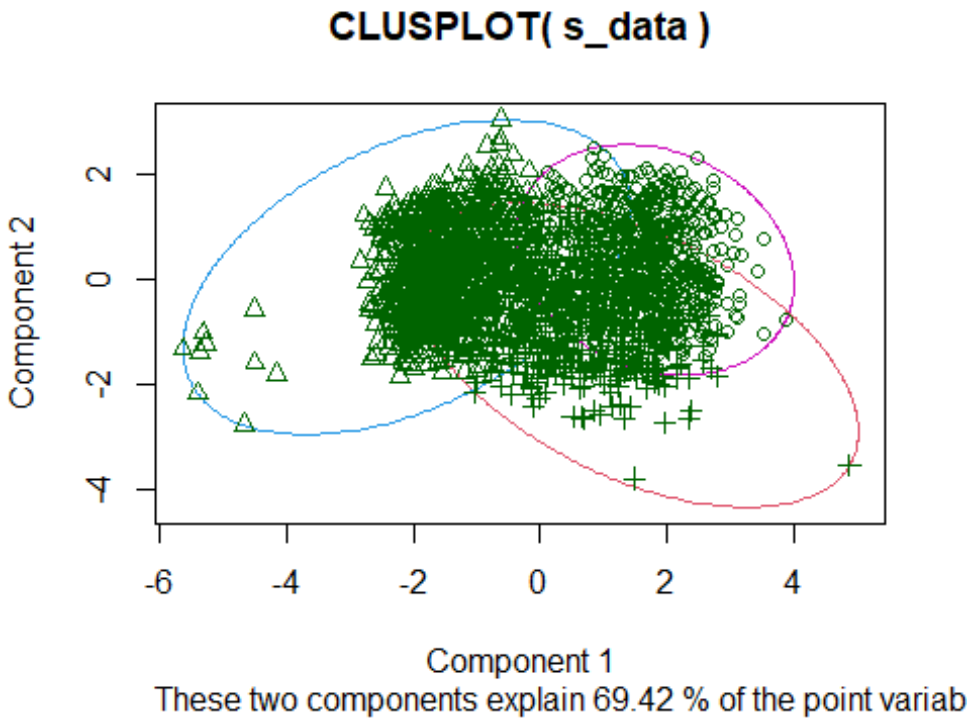
```
##       Income      Recency NumWebPurchases NumStorePurchases
NumWebVisitsMonth
## 1  1.0393413  0.046802395       0.1034245         0.8471096        -
1.1642443
## 2 -0.8447615  0.004091853      -0.7308031        -0.8090752
0.4959118
## 3  0.3666800 -0.062137354       1.2437777         0.5239075
0.4299703
```

```r
pca_data <- prcomp(s_data, center = TRUE, scale. = TRUE)
pca_df <- as.data.frame(pca_data$x[, 1:2])
pca_df$Cluster <- data$cluster
ggplot(pca_df, aes(x = PC1, y = PC2, color = as.factor(Cluster))) +
  geom_point() +
  scale_color_discrete(name = "Cluster") +
  labs(x = "PC 1", y = "PC 2") +
  ggtitle("PCA Cluster Plot ")
```



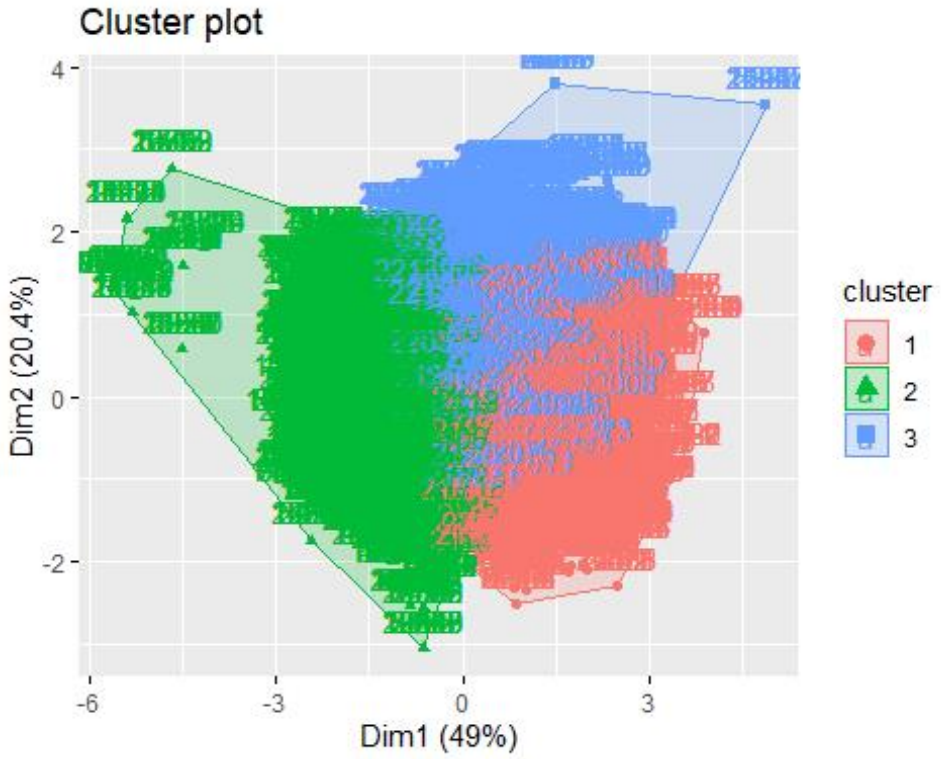
visual of the kmean cluster

```r
clusplot(s_data, KmeansModel$cluster, color = TRUE)
```



cluster visualization

```
fviz_cluster(KmeansModel, data = s_data)
```



cluster Members

```
dataDistance <- dist(x = s_data, method = "euclidean")
Hc_Data <- hclust(d = dataDistance, method = "complete")
cluster_mem = cutree(Hc_Data,3)
table(cluster_mem)

## cluster_mem
##     1     2     3
## 21992    30    89

aggregate(s_data,list(cluster_mem),mean)

##   Group.1      Income      Recency NumWebPurchases NumStorePurchases
## 1       1  0.01068181  0.001139378    -0.004511556       0.009282846
## 2       2 -1.16826300 -0.140822874     7.413658433      -1.656759839
## 3       3 -2.24569018 -0.234073137    -1.384175383      -1.735343288
##   NumWebVisitsMonth
## 1       -0.01710786
## 2       -1.97686340
## 3        4.89373099
```

Key Findings from Cluster Analysis

Explanation in Business Terms

The analysis of customer segments through K-means clustering offers valuable insights into customer behavior and preferences. Each segment demonstrates unique features that can be utilized to develop targeted marketing strategies and enhance the effectiveness of campaigns. The following insights are presented:

- Cluster 1: This category comprises customers with relatively higher incomes and a moderate level of engagement, as indicated by their NumWebPurchases, NumStorePurchases, and NumWebVisitsMonth. These customers display average behavior across all metrics, lacking a strong preference for web or store purchases and exhibiting moderate engagement with both channels (store and websites).

- Cluster 2: Customers in this segment possess lower incomes and a recent history of interaction with the company, reflected in their Recency. Nevertheless, their overall purchasing activity is lower compared to other segments. Additionally, customers show high involvement in online shopping but demonstrate minimal interest in visiting physical stores. They belong to lower income brackets and engage in frequent web purchases.

- Cluster 3: This group comprises customers with notably low incomes and limited interaction with the company across all measured parameters. Customers in this cluster visit the website frequently but exhibit low conversion rates for both web and store purchases.

Conclusion

Based on the examination of all four models, it can be inferred that the decision-making process concerning campaign offers in business can rely on the KNN model. This specific model distinguishes itself through its capacity to accurately discern between positive and negative categories, particularly by prioritizing a randomly chosen positive example over a randomly selected negative example 99.86% of the time, While the logistic regression model also demonstrates commendable performance by prioritizing customers based on their likelihood of accepting the offer and providing valuable data insights, its efficacy falls short when compared to that of the KNN model. Additionally, the cluster analysis offers invaluable insights into customer behavior, which can be leveraged to tailor campaign offers to specific customers.

Recommendations

Based on the deductions drawn from the exploration and model outcomes, the subsequent recommendations should be implemented:

- Sustain engagement through well-rounded and personalized campaigns to enhance the sensitivity of the logistic regression model.
- Improve online user experiences and encourage occasional visits to physical stores.
- Transform high web traffic into actual purchases by offering targeted incentives and compelling content.

Taking into account these recommendations will contribute to the improvement of customer satisfaction, heightened engagement, and increased conversion rates, thereby enhancing the overall effectiveness of campaign offers.

References

- E. Ascarza Retention futility: Targeting high-risk customers might be ineffective Journal of Marketing Research (2018).
- Ling, C.X., Huang, J., Zang, H.: Auc: a better measure than accuracy in comparing learning algorithms. Canadian Conference on AI (2003) 329–341.
- M.R. Colgate et al.Implementing a customer relationship strategy: The asymmetric impact of poor versus excellent execution Journal of the Academy of Marketing Science (2000).