MARKETING CAMPAIGN EFFECTIVENESS ANALYSIS

A MINI-PROJECT REPORT

Submitted by

SAROSHMI B 211701047
SHARON STEVE J 211701050

In partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND DESIGN

RAJALAKSHMI ENGINEERING COLLEGE, THANDALAM

ANNA UNIVERSITY: CHENNAI 600 025

NOVEMBER 2024

RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY: CHENNAI 602105

BONAFIDE CERTIFICATE

Certified that this project report **“MARKETING CAMPAIGN EFFECTIVENESS ANALYSIS”** is the bonafide work of **“SAROSHMI B (211701047) & SHARON STEVE J (211701050)”** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Mr. S. Uma Maheswara Rao	Dr. P. Revathy, M.E., Ph.D.,
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Associate Professor	Professor
Department of	Department of
Computer Science and Design	Computer Science and Design
Rajalakshmi Engineering College	Rajalakshmi Engineering College
Chennai - 602105	Chennai - 602105

Submitted to project viva-voce examination for the subject CD19P10 Foundations of Data Science held on ______________________

Internal Examiner	**External Examiner**

ABSTRACT

Marketing Campaign Effectiveness analyses customer data to uncover insights that drive marketing effectiveness and enhance customer segmentation strategies. The dataset includes demographic information, purchase behaviour, and promotional engagement, providing a rich basis for evaluating customer preferences and responses to marketing efforts over time. Key objectives include examining the correlation between customer attributes (such as age, income, and household composition) and spending patterns across various product categories, including wines, meats, and luxury goods. By analysing these trends, the project seeks to identify high-value customer segments and understand their unique purchasing behaviours. Another focal area is the assessment of promotional campaign success. Using customer responses to five previous campaigns, along with engagement metrics like purchase frequency and recency, this project will evaluate which campaigns were most effective and why. Additionally, analysing channel-specific data—such as web, catalogue, and in-store purchases—will help determine optimal platforms for customer outreach. Machine learning techniques will be applied to build predictive models for campaign responsiveness, aiming to improve targeting strategies. Ultimately, this analysis will generate actionable insights, allowing the marketing team to personalize campaigns, optimize resource allocation, and boost customer retention, satisfaction, and lifetime value. This solution is designed to empower marketing teams with actionable intelligence, aligning offers with customer preferences to enhance satisfaction and drive sustained revenue growth.

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavour to put forth this report. Our sincere thanks to our Chairman **Mr. S.Meganathan, B.E, F.I.E.,** our Vice Chairman **Mr. Abhay Shankar Meganathan, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) Thangam Meganathan, Ph.D.,** for providing us with the requisite infrastructure and sincere endeavouring in educating us in their premier institution.

Our sincere thanks to **Dr.S.N.Murugesan, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Mr.S.Uma Maheswara Rao.,** Professor and Head of the Department of Computer Science and Design for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide and project coordinator **Dr.P.Revathy, M.E., Ph.D.,** Professor, Department of Computer Science and Design, Rajalakshmi Engineering College for her valuable guidance throughout the course of the project.

SAROSHMI B (211701047)

SHARON STEVE J (211701050)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	**EXPANSION**
CSV	Comma Separated Values
ROI	Return on Investment
IQR	Inter Quartile Range
EDA	Exploratory Data Analysis
SMOTE	Synthetic Minority Oversampling Technique
CV	Cross Validation
TN	True Negative
TP	True Positive
FN	False Negative
FP	False Positive

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Evaluating the effectiveness of marketing campaigns is crucial for optimizing resource allocation and maximizing return on investment (ROI). In today's competitive landscape, businesses invest heavily in diverse campaigns, ranging from digital advertising to traditional media, making it imperative to measure their impact accurately. Analyzing effectiveness enables companies to understand customer behavior, improve targeting strategies, and achieve better conversion rates. Current trends in marketing analytics emphasize data-driven decision-making, leveraging tools such as predictive modeling, machine learning, and customer segmentation. The rise of omnichannel marketing has also necessitated a holistic approach to measuring campaign success across platforms. Key metrics, including click-through rates, customer lifetime value, and sentiment analysis, are central to crafting more effective strategies.

1.2 SCOPE OF THE PROJECT

This project evaluates the effectiveness of marketing campaigns by analyzing customer behavior, product preferences, promotional responses, and purchasing patterns. The dataset includes detailed customer demographics, such as age, marital status, education level, income, and household composition, providing insights into the profiles of campaign participants. The analysis focuses on customer spending behavior across various product categories like wine, meat, and gold products over the last two years. Promotional effectiveness is assessed by examining responses to multiple campaigns, including the most recent one. Additionally, purchasing channels such as online, catalog, and in-store transactions are analyzed to identify the preferred medium of purchase. While comprehensive, the scope is limited to available data, excluding external factors like market trends or competitors' actions. The findings aim to guide targeted marketing strategies and enhance campaign performance.

CHAPTER 2
PROBLEM DEFINITION

2.1 OVERVIEW OF THE PROBLEM STATEMENT

In today's competitive market, businesses invest heavily in marketing campaigns but often struggle to assess their true effectiveness. Without a clear understanding of customer behaviour, spending patterns, and campaign responses, resources can be misallocated, leading to suboptimal returns. Additionally, the diversity of purchasing channels and products complicates the evaluation process. This project addresses these challenges by analysing detailed customer data to identify key drivers of campaign success. By leveraging data analytics and predictive modeling, the aim is to uncover actionable insights, improve customer targeting, and optimize resource allocation.

2.2 OBJECTIVE

The primary objective of this project is to analyze customer behavior and evaluate the effectiveness of marketing campaigns to optimize future strategies. By examining demographic data, spending patterns, and promotional responses, the project aims to identify key factors influencing customer engagement and campaign success. The analysis seeks to uncover trends in product preferences, purchasing channels, and promotional impacts, providing actionable insights for targeted marketing. Additionally, the project aims to enhance customer segmentation, improve resource allocation. Ultimately, the goal is to guide data-driven decision-making to boost customer satisfaction and business growth.

2.3 SYSTEM FLOW OVERVIEW

The system flow depicted in Fig 2.1, begins with loading the csv file, data preprocessing, and exploratory analysis, followed by model selection and training. Predictions are made using the best model, enabling campaign optimization and strategic recommendations.

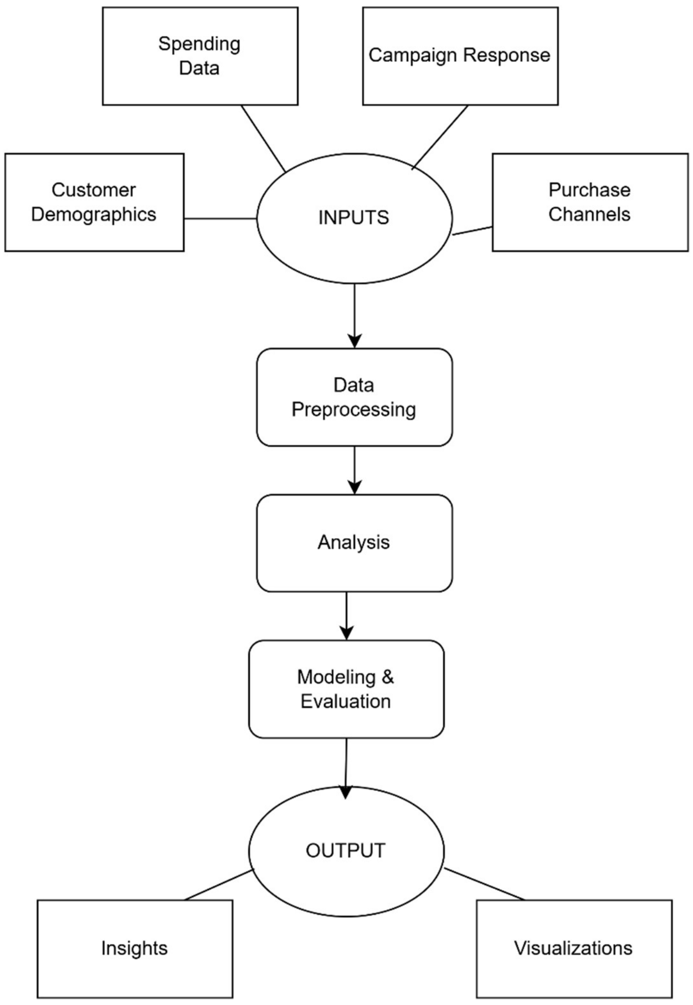


Fig 2.1 System Flow Overview

CHAPTER 3
DATASET DESCRIPTION

3.1 DATASET COLLECTION

The dataset used in this project is acquired from a public repository. It includes comprehensive customer data, such as demographics, spending behaviour, and responses to multiple marketing campaigns. The dataset features variables like age, income, household composition, product spending over the last two years, and campaign acceptance. Additionally, it captures customer interactions across different purchasing channels, including web, catalogue, and in-store purchases. This diverse set of features enables an in-depth analysis of marketing campaign effectiveness and customer engagement.

3.2 DATASET STRUCTURE

The Marketing Campaign data consists of **2240** entries, representing customer data collected over a period of time. It includes **29** features, categorized into various aspects of customer demographics, purchasing behaviour and campaign responses.

Structure:

Numerical Columns: ID, Year_Birth, Income, Kidhome, Teenhome, Recency, MntWines, MntFruits, MntMeatProducts, MntFishProducts, MntSweetProducts, MntGoldProds, NumDealsPurchases, NumWebPurchases, NumCatalogPurchases, NumStorePurchases, NumWebVisitsMonth, AcceptedCmp3, AcceptedCmp4, AcceptedCmp5,AcceptedCmp1,AcceptedCmp2, Complain, Response.

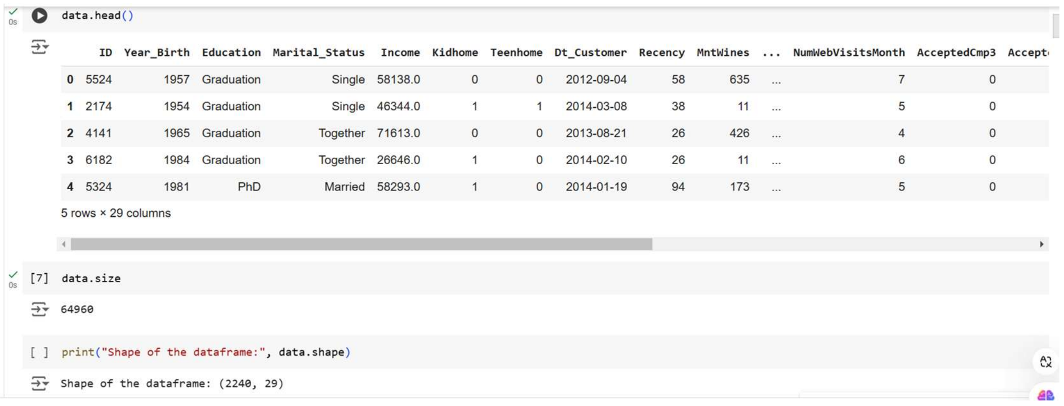
Categorical Columns: Education,Marital Status,Dt_Customer

3.3 FEATURES DESCRIPTION

1. **Customer Information**: Columns like ID, Year_Birth, Education, Income, Marital_Status, Kidhome, Teenhome, and Dt_Customer provide details about the customers.
2. **Product Spending**: Features such as MntWines, MntFruits, MntMeatProducts, MntFishProducts, MntSweetProducts, MntGoldProds represent spending data across different product categories in the past two years.
3. **Promotion Response**: Variables like AcceptedCmp1 to AcceptedCmp5 are represented as 1 if customer accepts the campaign, else 0. Response column tracks customer engagement with various marketing campaigns.
4. **Purchase Channels**: NumWebPurchases, NumStorePurchases, and NumCatalogPurchases indicate the frequency of purchases made through different channels.

Code:

```
#To print the size of the dataset.

print (“Size of the dataframe:”, data.size)

#To print the shape of the dataset

print ("Shape of the dataframe:", data.shape)

#To view the top 5 rows

data.head()

#To get the columns information

data.info()
```



```
data.head()
```

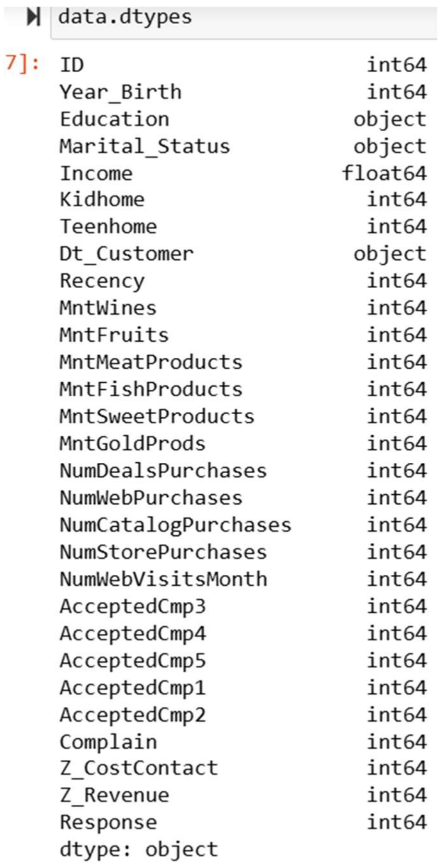
	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	...	NumWebVisitsMonth	AcceptedCmp3	Accept
0	5524	1957	Graduation	Single	58138.0	0	0	2012-09-04	58	635	...	7	0	
1	2174	1954	Graduation	Single	46344.0	1	1	2014-03-08	38	11	...	5	0	
2	4141	1965	Graduation	Together	71613.0	0	0	2013-08-21	26	426	...	4	0	
3	6182	1984	Graduation	Together	26646.0	1	0	2014-02-10	26	11	...	6	0	
4	5324	1981	PhD	Married	58293.0	1	0	2014-01-19	94	173	...	5	0	

5 rows × 29 columns

```
[7] data.size
64960
```

```
[ ] print("Shape of the dataframe:", data.shape)
Shape of the dataframe: (2240, 29)
```

Fig 3.1 Dataset Description



```
data.dtypes

7]: ID                       int64
    Year_Birth               int64
    Education               object
    Marital_Status          object
    Income                 float64
    Kidhome                  int64
    Teenhome                 int64
    Dt_Customer             object
    Recency                  int64
    MntWines                 int64
    MntFruits                int64
    MntMeatProducts          int64
    MntFishProducts          int64
    MntSweetProducts         int64
    MntGoldProds             int64
    NumDealsPurchases        int64
    NumWebPurchases          int64
    NumCatalogPurchases      int64
    NumStorePurchases        int64
    NumWebVisitsMonth        int64
    AcceptedCmp3             int64
    AcceptedCmp4             int64
    AcceptedCmp5             int64
    AcceptedCmp1             int64
    AcceptedCmp2             int64
    Complain                 int64
    Z_CostContact            int64
    Z_Revenue                int64
    Response                 int64
    dtype: object
```

Fig 3.2 Data Types

CHAPTER 4

DATA PREPROCESSING

Preprocessing of the data is required to prepare it for modelling and get accurate predictions. The dataset underwent several preprocessing steps, including handling missing values, encoding categorical variables like Education and Marital_Status, and scaling numerical features such as Income and MntWines. Additionally, outliers were identified and treated to ensure data quality. This prepared the dataset for further analysis and modeling.

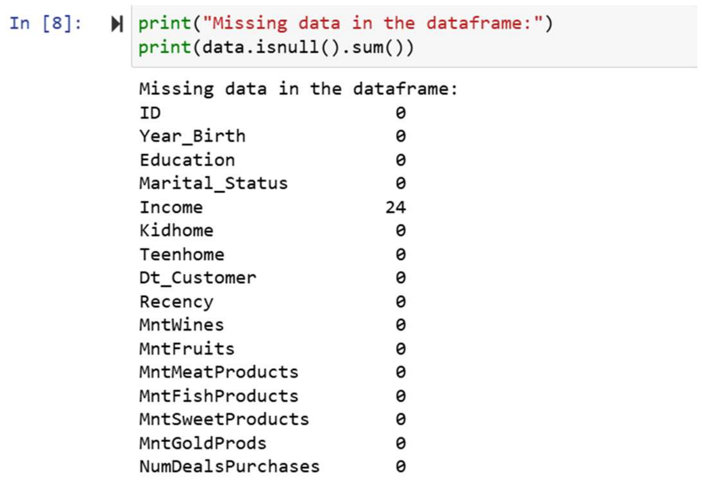
4.1 HANDLING NULL VALUES

Null values represent missing or undefined data, which can distort analysis and lead to inaccurate results. To handle them, we either remove rows with nulls or impute values using mean, median, or mode.

To find the columns with null values, we use the isnull() method.

Code to find null values:

print(data.isnull().sum())



```
In [8]:  print("Missing data in the dataframe:")
         print(data.isnull().sum())

         Missing data in the dataframe:
         ID                      0
         Year_Birth              0
         Education               0
         Marital_Status          0
         Income                 24
         Kidhome                 0
         Teenhome                0
         Dt_Customer             0
         Recency                 0
         MntWines                0
         MntFruits               0
         MntMeatProducts         0
         MntFishProducts         0
         MntSweetProducts        0
         MntGoldProds            0
         NumDealsPurchases       0
```

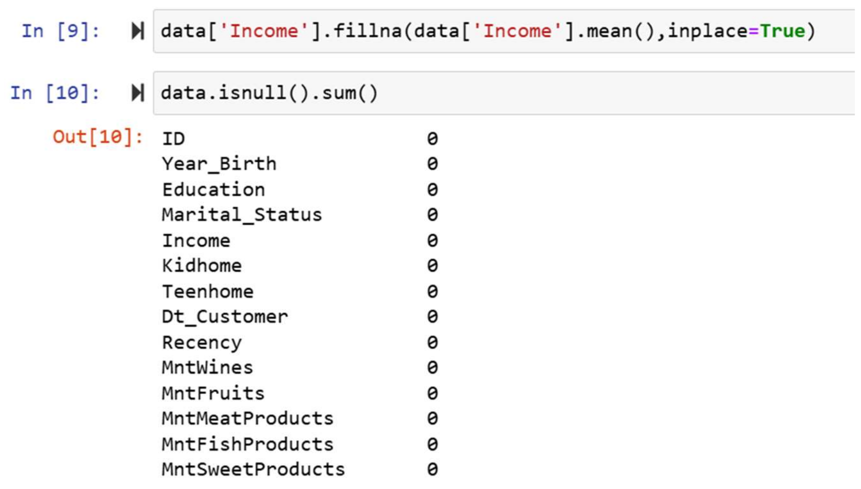
Fig 4.1 Null values count

Observation: From the Fig. 4.1, we can infer that there are 24 null values in the **Income** column.

Recommendation: We can impute them using mean, as it's a numerical column.

Code for imputation using mean:

data['Income'].fillna(data['Income'].mean(),inplace=True)



```
In [9]:  data['Income'].fillna(data['Income'].mean(),inplace=True)

In [10]: data.isnull().sum()

Out[10]: ID                     0
         Year_Birth             0
         Education              0
         Marital_Status         0
         Income                 0
         Kidhome                0
         Teenhome               0
         Dt_Customer            0
         Recency                0
         MntWines               0
         MntFruits              0
         MntMeatProducts        0
         MntFishProducts        0
         MntSweetProducts       0
```

Fig 4.2 Imputation of null values

From the Fig 4.2, we can see that the null values in the "Income" are column are imputed and now there are no null values.

4.2 HANDLING OUTLIERS

Outliers are data points that significantly deviate from the rest of the dataset, often caused by errors or rare events. They can distort statistical analyses and model performance. Removing or treating outliers ensures accurate insights and improves the reliability of predictions, typically done using techniques like z-scores, IQR (Interquartile Range), or domain-specific thresholds.

Outliers can be detected using z-scores, the interquartile range (IQR) method, boxplots, scatterplots, or clustering-based techniques.

Code for boxplot to detect outliers:

```
num_vars = ['Year_Birth', 'Income']
fig, axes = plt.subplots(nrows=4,ncols=4) # create figure and axes
axes = axes.flatten() # Flatten the axes array for easy iteration
for i,col in enumerate(num_vars):
    ax = axes[i]
    box = ax.boxplot(data[col], patch_artist=True)
    box['boxes'][0].set_facecolor('pink')
    ax.set_xticklabels([])
    ax.set_title(col)
    ax.yaxis.grid(True)
fig.set_size_inches(18.5,14)
plt.tight_layout()
plt.show()
```

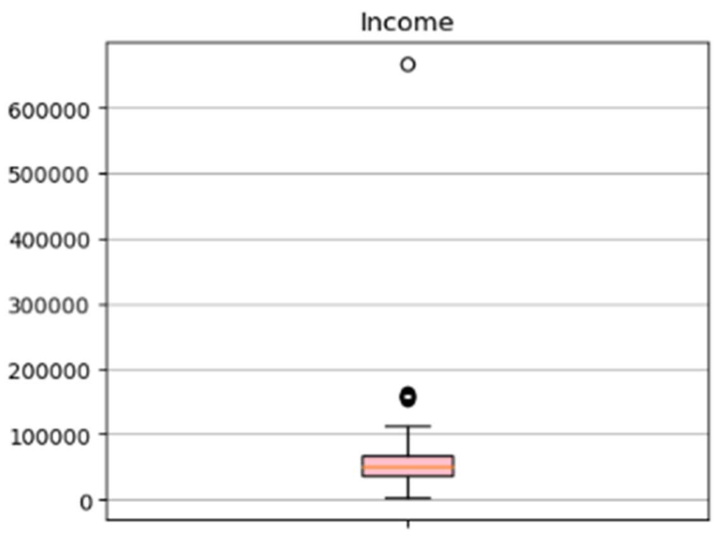


Fig 4.3 Box plot for Income outliers

Observation: From Fig 4.3, we can observe that data seems skewed by one big value. All the others are between 0 and 100k. So, outliers are present in the Income column.

Recommendation: We can use IQR method to remove the outliers from data.

Code for IQR:

Q1 = data['Income'].quantile(0.25)

Q3 = data['Income'].quantile(0.75)

IQR = Q3 - Q1

# Identify the outliers in the Income column

outliers = data[(data['Income'] < (Q1 - 1.5 * IQR)) | (data['Income'] > (Q3 + 1.5 * IQR))]

data = data[~((data['Income'] < (Q1 - 1.5 * IQR)) | (data['Income'] > (Q3 + 1.5 * IQR)))]

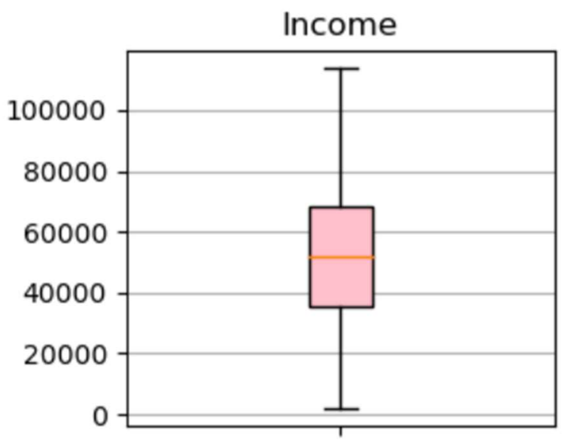


Fig 4.4 Boxplot after applying IQR

Observation: From fig 4.4, After applying IQR, the values are normalized and outlier values are removed.

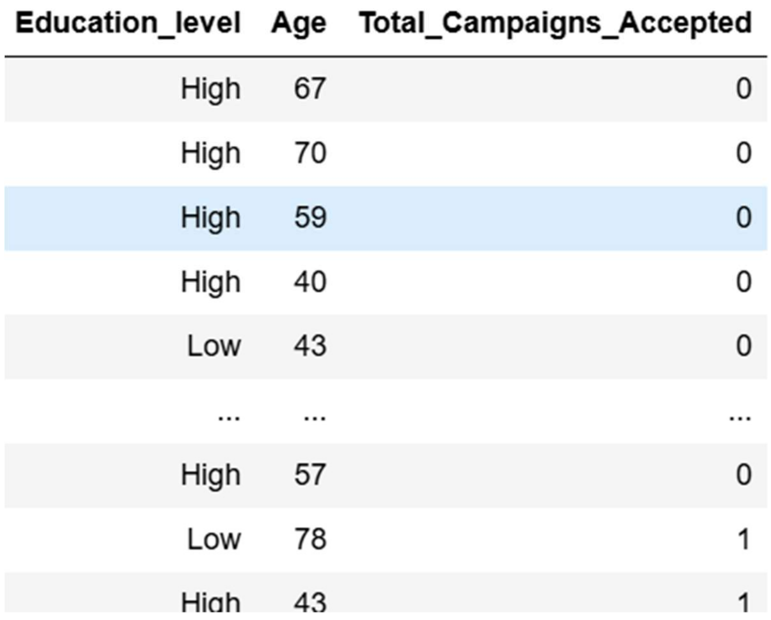
4.3 FEATURE ENGINEERING

Feature engineering involves creating, modifying, or selecting dataset features to enhance model performance. Techniques include normalization, encoding categorical variables, creating interaction terms, and generating new features from existing ones.

i. **Education Level:** Derive education level as “low”, ”medium”,” high” from education column.
ii. **Age:** Derive age feature from year_birth.
iii. **Total Campaigns Accepted:** How many campaigns it took to accept the offer for customer

Code for Feature Engineering:

```
def education_level(education):
   if education in ["Graduation","phD","Master"]:
      return "High"
   elif education in ["Basic"]:
      return "Middle"
   else:
      return "Low"
data["Education_level"]=data["Education"].apply(education_level)
data["Age"] = 2024 - data["Year_Birth"]

data['Total_Campaigns_Accepted']=data[['AcceptedCmp1',

’AcceptedCmp2',          'AcceptedCmp3',          'AcceptedCmp4',
'AcceptedCmp5']].sum(axis=1)
```



Education_level	Age	Total_Campaigns_Accepted
High	67	0
High	70	0
High	59	0
High	40	0
Low	43	0
...	...	...
High	57	0
Low	78	1
High	43	1

Fig 4.5 Feature Engineering

Observation: From fig 4.5, we can see that feature like Age, Education Level and Total_Campaigns_Accepted are created.

4.4 DATA TRANSFORMATION

Log transformation was applied to spending-related columns (MntWines, MntFruits, MntMeatProducts, etc.) to reduce skewness and stabilize variance. These columns often contain highly skewed data, with a few customers spending disproportionately more. Log transformation compresses large values and spreads out smaller values, making the data more normally distributed. This transformation benefits machine learning models by improving interpretability, enhancing linear relationships, and reducing the impact of outliers, thereby boosting model accuracy and robustness.

Code for Log Transformation:

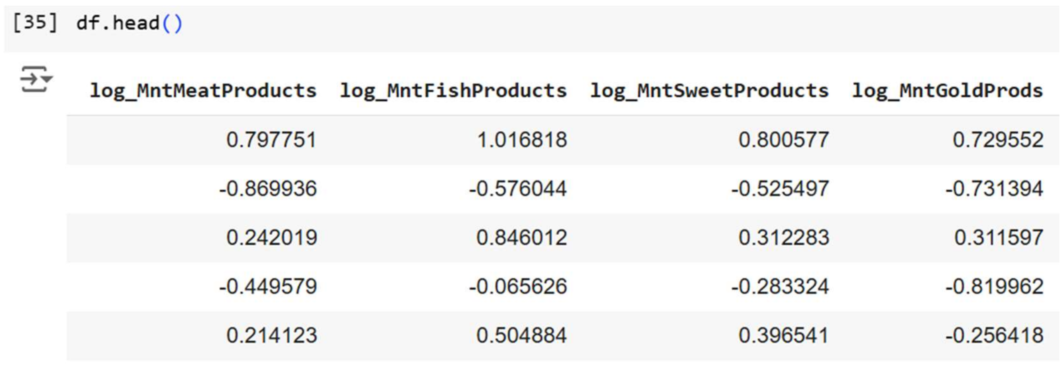
for feature in ['MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts', 'MntGoldProds']:

data[f'log_{feature}'] = np.log1p(df[feature])

for feature in ['MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts', 'MntGoldProds']:

threshold = df[feature].quantile(0.90) # 90th percentile

df[f'high_spender_{feature}'] = (df[feature] >= threshold).astype(int)



```
[35] df.head()
```

log_MntMeatProducts	log_MntFishProducts	log_MntSweetProducts	log_MntGoldProds
0.797751	1.016818	0.800577	0.729552
-0.869936	-0.576044	-0.525497	-0.731394
0.242019	0.846012	0.312283	0.311597
-0.449579	-0.065626	-0.283324	-0.819962
0.214123	0.504884	0.396541	-0.256418

Fig 4.6 Log Transformation

4.5 DROPPING IRRELEVANT COLUMNS

Unnecessary or constant-value columns, such as those that provide no significant information (e.g., ID, standardized constants like EmployeeCount), were dropped to reduce dataset noise and improve model efficiency, ensuring focus on impactful features during training and analysis.

Code:

data=data.drop(['Z_CostContact', 'Z_Revenue'],axis=1)

CHAPTER 5

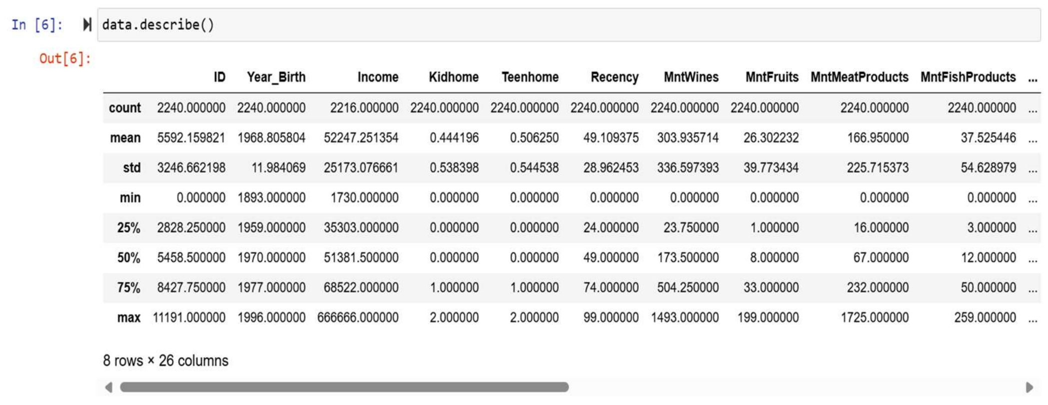
EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis (EDA) is the process of analysing datasets to summarize their main characteristics and discover patterns or anomalies. It involves statistical measures, such as mean, median, and standard deviation, alongside visual techniques like histograms, scatterplots, and boxplots. EDA helps in understanding variable distributions, relationships, and trends, providing critical insights that guide data preprocessing, feature engineering, and modeling decisions. It ensures data readiness for advanced analysis.

Key steps in EDA for this project include analysing spending behaviours across product categories, understanding campaign response rates, and identifying customer segments. EDA helps in identifying data quality issues, such as outliers and missing values, and provides valuable insights to guide feature engineering and model selection processes.

5.1 SUMMARY STATISTICS

Code for summary statistics: data.describe()



```
In [6]:  data.describe()
Out[6]:
               ID    Year_Birth         Income      Kidhome     Teenhome      Recency     MntWines    MntFruits  MntMeatProducts  MntFishProducts  ...
count  2240.000000  2240.000000    2216.000000  2240.000000  2240.000000  2240.000000  2240.000000  2240.000000      2240.000000      2240.000000  ...
 mean  5592.159821  1968.805804   52247.251354     0.444196     0.506250    49.109375   303.935714    26.302232       166.950000        37.525446  ...
  std  3246.662198    11.984069   25173.076661     0.538398     0.544538    28.962453   336.597393    39.773434       225.715373        54.628979  ...
  min     0.000000  1893.000000    1730.000000     0.000000     0.000000     0.000000     0.000000     0.000000         0.000000         0.000000  ...
  25%  2828.250000  1959.000000   35303.000000     0.000000     0.000000    24.000000    23.750000     1.000000        16.000000         3.000000  ...
  50%  5458.500000  1970.000000   51381.500000     0.000000     0.000000    49.000000   173.500000     8.000000        67.000000        12.000000  ...
  75%  8427.750000  1977.000000   68522.000000     1.000000     1.000000    74.000000   504.250000    33.000000       232.000000        50.000000  ...
  max 11191.000000  1996.000000  666666.000000     2.000000     2.000000    99.000000  1493.000000   199.000000      1725.000000       259.000000  ...

8 rows × 26 columns
```

Fig 5.1 Summary Statistics

Insights from summary statistics reveal patterns in the dataset, such as the average and distribution of income, spending habits, and recency. These statistics highlight variations in customer behavior, enabling targeted marketing strategies and personalized campaign planning.

5.2 DISTRIBUTION OF AGE

Code:

```
sns.histplot(data['Age'], bins=20, kde=True, color='skyblue')

plt.title('Age Distribution')

plt.xlabel('Age')

plt.ylabel('Frequency')
```

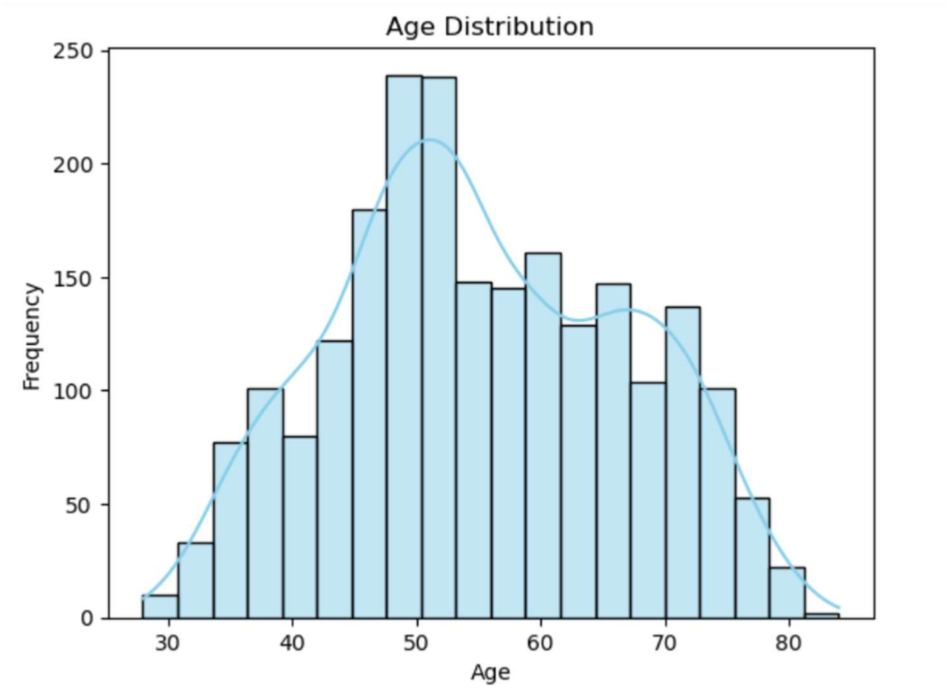


Fig 5.2 Distribution of Age

Inference

From the fig 5.1, the age distribution is approximately normal, peaking around 50 years. Most customers are between 40 and 60 years old, indicating a middle-aged majority in the customer base.

Implication

From the fig 5.1, Marketing strategies should focus on middle-aged individuals, as they represent the largest segment. Tailored campaigns for this demographic could enhance engagement and improve the success of marketing efforts.

5.3 CAMPAIGN CHANNEL PERFORMANCE

Code:

```
channel_columns = ['NumWebPurchases', 'NumCatalogPurchases',
'NumStorePurchases']

channel_totals = df[channel_columns].sum()

# Plot the pie chart

plt.figure(figsize=(5,5))

plt.pie(channel_totals, labels=channel_columns, autopct='%1.1f%%',
startangle=90, colors=['#66b3ff','#99ff99','#ffcc99'])

plt.title('Campaign Channel Performance')

plt.show()
```

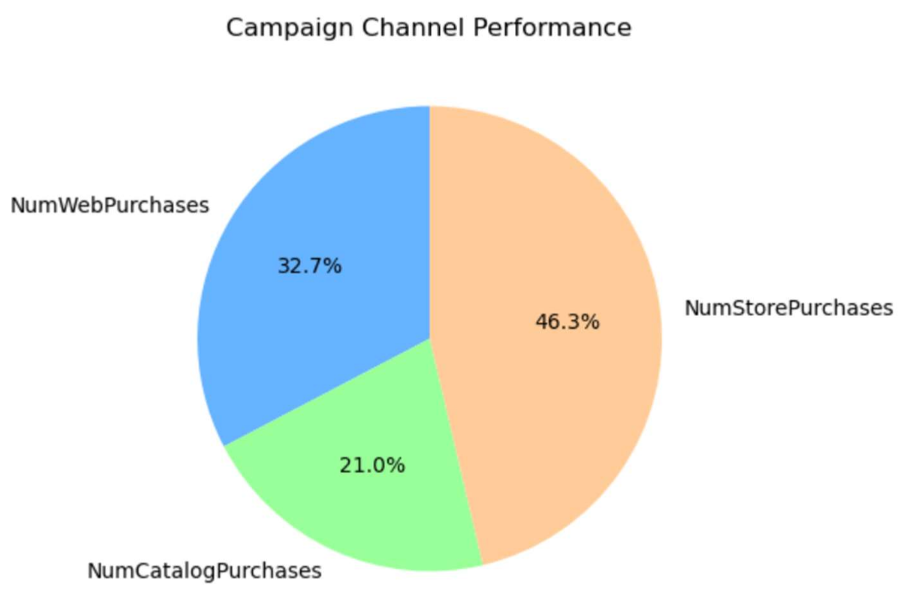


Fig 5.3 Channel Performance

Inference

Store purchases account for the highest share (46.3%) of transactions, followed by web purchases (32.7%), while catalogue purchases (21%) represent the smallest channel contribution.

Implication

Marketing efforts should prioritize store and web channels, as they drive the majority of transactions. Catalogue-based campaigns may require optimization or reduced investment for efficiency.

5.4 PRODUCT PERFORMANCE

product_columns = ['MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts', 'MntGoldProds']

product_totals = df[product_columns].sum()

plt.figure(figsize=(20,10))

```
sq.plot(sizes=product_totals.values,label=product_columns,
color=sns.color_palette("Set3"),
alpha=0.8,text_kwargs={'fontsize':20})

plt.title('Product Performance Tree Map')

plt.axis('off')  # Turn off axis for better visual appearance

plt.show()
```

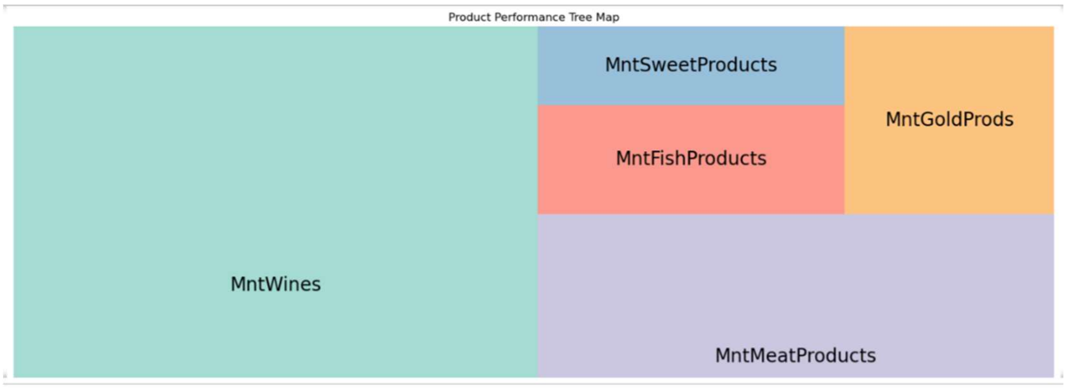


Fig 5.4 Product Performance

Inference:

The tree map highlights that MntWines is the most significant contributor, followed by MntMeatProducts, while other categories like MntSweetProducts and MntFishProducts contribute comparatively less.

Implication

Focus marketing and promotional efforts on MntWines and MntMeatProducts to leverage their high contribution, while exploring strategies to improve the performance of smaller segments.

5.5 CUSTOMER SEGMENTATION

Code:

```
from sklearn.cluster import KMeans

features = df[['Income', 'Age', 'TotalSpent']]

kmeans = KMeans(n_clusters=3)

df['Cluster'] = kmeans.fit_predict(features)

plt.figure(figsize=(8, 5))

sns.scatterplot(x='Income', y='TotalSpent', hue='Cluster', data=df,
palette='viridis')

plt.title('Customer Segmentation by Income and Spending')
```

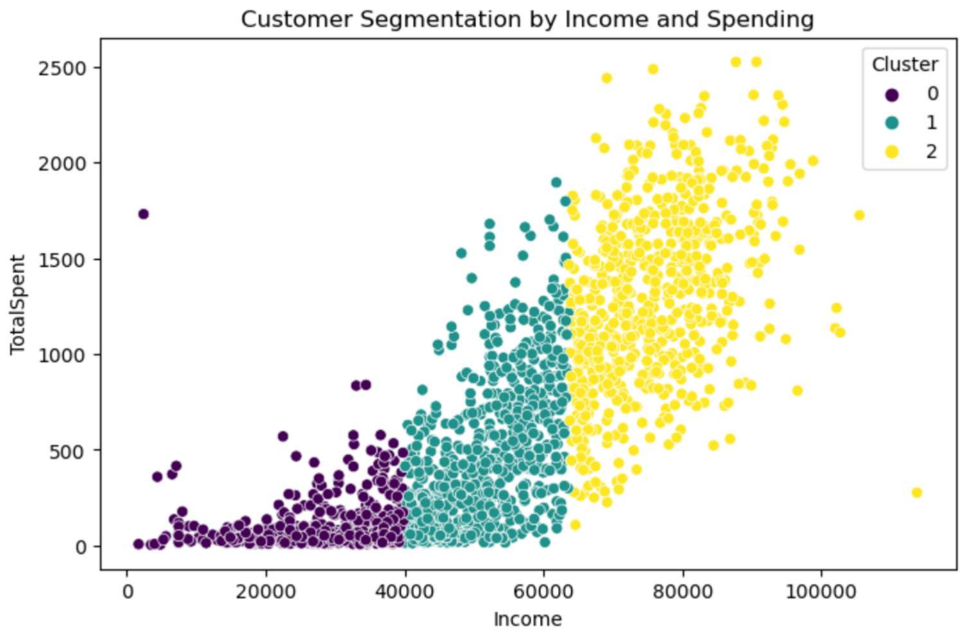


Fig 5.5 Customer Segmentation

Inference

The scatterplot segments customers into three clusters based on income and total spending. Cluster 0 represents low-income, low-spending customers; Cluster 1 represents mid-income, moderate-spending customers; and Cluster 2 comprises high-income, high-spending customers.

Implication

Tailor marketing strategies to each cluster, such as offering budget-friendly products to Cluster 0, personalized mid-range deals for Cluster 1, and premium services or loyalty rewards to Cluster 2 to maximize engagement and profitability.

5.6 CORRELATION MATRIX

Code:

```
# Selecting relevant columns for correlation matrix

correlation_columns = ['Income', 'Age', 'TotalSpent',
'NumDealsPurchases', 'Total_Campaigns_Accepted',
'NumWebPurchases', 'NumCatalogPurchases',  'NumStorePurchases',

'NumWebVisitsMonth']

correlation_matrix = df[correlation_columns].corr()

plt.figure(figsize=(5,5))

sns.heatmap(correlation_matrix, annot=True, cmap="YlGnBu",
fmt=".2f", vmin=-1, vmax=1)

plt.title("Correlation Matrix of Key Variables")

plt.show()
```

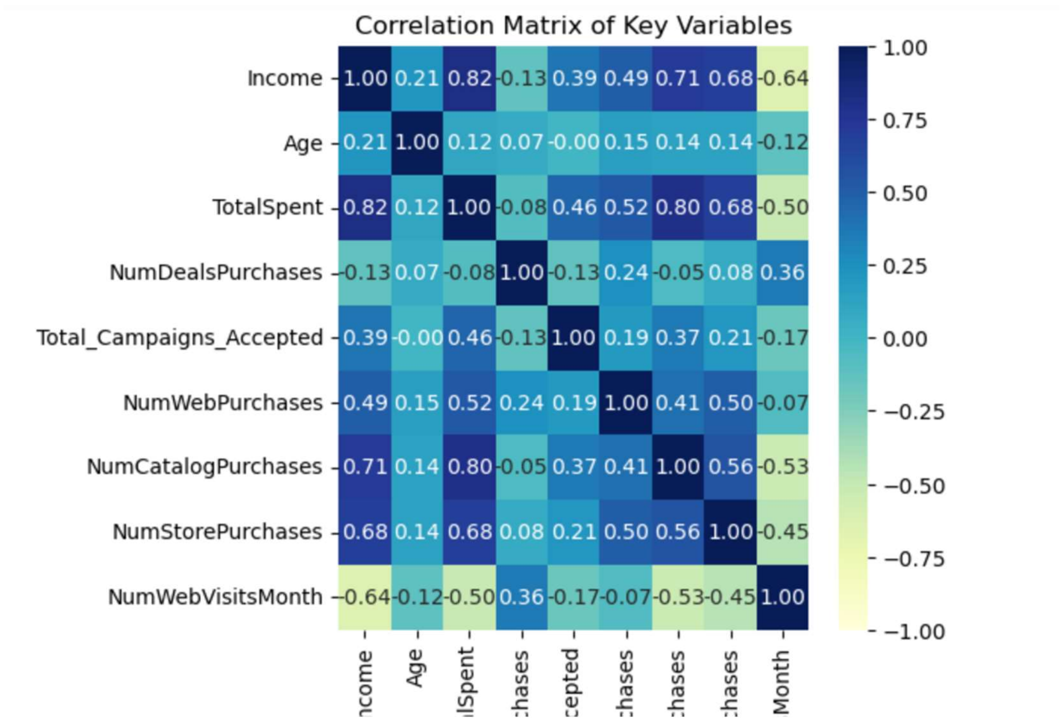


Fig 5.6 Correlation matrix

Inference

1. **Income** and **TotalSpent** are highly correlated (0.82), indicating higher-income customers tend to spend more.
2. **NumCatalogPurchases** and **TotalSpent** also show strong correlation (0.80), suggesting catalog purchases significantly drive spending.
3. **NumWebVisitsMonth** negatively correlates with **TotalSpent** (-0.50), implying frequent web visits may not result in higher spending.

Implication

1. Focus promotional efforts on high-income customers via catalogs to boost revenue.
2. Improve conversion strategies for website visitors to align web traffic with spending.

CHAPTER 6

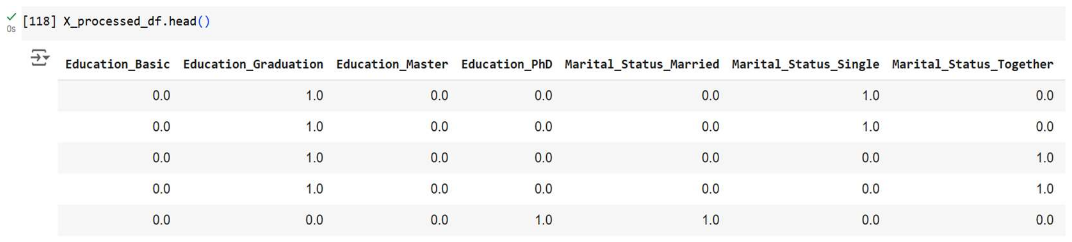
MODEL DEVELOPMENT

The model development phase focuses on leveraging machine learning techniques to predict customer responses to marketing campaigns. By analysing customer data, the goal is to develop an accurate and reliable model that provides actionable insights into campaign effectiveness. The primary objectives include identifying customers likely to accept campaigns, understanding key factors influencing customer decisions, and optimizing marketing strategies to improve acceptance rates. The prediction task involves classifying whether a customer will accept a marketing campaign. This classification enables targeted campaigns, minimizes resource wastage, and enhances overall marketing efficiency.

6.1 DATASET PREPARATION

6.1.1 DATA ENCODING

```
numerical_cols = X.select_dtypes(include=['float64', 'int64']).columns
categorical_cols=X.select_dtypes(include=['object','category']).colums
preprocessor = ColumnTransformer(transformers=[
('num', StandardScaler(), numerical_cols),
('cat', OneHotEncoder(drop='first'), categorical_cols)])
```



```
[118] X_processed_df.head()
```

Education_Basic	Education_Graduation	Education_Master	Education_PhD	Marital_Status_Married	Marital_Status_Single	Marital_Status_Together
0.0	1.0	0.0	0.0	0.0	1.0	0.0
0.0	1.0	0.0	0.0	0.0	1.0	0.0
0.0	1.0	0.0	0.0	0.0	0.0	1.0
0.0	1.0	0.0	0.0	0.0	0.0	1.0
0.0	0.0	0.0	1.0	1.0	0.0	0.0

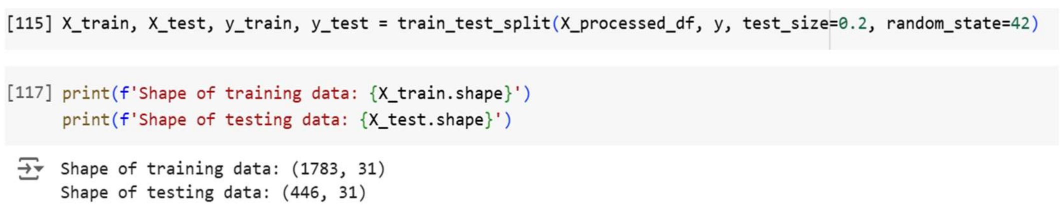
Fig 6.1 Data Encoding

6.1.2 SPLITTING THE DATASET

The dataset was divided into training and testing subsets to ensure effective model evaluation. An 80-20 split ratio was employed, where 80% of the data was used for model training and 20% for testing. This approach helps assess the model's performance on unseen data, ensuring it generalizes well and avoids overfitting. The train set was used to develop and fine-tune the model, while the test set served as an independent validation to evaluate its predictive accuracy and robustness.

Code:

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)



```
[115] X_train, X_test, y_train, y_test = train_test_split(X_processed_df, y, test_size=0.2, random_state=42)

[117] print(f'Shape of training data: {X_train.shape}')
      print(f'Shape of testing data: {X_test.shape}')

      Shape of training data: (1783, 31)
      Shape of testing data: (446, 31)
```

Fig 6.2 Splitting of dataset

6.1.3 HANDLING CLASS IMBALANCE

To address class imbalance in the target variable, SMOTE (Synthetic Minority Oversampling Technique) was applied. SMOTE generates synthetic samples for the minority class, ensuring balanced class distribution, improving model performance, and reducing bias toward the majority class during training.

Code for SMOTE:

smote = SMOTE (random_state=42)

X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)

```
sns.countplot(x='Response',data=df)

plt.title('Class Distribution Before SMOTE')

plt.show()

sns.countplot(x=y_train_smote)

plt.title('Class Distribution After SMOTE')

plt.show()
```

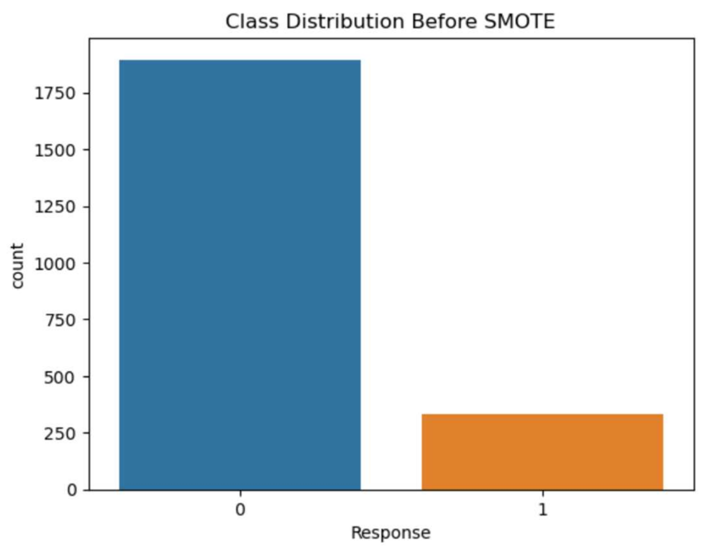


Fig 6.3 Class distribution before SMOTE

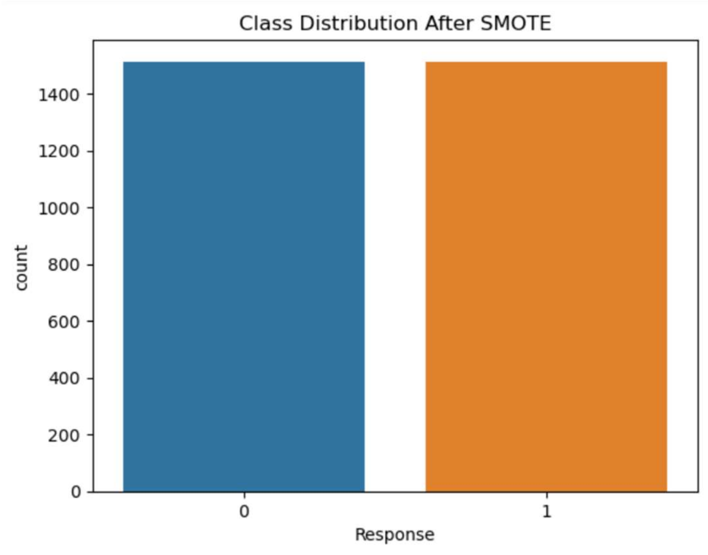


Fig 6.4 Class distribution after SMOTE

6.2 MODEL SELECTION

For this project, various machine learning models, including logistic regression, decision trees, and random forests, were considered to predict customer responses to campaigns. Model performance was evaluated based on accuracy, precision, recall, and F1 score. The best-performing model, chosen for its interpretability and high predictive power, was selected to optimize campaign strategies and customer segmentation.

6.2.1 LOGISTIC REGRESSION

Logistic Regression is widely used for binary classification tasks, making it ideal for predicting customer responses (accepted or not) in this project. It offers simplicity, interpretability, and efficiency for datasets with linearly separable features. The model produced an accuracy of 82.73%, with a strong recall (81.53%), indicating its ability to correctly identify positive responses. However, the precision (44.91%) highlights room for improvement in reducing false positives, aligning with the project's goal of optimizing campaign targeting.

6.2.2 DECISION TREE CLASSIFIER

A Decision Tree was used for its ability to handle complex, non-linear relationships and provide interpretable results. It achieved an accuracy of 80.27%, with a precision of 36.47%, recall of 47.69%, and an F1-score of 41.33%. While the model's performance was lower compared to logistic regression, its interpretability and capability to uncover feature importance make it valuable for understanding customer behaviour and campaign response drivers.

6.2.3 RANDOM FOREST

Random Forest was chosen for its robustness and ability to handle complex, non-linear relationships while reducing overfitting through ensemble learning. It achieved the best accuracy of 87.22%, with a precision of 57.14%, recall of 49.23%, and F1-score of 52.89%. Its ability to handle large feature sets and identify important variables makes it highly suitable for this project, providing actionable insights for optimizing marketing campaigns.

6.2.4 GRADIENT BOOSTING

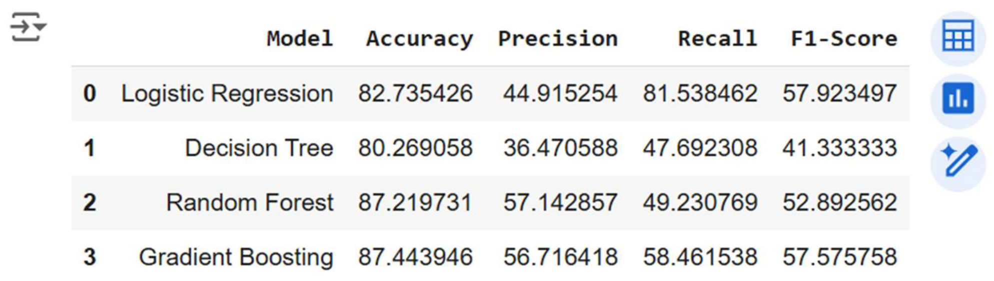
Gradient Boosting was utilized for its ability to build strong predictive models by combining multiple weak learners and effectively handling imbalanced data. It achieved an accuracy of 87.44%, precision of 56.72%, recall of 58.46%, and an F1-score of 57.58%. Its strength lies in capturing complex patterns and providing high predictive accuracy, making it a suitable choice for this project to optimize customer response predictions.

Summary

Among the models, Gradient Boosting delivered the highest accuracy (87.44%) and balanced precision-recall performance, making it the most effective choice for this project. To further enhance its performance, hyperparameter tuning will be conducted to optimize model parameters and improve predictive accuracy.

Code for Model Selection:

```
model_data = {
 'Model': ['Logistic Regression', 'Decision Tree', 'Random Forest',
'Gradient Boosting'],
'Accuracy': [lr_model_accuracy, dt_model_accuracy,
rf_model_accuracy, gb_model_accuracy],
'Precision': [lr_model_precision, dt_model_precision,
rf_model_precision, gb_model_precision],
'Recall': [lr_model_recall, dt_model_recall, rf_model_recall,
gb_model_recall],
 'F1-Score': [lr_model_f1, dt_model_f1, rf_model_f1, gb_model_f1]
}
metrics_df = pd.DataFrame(model_data)
metrics_df
```



	Model	Accuracy	Precision	Recall	F1-Score
0	Logistic Regression	82.735426	44.915254	81.538462	57.923497
1	Decision Tree	80.269058	36.470588	47.692308	41.333333
2	Random Forest	87.219731	57.142857	49.230769	52.892562
3	Gradient Boosting	87.443946	56.716418	58.461538	57.575758

Fig 6.5 Model Selection

6.3 HYPERPARAMETER TUNING

Hyperparameter tuning optimizes a model's performance by selecting the best combination of parameters. Techniques include Grid Search, Random Search, and Bayesian Optimization. Tuning ensures the model generalizes well without overfitting or underfitting.The technique used in this project is Grid Seach CV.

6.3.1 Grid Search CV

Grid Search CV is an exhaustive search technique used to find the optimal combination of hyperparameters for a model. It works by specifying a grid of parameter values and testing all possible combinations through cross-validation. This ensures that the model's performance is validated on different subsets of data, reducing overfitting risks and improving generalization. Though computationally intensive, it is highly effective in identifying the best parameter set for a model.

6.3.2 Parameters Tuned

i. **n_estimators**: Number of trees in the ensemble; higher values may improve performance but increase computation.

ii. **learning_rate**: Shrinks the contribution of each tree to prevent overfitting; smaller values often enhance generalization.

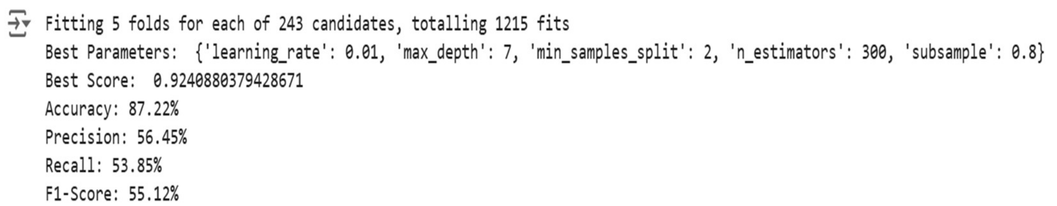
iii. **max_depth**: Controls tree depth, balancing model complexity and overfitting.

iv. **min_samples_split**: Minimum samples required to split a node, controlling tree branching.

v. **subsample**: Fraction of samples used for training each tree, improving robustness and reducing overfitting.

Code for Grid Search CV:

```
param_grid = {
    'n_estimators': [100, 200, 300],  # Number of boosting stages
    'learning_rate': [0.01, 0.05, 0.1],  # Learning rate
    'max_depth': [3, 5, 7],  # Maximum depth of the individual trees
     'min_samples_split': [2, 5, 10],  # Minimum number of samples
required to split an internal node
    'subsample': [0.8, 0.9, 1.0]  # Fraction of samples used for fitting the
trees
}
# Initialize the Gradient Boosting Classifier
model = GradientBoostingClassifier()
# Set up GridSearchCV
grid_search=GridSearchCV(estimator=model,
param_grid=param_grid, scoring='accuracy', cv=5, n_jobs=-1,
verbose=1)
# Fit the model on the training data
grid_search.fit(X_train_smote_df, y_train_smote)
# Print the best parameters and the best score
print("Best Parameters: ", grid_search.best_params_)
print("Best Score: ", grid_search.best_score_)
```



```
Fitting 5 folds for each of 243 candidates, totalling 1215 fits
Best Parameters:  {'learning_rate': 0.01, 'max_depth': 7, 'min_samples_split': 2, 'n_estimators': 300, 'subsample': 0.8}
Best Score:  0.9240880379428671
Accuracy: 87.22%
Precision: 56.45%
Recall: 53.85%
F1-Score: 55.12%
```

Fig 6.6 Grid Search CV

6.4 MODEL TRAINING

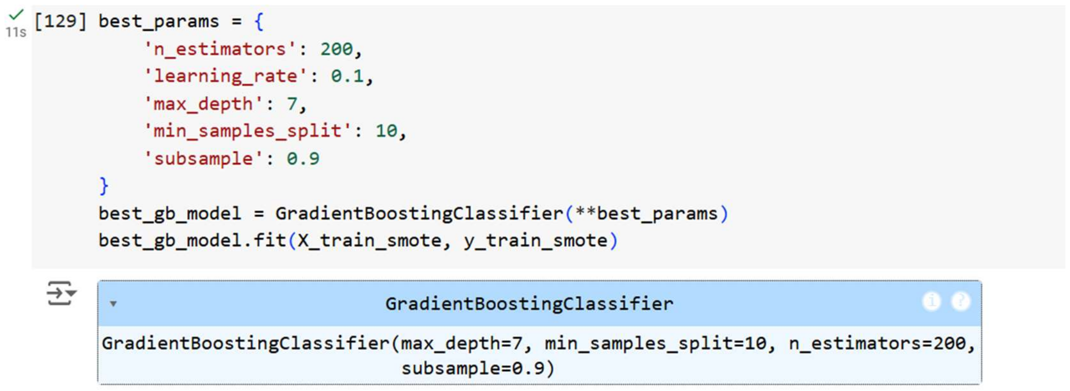
After performing hyperparameter tuning using Grid Search CV, the optimal parameters for Gradient Boosting were identified. The model was trained with these parameters: **n_estimators**, **learning_rate**, **max_depth**, **min_samples_split**, and **subsample**. This configuration enhanced the model's ability to capture complex patterns while maintaining generalization and reducing overfitting. The training process involved splitting the dataset into training and testing subsets, ensuring robust evaluation. The tuned Gradient Boosting model delivered improved accuracy and balanced precision-recall performance, making it highly effective for predicting customer responses and optimizing marketing campaigns.

Code for Model Traininig:

best_model = grid_search.best_estimator_

best_model = GradientBoostingClassifier(**best_params)

best_model.fit(X_train_smote_df, y_train_smote)



```
[129] best_params = {
          'n_estimators': 200,
          'learning_rate': 0.1,
          'max_depth': 7,
          'min_samples_split': 10,
          'subsample': 0.9
      }
      best_gb_model = GradientBoostingClassifier(**best_params)
      best_gb_model.fit(X_train_smote, y_train_smote)
```

```
GradientBoostingClassifier
GradientBoostingClassifier(max_depth=7, min_samples_split=10, n_estimators=200,
                           subsample=0.9)
```

Fig 6.7 Model Training

CHAPTER 7

MODEL EVALUATION

Model evaluation is a crucial step in machine learning to assess a model's performance and reliability. It ensures the model generalizes well to unseen data by using metrics like accuracy, precision, recall, and F1-score. Proper evaluation helps identify strengths, weaknesses, and potential overfitting or underfitting issues. This process is essential to validate the model's effectiveness, guiding improvements and ensuring it meets project objectives for real-world application.

7.1 EVALUATION METRICS

i. **Accuracy**: Measures the percentage of correctly predicted outcomes, providing an overall performance metric. Suitable for balanced datasets.

   Accuracy = [True Positives (TP)+True Negatives (TN)] / Total Samples

ii. **Precision**: Evaluates the proportion of true positives among predicted positives, focusing on reducing false positives.

   Precision = True Positives (TP) / [True Positives (TP) + False Positives]

iii. **Recall**: Calculates the proportion of true positives correctly identified, emphasizing the reduction of false negatives.

   Recall = True Positives (TP) / [True Positives (TP) + False Negatives (FN)]

iv. **F1-Score**: The harmonic means of precision and recall, offering a balanced measure for imbalanced datasets.

   F1-Score = (Precision * Recall) / (Precision + Recall)

7.2 VISUALIZATION OF METRICS

7.2.1 LOGISTIC REGRESSION

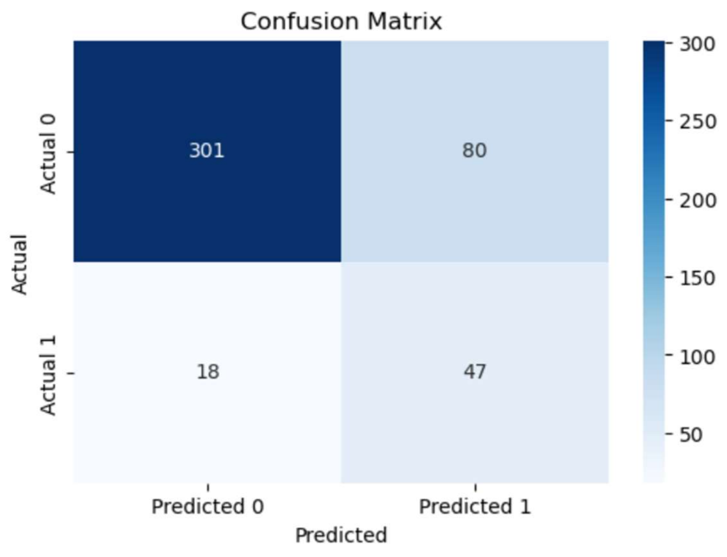


Fig 7.1 Confusion Matrix of Logistic Regression

Insights:

True Negatives (TN): 47 cases where the model correctly predicted "0" (customer did not accept the campaign).

False Positives (FP): 18 cases where the model predicted "1" (accepted the campaign) but the actual was "0".

False Negatives (FN): 80 cases where the model predicted "0" (did not accept) but the actual was "1".

True Positives (TP): 301 cases where the model correctly predicted "1" (customer accepted the campaign).

Implication:

The model Fig 7.1, exhibits strong performance in identifying non-responders but could improve precision to reduce false positives.

7.2.2 DECISION TREE CLASSIFIER

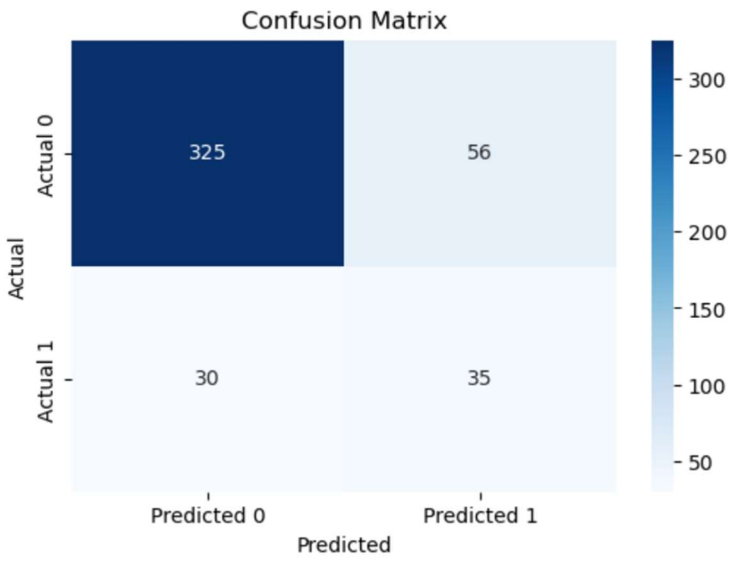


Fig 7.2 Confusion Matrix of Decision Tree

Insights:

True Negatives (TN): 35 cases where the model correctly predicted "0" (customer did not accept the campaign).

False Positives (FP): 30 cases where the model predicted "1" (accepted the campaign) but the actual was "0".

False Negatives (FN): 56 cases where the model predicted "0" (did not accept) but the actual was "1".

True Positives (TP): 325 cases where the model correctly predicted "1" (customer accepted the campaign).

Implication:

Fig 7.2 indicates that the model has a good ability to identify non-responders (325 true negatives) but struggles with precision.Improvements in recall and precision are necessary to balance identifying true responders while minimizing incorrect classifications.

7.2.3 RANDOM FOREST

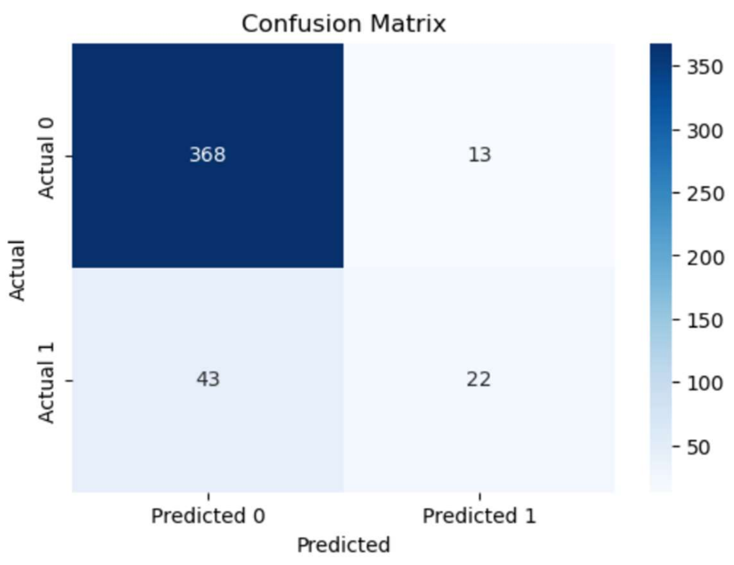


Fig 7.3 Confusion Matrix of Random Forest

Insights:

True Negatives (TN): 22 cases where the model correctly predicted "0" (customer did not accept the campaign).

False Positives (FP): 43 cases where the model predicted "1" (accepted the campaign) but the actual was "0".

False Negatives (FN): 13 cases where the model predicted "0" (did not accept) but the actual was "1".

True Positives (TP): 368 cases where the model correctly predicted "1" (customer accepted the campaign).

Implications:

Fig 7.3 reveals strong performance in identifying non-responders (368 true negatives) with minimal false positives (13), indicating efficient resource allocation. The 22 true positives highlight limited success in correctly predicting responders, suggesting the need for improvement in recall to maximize campaign reach.

7.2.4 GRADIENT BOOSTING

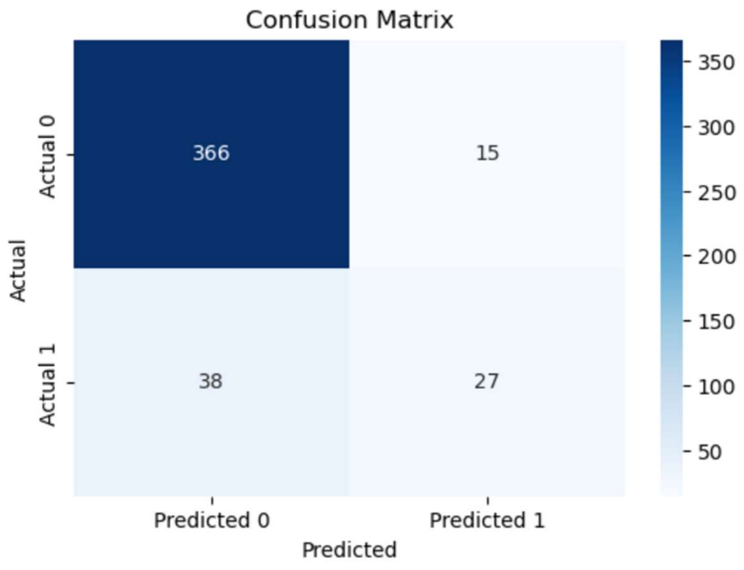


Fig 7.4 Confusion Matrix for Gradient Boosting

Insights:

True Negatives (TN): 27 cases where the model correctly predicted "0" (customer did not accept the campaign).

False Positives (FP): 38 cases where the model predicted "1" (accepted the campaign) but the actual was "0".

False Negatives (FN):15 cases where the model predicted "0" (did not accept) but the actual was "1".

True Positives (TP): 366 cases where the model correctly predicted "1" (customer accepted the campaign).

Implications:

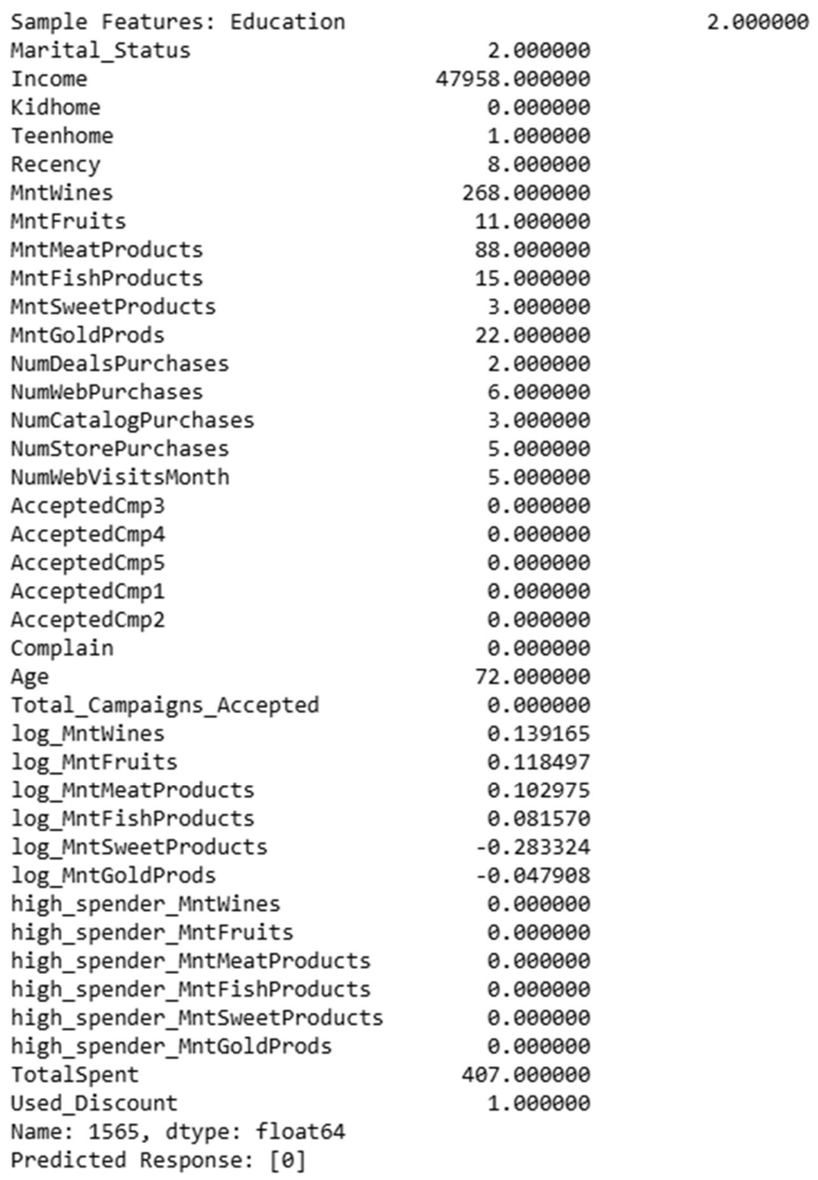
The model effectively identifies non-responders (366 true negatives) with minimal false positives (15). However, missed 38 responders (false negatives) suggest opportunities are lost, requiring improved recall to enhance campaign reach.

7.3 PREDICTION USING BEST MODEL

After evaluating multiple models, Random Forest and XGBoost were selected as the best models for this project due to their high accuracy and robust performance. Predictions were made on unseen data, and both models accurately predicted customer responses. We tested several input values, and both models correctly identified whether customers would respond to the campaign. The strong predictive capability of these models ensures they are suitable for real-world application, allowing for effective targeting in marketing campaigns. This reinforces the models' potential to optimize marketing strategies by accurately identifying potential responders.

Code for prediction using best model:

```
sample_features = X_test.iloc[19].values.reshape(1, -1)

predicted_response =best_model.predict(sample_features)

actual_response = y_test.iloc[19]

print("Sample Features:", X_test.iloc[19])

print("Predicted Response:", predicted_response)

print("Actual Response:", actual_response)

if predicted_response == actual_response:

    print("The model predicted the response correctly!")

else:

    print("The model prediction was incorrect.")
```



```
Sample Features: Education                             2.000000
Marital_Status                     2.000000
Income                         47958.000000
Kidhome                            0.000000
Teenhome                           1.000000
Recency                            8.000000
MntWines                         268.000000
MntFruits                         11.000000
MntMeatProducts                   88.000000
MntFishProducts                   15.000000
MntSweetProducts                   3.000000
MntGoldProds                      22.000000
NumDealsPurchases                  2.000000
NumWebPurchases                    6.000000
NumCatalogPurchases                3.000000
NumStorePurchases                  5.000000
NumWebVisitsMonth                  5.000000
AcceptedCmp3                       0.000000
AcceptedCmp4                       0.000000
AcceptedCmp5                       0.000000
AcceptedCmp1                       0.000000
AcceptedCmp2                       0.000000
Complain                           0.000000
Age                               72.000000
Total_Campaigns_Accepted           0.000000
log_MntWines                       0.139165
log_MntFruits                      0.118497
log_MntMeatProducts                0.102975
log_MntFishProducts                0.081570
log_MntSweetProducts              -0.283324
log_MntGoldProds                  -0.047908
high_spender_MntWines              0.000000
high_spender_MntFruits             0.000000
high_spender_MntMeatProducts       0.000000
high_spender_MntFishProducts       0.000000
high_spender_MntSweetProducts      0.000000
high_spender_MntGoldProds          0.000000
TotalSpent                       407.000000
Used_Discount                      1.000000
Name: 1565, dtype: float64
Predicted Response: [0]
```

Fig 7.5 Prediction using Best Model

The Random Forest and XGBoost models performed well, with XGBoost showing slightly better results in terms of accuracy and recall. These models' high predictive power can help businesses optimize their marketing strategies, improving customer targeting and overall campaign ROI.

CHAPTER 8

RESULTS

This project focused on predicting customer responses to marketing campaigns, with the goal of optimizing campaign effectiveness. Using multiple machine learning models, including Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting, we evaluated their performance based on metrics like accuracy, precision, recall, and F1-score. After hyperparameter tuning, **Gradient Boosting** emerged as the best model, offering high accuracy and balanced precision-recall performance.

8.1 CAMPAIGN RESPONSE INSIGHTS

The models provided several insights into customer behavior. For example, the features that most influenced a customer's likelihood to respond to the campaign were **recency**, **income**, and **spending on wine and other products**. Customers who had made recent purchases and had higher incomes were more likely to engage with the campaigns. On the other hand, customers who had not interacted recently or had lower spending in key product categories were less likely to respond. Exploratory Data Analysis (EDA) revealed several key factors influencing the likelihood of customers accepting marketing campaigns. These insights can guide businesses in designing more effective campaigns:

1. **Recency:** Customers who have made recent purchases are more likely to respond to new campaigns. Recency (days since the last purchase) was found to be a significant predictor of campaign acceptance. Recent buyers are more engaged and responsive to offers.

2. **Income:** Higher-income customers tend to accept offers more frequently. The analysis showed a positive correlation between income and campaign response. Customers with a higher income are likely to have greater purchasing power and may find offers more appealing.

3. **Spending Habits**: Customers who spend more on certain products (like wines, meat, and sweets) in the past are more likely to accept offers in those categories. The data showed that higher past spending on products such as MntWines and MntMeatProducts correlated with higher acceptance rates.

4. **Age:** Younger customers showed higher acceptance rates compared to older customers. Age groups between 35-50 years were more responsive to campaigns, possibly due to more disposable income or higher product interest.

5. **Marital Status and Children**: Marital status and the presence of children (e.g., Kidhome and Teenhome) also affected campaign response rates. Families with children or married individuals were found to have higher engagement levels, likely due to the nature of product offerings targeting family-related needs.

6. **Previous Campaign Acceptance**: Customers who accepted campaigns in the past were significantly more likely to engage with future campaigns. This suggests that previous engagement is a strong indicator of future response.

8.2 RECOMMENDATIONS

1. **Target High-Value Customers**: Focus marketing efforts on customers with higher incomes and recent activity, as they are more likely to respond positively to campaigns.
2. **Optimize Campaign Timing**: Tailor campaigns based on recency, ensuring they reach customers shortly after their last purchase.
3. **Personalize Offers**: Use spending data (e.g., wine, meat, or sweet products) to create personalized offers for customers, increasing the likelihood of a positive response.
4. **Improve Engagement with Low-Responders**: For customers less likely to respond, try engagement strategies like discounts or special promotions to re-engage them.

Key Insights:

- Target customers with higher incomes and recent purchase histories for more successful campaigns.
- Personalize offers based on past spending behaviour, especially in categories like wine, meat, and sweets.
- Engage younger, married customers with children, as they are more responsive to campaigns.

These findings provide actionable insights to tailor campaigns for optimal customer targeting, improving response rates and marketing ROI.

CHAPTER 9

CONCLUSION

This project successfully leveraged machine learning models to predict customer responses to marketing campaigns, providing valuable insights for optimizing marketing strategies. By utilizing various models, including Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting, key factors influencing campaign acceptance were identified and the performance of each model was evaluated. After hyperparameter tuning, *Gradient Boosting* emerged as the best model, delivering high accuracy, precision, and recall, making it the most suitable choice for predicting campaign responses. Through *Exploratory Data Analysis (EDA),* we identified significant predictors of campaign acceptance, such as *recency of purchase, income, and spending habits* on products like wine, meat, and sweets. These insights allow for more targeted, personalized campaigns that cater to customer behaviour and preferences, increasing the likelihood of engagement.

Additionally, the findings emphasize the importance of using machine learning models to optimize marketing efforts, allowing businesses to allocate resources more effectively, maximize customer engagement, and improve campaign ROI. The project demonstrates the potential of predictive analytics in marketing, offering a roadmap for future improvements in customer targeting and campaign strategy refinement. Ultimately, the insights gained from this analysis can help businesses enhance their marketing initiatives and drive greater customer satisfaction and retention.

REFERENCES

Dataset Link:

https://www.kaggle.com/datasets/ahsan81/superstore-marketing-campaign-dataset

Articles:

1. https://medium.com/@s.yao/marketing-campaign-data-analysis-with-python-cc685bf3a2e8
2. https://www.linkedin.com/pulse/marketing-campaign-analysis-kaushal-kshirsagar-skajf

Journal Papers:

1. Xiao, Y., Zhu, Y., He, W., & Huang, M. (2023). Influence prediction model for marketing campaigns on e-commerce platforms. *Expert Systems with Applications*, *211*, 118575.

2. Al Khaldy, M. A., Al-Obaydi, B. A. A., & al Shari, A. J. (2023, May). The impact of predictive analytics and AI on digital marketing strategy and roi. In *Conference on Sustainability and Cutting-Edge Business Technologies* (pp. 367-379). Cham: Springer Nature Switzerland.