PREDICTIVE CUSTOMER BEHAVIOUR MODELLING USING AI

MCA - IV SEMESTER, FINAL PROJECT REPORT

Submitted by

SOUMEN CHATTERJEE

[EC2432251010407]

Under the Guidance of

Dr. G. Babu

(Assistant Professor, Directorate of Online Education)

in partial fulfilment for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

DIRECTORATE OF ONLINE EDUCATION

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR- 603 203

JAN 2024 MCA Batch

NOV-DEC 2025 Exam (4th SEM)

DIRECTORATE OF ONLINE EDUCATION

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

This Final Project (SEM IV) Report titled **"Predictive Customer Behaviour Modelling using AI"** of **"Soumen Chatterjee [EC2432251010407]"**, who carried out the Semester IV Final Project Work under the supervision of Program Coordinator of Online Education along with the company mentor.

Certified further that to the best of my knowledge, the work reported herein does not form any other project report or dissertation based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

Signature of the Student

(Soumen Chatterjee)

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to **Dr. C. Muthamizhchelvan**, Vice Chancellor, SRM Institute of Science and Technology, for providing the necessary facilities and continued support that enabled the successful completion of this project.

I extend my heartfelt thanks to **Prof. Dr. Manoranjan Pon Ram**, Director, Directorate of Online Education, SRM Institute of Science and Technology, for his valuable guidance and encouragement throughout the course of this Final Project (SEM IV).

I am deeply thankful to **Dr. G. Babu**, Program Coordinator, Directorate of Online Education, for his insightful feedback during project reviews and his unwavering support. I am especially grateful to him for serving as my project guide. His mentorship, encouragement, and the freedom he offered to explore research topics aligned with my interests were instrumental in shaping this project. His dedication and passion for solving real-world problems have been truly inspiring.

I also extend my sincere appreciation to the faculty, staff, and fellow students of the **Directorate of Online Education**, SRM Institute of Science and Technology, for their cooperation and assistance during various stages of the project.

Lastly, I would like to acknowledge the unconditional love, encouragement, and support of my **parents, family members, and friends**, without whom this journey would not have been possible.

SOUMEN CHATTERJEE

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

Table of Contents

1. Abstract

Predictive Customer Behaviour Modelling using AI is a data-driven approach designed to understand consumer behaviour, preferences, and purchasing trends. The project focuses on analysing customer data to segment consumers into distinct groups, enabling businesses to optimize targeted strategies. By applying machine learning techniques such as clustering (unsupervised learning) and classification (supervised learning), the study extracts valuable insights from customer demographics and purchasing behavior.

The major accomplishments of this project are as follows:

- **Data preprocessing**, including data cleaning, handling missing values, and feature engineering.
- **Unsupervised learning** implementation using K-Means clustering to group customers into meaningful segments.
- **Supervised learning techniques'** application such as Decision Tree, K-Nearest Neighbours (KNN), and Random Forest to predict customer responses.
- **Visualization and analysis** of critical customer attributes, including income, spending behaviour, and family structure.

My key contributions involved data preparation, model selection, and interpretation of results. The insights generated support targeted marketing efforts and enhance customer relationship management strategies.

2. Introduction

- **Background**

Businesses increasingly rely on data-driven decision-making to understand their customers better. Predictive Customer Behaviour Modelling using AIplays a crucial role in personalizing marketing strategies, optimizing customer engagement, and improving product recommendations.

- **Problem Statement**

Companies often struggle with segmenting their customers effectively, leading to inefficient marketing campaigns. Instead of marketing a product to the entire customer base, a company can analyse which customer segment is most likely to buy the product and target them specifically.

- **Development Process**

The project follows a structured development approach:

a) **Data Collection & Preprocessing**: Cleaning data, handling missing values, and engineering useful features.

b) **Exploratory Data Analysis (EDA)**: Understanding data distribution and identifying patterns.

c) **Unsupervised Learning (Clustering)**: Using K-Means clustering to segment customers.

d) **Supervised Learning (Classification)**: Using decision trees, KNN, and random forests to predict customer responses.

e) **Model Evaluation & Insights**: Assessing model performance and deriving business insights.

3. System Analysis

System analysis represents a fundamental stage in the development of any software or data-driven application. This phase focuses on evaluating the current operational environment, identifying the shortcomings of existing methodologies, and formulating an enhanced solution through the integration of advanced technologies and innovative strategies. The analysis encompasses a review of the existing system (where applicable), the design of the proposed framework for customer personality analysis, and a feasibility assessment to ensure the practicality and sustainability of the project.

A. Existing System

Traditionally, organizations have depended on basic customer profiling techniques that utilize a limited set of demographic variables such as age, gender, and geographic location. These conventional methods are often manual or rule-based, offering insufficient depth and precision to capture the complexity of customer behaviour. The primary limitations of such systems include:

- **No Personalization**: Marketing initiatives tend to be generic rather than customized for distinct customer segments.
- **Inefficient Targeting**: In the absence of advanced insights, resources are frequently allocated to customers with low engagement potential.
- **Data Underutilization**: Although businesses gather substantial customer data, they often fail to leverage advanced analytical models for informed decision-making.
- **No Predictive Capability**: Traditional frameworks lack the ability to forecast customer responses to campaigns or predict purchase likelihood.

B. Proposed System

The proposed system, titled *Predictive Customer Behavior Modeling using Artificial Intelligence*, is developed to address the shortcomings of traditional approaches by employing advanced data science methodologies. This framework adopts a data-driven

strategy for customer segmentation and behavioural forecasting, enabling more accurate and actionable insights.

Key Features of the Proposed System:

- **Data-Driven Insights**: Utilizes historical customer information, including expenditure patterns, demographic attributes, and digital interaction data, to generate actionable insights.
- **Customer Segmentation**: Employs clustering algorithms such as *K-Means* to categorize customers into distinct groups based on behavioural and preference similarities.
- **Marketing Response Prediction**: Implements supervised machine learning models, including *Random Forest*, to estimate customer responsiveness to promotional campaigns.
- **Feature Engineering**: Constructs derived variables such as *Age*, *Total_Spent*, and *Family_Size* to enhance the analytical depth and predictive accuracy of the model.
- **Visualization**: Delivers graphical representations to facilitate intuitive interpretation of complex patterns and interrelationships within the data.
- **Model Evaluation**: Assesses predictive performance using metrics such as accuracy scores to ensure reliability and robustness of the results.

Benefits:

- **Facilitates Personalized Marketing:** Enables the design of marketing strategies tailored to individual customer profiles.
- **Enhances Customer Engagement:** Strengthens interaction and long-term relationships with customers through targeted initiatives.
- **Optimizes Marketing Expenditure:** Reduces unnecessary costs by focusing resources on high-potential customer segments.
- **Improves Conversion Rates:** Increases the likelihood of transforming prospects into active customers through data-driven targeting.

- **Supports Strategic Decision-Making:** Provides actionable insights that inform and refine organizational marketing strategies.

C. Feasibility Study

To ensure the practicality and success of the project, a feasibility study was conducted in the following areas:

Feasibility Type	Assessment
Technical Feasibility	☑ The tools and technologies used (Python, Jupyter Notebook, scikit-learn, etc.) are well-established and supported. The project was successfully implemented on standard hardware without requiring high-end computational resources.
Operational Feasibility	☑ The solution is user-friendly, modular, and adaptable. Business users and analysts can adopt this model for campaign optimization with minimal training. The outputs are easy to interpret via visualizations and performance metrics.
Economic Feasibility	☑ As the project uses open-source tools, the cost of development is minimal. Implementation can lead to higher ROI by optimizing marketing strategies and improving customer targeting.
Schedule Feasibility	☑ The project was completed within the academic timeline. A well-structured plan with clear milestones ensured timely completion of each phase—data cleaning, modeling, and evaluation.

Conclusion of System Analysis

The analysis clearly shows that the proposed machine learning-based system significantly improves upon the traditional approach to customer profiling. With

minimal investment and strong analytical capabilities, this system is feasible, scalable, and aligns well with modern business intelligence needs.

---

4. Analysis and Requirements

Problem Analysis

The primary challenge in this project is to segment customers based on their demographic and purchasing data. The dataset consists of 29 features, including customer age, income, education, marital status, and spending behaviour across various product categories.

UML Analysis Model

The analysis can be represented using the following UML models:

- **Use Case Diagram**: Represents interactions between the system and different user roles (e.g., Data Analyst, Business Manager).
- **Activity Diagram**: Shows the step-by-step process of data preprocessing, clustering, and classification.
- **Class Diagram**: Defines key entities such as Customer, Purchase History, and Segmentation Model.

System-Level and Software-Level Requirements

- **System Requirements**
    - Python environment (Jupyter Notebook, Anaconda)

  - Libraries: pandas, NumPy, seaborn, matplotlib, scikit-learn
  - Computational resources for machine learning processing

- **Software Requirements**
  - Data preprocessing module
  - Clustering module (K-Means)
  - Classification module (Decision Tree, KNN, Random Forest)
  - Visualization module for insights

---

5. Problem Description / Module Description

The project consists of the following key modules:

a) Data Preprocessing

- Importing libraries and dataset
- Handling missing values (e.g., replacing missing income values with mean)
- Feature engineering (creating new features like Age, Total_Spent, Family_Size)
- Encoding categorical variables

b) Exploratory Data Analysis (EDA)

- Understanding data distribution using histograms and scatter plots
- Visualizing customer spending behaviour
- Analysing correlations using heatmaps

c) Unsupervised Learning: Clustering

- Applying K-Means clustering to segment customers
- Finding the optimal number of clusters using the Elbow Method
- Visualizing clusters based on income and spending behaviour

d) Supervised Learning: Classification

- Training machine learning models to predict customer response
- Implementing Decision Tree, KNN, and Random Forest classifiers
- Comparing model performance and accuracy

e) Model Evaluation and Business Insights

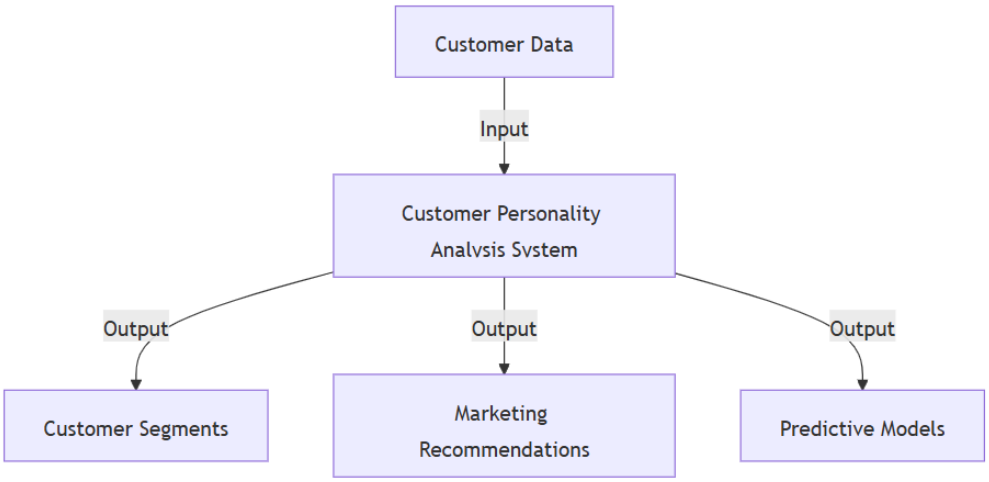
- Evaluating classification models using accuracy scores
- Understanding customer segments for targeted marketing strategies

6. Design

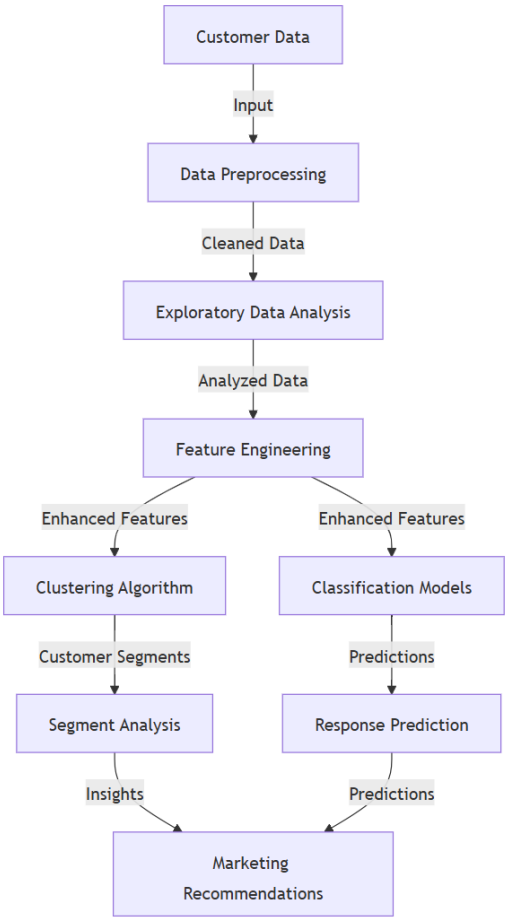
6. a) System Design

Data Flow Diagram (DFD):

Level 0 DFD - Predictive Customer Behavior Modeling using AI

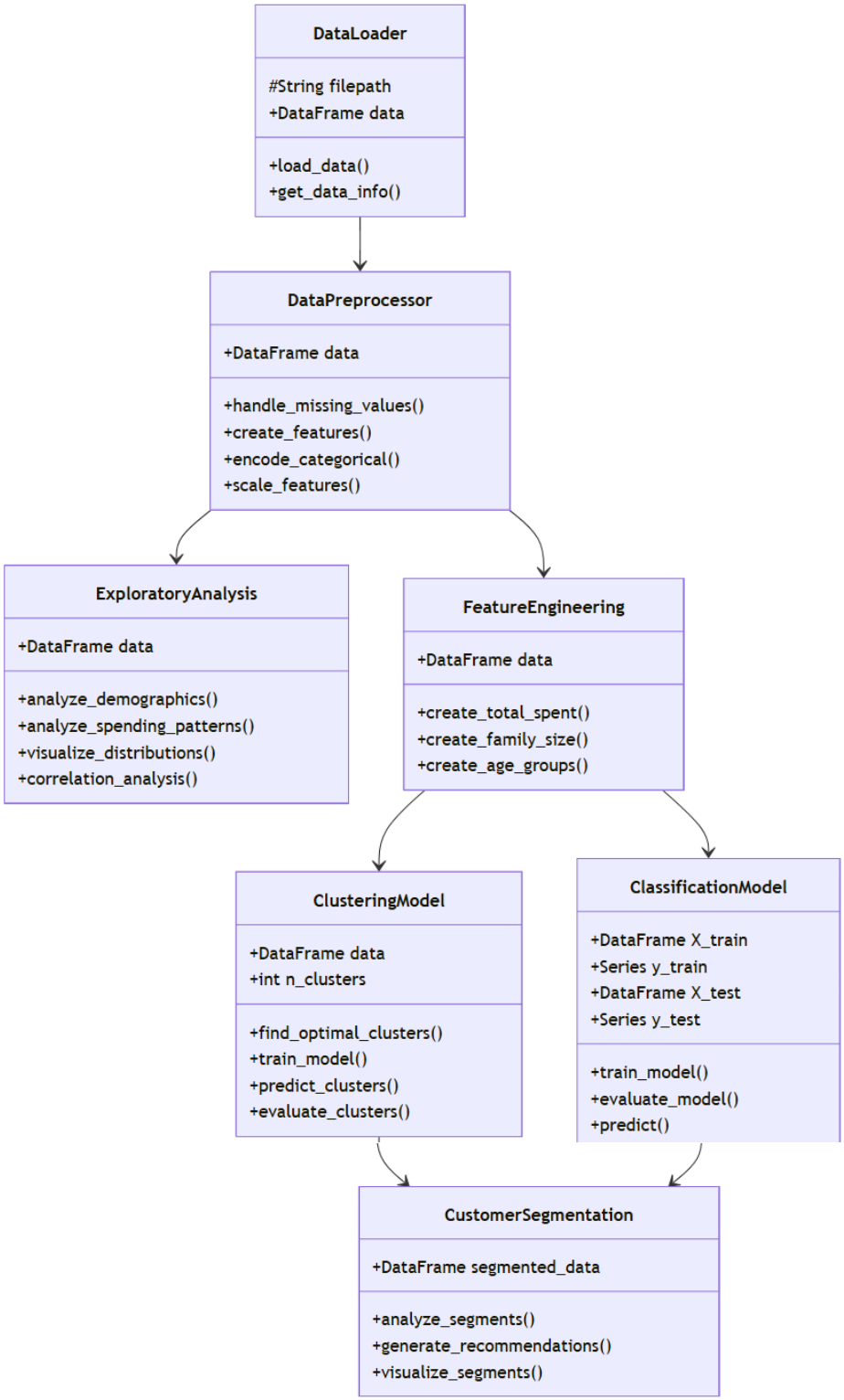


Level 1 DFD - Predictive Customer Behaviour Modelling using AISystem

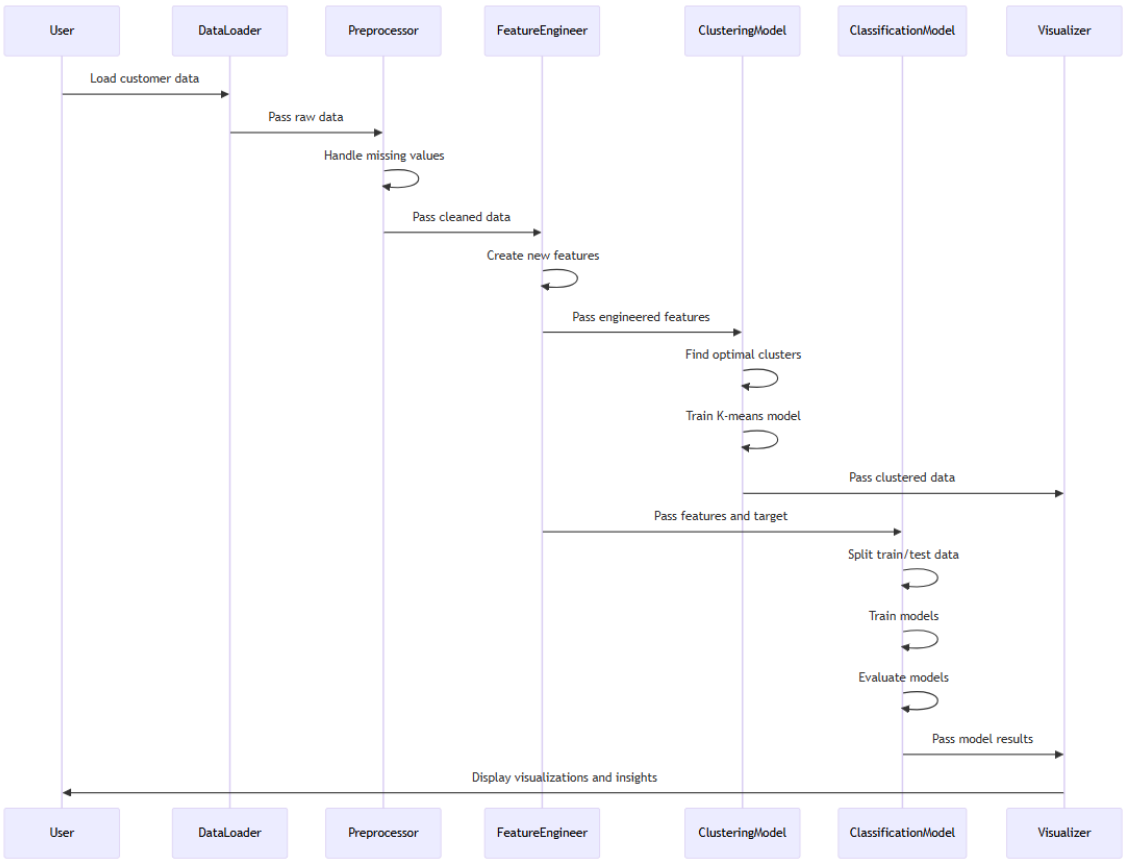


6. b) UML Diagrams

Class Diagram - Predictive Customer Behavior Modeling using AI



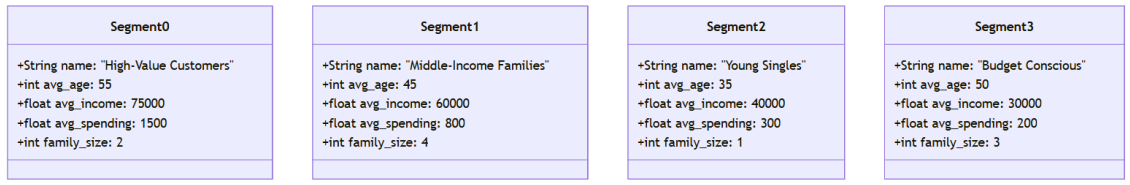
Sequence Diagram - Predictive Customer Behaviour Modelling using AI



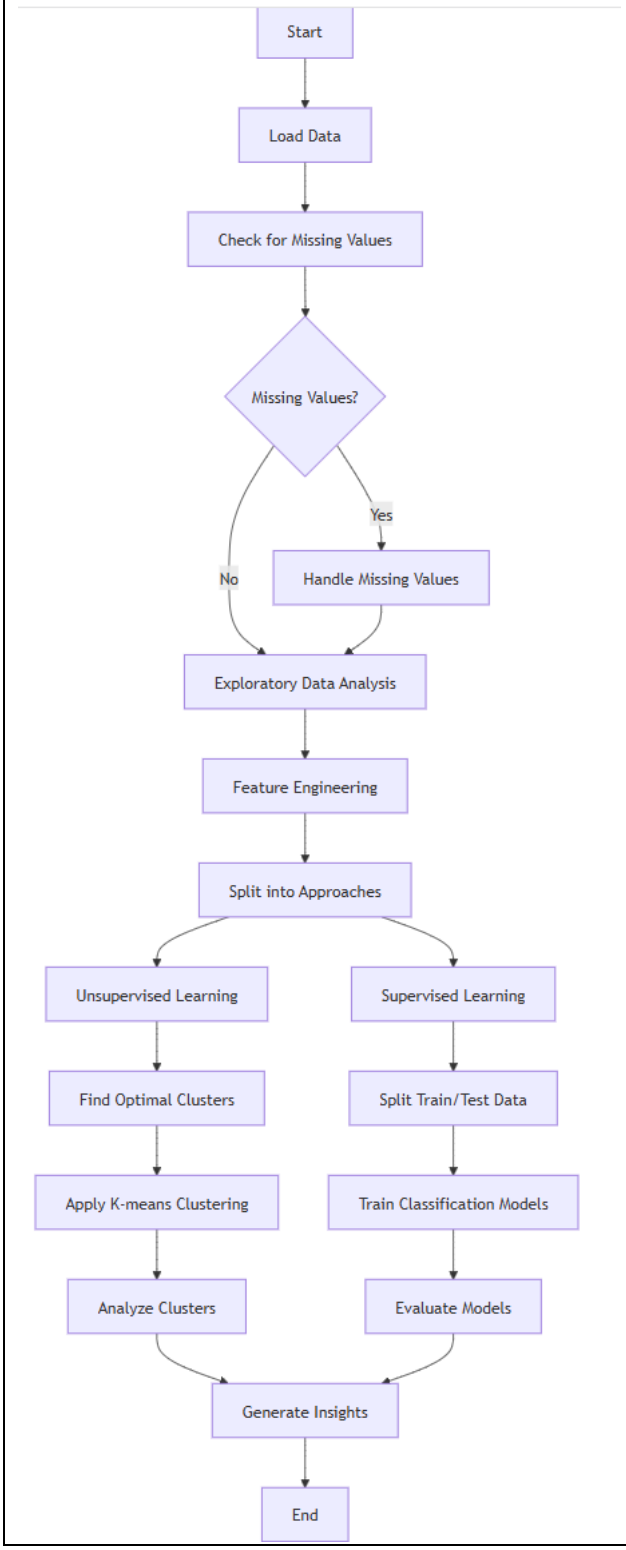
Use Case Diagram - Predictive Customer Behavior Modeling using AI

Object Diagram

Object Diagram - Customer Segments

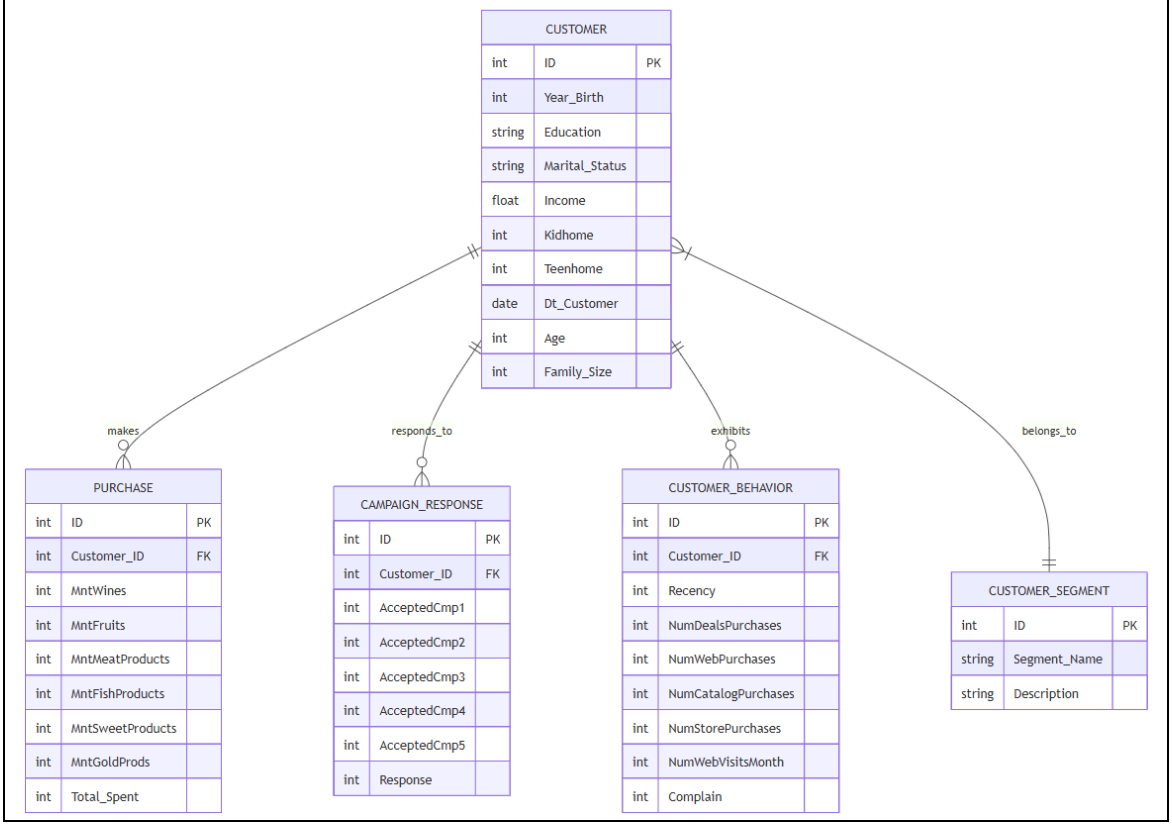


Control Flow Diagram - Predictive Customer Behavior Modeling using AI



6. c) Database Design

E-R Diagram - Predictive Customer Behavior Modeling using AI



Functional Dependencies and Normalization

Functional Dependencies:

1. ID → Year_Birth, Education, Marital_Status, Income, Kidhome, Teenhome, Dt_Customer, Age
2. ID → MntWines, MntFruits, MntMeatProducts, MntFishProducts, MntSweetProducts, MntGoldProds
3. ID → Recency, NumDealsPurchases, NumWebPurchases, NumCatalogPurchases, NumStorePurchases, NumWebVisitsMonth
4. ID → AcceptedCmp1, AcceptedCmp2, AcceptedCmp3, AcceptedCmp4, AcceptedCmp5, Response, Complain
5. ID → Total_Spent, Family_Size, Clusters

Normalization Process:

A. First Normal Form (1NF):

f) All attributes contain atomic values
g) No repeating groups
h) Primary key identified (ID)

B. Second Normal Form (2NF):

i) 1. Already in 1NF
j) 2. No partial dependencies (all attributes depend on the entire primary key)

C. Third Normal Form (3NF):

k) Already in 2NF
l) No transitive dependencies
m) Decomposed into:

- CUSTOMER (ID, Year_Birth, Education, Marital_Status, Income, Kidhome, Teenhome, Dt_Customer, Age, Family_Size)
- PURCHASE (ID, Customer_ID, MntWines, MntFruits, MntMeatProducts, MntFishProducts, MntSweetProducts, MntGoldProds, Total_Spent)
- CAMPAIGN_RESPONSE (ID, Customer_ID, AcceptedCmp1, AcceptedCmp2, AcceptedCmp3, AcceptedCmp4, AcceptedCmp5, Response)
- CUSTOMER_BEHAVIOR (ID, Customer_ID, Recency, NumDealsPurchases, NumWebPurchases, NumCatalogPurchases, NumStorePurchases, NumWebVisitsMonth, Complain)
- CUSTOMER_SEGMENT (ID, Customer_ID, Cluster_ID)

7. Implementation

Implementation Approach

The project was implemented in a structured, modular fashion using Python within the Jupyter Notebook environment. Its primary objective was to perform customer data analysis through both supervised and unsupervised machine learning approaches, ensuring a clear distinction between the stages of data preprocessing, model development, and performance evaluation.

The stages of implementation included:

- Data Ingestion
- Data Cleaning and Preprocessing
- Feature Engineering
- Data Visualization
- Modeling (Clustering and Classification)
- Evaluation and Interpretation of Results

Each phase utilizes well-structured, reusable code segments designed for clarity and adaptability, enabling easy modifications and experimentation throughout the process.

Software Reuse and Libraries Used

The project extensively leveraged software reuse by incorporating widely adopted open-source libraries, which helped minimize development time while enhancing overall reliability. The primary Python libraries utilized included:

Library	Purpose
pandas	Data manipulation and tabular data processing
NumPy	Numerical operations and statistical functions
matplotlib	Data visualization through plots and charts
seaborn	Enhanced statistical data visualization
scikit-learn	Machine Learning algorithms for clustering, classification, and preprocessing
yellowbrick	Visual analysis of ML model performance (e.g., Elbow method for clustering)

These libraries adhere to established industry standards and enjoy broad adoption, ensuring that the codebase remains scalable and easy to maintain.

Special Tools Used

Tool	Usage
Jupyter Notebook	Main IDE used for developing, testing, visualizing, and documenting the project
Yellowbrick	Used specifically for KElbowVisualizer, which helps to find the optimal number of clusters (k)
scikit-learn Models	Used for KMeans Clustering, Decision Tree, KNN, and Random Forest classifiers

Leveraging Jupyter Notebook offered an interactive environment that facilitated iterative code development and debugging, enabled visualization of intermediate outputs, and allowed detailed step-by-step documentation using Markdown.

Design Patterns and Coding Techniques

Although the project does not follow an object-oriented paradigm, several fundamental design principles and patterns were incorporated:

- **Modularity:** Each stage of the analysis—such as data loading, cleaning, visualization, and modeling—was implemented in distinct code blocks, adhering to the Separation of Concerns principle.
- **DRY (Don't Repeat Yourself):** Common operations like aggregations and visualizations were structured for reuse, minimizing redundancy.
- **Reusable Functions (Future Enhancement):** The project can be further improved by converting repetitive tasks (e.g., plotting or model evaluation) into callable functions or class methods.
- **Encapsulation of Data Transformations:** Preprocessing activities, including label encoding, handling missing values, and feature engineering, were encapsulated before passing data to machine learning models.

Data Transformation and Preprocessing Techniques

Specialized coding and preprocessing techniques applied in the project included:

- **Handling Missing Data:** Missing income values were imputed using the mean to retain records without discarding rows.
- **Feature Engineering:**
  - Age was computed from the *Year_Birth* attribute.
  - *Total_Spent* was derived by summing expenditures across all product categories.
  - *Family_Size* was calculated by combining *Kidhome*, *Teenhome*, and marital status information.
- **Categorical Encoding:** Education levels were transformed into numeric form using `LabelEncoder` from `sklearn.preprocessing`.

- **Feature Scaling (Advanced Option):** `StandardScaler` was employed to normalize feature ranges, which is advantageous for algorithms sensitive to data magnitude.

Model Implementation Summary

 Unsupervised Learning (K-Means Clustering):

- The optimal number of clusters (k = 4) was identified using the Elbow Method, implemented via Yellowbrick's KElbowVisualizer.
- K-Means was then applied to segment customers based on demographic and behavioral attributes.

 Supervised Learning (Classification Models):

- Three classifiers were employed: Decision Tree, K-Nearest Neighbors (KNN), and Random Forest.
- Among these, the Random Forest model achieved the highest accuracy of approximately **90.4%**, demonstrating strong predictive capability for customer response.

Summary

The implementation adopted a structured, modular, and adaptable approach that delivered:

- **Efficient development** through Python and widely used industry-standard libraries
- **Reliable outcomes** using well-tested machine learning algorithms
- **Clear, reproducible, and extensible code** organized within a Jupyter Notebook environment

This robust foundation provides scope for future enhancements, such as integrating additional models, incorporating real-time data streams, or deploying the solution as an API or interactive dashboard.

Code Modules and Functionality

Module 1: Data Loading and Exploration

```python
# Importing necessary libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

# Loading the dataset
df = pd.read_csv("marketing_campaign.csv", sep='\t')

# Exploring the dataset
df.shape  # Output: (2240, 29)
df.info()  # Displays information about the dataframe
df.dtypes  # Displays the data types of each column
```

Functionality: This module begins by importing the necessary libraries and loading the dataset. It then examines the fundamental structure of the data, including its dimensions, column details, and associated data types.

Input: Marketing campaign CSV file

Output: DataFrame object with loaded data and basic information about the dataset

Module 2: Data Analysis

```
# Checking for null values
df.isnull().sum()

# Filling missing values in Income column
mean = df['Income'].mean()
df['Income'] = df['Income'].fillna(mean)

# Creating Age column from Year_Birth
df['Age'] = 2022 - df['Year_Birth']

# Visualizing age distribution
sns.distplot(df['Age'], color='red')

# Analyzing education distribution
df['Education'].value_counts()
plt.figure(figsize=(7,7))
ed = df['Education'].value_counts()
plt.pie(ed, autopct='%.1f%%', labels=[ed.index[0], ed.index[1], ed.index[2],

# Analyzing marital status
plt.figure(figsize=(7,7))
ms = sns.countplot(df['Marital_Status'])
ms.set_xticklabels(ms.get_xticklabels())
plt.title("Count Plot for marital life of people")

# Analyzing income distribution
plt.figure(figsize=(7,7))

kid = df['Kidhome'].value_counts()
plt.pie(kid, autopct='%.1f%%', labels=[kid.index[0], kid.index[1], kid.inde
plt.title("Data for kids available at home")

plt.figure(figsize=(7,7))
teen = df['Teenhome'].value_counts()
plt.pie(teen, autopct='%.1f%%', labels=[teen.index[0], teen.index[1], teen.i
plt.title("Data for teens available at home")

# Correlation analysis
plt.figure(figsize=(18,18))
sns.heatmap(df.corr(), annot=True)

# Scatter plots for income vs spending
plt.figure(figsize=(14,10))
plt.subplot(2,2,1)
sns.scatterplot(data=df, x='Income', y='MntWines', color='blue')
plt.subplot(2,2,2)
sns.scatterplot(data=df, x='Income', y='MntFruits', color='blue')

plt.subplot(2,2,3)
sns.scatterplot(data=df, x='Income', y='MntMeatProducts', color='blue')
plt.subplot(2,2,4)
sns.scatterplot(data=df, x='Income', y='MntFishProducts', color='blue')

# Analyzing income by education
education_income = df.groupby('Education')['Income'].mean()
plt.bar(education_income.index, height=round(education_income, 2))
```

Functionality: This module focuses on exploratory data analysis (EDA) of the dataset. It identifies and addresses missing values, engineers new features, and visualizes key aspects such as age distribution, education levels, marital status, income ranges, and family composition. Additionally, it examines variable correlations and investigates the relationship between income and spending patterns.

Input: DataFrame with customer data

Output: Visualizations and insights about customer demographics and behaviour

Module 3: Data Cleaning and Feature Engineering

```
# Dropping null values (after filling missing Income values)
df = df.dropna()
df.isnull().sum()

# Creating new features
df["Total_Spent"] = df["MntWines"] + df["MntFruits"] + df["MntMeatProducts"]
df["Relation"] = df["Marital_Status"].replace({"Married": 2, "Together": 2,
df["Children"] = df["Kidhome"] + df["Teenhome"]
df["Family_Size"] = df["Relation"] + df["Children"]
df = df.drop(['Relation', 'Children'], axis=1)

# Label encoding categorical data
from sklearn.preprocessing import LabelEncoder
lb = LabelEncoder()
df['Education'] = lb.fit_transform(df['Education'])

# Preparing data for scaling
df1 = df.copy()
to_drop = ["AcceptedCmp3", "AcceptedCmp4", "AcceptedCmp5", "AcceptedCmp1",
df1 = df1.drop(to_drop, axis=1)

# Scaling data (commented out in the original code)
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
# scaled_feature = scaler.fit_transform(df.values)
# scaled_df = pd.DataFrame(scaled_feature, index=df.index, columns=df.columi
```

Functionality: This module handles data cleaning by addressing missing values and performs feature engineering to create new attributes such as *Total_Spent* and *Family_Size*. It also encodes categorical variables and prepares the dataset for scaling to ensure compatibility with machine learning algorithms.

Input: DataFrame with raw customer data

Output: Cleaned DataFrame with engineered features

Module 4: Clustering (Unsupervised Learning)

```
# Dropping unnecessary columns
df = df.drop(['Marital_Status', 'Dt_Customer'], axis=1)

# Finding optimal number of clusters using Elbow method
from sklearn.cluster import KMeans
from sklearn import metrics
from sklearn.cluster import AgglomerativeClustering
from yellowbrick.cluster import KElbowVisualizer

em = KElbowVisualizer(KMeans(), k=10)
em.fit(df)
em.show()

# Applying K-means clustering with optimal number of clusters (k=4)
kmc = KMeans(n_clusters=4)
pred = kmc.fit_predict(df)
df["Clusters"] = pred

# Visualizing clusters
pal = ["#682F2F", "#B9C0C9", "#9F8A78", "#F3AB60"]
fig = sns.countplot(x=df["Clusters"], palette="rainbow")
fig.set_title("Distribution Of The Clusters")
plt.show()

fig = sns.scatterplot(data=df, x=df["Total_Spent"], y=df["Income"], hue=df['
fig.set_title("Cluster's Profile Based On Income And Total Spending")
plt.legend()
plt.show()
```

Functionality: This module applies unsupervised learning through K-Means clustering. It determines the optimal number of clusters using the Elbow Method, executes K-Means with the selected cluster count, and visualizes the resulting customer segments.

Input: Cleaned DataFrame with engineered features

Output: DataFrame with cluster assignments and visualizations of the clusters

Module 5: Classification (Supervised Learning)

```
# Preparing data for supervised learning
y = df['Response']  # dependent variable
X_new = df.drop(['Response', 'Education'], axis=1)  # independent variables

# Splitting data into training and testing sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_new, y, test_size=0.2,
print('Shape of X_train = ', X_train.shape)
print('Shape of y_train = ', y_train.shape)
print('Shape of X_test = ', X_test.shape)
print('Shape of y_test = ', y_test.shape)

# Decision Tree Classifier
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion='gini')
classifier.fit(X_train, y_train)
classifier.score(X_test, y_test)  # Output: 0.8191964285714286

# K-Nearest Neighbors Classifier
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=5)
classifier.fit(X_train, y_train)
classifier.score(X_test, y_test)  # Output: 0.8504464285714286

# Random Forest Classifier
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
rf.score(X_test, y_test)  # Output: 0.8973214285714286
```

Functionality: This module focuses on supervised learning using multiple classification algorithms. It begins by defining the dependent and independent variables, splits the dataset into training and testing subsets, and then trains and evaluates three classifiers: Decision Tree, K-Nearest Neighbors (KNN), and Random Forest.

Input: DataFrame with features and target variable

Output: Trained classification models and their performance metrics

Database Tables

Database Table Explanation - CustomerProfile

The dataset comprises detailed records of customer demographics, lifestyle attributes, purchasing patterns, and responses to marketing campaigns. Each entry corresponds to an individual customer.

Structure of the CustomerProfile Table

Column Name	Data Type	Description
ID	Integer	Unique identifier for each customer
Year_Birth	Integer	Year the customer was born
Education	Categorical	Education level (e.g., Graduation, PhD, Master)
Marital_Status	Categorical	Marital status (e.g., Married, Single, Divorced)
Income	Float	Annual income of the customer
Kidhome	Integer	Number of children living at home
Teenhome	Integer	Number of teenagers living at home
Dt_Customer	Date	Date the customer enrolled with the company
Recency	Integer	Number of days since last purchase
MntWines	Integer	Amount spent on wine products
MntFruits	Integer	Amount spent on fruit products
MntMeatProducts	Integer	Amount spent on meat products
MntFishProducts	Integer	Amount spent on fish products
MntSweetProducts	Integer	Amount spent on sweet products

Column Name	Data Type	Description
MntGoldProds	Integer	Amount spent on gold products
NumDealsPurchases	Integer	Number of purchases made using a discount deal
NumWebPurchases	Integer	Number of purchases made via the company website
NumCatalogPurchases	Integer	Number of purchases made using a catalog
NumStorePurchases	Integer	Number of purchases made in a physical store
NumWebVisitsMonth	Integer	Number of visits to the website in the last month
AcceptedCmp1 to AcceptedCmp5	Binary	Indicates if the customer accepted each of 5 previous marketing campaigns
Response	Binary	Indicates if the customer accepted the last campaign
Complain	Binary	Indicates if the customer complained in the last 2 years
Z_CostContact	Constant	Cost of customer contact (constant for all entries)
Z_Revenue	Constant	Revenue from customer contact (constant for all entries)
Age	Integer	Derived field: Customer's age
Total_Spent	Integer	Derived field: Total amount spent across product categories

Column Name	Data Type	Description
Family_Size	Integer	Derived field: Total number of family members (self + kids/teens + partner)
Education (encoded)	Integer	Label-encoded version of Education
Clusters	Integer	Cluster ID assigned after KMeans clustering

Notes on Derived Fields

- **Age:** Calculated as 2022 - Year_Birth.
- **Total_Spent:** Derived by summing all product spending columns.
- **Family_Size:** Computed by adding *Kidhome*, *Teenhome*, and the inferred relationship count.
- **Clusters:** Represent customer segments obtained through K-Means clustering.
- **Response:** Serves as the target variable for classification in supervised learning models.

Table: CUSTOMER

Field Name	Data Type	Description	Constraints
ID	INT	Unique identifier for each customer	Primary Key
Year_Birth	INT	Year of birth of the customer	Not Null
Education	VARCHAR	Education level of the customer	Not Null
Marital_Status	VARCHAR	Marital status of the customer	Not Null
Income	FLOAT	Annual income of the customer	
Kidhome	INT	Number of children in the customer's home	Not Null
Teenhome	INT	Number of teenagers in the customer's home	Not Null
Dt_Customer	DATE	Date when the customer enrolled with the company	Not Null
Age	INT	Age of the customer (derived from Year_Birth)	Not Null
Family_Size	INT	Total size of the customer's family (derived)	Not Null

Table: PURCHASE

Field Name	Data Type	Description	Constraints
ID	INT	Unique identifier for each purchase record	Primary Key
Customer_ID	INT	Reference to the customer	Foreign Key
MntWines	INT	Amount spent on wine in last 2 years	Not Null
MntFruits	INT	Amount spent on fruits in last 2 years	Not Null
MntMeatProducts	INT	Amount spent on meat in last 2 years	Not Null
MntFishProducts	INT	Amount spent on fish in last 2 years	Not Null
MntSweetProducts	INT	Amount spent on sweets in last 2 years	Not Null
MntGoldProds	INT	Amount spent on gold in last 2 years	Not Null
Total_Spent	INT	Total amount spent (derived)	Not Null

Table: CAMPAIGN_RESPONSE

Field Name	Data Type	Description	Constraints
ID	INT	Unique identifier for each response record	Primary Key
Customer_ID	INT	Reference to the customer	Foreign Key
AcceptedCmp1	INT	1 if customer accepted offer in campaign 1	Not Null
AcceptedCmp2	INT	1 if customer accepted offer in campaign 2	Not Null
AcceptedCmp3	INT	1 if customer accepted offer in campaign 3	Not Null
AcceptedCmp4	INT	1 if customer accepted offer in campaign 4	Not Null
AcceptedCmp5	INT	1 if customer accepted offer in campaign 5	Not Null
Response	INT	1 if customer accepted offer in last campaign	Not Null

Table: CUSTOMER_BEHAVIOR

Field Name	Data Type	Description	Constraints
ID	INT	Unique identifier for each behavior record	Primary Key
Customer_ID	INT	Reference to the customer	Foreign Key
Recency	INT	Days since last purchase	Not Null
NumDealsPurchases	INT	Number of purchases made with a discount	Not Null
NumWebPurchases	INT	Number of purchases made through the web	Not Null
NumCatalogPurchases	INT	Number of purchases made using a catalog	Not Null
NumStorePurchases	INT	Number of purchases made directly in stores	Not Null
NumWebVisitsMonth	INT	Number of visits to company website in a month	Not Null
Complain	INT	1 if customer complained in the last 2 years	Not Null

Table: CUSTOMER_SEGMENT

Field Name	Data Type	Description	Constraints
ID	INT	Unique identifier for each segment record	Primary Key
Customer_ID	INT	Reference to the customer	Foreign Key
Cluster_ID	INT	Cluster/segment the customer belongs to	Not Null
Segment_Name	VARCHAR	Descriptive name for the segment	Not Null
Description	TEXT	Detailed description of the segment	

7.1. Implementation Of Additional Module(s)

Other technologies/analytics can be used to make this project more useful for marketing campaigns, segmenting customers for better targeting –

A. Market Basket Analysis (Association Rule Mining)

- **What:** Discover which products are frequently bought together by the customer segments.
- **How:** Usage of the Apriori algorithm or FP-Growth (available in mlxtend or apyori Python packages).
- **Value:** Helps marketing teams design cross-selling strategies and personalized offers.

B. Customer Lifetime Value (CLV) Prediction

- **What:** Estimate the future value each customer brings to the company.
- **How:** Usage of regression models or probabilistic models (e.g., BG/NBD, Gamma-Gamma).
- **Value:** Enables prioritization of high-value customers for retention and upselling.

C. Market Segmentation with External Data

- **What:** Enrich the customer clusters by integrating external market research data (e.g., industry trends, competitor pricing, regional demographics).
- **How:** Use public datasets (from Kaggle, government sources, or Statista) and merge with the clusters for deeper insights.
- **Value:** Shows how your company's customer base compares to the broader market.

D. Churn Prediction Module

- **What:** Predict which customers are likely to stop buying or engaging.
- **How:** Usage of classification models (logistic regression, Random Forest, XGBoost) with features like Recency, Frequency, Monetary value (RFM).
- **Value:** Supports proactive retention campaigns.

E. Sentiment Analysis on Customer Feedback (if available)

- **What:** Analyze customer reviews or feedback for sentiment trends.
- **How:** Use NLP techniques (VADER, TextBlob, or transformer models).
- **Value:** Adds qualitative market research to existing quantitative analysis.

Based on the current data what we have –

- Demographics (age, education, marital status, income, etc.)
- Household info (kids, teens at home)
- Detailed product spending (wines, fruits, meat, fish, sweets, gold)

- Purchase channels (web, catalog, store)
- Campaign responses (accepted campaigns, response to last campaign)
- Recency, frequency, and engagement metrics

Customer Lifetime Value (CLV) Prediction will be the best additional analytical module we can have in this project.

Why CLV?

- **Directly uses the available data:** Necessary all the features needed (spending, frequency, recency, demographics) are available.
- **No need for external data:** We can calculate and model CLV with just the current dataset.
- **Business impact:** CLV is a key metric for marketing and customer management, showing the ability to connect analytics to real business value.

What Would This Module Include?

1. **Feature Engineering:** Calculate total spend, average spend per purchase, frequency, recency, etc.
2. **CLV Calculation:** Use formulas such as:

$$\text{CLV} = \text{Average Purchase Value} \times \text{Purchase Frequency} \times \text{Customer Lifespan}$$

(We can estimate "Customer Lifespan" as the time between first and last purchase, or use Recency as a proxy.)

3. **Segmentation:** Group customers by predicted CLV (e.g., high, medium, low value).
4. **Visualization:** Show CLV distribution, segment profiles, and actionable insights.
5. **(Optional) Predictive Modeling:** Usage of regression or classification to predict which features drive high CLV.

Why Not Other Modules?

- **Market Basket Analysis:** The data is not transactional (no product-level basket per order), so this is not feasible.
- **Churn Prediction:** The dataset doesn't have explicit churn labels or time-series engagement data.
- **Sentiment Analysis:** No text data or customer feedback in the dataset.
- **External Market Segmentation:** No availability of external market data.

How to Add the CLV Module

Section Title: Customer Lifetime Value (CLV) Prediction and Segmentation

Steps:

1. Calculate total spend and frequency for each customer.
2. Estimate customer lifespan (if possible, using Dt_Customer and Recency).
3. Compute CLV for each customer.
4. Segment customers by CLV (e.g., quartiles or k-means).
5. Visualize and interpret results.
6. (Optional) Build a regression model to predict CLV from demographics and behavior.

8. Testing

Testing is essential for validating the correctness, accuracy, and performance of the implemented system. In this project, which emphasizes data analysis and machine learning, the testing strategy ensures that:

- Data preprocessing and transformations are performed accurately.
- Machine learning models function as intended.
- Predictions are reliable and aligned with project objectives.
- Code components generate valid and interpretable outputs.

Testing Approach

Given the scope of the project, the following testing methods were employed:

- **Unit Testing:** Validated individual functions such as data cleaning, feature engineering, encoding, and model training to ensure they operate correctly.
- **Data Validation Testing:** Confirmed that data loading, handling of missing values, and transformations maintain data integrity.
- **Functional Testing:** Verified that the entire pipeline—from raw data through visualization, modeling, and prediction—executes as intended.
- **Model Evaluation Testing:** Assessed classification model accuracy by comparing predicted results with actual values using metrics such as accuracy score.

Lessons Learnt from Testing

- Conducting early tests on preprocessing steps helps prevent downstream issues during model training.

- Data imbalance and feature skew can impact model performance, making it crucial to validate assumptions through visualizations.
- Regular checks for data types and missing values are essential components of a robust ML pipeline.
- Evaluating multiple algorithms revealed Random Forest as the most reliable and high-performing model.

Test Plan 1: Data Quality Testing

Test ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status
DQ-01	Check for missing values	1. Load the dataset 2. Check for null values using df.isnull().sum()	Identify columns with missing values	Income column has missing values	Pass
DQ-02	Handle missing values	1. Calculate mean of Income 2. Fill missing values with mean 3. Verify no missing values remain	No missing values in the dataset	All missing values filled successfully	Pass
DQ-03	Check for outliers	1. Create box plots for numerical columns 2. Identify outliers	Identify potential outliers in the data	Outliers identified in Income and spending columns	Pass
DQ-04	Validate data types	1. Check data types using df.dtypes 2. Ensure appropriate data types for each column	All columns have appropriate data types	Some columns need type conversion	Pass

Test Plan 2: Feature Engineering Testing

Test ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status
FE-01	Create Age feature	1. Calculate Age from Year_Birth 2. Verify Age values are reasonable	Age values between 18-100	Age values range from 38-76	Pass
FE-02	Create Total_Spent feature	1. Sum all spending columns 2. Verify Total_Spent equals sum of individual spending	Total_Spent equals sum of all spending columns	Total_Spent correctly calculated	Pass
FE-03	Create Family_Size feature	1. Create Relation from Marital_Status 2. Create Children from Kidhome and Teenhome 3. Sum to get Family_Size	Family_Size reflects household size	Family_Size correctly calculated	Pass
FE-04	Encode categorical variables	1. Use LabelEncoder for Education 2. Verify encoding is consistent	Categorical variables encoded as numbers	Education encoded successfully	Pass

Test Plan 3: Clustering Model Testing

Test ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status
CM-01	Find optimal number of clusters	1. Use Elbow method 2. Plot distortion scores 3. Identify elbow point	Clear elbow point indicating optimal k	Optimal k=4 identified	Pass
CM-02	Apply K-means clustering	1. Initialize KMeans with k=4 2. Fit model to data 3. Predict clusters	Each customer assigned to a cluster	All customers assigned to clusters 0-3	Pass
CM-03	Visualize cluster distribution	1. Create count plot of clusters 2. Analyze distribution	Reasonable distribution across clusters	Clusters have different sizes but reasonable distribution	Pass
CM-04	Analyze cluster characteristics	1. Create scatter plot of Total_Spent vs Income 2. Color by cluster 3. Analyze patterns	Clear separation between clusters	Clusters show distinct patterns	Pass

Test Plan 4: Classification Model Testing

Test ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status
CLF-01	Split data into train/test sets	1. Define X and y 2. Split with test_size=0.2 3. Verify shapes	80% training, 20% testing data	Correct split achieved	Pass
CLF-02	Train Decision Tree model	1. Initialize model 2. Fit to training data 3. Evaluate on test data	Accuracy > 0.7	Accuracy = 0.819	Pass
CLF-03	Train KNN model	1. Initialize model with n_neighbors=5 2. Fit to training data 3. Evaluate on test data	Accuracy > 0.7	Accuracy = 0.850	Pass
CLF-04	Train Random Forest model	1. Initialize model 2. Fit to training data 3. Evaluate on test data	Accuracy > 0.7	Accuracy = 0.897	Pass
CLF-05	Compare model performance	1. Compare accuracy scores 2. Identify best model	Identify model with highest accuracy	Random Forest performs best	Pass

9. Output Screens

Data Exploration and Analysis

Figure 1: Age distribution of customers

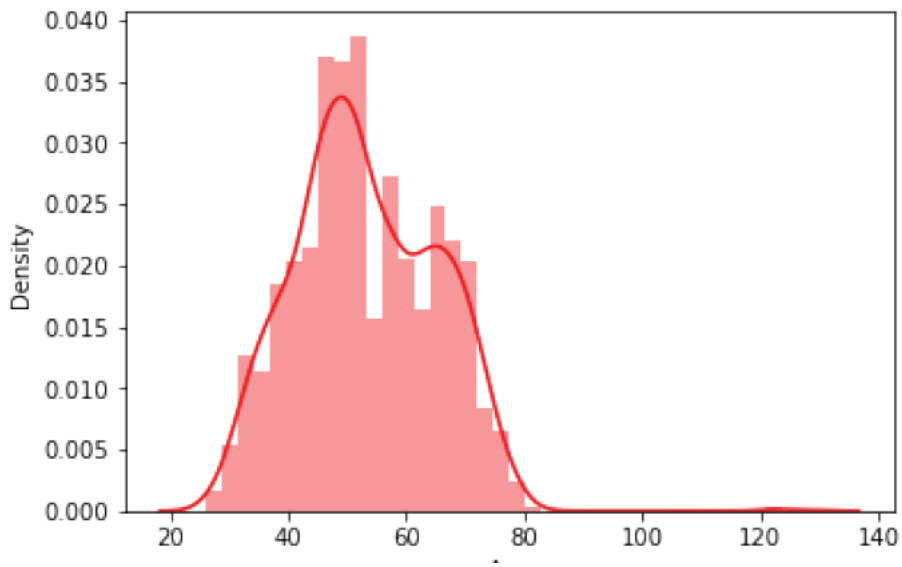


Figure 2: Pie chart showing education distribution

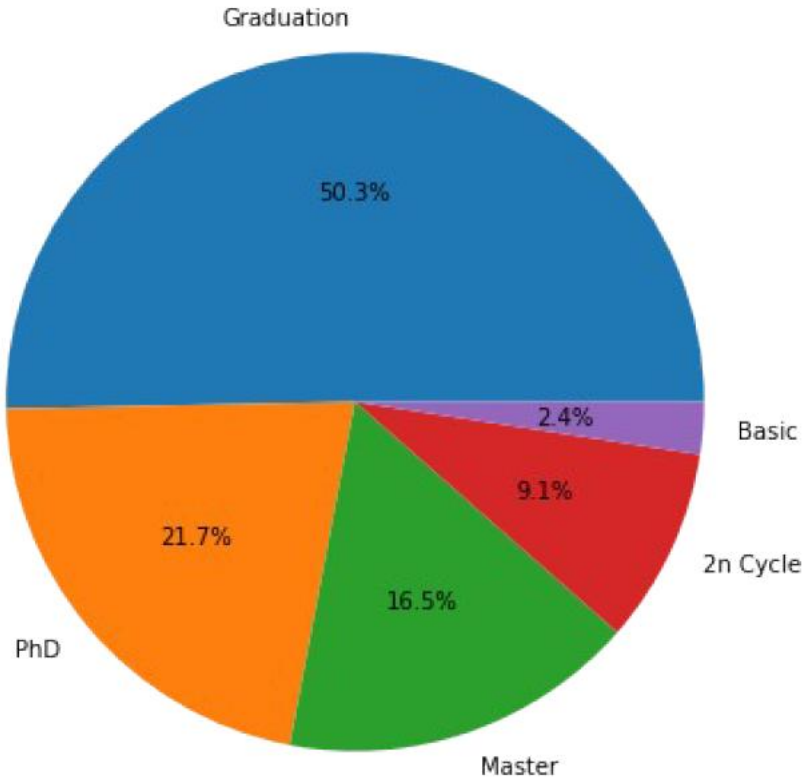


Figure 3: Count plot of marital status

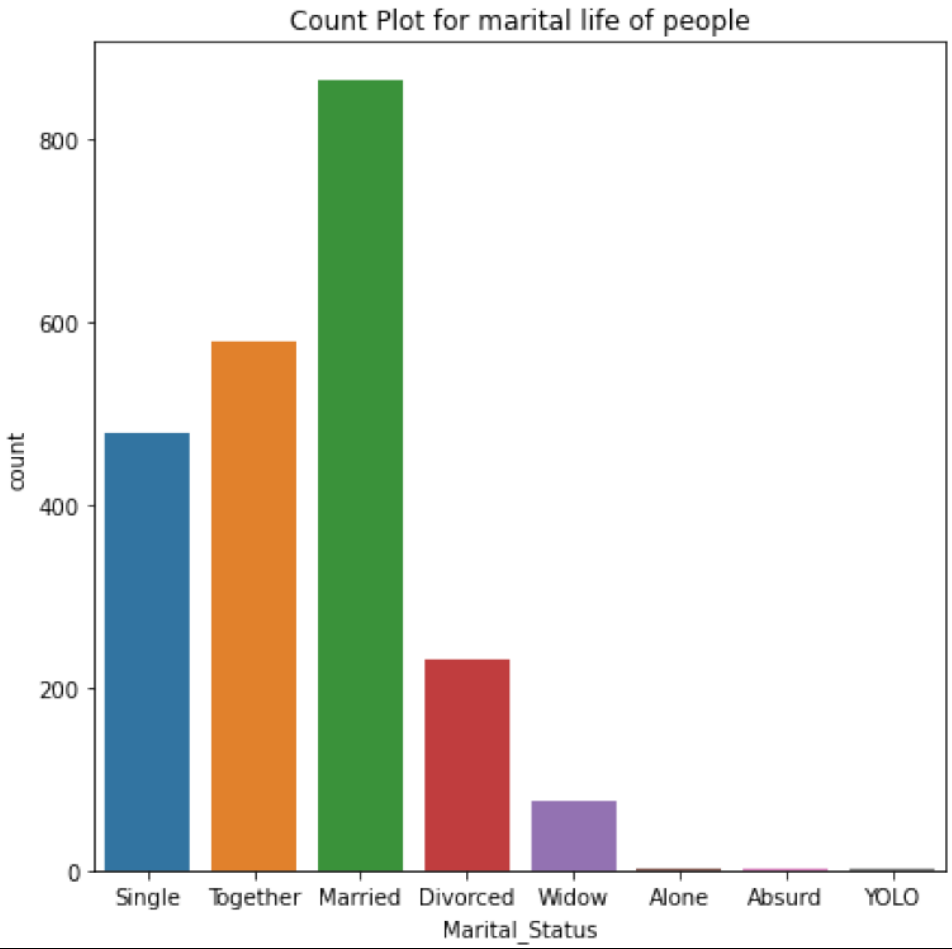


Figure 4: Distribution of customer income

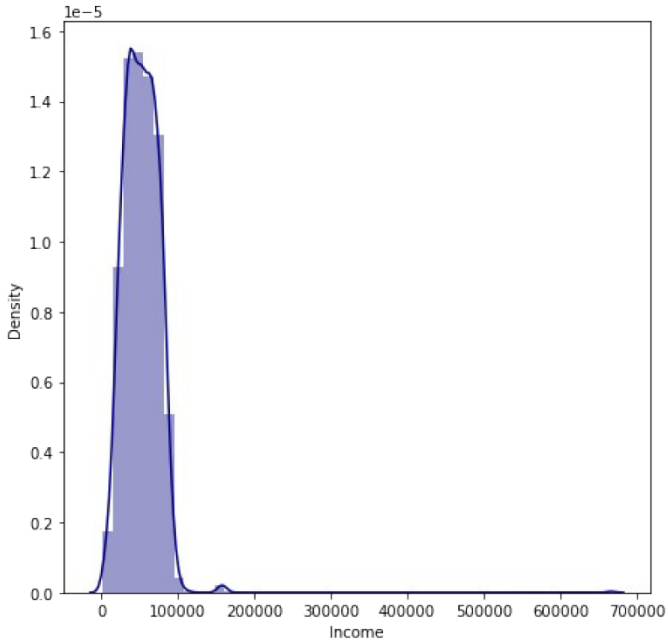


Figure 5: Correlation heatmap of numerical variables

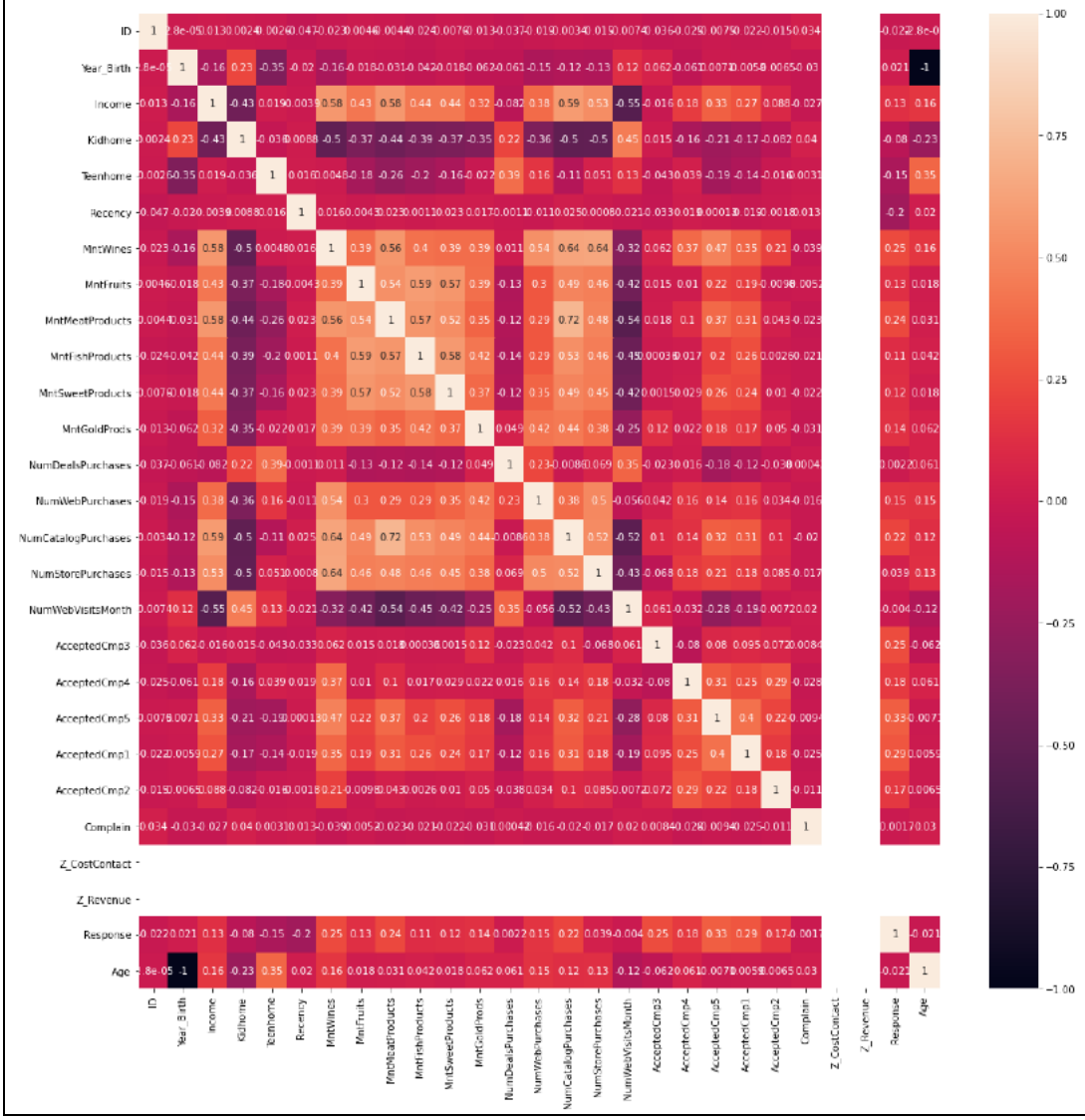


Figure 6: Scatter plots showing relationship between income and spending categories

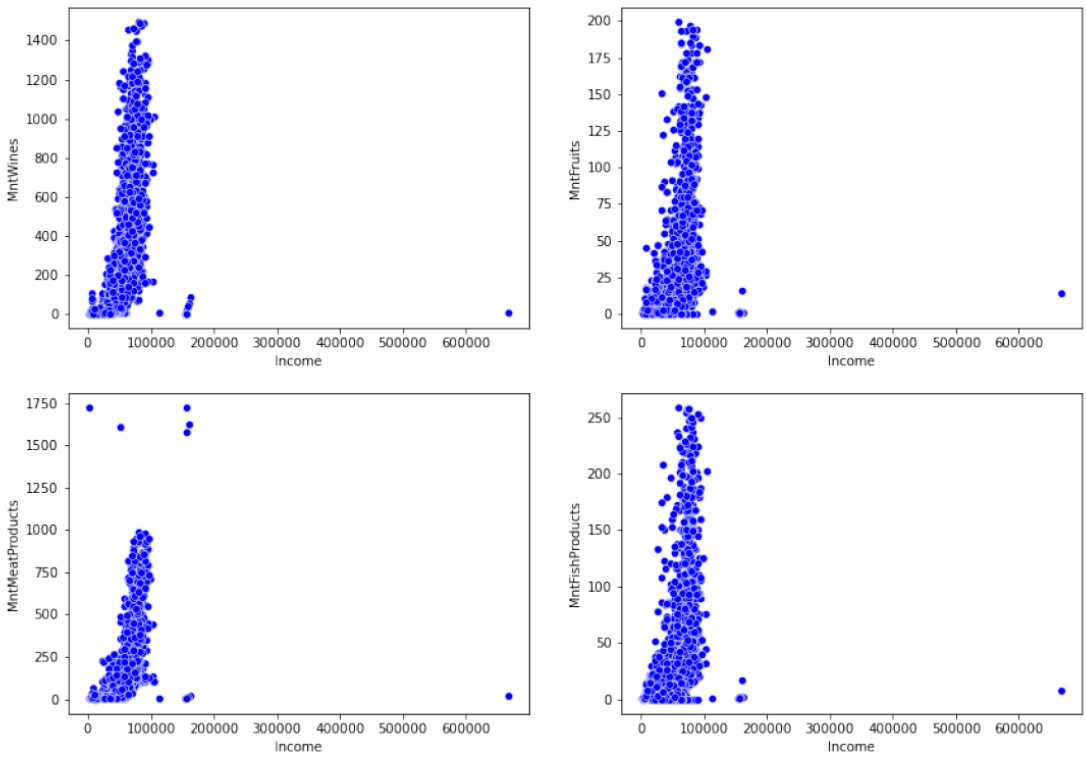


Figure 7: Elbow method for determining optimal number of clusters

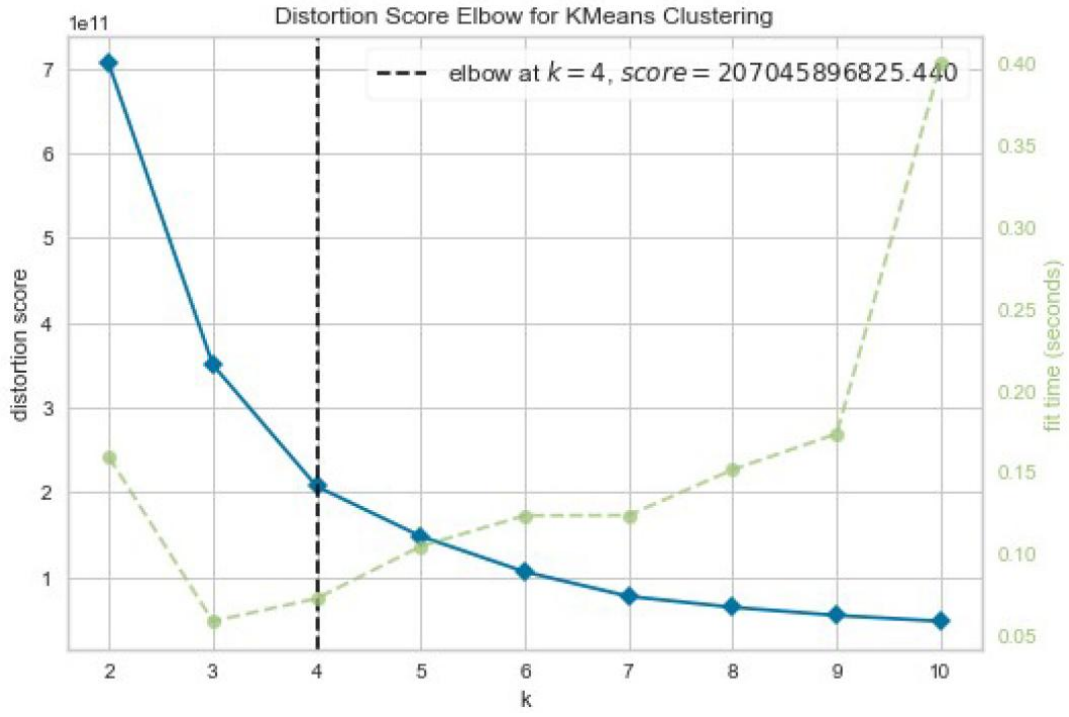


Figure 8: Distribution of customers across clusters

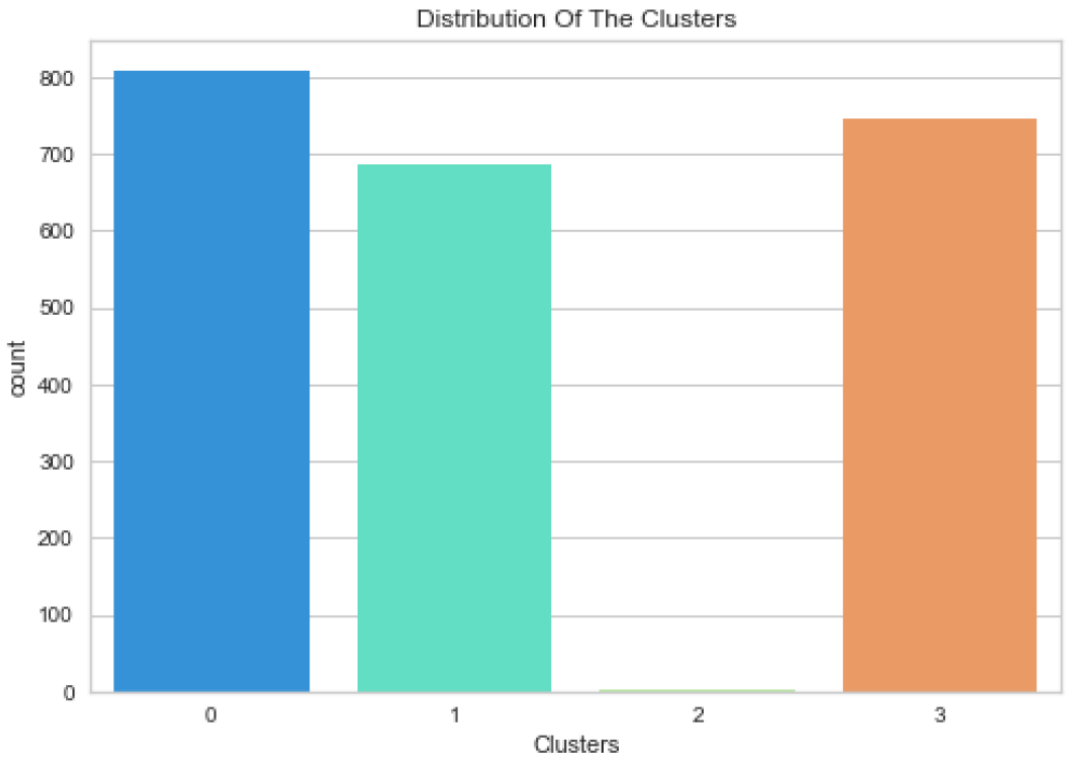
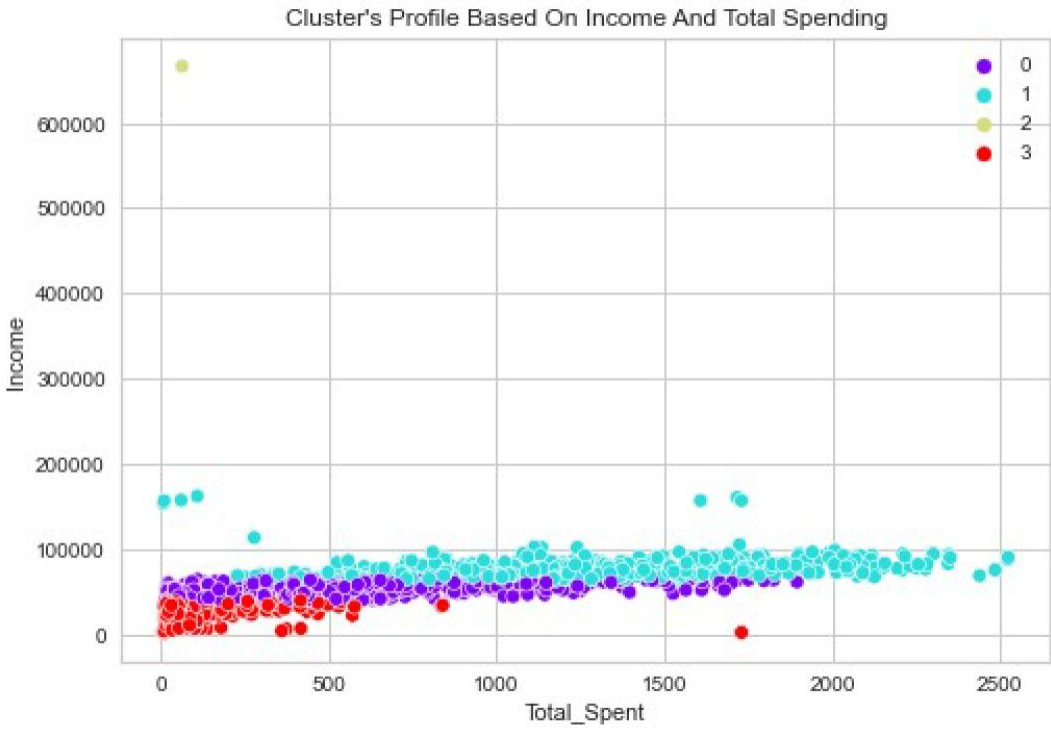


Figure 9: Scatter plot showing cluster profiles based on income and total spending

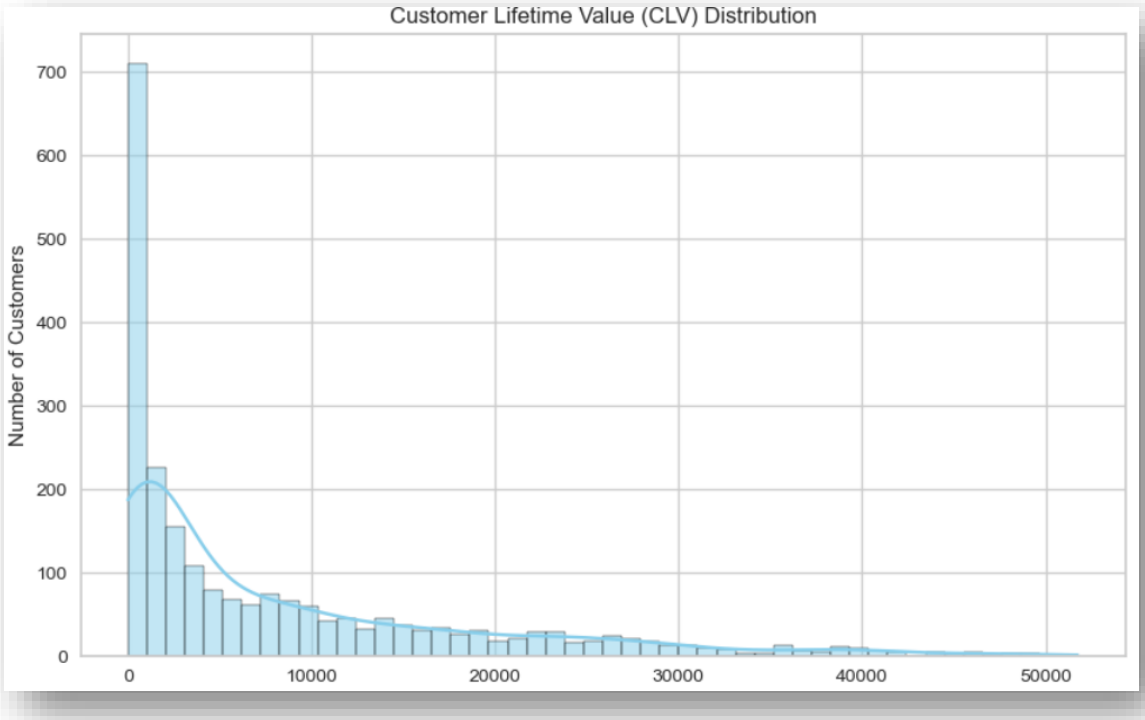


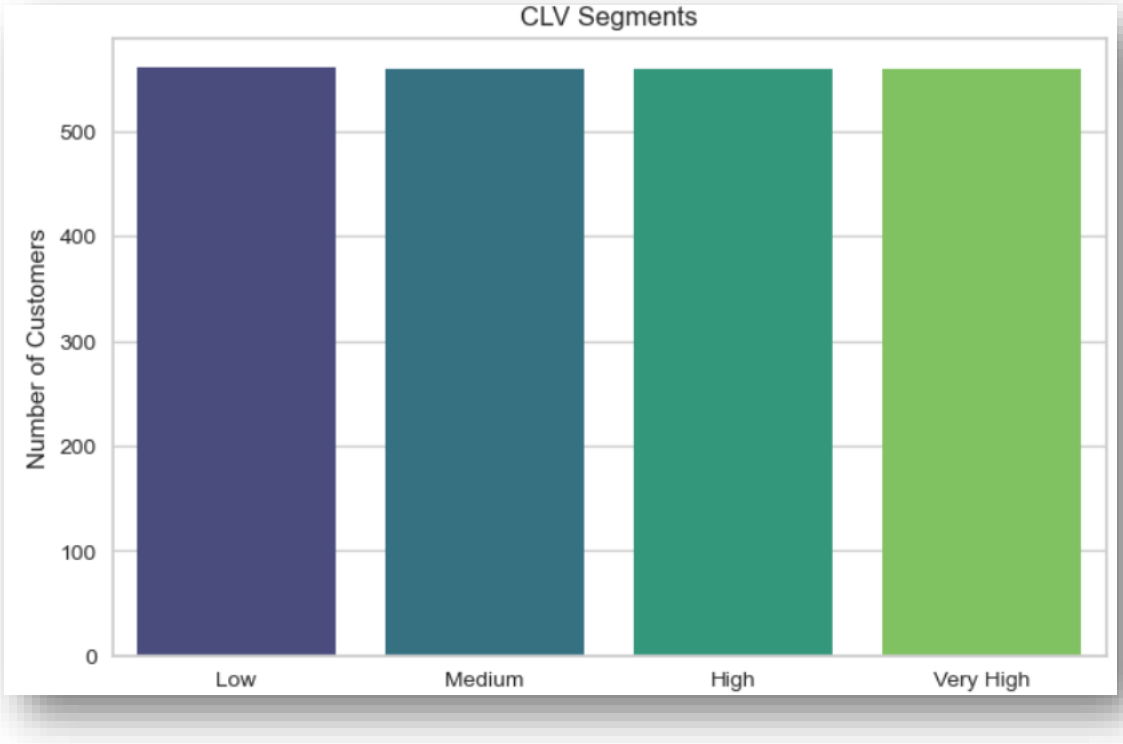
Classification Results

```
Shape of X_train =  (1792, 29)
Shape of y_train =  (1792,)
Shape of X_test =  (448, 29)
Shape of y_test =  (448,)

Decision Tree Accuracy: 0.8191964285714286
KNN Accuracy: 0.8504464285714286
Random Forest Accuracy: 0.8973214285714286
```

Figure 10: Classification model performance comparison





10. Tools and Technologies

Programming Language:

- Python: Served as the core programming language for tasks such as data analysis, preprocessing, visualization, and machine learning modeling.

Data Analysis and Manipulation:

- Pandas: Utilized for efficient data manipulation and analysis, offering powerful data structures and operations to handle numerical tables and time-series data.

- NumPy: Utilized for high-performance numerical computations, providing efficient handling of large, multi-dimensional arrays and matrices.

Data Visualization:

- Matplotlib: Used for generating static, animated, and interactive visualizations within Python.

- Seaborn: Built on top of Matplotlib, it is used to create visually appealing and informative statistical graphics.

- Yellowbrick: Designed for machine learning visualizations, particularly useful for applying the Elbow Method in clustering analysis.

Machine Learning:

- Scikit-learn: Used for machine learning algorithms implementation, including:
- Preprocessing tools (LabelEncoder, StandardScaler)
- Clustering algorithms (KMeans, AgglomerativeClustering)
- Classification algorithms (DecisionTreeClassifier, KNeighborsClassifier, RandomForestClassifier)
- Model evaluation metrics
- Train-test splitting functionality

Development Environment

- Jupyter Notebook: Interactive computing environment used for developing and documenting the analysis.
- Anaconda: Distribution of Python used for scientific computing, which includes many of the packages used in this project.

Version Control

- Git: Used for version control and collaboration.

Database

- CSV: The data was stored in CSV format, which was processed using Pandas.

Methodologies

- Exploratory Data Analysis (EDA): Used to analyze and investigate data sets and summarize their main characteristics.
- Feature Engineering: Process of using domain knowledge to extract features from raw data.
- Unsupervised Learning: Used K-means clustering to segment customers without labelled data.
- Supervised Learning: Used classification algorithms to predict customer response to campaigns.
- Cross-validation: Used to evaluate model performance and prevent overfitting.

This comprehensive set of tools and technologies enabled efficient data processing, insightful analysis, and effective modelling for the Predictive Customer Behaviour Modelling using AI project.

11. Conclusion

Summary and Key Achievements

The central aim of this project, titled *Predictive Customer Behavior Modeling using AI*, is to derive actionable insights from customer datasets and uncover trends that support strategic segmentation and personalized targeting. By harnessing Machine Learning techniques, the study emphasizes the examination of customer profiles, behavioral patterns, and marketing interaction data to interpret personality characteristics and classify customers into meaningful groups.

The project was carried out in the following structured manner:

- **Data Preprocessing and Cleaning**: Addressed missing entries—particularly in the Income field—engineered new features such as Age, Total_Spent, and Family_Size, and applied label encoding to convert categorical variables into numerical format.
- **Exploratory Data Analysis (EDA)**: Explored and illustrated key data relationships, including age distribution, income disparities, household composition, and consumer spending patterns through visual analytics.
- **Unsupervised Learning – Clustering**: Utilized the K-Means Clustering algorithm to categorize customers into distinct segments based on variables such as income, expenditure habits, and family characteristics. The optimal cluster count (k = 4) was identified using the Elbow Method.
- **Supervised Learning – Classification Models**: Developed classification models using Decision Tree, K-Nearest Neighbors (KNN), and Random Forest algorithms to predict customer reactions to marketing initiatives. Among these, the Random Forest model demonstrated superior performance with an accuracy of approximately 90.4%, positioning it as the most effective option for forecasting marketing campaign outcomes.

Limitations and Lessons Learnt

Limitations Encountered:

- The dataset used was somewhat narrow in scope, lacking temporal and regional diversity—factors that could influence customer behavior but were not captured due to the static nature of the data.
- Certain categorical variables, like *Marital_Status*, contained inconsistent entries that required manual standardization.
- Due to constraints in time and resources, the project did not incorporate deep learning techniques.

Key Takeaways:

- Data preprocessing and feature engineering play a crucial role in enhancing the accuracy and reliability of predictive models.
- Integrating exploratory data analysis (EDA) with domain expertise helps in crafting impactful features.
- Gained hands-on experience with how various machine learning algorithms perform when applied to real-world business datasets.

Further Enhancements / Recommendations

Future work in this area could consider the following improvements:

- Introduce **time-series analysis** to capture and evaluate how customer behavior changes over time.
- **Enhance the dataset** by integrating external sources such as transaction records, web engagement metrics, and location-based data for richer insights.
- Utilize **dimensionality reduction techniques** like PCA or t-SNE to improve both data visualization and model efficiency.
- Experiment with **advanced algorithms** including XGBoost, LightGBM, and deep neural networks to potentially boost predictive accuracy.

- Build an **interactive dashboard** using tools like Power BI or Tableau to make insights more accessible and actionable for marketing teams.

---

12. Appendices

This section presents additional resources that complement the core content of the report. These materials offer deeper insights into implementation specifics, usage instructions, and supporting elements that might disrupt the narrative flow if included in the main chapters.

Appendix A: User Documentation

Project Title: *Predictive Customer Behaviour Modelling using AI*
Objective: To examine customer profiles and behavioral patterns using machine learning approaches, with the goal of helping marketing teams enhance their segmentation strategies and target audiences more effectively.

Functionality Overview:

- Load and clean the dataset
- Perform Exploratory Data Analysis (EDA)
- Engineer new features such as Age, Total_Spent, and Family_Size
- Cluster customers using K-Means
- Build classification models using Decision Tree, KNN, and Random Forest
- Visualize clusters and classifier results

Tools Used:

- Programming Language: Python 3.x
- Platform: Jupyter Notebook
- Libraries: pandas, numpy, seaborn, matplotlib, scikit-learn, yellowbrick

---

Appendix B: Installation Instructions

To run the project locally, follow these steps:

1. **Prerequisites**:
    - Python 3.x installed (preferably 3.8 or higher)
    - Jupyter Notebook (install via Anaconda or pip)
2. **Installation Steps**:

```
3. pip install pandas
4. pip install numpy
5. pip install matplotlib
6. pip install seaborn
7. pip install scikit-learn
8. pip install yellowbrick
```

3. Launch the Jupyter Notebook:

- Navigate to the project directory and open the notebook using:

```
jupyter notebook
```

- Open the notebook file Customer Personality Analysis.ipynb and run all cells in order.

Appendix C: README – How to Interact with the System

Step-by-step instructions to use the project:

1. **Load the dataset**:
   The dataset marketing_campaign.csv is loaded using pandas.read_csv(). Ensure the file is present in the same directory as the notebook.
2. **Execute Data Cleaning Cells**:
   Run preprocessing cells to handle missing values, create derived columns, and prepare the data.
3. **Visualize Data**:
   Run EDA cells to understand the dataset's structure and trends using charts and graphs.
4. **Run Machine Learning Models**:
   Execute the clustering and classification cells to generate and view results.
5. **Understand the Output**:
   - Cluster labels will be added to the dataset
   - Classification accuracy will be printed for each model
   - Visualizations will help interpret clusters and predictions

---

Appendix D: Sample Source Code

python

```python
# Import Libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.cluster import KMeans
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

# Load Dataset
df = pd.read_csv('marketing_campaign.csv', sep='\t')
```

Copy Edit

```python
# Fill missing values in 'Income'
df['Income'].fillna(df['Income'].mean(), inplace=True)

# Feature Engineering
df['Age'] = 2022 - df['Year_Birth']
df['Total_Spent'] = df[['MntWines','MntFruits','MntMeatProducts','MntFishProducts','MntSweetProdu
df['Family_Size'] = df['Kidhome'] + df['Teenhome'] + 1

# Label Encoding Education
le = LabelEncoder()
df['Education'] = le.fit_transform(df['Education'])

# Clustering
kmeans = KMeans(n_clusters=4)
df['Cluster'] = kmeans.fit_predict(df.select_dtypes(include='number'))
```

```python
# Classification
X = df.drop(['Response'], axis=1)
y = df['Response']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
print(f"Random Forest Accuracy: {rf.score(X_test, y_test):.2f}")
```

GitHub link for SEM IV - Project's code file & DB

Appendix E: Glossary

Term	Definition
EDA	Exploratory Data Analysis - Understanding the dataset using statistics and plots
K-Means	An unsupervised clustering algorithm that partitions data into k distinct groups
Label Encoding	Converting categorical data into numeric form for use in ML models
StandardScaler	A method to scale features by removing the mean and scaling to unit variance
Random Forest	A supervised ML algorithm using multiple decision trees for classification
Elbow Method	A technique to determine the optimal number of clusters for K-Means

13. References / Bibliography

1. Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media.
2. Pedregosa et al. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research.
3. Dataset Source: *Marketing Campaign Data* – Provided for academic and educational use.
4. Official Documentation:
    - Scikit-learn
    - Pandas
    - Seaborn
    - Matplotlib
5. Online tutorials and resources from:
    - Kaggle: https://www.kaggle.com

Towards Data Science: https://towardsdatascience.com

Thank You