CUSTOMER PERSONALITY ANALYSIS

USING MACHINE LEARNING

AN INTERNSHIP 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

MAY-JUNE 2025 Exam (3rd SEM)

DIRECTORATE OF ONLINE EDUCATION

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

This Internship Report titled **"Customer Personality Analysis using Machine Learning"** of **"Soumen Chatterjee [EC2432251010407]"**, who carried out the Internship 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 internship 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)

INTERNSHIP OFFER LETTER

iNeuron

iNeuron Intelligence Pvt Ltd

17th Floor Tower A, Brigade Signature Towers,
Sannatammanahalli, Bengaluru, Karnataka -
562129.

DATE: 15th February 2025

Internship Offer Letter

Dear Soumen Chatterjee,

Following your application, we are pleased to inform you that you have been considered for an internship with iNeuron for **Customer Personality Analysis** project. As a result, you will be contributing to our project from 15th February 2025.

As a part of your internship, you will be proactively contributing to your selected project, besides product development & PoCs. In addition, you will be required to complete performance & learning goals for your current project with us.

We hope that your association with the company will be successful and rewarding.

Regards,
Sudhanshu Kumar
CEO & Chief AI Engineer at iNeuron.ai

I accept the offer with the company on the terms and conditions set out in this letter.

Soumen Chatterjee

DATE: 15th February 2025

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 internship.

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

Table of Contents

1. Abstract

Customer Personality Analysis is a data-driven approach to understanding consumer behaviour, preferences, and purchasing patterns. This project aims to analyse customer data to segment consumers into distinct groups, allowing businesses to target them effectively. The study leverages machine learning techniques, including clustering (unsupervised learning) and classification (supervised learning), to gain insights from customer demographics and purchase behaviour.

Key accomplishments of this project include:

- **Data preprocessing**, including cleaning, handling missing values, and feature engineering.
- **Unsupervised learning** using K-Means clustering to segment customers into groups.
- **Supervised learning** using Decision Tree, K-Nearest Neighbours (KNN), and Random Forest classifiers to predict customer response.
- **Visualization and analysis** of key customer attributes such as income, spending behaviour, and family structure.

My key contributions include data preparation, model selection, and result interpretation. The findings help in targeted marketing and customer relationship management strategies.

2. Introduction

- **Background**

Businesses increasingly rely on data-driven decision-making to understand their customers better. Customer Personality Analysis plays 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 is a critical phase in any software or data science project. It involves examining the current business scenario, identifying limitations of the existing approach, and proposing an improved solution through technology and innovation. This section includes an overview of the existing system (if any), the proposed system for customer personality analysis, and the results of a feasibility study validating the project's viability.

A. Existing System

Traditionally, businesses rely on basic customer profiling methods using limited demographic attributes such as age, gender, and location. These approaches are typically **manual or rule-based**, lacking the depth and accuracy required to truly understand complex customer behavior. The limitations of such systems include:

- **No Personalization**: Marketing campaigns are often generalized rather than tailored to specific customer segments.
- **Inefficient Targeting**: Without deeper insights, resources are wasted on customers who are unlikely to respond.
- **Data Underutilization**: Many businesses collect extensive customer data but fail to apply advanced analytics for better decision-making.
- **No Predictive Capability**: Traditional systems cannot predict which customers are more likely to respond to a campaign or make a purchase.

B. Proposed System

The proposed system, *Customer Personality Analysis using Machine Learning*, is designed to overcome these limitations by leveraging modern data science techniques. It provides a **data-driven approach** to customer segmentation and behavioral prediction.

Key Features of the Proposed System:

- **Data-Driven Insights**: Uses historical customer data including spending patterns, demographics, and online activity.

- **Customer Segmentation**: Applies clustering algorithms (e.g., KMeans) to group customers based on similar behaviors and preferences.
- **Marketing Response Prediction**: Implements supervised learning models (e.g., Random Forest) to predict customer responses to marketing campaigns.
- **Feature Engineering**: Derives new features such as Age, Total_Spent, and Family_Size to improve analytical depth.
- **Visualization**: Provides graphical insights for easy interpretation of complex relationships and patterns.
- **Model Evaluation**: Measures model performance using accuracy scores, ensuring reliable results.

Benefits:

- Enables **personalized marketing**
- Improves **customer engagement**
- Reduces **marketing costs**
- Increases **conversion rates**
- Enhances **strategic decision-making**

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.

Feasibility Type	Assessment
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

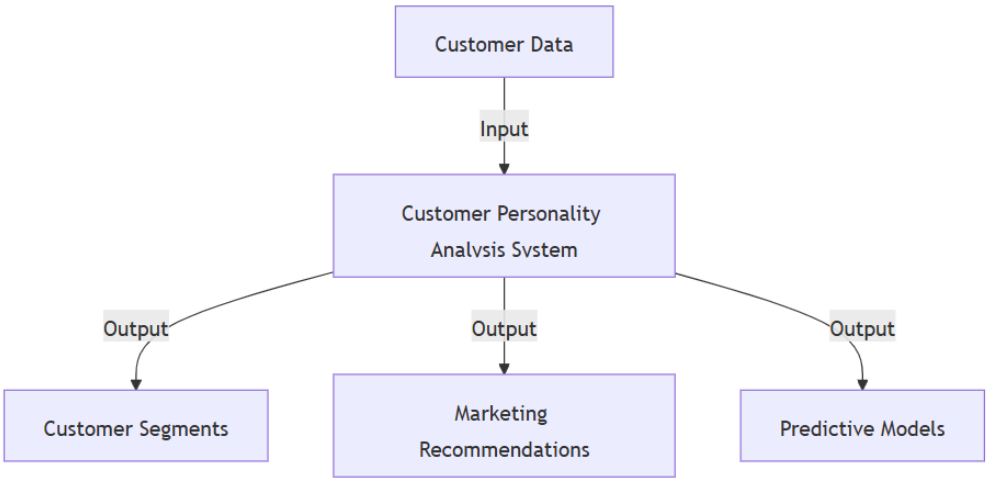
- Evaluating classification models using accuracy scores
- Understanding customer segments for targeted marketing strategies

6. Design

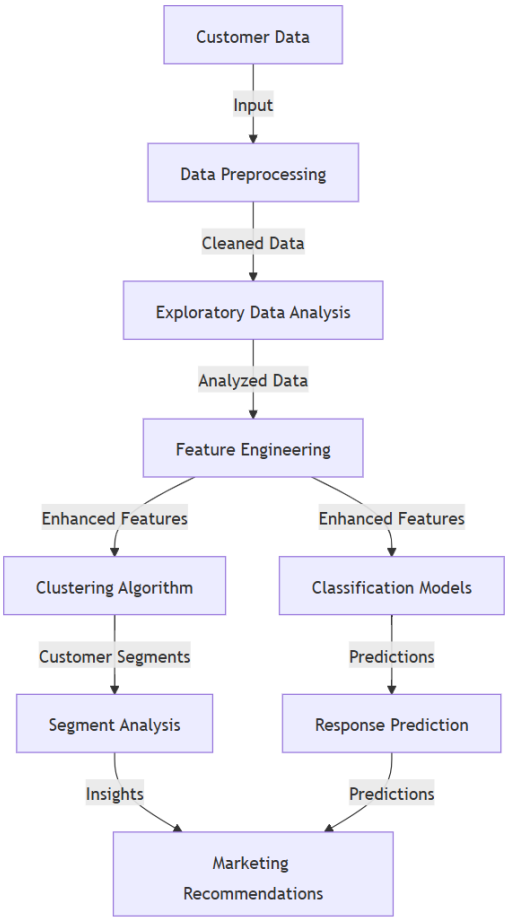
6. a) System Design

Data Flow Diagram (DFD):

Level 0 DFD - Customer Personality Analysis System

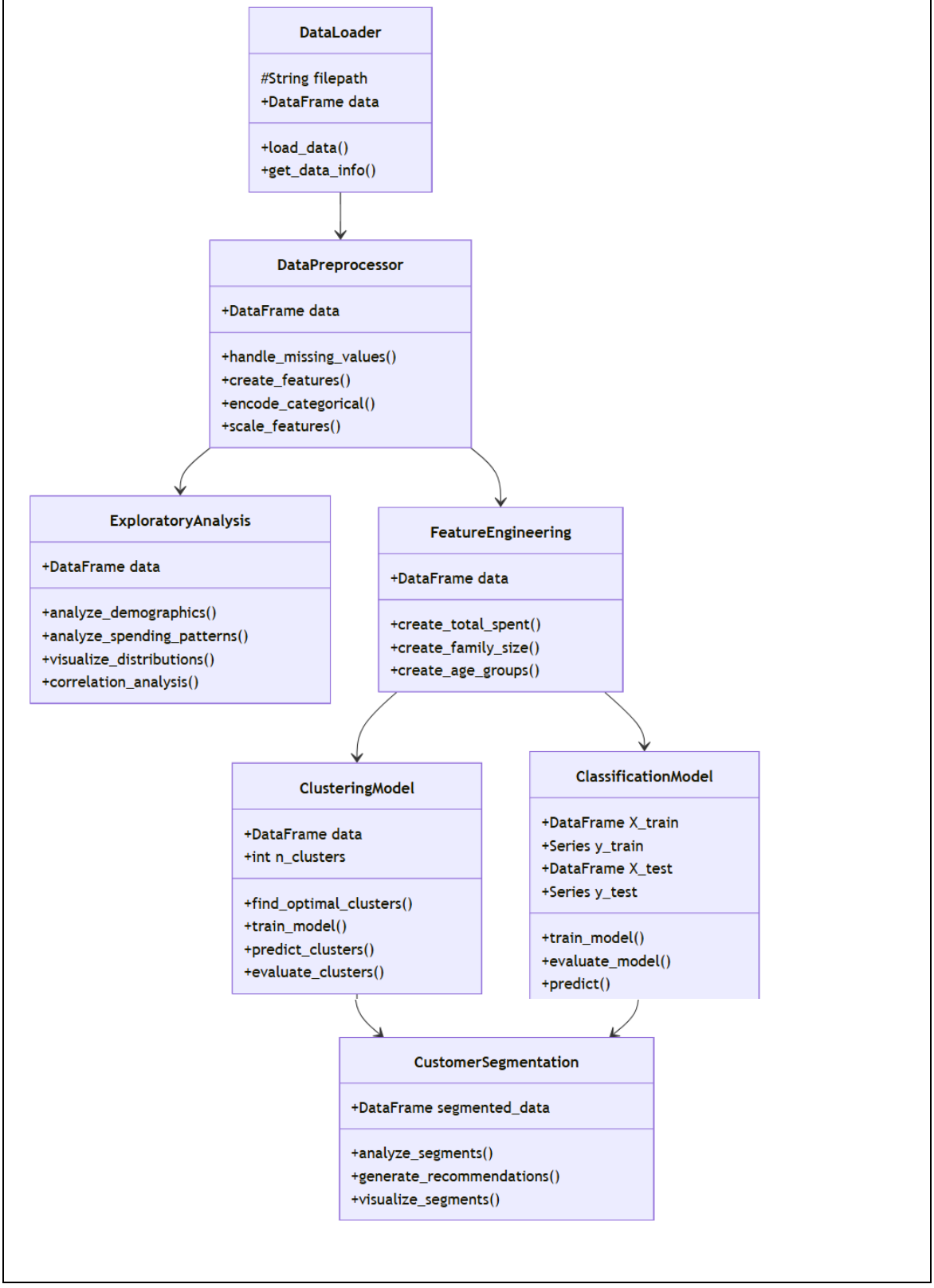


Level 1 DFD - Customer Personality Analysis System

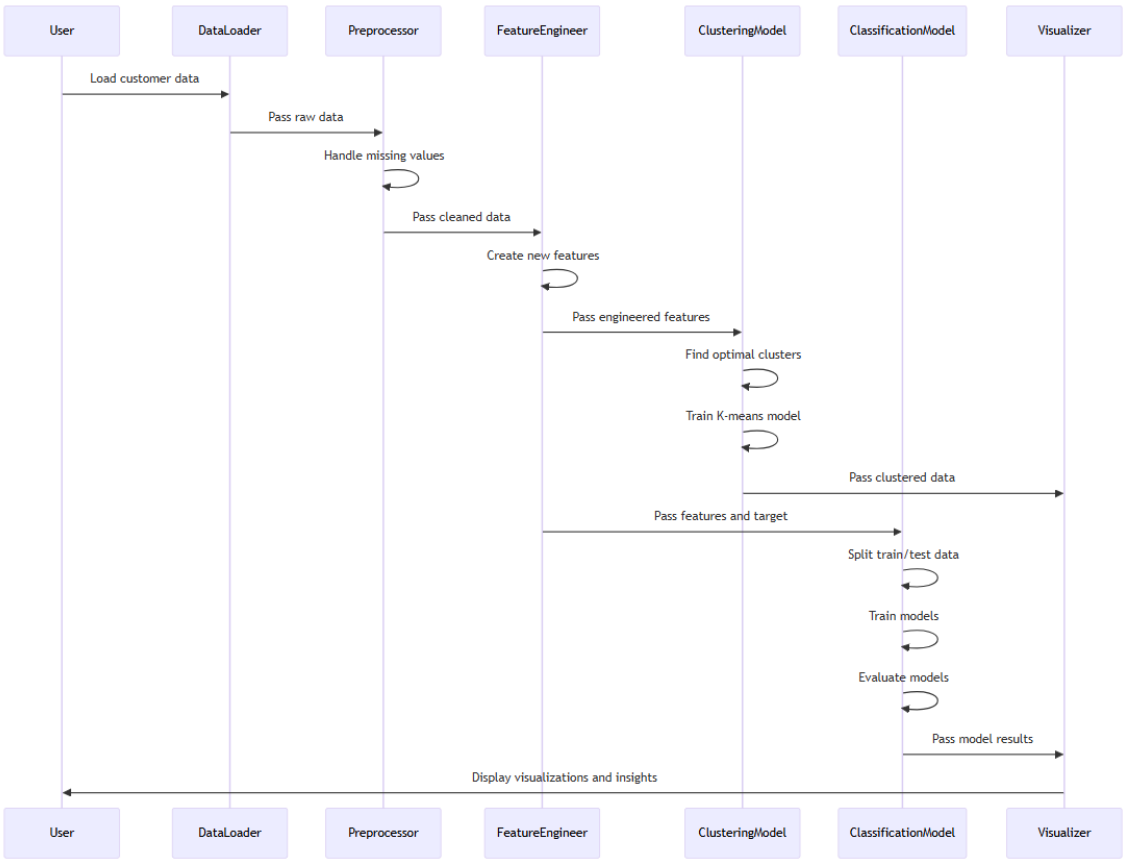


6. b) UML Diagrams

Class Diagram - Customer Personality Analysis



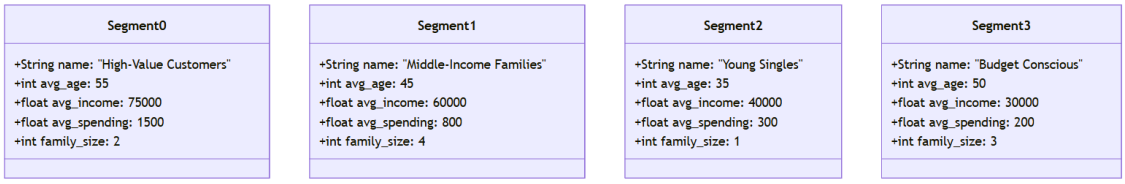
Sequence Diagram - Customer Personality Analysis Process



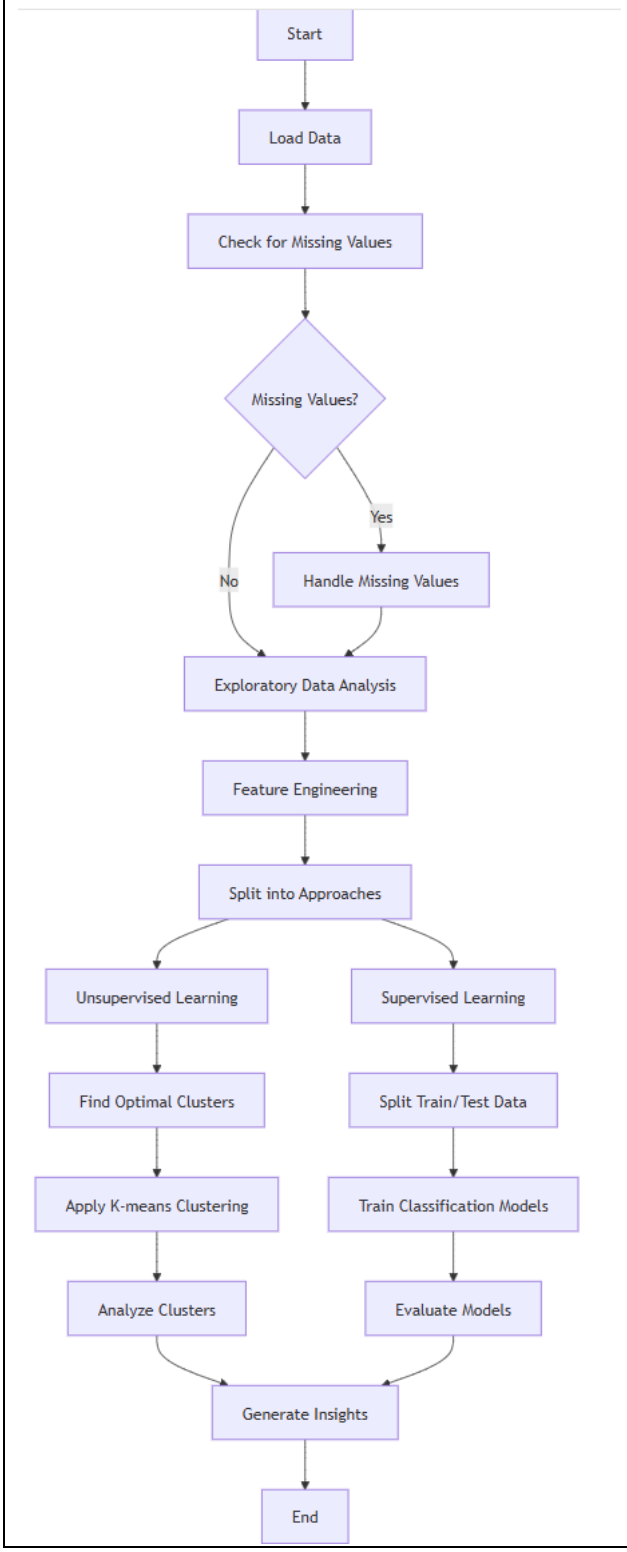
Use Case Diagram - Customer Personality Analysis

Object Diagram

Object Diagram - Customer Segments

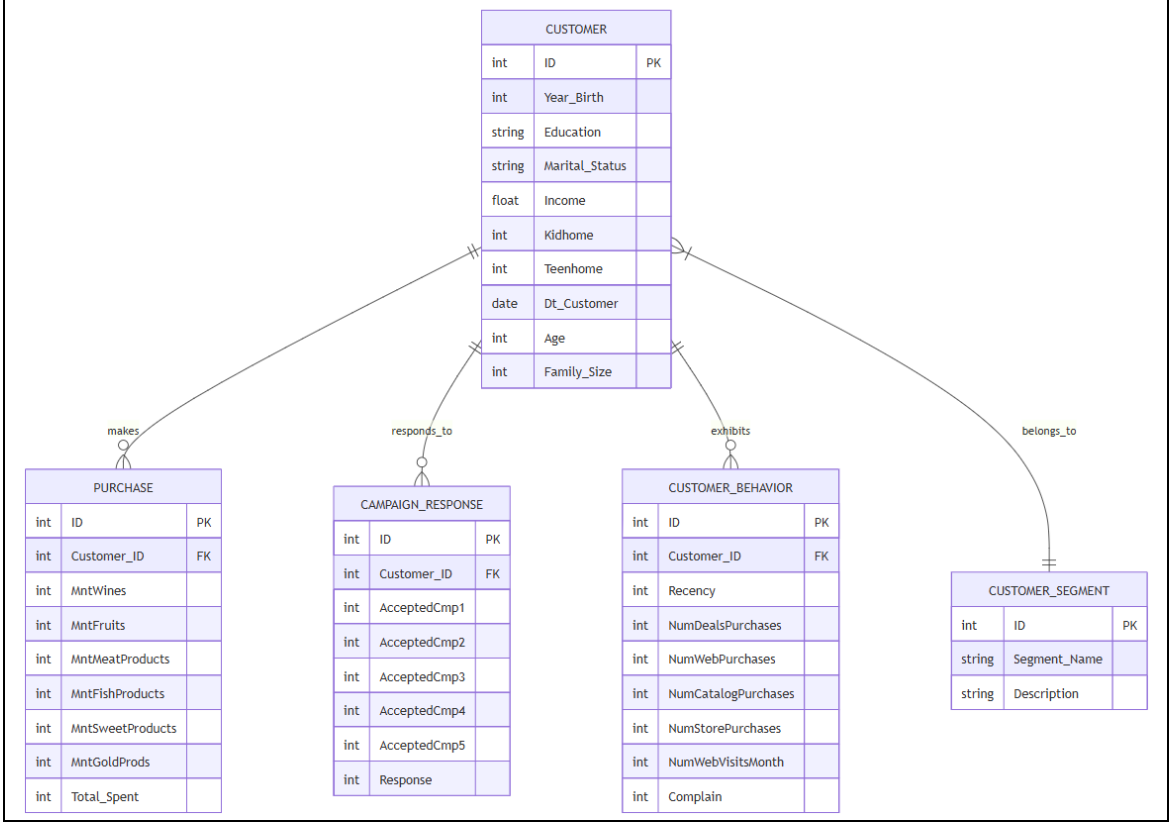


Control Flow Diagram - Customer Personality Analysis



6. c) Database Design

E-R Diagram - Customer Personality Analysis



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 implementation of this project was carried out in a modular, step-by-step manner within a Jupyter Notebook environment using Python programming language. The overall goal was to analyze customer data using both unsupervised and supervised machine learning techniques, with clear separation between data processing, model training, and evaluation phases.

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 of these stages was implemented using reusable and readable code blocks, allowing flexibility for changes and experimentation.

Software Reuse and Libraries Used

The project heavily relied on software reuse through existing open-source libraries, which significantly reduced development time and improved reliability. The following key Python libraries were reused:

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 follow the industry's best practices and are widely adopted, ensuring scalability and maintainability of the codebase.

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

The use of Jupyter Notebook provided an interactive platform to iteratively develop and debug code, visualize intermediate results, and annotate each step with Markdown documentation.

Design Patterns and Coding Techniques

Although the project is not object-oriented in nature, some core design principles and patterns were as follows:

- Modularity: Each step in the analysis process—such as loading data, cleaning, visualization, and modeling—was implemented in a separate code block. This aligns with the principle of *Separation of Concerns*.
- DRY (Don't Repeat Yourself): Common logic such as aggregations and visualizations were written in a reusable way with minimal repetition.
- Reusable Functions *(Optional for enhancement)*: The project could be extended further by converting repeated tasks (e.g., plot generation or model evaluation) into callable functions or class methods.
- Encapsulation of Data Transformations: Preprocessing steps such as label encoding, handling null values, and feature creation were encapsulated before feeding data to ML models.

Data Transformation and Preprocessing Techniques

Special coding and data preprocessing techniques included:

- Handling Missing Data: Missing Income values were replaced with the mean to preserve data without discarding rows.
- Feature Engineering:
    - Age was derived from Year_Birth.
    - Total_Spent was calculated as the total of all product category spendings.
    - Family_Size was derived using the sum of Kidhome, Teenhome, and marital relationship status.
- Encoding Categorical Variables: LabelEncoder from sklearn.preprocessing was used to convert Education levels into numerical values.

- Feature Scaling *(Optional in advanced modeling)*: The use of StandardScaler was initiated to normalize feature ranges, which is beneficial for algorithms sensitive to data magnitude.

Model Implementation Summary

- Unsupervised Learning (KMeans Clustering):
  - The Elbow Method (via Yellowbrick's KElbowVisualizer) was used to determine the optimal number of clusters (k=4).
  - KMeans then grouped customers into segments based on behavioral and demographic variables.
- Supervised Learning (Classification Models):
  - Three classifiers were implemented: Decision Tree, K-Nearest Neighbors (KNN), and Random Forest.
  - The Random Forest Classifier performed best with an accuracy of approximately 90.4%, indicating high prediction reliability for customer response.

Summary

The implementation used a structured, modular, and flexible approach that enabled:

- Efficient development using Python and industry-standard libraries
- Accurate results using tested machine learning models
- Readable, reproducible, and extensible code in a Jupyter Notebook format

This foundation allows the project to be enhanced in the future with additional models, real-time data integration, or deployment as an API or 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 imports the required libraries and loads the dataset. It then explores the basic structure of the data, including its shape, information about columns, and 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 performs exploratory data analysis on the dataset. It checks for null values, fills missing values, creates new features, and visualizes various aspects of the data such as age distribution, education distribution, marital status, income distribution, and family composition. It also analyzes correlations between variables and explores relationships 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.columı
```

Functionality: This module cleans the data by handling missing values and performs feature engineering by creating new features such as Total_Spent, Family_Size, etc. It also encodes categorical variables and prepares the data for scaling.

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 implements unsupervised learning using K-means clustering. It finds the optimal number of clusters using the Elbow method, applies K-means clustering with the optimal number of clusters, and visualizes the resulting clusters.

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 implements supervised learning using various classification algorithms. It prepares the data by defining dependent and independent variables, splits the data into training and testing sets, and trains and evaluates three different classification models: Decision Tree, K-Nearest Neighbors, and Random Forest.

Input: DataFrame with features and target variable

Output: Trained classification models and their performance metrics

Database Tables

Database Table Explanation - CustomerProfile

This table contains detailed records of customer demographic information, lifestyle indicators, purchasing behavior, and responses to marketing campaigns. Each row represents one 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 = 2022 - Year_Birth
- Total_Spent = Sum of all product spending columns
- Family_Size = Sum of Kidhome, Teenhome, and inferred relationship count
- Clusters are results from unsupervised KMeans clustering
- Response is the target variable used for classification in supervised learning

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	

8. Testing

Testing plays a crucial role in verifying the correctness, accuracy, and performance of the implemented system. For this project, which focuses on data analysis and machine learning, the testing approach ensures that:

- The data is correctly preprocessed and transformed,
- The machine learning models behave as expected,
- Predictions are accurate and aligned with the objective,
- Code functions produce valid and interpretable outputs.

Testing Approach

Given the nature of the project, the following types of testing were applied:

- **Unit Testing**: To verify that individual functions such as data cleaning, feature engineering, encoding, and model training behave as expected.
- **Data Validation Testing**: Ensuring that data loading, null value handling, and transformations preserve data integrity.
- **Functional Testing**: Testing whether the pipeline—from raw data to visualization, modeling, and prediction—executes as intended.
- **Model Evaluation Testing**: Comparing predicted outcomes with actual values to determine classification model accuracy using performance metrics like accuracy score.

Lessons Learnt from Testing

- Early testing of preprocessing steps prevents downstream issues in model training.
- Data imbalance and feature skew can affect model performance—important to validate assumptions with plots.
- Regular checking of data types and nulls is essential in an ML pipeline.
- Model testing using different algorithms helped identify Random Forest as the most robust performer.

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

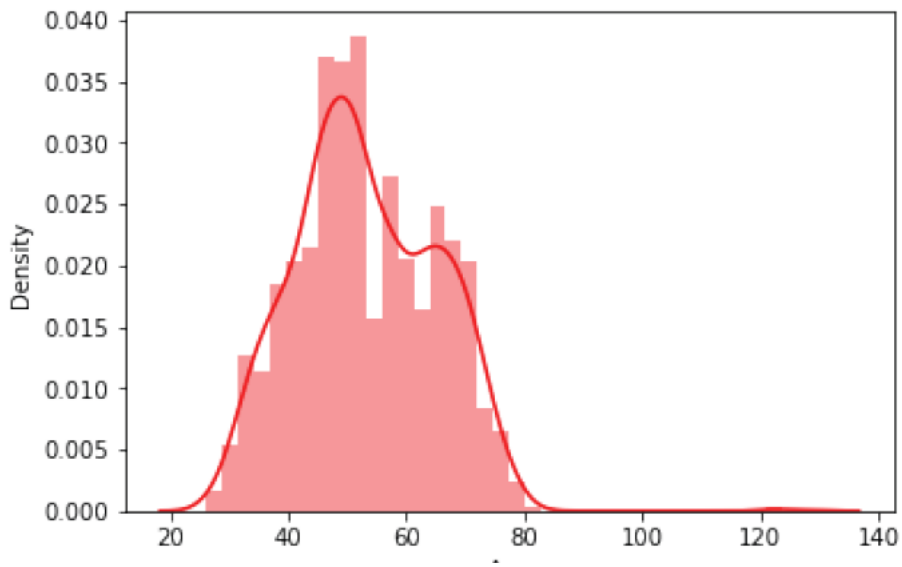


Figure 2: Pie chart showing education distribution

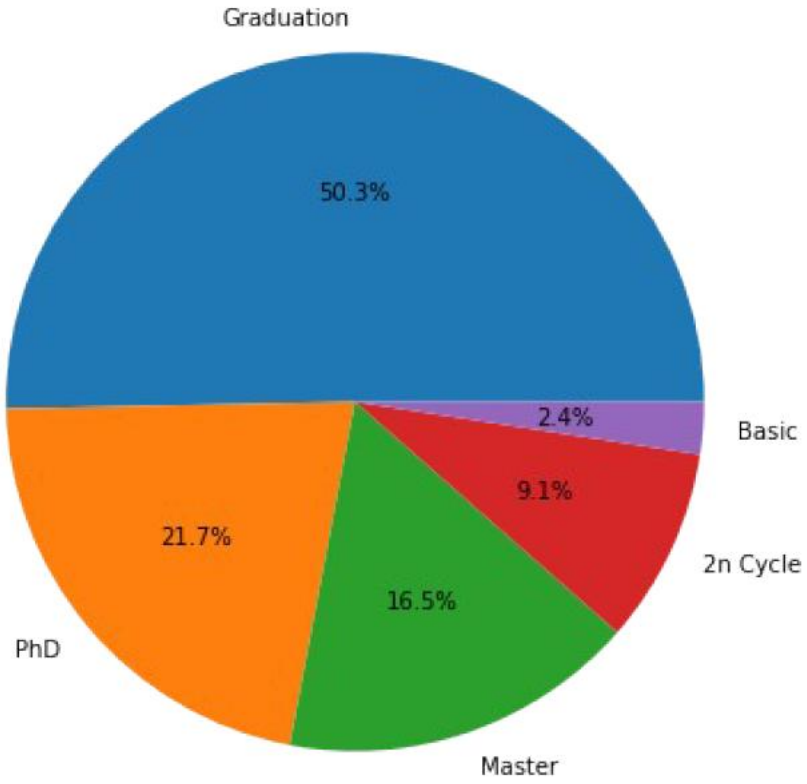


Figure 3: Count plot of marital status

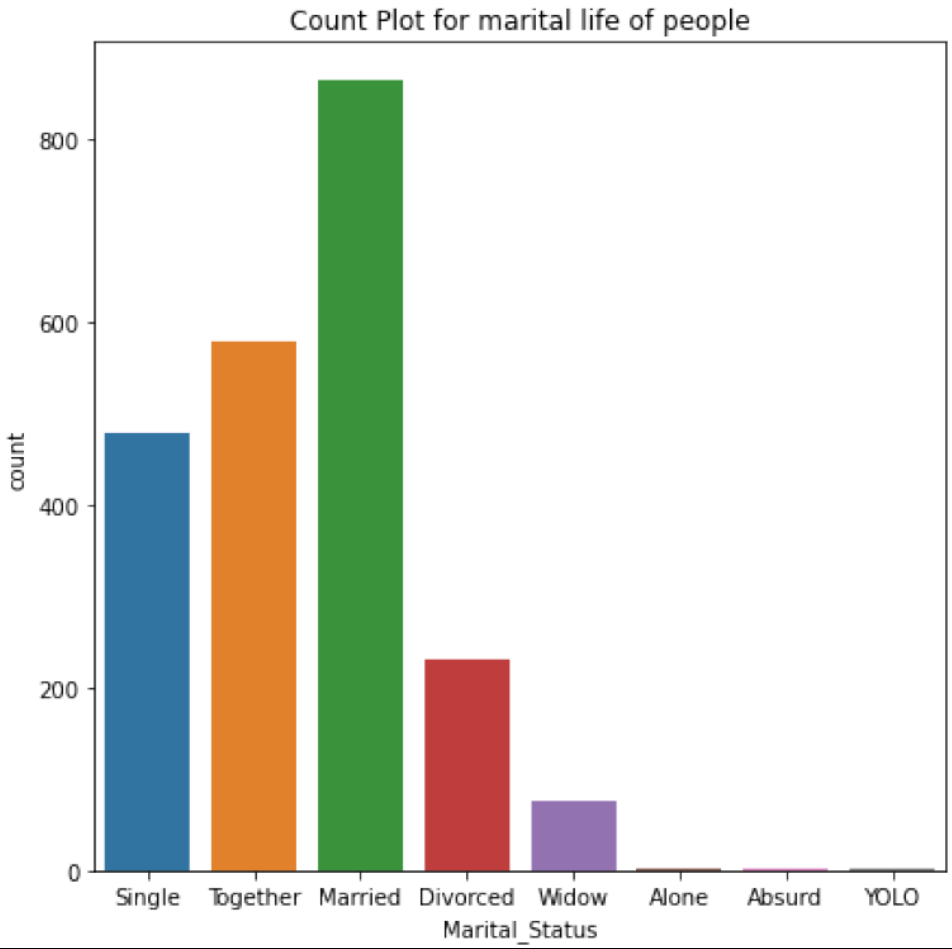


Figure 4: Distribution of customer income

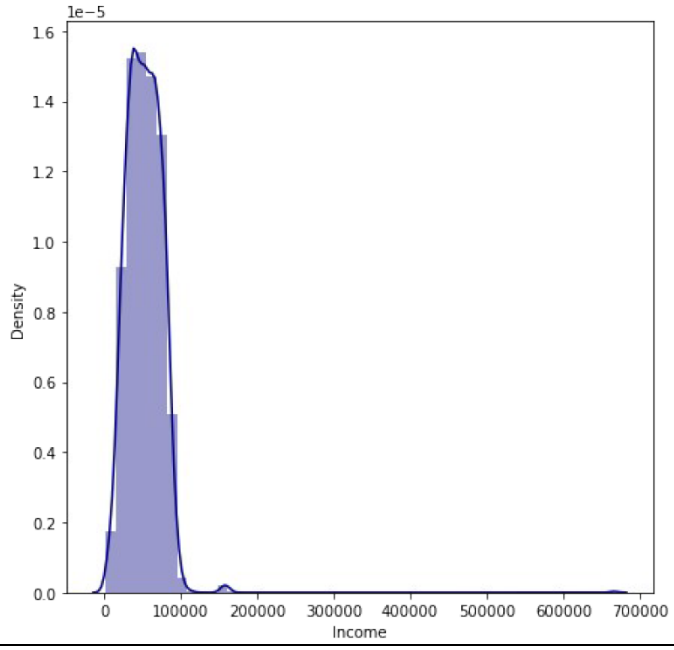


Figure 5: Correlation heatmap of numerical variables

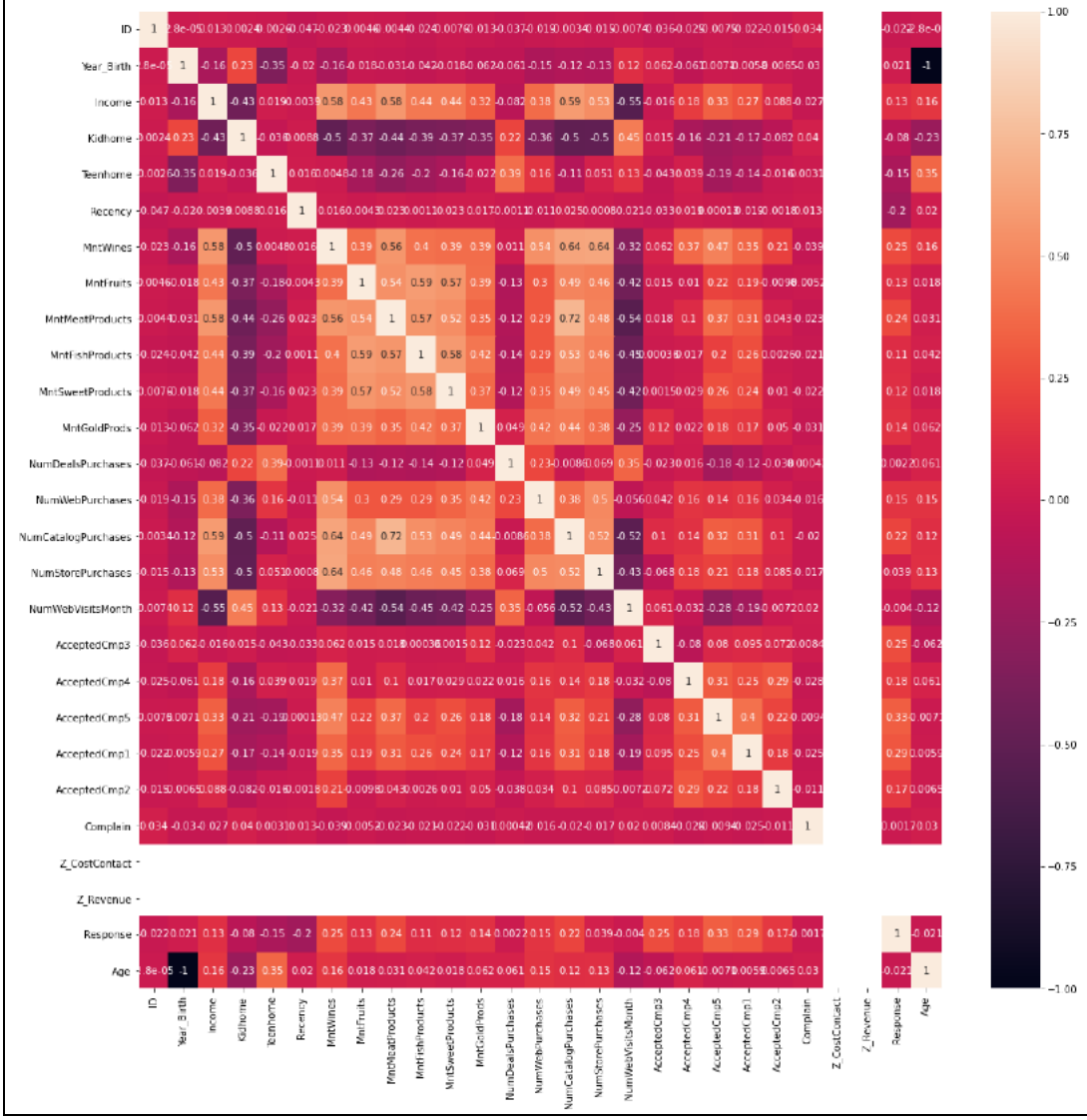


Figure 6: Scatter plots showing relationship between income and spending categories

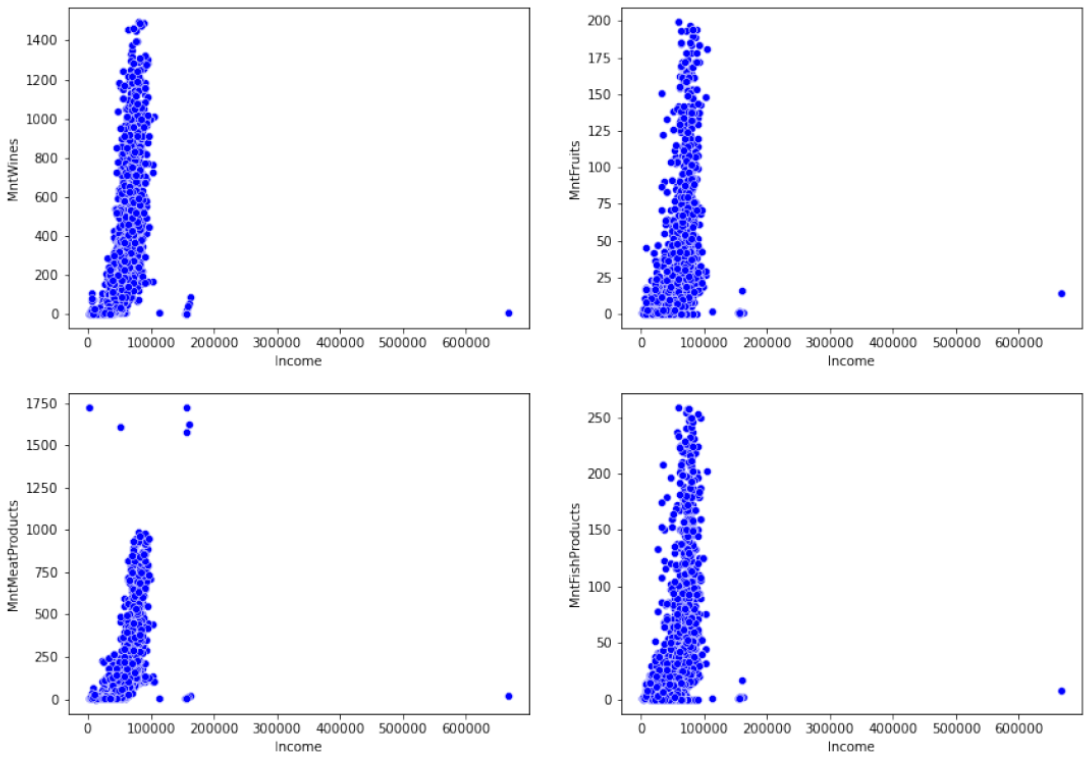


Figure 7: Elbow method for determining optimal number of clusters

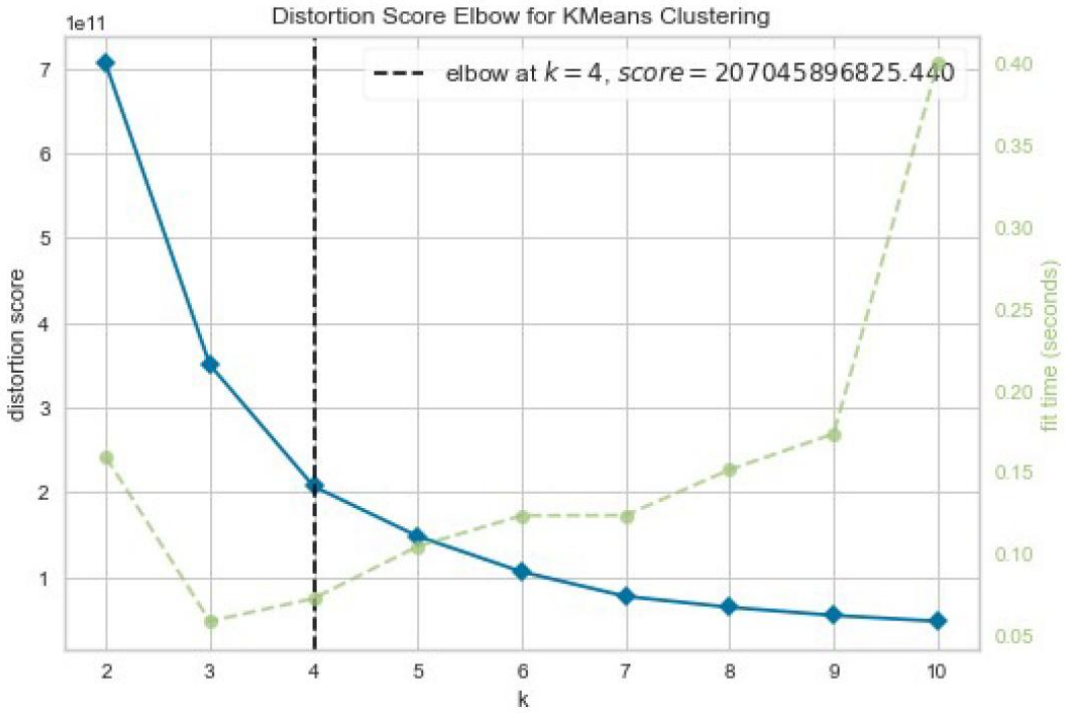


Figure 8: Distribution of customers across clusters

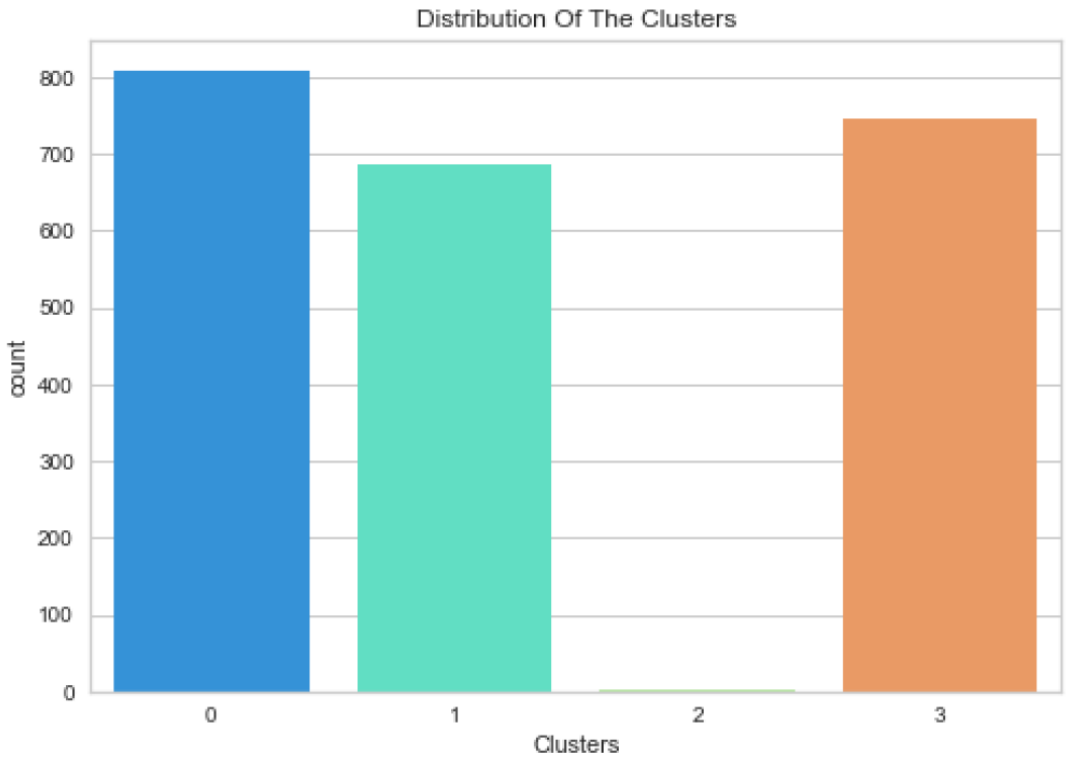
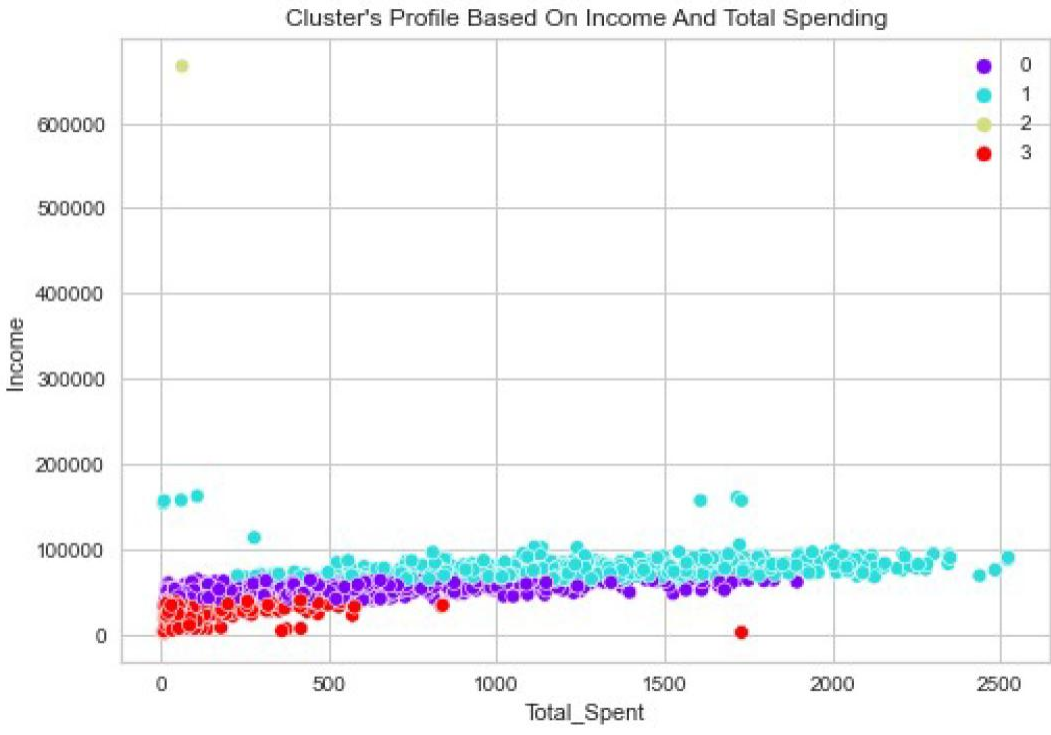


Figure 9: Scatter plot showing cluster profiles based on income and total spending



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: Primary programming language used for data analysis, preprocessing, visualization, and modeling.

Data Analysis and Manipulation:

- Pandas: Used for data manipulation and analysis, providing data structures and operations for manipulating numerical tables and time series.

- NumPy: Used for numerical computations, providing support for large, multi-dimensional arrays and matrices.

Data Visualization:

- Matplotlib: Used for creating static, animated, and interactive visualizations in Python.

- Seaborn: Built on top of Matplotlib, used for making statistical graphics more attractive and informative.

- Yellowbrick: Used specifically for machine learning visualization, particularly for the Elbow method in clustering.

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 Customer Personality Analysis project.

11. Conclusion

Summary and Key Achievements

The primary objective of this project, *Customer Personality Analysis*, was to extract meaningful insights from customer data and identify patterns that can help in effective customer segmentation and targeting. Leveraging the power of **Machine Learning**, the project focused on analyzing customer demographics, behavioral attributes, and marketing response data to understand the personality traits of customers and group them accordingly.

The project was carried out in the following structured manner:

- **Data Preprocessing and Cleaning**: Handled missing values (especially in the Income column), created derived features like Age, Total_Spent, and Family_Size, and performed label encoding.
- **Exploratory Data Analysis (EDA)**: Visualized various relationships within the data such as age distribution, income variation, family structure, and spending behavior.
- **Unsupervised Learning – Clustering**: Implemented **K-Means Clustering** to group customers into distinct clusters based on attributes like income, spending, and family profile. The **Elbow Method** was used to determine the optimal number of clusters (k = 4).
- **Supervised Learning – Classification Models**: Built predictive models using **Decision Tree**, **K-Nearest Neighbors (KNN)**, and **Random Forest** to classify customer responses to marketing campaigns. Among these, the **Random Forest** classifier achieved the highest accuracy (~90.4%), making it a strong choice for marketing response prediction.

Limitations and Lessons Learnt

Despite the success of the project, a few limitations were observed:

- The dataset was relatively limited in scope, and customer behaviors may vary over time and across regions—an aspect not covered due to static data.

- Some categorical fields, such as Marital_Status, had inconsistent values that required manual consolidation.
- No deep learning models were explored due to time and resource constraints.

Key lessons learnt include:

- The importance of **data preprocessing and feature engineering** in improving model performance.
- How combining **EDA with domain understanding** can guide meaningful feature creation.
- Practical knowledge of how different machine learning algorithms behave with real-world business data.

Further Enhancements / Recommendations

Future work in this area could consider the following enhancements:

- **Incorporate time-series elements** to study how customer behavior evolves over time.
- **Enrich the dataset** with external data sources (e.g., transaction history, website interactions, geographic data).
- Apply **Dimensionality Reduction techniques** like PCA or t-SNE for improved visualization and model performance.
- Explore advanced models such as **XGBoost**, **LightGBM**, or **Deep Neural Networks** for potentially better prediction accuracy.
- Develop an **interactive dashboard** using tools like Power BI or Tableau to make the insights accessible to marketing teams.

---

12. Appendices

This section includes supplementary information that supports the main content of the report. The materials provided here offer deeper insight into the implementation details, usage guidelines, and supporting artifacts that may otherwise interrupt the flow if placed within the main chapters.

Appendix A: User Documentation

Project Title: *Customer Personality Analysis using Machine Learning*
Objective: To analyze customer profiles and behavior using Machine Learning techniques to assist marketing teams in better segmenting and targeting their customer base.

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}")
```

Customer Personality Analysis.pdf

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

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

Thank You