CUSTOMER SEGMENTATION USING MACHINE LEARNING & HYBRID MODEL

A Minor Project Report

Submitted in Partial fulfillment for the award of

Degree of Bachelor of Technology (B.Tech.)

In

Computer Science & Engineering

(Session: 2023-24)

Submitted By

Swechchha Agrawal (Roll no-0205CS211112)

Vanshika Yadav (Roll no-0205CS211117)

Raja Shoaib (Roll no-0205CS223D07)

Under the Guidance of

Prof. Sweta Kriplani

(Computer Sc. & Engg.)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SHRI RAM INSTITUTE OF TECHNOLOGY, JABALPUR (M.P.)

RAJEEV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL (M.P.)

SHRI RAM INSTITUTE OF TECHNOLOGY
Approved by AICTE New Delhi & Govt. of M.P.
(Affiliated to R.G.P.V.V. - University of Technology of Madhya Pradesh)
ISO 9001 : 2000 Certified Institution
Near ITI, MADHOTAL, JABALPUR - 482 002 (M.P.)
Ph. No. : 0761 - 2640291, 94 Fax No. : 2640294 Mobile - 9300104815

CERTIFICATE

This is to certify that

Swechchha Agrawal (Roll no-0205CS211112)
Vanshika Yadav (Roll no-0205CS211117)
Raja Shoaib (Roll no-0205CS223D07)

students of 6th semester, Computer Sc. & Engg. S.R.I.T., Jabalpur have duly completed their final year project entitled ***"Customer Segmentation Using Machine Learning & Hybrid Model"*** *for the partial fulfillment of the requirement for the degree of Bachelor of Technology as per R.G.P.V., Bhopal.*

They have successfully implemented and tested this project, which meets all the requirements specified under my guidance.

Prof. Sweta Kriplani	*Prof. Brajesh Patel*	*Dr. Shailesh Gupta*
(Project Guide)	*(H.O.D.)*	*(Principal)*
Computer Sc. & Engg Deptt., SRIT	*Computer Sc. & Engg Deptt., SRIT*	*SRIT, Jabalpur*

SHRI RAM INSTITUTE OF TECHNOLOGY
Approved by AICTE New Delhi & Govt. of M.P.
(Affiliated to R.G.P.V.V. - University of Technology of Madhya Pradesh)
ISO 9001 : 2000 Certified Institution
Near ITI, MADHOTAL, JABALPUR - 482 002 (M.P.)
Ph. No. : 0761 - 2640291, 94 Fax No. : 2640294 Mobile - 9300104815

CERTIFICATE

This is to certify that

Swechchha Agrawal (Roll no-0205CS211112)
Vanshika Yadav (Roll no-0205CS211117)
Raja Shoaib (Roll no-0205CS223D07)

*students of 6th semester, Computer Sc. & Engg. S.R.I.T., Jabalpur have duly completed their third year project entitled "**Customer Segmentation Using Machine Learning & Hybrid Model**" for the partial fulfillment of the requirement for the degree of Bachelor of Technology as per R.G.P.V., Bhopal.*

They have successfully implemented and tested this project, which meets all the requirements.

INTERNAL EXAMINER

Date:

EXTERNAL EXAMINER

Date:

PREFACE

In the contemporary business landscape, the significance of understanding and effectively engaging with customers cannot be overstated. As markets become increasingly competitive, businesses must leverage advanced analytical techniques to uncover deep insights into customer behavior and preferences. Customer segmentation, the process of dividing a customer base into distinct groups based on specific characteristics, is a pivotal strategy in achieving this goal. Traditionally, segmentation methods have relied on either demographic data or behavioral metrics, but the advent of big data and machine learning has opened up new possibilities for more nuanced and effective segmentation strategies.

This Report, "Customer Segmentation Using Hybrid Model," presents an innovative approach to customer segmentation that integrates various data sources and analytical techniques to provide a comprehensive understanding of customer segments. By combining demographic, psychographic, behavioral, and transactional data, the hybrid model offers a multi-faceted view of customers, enabling businesses to tailor their marketing strategies more precisely and effectively.

The hybrid model discussed in this book leverages both supervised and unsupervised machine learning algorithms, ensuring that the segmentation process is both data-driven and adaptable to the unique needs of different businesses. The integration of clustering techniques, such as K-means and hierarchical clustering, with classification algorithms like decision trees and neural networks, allows for the creation of dynamic and actionable customer segments.

We extend our gratitude to the researchers, practitioners, and businesses who have contributed their knowledge and experiences, making this comprehensive guide possible. It is our hope that this book will serve as a valuable resource for those seeking to enhance their understanding of customer segmentation and its critical role in modern business strategy.

Swechchha Agrawal (Roll no-0205CS211112)
Vanshika Yadav (Roll no-0205CS211117)
Raja Shoaib (Roll no-0205CS223D07)

ACKNOWLEDGEMENT

First of all, we thank the ***GOD*** who is constantly showering his blessing and love on us.

As one self can accomplish nothing, this project is also not an exception. I would like to have some space to acknowledge some of them that frequently fade in to the background during the course of our project work; we got help, advices, suggestions and co-operation from many people. Some influenced us, some inspired us, and some helped us in completing our project work.

It is our great privilege; pride and honor in expressing our deepest sense of gratitude to my esteemed inspiring, mentor Prof. Sweta Kriplani in presenting this project is entitled "CUSTOMER SEGMENTATION USING MACHINE LEARNING & HYBRID MODEL", his constant inspiration, memorable guidance, innovative ideas, at various stages and above all her untiring valuable support has brought out this project to an exquisite workmanship and great success.

We express my profound and sincere gratitude to Mr. R. Karsoliya (Chairman) Shri Ram Institute of Technology, Jabalpur (M.P.), affiliated to Rajiv Gandhi Technical University, Bhopal, (M.P.) for providing facilities needed in connection with our project work.

We express our gratitude to Dr. Shailesh Gupta Principal, Dr. Ruchi K. Patel , Prof. Brajesh Patel, Prof. Rajendra Arakh, Prof. Deepak Singh Rajput, Prof Aproov Khare ,Prof Sweta Kriplani, Prof Richa Shukla for their support and help we received from him throughout this project work and encouragement to carry out this project.

Swechchha Agrawal (Roll no-0205CS211112)
Vanshika Yadav (Roll no-0205CS211117)
Raja Shoaib (Roll no-0205CS223D07)

SHRI RAM INSTITUTE OF TECHNOLOGY
Approved by AICTE New Delhi & Govt. of M.P.
(Affiliated to R.G.P.V.V. - University of Technology of Madhya Pradesh)
ISO 9001 : 2000 Certified Institution
Near ITI, MADHOTAL, JABALPUR - 482 002 (M.P.)
Ph. No. : 0761 - 2640291, 94 Fax No. : 2640294 Mobile - 9300104815

DECLARATION

We, hereby declare that the work presented in this document, titled "CUSTOMER SEGMENTATION USING MACHINE LEARNING & HYBRID MODEL" is the result of our original research and has been carried out under the guidance of Prof. Sweta Kriplani at Shri Ram Institute of Technology, Jabalpur (M.P.).

We affirm that this research represents an authentic and original contribution to the field of engineering and technology. Any sources, data, or literature used in this work have been duly acknowledged and referenced. The ideas, methodologies, and findings presented in this document are the outcome of our independent investigation and analysis.

We acknowledge that ethical considerations and scientific integrity have been paramount throughout the research process. All experiments, data collection, and analysis have been conducted in accordance with the ethical standards and guidelines of Shri Ram Institute of Technology, Jabalpur (M.P.).

We also confirm that this work has not been submitted in part or in full for the award of any other degree or qualification in this institution or any other academic institution. Any contributions from other individuals or organizations have been appropriately acknowledged. We take full responsibility for the content of this document and are aware of the consequences of any form of plagiarism or academic misconduct. By signing this declaration, we affirm our commitment to the principles of honesty, integrity, and academic excellence.

Swechchha Agrawal (Roll no-0205CS211112)
Vanshika Yadav (Roll no-0205CS211117)
Raja Shoaib (Roll no-0205CS223D07)

LIST OF FIGURES

INDEX

1. INTRODUCTION

Customer segmentation is a cornerstone of modern marketing strategies, aiming to divide a heterogeneous customer base into distinct groups with shared characteristics and behaviors. This segmentation facilitates targeted marketing campaigns, personalized communication, and enhanced customer satisfaction. While traditional segmentation methods have been effective to some extent, they often fail to capture the intricate nuances of customer behavior in today's data-rich environment.

Machine learning (ML) techniques present a promising avenue for refining customer segmentation by leveraging advanced algorithms to analyze vast amounts of data and extract meaningful insights. However, despite the effectiveness of ML-based segmentation, it may face challenges in certain scenarios, such as dealing with noisy or unstructured data, or when interpretability is crucial for decision-making. To address these challenges, hybrid models that combine the strengths of both ML algorithms and traditional statistical methods have emerged as a compelling approach.

This introduction provides an overview of customer segmentation using machine learning, highlights its benefits and challenges, and introduces the concept of hybrid models as a means to enhance segmentation accuracy and interpretability. Through the integration of ML techniques with traditional statistical approaches, hybrid models offer a balanced solution for customer segmentation, catering to the diverse needs of businesses across various industries.

2. ANALYSIS

Customer segmentation is an essential process for businesses aiming to understand their customer base more profoundly and tailor their marketing efforts to specific groups. Traditional segmentation methods, such as demographic or psychographic segmentation, provide valuable insights but often lack the depth and dynamism needed to fully capture the complexity of customer behavior in today's data-rich environment.

2.1 OBJECTIVE

The primary objective of customer segmentation using a hybrid model is to achieve a more accurate, comprehensive, and actionable understanding of the customer base. By integrating multiple data sources and analytical techniques, the hybrid model aims to address the limitations of traditional segmentation methods and provide deeper insights that can drive more effective marketing strategies and business decisions. The specific objectives include:

1. Enhance Customer Understanding:

- **Multidimensional Insights:** Combine demographic, psychographic, behavioral, and transactional data to gain a 360-degree view of customers.
- **Identify Hidden Patterns:** Use advanced machine learning techniques to uncover latent patterns and relationships within the customer data that traditional methods may miss.

2. Improve Marketing Effectiveness:

- **Targeted Campaigns:** Develop more precise and personalized marketing campaigns that resonate with specific customer segments.
- **Optimized Resource Allocation:** Allocate marketing resources more efficiently by focusing on high-potential customer segments.

2.2 REQUIREMENT GATHERING

Effective customer segmentation using a hybrid model requires meticulous requirement gathering to ensure that the process is aligned with business objectives and leverages the appropriate data and tools. Below is a comprehensive guide to the requirement gathering process for implementing a hybrid customer segmentation model:

1. Define Business Objectives

- **Understand the Purpose:** Clarify why the segmentation is being undertaken (e.g., targeted marketing, personalized customer experiences, new product development).
- **Set Goals:** Define specific, measurable goals such as increasing customer retention rates, improving campaign ROI, or identifying new market opportunities.

2. Data Requirements

- **Data Sources:** Identify all relevant data sources, including CRM systems, transactional databases, web analytics, social media, and third-party data providers.
- **Data Types:**
    - **Mall Customer:** CustomerID, Age, gender, Annual Income (k$), Spending Score (1-100).

- Data Quality: Assess the quality and completeness of the data available. Plan for data cleaning and preprocessing if necessary.

3. Analytical Requirements

- **Analytical Tools:** Identify the tools and software needed for data analysis (e.g., Python, Excel, machine learning platforms).
- **Techniques and Algorithms:**
    - Clustering: K-means, hierarchical clustering, DBSCAN.
    - Classification: Decision trees, random forests, neural networks.

2.3 HARDWARE REQUIREMENT

Implementing a customer segmentation model using a hybrid approach requires robust hardware to handle large datasets, perform complex computations, and ensure efficient processing. The specific hardware requirements can vary depending on the scale of the data and the complexity of the algorithms used, but the following guidelines provide a comprehensive overview:

1. Computing Power

- **CPU (Central Processing Unit):**
    - Type: Multi-core processors (preferably Intel i7/i9 or AMD Ryzen 7/9)
    - Clock Speed: High clock speed (3.5 GHz or higher) for faster data processing.
- **GPU (Graphics Processing Unit):**
    - Memory: At least 8 GB of GPU memory; 16 GB or more for handling large models and datasets.

2. Memory (RAM)

- Minimum Requirement: 32 GB of RAM for moderate data sizes.
- Recommended: 64 GB or more for large datasets and complex models to ensure smooth operation and prevent bottlenecks.

3. Storage

- Type: Solid State Drive (SSD) for faster read/write speeds.
- Capacity: At least 1 TB SSD; additional storage (HDD) may be required for archival purposes.

2.4 SOFTWARE REQUIREMENT

Implementing a hybrid model for customer segmentation requires a combination of software tools and platforms for data collection, preprocessing, analysis, visualization, and deployment. Here are the essential software requirements:

1. Operating System

- Windows 10

2. IDE Tool

- VS Code

3. Programming Language

- **Python**

4. Machine Learning Libraries and Frameworks:

- **Python**: scikit-learn, TensorFlow, Keras, PyTorch.

5. Visualization Tools:

- Python (matplotlib, seaborn, plotly) for custom visualizations.

2.5 Feasibility Study

A feasibility study for implementing a customer segmentation model using a hybrid approach involves analyzing various factors to determine the viability, benefits, and potential challenges of the project. This includes examining technical, operational, economic, and legal aspects to ensure a comprehensive understanding of the project's potential success.

1. Technical Feasibility

- **Data Availability and Quality:**
  - Data Sources: Assess the availability of necessary data from CRM systems, transactional databases, web analytics, social media, and third-party data providers.
- **Technical Expertise:**
  - Skills and Knowledge: Ensure the availability of skilled personnel with expertise in data science, machine learning, data engineering, and domain-specific knowledge.
- **Infrastructure and Tools:**
  - Hardware Requirements: Assess the current hardware infrastructure and determine if upgrades are needed (e.g., high-performance CPUs, GPUs, sufficient RAM, and storage).

2. Operational Feasibility

- **Business Objectives Alignment:**
  - Strategic Fit: Ensure the project aligns with the organization's strategic goals, such as enhancing marketing effectiveness, improving customer satisfaction, and driving business growth.
  - Stakeholder Support: Secure buy-in from key stakeholders, including marketing, sales, product development, and senior management.
- **Process Changes:**
  - Workflow Integration: Assess how the hybrid segmentation model will integrate with existing business processes and workflows.
  - Change Management: Develop a change management plan to address potential resistance and ensure smooth adoption of new practices.
- **Resource Allocation:**
  - Human Resources: Ensure the availability of dedicated personnel to manage the project, including data scientists, analysts, and IT support.
  - Time and Effort: Estimate the time and effort required for each phase of the project, including data collection, model development, validation, and deployment.

2.6 Software Model

Creating a software model for a customer segmentation project involves several steps, from data collection and preprocessing to model building, evaluation, and deployment. Here's an outline of the process and the tools commonly used at each stage:

1. Data Collection

- Sources: CRM systems, transaction logs, website analytics, social media, surveys, etc.
- Tools: SQL databases, APIs, web scraping tools.

2. Data Preprocessing

- Data Cleaning: Handling missing values, removing duplicates, correcting errors.
- Feature Engineering: Creating new features from existing data, normalizing or standardizing features.
- Tools: Pandas, NumPy, scikit-learn, PySpark.

3. Exploratory Data Analysis (EDA)

- Visualization: Understanding the distribution of data, relationships between features.
- Tools: Matplotlib, Seaborn, Plotly.

4. Model Building

- Choosing Algorithms: K-Means, Hierarchical Clustering, DBSCAN, Gaussian Mixture Models, etc.
- Model Training: Training the selected algorithm on the preprocessed data.
- Tools: scikit-learn, TensorFlow, PyTorch.

5. Model Evaluation

- Metrics: Silhouette score, Davies-Bouldin index, Elbow method for K-Means.
- Validation: Cross-validation, holdout validation.
- Tools: scikit-learn.

2.7 SOFTWARE DEVELOPMENT LIFE CYCLE

Implementing a customer segmentation model using a hybrid approach involves a comprehensive Software Development Life Cycle (SDLC) to ensure the project is well-planned, executed, and maintained. The SDLC stages include requirements gathering, design, development, testing, deployment, and maintenance.

1. Requirement Gathering and Analysis

- **Objectives:**
  - Define business goals and objectives.
  - Identify stakeholders and their needs.

    - Gather detailed requirements for data sources, processing, analysis, and visualization.
- **Activities:**
    - Conduct meetings and interviews with stakeholders.
    - Analyze existing data sources and systems.
    - Define technical and functional requirements.
    - Document business use cases and user stories.
- **Deliverables:**
    - Requirements specification document.
    - Use case diagrams and user stories.

2. System Design

- **Objectives:**
    - Design the architecture of the hybrid model.
    - Plan the integration of data sources and preprocessing steps.
    - Define the machine learning models and algorithms to be used.
- **Activities:**
    - Create high-level system architecture diagrams.
    - Design database schema and data flow diagrams.
    - Select appropriate machine learning algorithms and tools.
    - Develop data preprocessing pipelines.
- **Deliverables:**
    - System architecture diagram.
    - Database schema.
    - Data flow diagrams.
    - Algorithm selection and model design documentation.

3. Development

- **Objectives:**
    - Develop data collection, preprocessing, and integration modules.
    - Implement machine learning models for clustering and classification.
    - Create visualization dashboards and reporting tools.
- **Activities:**
    - Write code for data extraction, transformation, and loading (ETL).
    - Develop scripts for data cleaning and preprocessing.
- **Deliverables:**
    - ETL scripts and data preprocessing code.
    - Machine learning model scripts.
    - Visualization and reporting tools.

4. Testing

- **Objectives:**
    - Validate the accuracy and performance of the machine learning models.

    - Ensure the system meets functional and non-functional requirements.
    - Identify and fix any bugs or issues.
- **Activities:**
    - Perform unit testing on individual modules.
    - Conduct integration testing to ensure modules work together seamlessly.
    - Validate model performance using cross-validation and performance metrics (e.g., accuracy, precision, recall).
    - Conduct user acceptance testing (UAT) with stakeholders.
- **Deliverables:**
    - Test plans and test cases.
    - Test reports and results.
    - Bug and issue logs.

5. Deployment

- **Objectives:**
    - Ensure the system is scalable and reliable.
    - Provide training and documentation for users.
- **Activities:**
    - Set up the production environment (cloud or on-premises).
    - Create user manuals and technical documentation.
- **Deliverables:**
    - Deployed system in production.
    - Training materials and user manuals.

6. Maintenance and Monitoring

- **Objectives:**
    - Ensure the system operates smoothly and efficiently.
    - Monitor model performance and update models as needed.
    - Address any issues or bugs that arise.
- **Activities:**
    - Monitor system performance and resource usage.
    - Implement logging and alerting for system issues.
- **Deliverables:**
    - Updated models and retraining logs.
    - Support and maintenance logs.

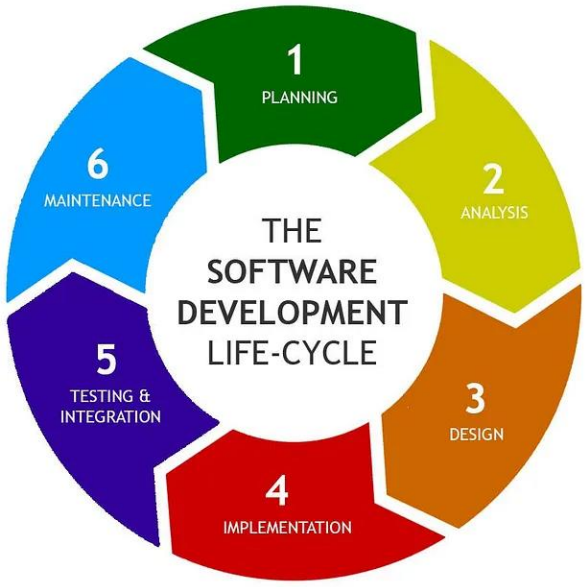

Figure 1: Software Development Life- Cycle

2.8 COST ESTIMATION

For a college minor project, the scope is smaller, and the budget is often more constrained. The focus is on learning and demonstrating the concepts rather than deploying a production-grade system. Here is a cost estimation for a college minor project on customer segmentation using a hybrid model:

Summary of Estimated Costs

Cost Category	Estimated Cost (Low)	Estimated Cost (High)
Hardware	$2,050	$3,100
Software	$0	$0
Personnel	$0	$4,000
Data Acquisition	$0	$500
Deployment and Maintenance	$0	$300
Total Estimated Cost	$2,050	$7,900

Explanation

1. **Hardware Costs:** Laptops or desktops are necessary for students to work on the project. External storage is optional but useful for backup.

2. **Software Costs:** Using open-source and free-tier software tools significantly reduces costs.

3. **Personnel Costs:** Assuming students are working as part of their coursework or for a stipend, which is minimal.

4. **Data Acquisition Costs:** Public datasets are preferred, but minimal budget is allocated for any additional data if needed.

5. **Deployment and Maintenance Costs:** Using free-tier cloud services for deployment and storage keeps costs low.

3. DESIGN

Implementing a customer segmentation system using a hybrid model requires gathering various types of input data and ensuring that these data sets are comprehensive, accurate, and relevant to the segmentation objectives. Below are the detailed input requirements categorized by data types, technical specifications, and stakeholder inputs.

3.1 INPUT REQUIREMENT

1**. Data Requirements**

- **Transactional Data:**
  - Purchase History: Detailed records of each customer's purchases, including transaction dates, products purchased, quantities, and amounts spent.
  - Transaction Frequency: Number of transactions per customer over a specific period.
- **Demographic Data:**
  - Customer Profiles: Information on age, gender, income level, education, marital status, and occupation.
  - Geographic Information: Customer locations, including addresses, cities, states, and countries.
- **Behavioral Data:**
  - Website Interactions: Data on customer activities on the website, such as page views, click-through rates, time spent on site, and browsing history.
  - Engagement Metrics: Email open rates, click-through rates, social media interactions, and customer service interactions.
- **Psychographic Data:**
  - Interests and Preferences: Data on customer interests, hobbies, preferences, and lifestyle information.

- Customer Feedback: Reviews, ratings, and feedback provided by customers through surveys or social media.

3.2 DATA FLOW DIAGRAM

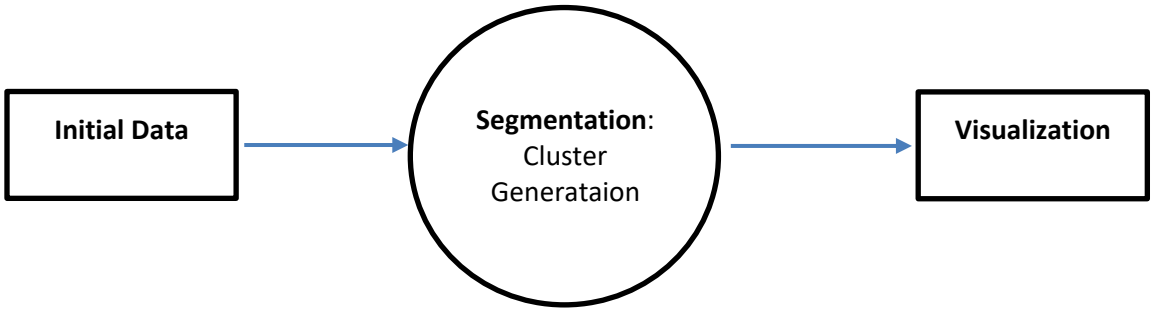

Figure 2: 0 Level Data Flow Diagram

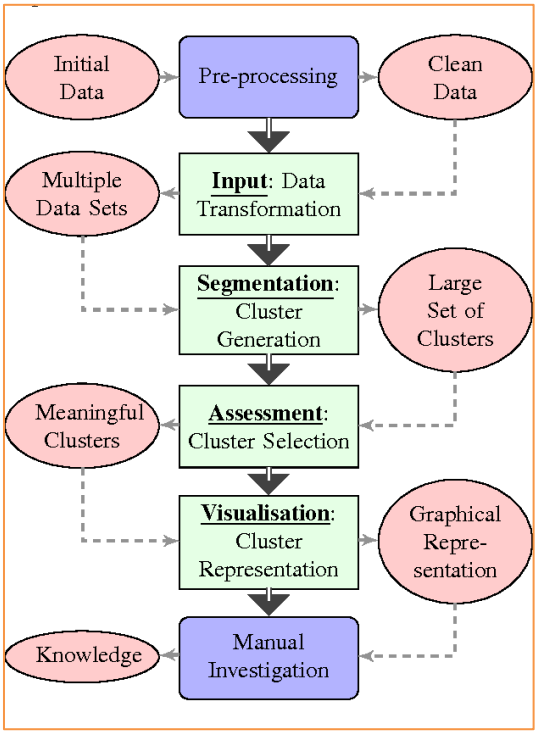

Figure 2.1: Data Flow Diagram

3.3 SYSTEM FLOW DIAGRAM

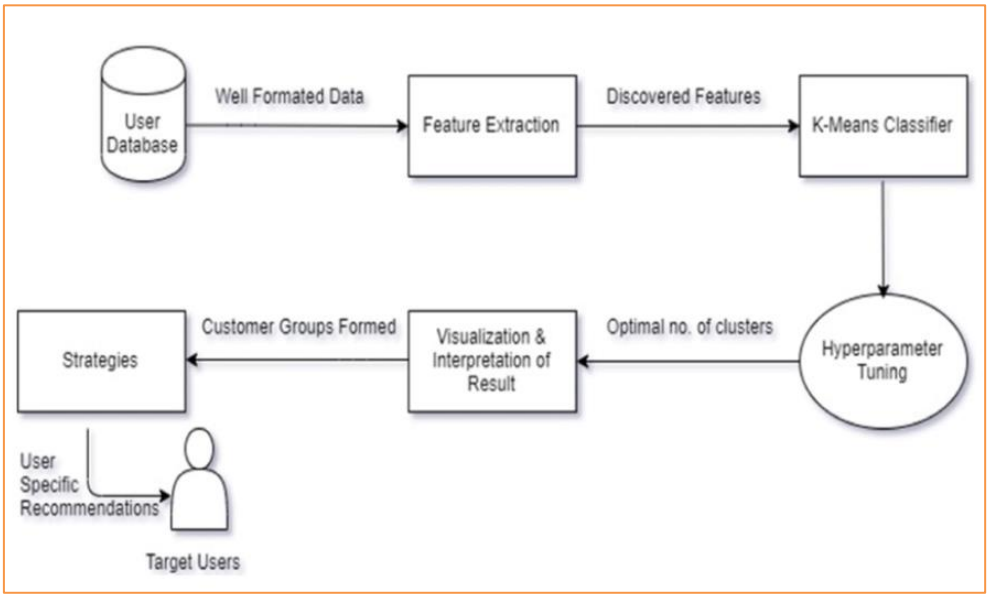

Figure 3: System Flow Diagram

3.4 USER INTERFACE DIAGRAM

Figure 4 : User Interface Diagram

4. TOOLS AND TECHNOLOGY USED

1. Data Collection and Storage

- Microsoft Excel

2. Programming Language

- Python
  - pandas: Essential for data manipulation and analysis.
  - scikit-learn: For preprocessing and feature engineering.
  - PyTorch: Deep learning platform that provides flexibility and speed.
  - Matplotlib: For Plotting the graph.
- Clustering Algorithms:
  - K-Means Clustering
  - Hierarchical Clustering
  - DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

3. Tools

- VSCode

5. CODING

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
from sklearn.cluster import KMeans, AgglomerativeClustering, DBSCAN,
AffinityPropagation
from sklearn.metrics import silhouette_score
import plotly as py
import plotly.graph_objs as go
import scipy.cluster.hierarchy as sch
from itertools import product
from sklearn.preprocessing import StandardScaler
import plotly.express as px
data = pd.read_csv('Mall_Customers.csv')
data.head()
print(data.shape)
data.describe()
data.isnull().sum()
plt.rcParams['figure.figsize'] = (15, 10)
pd.plotting.andrews_curves(data.drop("CustomerID", axis=1), "Gender")
plt.title('Andrew Curves for Gender', fontsize = 20)
plt.show()
plt.figure(1 , figsize = (15 , 6))
n = 0
for x in ['Age' , 'Annual Income (k$)' , 'Spending Score (1-100)']:
    n += 1
    plt.subplot(1 , 3 , n)
    plt.subplots_adjust(hspace = 0.5 , wspace = 0.5)
    sns.distplot(data[x] , bins = 15)
    plt.title('Distplot of {}'.format(x))
  plt.figure(figsize=(6, 4))
sns.countplot(x='Gender', data=data)
plt.title('Countplot for Gender')
plt.show()
plt.figure(figsize=(12, 8))
sns.countplot(x='Age', data=data)
plt.title('Countplot for Age')
plt.show()
data_male = data[data['Gender'] == 'Male']
data_female = data[data['Gender'] == 'Female']
plt.figure(1 , figsize = (15 , 6))
```

```
n = 0
for x in ['Age' , 'Annual Income (k$)' , 'Spending Score (1-100)']:
    n += 1
    plt.subplot(1 , 3 , n)
   plt.subplots_adjust(hspace = 0.5 , wspace = 0.5)
    sns.distplot(data_male[x] , bins = 20)
    plt.title('Distplot of male {}'.format(x))
plt.show()
plt.figure(1 , figsize = (15 , 6))
n = 0
for x in ['Age' , 'Annual Income (k$)' , 'Spending Score (1-100)']:
    n += 1
    plt.subplot(1 , 3 , n)
    plt.subplots_adjust(hspace = 0.5 , wspace = 0.5)
    sns.distplot(data_female[x] , bins = 20)
    plt.title('Distplot of Female {}'.format(x))
plt.show()
males_age = data[data['Gender']=='Male']['Age']
females_age = data[data['Gender']=='Female']['Age']
age_bins = range(15,75,5)
fig2, (ax1, ax2) = plt.subplots(1, 2, figsize=(12,5), sharey=True)
sns.distplot(males_age, bins=age_bins, kde=False, color='#0066ff', ax=ax1,
hist_kws=dict(edgecolor="k", linewidth=2))
ax1.set_xticks(age_bins)
ax1.set_ylim(top=25)
ax1.set_title('Males')
ax1.set_ylabel('Count')
ax1.text(45,23, "TOTAL count: {}".format(males_age.count()))
ax1.text(45,22, "Mean age: {:.1f}".format(males_age.mean()))
sns.distplot(females_age, bins=age_bins, kde=False, color='#cc66ff', ax=ax2,
hist_kws=dict(edgecolor="k", linewidth=2))
ax2.set_xticks(age_bins)
ax2.set_ylim(top=25)
ax2.set_title('Females')
ax2.set_ylabel('Count')
ax2.text(45,23, "TOTAL count: {}".format(females_age.count()))
ax2.text(45,22, "Mean age: {:.1f}".format(females_age.mean()))
plt.show()
male_spendings = data[data['Gender']=='Male']['Spending Score (1-100)'
female_spendings = data[data['Gender']=='Female']['Spending Score (1-100)']
spending_bin = range(0,105,5)
fig2, (ax1, ax2) = plt.subplots(1, 2, figsize=(18,5), sharey=True)
sns.distplot(male_spendings, bins=spending_bin, kde=False, color='#0066ff', ax=ax1,
hist_kws=dict(edgecolor="k", linewidth=2))
ax1.set_xticks(spending_bin)
```

```
ax1.set_ylim(top=25)
ax1.set_title('Males')
ax1.set_ylabel('Count')
ax1.text(50,15, "Mean spending score: {:.1f}".format(male_spendings.mean()))
ax1.text(50,14, "Median spending score: {:.1f}".format(male_spendings.median()))
ax1.text(50,13, "Std. deviation score: {:.1f}".format(male_spendings.std()))
ns.distplot(female_spendings, bins=spending_bin, kde=False, color='#cc66ff',
ax=ax2, hist_kws=dict(edgecolor="k", linewidth=2))
ax2.set_xticks(spending_bin)
ax2.set_ylim(top=25)
ax2.set_title('Females')
ax2.set_ylabel('Count')
ax2.text(50,15, "Mean spending score: {:.1f}".format(female_spendings.mean()))
ax2.text(50,14, "Median spending score: {:.1f}".format(female_spendings.median()))
ax2.text(50,13, "Std. deviation score: {:.1f}".format(female_spendings.std()))
medians_by_age_group = data.groupby(["Gender",pd.cut(data['Age'],
age_bins)]).median()
medians_by_age_group.index = medians_by_age_group.index.set_names(['Gender',
'Age_group'])
medians_by_age_group.reset_index(inplace=True)
medians_by_age_group.head(10)
fig, ax = plt.subplots(figsize=(12,5))
sns.barplot(x='Age_group', y='Annual Income (k$)', hue='Gender',
data=medians_by_age_group,
        palette=['#cc66ff','#0066ff'],
        alpha=0.7,edgecolor='k',
        ax=ax)
ax.set_title('Median annual income of male and female customers')
ax.set_xlabel('Age group')
plt.show()
fig, ax = plt.subplots(figsize=(12,5))
sns.barplot(x='Age_group', y='Spending Score (1-100)', hue='Gender',
data=medians_by_age_group,
        palette=['#cc66ff','#0066ff'],
        alpha=0.7,edgecolor='k',
        ax=ax)
ax.set_title('Median spending power of male and female customers')
ax.set_xlabel('Age group')
plt.show()
from scipy.stats import pearsonr
import seaborn as sns
import matplotlib.pyplot as plt
corr, _ = pearsonr(data['Age'], data['Spending Score (1-100)'])
jp = sns.jointplot(x='Age', y='Spending Score (1-100)', data=data, kind='reg')
jp.plot_joint(sns.kdeplot, zorder=0, n_levels=6)
```

```
plt.text(5, 120, 'Pearson: {:.2f}'.format(corr))
plt.show()
corr1, _ = pearsonr(males_age.values, male_spendings.values)
corr2, _ = pearsonr(females_age.values, female_spendings.values)
sns.lmplot(x='Age', y='Spending Score (1-100)', data=data, hue='Gender', aspect=1.5)
plt.text(15, 87, 'Pearson (Male): {:.2f}'.format(corr1), color='blue')
plt.text(65, 80, 'Pearson (Female): {:.2f}'.format(corr2), color='orange')
plt.show()
X1 = data[['Age' , 'Spending Score (1-100)']].iloc[: , :].values
inertia = []
s_scores = []
for n in range(2 , 11):
    algorithm = (KMeans(n_clusters = n ,init='k-means++', n_init = 10 ,max_iter=300,
                        tol=0.0001,  random_state= 111  , algorithm='elkan') )
    algorithm.fit(X1)
    inertia.append(algorithm.inertia_)
    silhouette_avg = silhouette_score(X1, algorithm.labels_)
    s_scores.append(silhouette_avg)
plt.figure(1 , figsize = (15 ,6))
plt.plot(np.arange(2 , 11) , inertia , 'o')
plt.plot(np.arange(2 , 11) , inertia , '-' , alpha = 0.5)
plt.xlabel('Number of Clusters') , plt.ylabel('Inertia')
plt.show()
fig, ax = plt.subplots(figsize=(12,5))
sns.lineplot(x=np.arange(2, 11), y=s_scores, marker='o', ax=ax)
ax.set_title("Silhouette score method")
ax.set_xlabel("Number of clusters")
ax.set_ylabel("Silhouette score")
ax.axvline(4, ls="--", c="red")
ax.axvline(5, ls="--", c="red")
ax.axvline(6, ls="--", c="red")
plt.grid()
plt.show()
algorithm = (KMeans(n_clusters = 4 ,init='k-means++', n_init = 10 ,max_iter=300,
                        tol=0.0001,  random_state= 111  , algorithm='elkan') )
algorithm.fit(X1)
labels1 = algorithm.labels_
centroids1 = algorithm.cluster_centers_
h = 0.02
x_min, x_max = X1[:, 0].min() - 1, X1[:, 0].max() + 1
y_min, y_max = X1[:, 1].min() - 1, X1[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))
Z = algorithm.predict(np.c_[xx.ravel(), yy.ravel()])
plt.figure(1 , figsize = (15 , 7) )
plt.clf()
```

```
Z = Z.reshape(xx.shape)
plt.imshow(Z , interpolation='nearest',
           extent=(xx.min(), xx.max(), yy.min(), yy.max()),
           cmap = plt.cm.Pastel2, aspect = 'auto', origin='lower')
plt.scatter( x = 'Age' ,y = 'Spending Score (1-100)' , data = data , c = labels1 ,
            s = 200 )
plt.scatter(x = centroids1[: , 0] , y =  centroids1[: , 1] , s = 300 , c = 'red' , alpha = 0.5)
plt.ylabel('Spending Score (1-100)') , plt.xlabel('Age')
plt.show()
X = data[['Age' ,'Annual Income (k$)' ,'Spending Score (1-100)']]
inertia = []
s_scores = []
for n in range(2 , 11):
    algorithm = KMeans(n_clusters = n ,init='k-means++', n_init = 10 ,max_iter=300,
                       tol=0.0001,  random_state= 111  , algorithm='elkan').fit(X)
    inertia.append(algorithm.inertia_)
    silhouette_avg = silhouette_score(X, algorithm.labels_)
    s_scores.append(silhouette_avg)
fig, ax = plt.subplots(figsize=(12,5))
sns.lineplot(x=np.arange(2, 11), y=inertia, marker='o', ax=ax)
ax.set_title("Elbow method")
ax.set_xlabel("Number of clusters")
ax.set_ylabel("Clusters inertia")
ax.axvline(5, ls="--", c="red")
ax.axvline(6, ls="--", c="red")
plt.grid()
plt.show()

fig, ax = plt.subplots(figsize=(12,5))
sns.lineplot(x=np.arange(2, 11), y=s_scores, marker='o', ax=ax)
ax.set_title("Silhouette score method")
ax.set_xlabel("Number of clusters")
ax.set_ylabel("Silhouette score")
ax.axvline(6, ls="--", c="red")
plt.grid()
plt.show()
KM6 = (KMeans(n_clusters = 6 ,init='k-means++', n_init = 10 ,max_iter=300,
                       tol=0.0001,  random_state= 111  , algorithm='elkan') )
KM6.fit(X)
labels6 = KM6.labels_
centroids6 = KM6.cluster_centers_
KM6_df = data.copy()
KM6_df['labels'] = labels6
fig1, axes = plt.subplots(1, 2, figsize=(12, 5))
```

```
scat_1 = sns.scatterplot(x='Annual Income (k$)', y='Spending Score (1-100)',
data=KM6_df,
                hue='labels', ax=axes[0], palette='Set1', legend='full')
sns.scatterplot(x='Age', y='Spending Score (1-100)', data=KM6_df,
                hue='labels', palette='Set1', ax=axes[1], legend='full')
axes[0].scatter(centroids6[:,1], centroids6[:,2], marker='s', s=40, c="blue")
axes[1].scatter(centroids6[:,0], centroids6[:,2], marker='s', s=40, c="blue")
plt.show()
KM_clust_sizes = KM6_df.groupby('labels').size().to_frame()
KM_clust_sizes.columns = ["KM_size"]
KM_clust_sizes
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure(figsize=(7, 7))
ax = Axes3D(fig, rect=[0, 0, .99, 1], elev=20, azim=210)
ax.scatter(KM6_df['Age'],
           KM6_df['Annual Income (k$)'],
           KM6_df['Spending Score (1-100)'],
           c=KM6_df['labels'],
           s=35, edgecolor='k', cmap=plt.cm.Set1)
ax.set_xticklabels([])
ax.set_yticklabels([])
ax.set_zticklabels([])
ax.set_xlabel('Age')
ax.set_ylabel('Annual Income (k$)')
ax.set_zlabel('Spending Score (1-100)')
ax.set_title('3D view of K-Means 5 clusters')
ax.dist = 12
plt.show()
import plotly as py
import plotly.graph_objs as go
def tracer(db, n, name):
    '''
    This function returns trace object for Plotly
    '''
    return go.Scatter3d(
        x = db[db['labels']==n]['Age'],
        y = db[db['labels']==n]['Spending Score (1-100)'],
        z = db[db['labels']==n]['Annual Income (k$)'],
        mode = 'markers',
        name = name,
        marker = dict(
            color = KM6_df['labels'],
            size = 5,
            line=dict(
            color= KM6_df['labels'],
```

```
            width= 12
        ),
        opacity=0.8
        )
    )
trace0 = tracer(KM6_df, 0, 'Cluster 0')
trace1 = tracer(KM6_df, 1, 'Cluster 1')
trace2 = tracer(KM6_df, 2, 'Cluster 2')
trace3 = tracer(KM6_df, 3, 'Cluster 3')
trace4 = tracer(KM6_df, 4, 'Cluster 4')
trace5 = tracer(KM6_df, 5, 'Cluster 5')
trace_data = [trace0, trace1, trace2, trace3, trace4, trace5]
layout = go.Layout(
   title = 'Clusters with k=6 wrt Age, Income and Spending Scores',
   scene = dict(
           xaxis = dict(title = 'Age'),
           yaxis = dict(title = 'Spending Score'),
           zaxis = dict(title = 'Annual Income')
       )
)
fig = go.Figure(data=trace_data, layout=layout)
py.offline.plot(fig, filename='3d_plot.html')
KM5 = (KMeans(n_clusters = 5 ,init='k-means++', n_init = 10 ,max_iter=300,
                    tol=0.0001,  random_state= 111  , algorithm='elkan') )
KM5.fit(X)
labels5 = KM5.labels_
centroids5 = KM5.cluster_centers_
KM5_df = data.copy()
KM5_df['labels'] = labels5
fig1, axes = plt.subplots(1, 2, figsize=(12, 5))
scat_1 = sns.scatterplot(x='Annual Income (k$)', y='Spending Score (1-100)',
data=KM5_df,
                     hue='labels', ax=axes[0], palette='Set1', legend='full')
scat_2 = sns.scatterplot(x='Age', y='Spending Score (1-100)', data=KM5_df,
                     hue='labels', ax=axes[1], palette='Set1', legend='full')
axes[0].scatter(centroids5[:, 1], centroids5[:, 2], marker='s', s=40, c="blue")
axes[1].scatter(centroids5[:, 0], centroids5[:, 2], marker='s', s=40, c="blue")
plt.show()
KM_clust_sizes5 = KM5_df.groupby('labels').size().to_frame()
KM_clust_sizes5.columns = ["KM_size"]
KM_clust_sizes5
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure(figsize=(7, 7))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(KM5_df['Age'],
```

```
        KM5_df['Annual Income (k$)'],
        KM5_df['Spending Score (1-100)'],
        c=KM5_df['labels'],
        s=35, edgecolor='k', cmap=plt.cm.Set1)
ax.set_xticklabels([])
ax.set_yticklabels([])
ax.set_zticklabels([])
ax.set_xlabel('Age')
ax.set_ylabel('Annual Income (k$)')
ax.set_zlabel('Spending Score (1-100)')
ax.set_title('3D view of K-Means 5 clusters')
ax.dist = 12
plt.show()
import plotly.offline as py_offline
fig = go.Figure(data=trace_data, layout=layout)
py_offline.plot(fig, filename='plot.html')
dendrogram = sch.dendrogram(sch.linkage(X, method = 'ward'))
plt.title('Dendrogram', fontsize = 20)
plt.xlabel('Customers')
plt.ylabel('Ecuclidean Distance')
plt.show()
from sklearn.cluster import AgglomerativeClustering
hc = AgglomerativeClustering(n_clusters=5, linkage='ward')
hc.fit(X)
labels_hc = hc.labels_
hc_df = data.copy()
hc_df['labels'] = labels_hc
fig1, axes = plt.subplots(1, 2, figsize=(12, 5))
sns.scatterplot(x='Annual Income (k$)', y='Spending Score (1-100)', data=hc_df,
            hue='labels', palette='Set1', ax=axes[0], legend='full')
sns.scatterplot(x='Age', y='Spending Score (1-100)', data=hc_df,
            hue='labels', palette='Set1', ax=axes[1], legend='full')
plt.show()
eps_values = np.arange(8,12.75,0.25) # eps values to be investigated
min_samples = np.arange(3,10) # min_samples values to be investigated
DBSCAN_params = list(product(eps_values, min_samples))
no_of_clusters = []
sil_score = []
for p in DBSCAN_params:
   DBS_clustering = DBSCAN(eps=p[0], min_samples=p[1]).fit(X)
   no_of_clusters.append(len(np.unique(DBS_clustering.labels_)))
   sil_score.append(silhouette_score(X, DBS_clustering.labels_))
tmp = pd.DataFrame.from_records(DBSCAN_params, columns =['Eps',
'Min_samples'])
tmp['No_of_clusters'] = no_of_clusters
```

```
pivot_1 = pd.pivot_table(tmp, values='No_of_clusters', index='Min_samples',
columns='Eps')
fig, ax = plt.subplots(figsize=(12,6))
sns.heatmap(pivot_1, annot=True,annot_kws={"size": 16}, cmap="YlGnBu", ax=ax)
ax.set_title('Number of clusters')
plt.show()
tmp = pd.DataFrame.from_records(DBSCAN_params, columns =['Eps',
'Min_samples'])
tmp['Sil_score'] = sil_score
pivot_1 = pd.pivot_table(tmp, values='Sil_score', index='Min_samples',
columns='Eps')

fig, ax = plt.subplots(figsize=(18,6))
sns.heatmap(pivot_1, annot=True, annot_kws={"size": 10}, cmap="YlGnBu", ax=ax)
plt.show()

DBS_clustering = DBSCAN(eps=12.5, min_samples=4).fit(X)

DBSCAN_clustered = X.copy()
DBSCAN_clustered.loc[:,'Cluster'] = DBS_clustering.labels_ # append labels to
points
DBSCAN_clust_sizes = DBSCAN_clustered.groupby('Cluster').size().to_frame()
DBSCAN_clust_sizes.columns = ["DBSCAN_size"]
DBSCAN_clust_sizes
outliers = DBSCAN_clustered[DBSCAN_clustered['Cluster'] == -1]

fig2, axes = plt.subplots(1, 2, figsize=(12, 5))
sns.scatterplot(x='Annual Income (k$)', y='Spending Score (1-100)',
        data=DBSCAN_clustered[DBSCAN_clustered['Cluster'] != -1],
        hue='Cluster', ax=axes[0], palette='Set1', legend='full', s=45)
sns.scatterplot(x='Age', y='Spending Score (1-100)',
        data=DBSCAN_clustered[DBSCAN_clustered['Cluster'] != -1],
        hue='Cluster', palette='Set1', ax=axes[1], legend='full', s=45)
axes[0].scatter(outliers['Annual Income (k$)'], outliers['Spending Score (1-100)'],
        s=5, label='outliers', c="k")
axes[1].scatter(outliers['Age'], outliers['Spending Score (1-100)'],
        s=5, label='outliers', c="k")
axes[0].legend()
axes[1].legend()
plt.setp(axes[0].get_legend().get_texts(), fontsize='10')
plt.setp(axes[1].get_legend().get_texts(), fontsize='10')
plt.show()
no_of_clusters = []
preferences = range(-20000,-500,200)
af_sil_score = []
```

```
for p in preferences:
    AF = AffinityPropagation(preference=p, max_iter=200).fit(X)
    no_of_clusters.append((len(np.unique(AF.labels_))))
    af_sil_score.append(silhouette_score(X, AF.labels_))

af_results = pd.DataFrame([preferences, no_of_clusters, af_sil_score],
index=['preference','clusters', 'sil_score']).T
af_results.sort_values(by='sil_score', ascending=False).head()

fig, ax = plt.subplots(figsize=(12,5))
sns.lineplot(x=preferences, y=af_sil_score, marker='o', ax=ax)
ax.set_title("Silhouette score method")
ax.set_xlabel("number of clusters")
ax.set_ylabel("Silhouette score")
ax.axvline(-11800, ls="--", c="red")
plt.grid()
plt.show()
AF = AffinityPropagation(preference=-11800).fit(X)
AF_clustered = X.copy()
AF_clustered.loc[:,'Cluster'] = AF.labels_ # append labels to points
AF_clust_sizes = AF_clustered.groupby('Cluster').size().to_frame()
AF_clust_sizes.columns = ["AF_size"]
AF_clust_sizes

fig3, (ax_af) = plt.subplots(1, 2, figsize=(12,5))
scat_1 = sns.scatterplot(x='Annual Income (k$)', y='Spending Score (1-100)',
data=AF_clustered,
            hue='Cluster', ax=ax_af[0], palette='Set1', legend='full')
sns.scatterplot(x='Age', y='Spending Score (1-100)', data=AF_clustered,
            hue='Cluster', palette='Set1', ax=ax_af[1], legend='full')
plt.setp(ax_af[0].get_legend().get_texts(), fontsize='10')
plt.setp(ax_af[1].get_legend().get_texts(), fontsize='10')
plt.show()
clusters = pd.concat([KM_clust_sizes,KM_clust_sizes5, DBSCAN_clust_sizes,
AF_clust_sizes],axis=1, sort=False)
clusters
```

6. TESTING

Testing a customer segmentation model involves several key steps to ensure its accuracy, robustness, and generalizability. Here are the details:

1. Train-Test Split
Divide the dataset into training and testing sets to evaluate the model on unseen data.

2. Model Validation
Validate the model using techniques like cross-validation to assess its performance on different subsets of the data.

3. Cluster Analysis
Analyze the clusters to ensure they are meaningful and actionable.

- Centroid Analysis: Examine the centroids of each cluster to understand the central tendencies.
- Cluster Distribution: Check the number of data points in each cluster.

7. Integration Testing
Test the integration of the model within the broader system to ensure it works well with other components.

- API Testing: Use tools like Postman to test the API endpoints.
- End-to-End Testing: Simulate real-world scenarios to ensure the entire system works as expected.

8. Performance Testing
Ensure the model can handle the expected load and performs efficiently.

- Latency Measurement: Measure the time taken for predictions.
- Scalability Testing: Test the system's performance with large datasets.

7. OUTPUT

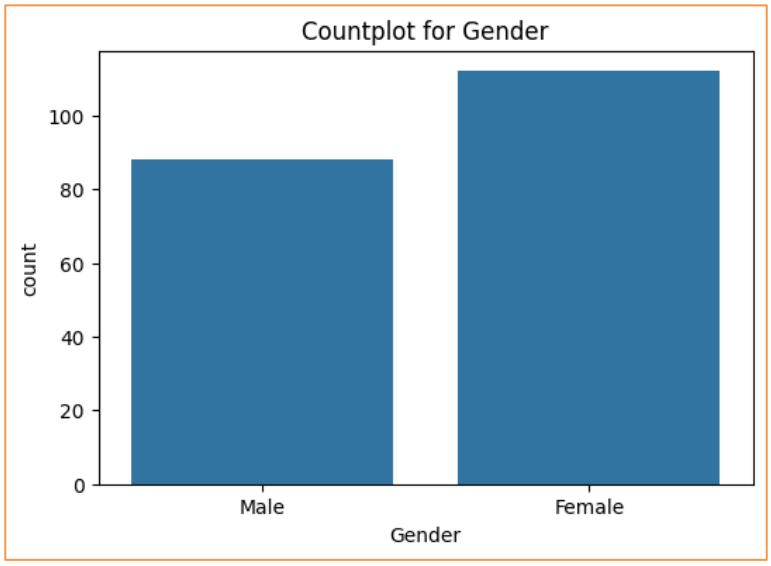

Figure 5 : Distribution of Gender

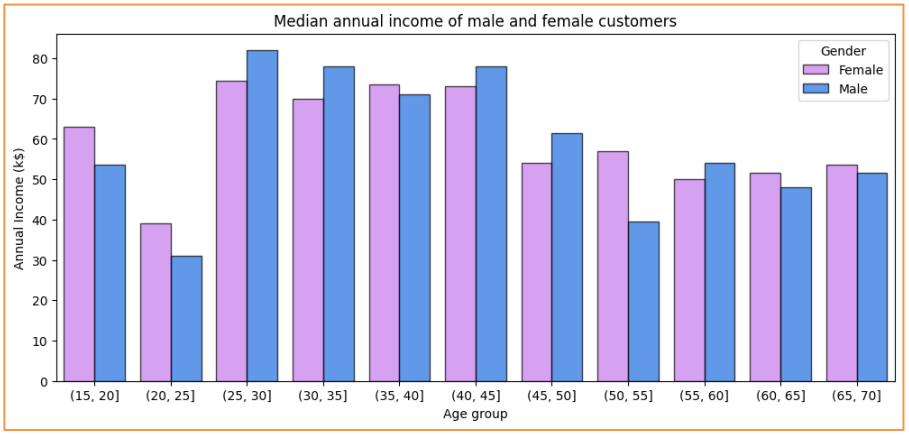

Figure 6: Median annual income of male and female

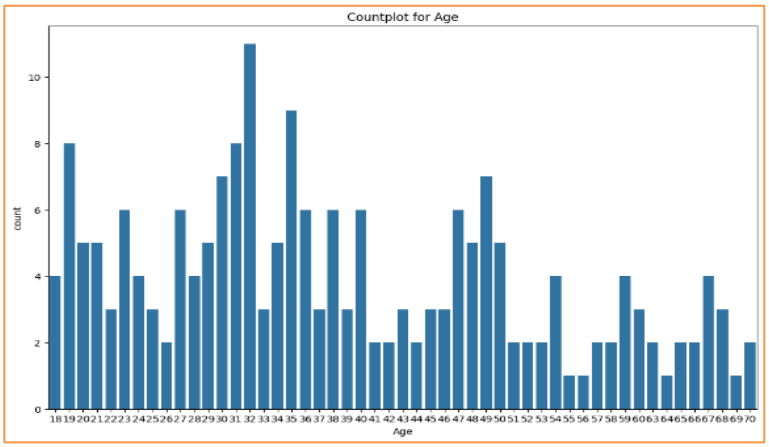

Figure 7: Distribution of age

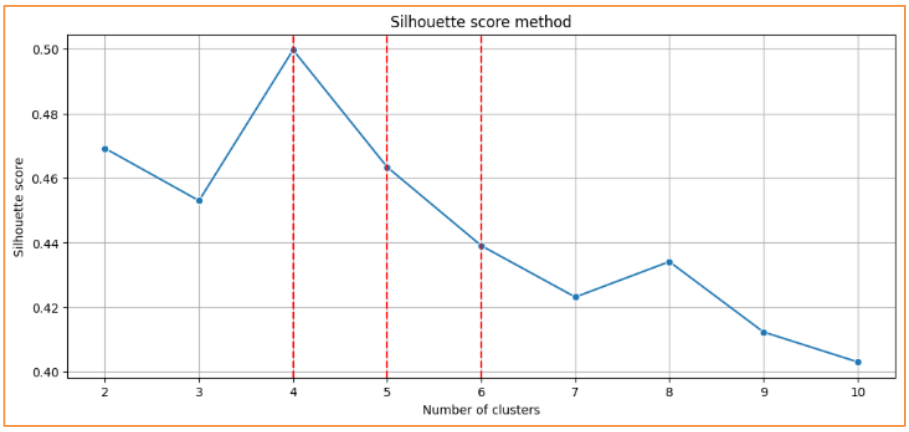

Figure 8: Calculating Pearson's correlation

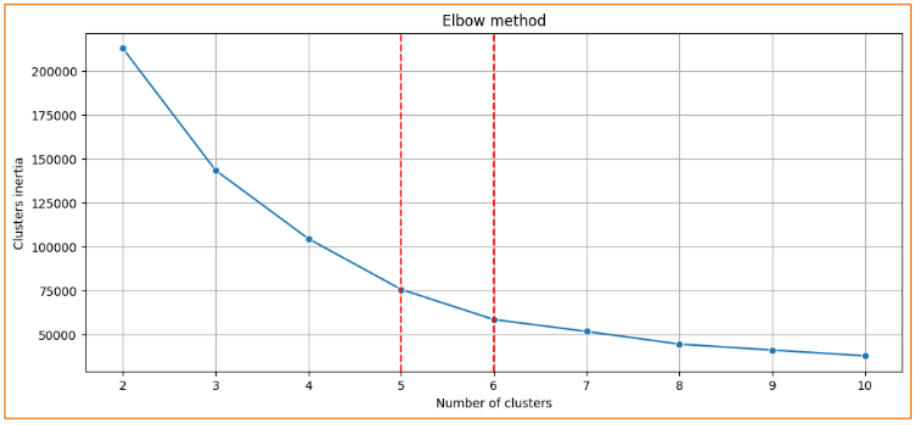

Figure 9: Elbow Method

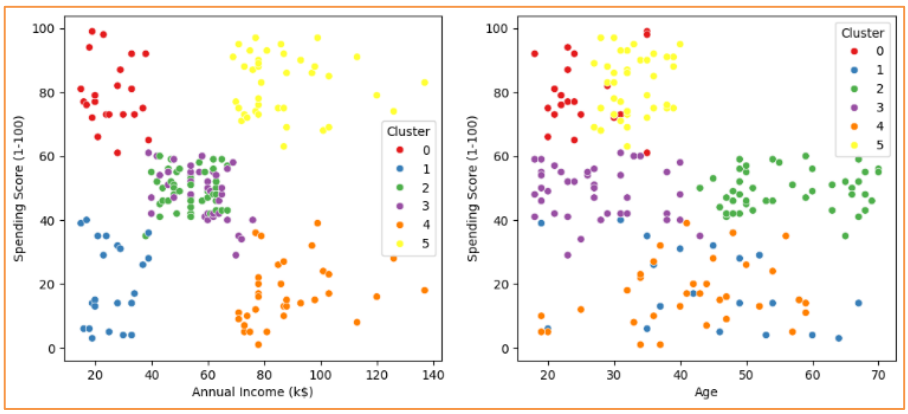

Figure 10: Spending score of Annual income and Age

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

Figure 11: Output of dataset (0 to 4)

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

Figure 12: Output of dataset for mean, max, count, min

8. SWOT ANALYSIS

Strengths

1. High Precision and Accuracy:

- Machine learning models can process vast amounts of data with high precision, leading to more accurate customer segments.
- Algorithms can uncover hidden patterns and correlations that traditional methods might miss.

2. Scalability:

- Machine learning solutions can handle large datasets, making them suitable for businesses of all sizes, from small startups to large enterprises.

3. Automation:

- Automation of segmentation processes reduces the need for manual intervention, saving time and resources.

4. Personalization:

- Improves customer experience by providing tailored product recommendations and offers.

Weaknesses

1. Complexity:

- Implementing machine learning models requires technical expertise and knowledge, which might be a barrier for some organizations.

2. Data Quality and Availability:

- The effectiveness of machine learning models depends heavily on the quality and quantity of data available.

3. Cost:

- Requires investment in infrastructure, software, and skilled personnel.

9. CONCLUSION

In conclusion, the integration of machine learning techniques into customer segmentation represents a transformative evolution in how businesses understand and interact with their customer base. By combining traditional clustering algorithms with advanced machine learning methods, companies can achieve unprecedented levels of accuracy and efficiency in segmenting their clientele. This hybrid approach leverages both supervised and unsupervised learning, utilizing labeled and unlabeled data to refine segment definitions and accommodate the dynamic nature of customer preferences.

The future of customer segmentation using machine learning holds immense promise, characterized by automated, dynamic, and behavior-based segmentation driven by predictive analytics and ethical considerations. Through continuous refinement and adaptation, businesses can ensure that their segmentation strategies remain relevant and actionable in an ever-changing marketplace. Moreover, by prioritizing customer privacy and data protection, businesses can foster trust and loyalty among their clientele, ultimately leading to sustainable growth and success in competitive markets.

10. GITHUB LINK OF PROJECT

https://github.com/RajShoaib/Customer-Segmentation-using-hybrid-model

11. REFERENCE

[1] Customer Segmentation Tutorial Python Projects K-Means Algorithm Python Training Edureka. https://www.youtube.com/watch?v=4jv1pUrG0Zk&t=1510s

[2] Machine Learning Project With Code Customer Segmentation End To End Implementation by Data Science Diaries. https://www.youtube.com/watch?v=frvailWW6Iw&t=1046s

[3] Implementing Customer Segmentation Using Machine Learning [Beginners Guide] by Dhiraj Kumar. https://neptune.ai/blog/customer-segmentation-using-machine-learning

[4] Customer Segmentation using K-means Clustering J Madhu1 , Kavita K Revanakar2 , Lavanya3 , Akash4 Department of Computer Science and Engineering Srinivas Institute of Technology, Mangalore, Karnataka, India. https://ijaem.net/issue_dcp/Customer%20Segmentation%20using%20K%20means%20Clustering.pdf

[5] How to Perform Customer Segmentation in Python– Machine Learning Tutorial by Ibrahim Abayomi Ogunbiyi. https://www.freecodecamp.org/news/customer-segmentation- python-machine-learning/

[6] Customer Segmentation using K-Means Clustering l Machine Learning Project by Engineering WalaBhaiya. https://www.youtube.com/watch?v=he9FrAo6pms&t=1048s

[7] Customer Segmentation Using Machine Learning.

https://www.javatpoint.com/customer-segmentation-using- machine-learning.

[8] Customer Segmentation Using Machine Learning Prof. Nikhil Patankar a ,1, Soham Dixit a , Akshay Bhamare a , Ashutosh Darpel a and Ritik Raina a a Dept. Of Information Technology Sanjivani College of Engineering, Kopargaon 423601 (MH), India. https://www.researchgate.net/publication/356756320_Customer_Segmentation_Using_Machine_Learning

[9] How to Use Machine Learning For Customer Segmentation by Eric Vardon. https://hawke.ai/blog/machine-learning-for-customer- segmentation/

[10] How To Solve Customer Segmentation Problem With Machine Learning by Mrinal Singh Walia. https://www.analyticsvidhya.com/blog/2021/06/how-to-solve-customer-segmentation-problem-with-machine-learning/