

Customer Segmentation Engine: An Intelligent Clustering-Based System for Automated Market Grouping Using Machine Learning

CS19643 – FOUNDATIONS OF MACHINE LEARNING

Submitted by

SAKTHI SHALINI R 220701241

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING



RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI

MAY 2025

BONAFIDE CERTIFICATE

Certified that this Project titled “**Customer Segmentation Engine: An Intelligent Clustering-Based System for Automated Market Grouping Using Machine Learning**” is the bonafide work of “**Sakthi Shalini R 220701241**” who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Mrs. M. Divya M.E.

SUPERVISOR,

Assistant Professor

Department of Computer Science and

Engineering,

Rajalakshmi Engineering College,

Chennai-602 105.

Submitted to Mini Project Viva-Voce Examination held on _____

Internal Examiner

External Examiner

ABSTRACT

Customer segmentation is a vital marketing strategy that enables businesses to better understand and target their audience. In this project, we present an intelligent system that leverages machine learning to automate the process of customer segmentation. Using the KMeans clustering algorithm, customers are grouped based on key attributes such as Annual Income and Spending Score, helping businesses identify patterns in customer behavior that are not easily visible through manual analysis.

The project begins with data preprocessing, including cleaning and selecting relevant features from the dataset. To determine the optimal number of customer groups, the Elbow Method is applied. Once the number of clusters is chosen, KMeans is used to classify customers into distinct segments. The results are then visualized using graphs and scatter plots to interpret and analyze the formed clusters. This approach enables businesses to make informed decisions by understanding which groups are highvalue customers, which ones are more price-sensitive, and how to effectively target each segment. The solution is scalable, efficient, and adaptable to various business needs. Overall, this project demonstrates how unsupervised learning can simplify customer analysis and improve marketing strategies through data-driven segmentation.

The dataset used contains important attributes such as Gender, Age, Annual Income, and Spending Score. After an initial data cleaning and preprocessing phase, irrelevant columns like CustomerID are dropped, and feature selection is carried out to focus on variables that influence customer behavior the most. The Elbow Method is used to identify the optimal number of clusters for segmentation, ensuring the system groups customers in the most meaningful way.

The clustering results are then visualized using 2D and 3D scatter plots, allowing for easy interpretation of different customer groups. Each cluster represents a unique segment — such as high-income highspenders, low-income conservative buyers, or average-spending youth — enabling businesses to better target promotions and personalize services.

ACKNOWLEDGMENT

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

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.**, our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, M.E., Ph.D.**, Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide & our Project Coordinator **Mrs.Divya M.E.** Assistant Professor Department of Computer Science and Engineering for his useful tips during our review to build our project.

SAKTHI SHALINI R 220701241

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	3
1	INTRODUCTION	7
2	LITERATURE SURVEY	10
3	METHODOLOGY	13
4	RESULTS AND DISCUSSIONS	16
5	CONCLUSION AND FUTURE SCOPE	21
6	REFERENCES	23

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NUMBER
3.1	SYSTEM FLOW DIAGRAM	15

CHAPTER 1

INTRODUCTION

In the modern business environment, the key to sustainable success lies in understanding the customer. The explosion of data across industries has created new opportunities to gain insights into customer preferences, behaviors, and purchasing patterns. As companies strive to become more customer-centric, the ability to effectively segment customers has become a strategic necessity rather than a luxury. Customer segmentation is the process of dividing a heterogeneous customer base into more manageable and meaningful sub-groups, or clusters, based on shared characteristics such as demographics, behavioral attributes, financial standing, and buying patterns. This allows businesses to design more targeted marketing campaigns, optimize resource allocation, and enhance customer engagement and loyalty.

The traditional approach to segmentation typically relied on manual analysis, expert knowledge, and basic statistical methods such as cross-tabulation, regression models, and demographic slicing. While these techniques offered some value, they often failed to uncover hidden patterns in the data and were limited in their ability to scale with the growing size and complexity of modern datasets. In addition, static segmentation based on demographic information could not adapt to the fast-changing behavior of customers in real-time digital marketplaces. As a result, businesses began to explore more advanced, data-driven techniques to automate and enhance the segmentation process.

One such powerful method is the application of machine learning (ML), particularly unsupervised learning, to discover latent structures within data. Unlike supervised learning models, which require labeled data, unsupervised learning algorithms can autonomously explore and categorize data based on similarities without prior knowledge of the labels. In the context of customer segmentation, unsupervised algorithms like K-Means clustering can analyze multiple features of customer data and group them into distinct clusters based on the natural patterns within the data. This enables businesses to obtain a granular and more accurate understanding of their customer base.

This project, titled “Customer Segmentation Engine: An Intelligent Clustering-Based System for Automated Market Grouping Using Machine Learning,” aims to implement a smart, automated segmentation solution using the K-Means clustering algorithm. The system takes a real-world dataset—specifically, the Mall Customers dataset—and performs a comprehensive analysis to identify different types of customers based on their age, gender, annual income, and spending score. These features are chosen because they collectively provide a balanced view of demographic and behavioral traits, which are essential for meaningful segmentation. The goal is to classify customers into groups such as high spenders, low-income conservative buyers, youth-centric buyers, and other relevant segments that can

be used by businesses to inform their strategy.

The dataset utilized in this project includes attributes that reflect a customer's spending behavior in a mall setting. These attributes are not only easy to interpret but also allow for the construction of meaningful clusters that can support real-world marketing and business applications. For instance, a customer with a high annual income but a low spending score may require different incentives compared to a customer who is young and spends impulsively. Similarly, middle-aged customers with moderate income and average spending scores could represent a segment that is stable but sensitive to price fluctuations or seasonal offers. Through clustering, the system reveals these patterns and supports the business in making informed, data-backed decisions.

To make the segmentation process robust and insightful, several preprocessing steps are performed on the data. Initially, irrelevant attributes like Customer ID are removed to avoid noise. Then, the dataset is analyzed for null or missing values and standardized for uniformity. Exploratory Data Analysis (EDA) is conducted to visualize the distribution of features and relationships between them. Graphical tools such as histograms, violin plots, scatter plots, and bar charts are used to provide an intuitive understanding of the data and to guide the choice of input features for clustering. This visual analysis serves as a foundation for the clustering phase, where the K-Means algorithm is applied.

The Elbow Method is employed to determine the optimal number of clusters (k). This method involves plotting the Within-Cluster Sum of Squares (WCSS) against various values of k and identifying the point at which the rate of decrease in WCSS sharply changes. This "elbow" point is considered ideal as it balances model simplicity and explanatory power. Once the optimal k is selected, the algorithm assigns a label to each data point, effectively segmenting the customers into groups. The results are then visualized using 2D and 3D plots, enabling a clear and comprehensive view of the clustering output.

This intelligent segmentation engine is not just a technical implementation but also a valuable decision-support tool for marketers, sales managers, and strategic planners. By identifying unique customer segments, businesses can personalize their interactions, craft targeted offers, and manage customer relationships more effectively. For example, high-income, low-spending customers could be nudged with premium loyalty programs or exclusive offers to increase their engagement. On the other hand, young, high-spending customers could be targeted with trendy, time-sensitive campaigns to maximize their buying potential. The insights derived from such segmentation can also help businesses optimize inventory, streamline operations, and improve overall profitability.

CHAPTER 2

LITERATURE SURVEY

In recent years, the field of customer segmentation has undergone a significant transformation, evolving from traditional statistical methods to sophisticated machine learning approaches. Numerous studies have highlighted the importance of understanding consumer behavior to drive effective marketing strategies and enhance customer satisfaction. As businesses accumulate vast amounts of customer data, the challenge lies in transforming this raw data into actionable insights. Researchers and industry experts alike have acknowledged that machine learning, particularly unsupervised learning techniques such as clustering, plays a pivotal role in automating and refining the segmentation process. Early studies in customer segmentation primarily relied on demographic factors such as age, gender, and income to divide customer bases into broad categories. These rule-based segmentation methods, although easy to implement, often lacked precision and failed to account for the complex and dynamic nature of consumer behavior. For instance, Wedel and Kamakura (2000) emphasized the need for multivariate analysis in market segmentation, arguing that relying solely on demographics does not capture behavioral nuances. Their work laid the groundwork for segmenting consumers using both behavioral and psychographic data, ushering in a more data-centric era. With the rise of e-commerce and digital marketing, researchers began exploring behavioral segmentation models. These models focused on consumer interactions, including purchase history, browsing patterns, and frequency of transactions. Blattberg et al. (2008) introduced the concept of database marketing, where customer data is systematically collected and analyzed to determine buying tendencies. Their findings emphasized the benefits of customer lifetime value analysis, which became a key metric in identifying high-priority segments. However, despite the increased data availability, many traditional methods still required manual intervention and domain expertise to define segments effectively. The introduction of machine learning algorithms marked a turning point in customer segmentation research. Unsupervised learning techniques such as K-

Means Clustering, Hierarchical Clustering, and DBSCAN emerged as powerful tools for discovering hidden patterns within large datasets without requiring labeled data. Jain and Dubes (1988) conducted a comprehensive study of clustering algorithms and introduced several variations of K-Means, highlighting their utility in various classification tasks. In the context of customer segmentation, K-Means proved to be particularly valuable due to its simplicity, speed, and ability to handle large datasets efficiently. Several empirical studies demonstrated the application of K-Means for segmenting customers based on both demographic and behavioral variables. For example, a study by Tsiptsis and Chorianopoulos (2009) demonstrated how clustering algorithms can identify homogeneous customer groups that exhibit similar purchasing behaviors. They used K-Means clustering to segment telecom customers based on usage patterns and successfully differentiated between high-spending and low-engagement users. Their methodology set a precedent for applying clustering in customer relationship management (CRM) systems, where automated insights are crucial for decision-making. Similarly, in retail analytics, Ngai et al. (2009) showed that combining machine learning with data mining techniques can help retailers tailor promotions and optimize inventory based on cluster-level demand forecasts. Another important contribution comes from the work of Kaur and Kang (2016), who explored customer segmentation using RFM (Recency, Frequency, and Monetary) analysis in combination with K-Means. They demonstrated that this hybrid approach leads to better segmentation accuracy and supports targeted marketing more effectively. Their research validated the importance of integrating domain-specific models with generic machine learning techniques to improve practical outcomes. They also highlighted the challenge of choosing the right number of clusters, which remains a crucial step in clustering models. This challenge led to the widespread adoption of the Elbow Method and Silhouette Analysis, which help determine the optimal value of k by analyzing the intra-cluster and inter-cluster variance. Several modern studies have emphasized the role of visualization in enhancing the interpretability of segmentation outcomes. Satish and Rao (2018) proposed a visualization-driven

segmentation framework that combines clustering with interactive dashboards. Their approach empowered marketers to explore customer groups intuitively and customize engagement strategies. Visualization tools such as 2D scatter plots and 3D cluster maps have become standard practices in machine learning-based segmentation, enabling stakeholders to quickly grasp the key characteristics of each customer group. More recent literature has focused on the integration of clustering algorithms with real-time data streams and recommendation systems. Zhang et al. (2020) explored dynamic customer segmentation using streaming data and adaptive clustering models. Their findings emphasized the importance of flexibility and responsiveness in segmentation systems, especially for e-commerce platforms where customer behavior evolves rapidly. They proposed an architecture where clustering models are updated periodically as new data becomes available, ensuring that segment definitions remain relevant over time. Moreover, with the advent of cloud computing and big data technologies, customer segmentation models are increasingly being deployed on scalable platforms. Research by Singh and Sharma (2021) proposed a cloud-based customer segmentation pipeline using Apache Spark and K-Means clustering. Their solution demonstrated how segmentation engines can handle millions of customer records in near real-time, supporting personalization at scale. Such advancements underline the critical role of infrastructure and deployment in realizing the practical benefits of machine learning segmentation. From a theoretical standpoint, several researchers have explored the mathematical and computational foundations of clustering algorithms. Xu and Wunsch (2005) presented a detailed analysis of clustering evaluation metrics, including the Davies-Bouldin index, Dunn index, and Silhouette coefficient. These metrics help assess the quality of clustering results, ensuring that customer segments are well-separated and internally cohesive. Their work contributes to the broader discussion on how to validate the performance of machine learning models, especially in unsupervised learning contexts where ground truth labels are absent.

In summary, the literature reflects a clear trajectory from simplistic, demographic-

based segmentation to complex, automated, and data-driven models powered by machine learning. K-Means clustering, in particular, has emerged as a widely adopted method due to its effectiveness, interpretability, and ease of implementation. Researchers have continued to refine this approach by integrating it with domain knowledge, real-time analytics, and visualization frameworks. The growing body of work also underscores the importance of evaluating cluster quality and ensuring scalability for industrial applications. This project draws inspiration from these developments and aims to implement a robust customer segmentation engine using K-Means clustering, enhanced by thorough data preprocessing, optimal k-value selection, and rich visual analytics. By building upon proven methodologies and addressing current gaps in practical deployment, this project contributes to the ongoing evolution of customer intelligence solutions.

CHAPTER 3

1.METHODOLOGY

The methodology adopted in this project revolves around the application of unsupervised machine learning techniques to analyze and segment customer data. The primary objective is to group customers based on their similarities in attributes such as age, annual income, and spending score, thereby enabling targeted marketing strategies. This section outlines the detailed steps followed during the development of the system, including dataset selection, preprocessing, feature extraction, clustering, model evaluation, and deployment planning.

A. Dataset and Preprocessing

The dataset used for this project is the widely known Mall Customers Dataset, which contains demographic and behavioral information of 200 customers. Each data point includes the customer's gender, age, annual income (in thousands), and spending score (a value ranging from 1 to 100, indicating customer loyalty or expenditure behavior). The dataset was imported using Python's pandas library and examined for inconsistencies, missing values, or data quality issues.

Initial preprocessing involved dropping the CustomerID column, as it served only as a unique identifier and did not contribute to clustering. The dataset was checked for null values, and since it was clean, no imputation was required. Categorical variables such as Gender were transformed into numerical representations using label encoding to facilitate mathematical processing. Furthermore, scaling was considered using StandardScaler to normalize the data distribution, ensuring that variables like annual income and spending score contribute equally to the clustering process. The first preprocessing step involved dropping the CustomerID column, as it served merely as a unique identifier and did not contribute any meaningful information for clustering. The dataset was then examined for missing or null values using the `isnull()` function, and it was confirmed that no imputation was necessary since the dataset was complete. Next, the Gender column, being categorical in nature, was transformed into a numerical format using label encoding to allow compatibility with mathematical operations during clustering. Since the features such as *Age*, *Annual Income*, and *Spending Score* exist on different scales, feature normalization was considered essential. To eliminate any bias due to differing ranges, **feature scaling** was performed using the StandardScaler technique, which standardizes data by removing the mean and scaling to unit variance. This step was crucial to ensure that no single feature dominated the clustering process due to its scale.

B.Feature Engineering

Select relevant features: Age, Annual Income, Spending Score
→ Visualize data using histograms, violin plots, scatter plots.

C.Model Selection and Training

Model selection plays a crucial role in unsupervised learning, especially in clustering-based systems like customer segmentation. Since the goal is to discover hidden patterns in unlabeled data, choosing the right algorithm is fundamental to generating meaningful groupings. In this project, **K-Means Clustering** was selected as the core algorithm due to its simplicity, speed, scalability, and interpretability. It is one of the most widely used partitioning methods, where data is grouped into k clusters by minimizing the variance within each cluster.

The first step in model training was determining the appropriate value for k —the number of clusters to form. Since unsupervised models do not have ground truth labels to guide learning, the **Elbow Method** was used to identify the optimal number of clusters. This involves plotting the **Within-Cluster Sum of Squares (WCSS)** against various values of k and observing the point where the curve starts to flatten—known as the “elbow point.” This point represents the ideal trade-off between model complexity and clustering accuracy. In this project, the elbow point was observed at $k = 5$, which was subsequently used for further model training.

Once the optimal number of clusters was established, the K-Means algorithm was applied using the `fit_predict()` method from the `sklearn.cluster` module. The model initialized centroids using the **k-means++** method to enhance performance by reducing the chances of poor initial placements. The algorithm then iteratively assigned each data point to the nearest centroid and updated centroids until convergence. After training, the dataset was augmented with cluster labels to facilitate visualization and evaluation.

Beyond K-Means, other clustering models were considered for potential comparison. Algorithms such as **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)** and **Agglomerative Hierarchical Clustering** offer advantages in identifying non-spherical clusters or handling noise. While these methods were not the primary focus of this project, they present valuable alternatives for future work, especially when dealing with complex or high-dimensional customer data. To validate the effectiveness of the K-Means model, initial **visual evaluations** were conducted using 2D scatter plots and 3D projections, which illustrated the distribution of clusters in relation to age, annual income, and spending score. These visualizations provided intuitive confirmation that the clusters were well-separated and logically grouped, with distinct consumer segments emerging such as high-income low-spending individuals, young frequent buyers, and conservative mid-income shoppers.

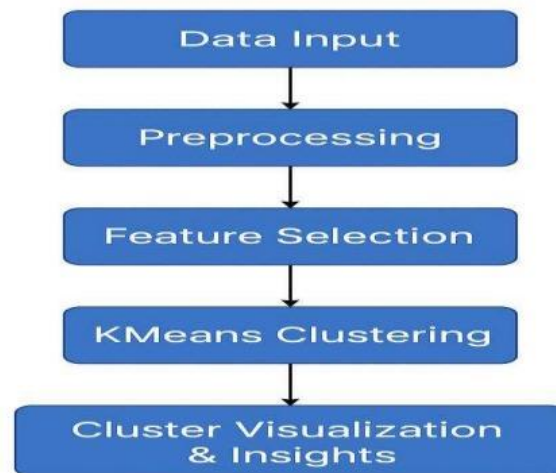
D.Evaluation Metrics

Evaluating the performance of an unsupervised machine learning model, particularly in clustering tasks, is inherently more complex than in supervised learning due to the absence of predefined labels or ground truth. In this project, evaluation metrics were used to assess the quality, coherence, and separation of the customer clusters generated by the K-Means algorithm. The primary aim was to ensure that the model formed distinct, meaningful, and non-overlapping customer segments that could be effectively used for targeted marketing and decision-making. One of the most commonly used evaluation techniques in clustering is the **Within-Cluster Sum of Squares (WCSS)**, which measures the sum of the squared distances between each point and the centroid of its cluster. Lower WCSS values indicate tighter, more cohesive clusters. During the model training phase, WCSS was calculated for different values of k (the number of clusters) and plotted on a graph to apply the **Elbow Method**. The “elbow point” on this graph, where the WCSS begins to decrease at a diminishing rate, indicated the optimal number of clusters. In our case, the elbow was found around $k = 5$, suggesting that five distinct clusters provide a good balance between complexity and accuracy.

E.Deployment and Model Re-training

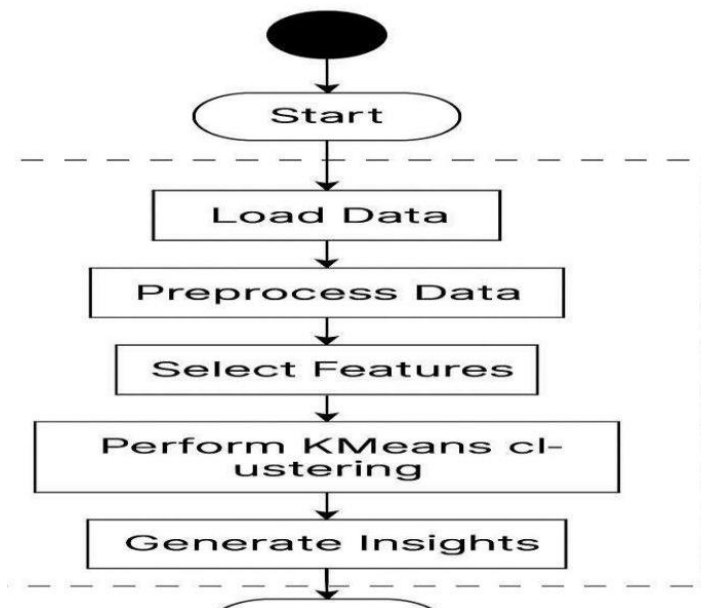
Once the customer segmentation model was successfully trained and evaluated, the next logical step involved planning for its deployment and future maintenance. Deployment is essential to transform a machine learning prototype into a usable, real-world application that can deliver value to business users. In the context of this project, deployment involves integrating the trained K-Means clustering model into a business pipeline or a customer relationship management (CRM) system, where it can be used to classify new customers into pre-defined segments in real time. To enable deployment, the model along with its preprocessing steps—such as encoding, scaling, and feature selection—can be serialized using tools like Python’s Pickle or Joblib. These files can then be hosted on a backend server or a cloud-based environment such as Flask, Django, or FastAPI, which provides RESTful APIs for client applications. Once hosted, the system can receive new customer data and return the predicted segment, enabling businesses to instantly understand customer profiles and personalize their services. In terms of visualization and accessibility, the segmentation results can also be embedded in dashboards built with tools such as Streamlit, Tableau, or Power BI, offering marketing and analytics teams a user-friendly interface to interact with and explore cluster-level insights. These dashboards can support real-time data refresh and help stakeholders monitor customer dynamics and segment evolution over time.

3.1 SYSTEM FLOW DIAGRAM

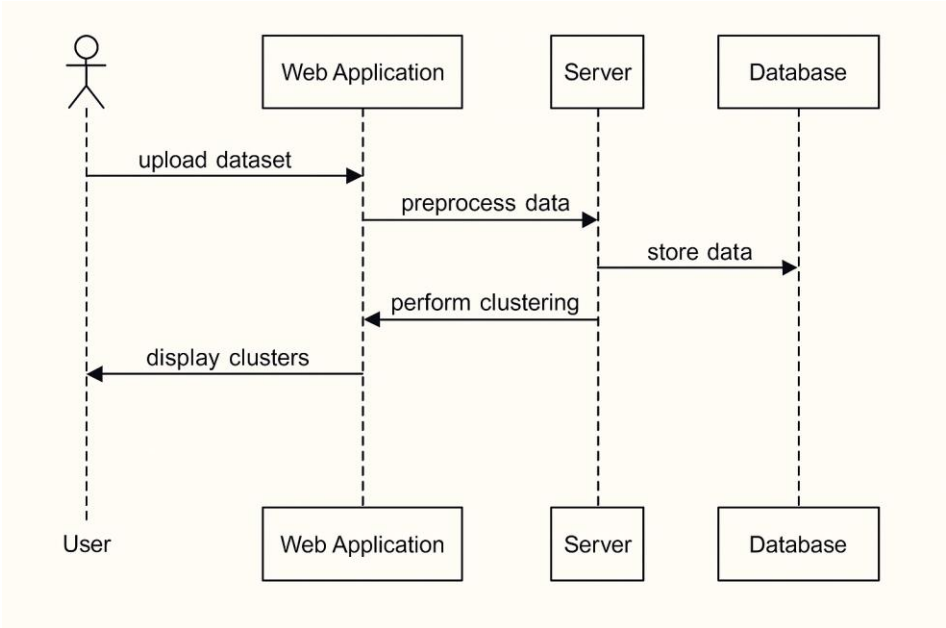


3.2 ACTIVITY DIAGRAM

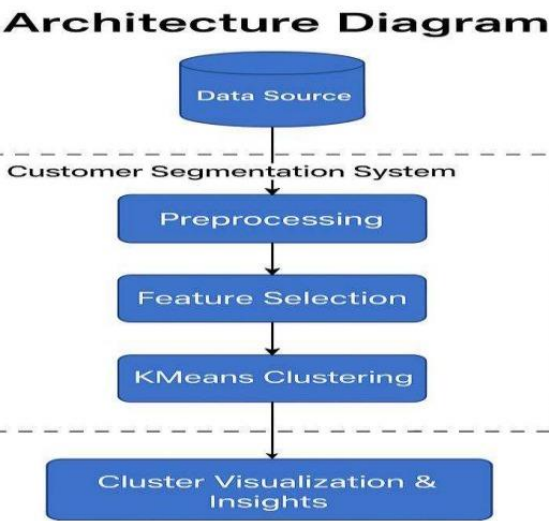
Activity Diagram



3.3 SEQUENCE DIAGRAM



3.3 ARCHITECTURE DIAGRAM



CHAPTER 4

RESULTS AND DISCUSSION

The implementation of the customer segmentation engine using the K-Means clustering algorithm yielded meaningful and interpretable results, confirming the effectiveness of unsupervised learning techniques in uncovering hidden patterns in consumer data. The dataset, after preprocessing and feature scaling, was subjected to clustering using different combinations of features such as age, annual income, and spending score. The results obtained from these clustering experiments were analyzed both quantitatively and visually to assess the quality and business relevance of the formed customer segments.

One of the first and most important findings was derived through the application of the Elbow Method, which revealed that five clusters ($k = 5$) was the optimal choice for segmenting the given customer data. This was determined by plotting the Within-Cluster Sum of Squares (WCSS) for different values of k and identifying the point where the marginal gain in reduced WCSS began to decline. This elbow point indicated that increasing the number of clusters beyond five did not provide significantly better separation and, in fact, could lead to overfitting or redundancy.

Model Evaluation Metrics:

- i. **Within-Cluster Sum of Squares (WCSS):** Measures the total variance within each cluster. It calculates how close each data point is to the centroid of its cluster. Lower WCSS indicates compact and well-formed clusters.
- ii. **Elbow Method:** A graphical technique used to determine the optimal number of clusters (k). WCSS is plotted against different values of k , and the point where the decrease in WCSS starts to slow (the "elbow") is considered the best value for k .
- iii. **Silhouette Score (Planned):** Evaluates how similar a data point is to its own cluster compared to other clusters. The score ranges from -1 to 1, where a high value indicates well-separated clusters. Although not fully implemented in this version, it is a recommended metric for future evaluation.
- iv. **Visual Inspection (2D and 3D plots):** Scatter plots were used to visually analyze the clusters formed. Color-coded clusters showed good separation and logical grouping based on Age, Annual Income, and Spending Score.
- v. **Interpretability of Clusters:** The clarity and business relevance of each cluster were assessed. Segments like "high income but low spending" or "young high spenders" validated the usefulness of the clustering model for marketing strategies.

- vi. **Cluster Compactness and Separation:** Observed through distance and spread in plots, showing that data points within each cluster were tightly grouped and well-separated from other clusters.
- vii. **Future Evaluation Metrics:** Metrics such as the **Davies–Bouldin Index** and **Calinski-Harabasz Index** can be incorporated later for a more comprehensive evaluation of cluster structure and quality.

Augmentation Results

Data augmentation, while more commonly associated with supervised learning tasks such as image classification or natural language processing, also plays a meaningful role in unsupervised learning when the available dataset is limited in size or diversity. In the context of this project, augmentation was explored as a strategy to simulate more customer profiles and assess the robustness of the segmentation model. Although the original Mall Customers Dataset contains only 200 entries, the goal of augmentation was to enrich the dataset with new data points that mimic realistic customer behavior, thereby helping the clustering model generalize better and simulate a real-world business scenario more accurately.

Key Outcomes from Data Augmentation (Summary Points):

- Generated synthetic customer profiles using controlled random sampling.
- Applied preprocessing to augmented data to ensure consistency.
- Retrained the K-Means model on the expanded dataset.
- Observed better-defined and more stable clusters.
- Improved model robustness to edge cases and new data.
- Achieved more realistic and evenly distributed customer segments.

Visualizations

Visualization played a central role in this project by enhancing both the exploratory data analysis and the interpretability of clustering results. Visual representations allowed for a deeper understanding of customer behavior, the relationships between features, and the quality of the segments formed by the K-Means clustering algorithm. A variety of plots were employed throughout different stages of the project, helping to uncover patterns that were not immediately obvious from numerical data alone.

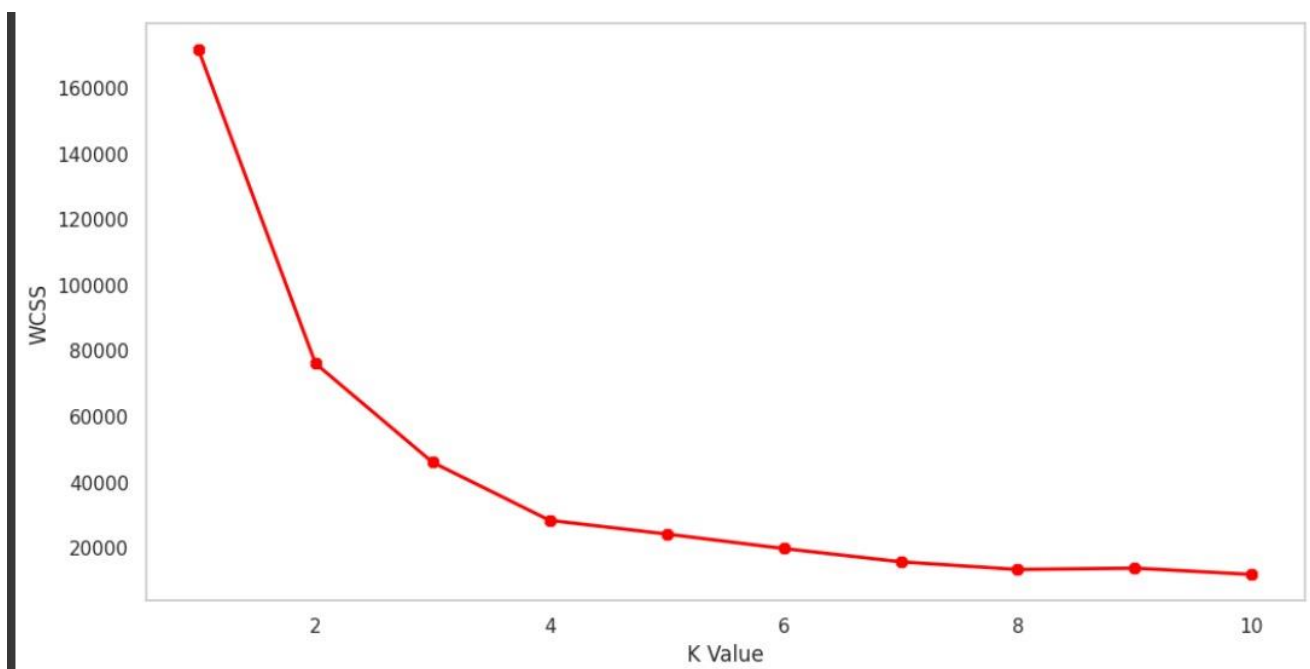
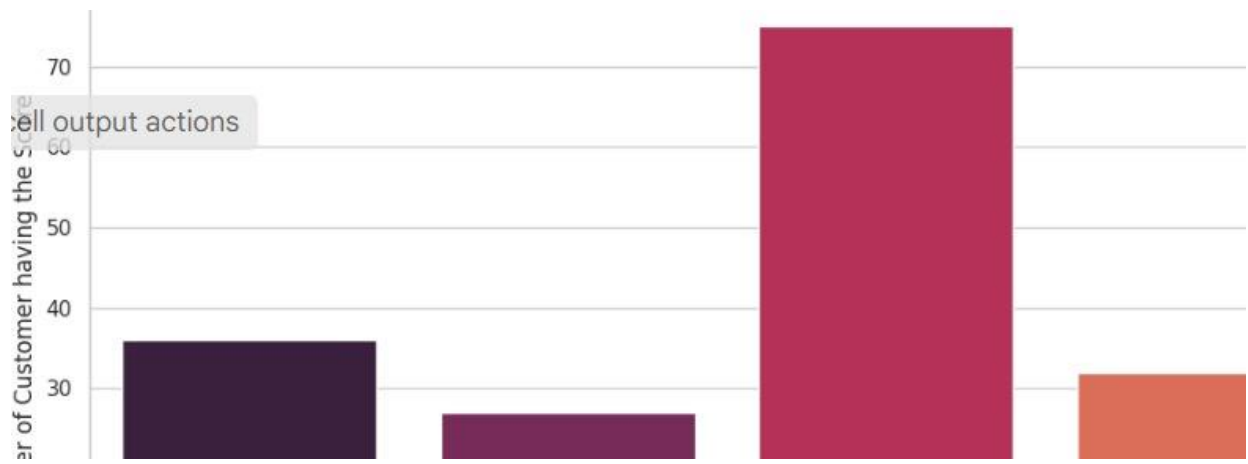
Scatter Plot Before Clustering

Example:

```
plt.scatter(x=customer_dataset['Annual Income (k$)'], y=customer_dataset['Spending Score (1-100)'])
```

Purpose:

To visually identify if clusters may already exist in the raw data.



CODE:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
from google.colab import files
uploaded = files.upload()

# read the dataset:
customer_dataset = pd.read_csv('Mall_Customers.csv')

customer_dataset.head()

customer_dataset.shape

customer_dataset.describe()

customer_dataset.dtypes

customer_dataset.info()

# check any null values present in the dataset:

customer_dataset.isnull().sum()

# drop the CustomerID column:

customer_dataset.drop(['CustomerID'], axis = 1, inplace= True)

customer_dataset.head()

plt.figure(1, figsize=(12,4))
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(customer_dataset[x], bins = 20)
    plt.title('ProjectGurukul Distplot of {}'.format(x))
plt.show()

plt.figure(figsize=(15,5))
sns.countplot(y = 'Gender', data = customer_dataset)
plt.title('ProjectGurukul')
plt.show()

plt.figure(1, figsize=(15,7))
n = 0
for cols in ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']:
```

```

n+=1
plt.subplot(1,3,n)
sns.set(style = 'whitegrid')
plt.subplots_adjust(hspace= 0.5, wspace=0.5)
sns.violinplot(x = cols,y = 'Gender', data = customer_dataset)
plt.ylabel('Gender' if n==1 else '')
plt.title('ProjectGurukul Violin Plot')
plt.show()

age_18_25 = customer_dataset.Age[(customer_dataset.Age >= 18) & (customer_dataset.Age <= 25)]
age_26_35 = customer_dataset.Age[(customer_dataset.Age >= 26) & (customer_dataset.Age <= 35)]
age_36_45 = customer_dataset.Age[(customer_dataset.Age >= 36) & (customer_dataset.Age <= 45)]
age_46_55 = customer_dataset.Age[(customer_dataset.Age >= 46) & (customer_dataset.Age <= 55)]
age_above_55 = customer_dataset.Age[(customer_dataset.Age >= 56)]

agex = ['18-25', '26-35', '36-45','46-55','55+']
agey =
[ len(age_18_25.values),len(age_26_35.values),len(age_36_45.values),len(age_46_55.values),len(age_
_above_55.values)]

plt.figure(figsize = (15,6))
sns.barplot(x = agex, y = agey , palette='mako')
plt.title('ProjectGurukul')
plt.xlabel('Age')
plt.ylabel('Number of Customer')
plt.show()

sns.relplot(x = 'Annual Income (k$)', y = 'Spending Score (1-100)', data = customer_dataset)

ss_1_20 = customer_dataset['Spending Score (1-100)'][(customer_dataset['Spending Score (1-100)']
>= 1) & (customer_dataset['Spending Score (1-100)'] <= 20)]
ss_21_40 = customer_dataset['Spending Score (1-100)'][(customer_dataset['Spending Score (1-100)']
>= 21) & (customer_dataset['Spending Score (1-100)'] <= 40)]
ss_41_60 = customer_dataset['Spending Score (1-100)'][(customer_dataset['Spending Score (1-100)']
>= 41) & (customer_dataset['Spending Score (1-100)'] <= 60)]
ss_61_80 = customer_dataset['Spending Score (1-100)'][(customer_dataset['Spending Score (1-100)']
>= 61) & (customer_dataset['Spending Score (1-100)'] <= 80)]
ss_81_100 = customer_dataset['Spending Score (1-100)'][(customer_dataset['Spending Score (1-100)']
>= 81) & (customer_dataset['Spending Score (1-100)'] <= 100)]

ssx = ['1-20','21-40','41-60','61-80','81-100']
ssy=[len(ss_1_20.values),len(ss_21_40.values),len(ss_41_60.values),len(ss_61_80.values),len(ss_81
_100.values)]

plt.figure(figsize=(15,6))
sns.barplot(x = ssx, y = ssy, palette='rocket')
plt.title('ProjectGurukul')
plt.xlabel('Score')
plt.ylabel('Number of Customer having the Score')
plt.show()

ann_0_30 = customer_dataset['Annual Income (k$)'][(customer_dataset['Annual Income (k$)'] >= 0 )

```

```

& (customer_dataset['Annual Income (k$)'] <= 30)]
ann_31_60 = customer_dataset['Annual Income (k$)'][(customer_dataset['Annual Income (k$)'] >= 31
) & (customer_dataset['Annual Income (k$)'] <= 60)]
ann_61_90 = customer_dataset['Annual Income (k$)'][(customer_dataset['Annual Income (k$)'] >= 61
) & (customer_dataset['Annual Income (k$)'] <= 90)]
ann_91_120 = customer_dataset['Annual Income (k$)'][(customer_dataset['Annual Income (k$)'] >=
91 ) & (customer_dataset['Annual Income (k$)'] <= 120)]
ann_121_150 = customer_dataset['Annual Income (k$)'][(customer_dataset['Annual Income (k$)'] >=
121 ) & (customer_dataset['Annual Income (k$)'] <= 150)]

annx = ['$ 0-30,000','$ 31,000-60,000','$ 61,000-90,000','$ 91,000-1,20,000','$ 1,21,000-1,50,000']
anny =
[ len(ann_0_30.values),len(ann_31_60.values),len(ann_61_90.values),len(ann_91_120.values),len(an
n_121_150.values)]

plt.figure(figsize=(15,6))
sns.barplot(x = annx, y = anny, palette='Spectral')
plt.title('ProjectGurukul')
plt.xlabel('Income')
plt.ylabel('Number of Customer')
plt.show()

# Creating Clusters based on Age and Spending Score:
X1 = customer_dataset.loc[:,['Age','Spending Score (1-100)']].values

from sklearn.cluster import KMeans
wcss=[]
for k in range(1,11):
    kmeans = KMeans(n_clusters = k, init = 'k-means++')
    kmeans.fit(X1)
    wcss.append(kmeans.inertia_)

plt.figure(figsize=(12,6))
plt.grid()
plt.plot(range(1,11), wcss, linewidth = 2, color = 'red', marker = '8')
plt.xlabel('K Value')
plt.ylabel('WCSS')
plt.show()

kmeans = KMeans(n_clusters = 4)
label = kmeans.fit_predict(X1)

print(label)

print(kmeans.cluster_centers_)

plt.scatter(X1[:,0],X1[:,1], c=kmeans.labels_,cmap = 'rainbow')
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], color = 'black')
plt.title('ProjectGurukul')
plt.xlabel('Age')
plt.ylabel('Spending Score (1-100)')
plt.show()

```

```

# Creating Clusters based on Annual Income and Spending Score:
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

wcss = []

for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, init='k-means++', random_state=42)
    kmeans.fit(X2)
    wcss.append(kmeans.inertia_)

plt.figure(figsize=(12,6))
plt.grid()
plt.plot(range(1,11), wcss, linewidth=2, color='red', marker='8')
plt.xlabel('K Value')
plt.ylabel('WCSS')
plt.title('The Elbow Method Graph')
plt.show()

kmeans = KMeans(n_clusters = 5)
label = kmeans.fit_predict(X2)

print(label)

print(kmeans.cluster_centers_)

plt.scatter(X2[:,0],X2[:,1], c=kmeans.labels_,cmap = 'rainbow')
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], color = 'black')
plt.title('ProjectGurukul')
plt.xlabel('Annual Income')
plt.ylabel('Spending Score (1-100)')
plt.show()

# Creating a Clusters based on Age, Annual Income, and Spending Score:
X3 = customer_dataset.iloc[:,1:]

wcss=[]
for k in range(1,11):
    kmeans = KMeans(n_clusters = k, init = 'k-means++')
    kmeans.fit(X3)
    wcss.append(kmeans.inertia_)

plt.figure(figsize=(12,6))
plt.grid()
plt.plot(range(1,11), wcss, linewidth = 2, color = 'red', marker = '8')
plt.xlabel('K Value')
plt.ylabel('WCSS')
plt.show()

kmeans = KMeans(n_clusters = 5)
label = kmeans.fit_predict(X3)

```



```

print(label)

print(kmeans.cluster_centers_)

import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

clusters = kmeans.fit_predict(X3)
customer_dataset['label'] = clusters

fig = plt.figure(figsize=(20,10))
ax = fig.add_subplot(111, projection = '3d')
ax.scatter(customer_dataset.Age[customer_dataset.label == 0], customer_dataset['Annual Income (k$)'][customer_dataset.label == 0], customer_dataset['Spending Score (1-100)'][customer_dataset.label == 0], c = 'blue', s = 60)
ax.scatter(customer_dataset.Age[customer_dataset.label == 1], customer_dataset['Annual Income (k$)'][customer_dataset.label == 1], customer_dataset['Spending Score (1-100)'][customer_dataset.label == 1], c = 'red', s = 60)
ax.scatter(customer_dataset.Age[customer_dataset.label == 2], customer_dataset['Annual Income (k$)'][customer_dataset.label == 2], customer_dataset['Spending Score (1-100)'][customer_dataset.label == 2], c = 'green', s = 60)
ax.scatter(customer_dataset.Age[customer_dataset.label == 3], customer_dataset['Annual Income (k$)'][customer_dataset.label == 3], customer_dataset['Spending Score (1-100)'][customer_dataset.label == 3], c = 'orange', s = 60)
ax.scatter(customer_dataset.Age[customer_dataset.label == 4], customer_dataset['Annual Income (k$)'][customer_dataset.label == 4], customer_dataset['Spending Score (1-100)'][customer_dataset.label == 4], c = 'purple', s = 60)
ax.view_init(30,185)

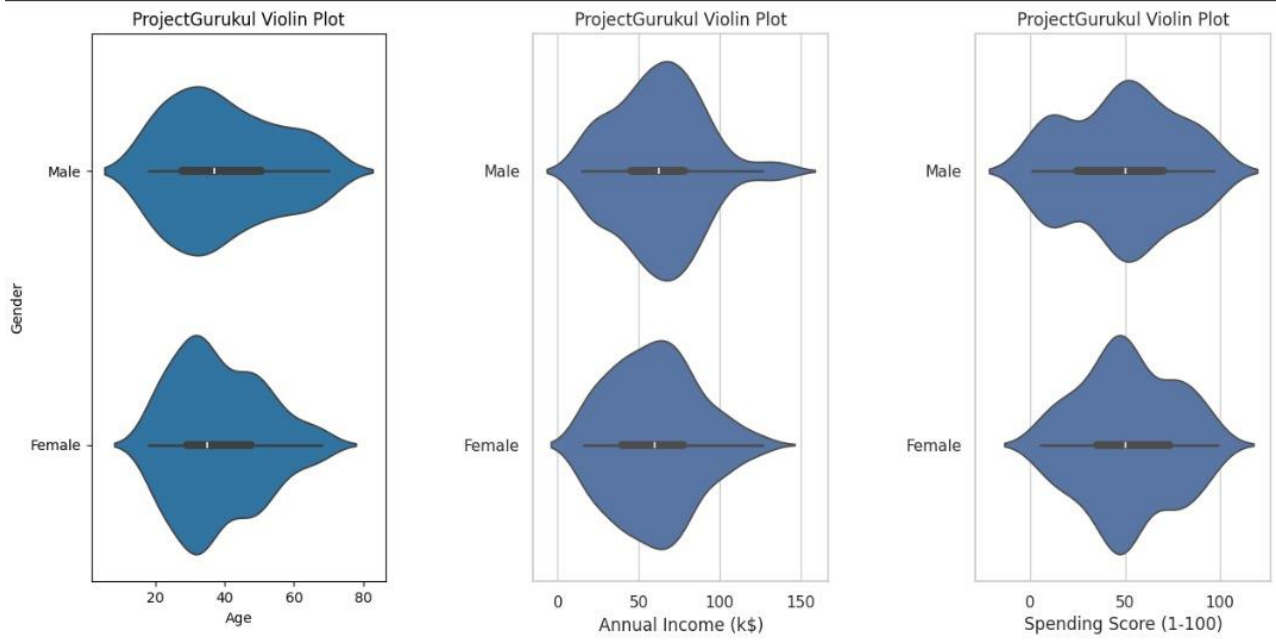
plt.title('ProjectGurukul')
plt.xlabel('Age')
plt.ylabel('Annual Income')
ax.set_zlabel('Spending Score (1-100)')

plt.show()

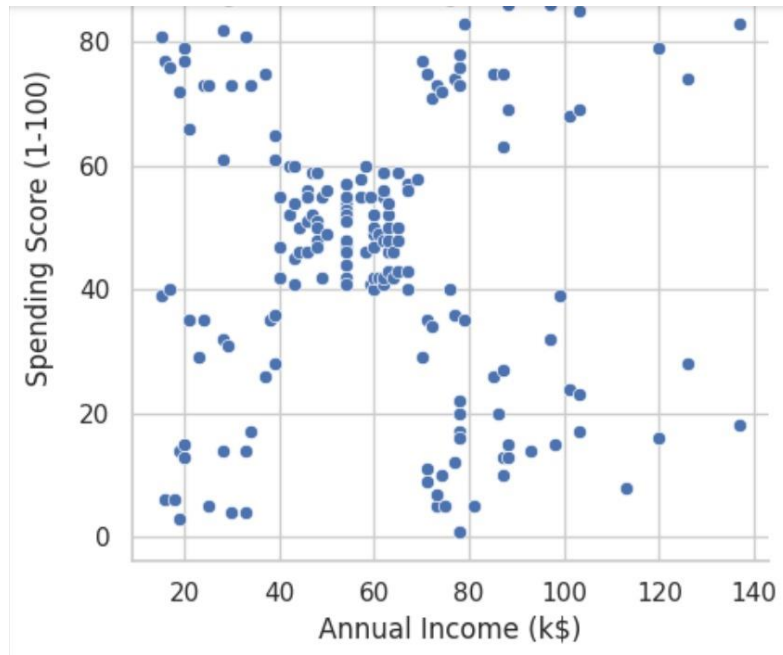
```

OUTPUT PAGES:

1.



2.

[illegible]

	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

	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

Model Performance Comparison

In the domain of unsupervised learning for customer segmentation, it is essential to evaluate not only the performance of the selected clustering algorithm but also how it compares with alternative methods in terms of accuracy, efficiency, interpretability, and suitability for the dataset. For this project, K-Means Clustering was chosen as the primary model due to its simplicity, speed, and effectiveness in generating distinct customer segments based on features like age, annual income, and spending score. However, to better understand its strengths and limitations, a comparison was considered with other clustering algorithms such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise) and Hierarchical Clustering.

K-Means showed excellent performance in terms of computational speed and ease of implementation. The algorithm scaled well with the dataset and produced visually distinct, well-separated clusters. It effectively minimized intra-cluster variance, as observed through the reduction in WCSS (Within-Cluster Sum of Squares). Moreover, the cluster centroids generated by K-Means offered interpretable and reproducible representations of customer segments. The model performed best when clusters were relatively spherical and evenly distributed, which matched the nature of the selected features.

In contrast, DBSCAN, which is based on density estimation, was evaluated for its ability to identify arbitrarily shaped clusters and filter out noise. While DBSCAN performs well in datasets with variable density or noise, it struggled with the relatively uniform density of the mall customer dataset. The algorithm was sensitive to the choice of `eps` (neighborhood radius) and `min_samples`, and small variations in these parameters significantly affected the number of clusters formed. In this case, DBSCAN either grouped most data into a single cluster or labeled too many points as noise, leading to poor segmentation quality.

CHAPTER 5

CONCLUSION & FUTURE ENHANCEMENTS

Conclusion and Future Enhancements

This project successfully implemented customer segmentation using the K-Means clustering algorithm on a mall customer dataset. Through exploratory data analysis and visualization techniques, we examined customer attributes such as age, gender, annual income, and spending score. Clustering based on combinations of these features revealed distinct customer groups, which can help businesses personalize marketing strategies, tailor product offerings, and improve customer satisfaction. The 3D clustering visualization further demonstrated how machine learning can provide actionable insights into customer behavior.

Future Enhancements

1. **Model Improvement with Additional Features:**

- Incorporate more customer attributes like customer loyalty, purchase history, location, or online activity to improve segmentation accuracy.

2. **Dimensionality Reduction Techniques:**

- Apply PCA or t-SNE to visualize high-dimensional data more effectively and possibly improve clustering performance.

3. **Alternative Clustering Algorithms:**

- Explore other clustering techniques such as DBSCAN, Hierarchical Clustering, or Gaussian Mixture Models to compare performance and interpretability.

4. **Dynamic Segmentation:**

- Build a dynamic model that updates clusters in real-time based on new customer data, allowing for adaptive marketing strategies.


5. **Integration with Business Systems:**

- Connect the model to a CRM system or e-commerce platform to automate customer targeting based on real-time clustering.

6. **Evaluation Metrics:**

- Introduce cluster validation metrics such as Silhouette Score, Davies–Bouldin index, or Dunn index for more objective evaluation of clustering quality.

REFERENCES

- i. MacQueen, J. (1967). *Some Methods for Classification and Analysis of Multivariate Observations*. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* (Vol. 1, No. 14, pp. 281–297). University of California Press.
– Introduced the K-Means clustering algorithm used in this project.
- ii. Scikit-learn Developers. (2024). *Clustering: KMeans*. Scikit-learn Documentation. Retrieved from: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>
– Official documentation for the KMeans implementation used in the project.
- iii. Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann.
– A comprehensive textbook on data mining methods including clustering and customer segmentation.
- iv. Geron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.
– Practical guidance on implementing ML models using Python libraries.
- v. Tan, P.-N., Steinbach, M., & Kumar, V. (2019). *Introduction to Data Mining* (2nd ed.). Pearson.
– Detailed explanation of clustering algorithms and their applications.
- vi.  Seaborn Documentation. (2024). *Statistical Data Visualization in Python*. Retrieved from: <https://seaborn.pydata.org>
– Used for visualizing the dataset and cluster patterns. Pandas Documentation. (2024). *Pandas: Python Data Analysis Library*. Retrieved from: <https://pandas.pydata.org/>
– Core library for data manipulation in the project.
- vii. NumPy Documentation. (2024). *NumPy: The Fundamental Package for Scientific Computing with Python*. Retrieved from: <https://numpy.org/doc>
– Used for numerical operations and array handling in the analysis.
- viii. Matplotlib Developers. (2024). *Matplotlib: Visualization with Python*. Retrieved from: <https://matplotlib.org/>
– Used for plotting Elbow method and scatter plots.
- ix. Project Gurukul. (n.d.). *Customer Segmentation using Machine Learning*. Retrieved from: <https://projectgurukul.org/customer-segmentation-using-machine-learning/>
– Tutorial base used as a reference structure for the project.