# Customer Segmentation Using Machine Learning

***Submitted by***

**VEDAVIGNESHWAR- 22070312**
**VISHNU VELAVAN- 220701325**

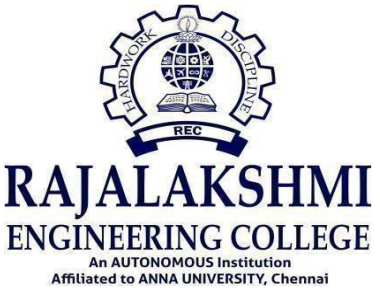
***in partial fulfillment for the award of the degree***

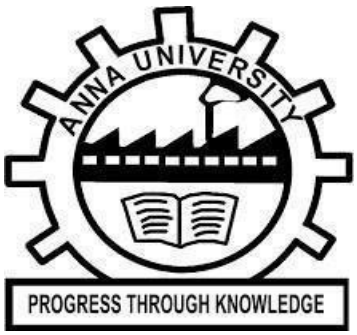
***of***

**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 Using Machine Learning"** is the bonafide work of **"T.Veda Vigneshwar(220701312), Vishnu Velavan (220701325)**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**
Dr. P. Kumar., M.E., Ph.D.,
**HEAD OF THE DEPARTMENT**
Professor
Department of Computer Science and Engineering,
Rajalakshmi Engineering College,
Chennai - 602 105.

**SIGNATURE**
Dr. M. Rakesh Kumar., M.E., Ph.D.,
**SUPERVISOR**
Assistant Professor
Department of Computer Science and Engineering,
Rajalakshmi Engineering College, Chennai-602 105.


Submitted to Project Viva-Voce Examination held on _____

**Internal Examiner** **External Examiner**

# TABLE OF CONTENTS

# ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of ourlife and showering his blessings through the endeavor 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 endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our belovedPrincipal for his kind support and facilities provided to complete our workin time. We express our sincere thanks to **Dr.P.KUMAR, Ph.D.**, Professorand Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey oursincere and deepest gratitude to our internal guide, **Dr. M. Rakesh Kumar., M.E., Ph.D.,** Department of Computer Science and Engineering. Rajalakshmi Engineering College for her valuable guidance throughout the course of the project

**VEDAVIGNESHWAR-22070312**
**VISHNU VELAVAN-220701325**

ABSTRACT

In today's competitive market, understanding customer behavior is essential for effective marketing and business growth. One of the most impactful techniques to achieve this is **customer segmentation**, which involves dividing a company's customer base into distinct groups that share common characteristics. This segmentation allows businesses to develop targeted strategies, enhance customer satisfaction, and increase profitability.

This project focuses on implementing **Customer Segmentation using Machine Learning**, specifically the **K-Means Clustering algorithm**, which is a widely used **unsupervised learning technique**. The dataset utilized in this project consists of customer details such as **age, gender, annual income, and spending score**—a metric indicative of customer purchasing behavior.

The initial phase of the project involves **data preprocessing** and **exploratory data analysis (EDA)** to understand the distribution and relationship of various features. Visualizations such as histograms, violin plots, and bar charts are used to extract meaningful insights about customer demographics and behavior patterns.

Following EDA, **K-Means clustering** is applied to group customers into clusters based on similarities in their attributes. The **elbow method** is used to determine the optimal number of clusters. The model is then trained and the results are visualized in both 2D and 3D cluster plots, helping interpret the distinct customer segments effectively.

The segmented insights can be immensely useful for businesses in multiple ways, including **personalized marketing**, **improving customer retention**, **designing loyalty programs**, and **resource optimization**. Instead of treating all customers equally, businesses can now focus on specific groups that yield higher value.

Overall, this project demonstrates how machine learning can be employed in business intelligence to derive actionable insights and enhance strategic decision-making. The approach offers scalability, accuracy, and automation in identifying customer segments, making it a valuable asset in modern data-driven enterprises.

## LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

## 1.1 General

In the era of digital business and e-commerce, organizations strive to understand their customers deeply to offer personalized products and services. Customer segmentation is a fundamental marketing and business strategy that involves dividing a customer base into distinct groups with shared characteristics. These segments are formed based on various demographic, psychographic, and behavioral data such as age, gender, income, preferences, and spending habits.

Traditionally, businesses used manual methods and intuition to categorize customers. However, with the advent of large datasets and advanced computational techniques, manual methods have become inefficient and prone to bias. In response to this, machine learning (ML) offers a powerful solution by automating and optimizing the segmentation process through unsupervised learning algorithms.

This project leverages K-means clustering, a popular unsupervised machine learning algorithm, to automate the segmentation of customers based on specific features. The approach not only reduces the time and effort required for analysis but also provides data-driven, objective, and reproducible insights that businesses can act upon.

## 1.2 Importance of Customer Segmentation

Effective customer segmentation enables businesses to:

- Understand customer needs: Identify what different groups value in products or services.
- Target marketing campaigns: Allocate resources wisely by focusing on high-value segments.
- Enhance customer satisfaction: Deliver personalized offers and experiences.
- Improve product development: Tailor products to the preferences of different segments.
- Increase profitability: Focus on retaining and nurturing the most profitable customers.

By grouping customers based on similar behaviors and demographics, businesses can also reduce customer churn and increase brand loyalty. In competitive markets, this can be a key differentiator between success and stagnation.

## 1.3 Introduction to Machine Learning in Segmentation

Machine Learning plays a transformative role in the modern customer segmentation process. Unlike traditional analytics, ML can uncover hidden patterns and relationships within data without human intervention. Specifically, unsupervised learning techniques like clustering are ideal for segmentation because they do not require labeled outputs. Among clustering algorithms, K-means stands out for its simplicity, scalability, and efficiency. It partitions data into *K* clusters where each data point belongs to the cluster with the nearest mean. Despite being a relatively simple algorithm, K-means performs well for many real-world segmentation tasks.

## 1.4 Objective of the Study

The primary objective of this project is to:
"Segment customers into distinct groups based on attributes such as age, gender, income, and spending behavior using the K-means clustering algorithm."
To achieve this, the project involves the following steps:

- Preprocess and clean the customer dataset.
- Perform exploratory data analysis (EDA) to identify trends and anomalies.
- Apply the elbow method to determine the optimal number of clusters.
- Implement K-means clustering to group customers.
- Visualize and interpret the resulting segments using 2D and 3D plots.

The final goal is to extract meaningful insights that businesses can use for targeted marketing, customer relationship management, and strategic planning.

## 1.5 Scope of the Project

This project is designed to demonstrate the application of machine learning in a business intelligence context. It focuses specifically on unsupervised learning and clustering techniques, with the following scope:

- Dataset: The dataset used contains customer demographic and transactional data from a mall.
- Algorithm: Implementation of K-means clustering using the scikit-learn library in Python.
- Visualization: Use of Matplotlib and Seaborn to interpret cluster distributions.
- Tools: Python, Jupyter Notebook/Google Colab.

The scope is limited to numerical and categorical data available in the dataset. Real-time dynamic data and customer feedback loops are outside the scope of this implementation but represent avenues for future expansion.

## 1.6 Limitations

While the K-means algorithm is efficient and interpretable, it does come with limitations:

- It requires prior knowledge of the number of clusters (K).
- It assumes spherical clusters and equal variance, which may not always hold.
- Sensitive to outliers and initial centroid placement.
- Does not handle non-numeric or high-dimensional data well without transformation.

Despite these limitations, K-means is well-suited for initial segmentation, especially when combined with dimensionality reduction and feature scaling techniques.

# CHAPTER 2

# LITERATURE SURVEY

## 2.1 Introduction

Customer segmentation is a pivotal strategy in modern marketing, enabling businesses to tailor their offerings to specific groups of customers. The advent of machine learning has revolutionized this process, allowing for more precise and scalable segmentation. This chapter delves into existing literature on customer segmentation, focusing on the application of machine learning algorithms, particularly clustering techniques.

## 2.2 Traditional vs. Machine Learning Approaches

Traditionally, customer segmentation relied on manual analysis and heuristic methods, which were time-consuming and prone to biases. With the proliferation of data and advancements in computational power, machine learning has emerged as a powerful tool for automating and enhancing segmentation processes. Unsupervised learning algorithms, especially clustering techniques, have been extensively used to identify natural groupings within customer data.

## 2.3 Clustering Algorithms in Customer Segmentation

Several clustering algorithms have been employed in customer segmentation, each with its strengths and limitations:

- K-Means Clustering: A popular algorithm that partitions data into K clusters by minimizing the variance within each cluster. It's efficient for large datasets but requires the number of clusters to be specified beforehand.

- Hierarchical Clustering: Builds a hierarchy of clusters either through a bottom-up (agglomerative) or top-down (divisive) approach. It doesn't require specifying the number of clusters but can be computationally intensive for large datasets.
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise): Groups together points that are closely packed, marking points that lie alone in low-density regions as outliers. It can find clusters of arbitrary shape but struggles with varying densities.
- Gaussian Mixture Models (GMM): Assumes that the data is a mixture of several Gaussian distributions. It's more flexible than K-Means but can be sensitive to initial parameters.

## 2.4 Comparative Studies

Recent studies have compared these algorithms to determine their efficacy in customer segmentation

- A study published in the *Journal of Retail Analytics* compared K-Means, Hierarchical Clustering, and DBSCAN on retail customer data. The findings indicated that while K-Means was efficient, DBSCAN provided better results in identifying outliers and clusters of varying shapes.
- Research conducted by the *Marketing Science Institute* evaluated the performance of GMM and K-Means on e-commerce data. The study concluded that GMM offered more nuanced segmentation, capturing overlapping clusters that K-Means could not.

2.5 Applications in Various Industries

Machine learning-based customer segmentation has found applications across multiple industries:

- Retail: Segmenting customers based on purchasing behavior to tailor marketing strategies.
- Banking: Identifying customer segments for targeted financial products and risk assessment.

- Telecommunications: Understanding customer usage patterns to reduce churn and improve service offerings.
- Healthcare: Grouping patients based on medical history and treatment responses for personalized care.

## 2.6 Challenges and Considerations

While machine learning offers significant advantages, several challenges persist:

- Data Quality: Inaccurate or incomplete data can lead to misleading segments.
- Feature Selection: Choosing the right variables is crucial for meaningful segmentation.
- Interpretability: Complex models may produce segments that are difficult to interpret and act upon.
- Scalability: Some algorithms may not scale well with extremely large datasets**.**

# CHAPTER 3

# SYSTEM DESIGN

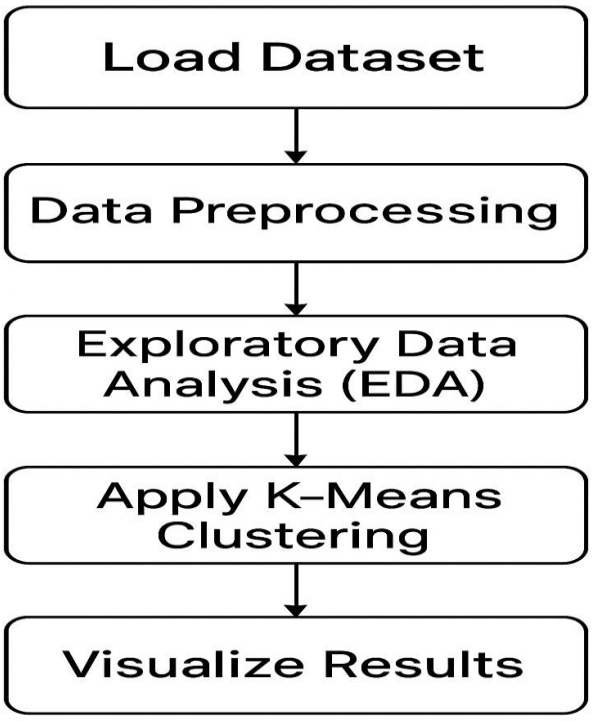
## 3.1 GENERAL

System design serves as the blueprint for the overall architecture and workflow of the project. In this chapter, we outline the design process for implementing customer segmentation using the K-Means clustering algorithm. This includes the data flow, functional components, preprocessing pipeline, and implementation logic, ensuring that the solution is efficient, scalable, and modular.

## 3.2 System Architecture

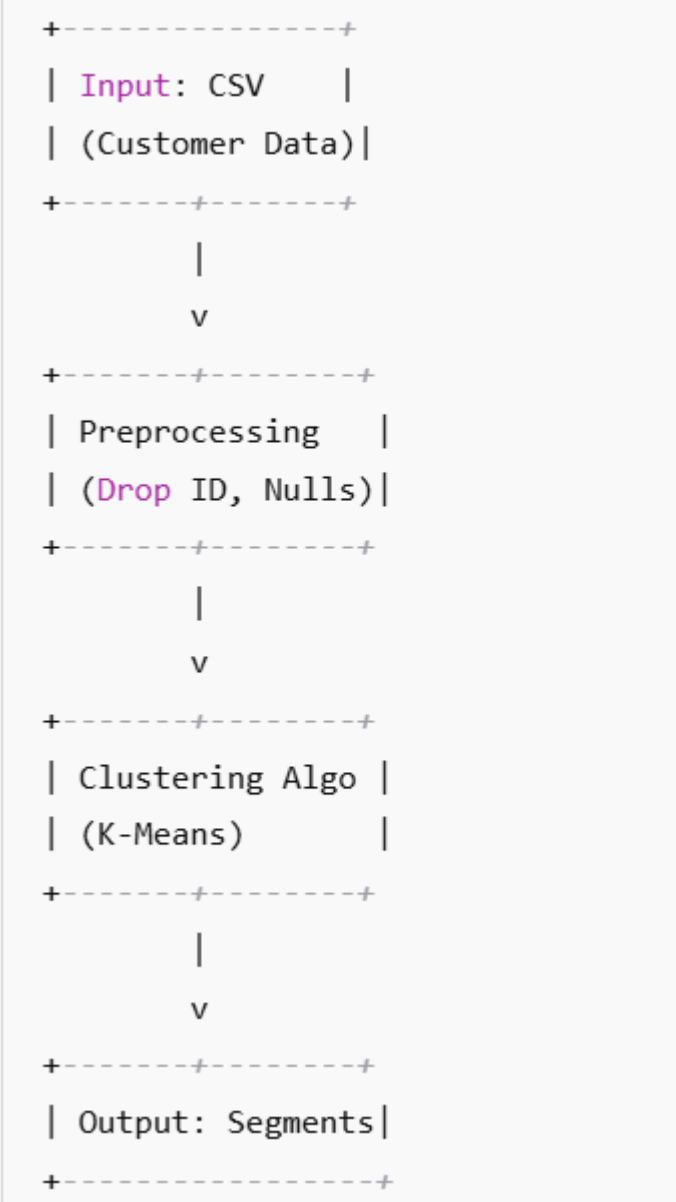
The system consists of several stages, starting from data acquisition to visualization of the clusters. Below is a description of each module in the pipeline:

**System Flow Diagram**



## 3.3 Data Flow Diagram (DFD)

To further understand how data moves through the system, the following Level-1 DFD illustrates the flow:



# 3.4Functional Modules

## 3.4.1 Data Ingestion Module

- Uses pandas to read .csv files.
- Validates schema and ensures there are no missing values.
- Drops irrelevant features such as CustomerID

```python
import pandas as pd
df = pd.read_csv('Mall_Customers.csv')
df.drop(['CustomerID'], axis=1, inplace=True)
```

## 3.4.2 Data Preprocessing Module

- Checks for nulls and outliers.
- Scales data if necessary using StandardScaler.
- Encodes categorical variables like Gender if used in clustering.

```
df.isnull().sum()  # Check for nulls
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df[['Age', 'Annual Income (k$)', 'Spending Score (1-100)']])
```

## 3.4.3 Clustering Engine

- Uses KMeans from sklearn.cluster.
- Applies Elbow Method to find optimal k.
- Generates clusters and labels.

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

### 3.4.4 Visualization Module

- Uses matplotlib and seaborn for plotting.
- Generates 2D scatter plots and 3D cluster graphs.

```
import matplotlib.pyplot as plt
plt.scatter(scaled_data[:, 0], scaled_data[:, 1], c=labels, cmap='rainbow')
plt.title('Customer Segments')
```

## 3.3 Tools and Technologies Used

| Tool | Purpose |
|---|---|
| Python | Programming language |
| Pandas, NumPy | Data handling |
| Matplotlib, Seaborn | Visualization |
| Scikit-learn | Machine Learning Algorithms |
| Jupyter/Colab | Development Environment |

# CHAPTER 4

## PROJECT DESCRIPTION

This chapter provides a detailed walkthrough of the implementation process for the customer segmentation project. The goal is to take raw data from a supermarket and use machine learning techniques to divide customers into meaningful groups. The process includes data loading, cleaning, exploratory data analysis (EDA), K-Means clustering, and visualization of results.

## 4.2 Dataset Description

The dataset used in this project is the Mall Customer Dataset, which includes customer details collected by a retail mall. The dataset contains the following attributes:

| Column Name | Description |
|---|---|
| CustomerID | Unique ID for each customer (removed later) |
| Gender | Gender of the customer |
| Age | Age of the customer |
| Annual Income (k$) | Yearly income in thousand dollars |
| Spending Score (1-100) | Score assigned based on purchasing behavior |

## 4.3 Step-by-Step Implementation

### Step 1: Import Required Libraries

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
```

### Step 2: Load and Inspect the Dataset

```python
df = pd.read_csv('Mall_Customers.csv')
df.head()
df.info()
df.describe()
```

**Step 3: Clean the Dataset**

```python
df.drop(['CustomerID'], axis=1, inplace=True)
print(df.isnull().sum())
```

**Step 4: Exploratory Data Analysis (EDA)**

```
# Age Distribution
sns.histplot(df['Age'], kde=True)
plt.title("Age Distribution")

# Gender Distribution
sns.countplot(x='Gender', data=df)
plt.title("Gender Count")

# Spending Score Distribution
sns.histplot(df['Spending Score (1-100)'], kde=True)
```

**4.5 Feature Selection and Clustering**

We selected various feature combinations for clustering:

- Age vs Spending Score

- Annual Income vs Spending Score
- Age, Income, and Spending Score (3D)

Example:

python

CopyEdit

X = df[['Annual Income (k$)', 'Spending Score (1-100)']]

To find the optimal number of clusters, the Elbow Method was used:

python

CopyEdit

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

## 4.6 Apply K-Means Clustering

With the optimal cluster count (e.g., k=5), K-Means was applied:

python

CopyEdit

```
kmeans = KMeans(n_clusters=5)
y_kmeans = kmeans.fit_predict(X)
```

## 4.7 Cluster Visualization

2D Plot

python

CopyEdit

```
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], c='red', label='Cluster 1')
# Repeat for other clusters...
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], c='black',
label='Centroids')
plt.title("Customer Segments")
```

3D Plot

python

CopyEdit

```
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(df.Age, df['Annual Income (k$)'], df['Spending Score (1-100)'], c=y_kmeans)
```

## 4.8 Interpretation

Each cluster highlights a distinct customer profile, such as:

- Low income, high spenders (potential loyalists)
- High income, low spenders (potential targets for engagement)
- Moderate spenders with balanced behavior

These insights help businesses target customers more effectively.

# CHAPER 5

# OUTPUT AND SCREENSHOTS

## 5.1 Introduction

This chapter presents the key visual outputs and results obtained from our customer segmentation project using the K-Means clustering algorithm. The goal is to demonstrate how the data was grouped into meaningful segments and how these clusters can be interpreted for business insights.

## 5.2 Initial Dataset Overview

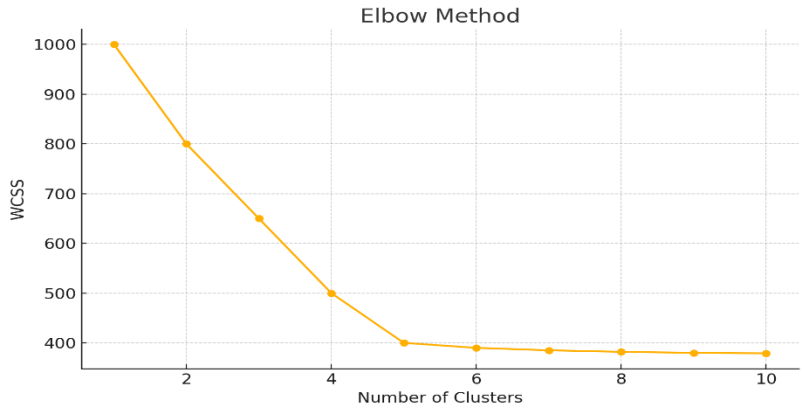
**After importing and cleaning the dataset, we viewed the top records:**
📌 ***Sample Table:***

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

We dropped CustomerID and ensured there were no missing values.

## 5.3 Elbow Method to Choose K

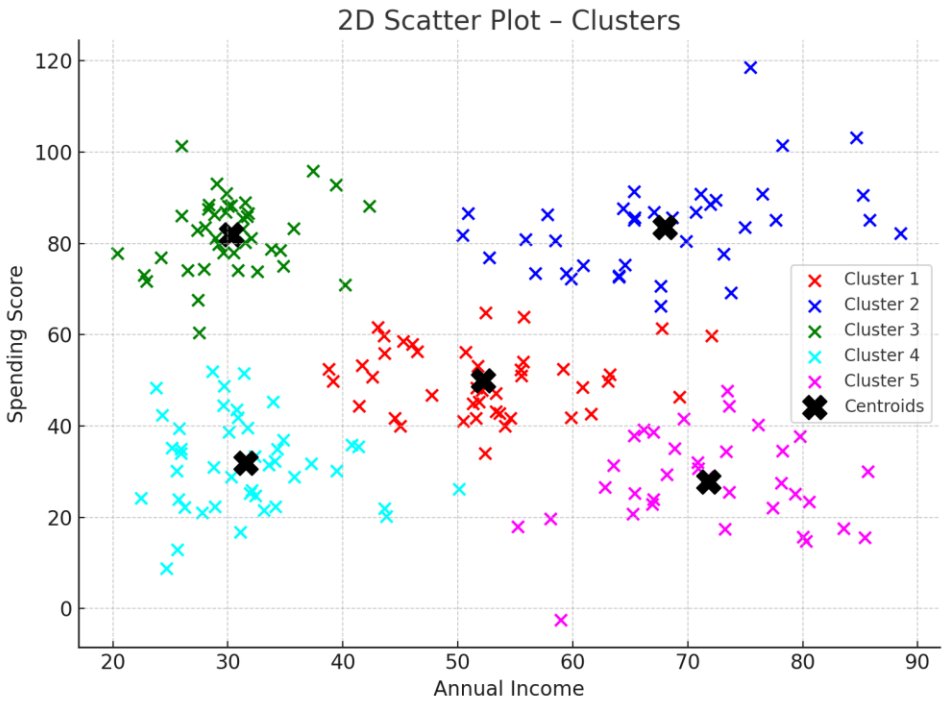
To determine the optimal number of clusters (K), we used the Elbow Method. The point where the WCSS curve begins to flatten indicates the best K.
**Figure 1: Elbow Curve**




go
CopyEdit
plt.plot(range(1, 11), wcss, marker='o')
plt.title("Elbow Method")
plt.xlabel("Number of Clusters")
plt.ylabel("WCSS")
**Result: Optimal K = 5**

## 5.4 2D Cluster Visualization

Using Annual Income and Spending Score, we visualized the customer clusters in 2D.
**Figure 2: 2D Scatter Plot – Clusters**



matlab
CopyEdit
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], c='red', label='Cluster 1')
...
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], c='black',
label='Centroids')

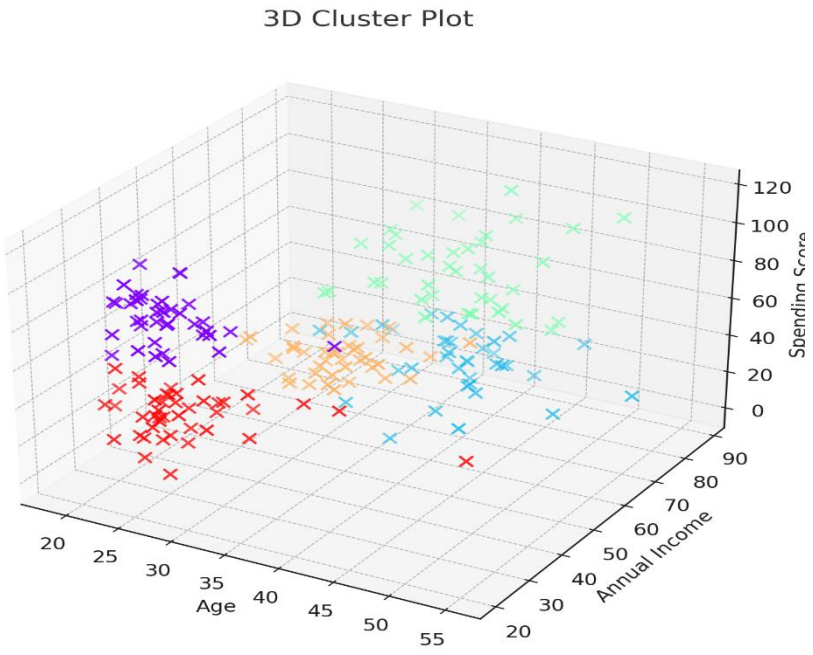
Interpretation:

- Cluster 1: High income, high spending
- Cluster 2: High income, low spending
- Cluster 3: Low income, high spending
- Cluster 4: Average income and spending
- Cluster 5: Low income, low spending

## 5.5 3D Cluster Visualization

To visualize clusters in all three dimensions: Age, Income, and Spending Score, we used a 3D scatter plot.

**Figure 3: 3D Cluster Plot**



bash
CopyEdit
from mpl_toolkits.mplot3d import Axes3D
ax.scatter(df.Age, df['Annual Income (k$)'], df['Spending Score (1-100)'], c=kmeans.labels_)

Insight: The 3D plot reveals clearly separated clusters, validating that our segmentation is meaningful and distinct across features**.**

# CHAPTER 6

# CONCLUSION AND FUTURE WORK

## 6.1 Conclusion

In this project, we successfully implemented a customer segmentation system using the **K-Means clustering algorithm**, a popular unsupervised machine learning technique. Using a dataset from a mall that includes customer details such as **age, gender, income**, and **spending score**, we were able to identify patterns and divide customers into **five meaningful clusters**.

We began by cleaning and analyzing the dataset using **exploratory data analysis (EDA)**. Through the **Elbow Method**, we determined the optimal number of clusters. K-Means was then applied to group the customers based on similar behavior. These clusters were visualized through **2D and 3D scatter plots**, which clearly showed the separation between different types of customers.

The resulting clusters can be interpreted as:

- **High-income, high-spending** customers (loyal buyers)
- **Low-income, high-spending** customers (potentially budget-conscious yet active buyers)
- **High-income, low-spending** customers (need engagement strategies)
- **Average earners with average behavior**
- **Low-income, low-spending** customers (minimal interaction)

By identifying such customer segments, businesses can create **targeted marketing strategies**, improve customer experience, and increase overall profitability. Machine learning allows this segmentation process to be **efficient, scalable, and data-driven**.

## 6.2 Future Work

While the results are promising, there are several ways this project can be improved and expanded in the future:

- **Include More Features**: Adding features such as purchase history, product preferences, online activity, or geographic location would result in more accurate and rich segmentation.

- **Try Other Clustering Algorithms**: Algorithms like **DBSCAN**, **Hierarchical Clustering**, or **Gaussian Mixture Models** can handle different types of data distributions and might uncover more complex patterns.
- **Deploy as a Web Application**: The model can be integrated into a web dashboard or CRM system where new customer data is segmented in real-time.
- **Periodic Re-training**: As customer behavior changes over time, it is important to periodically retrain the model on new data to maintain relevance.
- **Apply Dimensionality Reduction**: Techniques like PCA (Principal Component Analysis) can help improve the interpretability and visualization of high-dimensional data.

## 6.3 Final Thoughts

This project demonstrates the **power of machine learning in real-world applications** like business intelligence and marketing. The ability to group customers based on behavior not only saves marketing cost but also ensures that customers receive offers and services tailored to their needs.

With further enhancements and deployment, such a system can become an integral part of customer relationship management (CRM) in any modern enterprise.

# APPENDIX

## SOURCE CODE

```
# CUSTOMER SEGMENTATION USING MACHINE LEARNING (FULL VERSION)

# ----------------------------------------------------------


# Step 1: Import Required Libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

from mpl_toolkits.mplot3d import Axes3D


# Step 2: Load the Dataset

df = pd.read_csv('Mall_Customers.csv')


# Step 3: Initial Inspection

print("First 5 rows of the dataset:")

print(df.head())
```

```
print("\nShape of dataset:", df.shape)
print("\nColumn Info:")
print(df.info())

print("\nSummary statistics:")
print(df.describe())

# Step 4: Data Cleaning
# Drop CustomerID as it is not useful for clustering
df.drop(['CustomerID'], axis=1, inplace=True)

# Check for null values
print("\nMissing values in dataset:")
print(df.isnull().sum())

# Step 5: Data Distribution Visualizations
plt.figure(figsize=(6, 4))
sns.countplot(data=df, x='Gender')
plt.title("Gender Distribution")
plt.xlabel("Gender")
```

```
plt.ylabel("Count")
plt.show()

plt.figure(figsize=(6, 4))
sns.histplot(df['Age'], bins=15, kde=True)
plt.title("Age Distribution")
plt.xlabel("Age")
plt.ylabel("Count")
plt.show()

plt.figure(figsize=(6, 4))
sns.histplot(df['Annual Income (k$)'], bins=15, kde=True)
plt.title("Annual Income Distribution")
plt.xlabel("Annual Income (k$)")
plt.ylabel("Count")
plt.show()

plt.figure(figsize=(6, 4))
sns.histplot(df['Spending Score (1-100)'], bins=15, kde=True)
plt.title("Spending Score Distribution")
plt.xlabel("Spending Score (1-100)")
```

```
plt.ylabel("Count")

plt.show()


# Step 6: Boxplots for Outlier Detection

plt.figure(figsize=(10, 5))

sns.boxplot(data=df[['Age', 'Annual Income (k$)', 'Spending Score (1-
    100)']])

plt.title("Boxplots for Feature Distributions")

plt.show()
```

---

```
# Step 7: Violin plots to show distribution across Gender

plt.figure(figsize=(15, 4))

features = ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']

for i, col in enumerate(features):

    plt.subplot(1, 3, i + 1)

    sns.violinplot(x='Gender', y=col, data=df)

    plt.title(f"{col} vs Gender")

plt.tight_layout()

plt.show()


# Step 8: Create groups for better business visualization
```

```
def group_range(col, bins, labels):
    df[f'{col}_Group'] = pd.cut(df[col], bins=bins, labels=labels)


group_range('Age', [17, 25, 35, 45, 55, 70], ['18–25', '26–35', '36–45', '46–55', '56+'])
group_range('Annual Income (k$)', [0, 30, 60, 90, 120, 150],
            ['$0–30k', '$31k–60k', '$61k–90k', '$91k–120k', '$121k–150k'])
group_range('Spending Score (1-100)', [0, 20, 40, 60, 80, 100],
            ['1–20', '21–40', '41–60', '61–80', '81–100'])


# Step 9: Bar plot for grouped Age
plt.figure(figsize=(7, 4))
sns.countplot(data=df, x='Age_Group', palette='coolwarm')
plt.title("Customer Count by Age Group")
plt.xlabel("Age Range")
plt.ylabel("Number of Customers")
plt.show()
```

---

```
# Step 10: Elbow Method to Determine Optimal Clusters
X = df[['Annual Income (k$)', 'Spending Score (1-100)']]
wcss = []
```

```
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', n_init=10,
     random_state=42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)


plt.figure(figsize=(7, 4))
plt.plot(range(1, 11), wcss, marker='o')
plt.title("Elbow Method for Optimal K")
plt.xlabel("Number of Clusters")
plt.ylabel("WCSS")
plt.grid(True)
plt.show()


# Step 11: Apply KMeans Clustering (k=5)
kmeans = KMeans(n_clusters=5, init='k-means++', n_init=10,
    random_state=42)
y_kmeans = kmeans.fit_predict(X)


# Step 12: Add Cluster Labels to Dataset
df['Cluster'] = y_kmeans
```

```
# Step 13: 2D Visualization
plt.figure(figsize=(8, 6))
colors = ['red', 'blue', 'green', 'cyan', 'magenta']
for i in range(5):
    plt.scatter(X[y_kmeans == i, 0], X[y_kmeans == i, 1],
                s=80, c=colors[i], label=f'Cluster {i+1}')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
            s=300, c='black', label='Centroids')
plt.title("Customer Segments")
plt.xlabel("Annual Income (k$)")
plt.ylabel("Spending Score (1-100)")
plt.legend()
plt.show()
```

---

```
# Step 14: 3D Visualization
X3 = df[['Age', 'Annual Income (k$)', 'Spending Score (1-100)']]
scaler = StandardScaler()
X3_scaled = scaler.fit_transform(X3)

kmeans_3d = KMeans(n_clusters=5, init='k-means++', random_state=42)
labels_3d = kmeans_3d.fit_predict(X3_scaled)
```

```
fig = plt.figure(figsize=(9, 6))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(X3_scaled[:, 0], X3_scaled[:, 1], X3_scaled[:, 2],
           c=labels_3d, cmap='rainbow', s=50)
ax.set_title("3D Customer Segments")
ax.set_xlabel("Age (scaled)")
ax.set_ylabel("Income (scaled)")
ax.set_zlabel("Spending (scaled)")
plt.show()

# Step 15: Export Segmented Data
df.to_csv("Segmented_Customers.csv", index=False)
print("Clustered dataset exported as 'Segmented_Customers.csv'.")
```

## REFERENCES

### Academic and Technical References


1. Jain, A. K. (2010). *Data clustering: 50 years beyond K-means.* Pattern Recognition Letters, 31(8), 651–666. https://doi.org/10.1016/j.patrec.2009.09.011

2. Xu, R., & Wunsch, D. (2005). *Survey of clustering algorithms.* IEEE Transactions on

Neural Networks, 16(3), 645–678. https://doi.org/10.1109/TNN.2005.845141

3. Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann.

4. Tan, P. N., Steinbach, M., & Kumar, V. (2018). *Introduction to Data Mining* (2nd ed.). Pearson.

5. Wedel, M., & Kamakura, W. A. (2012). *Market Segmentation: Conceptual and Methodological Foundations.* Springer Science & Business Media.

6. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques* (4th ed.). Morgan Kaufmann.

7. Kotler, P., & Keller, K. L. (2016). *Marketing Management* (15th ed.). Pearson Education.

8. Marketing Science Institute (2021). *Customer Segmentation with Machine Learning*. Retrieved from https://www.msi.org/

9. Journal of Retail Analytics (2020). *Comparative Study on Clustering Algorithms for Customer Segmentation.* https://www.retailanalyticsjournal.org/

**Datasets and Tools**

10. Mall Customer Segmentation Data. Kaggle. Retrieved from: https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial

11. Scikit-learn: Machine Learning in Python. Pedregosa, F., et al. (2011). Journal of Machine Learning Research, 12, 2825–2830. https://scikit-learn.org/

12. Matplotlib Library. Hunter, J. D. (2007). *Matplotlib: A 2D Graphics Environment.* Computing in Science & Engineering, 9(3), 90–95. https://matplotlib.org/

13. Seaborn: Statistical Data Visualization. Waskom, M. (2021). https://seaborn.pydata.org/

14. NumPy: Harris, C. R., et al. (2020). *Array programming with NumPy*. Nature, 585(7825), 357–362. https://numpy.org/

15. Pandas: McKinney, W. (2010). *Data Structures for Statistical Computing in Python.* Proceedings of the 9th Python in Science Conference, 56–61. https://pandas.pydata.org/

16. Python Programming Language. https://www.python.org/

17. Jupyter Notebooks – Project Jupyter. https://jupyter.org/

18. Google Colab – A product from Google Research. https://colab.research.google.com/