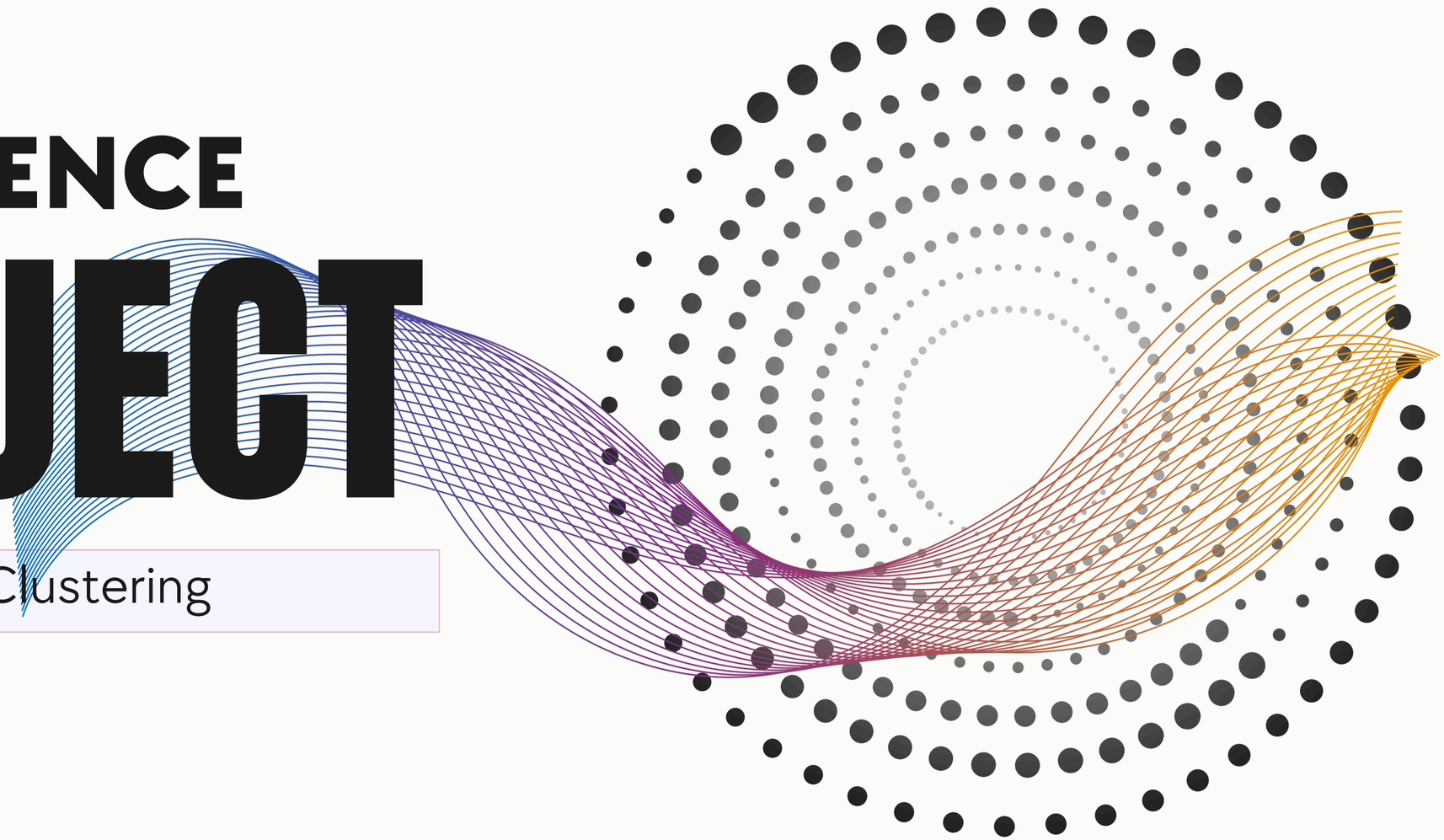
RAJALAKSHMI EDUVERSE

20-02-2025

DATA SCIENCE

PROJECT



Machine Learning - Clustering

Instructor - Gowthami Shyam

Presented by Keerthana Praveen

Acknowledgement

I would like to express my sincere gratitude to Raja Lakshmi Eduverse and Mrs. Gowthami Shyam, Instructor, for providing valuable guidance throughout this data science course. Their insights into machine learning and clustering techniques have been instrumental in shaping this project.

I would also like to acknowledge Kaggle for providing the dataset, which helped in applying clustering techniques to real-world customer segmentation.

Finally, I extend my appreciation to my mentors, peers, and online resources that contributed to my understanding of unsupervised learning and clustering algorithms.

TABLE
of contents

Introduction

Problem Statement

Understanding customer behavior is crucial for businesses to optimize marketing strategies, enhance customer experience, and increase sales. However, identifying distinct customer segments based on purchasing behavior and demographics is challenging. Traditional segmentation approaches may not capture complex patterns in customer data. This project aims to apply unsupervised learning techniques, specifically clustering algorithms, to segment customers based on key attributes such as age, income, and spending habits. These insights can help businesses tailor personalized marketing campaigns and improve customer retention.

Motivation

Customer segmentation is a vital tool in retail and marketing analytics. By grouping customers with similar behaviors, businesses can:

- Develop targeted marketing strategies to enhance customer engagement.
- Improve customer experience through personalized promotions and product recommendations.
- Optimize business strategies by understanding high-value customer groups.

Objectives

1 Data Exploration & Preprocessing

- Understand the structure of the dataset and handle any missing or inconsistent values.
- Perform necessary transformations to prepare the data for clustering.

2 Feature Selection & Engineering

- Identify key attributes such as age, annual income, and spending score that influence customer segmentation.
- Scale the features appropriately to ensure effective clustering.

3 Applying Clustering Algorithms

- Implement K-Means Clustering to identify distinct customer groups.
- Experiment with DBSCAN and Hierarchical Clustering to compare different clustering approaches.

4 Evaluation & Visualization

- Use elbow method and silhouette score to determine the optimal number of clusters.
- Visualize customer segments using scatter plots and cluster distributions for better interpretation.

5 Business Insights & Recommendations

- Interpret the characteristics of different customer groups.
- Provide actionable insights for businesses to enhance marketing strategies, optimize promotions, and improve customer engagement.

Data Description

Data Sources

The dataset used in this project is the Mall Customers Dataset, which is commonly available in open-source repositories. It contains customer information collected by a shopping mall to analyze customer behavior and spending patterns.

Data Characteristics

The dataset contains 200 entries and 5 features, including categorical (Gender) and continuous (Age, Annual Income, Spending Score) variables.

- Age: Ranges from 18 to 70, with a mean of 38.85.
- Annual Income (k$): Varies from 15k to 137k, with a mean of 60.56k.
- Spending Score (1-100): Ranges from 1 to 99, averaging 50.20.

The dataset is clean with no missing values, making it well-suited for clustering analysis.

Methodology

• Phase 1 - Apporach

Data Preprocessing

- Checked for outliers using boxplots.
- Removed the CustomerID column as it is not relevant for clustering.
- Standardized numerical features using StandardScaler for better clustering performance.

• Phase 2 - Algorithms Used

Model Selection & Training

- Applied K-Means clustering and used the Elbow Method to determine the optimal number of clusters.
- Trained the K-Means model using the optimal number of clusters.
-

Insights & Visualization

- Analyzed the characteristics of each cluster.
- Focused on identifying high-spending customer segments for business insights.

Evaluation Metrics

Silhouette Score

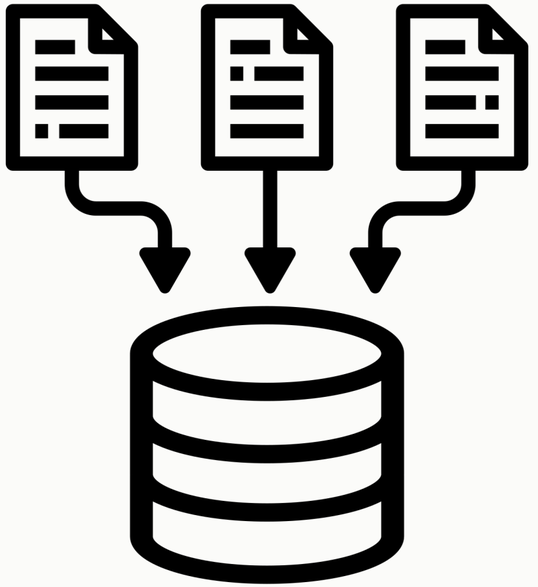
- Measures how similar a data point is to its own cluster compared to other clusters.
- A higher score (closer to 1) indicates well-defined clusters, while a lower score (closer to -1) suggests overlapping clusters.

Davies-Bouldin Score

- Computes the average similarity ratio between each cluster and its most similar cluster.
- Lower values indicate better clustering with well-separated clusters

Calinski-Harabasz Index (Variance Ratio Criterion)

- Evaluates the ratio of intra-cluster and inter-cluster dispersion.
- Higher values indicate better-defined clusters.



Expected Outcomes

1.With this project, we anticipate achieving the following:
2.Customer Segmentation
3.Identifying distinct customer groups based on spending behavior and income levels.
4.Targeted Marketing Strategies
5.Helping businesses tailor promotions and offers for high-spending and budget-conscious customers.
6.Improved Customer Insights
7.Gaining a deeper understanding of customer demographics and spending habits for better business decision-making.
8.Enhanced Customer Experience
9.Personalizing services based on cluster characteristics to improve customer satisfaction and loyalty.

Conclusion

This project successfully applied K-Means clustering to segment mall customers based on age, annual income, and spending score. The key findings include:

- Customers were grouped into distinct segments, highlighting differences in spending behavior and income levels.
- High-spending clusters were identified, which can help businesses target premium customers with personalized offers.
- The Elbow Method and Silhouette Score were used to determine the optimal number of clusters, ensuring effective segmentation.
- Insights from clustering can be leveraged to enhance marketing strategies, improve customer engagement, and drive business growth.

This segmentation approach provides a data-driven foundation for businesses to optimize their marketing and customer relationship strategies.

Code

Clustering Project - Mall customers segmentation

[1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
#Load dataset
df = pd.read_csv("Mall_Customers.csv")
df.head()
```

[1]:

	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

[2]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   CustomerID              200 non-null    int64 
 1   Gender                  200 non-null    object
 2   Age                     200 non-null    int64 
 3   Annual Income (k$)      200 non-null    int64 
 4   Spending Score (1-100)  200 non-null    int64 
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

[3]:
```
df.describe()
```

[3]:

	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

[4]:
```
#Checking for missing values
df.isnull().sum()
```

[4]:
```
CustomerID              0
Gender                  0
Age                     0
Annual Income (k$)      0
Spending Score (1-100)  0
dtype: int64
```

[5]:
```
#checking for duplicates
df.duplicated().sum()
```

[5]:
```
0
```

[7]:
```
#encoding the gender feature
df = pd.get_dummies(df, columns = ['Gender'])
```

[8]:
```
df['Gender_Male'] = df['Gender_Male'].astype(int)
```

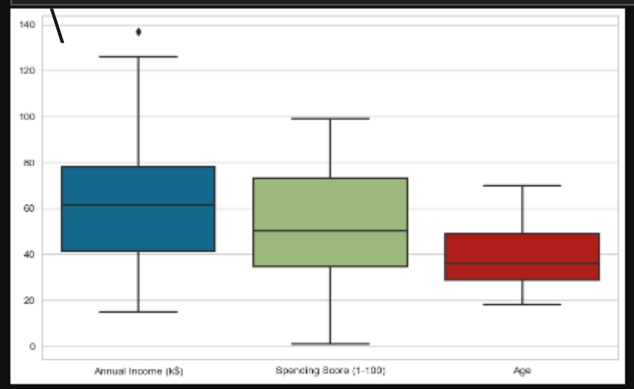
[9]:
```
df.drop(columns = ['Gender_Female'],inplace = True)
df.head()
```

[9]:

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

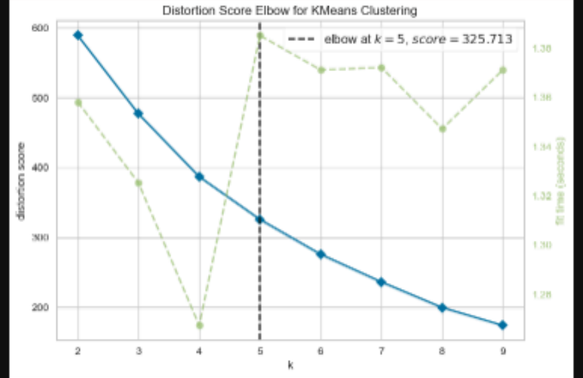
Code

[25]:
```
#checking for outliers
plt.figure(figsize=(10, 6))
sns.boxplot(data=df[['Annual Income (k$)', 'Spending Score (1-100)', 'Age']])
plt.show()
```



[10]:
```
#Feature selection
df = df.drop(columns=['CustomerID'], errors='ignore')

#scaling the features
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
df_scaled = scaler.fit_transform(df)
```

[11]:
```
#applying kmeans algorithm
from sklearn.cluster import KMeans
from yellowbrick.cluster import KElbowVisualizer
import matplotlib.pyplot as plt

# Apply K-Means clustering
model = KMeans(random_state=1)
visualizer = KElbowVisualizer(model, k=(2,10))

visualizer.fit(df_scaled)
visualizer.show()
plt.show()
```



[19]:
```
# Choose the optimal K based on the Elbow Curve
k_optimal = visualizer.elbow_value_

# Apply K-Means clustering with the optimal K
kmeans = KMeans(n_clusters=k_optimal, init='k-means++', random_state=42)
kmeans.fit(df_scaled)
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, calinski_harabasz_score, davies_bouldin_score

# Apply K-Means clustering with the optimal K
kmeans = KMeans(n_clusters=k_optimal, init='k-means++', random_state=42)
kmeans.fit(df_scaled)

# Get cluster labels
labels = kmeans.labels_

# Compute clustering metrics
inertia = kmeans.inertia_
silhouette = silhouette_score(df_scaled, labels)
calinski = calinski_harabasz_score(df_scaled, labels)
davies = davies_bouldin_score(df_scaled, labels)

# Print results

print(f"Calinski-Harabasz Index: {calinski:.2f}")
```

Code

\

```
  warnings.warn(
Calinski-Harabasz Index: 71.21
```

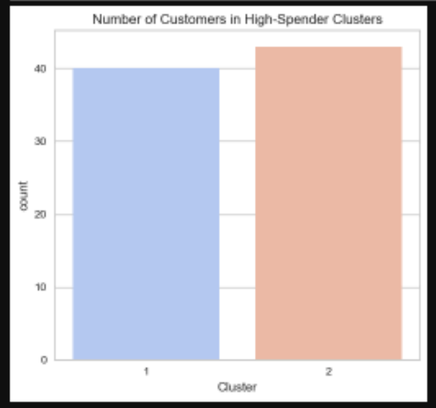
[13]:
```
# Add the cluster labels to the original dataframe
df['Cluster'] = labels

# Analyze the clusters
cluster_characteristics = []
for i in range(k_optimal):
    cluster_df = df[df['Cluster'] == i]
    characteristics = cluster_df.describe().iloc[1]  # Get the mean row
    characteristics['Cluster'] = f'Cluster {i+1}'
    cluster_characteristics.append(characteristics)

# Create a DataFrame
cluster_characteristics_df = pd.DataFrame(cluster_characteristics)
cluster_characteristics_df
```

```
C:\Users\Lenova\AppData\Local\Temp\ipykernel_11288\2561784790.py:12: FutureWarning: Setting an item of incompatible dtype is deprecated and will raise in a future error of pandas. Value 'Cluster 1' has dtype incompatible with float64, please explicitly cast to a compatible dtype first.
  characteristics['Cluster'] = f'Cluster {i+1}'
C:\Users\Lenova\AppData\Local\Temp\ipykernel_11288\2561784790.py:12: FutureWarning: Setting an item of incompatible dtype is deprecated and will raise in a future error of pandas. Value 'Cluster 2' has dtype incompatible with float64, please explicitly cast to a compatible dtype first.
  characteristics['Cluster'] = f'Cluster {i+1}'
C:\Users\Lenova\AppData\Local\Temp\ipykernel_11288\2561784790.py:12: FutureWarning: Setting an item of incompatible dtype is deprecated and will raise in a future error of pandas. Value 'Cluster 3' has dtype incompatible with float64, please explicitly cast to a compatible dtype first.
  characteristics['Cluster'] = f'Cluster {i+1}'
C:\Users\Lenova\AppData\Local\Temp\ipykernel_11288\2561784790.py:12: FutureWarning: Setting an item of incompatible dtype is deprecated and will raise in a future error of pandas. Value 'Cluster 4' has dtype incompatible with float64, please explicitly cast to a compatible dtype first.
  characteristics['Cluster'] = f'Cluster {i+1}'
C:\Users\Lenova\AppData\Local\Temp\ipykernel_11288\2561784790.py:12: FutureWarning: Setting an item of incompatible dtype is deprecated and will raise in a future error of pandas. Value 'Cluster 5' has dtype incompatible with float64, please explicitly cast to a compatible dtype first.
  characteristics['Cluster'] = f'Cluster {i+1}'
```

[13]:

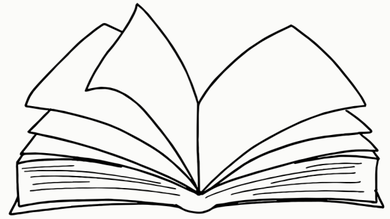
	Age	Annual Income (k$)	Spending Score (1-100)	Gender_Male	Cluster
mean	28.345455	60.800000	68.654545	0.000000	Cluster 1
mean	28.250000	62.000000	71.675000	1.000000	Cluster 2
mean	48.720930	46.186047	39.674419	0.000000	Cluster 3
mean	55.903226	48.774194	38.806452	1.000000	Cluster 4
mean	40.419355	90.000000	15.741935	0.548387	Cluster 5

[17]:
```
#targeting the high spenders
import matplotlib.pyplot as plt
import seaborn as sns

# Filter data for high-spending clusters (1 & 2)
high_spenders = df[df['Cluster'].isin([1, 2])]

# Set up the figure
plt.figure(figsize=(15, 5))

# Customer Count in Each Cluster
plt.subplot(1, 3, 3)
sns.countplot(x='Cluster', data=high_spenders, palette='coolwarm')
plt.title("Number of Customers in High-Spender Clusters")

# Show the dashboard
plt.tight_layout()
plt.show()
```



In conclusion we can keep targeting the cluster one and two because they are high in spending score and we can do marketing to ather clusters .

References



Raja Lakshmi Eduverse Data Science Course – Mrs. Gowthami Shyam, Instructor

- Course: Data Science
- Institution: Rajalakshmi Eduverse
- This course provided foundational knowledge and practical skills in data science, including data preprocessing, machine learning algorithms, and model evaluation.

Kaggle (2025). "Mall Customer Segmentation Dataset."
Retrieved from https://www.kaggle.com/datasets
The dataset used for this project was sourced from Kaggle, which provided relevant customer data for clustering and analysis.

Thank You