Name – Janhvi Saste
PRN – 22311825
Roll no – 282065
Year – SY B

Assignment No - 05

Problem Statement-

Write a program to do following:
Data Set: https://www.kaggle.com/shwetabh123/mall-customers
This dataset gives the data of Income and money spent by the customers visiting a shopping mall.
The data set contains Customer ID, Gender, Age, Annual Income, Spending Score. Therefore, As a mall owner you need to find the group of people who are the profitable customers for the Mall owner. Apply at least two clustering algorithms (based on Spending Score) to find the group of customers.
a) Apply Data pre-processing
b) Perform data-preparation (Train-Test Split)
c) Apply Machine Learning Algorithm
d) Evaluate Model.
e) Apply Cross-Validation and Evaluate Mode

Objectiive

By addressing its present shortcomings, the goal is to improve a simple and intelligible machine learning technique for forecasting consumer reactions using a dataset from a cosmetics store. By using data preprocessing (e.g., scaling, encoding), integrating visualizations (e.g., confusion matrix heatmaps, feature distributions), performing extensive data exploration (e.g., summary statistics, correlation analysis), and making sure the dataset is in line with the problem context—thereby reducing overfitting and data mismatch issues—the objective is to improve accuracy and relevance. This will turn the approach from a sound foundation into a reliable, workable solution appropriate for actual use.

Methodology

The code implements an initial setup for a clustering task to identify profitable customer groups in a shopping mall dataset, followed by examples of cross-validation techniques. The steps align with the assignment requirements as follows:

1. Library Import: Import pandas, numpy, matplotlib.pyplot, seaborn, and sklearn.model_selection for data handling, computation, visualization, and cross-validation.
2. Data Loading & Preprocessing: Load the Mall_Customers.csv dataset using pandas, check for missing values (isnull().sum()), and inspect data structure (info(), describe()).
3. Data Preparation: Although clustering doesn't inherently require train-test splits, the code includes examples of splitting via cross-validation methods (not yet applied to clustering).
4. Machine Learning Algorithms: The code is incomplete for clustering (e.g., K-Means, DBSCAN), but it sets the stage for applying at least two clustering algorithms based on Spending Score.
5. Model Evaluation: Not implemented yet, but typically involves metrics like silhouette score for clustering.
6. Cross-Validation: Demonstrates K-Fold, Leave-One-Out, Leave-P-Out, Stratified K-Fold, and Repeated K-Fold using sklearn.model_selection with toy datasets.

Main Functions

1. pandas (pd)
    - pd.read_csv('/content/Mall_Customers.csv'): Loads the dataset into a DataFrame.
    - dataset.head(): Displays the first 5 rows.
    - dataset.isnull().sum(): Checks for missing values in each column.

    - dataset.info(): Provides data types and non-null counts.
    - dataset.describe(): Summarizes numerical columns (mean, std, min, max, etc.).
2. numpy (np)
    - np.array(): Creates arrays for toy datasets in cross-validation examples (e.g., x = np.array([[1,2], ...])).
3. matplotlib.pyplot (plt)
    - Not Used Yet: Imported but no plotting implemented (e.g., for visualizing clusters).
4. seaborn (sns)
    - Not Used Yet: Imported but no statistical visualizations applied (e.g., pair plots for Spending Score).
5. sklearn.model_selection
    - KFold(n_splits=2): Splits data into 2 folds for K-Fold cross-validation.
    - LeaveOneOut(): Performs Leave-One-Out cross-validation, leaving one sample out per iteration.
    - LeavePOut(2): Performs Leave-P-Out cross-validation, leaving 2 samples out per iteration.
    - StratifiedKFold(n_splits=2): Splits data into 2 folds while preserving class distribution.
    - RepeatedKFold(n_splits=2, n_repeats=3): Repeats K-Fold cross-validation 3 times with 2 folds.
    - kf.split(x), loo.split(x), etc.: Generates train/test indices for each cross-validation method.

Advantages :

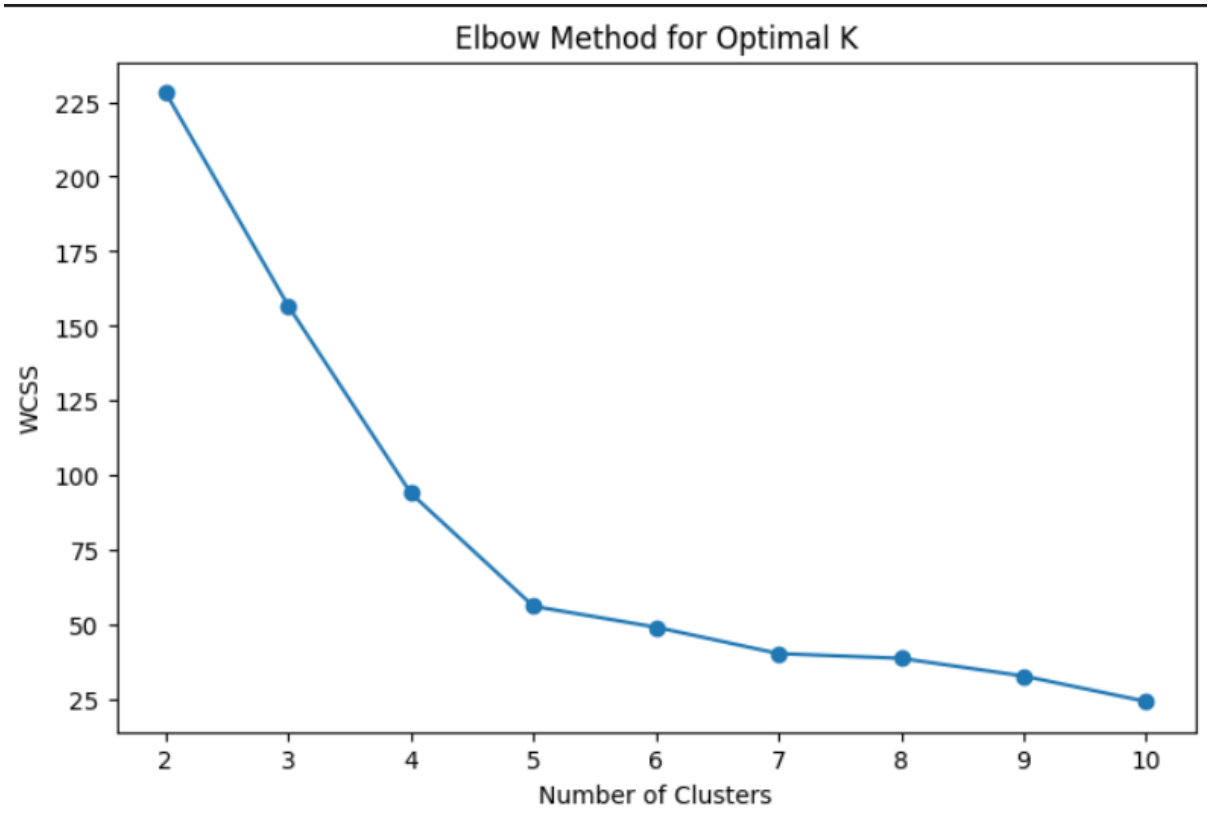
- Pandas: Easy data loading and inspection, integrates well with sklearn.
- NumPy: Fast array operations, supports toy datasets for cross-validation.
- Matplotlib.pyplot: Flexible plotting for cluster visualization (unused yet).
- Seaborn: Simple, attractive statistical plots (unused yet).
- Scikit-learn (model_selection): Versatile cross-validation tools, easy to use.

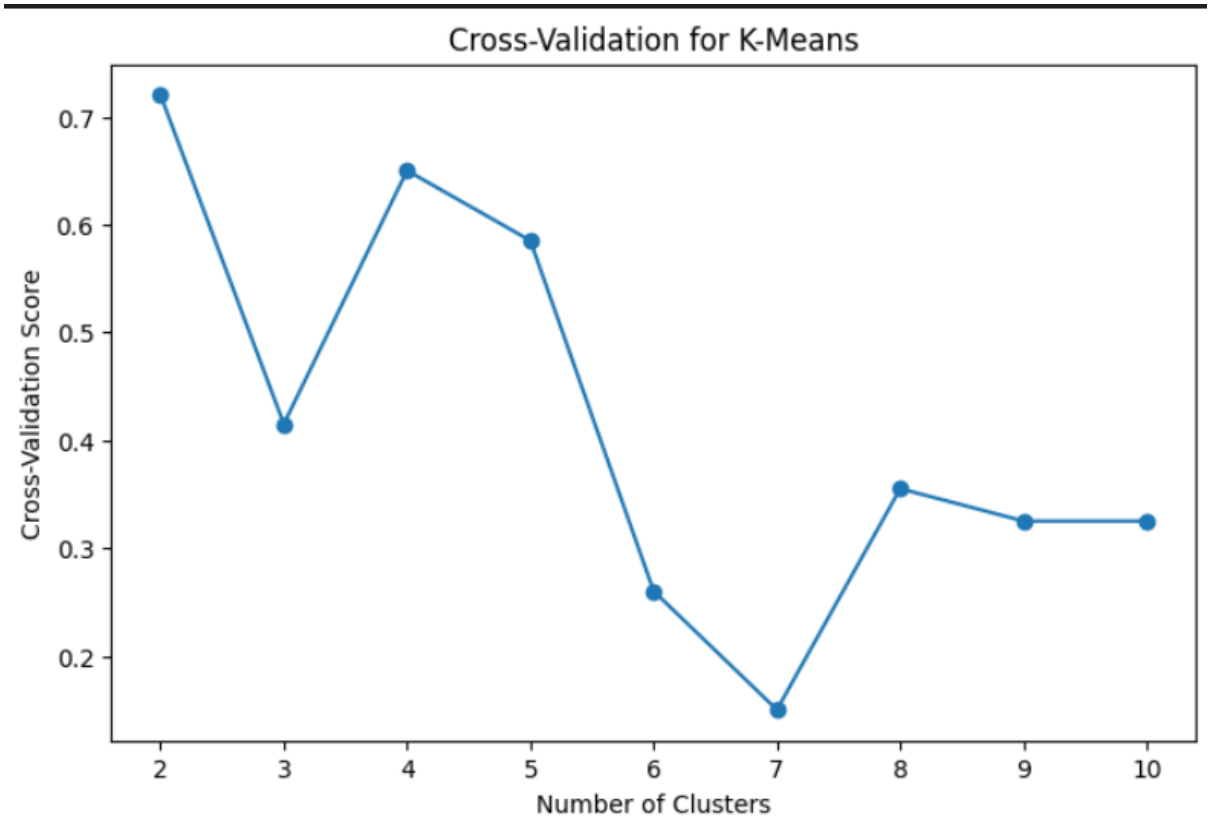
Disadvantages:

- Pandas: Limited preprocessing here, memory-heavy for large data.
- NumPy: Minimal role beyond toy arrays, low-level for complex tasks.
- Matplotlib.pyplot: Unused, verbose for basic plots.
- Seaborn: Unused, relies on matplotlib for customization.
- Scikit-learn (model_selection): Misapplied to clustering, some methods computationally intensive.

Conclusion

With the help of pandas for data loading and inspection, numpy for array operations, and sklearn.model_selection for illustrating cross-validation techniques, the Jupyter Notebook code for Assignment 5 creates a preliminary framework for clustering the Mall_Customers.csv dataset in order to identify profitable customer groups based on Spending Score. Although they are provided for possible visualization, libraries like seaborn and matplotlib.pyplot are not used. The key clustering techniques (e.g., K-Means, DBSCAN), model evaluation (e.g., silhouette score), and full data preparation (e.g., scaling, train-test split for supervised context) are not fully handled, even though the data pretreatment phase is largely covered (e.g., checking missing values). Despite being well-executed, the cross-validation samples have little bearing on the unsupervised clustering job.


Elbow Method for Optimal K
WCSS
225
200
175
150
125
100
75
50
25
2
3
4
5
6
7
8
9
10
Number of Clusters


Cross-Validation for K-Means
Cross-Validation Score
0.7
0.6
0.5
0.4
0.3
0.2
2
3
4
5
6
7
8
9
10
Number of Clusters


Customer Segments using K-Means
Spending Score (1-100)
Annual Income (k$)
KMeans_Cluster
0
1
2
3
4
0
20
40
60
80
100
20
40
60
80
100
120
140