Final Project Report

1. Introduction

The objective of this project was to apply clustering techniques to gain insights into our dataset, identify natural groupings, and provide valuable information for decision-makers. In pursuit of this goal, we employed three distinct clustering algorithms: K-Means, Hierarchical Agglomerative Clustering (HAC), and Mean Shift. This report outlines our findings, the common results observed across the algorithms, and their implications.

2. Data Overview

The dataset employed in this report was sourced from Kaggle.com and comprises information pertaining to 200 mall customers. This dataset encompasses five distinct features, each providing valuable insights into customer attributes and behavior:

- CustomerID: A unique identifier assigned to each customer, facilitating individual identification within the dataset.
- Gender: This categorical attribute delineates the gender of the mall customers, categorizing them as either "male" or "female".
- Age: Representing the age of each customer, this numerical attribute quantifies the chronological age of the individuals.
- Annual Income (k$): A quantitative feature, this attribute delineates the annual income of each customer in thousands of dollars (k$).
- Spending Score (1-100): This numerical attribute measures the spending behavior of customers, providing a score ranging from 1 to 100, with higher scores indicating higher spending propensity.

The primary objective of this analysis is to apply clustering techniques to the "Annual Income" and "Spending Score" attributes to categorize customers into distinct groups based on their income and spending behavior.

3. Data preprocessing and Feature engineering

In this section, we provide a brief overview of the steps taken during data exploration and preprocessing:

1. **Data Exploration:**

Summary Statistics:

- Age: The average age of customers is approximately 38, with ages ranging from 18 to 70.
- Annual Income (k$): On average, customers report an annual income of around 60 thousand dollars, with incomes spanning from 15 to 137 k$.
- Spending Score (1-100): The spending score exhibits an average value of 50, with scores varying between 1 and 99.

Data Distribution:

Data in this dataset is not normally distributed.

Categorical Attributes:

Data in the feature "Gender" is not equally divided, about 42% for Male and 58% for Female.

2. **Feature Engineering:**

The categorical feature "Gender" was encoded to facilitate its inclusion in the clustering analysis. The encoding process transformed the "Gender" feature into numerical format for model compatibility. The encoding scheme used is as follows:

- "Male" was encoded as 1.
- "Female" was encoded as 0.

4. Clustering Models:

1. **Kmeans:**

The initial step in our analysis involved the application of the K-Means clustering algorithm. K-Means is a partitioning method that segregates customers into distinct groups, commonly referred to as clusters. The model was trained multiple times using varying values of k, specifically k = 3, k = 4, and k = 5. These different values of k were determined following the identification of elbow points in order to assess the optimal number of clusters. The elbow point analysis aids in pinpointing the most suitable value of k, offering insights into the inherent structure of the data.

2. **Hierarchical Agglomerative Clustering (HAC):**

Subsequently, Hierarchical Agglomerative Clustering (HAC) was employed as an alternative clustering technique. HAC is characterized by its hierarchical nature, creating a tree-like structure of clusters. In this analysis, HAC was utilized to group customers into clusters with varying sizes, specifically k = 3, k = 4, and k = 5. Euclidean distance was employed as the distance metric, while the ward linkage method was chosen as the linkage criterion. These parameters were selected to ensure the hierarchical clustering process yielded meaningful results.

3. **Mean Shift:**

   The final clustering approach employed in our analysis was Mean Shift clustering. Mean Shift is a density-based method that identifies clusters based on density peaks within the data distribution. In this case, the model was trained with Mean Shift clustering using an estimated bandwidth parameter. The bandwidth estimation was conducted using a quantile value of 0.1. This approach allows the model to discover clusters without the need for specifying the number of clusters a priori.

5. Model recommendations

Given the consistent clustering outcomes across all three unsupervised models, any of them can be chosen as the final clustering solution. The stability and uniformity in results indicate a clear inherent structure in the data.

6. Key findings and Insight

1. **Clear Segmentation by Income and Spending:**

A prominent and consistent pattern across all clustering models is the unequivocal segmentation of customers based on their "Annual Income" and "Spending Score". Customers have been effectively grouped into distinct clusters, each exhibiting unique characteristics in terms of their income levels and spending behavior.

2. **High Consistency Across Models:**

   The remarkable consistency in clustering results across multiple algorithms underscores the stability and robustness of the customer segmentation. Regardless of the technique employed, the clustering solutions uniformly reflect the underlying structure in the data.

7. Suggestion for next steps

1. **Feature Augmentation:**
   Consider augmenting the dataset with additional features that may influence customer behavior, such as customer demographics (e.g., marital status, education level) or external factors (e.g., economic indicators). The inclusion of new attributes can provide a richer context for understanding customer segments.
2. **External Data Integration:**
   Explore the possibility of integrating external data sources, such as economic indicators or competitor data, to gain a broader perspective on factors affecting customer behavior. External data can enrich the analysis and provide context.
3. **Apply Advanced Machine Learning Techniques:**
   Explore advanced machine learning techniques, such as deep learning or reinforcement learning, to develop predictive models for customer behavior.

8. Conclusion

In conclusion, this analysis represents a comprehensive exploration of mall customer data through the application of unsupervised learning techniques, including K-Means, Hierarchical Agglomerative Clustering (HAC), and Mean Shift. The main objectives of this endeavor were to identify customer segments, unveil underlying patterns, and assist data-driven decision-making for optimized marketing strategies.

The key findings and insights revealed a clear and consistent segmentation of customers based on their "Annual Income" and "Spending Score." Importantly, all three clustering models provided identical results, reinforcing the robustness of the identified customer segments. These insights empower businesses to tailor marketing campaigns and optimize resource allocation.

9. Python implementation

Import libraries

In [ ]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import metrics
import warnings
warnings.filterwarnings('ignore')
from sklearn.preprocessing import LabelEncoder
from sklearn.cluster import AgglomerativeClustering
from sklearn.cluster import KMeans
from sklearn.cluster import MeanShift, estimate_bandwidth
```

Read data and preprocessing

In [ ]:

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

Out[ ]:

	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

In [ ]:

```
df.shape
```

Out[ ]:

```
(200, 5)
```

In [ ]:

```
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
```

In [ ]:

```
df.describe().T
```

Out[ ]:

	count	mean	std	min	25%	50%	75%	max
CustomerID	200.0	100.50	57.879185	1.0	50.75	100.5	150.25	200.0
Age	200.0	38.85	13.969007	18.0	28.75	36.0	49.00	70.0
Annual Income (k$)	200.0	60.56	26.264721	15.0	41.50	61.5	78.00	137.0
Spending Score (1-100)	200.0	50.20	25.823522	1.0	34.75	50.0	73.00	99.0

In [ ]:

```
df.rename(index = str, columns = {'Annual Income (k$)':'Income',
                                  'Spending Score (1-100)':'Score'}, inplace = True)
```

In [ ]:

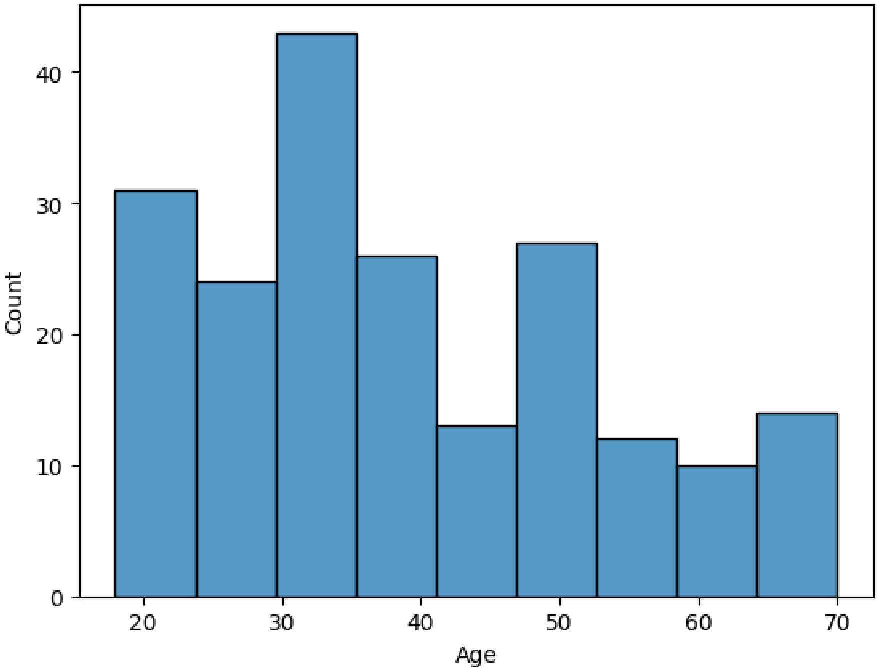
```
df.head()
```

Out[ ]:

	CustomerID	Gender	Age	Income	Score
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

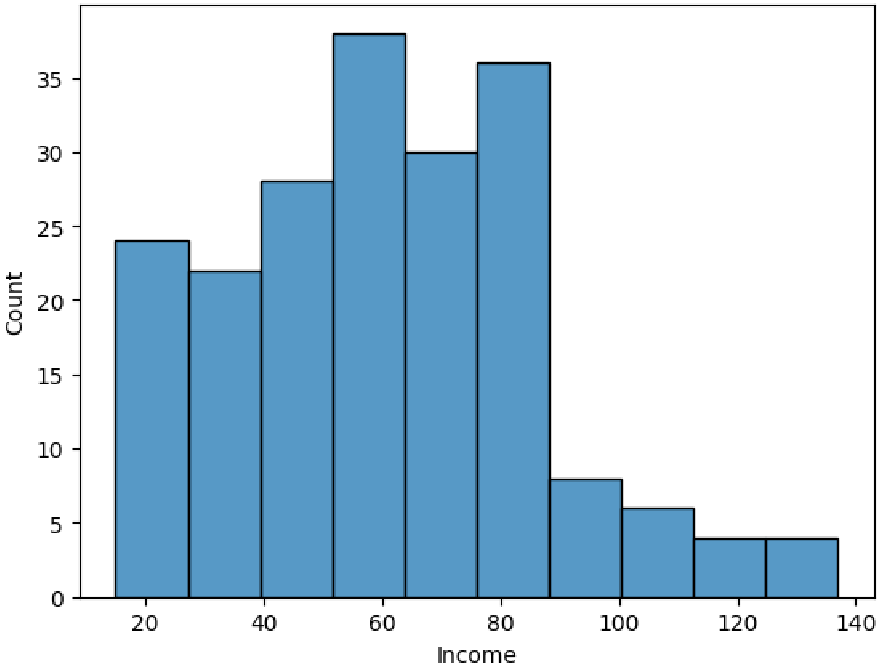
In [ ]:

```
sns.histplot(data = df['Age']);
```



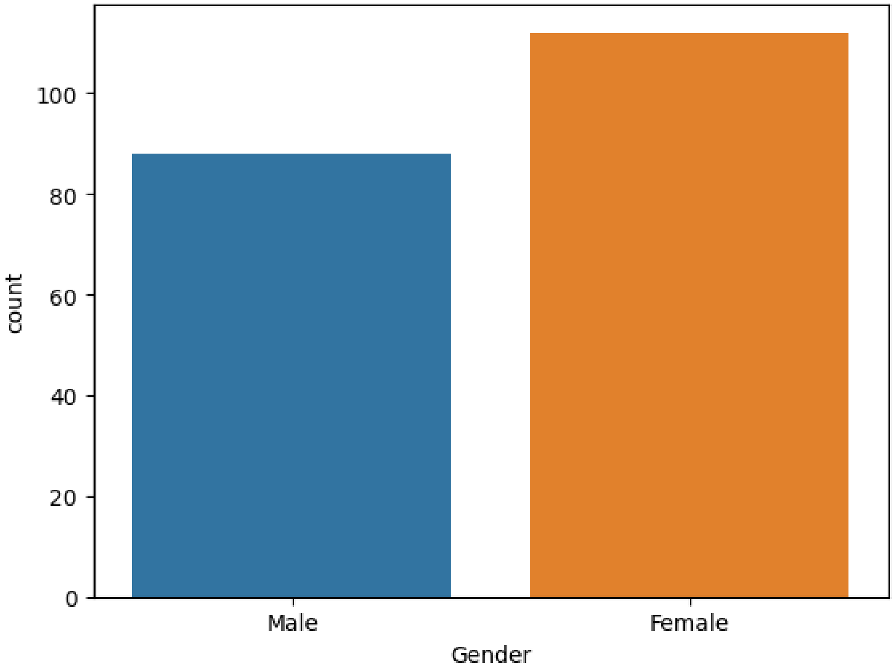
In [ ]:

```
sns.histplot(data = df['Income']);
```



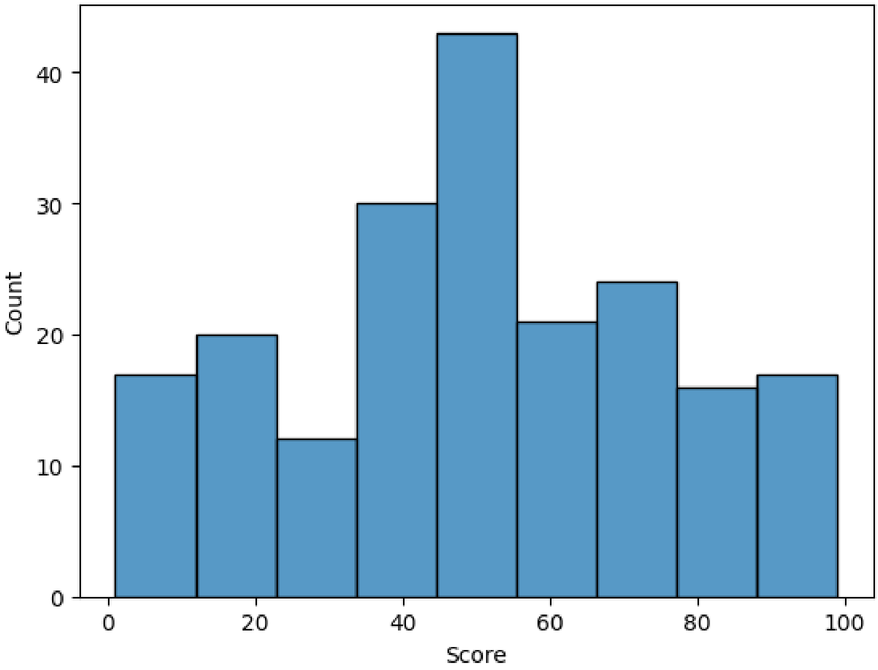
In [ ]:

```
sns.countplot(x = df['Gender']);
```



In [ ]:

```
sns.histplot (x = df['Score']);
```



It is obvious that the data in features [Age, Score, Income] are not normally distributed, while that in feature gender are not fairly divided

Encode column Gender

In [ ]:

```
label_encoder = LabelEncoder()

df['Gender'] = label_encoder.fit_transform(df['Gender'])
df
```

Out[ ]:

	CustomerID	Gender	Age	Income	Score
0	1	1	19	15	39
1	2	1	21	15	81
2	3	0	20	16	6
3	4	0	23	16	77
4	5	0	31	17	40
...	...	...	...	...	...
195	196	0	35	120	79
196	197	0	45	126	28
197	198	1	32	126	74
198	199	1	32	137	18
199	200	1	30	137	83

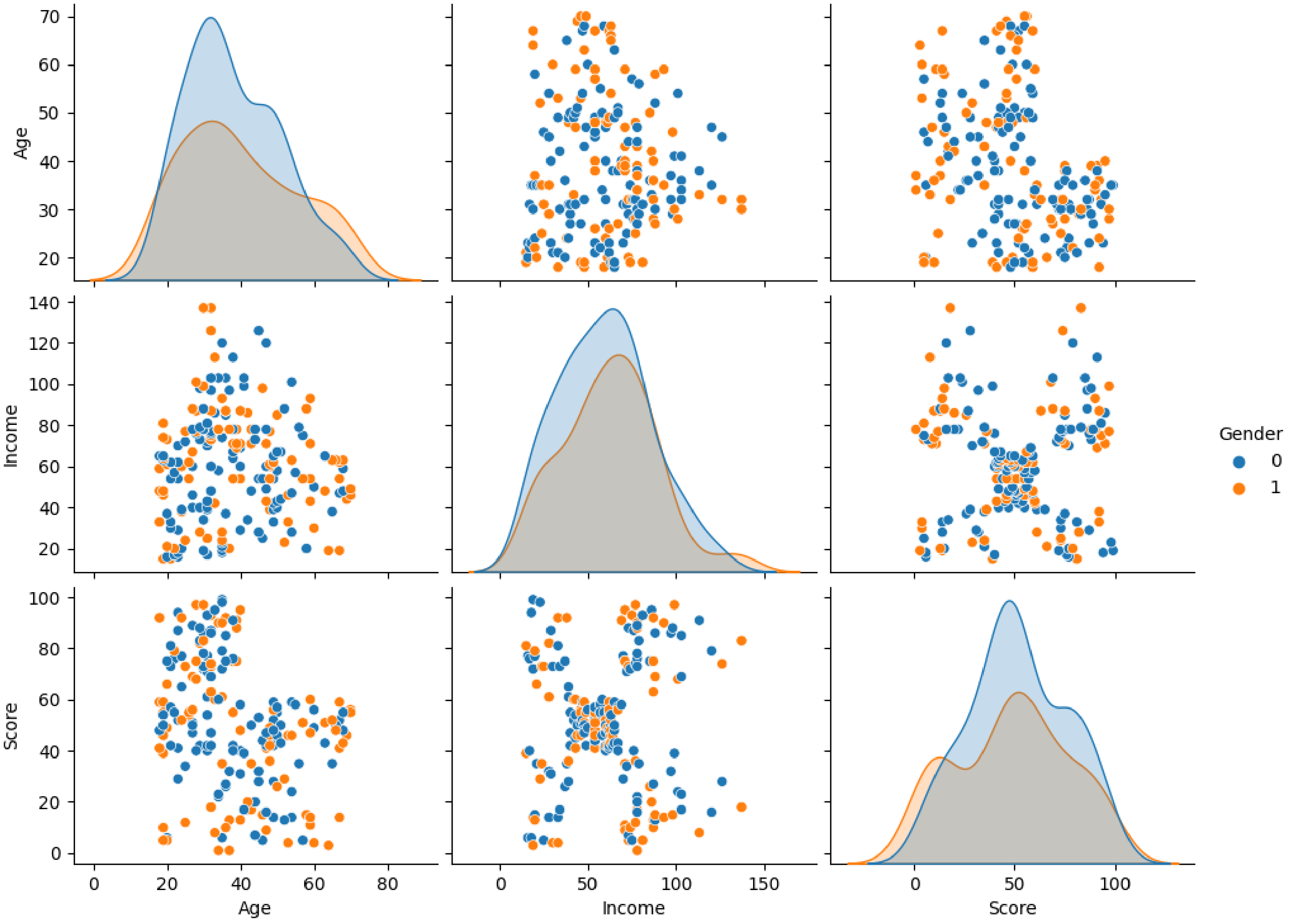
200 rows × 5 columns

In [ ]:

```
df = df.drop(columns = 'CustomerID', axis = 1)
```

In [ ]:

```
sns.pairplot (df, hue = 'Gender', aspect= 1.3);
```



Insight: the gender does not have any effects on customer clustering. Thus, it could be dropped

In [ ]:

```
df.head()
```

Out[ ]:

```
   Gender  Age  Income  Score
0       1   19      15     39
1       1   21      15     81
2       0   20      16      6
3       0   23      16     77
4       0   31      17     40
```

Set up train set and Train model

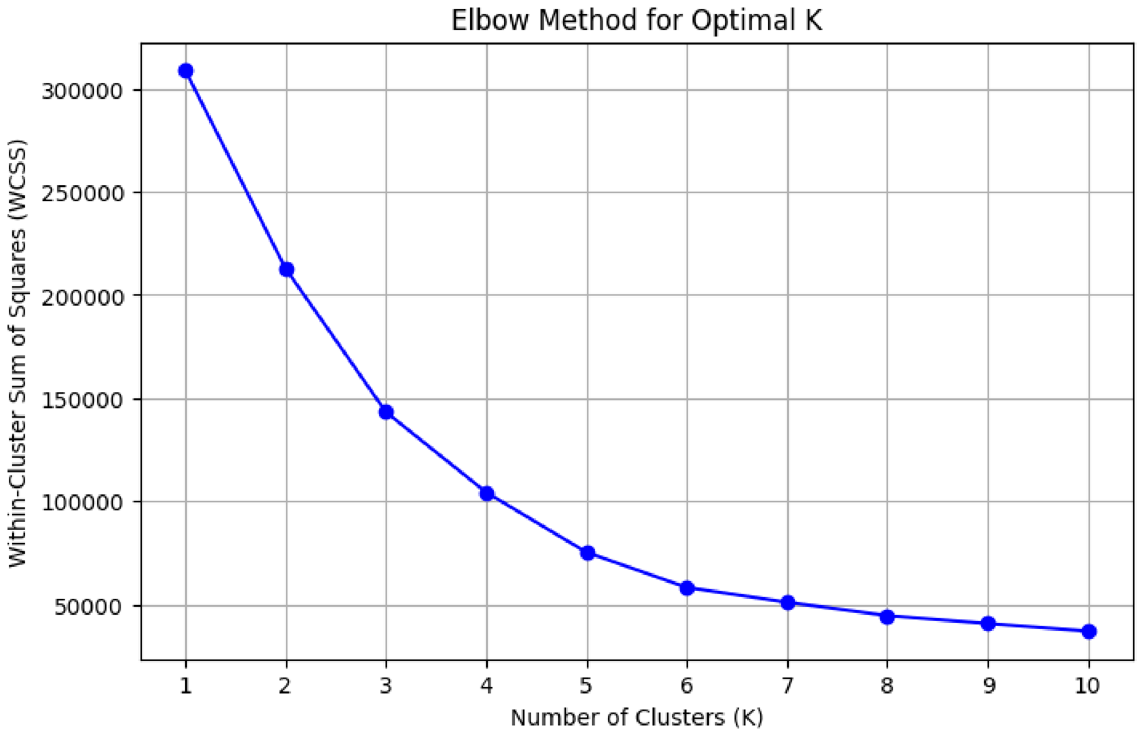
In [ ]:

```
X = df.drop('Gender', axis =1)
```

Train model with K_means

In [ ]:

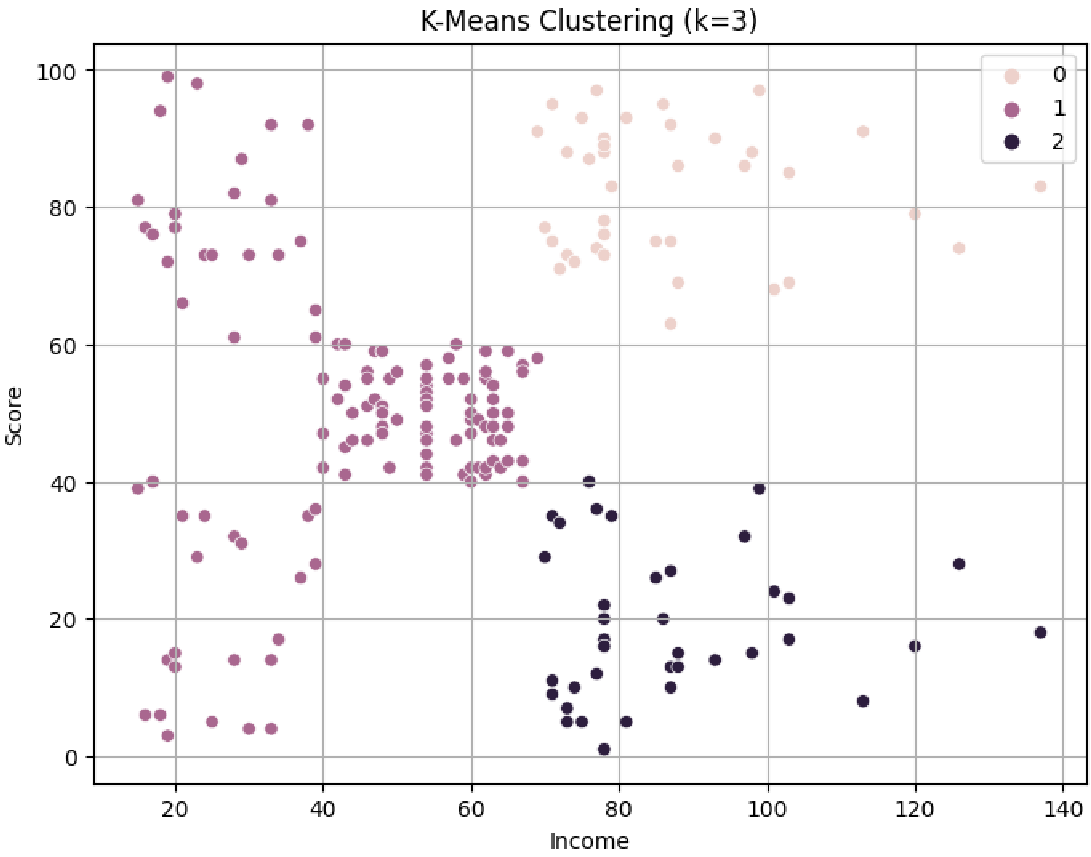
```
k_values = range(1, 11)
wcss = []
# calculate inertia
for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

# Plot the elbow curve
plt.figure(figsize=(8, 5))
plt.plot(k_values, wcss, marker='o', linestyle='-', color='b')
plt.title('Elbow Method for Optimal K')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('Within-Cluster Sum of Squares (WCSS)')
plt.xticks(k_values)
plt.grid(True)
plt.show()
```



Insight: k = 3, or k = 4 could be the elbow points

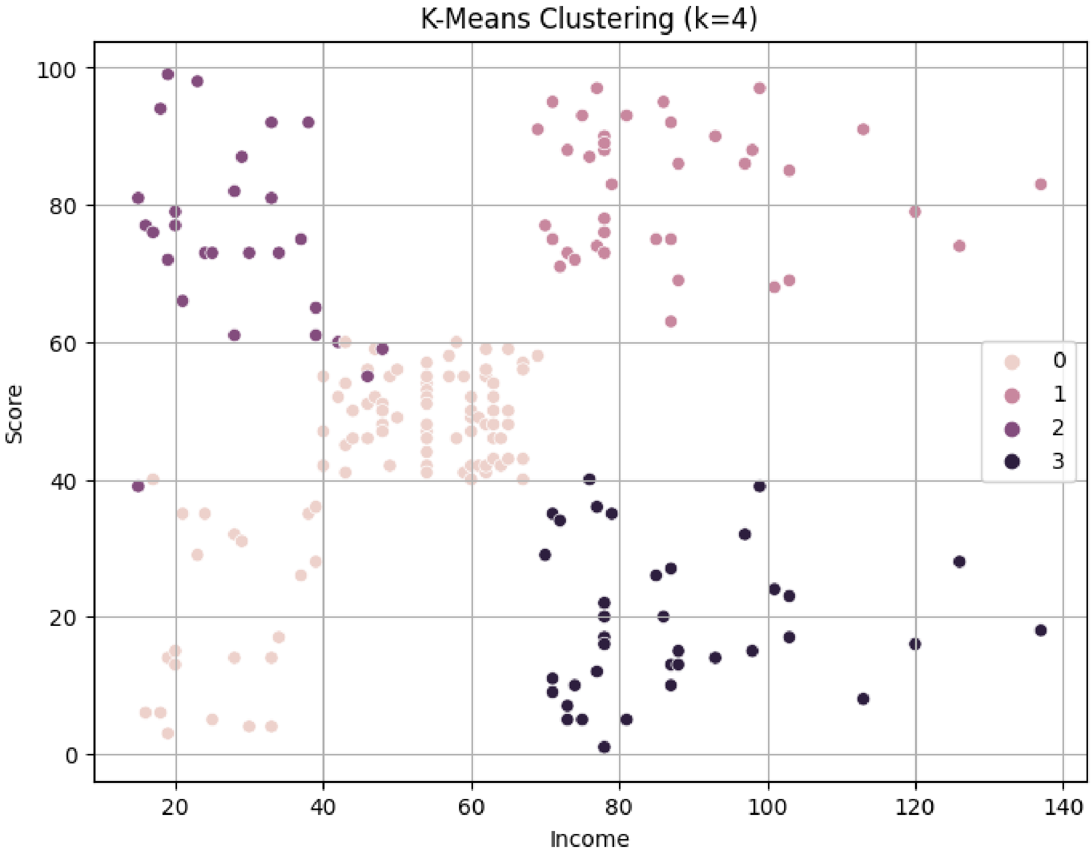
In [ ]:

```
# k = 3
kmeans = KMeans(n_clusters=k, random_state=42).fit(X)
X['Labels'] = kmeans.labels_

plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Income', y='Score', hue = X['Labels'])

plt.title('K-Means Clustering (k=3)')
plt.xlabel('Income')
plt.ylabel('Score')
plt.legend()
plt.grid(True)
plt.show();
```



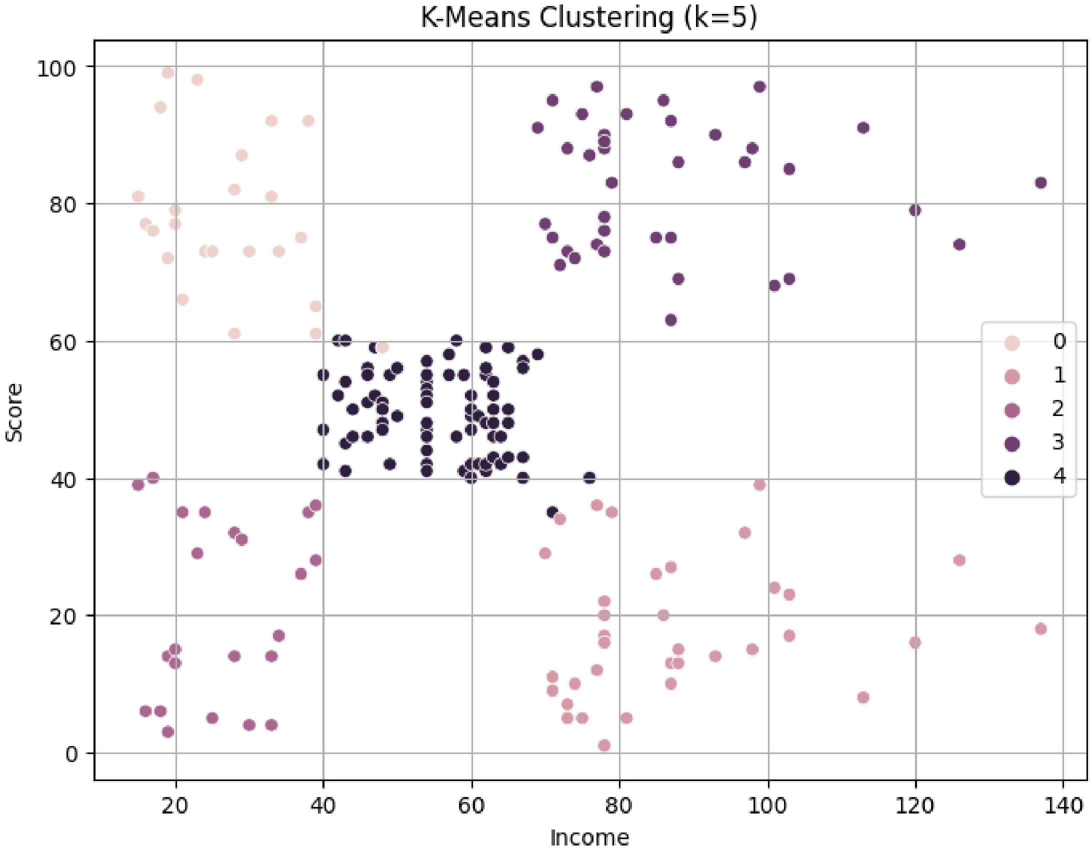
In [ ]:

```
# k = 4
kmeans = KMeans(n_clusters=k, random_state=42).fit(X)
X['Labels'] = kmeans.labels_

plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Income', y='Score', hue = X['Labels'])

plt.title('K-Means Clustering (k=4)')
plt.xlabel('Income')
plt.ylabel('Score')
plt.legend()
plt.grid(True)
plt.show();
```



In [ ]:

```
# k = 5
kmeans = KMeans(n_clusters=k, random_state=42).fit(X)
X['Labels'] = kmeans.labels_

plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Income', y='Score', hue = X['Labels'])

plt.title('K-Means Clustering (k=5)')
plt.xlabel('Income')
plt.ylabel('Score')
plt.legend()
plt.grid(True)
plt.show();
```

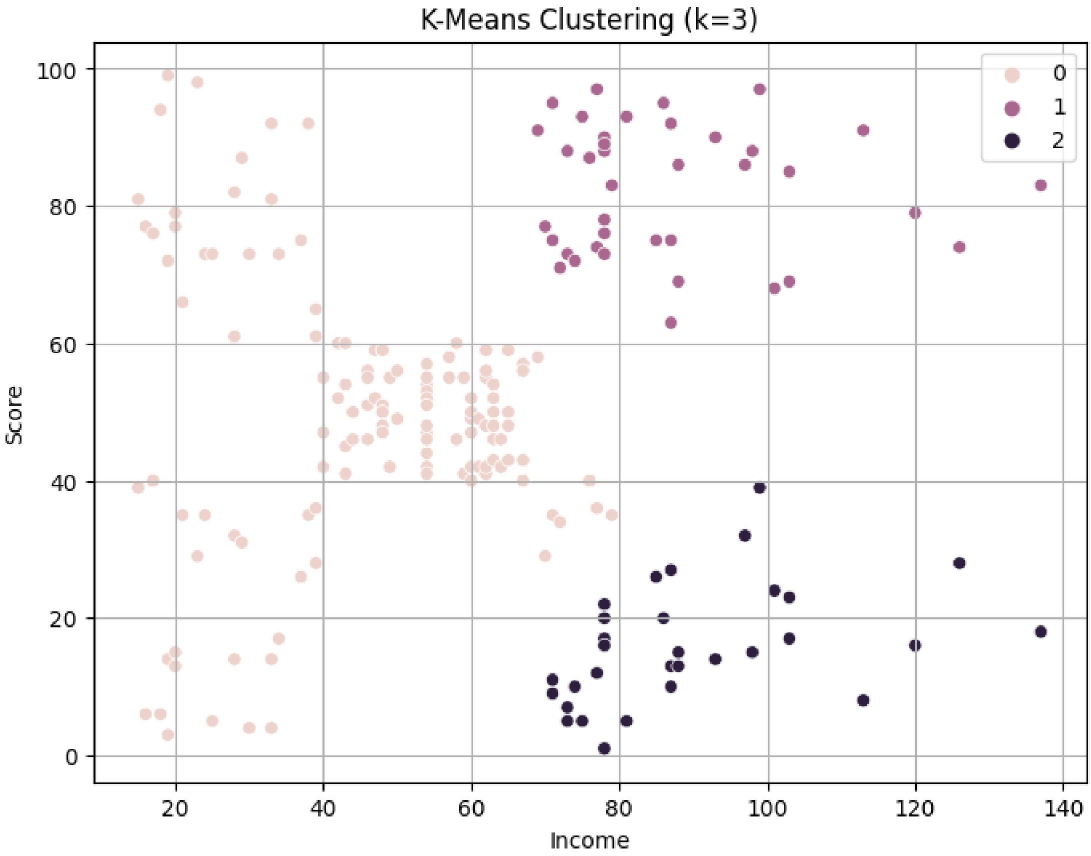


Insight: At k = 5, the segmentation seems to be rational with customers are divided into 5 groups based on their spending and income

Train model with Hierarchical Agglomerative Clustering (HAC)

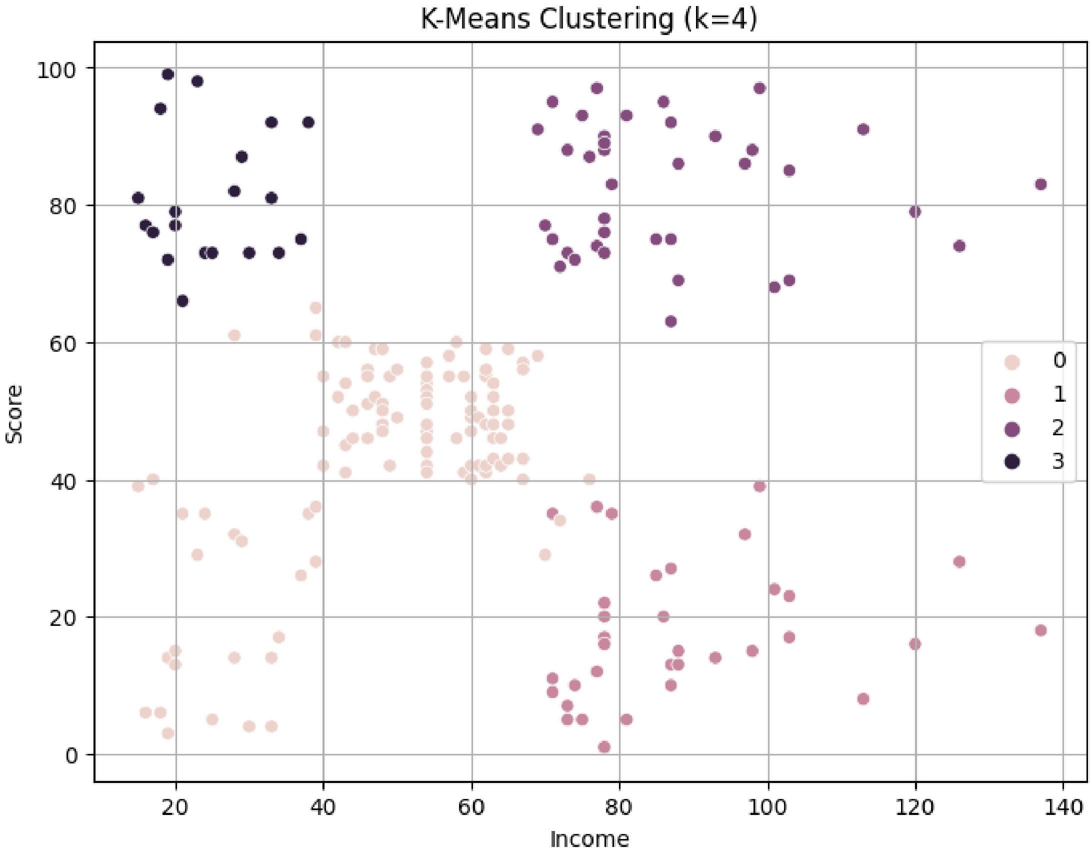
In [ ]:

```
k = 3
agg = AgglomerativeClustering(n_clusters=k, affinity='euclidean', linkage='ward').fit(X)
X['Labels'] = agg.labels_

plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Income', y='Score', hue = X['Labels'])

plt.title('K-Means Clustering (k=3)')
plt.xlabel('Income')
plt.ylabel('Score')
plt.legend()
plt.grid(True)
plt.show();
```



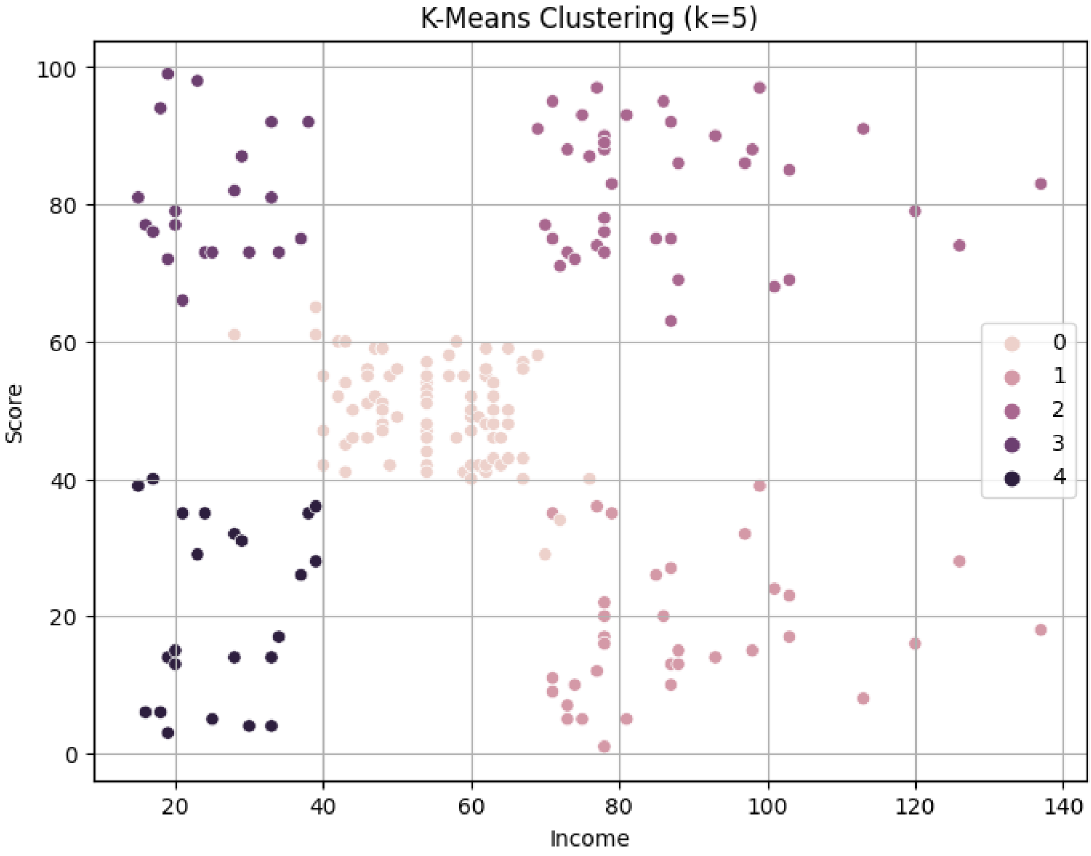
In [ ]:

```
k = 4
agg = AgglomerativeClustering(n_clusters=k, affinity='euclidean', linkage='ward').fit(X)
X['Labels'] = agg.labels_

plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Income', y='Score', hue = X['Labels'])

plt.title('K-Means Clustering (k=4)')
plt.xlabel('Income')
plt.ylabel('Score')
plt.legend()
plt.grid(True)
plt.show();
```



In [ ]:

```
k = 5
agg = AgglomerativeClustering(n_clusters=k, affinity='euclidean', linkage='ward').fit(X)
X['Labels'] = agg.labels_
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Income', y='Score', hue = X['Labels'])
plt.title('K-Means Clustering (k=5)')
plt.xlabel('Income')
plt.ylabel('Score')
plt.legend()
plt.grid(True)
plt.show();
```

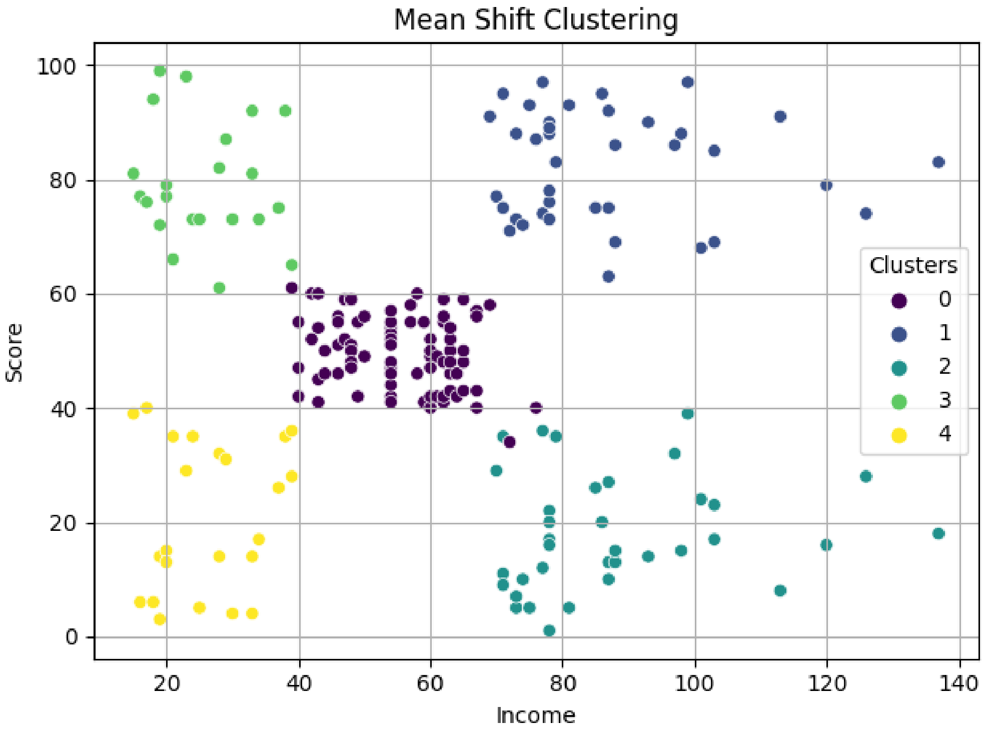


Hierarchical Clustering Agglomerative gave the same result as Kmeans did

Train model with Mean shift

In [ ]:

```
bandwidth = estimate_bandwidth(X, quantile=0.1)
ms = MeanShift(bandwidth = bandwidth).fit(X)
X['Meanshift_Labels'] = ms.labels_

sns.scatterplot(data=df, x='Income', y='Score', hue=X['Meanshift_Labels'], palette='viridis')
plt.title('Mean Shift Clustering')
plt.xlabel('Income')
plt.ylabel('Score')
plt.legend(title='Clusters')
plt.grid(True)
plt.tight_layout()
plt.show()
```



Conclusion

All of the algorithms used in this notebook return exactly the same results

Pros:

- Easy to understand and implement (and debug)
- Highly reasonable results that helps the customer segmentation easier and more understandable
- Consistently identified similar clusters in the dataset, which suggests the presence of meaningful groupings in the data.

Cons:

- All three algorithms (K-Means, HAC, and Mean Shift) require parameter tuning
- Some clustering algorithms may not scale well to very large datasets, impacting their efficiency and effectiveness
- Hierarchical Agglomerative Clustering (HAC) and Mean Shift can be computationally expensive, particularly for large datasets

Thank you reading the whole notebook, if something is wrong, please give me constructive feedbacks.

THE END