UNSUPERVISED MACHINE LEARNING

COURSE PROJECT

Algorithms belonging to the family of Unsupervised Learning have no variable to predict tied to the data. Instead of having an output, the data only has an input which would be multiple variables that describe the data. This is where clustering comes in.

Clustering is the task of grouping together a set of objects in a way that objects in the same cluster are more similar to each other than to objects in other clusters. Similarity is a metric that reflects the strength of relationship between two data objects. Clustering is mainly used for exploratory data mining. It has manifold usage in many fields such as machine learning, pattern recognition, image analysis, information retrieval, bio-informatics, data compression, and computer graphics.

ABOUT THE DATA

This file contains the basic information (ID, age, gender, income, spending score) about the customers. It is data related to customers spending and income. It is a https://www.kaggle.com/datasets/vjchoudhary7/customer-segmentation-tutorial-in-python?select=Mall_Customers.csv file.

OBJECTIVE

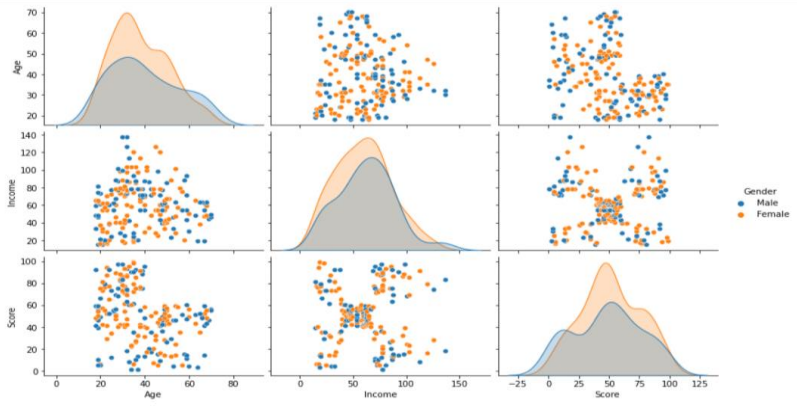
See customer relations over age, income and speding score.

1. Simple EDA
- Descriptive statistics and data cleaning
2. Applying various clustering algorithms
- K Means
- Agglomerative clustering
- DBSCAN
- Mean Shift
3. Comparing the algorithms
4. Conclusion

METHODS USED

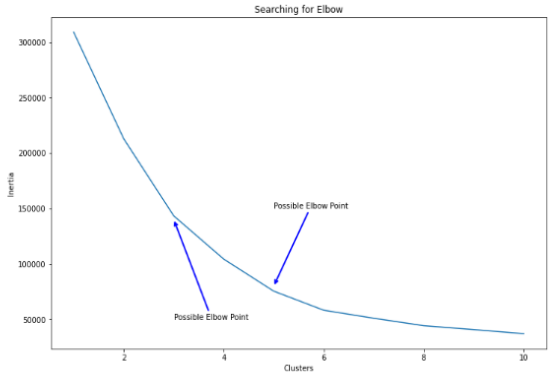
1. Import all libraries and the dataset.
2. We drop the columns gender and customer id as those are not needed for the algorithms.
3. Take the important columns and print a pairplot to see the relation between them.

```
In [14]: X = df.drop(['CustomerID', 'Gender'], axis=1)
         sns.pairplot(df.drop('CustomerID', axis=1), hue='Gender', aspect=1.5)
         plt.show()
```



4. Check foe the elbow using K Means method. Elbow method tells us to select the cluster when there is a significant change in inertia. As we can see from the graph, we can say this may be either 3 or 5. Let's see both results in graph and decide.

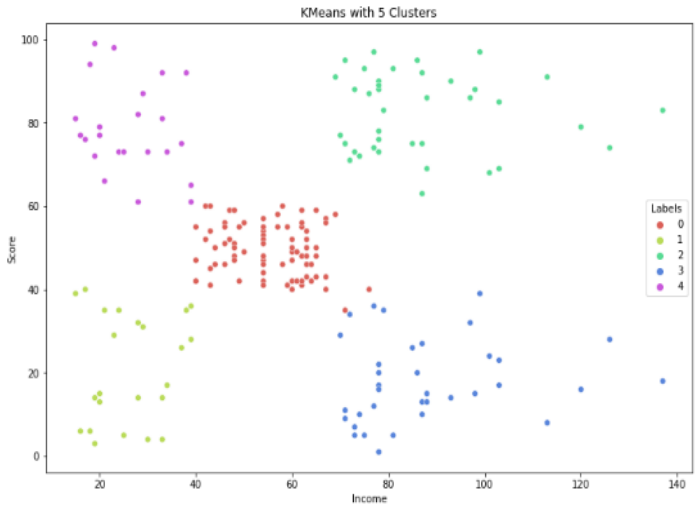
```
In [15]: from sklearn.cluster import KMeans

clusters = []

for i in range(1, 11):
    km = KMeans(n_clusters=i).fit(X)
    clusters.append(km.inertia_)

fig, ax = plt.subplots(figsize=(12, 8))
sns.lineplot(x=list(range(1, 11)), y=clusters, ax=ax)
ax.set_title('Searching for Elbow')
ax.set_xlabel('Clusters')
ax.set_ylabel('Inertia')

ax.annotate('Possible Elbow Point', xy=(3, 140000), xytext=(3, 50000), xycoords='data',
            arrowprops=dict(arrowstyle='->', connectionstyle='arc3', color='blue', lw=2))

ax.annotate('Possible Elbow Point', xy=(5, 80000), xytext=(5, 150000), xycoords='data',
            arrowprops=dict(arrowstyle='->', connectionstyle='arc3', color='blue', lw=2))

plt.show()
```



5. Apply K Means method with 3 clusters and 5 clusters to see which fits better


```
In [18]: km3 = KMeans(n_clusters=5).fit(X)

X['Labels'] = km3.labels_
plt.figure(figsize=(12, 8))
sns.scatterplot(X['Income'], X['Score'], hue=X['Labels'],
                palette=sns.color_palette('hls', 5))
plt.title('KMeans with 5 Clusters')
plt.show()
```

```
C:\Python310\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. Fr
om version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword wil
l result in an error or misinterpretation.
  warnings.warn(
```

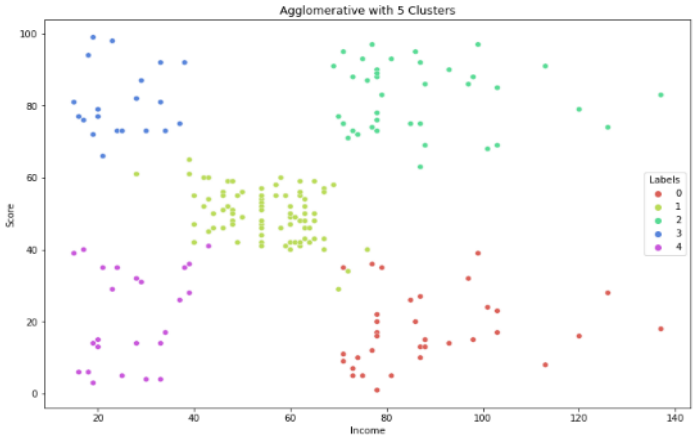



```
In [19]: fig = plt.figure(figsize=(20,8))
ax = fig.add_subplot(121)
sns.swarmplot(x='Labels', y='Income', data=X, ax=ax)
```

6. Apply agglomerative clustering and also plotted the deno diagram for it for different linkages.

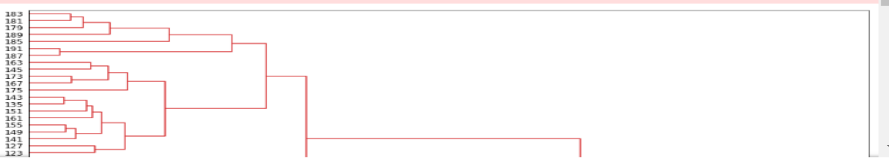
```
In [20]: from sklearn.cluster import AgglomerativeClustering

agglom = AgglomerativeClustering(n_clusters=5, linkage='average').fit(X)

X['Labels'] = agglom.labels_
plt.figure(figsize=(12, 8))
sns.scatterplot(X['Income'], X['Score'], hue=X['Labels'],
                palette=sns.color_palette('hls', 5))
plt.title('Agglomerative with 5 Clusters')
plt.show()
```

```
C:\Python310\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. Fr
om version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword wil
l result in an error or misinterpretation.
  warnings.warn(
```



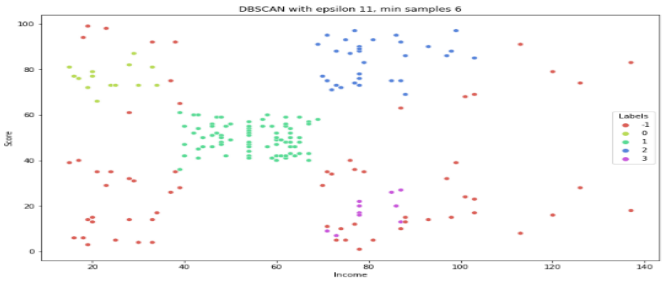
7. Apply DBSCAN

```
C:\Users\SHAH2H~1\AppData\Local\Temp\ipykernel_15428\2848005678.py:1: ClusterWarning: scipy.cluster: The symmetric non-negat
ive hollow observation matrix looks suspiciously like an uncondensed distance matrix
  Z = hierarchy.linkage(dist, 'average')
```



```
In [29]: from sklearn.cluster import DBSCAN

db = DBSCAN(eps=11, min_samples=6).fit(X)

X['Labels'] = db.labels_
plt.figure(figsize=(12, 8))
sns.scatterplot(X['Income'], X['Score'], hue=X['Labels'],
                palette=sns.color_palette('hls', np.unique(db.labels_).shape[0]))
plt.title('DBSCAN with epsilon 11, min samples 6')
plt.show()
```

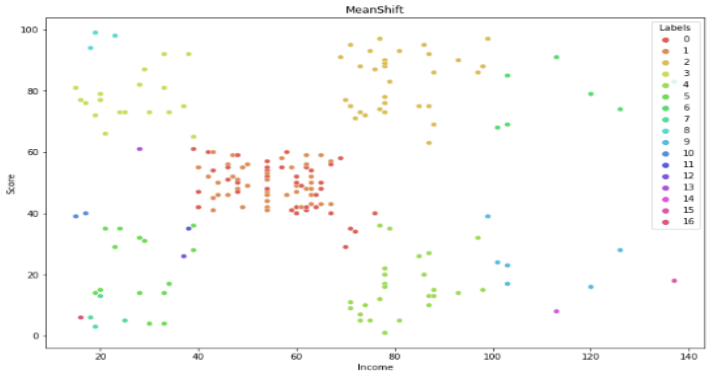
```
C:\Python310\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. Fr
om version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword wil
l result in an error or misinterpretation.
  warnings.warn(
```



8. Applied Mean Shift

```
In [39]: from sklearn.cluster import MeanShift, estimate_bandwidth

         # The following bandwidth can be automatically detected using
         bandwidth = estimate_bandwidth(X, quantile=0.1)
         ms = MeanShift(bandwidth=20).fit(X)

         X['Labels'] = ms.labels_
         plt.figure(figsize=(12, 8))
         sns.scatterplot(X['Income'], X['Score'], hue=X['Labels'],
                         palette=sns.color_palette('hls', np.unique(ms.labels_).shape[0]))
         plt.plot()
         plt.title('MeanShift')
         plt.show()
```

```
C:\Python310\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. Fr
om version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword wil
l result in an error or misinterpretation.
  warnings.warn(
```

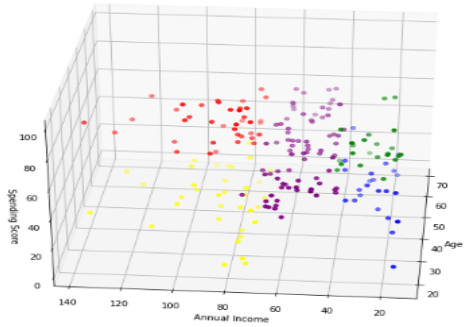


9. Applying the kmeans on all three of these paramaters and plot a 3D graph relating them.

```
In [50]: print(kmeans.cluster_centers_)
```

```
[[24.96       28.04       77.        ]
 [32.69230769 86.53846154 82.12820513]
 [45.2173913  26.30434783 20.91304348]
 [43.72727273 55.48051948 49.32467532]
 [40.66666667 87.75       17.58333333]]
```

```
In [56]: clusters=kmeans.fit_predict(x3)
         df["label"]= clusters
         from mpl_toolkits.mplot3d import Axes3D
         fig=plt.figure(figsize=(20, 10))
         ax=fig.add_subplot(111,projection="3d")
         ax.scatter(df.Age[df.label==0],df["Income"][df.label==0],df["Score"][df.label==0],c="blue" )
         ax.scatter(df.Age[df.label==1],df["Income"][df.label==1],df["Score"][df.label==1],c="red" )
         ax.scatter(df.Age[df.label==2],df["Income"][df.label==2],df["Score"][df.label==2],c="green" )
         ax.scatter(df.Age[df.label==3],df["Income"][df.label==3],df["Score"][df.label==3],c="yellow" )
         ax.scatter(df.Age[df.label==4],df["Income"][df.label==4],df["Score"][df.label==4],c="purple" )
         ax.view_init(30,185)
         plt.xlabel("Age")
         plt.ylabel("Annual Income")
         ax.set_zlabel("Spending Score")
         plt.show()
```



CONCLUSION

As we can see DBSCAN doesn't perform very well because the density in our data is not that strong. Label -1 means outliers so it will appear most as outliers. We may have performed better if we had had a bigger data. Even Mean Shift fails at it. So, we can see that k Means clustering is a preferred method for this analysis.