Case Study- Studying crime rate

Sahil Kumarwar-201060035
Risheel Chheda-201060030
Lavanya Tembhurne-201060023
Vishwesh Jadhav-201060071

Aim of Case Study

- The aim of this case study is to perform different types of clustering methods namely Hierarchical, K-means and DBSCAN clustering for the crime data.

- The aim of this case study is to identify the number of clusters formed and to draw inferences based on the number of clusters formed.

Why did we choose this topic ?

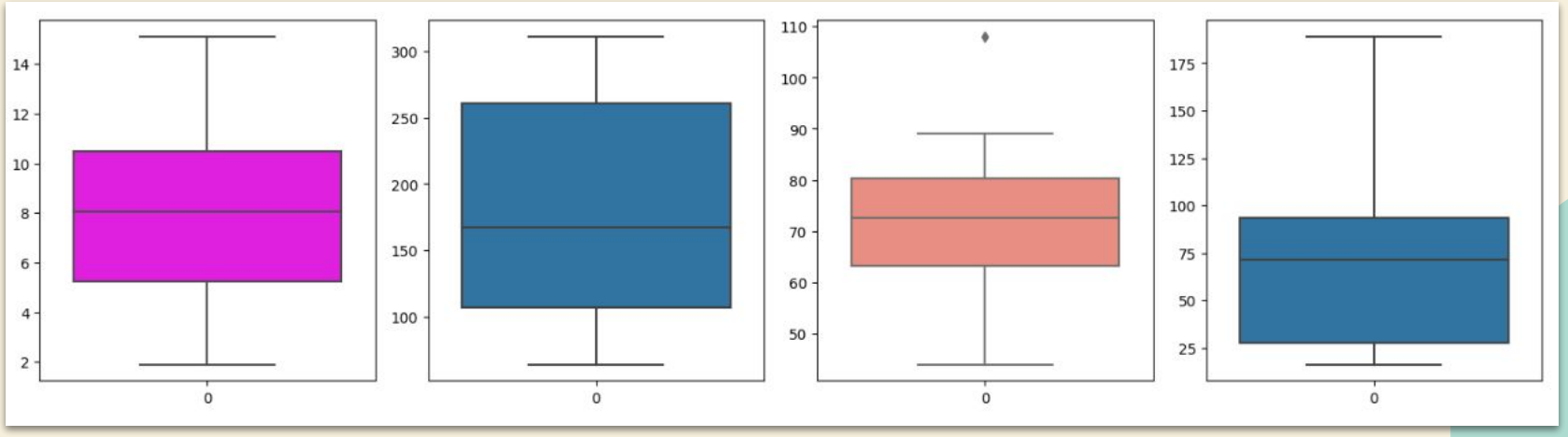
- Data science plays a huge role in Crime prediction.
- Crime rate is one of the most important indicators of safety and security in any country.
- Crime is a widespread issue among society and has significant impact on individuals,communities and the economy.
- In India, crime has been a major concern for policymakers, law enforcement agencies, and the general public. In recent years, data science and machine learning techniques have been used to analyze crime data and identify patterns that can help law enforcement agencies in their efforts to reduce crime rates.
- By analyzing vast amount of crime based data, data scientists are able to identify patterns, trends and correlations which can further be used to make accurate predictions about future criminal activities.
- Data science also helps in identifying patterns and relationships in crime data.
- This can help in preventing further crimes.

Steps

- Dataset: The dataset we will be using is the "crime_data_india.csv" dataset, which contains crime statistics for Indian states in a year. The dataset contains information on several types of crimes, including murder, assault, shoplifting and burglary.
- Data Cleaning and Preprocessing: Before we can apply any clustering algorithms, we need to preprocess the data. This includes removing any missing values, normalizing the data, and removing any irrelevant features.
- Hierarchical Clustering:Hierarchical clustering is a clustering technique that groups similar data points into clusters. It starts with each data point as a separate cluster and then combines them based on their similarity. The result is a dendrogram that shows the hierarchical relationship between the clusters.
- K-Means Clustering:K-means clustering is a clustering technique that divides data points into a predefined number of clusters based on their similarity. It starts by randomly selecting k centroids and then assigns each data point to the nearest centroid. It then updates the centroids based on the mean of the data points in each cluster.
- DBSCAN Clustering:DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering technique that groups data points based on their density. It defines clusters as areas of high density separated by areas of low density. Data points in low-density areas are considered as noise and are not assigned to any cluster.

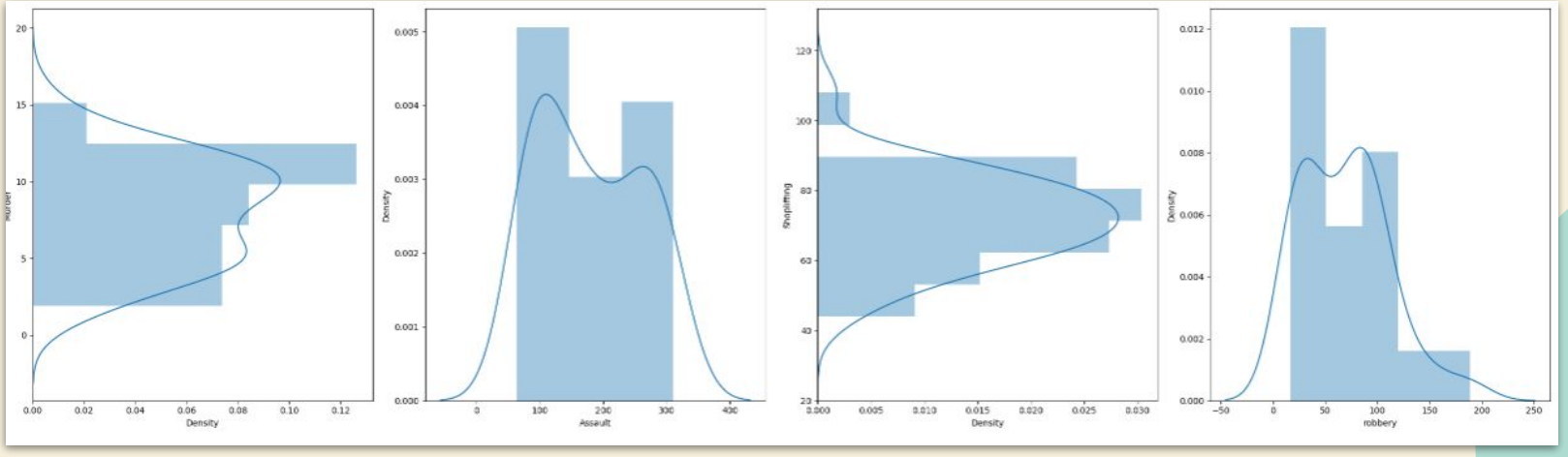
Checking for outliers using boxplot

```
fig, ax = plt.subplots(1, 4, figsize=(15,4))
sns.boxplot(crime.Murder, ax=ax[0],color='magenta')
sns.boxplot(crime.Assault, ax=ax[1])
sns.boxplot(crime.Shoplifting, ax=ax[2],color='salmon')
sns.boxplot(crime.robbery, ax=ax[3])
plt.tight_layout()
plt.show()
```


14
12
10
8
6
4
2
0
300
250
200
150
100
0
110
100
90
80
70
60
50
0
175
150
125
100
75
50
25
0

Checking for the normal distribution in the data using distplot

```
fig, ax = plt.subplots(1, 4, figsize=(25,7))
sns.distplot(crime.Murder, ax=ax[0], vertical=True)
sns.distplot(crime.Assault, ax=ax[1])
sns.distplot(crime.Shoplifting, ax=ax[2],vertical=True)
sns.distplot(crime.robbery, ax=ax[3])
plt.tight_layout()
plt.show()
```


Density
Assault
Density
Shoplifting
Density
robbery
Density

Hierarchical clustering

Hierarchical clustering is a clustering technique that groups similar data points into clusters. It starts with each data point as a separate cluster and then combines them based on their similarity. The result is a dendrogram that shows the hierarchical relationship between the clusters.

In our case, we will use hierarchical clustering to group Indian states based on their crime rates. We will use the complete linkage method to measure the distance between clusters. The complete linkage method calculates the distance between clusters based on the maximum distance between any two points in the clusters.

Applying normalization on data because different features have different range

```
def norm_func(i):
    x = (i-i.min())/(i.max()-i.min())
    return (x)
```

```
df_norm = norm_func(crime.iloc[:,1:])
```

Hierarchical clustering with 2 clusters

```
hc = AgglomerativeClustering(n_clusters=2, affinity = 'euclidean', linkage = 'single')
```

```
y_hc1 = hc.fit_predict(df_norm)
Clusters=pd.DataFrame(y_hc1,columns=['Clusters'])
```

```
crime['cluster'] = y_hc1
```

```
crime.groupby('cluster').agg(['mean']).reset_index()
```

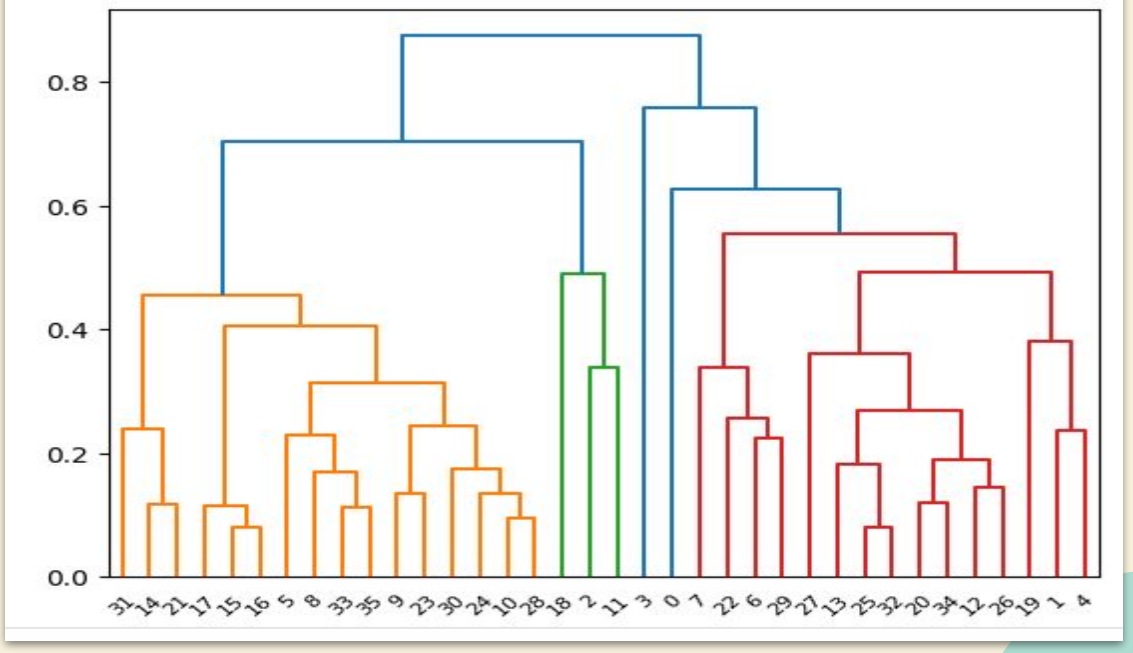
	cluster	Murder	Assault	Shoplifting	robbery
		mean	mean	mean	mean
0	0	7.788571	176.391429	71.402857	65.394286
1	1	15.100000	282.000000	88.000000	189.000000

```
cluster 0
The Members: Andhra Pradesh | Arunachal Pradesh | Assam | Chhattisgarh | Goa | Gujarat | Haryana | Himachal Pradesh | Jharkhand | Karnataka | Kerala | Madhya Pradesh | Maharashtra | Manipur | Meghal
Total Members: 35

cluster 1
The Members: Bihar
Total Members: 1
```

From the above result, it is evident that, this method is not giving proper result. So, we will try a different method for identify best number of clusters

Hierarchical clustering with average linkage

```
dendrogram = sch.dendrogram(sch.linkage(df_norm, method='average'))
```

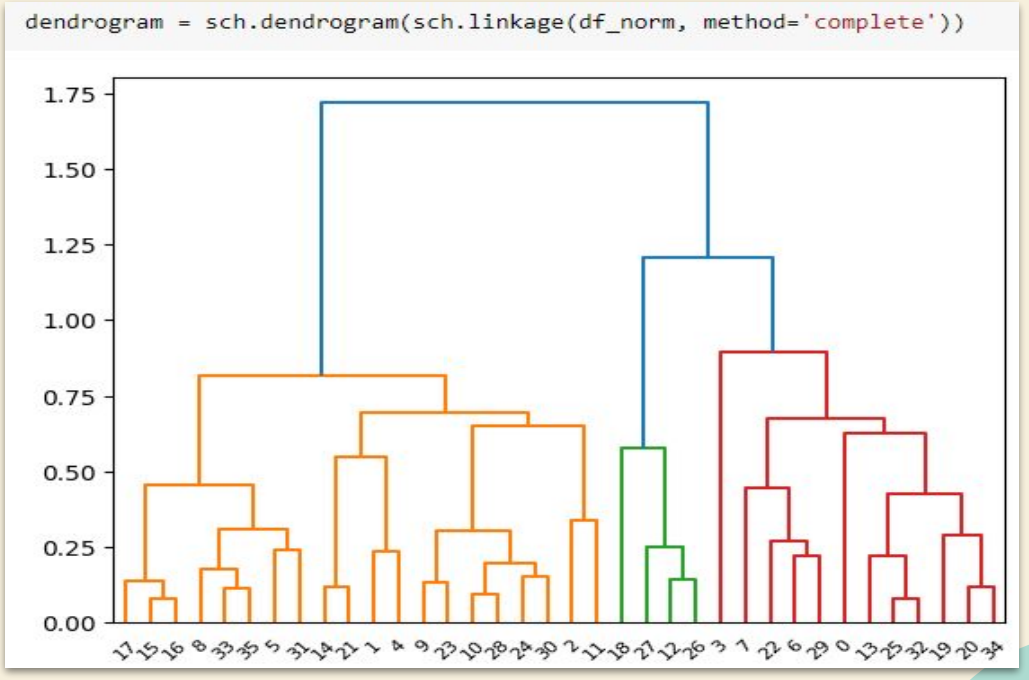


	cluster	Murder	Assault	Shoplifting	robbery
		mean	mean	mean	mean
0	0	10.450000	246.90000	78.043750	91.831250
1	1	9.866667	186.00000	48.666667	20.966667
2	2	15.100000	282.00000	88.000000	189.000000
3	3	4.737500	104.08125	69.025000	47.287500

```
cluster 0
The Members: Andhra Pradesh | Arunachal Pradesh | Chhattisgarh | Gujarat | Haryana | Madhya Pradesh | Maharashtra | Punjab | Rajasthan | Tamil Nadu | Uttar Pradesh | Uttarakhand | West Bengal | Chandigarh
Total Members: 16

cluster 1
The Members: Assam | Kerala | Odisha
Total Members: 3

cluster 2
The Members: Bihar
Total Members: 1

cluster 3
The Members: Goa | Himachal Pradesh | Jharkhand | Karnataka | Manipur | Meghalaya | Mizoram | Nagaland | Sikkim | Telangana | Tripura | Andaman and Nicobar Islands | Dadra and Nagar Haveli and Daman and D
Total Members: 16
```

Hierarchical clustering complete linkage mode

```
dendrogram = sch.dendrogram(sch.linkage(df_norm, method='complete'))
```

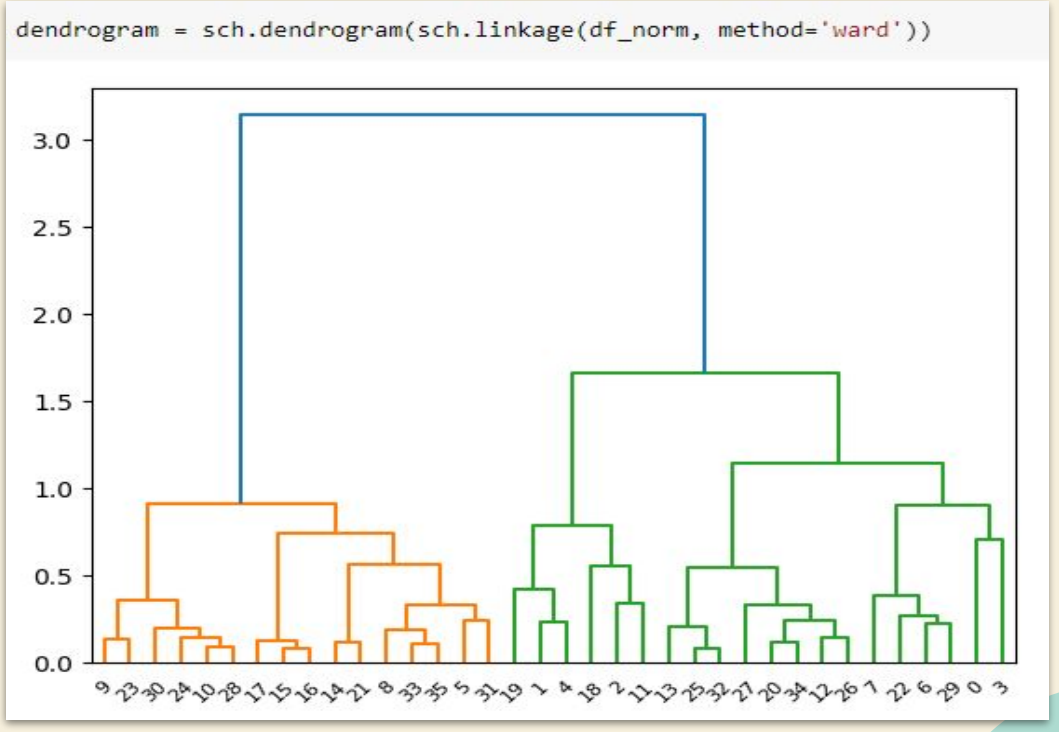


	cluster	Murder	Assault	Shoplifting	robbery
		mean	mean	mean	mean
0	0	5.610000	117.715000	67.620000	43.780000
1	1	10.745455	247.127273	82.790909	100.381818
2	2	10.550000	275.250000	59.000000	77.250000
3	3	15.100000	282.000000	88.000000	189.000000

```
cluster 0
The Members: Arunachal Pradesh | Assam | Chhattisgarh | Goa | Himachal Pradesh | Jharkhand | Karnataka | Kerala | Manipur | Meghalaya | Mizoram | Nagaland | Sikkim | Telangana | Tripura | Andaman and
Total Members: 20

cluster 1
The Members: Andhra Pradesh | Gujarat | Haryana | Maharashtra | Punjab | Rajasthan | Tamil Nadu | Uttar Pradesh | Chandigarh | Delhi | Jammu and Kashmir
Total Members: 11

cluster 2
The Members: Madhya Pradesh | Odisha | Uttarakhand | West Bengal
Total Members: 4

cluster 3
The Members: Bihar
Total Members: 1
```

Hierarchical clustering with centroid linkage method

```
dendrogram = sch.dendrogram(sch.linkage(df_norm, method='ward'))
```



	cluster	Murder	Assault	Shoplifting	robbery
		mean	mean	mean	mean
0	0	4.737500	104.081250	69.025000	47.287500
1	1	10.683333	212.566667	87.083333	130.450000
2	2	9.950000	200.000000	62.166667	32.016667
3	3	11.012500	289.375000	73.400000	93.300000

```
cluster 0
The Members: Goa | Himachal Pradesh | Jharkhand | Karnataka | Manipur | Meghalaya | Mizoram | Nagaland | Sikkim | Telangana | Tripura | Andaman and Nicobar Islands | Dadra and Nagar Haveli and Daman and
Total Members: 16

cluster 1
The Members: Andhra Pradesh | Bihar | Gujarat | Haryana | Tamil Nadu | Chandigarh
Total Members: 6

cluster 2
The Members: Arunachal Pradesh | Assam | Chhattisgarh | Kerala | Odisha | Punjab
Total Members: 6

cluster 3
The Members: Madhya Pradesh | Maharashtra | Rajasthan | Uttar Pradesh | Uttarakhand | West Bengal | Delhi | Jammu and Kashmir
Total Members: 8
```

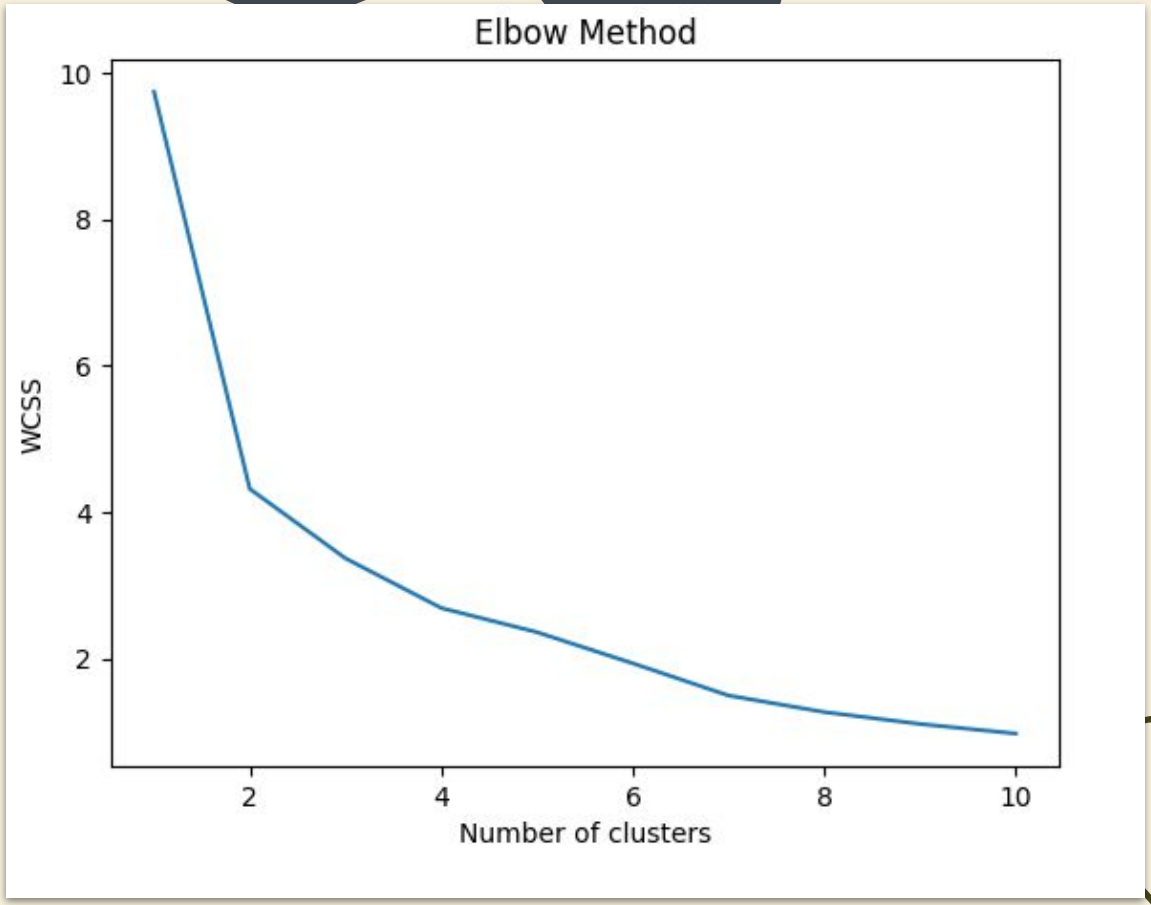
K-Means Clustering

K-means clustering is a clustering technique that divides data points into a predefined number of clusters based on their similarity. It starts by randomly selecting k centroids and then assigns each data point to the nearest centroid. It then updates the centroids based on the mean of the data points in each cluster.

In our case, we will use k-means clustering to group US states based on their crime rates. We will use the elbow method to determine the optimal number of clusters. The elbow method plots the within-cluster sum of squares (WSS) for different values of k and selects the value of k where the rate of decrease of WSS slows down.

Plotting elbow curve to determine the best number of clusters to be used in K-Means

```
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i,random_state=0)
    kmeans.fit(df_norm)
    wcss.append(kmeans.inertia_)

plt.plot(range(1, 11), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```


Elbow Method
10
8
6
4
2
WCSS
2
4
6
8
10
Number of clusters

Creating K-Means cluster with 4 groups

```
from sklearn.cluster import KMeans
clusters_new = KMeans(4, random_state=42)
clusters_new.fit(df_norm)
```

```
KMeans(n_clusters=4, random_state=42)
```

```
KM_label=clusters_new.labels_
```

```
crime['cluster'] = clusters_new.labels_
```

```
clusters_new.cluster_centers_
```

```
array([[0.7003367 , 0.89563653, 0.47256944, 0.41542817],
       [0.21496212, 0.16227227, 0.39101562, 0.17942965],
       [0.66540404, 0.60148448, 0.67317708, 0.66097279],
       [0.57575758, 0.50850202, 0.225     , 0.07481181]])
```

	cluster	Murder	Assault	Shoplifting	robbery
		mean	mean	mean	mean
0	0	11.144444	285.222222	74.244444	88.044444
1	1	4.737500	104.081250	69.025000	47.287500
2	2	10.683333	212.566667	87.083333	130.450000
3	3	9.500000	189.600000	58.400000	29.220000

DBSCAN Clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering technique that groups data points based on their density. It defines clusters as areas of high density separated by areas of low density. Data points in low-density areas are considered as noise and are not assigned to any cluster.

In our case, we will use DBSCAN clustering to group Indian states based on their crime rates. We will use the epsilon-neighborhood method to determine the optimal value of epsilon. The epsilon-neighborhood method plots the distance to the kth nearest neighbor for each data point and selects the value of epsilon where there is a significant change in the distance.

Applying DBSCAN clustering technique

```
from sklearn.cluster import DBSCAN
dbscan = DBSCAN(eps=0.3, min_samples=3)
dbscan.fit(df_norm)
```

```
DBSCAN(eps=0.3, min_samples=3)
```

```
dbscan.labels_
```

```
array([-1, -1, -1, -1, -1,  0,  0,  0,  0,  0,  0, -1,  1,  1,  0,  0,  0,
        0, -1,  1,  1,  0,  0,  0,  0,  1,  1,  1,  0,  0,  0,  0,  1,  0,
        1,  0])
```

```
cl=pd.DataFrame(dbscan.labels_,columns=['cluster_db'])
cl.head()
```

	cluster_db
0	-1
1	-1
2	-1
3	-1
4	-1

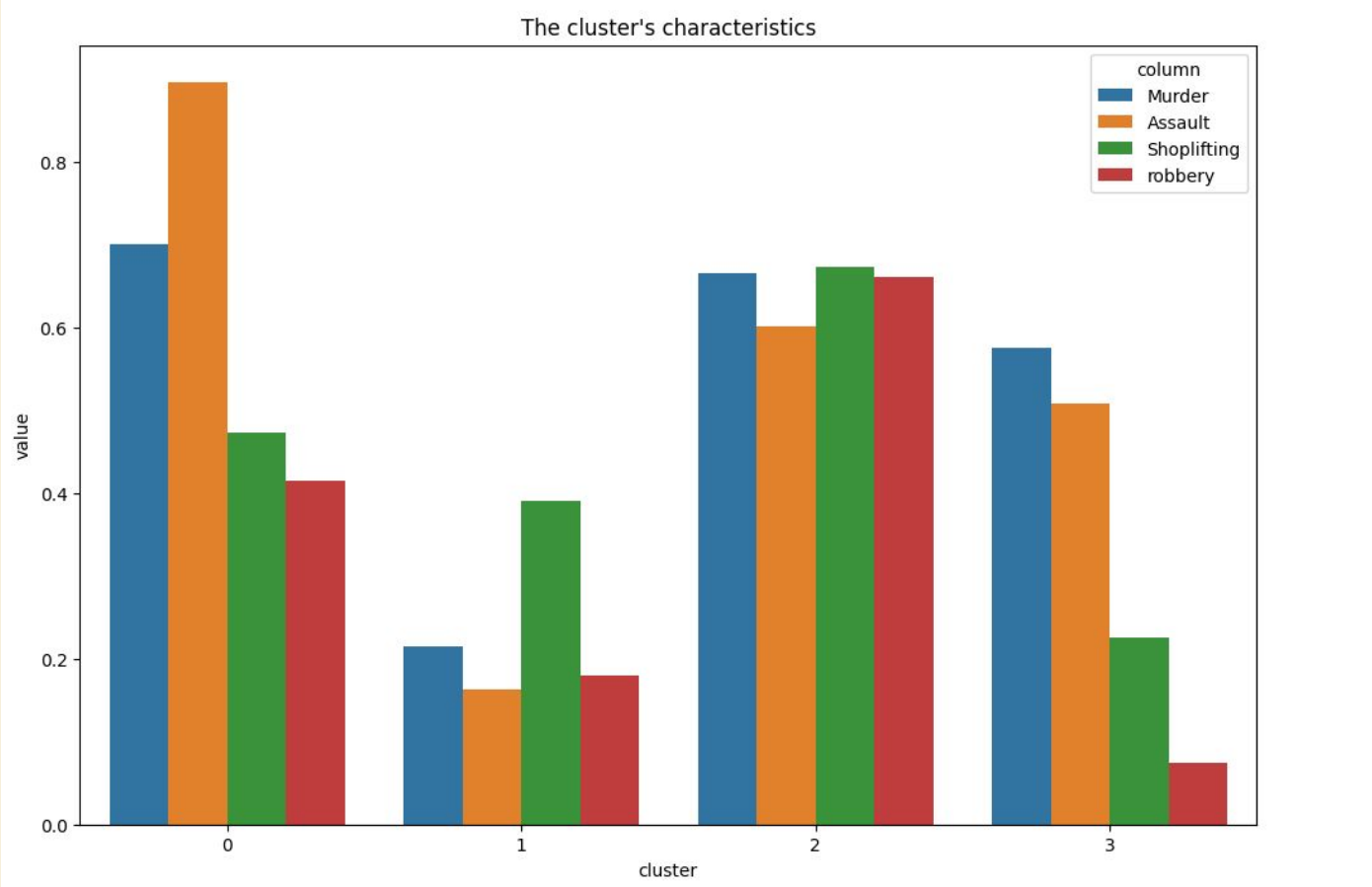
Calculating the Silhouette Score

```
t={'Model':['Single','Average','Complete','Ward','Kmeans','DBScan'],
  'Sillhouette score':[s1,s2,s3,s4,s5,s6]
  }
t=pd.DataFrame(t)
t
```

	Model	Sillhouette score
0	Single	0.317805
1	Average	0.347503
2	Complete	0.299844
3	Ward	0.366642
4	Kmeans	0.370526
5	DBScan	0.301718

Visualization

```
visualize = pd.DataFrame(clusters_new.cluster_centers_)
visualize = visualize.T
visualize['column'] = ['Murder','Assault','Shoplifting','robbery']
visualize = visualize.melt(id_vars=['column'], var_name='cluster')
visualize['cluster'] = visualize.cluster.astype('category')
plt.figure(figsize=(12, 8))
sns.barplot(x='cluster', y='value', hue='column', data=visualize)
plt.title('The cluster\'s characteristics')
plt.show()
```


The cluster's characteristics
column
Murder
Assault
Shoplifting
robbery
value
0.0
0.2
0.4
0.6
0.8
0
1
2
3
cluster

Inference

- Cluster0 has high murder& high assault rate with the K-Means clustering method
- Cluster1 has low murder& low assault rate with the K-Means clustering method
- Cluster2 has high shoplifting & robbery rate with the K-Means clustering method
- Cluster3 has low shoplifting & low robbery rate with the K-Means clustering method
- Four clusters are good to classify the crime rate states.
- From different models and visualizations, it is evident that the hierarchical clustering with average linkage method have the highest silhouette score.
- Higher the silhouette score, more far are the clusters separated from each other
- Four clusters are good to classify the crime rate states.
- From different models and visualizations, it is evident that the K-Means Clustering method have the highest silhouette score.
- Higher the silhouette score, more far are the clusters separated from each other.

Conclusion

In conclusion, the application of clustering algorithms such as Hierarchical, K-Means, and DBSCAN clustering to crime rate data in India has provided valuable insights into the distribution of crime rates across different states. Each algorithm has its own strengths and weaknesses, and the choice of algorithm depends on the specific needs and goals of the analysis.

Hierarchical clustering was able to identify a clear hierarchy of clusters based on crime rates, with some states clustering together due to similar crime rates. K-Means clustering was able to divide the states into a pre-defined number of clusters based on similarity, which could be useful for comparison and classification purposes. DBSCAN clustering was able to identify dense areas of crime rates and separate out low-density areas as noise.

Overall, the use of clustering algorithms in crime rate analysis can help inform policy decisions and resource allocation by identifying states with similar crime rates and highlighting areas that may require more targeted attention. Further analysis could include exploring the factors that contribute to high or low crime rates within clusters or identifying any temporal trends in crime rates over time.

Thank You