Data Preprocessing & EDA Steps:

1. We loaded the dataset *covtype_train*
2. We checked for null values, but since the data does not contain any null values, we did not drop any rows or columns
3. We plotted histograms of all the features and removed the skewed columns (columns having the same value of more than 99% of the data)
4. We checked for duplicate rows and found that more than 98% of the rows are duplicates. However, we decided not to drop those rows as this might change the sample distribution, thus affecting the clustering as well.
5. We converted the nominal features to ordinal (low:0, medium:1, high:2 etc) since most of the clustering algorithms work on the numerical features only.
6. We then plotted the correlation matrix and removed features having high correlation (> 0.6 correlation)
7. We then sampled 20% of the data to train the models (due to the resource limitation for most of the clustering algorithms), maintaining the cluster frequency of the original data
8. We used the target column to find the frequency of each cluster and saved it to map labels to the predicted clusters
9. Due to resource limitations for most of the clustering algorithms, we reduced the features using PCA, keeping the explained variance sufficiently high (more than 95%)

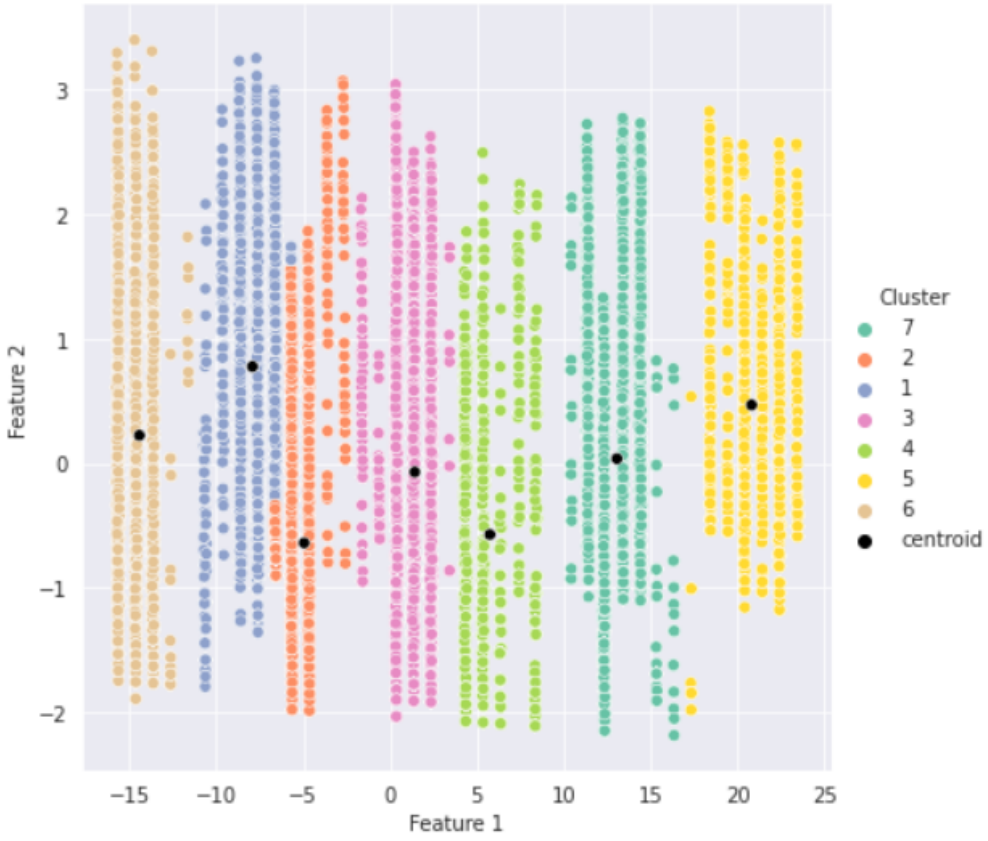
NOTE: We have used the PCA features for better visualizing the clusters.

K-Means

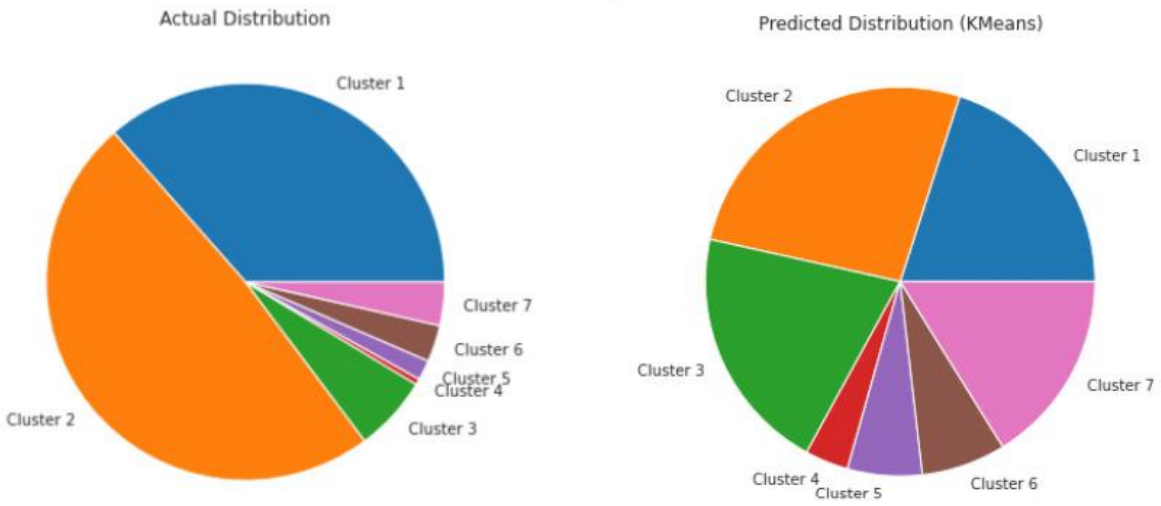
Centroids

	Aspect	Slope	Horizontal_Distance_To_Hydrology	Vertical_Distance_To_Hydrology	Horizontal_Distance_To_Fire_Points	Soil_Type	Wilderness
1	1.905038	0.360435	1.240517	1.308837	0.276414	31.302343	1.232868
2	0.715536	0.265576	0.873959	1.154603	0.455902	28.313153	0.664890
3	1.186124	0.351169	0.862270	1.194005	0.354310	21.956019	1.123978
4	0.760753	0.338009	0.663499	1.129513	0.405285	17.630283	1.022110
5	1.616339	0.515773	0.570048	1.191193	0.206488	2.598478	1.964777
6	1.449226	0.275395	1.258799	1.271632	0.376175	37.794702	0.777786
7	1.263662	0.432677	0.640587	1.170185	0.294682	10.335208	1.528314

Cluster Visualization



Cluster Distribution



Observations

On comparing the predicted distribution with the actual distribution, we have observed that:

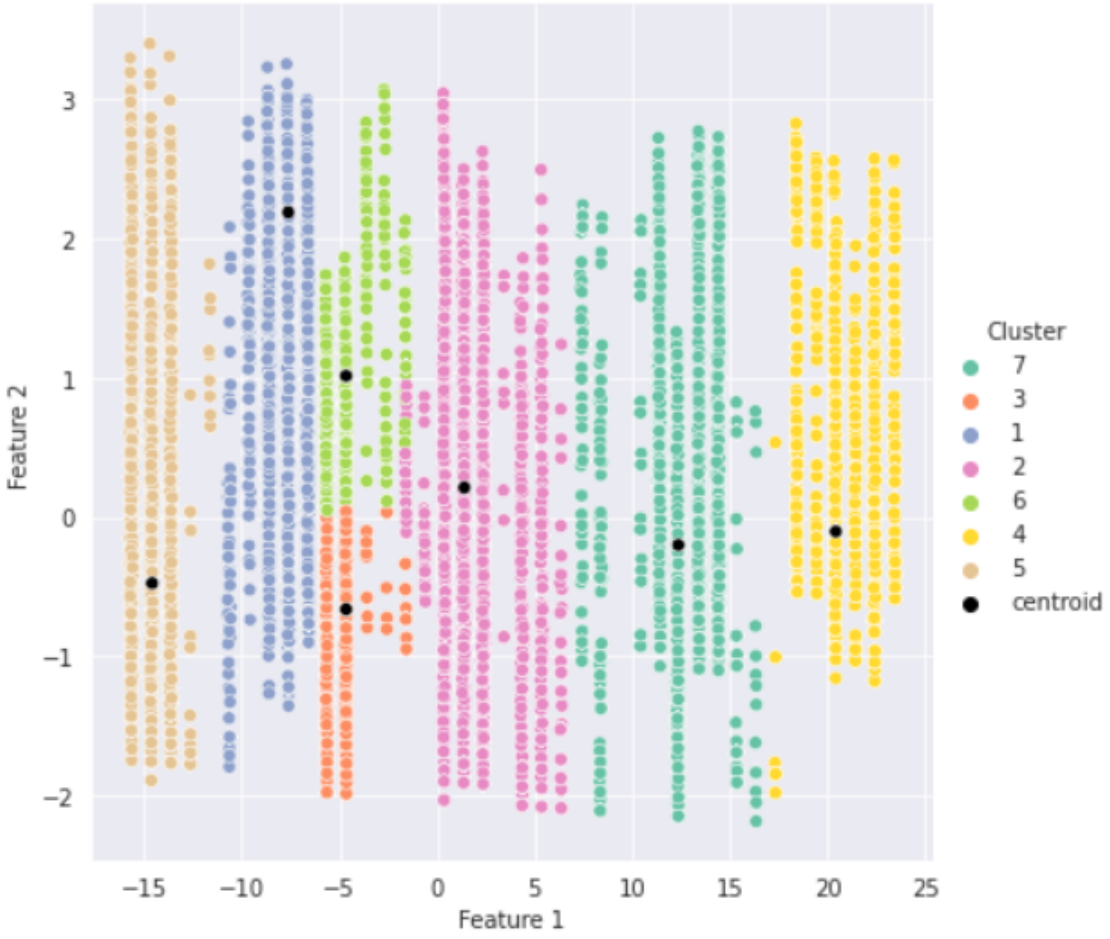
1) The order of the frequency distribution among the clusters remains almost the same i.e. Cluster 2 has the most frequency in the actual as well as in the predicted distribution, and so on.
2) However, the frequency distribution for a particular cluster when compared to the actual is not the same. In the actual distribution, about 50% of the data are in Cluster 2, while in the predicted only about 25% are in Cluster 2. Similarly, Cluster 7 is overrepresented in the predicted distribution in comparison with the actual distribution.

K-Medians

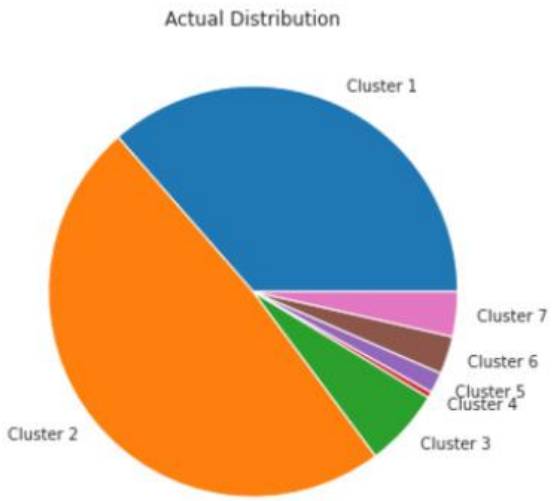
Centroids

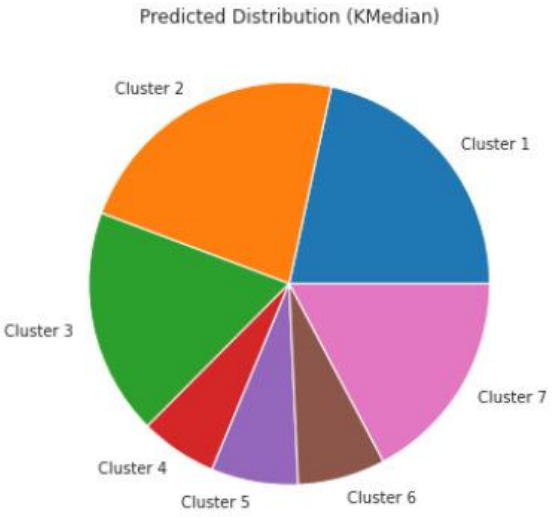
	Aspect	Slope	Horizontal_Distance_To_Hydrology	Vertical_Distance_To_Hydrology	Horizontal_Distance_To_Fire_Points	Soil_Type	Wilderness
1	3.087330	0.475536	1.536982	1.452775	0.084768	31.007871	1.898920
2	1.421910	0.373604	0.923090	1.222945	0.316421	21.978875	1.254273
3	0.696326	0.265587	0.863010	1.151418	0.457819	28.022867	0.663241
4	1.137807	0.467539	0.455420	1.133674	0.285104	2.975875	1.687166
5	0.864073	0.218378	1.112233	1.200415	0.471058	37.948324	0.447892
6	2.100665	0.399961	1.222799	1.323444	0.231672	28.040909	1.442949
7	1.065248	0.409235	0.604283	1.147884	0.329469	11.031699	1.396421

Cluster Visualization



Cluster Distribution





Observations

On comparing the predicted distribution with the actual distribution, we have observed that:

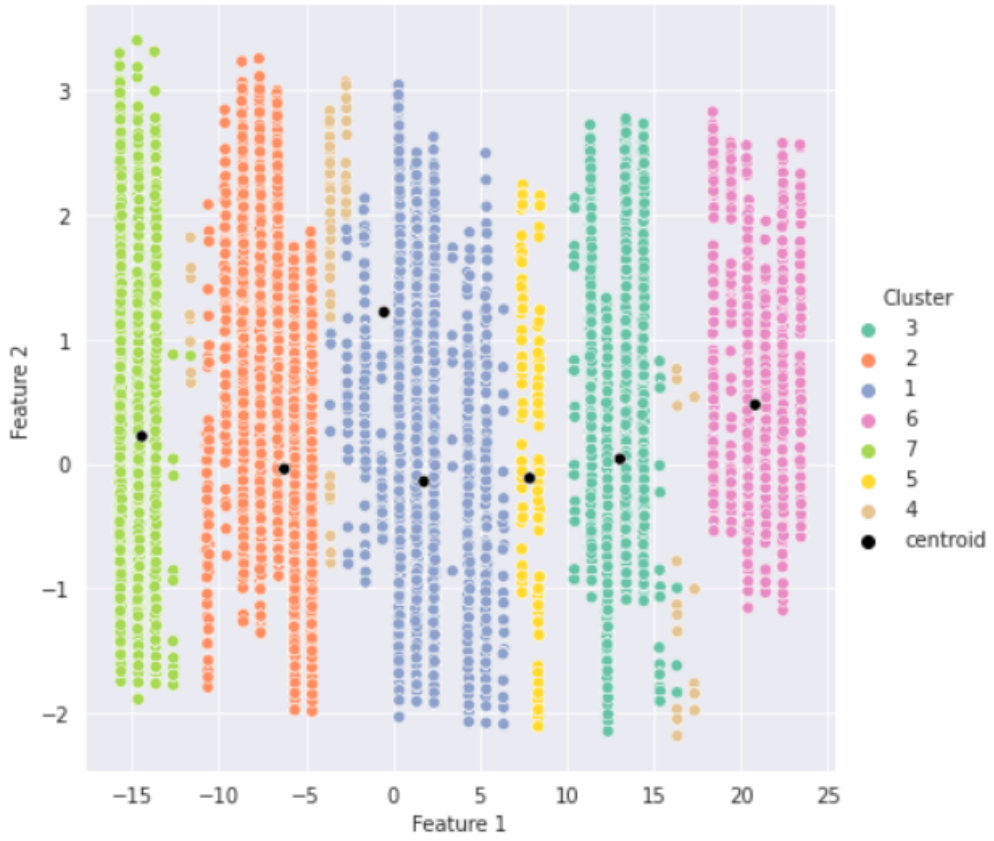
1) The order of the frequency distribution among the clusters remains almost the same i.e. Cluster 2 has the most frequency in the actual as well as in the predicted distribution, and so on.
2) However, the frequency distribution for a particular cluster when compared to the actual is not the same. In the actual distribution, about 50% of the data are in Cluster 2, while in the predicted only about 25% are in Cluster 2. Similarly, Cluster 7 is overrepresented in the predicted distribution in comparison with the actual distribution.

DBSCAN

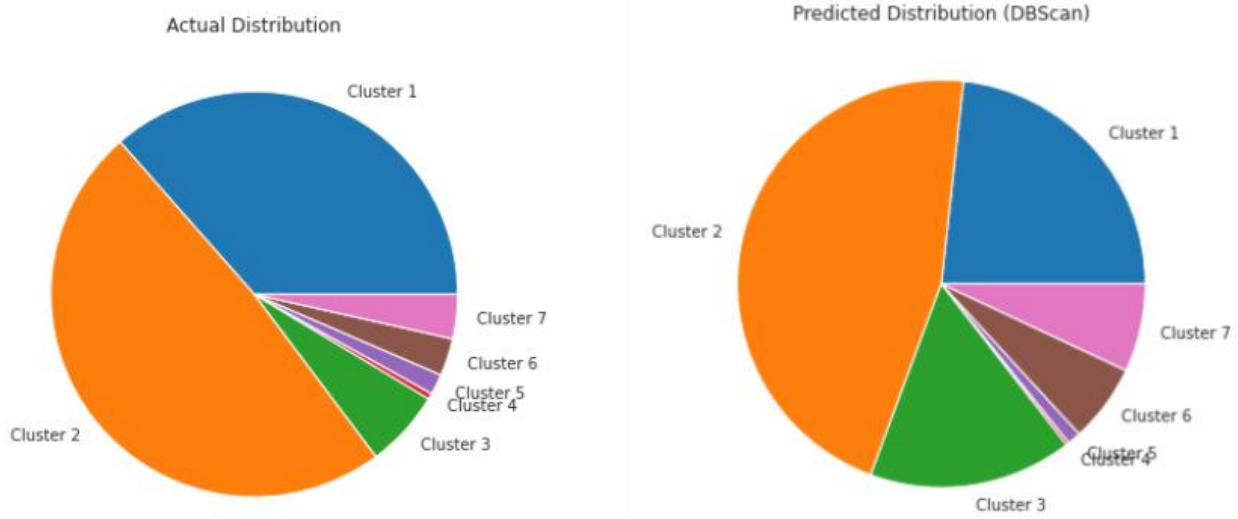
Centroids

	Aspect	Slope	Horizontal_Distance_To_Hydrology	Vertical_Distance_To_Hydrology	Horizontal_Distance_To_Fire_Points	Soil_Type	Wilderness
1	1.125020	0.347545	0.839371	1.185521	0.362736	21.606656	1.100889
2	1.217656	0.305358	1.029544	1.219827	0.380302	29.615922	0.903376
3	1.265125	0.432784	0.641070	1.170379	0.294468	10.340437	1.528964
4	2.266771	0.442231	1.179361	1.331929	0.188140	23.905563	1.663795
5	1.140526	0.387744	0.717121	1.170010	0.335605	15.533817	1.298289
6	1.621336	0.516317	0.571112	1.191775	0.205642	2.588153	1.967874
7	1.446361	0.275059	1.258267	1.271309	0.376676	37.804422	0.775892

Cluster Visualization



Cluster Distribution



Observations

On comparing the predicted distribution with the actual distribution, we have observed that:

1) The cluster having the most frequency i.e. Cluster 2 has almost the same frequency in the predicted distribution as in the actual distribution.
2) Cluster 3 is overrepresented in the predicted distribution in comparison with the actual distribution.

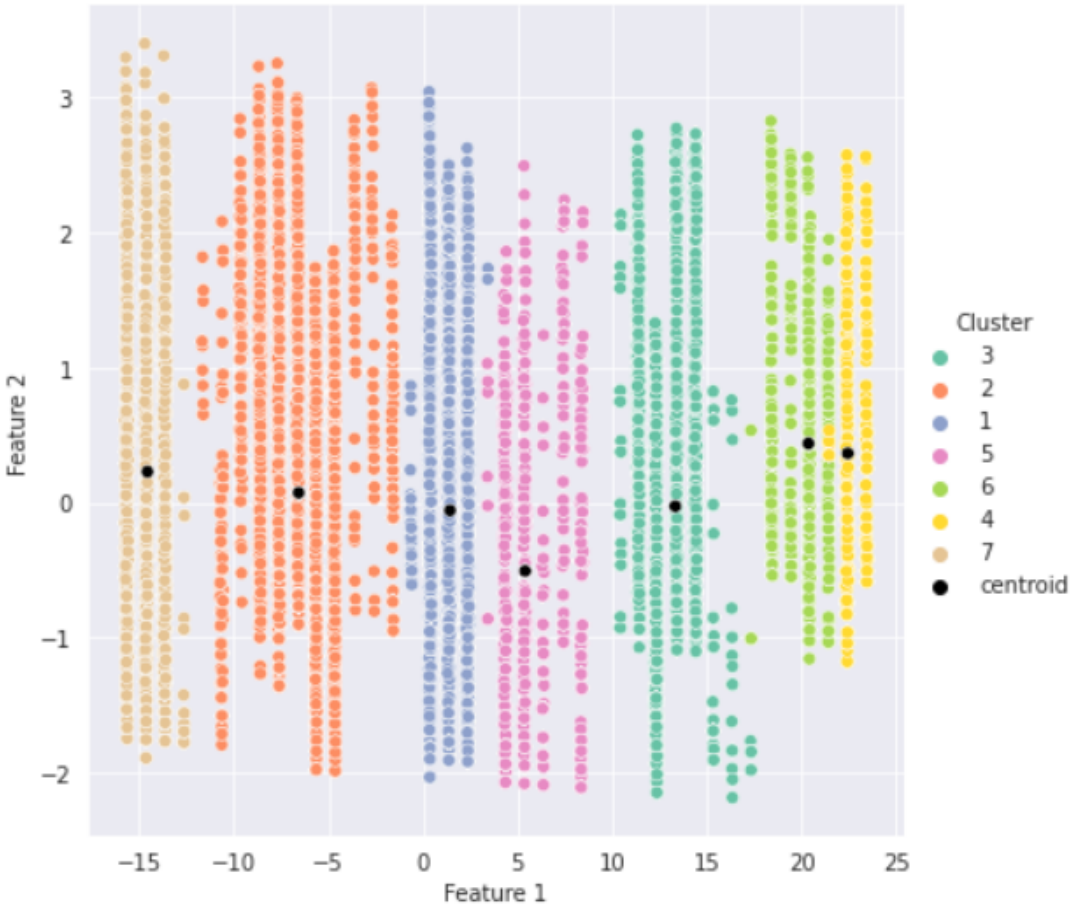
3) However, the order of the frequency distribution among the clusters remains almost the same i.e. Cluster 2 has the most frequency in the actual as well as in the predicted distribution, and so on.

GaussianMixture

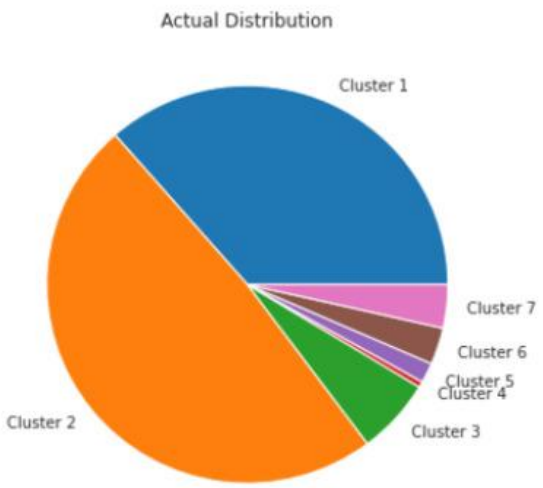
Centroids

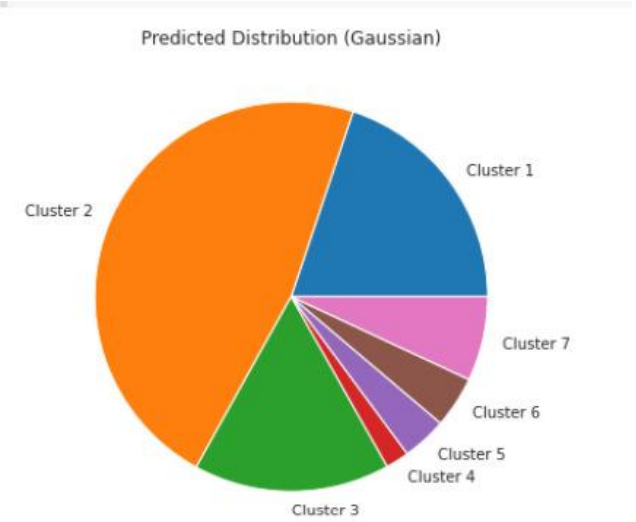
	Aspect	Slope	Horizontal_Distance_To_Hydrology	Vertical_Distance_To_Hydrology	Horizontal_Distance_To_Fire_Points	Soil_Type	Wilderness
1	1.196129	0.352147	0.864766	1.195221	0.352686	21.952906	1.129634
2	1.314182	0.312504	1.061089	1.232591	0.366088	29.945045	0.946775
3	1.210126	0.429268	0.621285	1.162856	0.302213	10.065730	1.506946
4	1.528446	0.517788	0.513540	1.175738	0.214009	0.962004	1.966815
5	0.816078	0.341032	0.685077	1.137312	0.397821	17.987226	1.041753
6	1.591478	0.510570	0.572884	1.189417	0.212289	3.041102	1.937204
7	1.453535	0.274953	1.262688	1.272544	0.376024	37.928778	0.776012

Cluster Visualization



Cluster Distribution





Observations

On comparing the predicted distribution with the actual distribution, we have observed that:

1) The cluster having the most frequency i.e. Cluster 2 has almost the same frequency in the predicted distribution as in the actual distribution.
2) Cluster 3 is overrepresented while cluster 1 is underrepresented in the predicted distribution in comparison with the actual distribution.
3) However, the order of the frequency distribution among the clusters remains almost the same i.e. Cluster 2 has the most frequency in the actual as well as in the predicted distribution, and so on.

Comparison with the gaussian based method

1. KMeans and KMedian both performed badly in comparison with the GaussianMixture. The clusters of GuassianMixture were closer to the actual ones in comparison with KMeans and KMedians. The primary reason behind it is that both KMeans and KMedian did not perform well with clusters of varying sizes and density (variance), while the GaussianMixture takes into account the variance and can easily handle uneven size (even oblong) clusters.
2. DBScan performs almost the same as GaussianMixture because DBScan calculates the clusters using the density distribution of the clusters and works well with uneven size clusters. Also, DBScan takes noise in data into account hence sometimes (with small dataset) it performs better than GaussianMixture.

Question 2

We calculated the score of each of our clustering algorithms used in 1st question:

```
f1_score(target_df['target'], pca_covtype_df['Cluster'], average='micro')
```

We used ‘micro’ as an average parameter in f1_score because it sums up the individual true positives, false positives, and false negatives of the system for different sets and then applies them to get the statistics.

KMean: 0.252

KMedian : 0.175

DBSCAN : 0.474

Gaussian Mixture : 0.466

It is clearly visible that DBSCAN and GuassianMixture are performing better than K-Mean and K-Median. Then, we decided that we will use Gaussian Mixture, even though DBSCAN is performing better than GuassianMixture because the time complexity of DBSCAN is much, much higher (computationally expensive).

Now, for creating clusters using GuassianMixture we splitted the given data in an 80:20 ratio (80% for model fitting and 20% for testing). After training the model, we have saved it as a pickle file and loaded the same for testing.