# DATA MINING
# Assignment 3
# REPORT


**PREPARED BY**

| | |
|---|---|
| Aditi Gupta | 2019292 |
| Mansi Singhal | 2019370 |

# Preprocessing

- Since in our dataset all of the features have a very small limited range, therefore, there are no outliers in the data.
- There are no null values in the dataset.
- First we converted categorical data to numerical form using ordinal encoder since the data was ordinal
- We have used MinMax Scaling.
- To reduce the dimensionality we have computed `Distance_To_Hydrology` from the vertical and horizontal distances provided.
- Since the data had class imbalance we have used random undersampling to make the data balanced.

# Question 1

We have used K-means, birch, Agglomerative Clustering and Gaussian Mixture clustering.

### Part 1

- K means

```
#centroids
cluster_centers = kmeans.cluster_centers_
cluster_centers
```

```
array([[3.73639306e-02, 1.75639894e-01, 2.82288909e-01, 1.00000000e+00,
        9.97793469e-01, 1.60982339e-15, 1.33274492e-01, 9.99117387e-01,
        1.92032910e-01],
       [7.12612613e-01, 1.29909910e-01, 1.14954955e-01, 1.00000000e+00,
        9.96216216e-01, 1.15405405e-01, 8.31919612e-01, 6.38198198e-01,
        2.68407794e-01],
       [5.42342342e-01, 1.67767768e-01, 1.58958959e-01, 1.00000000e+00,
        9.95795796e-01, 1.52655666e-16, 7.02733503e-01, 6.80680681e-03,
        2.14540327e-01],
       [1.87450980e-01, 8.89281046e-01, 2.36862745e-01, 9.33725490e-01,
        9.99607843e-01, 8.82352941e-03, 2.02513826e-01, 8.85751634e-01,
        2.02831062e-01],
       [6.19251642e-01, 2.32790631e-01, 9.11168238e-02, 1.00000000e+00,
        1.00000000e+00, 6.25107112e-01, 6.78421550e-01, 1.54241645e-02,
        2.00294794e-01],
       [6.97773412e-01, 9.08644401e-01, 9.11918795e-02, 9.87721022e-01,
        1.00000000e+00, 1.16650295e-01, 7.66006750e-01, 4.92305174e-01,
        2.68337772e-01],
       [3.60664245e-01, 1.64677391e-01, 1.29735340e-01, 1.00000000e+00,
        1.00000000e+00, 2.95796575e-02, 2.04795550e-01, 7.16658018e-01,
        2.04806637e-01]])
```

- Birch

```
clf_br = NearestCentroid()
clf_br.fit(X, y_predict_br)
print(clf_br.centroids_)
```

```
[[0.67812759 0.88304336 0.08933996 0.99544325 1.         0.1526512
  0.73642003 0.43938139 0.26031467]
 [0.15371361 0.46322044 0.23060012 1.         1.         0.00650624
  0.15318342 0.90932858 0.19772007]
 [0.5955431  0.13764345 0.12590072 1.         1.         0.12560048
  0.69097842 0.35528423 0.23056932]
 [0.5952381  0.         0.71428571 1.         0.         0.
  0.83699634 0.33333333 0.24708201]
 [0.12932605 0.9708561  0.5428051  0.         0.99453552 0.00546448
  0.26691887 0.91621129 0.21693963]
 [0.48396095 0.11854951 0.05020921 1.         1.         0.94769874
  0.51464435 0.0069735  0.19844564]
 [0.         0.06666667 0.66666667 1.         0.         0.
  0.13846154 1.         0.15016327]]
```

- Agglomerative Clustering

```
clf_ac = NearestCentroid()
clf_ac.fit(X, y_predict_ac)
print(clf_ac.centroids_)
```

```
[[0.2265014  0.92238454 0.24199498 0.91412129 0.99955733 0.01527224
  0.21853327 0.85878707 0.21025706]
 [0.68615225 0.21693946 0.11327618 1.         0.99259548 0.1159392
  0.78254092 0.61626396 0.25564124]
 [0.68461265 0.93159204 0.08795309 1.         1.         0.18390192
  0.75513914 0.38379531 0.26465473]
 [0.61603376 0.13888889 0.09563994 1.         1.         0.6149789
  0.68349021 0.00246132 0.1958502 ]
 [0.07192682 0.22801601 0.2436821  1.         1.         0.
  0.13501341 0.99771298 0.1959957 ]
 [0.51412819 0.12956582 0.16471399 1.         1.         0.
  0.69942922 0.         0.20586308]
 [0.3637621  0.17427386 0.15260489 1.         1.         0.
  0.20661773 0.66666667 0.20590097]]
```

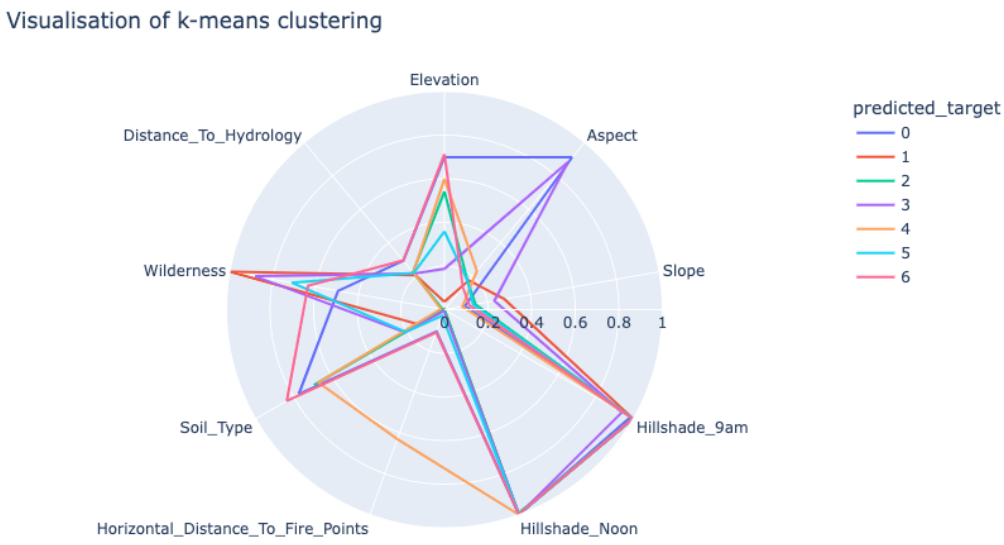
- Gaussian Mixture Model

```
clf_gm = NearestCentroid()
clf_gm.fit(X, y_predict_gm)
print(clf_gm.centroids_)
```

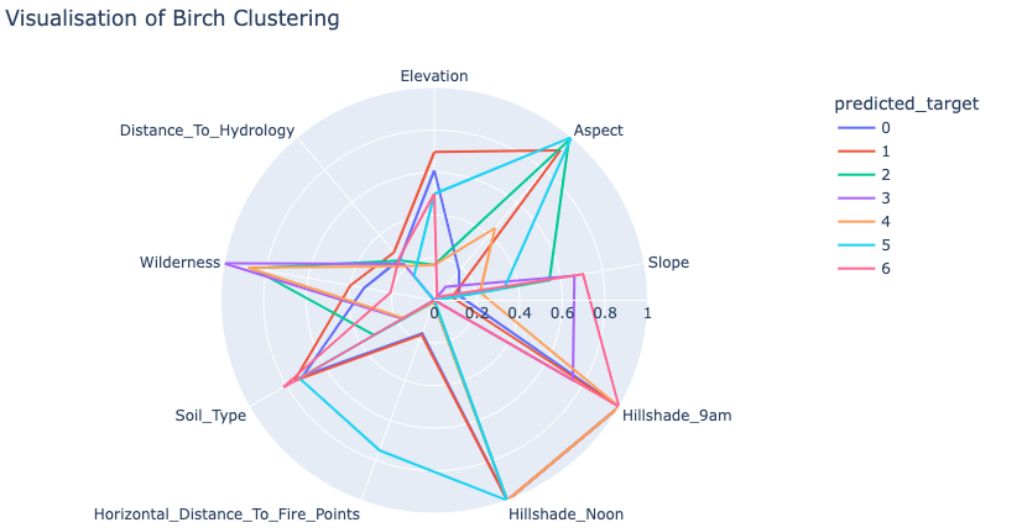
```
[[0.62215478 0.31562974 0.09307031 1.         1.         0.6107739
  0.68318353 0.01846232 0.20121926]
 [0.15151515 0.87878788 0.55892256 0.0959596  0.8989899  0.
  0.2951308  0.87542088 0.2117546 ]
 [0.63387001 0.44156231 0.14967203 1.         1.         0.20192308
  0.75489656 0.66666667 0.30975647]
 [0.18161138 0.43358197 0.21467492 1.         1.         0.
  0.17212978 0.88261123 0.1986639 ]
 [0.75       1.         0.52083333 0.0625     1.         0.21875
  0.83173077 0.3125     0.27441283]
 [0.33333333 0.1894452  0.21470456 1.         1.         0.
  0.67488984 0.         0.1920147 ]
 [0.73589647 0.43686393 0.0660887  1.         1.         0.
  0.7651715  0.3246639  0.2197706 ]]
```

## Part 2

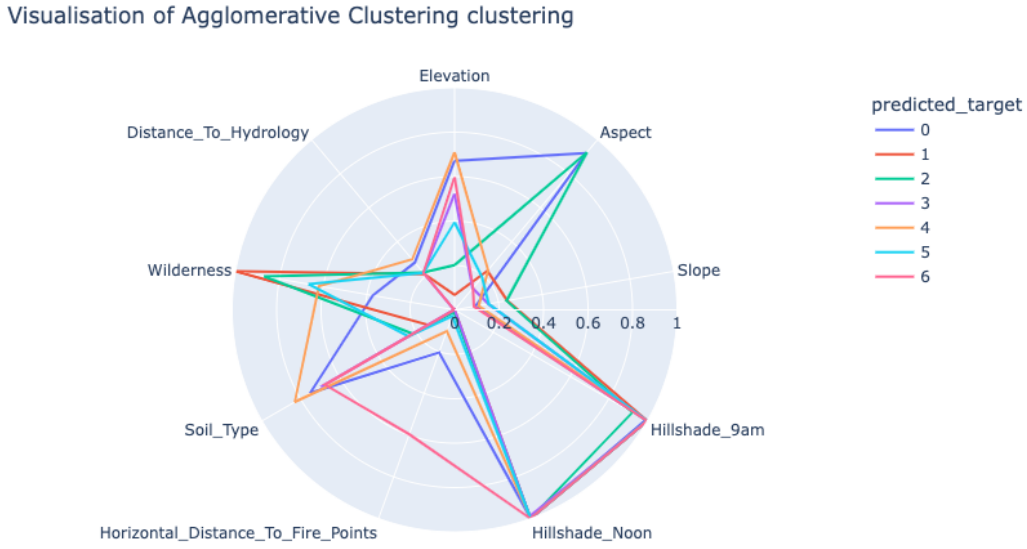
- K means



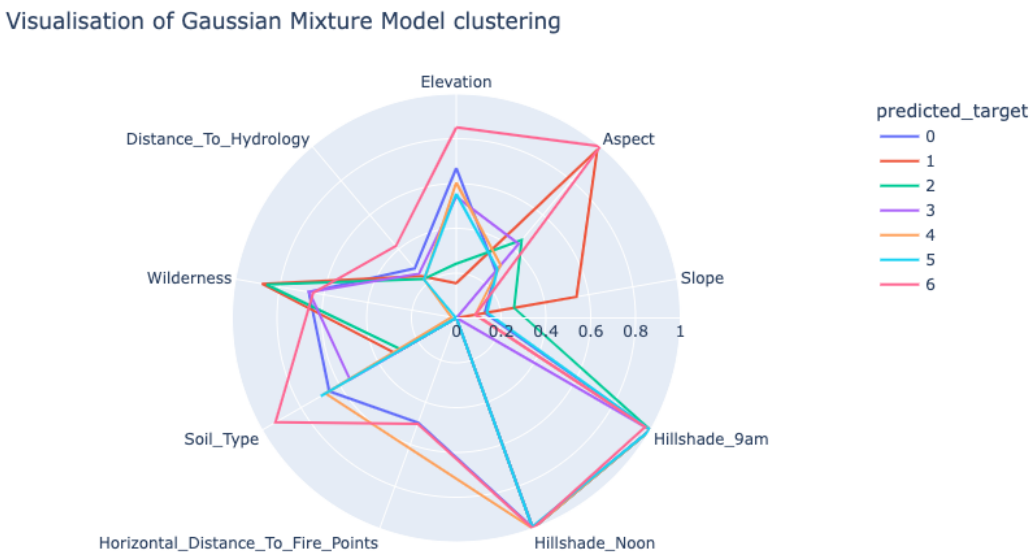
- Birch




- Agglomerative Clustering



- Gaussian Mixture Model




## Part 3

For part 3, we have used AMI score to compare the cluster distributions with the true label count distribution. Then we have used our mapping algorithm to assign target values to each cluster and calculated the f1 score for each model.

- K means

```
AMI score for K-means Clustering Model and target label:
0.3271954656782891
F1 score for  K-means model:
0.39383861774656026
```

- Birch

```
AMI score for Birch Clustering and target label:
0.29693917836667943
F1 score for Birch Clustering Model:
0.21335063105938967
```

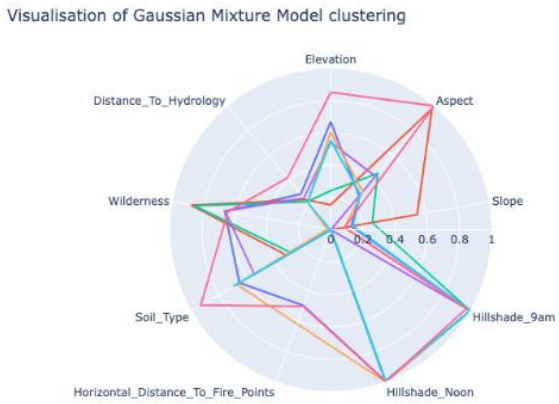
- Agglomerative Clustering

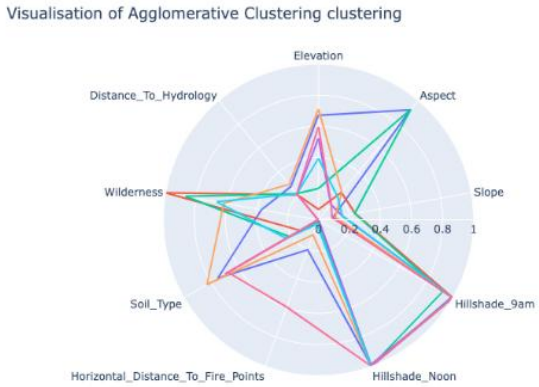
```
AMI score for Agglomerative Clustering and target label:
0.34335008076215795
F1 score for Agglomerative Clustering Model:
0.3956528769418069
```

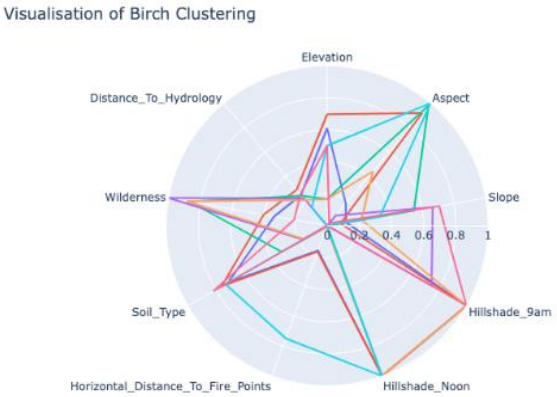
- Gaussian Mixture Model

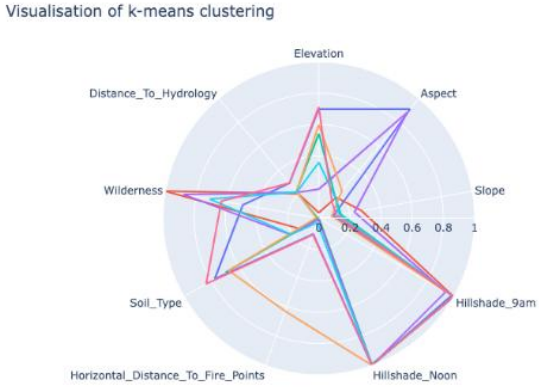
```
AMI score for Gaussian Mixture and target label:
0.3289733493692009
F1 score for Gaussian Mixture Model:
0.2206982189791166
```

## Part 4




When we see the graphs of Gaussian model and Agglomerative Clustering, we can observe that Gaussian model has comparatively formed bigger clusters compared to Agglomerative. The similarity between the two models according to AMI is `0.5574392687684919`.

When we see the graphs of Gaussian model and birch Clustering, we can observe that both the models have similar sized clusters. However, the overlap between clusters in Birch is more than clusters of Gaussian model.The similarity between the two models according to AMI is `0.46241293720836163`.

When we see the graphs of Gaussian model and K-means Clustering, we can observe that Gaussian model has comparatively formed bigger clusters compared to K-means and the overlap between clusters in K-means is more than clusters of Gaussian model. The similarity between the two models according to AMI is `0.5700402507139478`.

Clusters formed by k-means are the most similar to Gaussian clusters as compared to the other two clustering algorithms.

# Question 2

In preprocessing we have transformed categorical data into numeric using Ordinal Encoder.

We have also reduced the dimensionality of the dataset by introducing a new column: distance_to_hydrology.

Out of all the models KMeans performed the best for us so we went with that as our final model.

For mapping the predicted clustered labels with the actual dataset's labels, we found all the possible renamed predicted labels and took the best renaming out of them according to F1 scoring.

# References


- https://towardsdatascience.com/clustering-with-more-than-two-features-try-this-to-explain-your-findings-b053007d680a
- https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68
- https://medium.com/@evgen.ryzhkov/5-stages-of-data-preprocessing-for-k-means-clustering-b755426f9932
- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_mutual_info_score.html
- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html