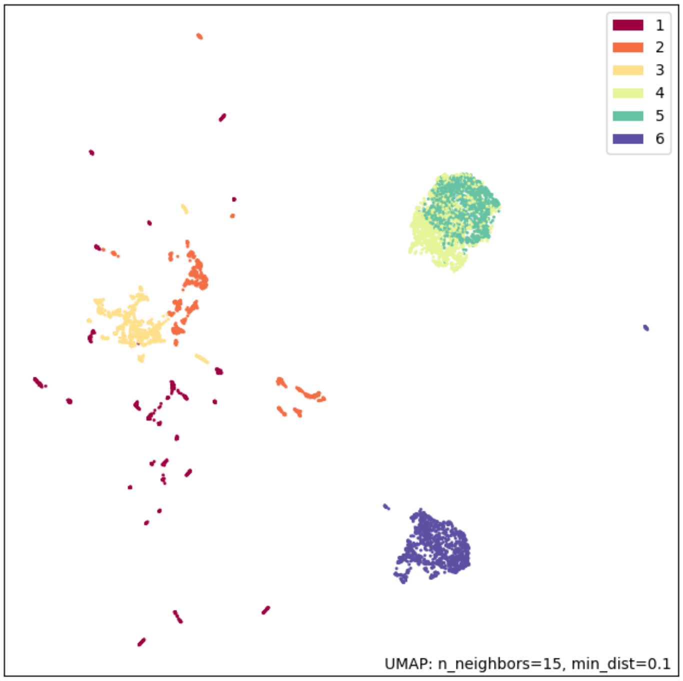
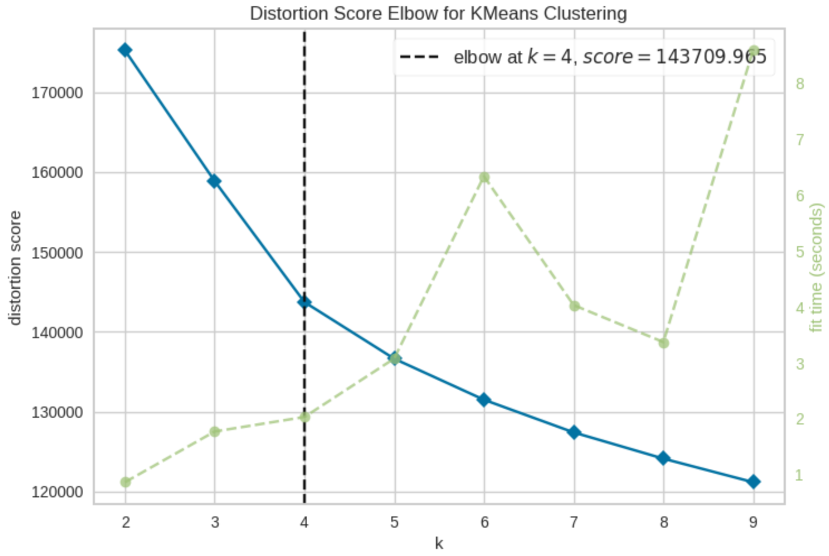
Ans 1 :

Firstly, I have downloaded the dataset and then did some pre-processing to convert the text file of train inputs/outputs and test inputs/outputs to pandas dataframe. Then I am using the inbuilt umap library for the umap plotting. Then for K-means, I am using the KElbowVisualizer library to plot the elbow curve. The optimal K came out to be 4. Then for spectral clustering, I am using the silhouette score as a metric to determine the optimal number of clusters. The optimal clusters came out to be 5. Then, I am running K-means for 4 clusters and spectral clustering for 5 clusters and then I am using silhouette score as an evaluation metric.

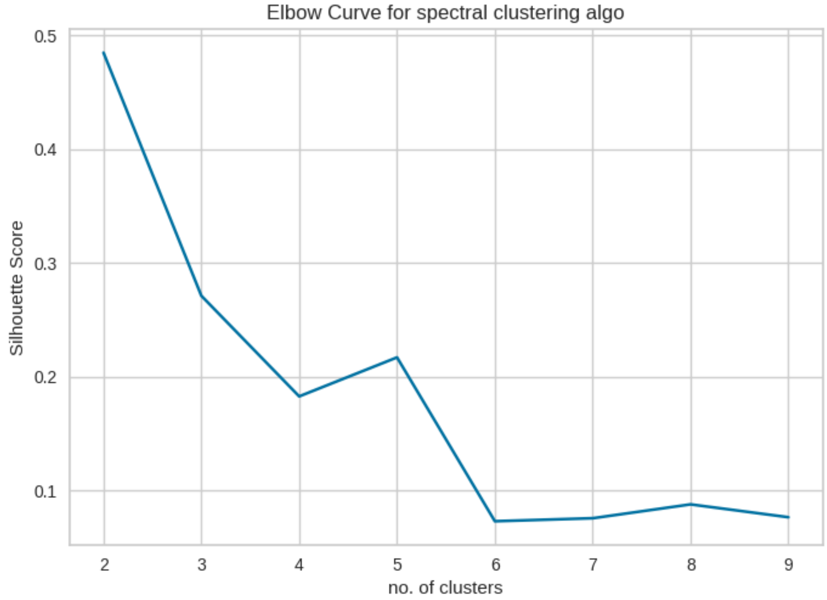
UMAP plot :



K-means elbow curve :



Spectral clustering silhouette score :



K-Means silhouette score : **0.187**
Spectral clustering silhouette score : **0.216**

Ans 2 :
Random forest classifier accuracy : 98.14%
Decision tree with bagging accuracy : 98.14%
Decision tree with ADABoost accuracy : 92.6%

The accuracy of the random forest classifier came out to be 98.14% which is very high. Random forest classifier combined multiple decision trees due to which it is more accurate. The algorithm did not take much time to run because the data was less otherwise random forest classifier is a computationally expensive algorithm.

The accuracy of the decision tree with bagging came out to be the same as random forest classifier i.e. 98.14%. This algorithm combined multiple decision trees results and used majority voting to get the final prediction. The accuracy came high because we're using multiple trees for decision making instead of one tree. The algorithm is computationally expensive but since the dataset was small the algorithm didn't take much time.

The accuracy of the decision tree with ADAboost came out to be 92.5%. The reason of low accuracy as compared to other models could be due to the fact that the dataset might have outliers. This algorithm is also computationally expensive.

Ans 4 :
In this problem, first of all I am creating a grid search function which is used for getting the training and testing accuracy list for each algorithm i.e. ID3, C4.5, C5.0 and CART. I am calculating training and testing accuracy for 10 different complexity parameters. Then I am

creating a dataframe displaying the training and testing accuracies for each algorithm. Then I am training different classifiers for each algorithm and obtaining the ccp_alpha values. Now, for these ccp_alpha values I am obtaining the training and testing accuracy and then displaying it on the dataframe.

ID3 classifier : It was observed that as I increase the max_depth the training accuracy also increases. The test accuracy kind of decreases as I move from max_depth = 2 to 26 but then starts increasing.

C4.5 classifier : It was observed that as I increase the max_depth the training accuracy also increases. The test accuracy on the other hand is lower as I increase the max_depth. C4.5 had higher accuracy as compared to ID3.

C5.0 classifier : It was observed that as I increase the max_depth the training accuracy also increases. The test accuracy on the other hand follows a similar pattern as that of ID3. The accuracy for C5.0 came out to be lower than C4.5 for the given dataset.

CART classifier : It was observed that as I increase the max_depth the training accuracy also increases. The training accuracy reached its maximum earlier than it did for the other algorithms. The CART algorithm test accuracy came out to be better than C5.0 and ID3 but not better than C4.5 for the given dataset.

Deviations from inbuilt :
For the ID3 classifier, using the cost complexity pruning path the max accuracy increased by ~0.8% as compared to earlier ID3 classifier. Similarly for C4.5, the max accuracy increased by 0.5% from 89.09% to 89.6% while classifying using cost complexity pruning path. For C5.0, the accuracy increased by approximately 0.8%. For the CART algorithm, the max accuracy increased by ~0.4%. It was observed that the max accuracy deviation is not significant. Also, the difference was highest for both ID3 and C5.0 and lowest for CART algorithm.