MACHINE LEARNING AND PATTERN RECOGNITION

**LAB WEEK 6**

1. Import the following libraries:

* *Pandas*
* *Matplotlib.pyplot*
* *Numpy*

1. Import the training and test datasets using *pandas*.
2. Plot the training and test data sets having ‘Rating’ on the y-axis and ‘Duration’ on the x-axis. Also, label the data into the two classes ‘Action’ and ‘Drama’ and annotate each data point with the name of the movie:

A diagram of a train set

Description automatically generated

1. Model Development and Evaluation:

* Import *KNeighborsClassifier* from *sklearn.neighbors*
* Import *confusion\_matrix, ConfusionMatrixDisplay* from *sklearn.metrics*
* Define a list: *knn\_neighbors* = [1, 3, 5, ……., 499]
* for each *k* in *knn\_neighbors*:

1. apply the k-NN algorithm on the train set

(For Reference: <https://www.geeksforgeeks.org/k-nearest-neighbor-algorithm-in-python/>)

1. use the test dataset to the find the predictions set
2. use the test dataset and the predictions set to get the confusion matrix for each *k*
3. use the Confusion Matrix to calculate the accuracy for each *k*:

Accuracy = (True Negatives + True Positives) / Total no. of data points

1. Plot Accuracy vs No. of Neighbors (*k*):

A graph showing a number of neighbors

Description automatically generated

1. Find the value of *k* for which the accuracy is the maximum. Use the *max()* and *list.index()* to find that value of *k* and the corresponding maximum accuracy.
2. For that value of *k*, display the confusion matrix and calculate the performance metrics i.e. precision, recall, overall\_precision, overall\_recall and F\_score.

A blue and pink squares with white text

Description automatically generated

DELIVERABLES:

1. Train dataset plot annotated with movie names
2. Test dataset plot annotated with movie names
3. Accuracy vs No. of Neighbors plot
4. Final confusion matrix and performance metrics (screenshot)