Pattern Recognition and Machine Learning
Lab - 8 Assignment
Feature Selection

Guidelines for submission

1. Perform all tasks in a single colab file.
2. Create a report regarding the steps followed while performing the given tasks. The report should not include excessive unscaled preprocessing plots.
3. Try to modularize the code for readability wherever possible
4. In-Lab Submission: Google Classroom

Naming convention: **RollNo_InLab8_Submission.ipynb**

Submit the In-lab work after the lab itself.

5. Submit the colab[.ipynb], python[.py] and report[.pdf] files in the Google Classroom with proper naming conventions.

Please follow the naming conventions:

RollNo_LabAssignment_8.ipynb

RollNo_LabAssignment_8.py

RollNo_LabAssignment_8.pdf

Not following this naming convention will make it difficult for evaluation. Kindly follow them.

6. Plagiarism will not be tolerated
7. For the final submission, you may take up to 4 extra days. No penalty will be there.

Guidelines for Report:

1. The report should be to the point. Justify the space you use!
2. Explanations for each task should be included in the report. You should know the ‘why’ behind whatever you do.
3. Do not paste code snippets in the report.

[In-Lab Submission: Question 1, first 4 subparts]
Question 01: [60 marks]: Sequential Feature Selection

Sequential feature selection algorithms are a family of greedy search algorithms that are used to reduce an initial d-dimensional feature space to a k-dimensional feature subspace where k < d. The motivation behind feature selection algorithms is to automatically select a subset of features that is most relevant to the problem. In a nutshell, SFAs remove or add one feature at a time based on the classifier performance until a feature subset of the desired size k is reached.

Install the below library for using SFS algorithms.
pip install mlxtend
Import SFS using the below
from mlxtend.feature_selection import SequentialFeatureSelector as SFS

Download the dataset from the given link AirlinePassenger and perform the following tasks. There is a separate file for train and testing. Download only the train.csv file.
1) Preprocess, clean and prepare the dataset based on the previous lab experience. Separate features and labels as X and Y respectively. [5 marks]
2) Create an object of SFS by embedding the Decision Tree classifier object, providing 10 features, forward as True, floating as False and scoring = accuracy. Train SFS and report accuracy for all 10 features. Also, list the names of the 10 best features selected by SFS. [10 marks]
3) Using the forward and Floating parameter toggle between SFS(forward True, floating False), SBS (forward False, floating False), SFFS (forward True, floating True), SBFS (forward False, floating True), and choose cross validation = 4 for each configuration. Also, report cv scores for each configuration. [5 marks]
4) Visualize the output from the feature selection in a pandas DataFrame format using the get_metric_dict for all four configurations. Finally, plot the results for each configuration (from mlxtend. plotting import plot_sequential_feature_selection as plot_sfs). [10 marks]
5) Implement Bi-directional Feature Set Generation Algorithm from scratch. It must take a Full Set of features as well as similarity measures as input. [10 marks]
6) Use the function implemented in part 5 and use selection criteria from the following: [10 marks]

- Accuracy Measures: using Decision Tree and SVM Classifiers
- Information Measures: Information gain
- Distance Measure: Angular Separation, Euclidian Distance and City-Block Distance
- Distance Measures. - Measures of separability, discrimination or divergence measures. The most typical is derived from the distance between the class conditional density functions.)

7) Train any classifier of your choice on the Selected features generated from each measure and report its classification results. [10 marks]

Question 02: [40 Marks]:

1. Make a Dataset of 1000 points sampled from a zero-centred gaussian distribution with a covariance matrix

$$\sum = \begin{bmatrix} 0.6006771 & 0.14889879 & 0.244939 \\ 0.14889879 & 0.58982531 & 0.24154981 \\ 0.244939 & 0.24154981 & 0.48778655 \end{bmatrix}$$

   Label the points as shown below:

$$class = \begin{cases} 0 & \vec{x}.\vec{v} > 0 \\ 1 & \vec{x}.\vec{v} <= 0 \end{cases} where \vec{v} = \begin{bmatrix} 1/sqrt(6) \\ 1/sqrt(6) \\ -2/sqrt(6) \end{bmatrix}$$

   and x is the data point. Visualize the data as a 3D scatter-plot using plotly's scatter_3d function.

   [10 marks]

2. Apply Principal Component analysis (using sklearn) with n_components=3 on the input data X and transform the data accordingly. [5 marks]
3. Perform Complete FS on the Transformed Data with a number of features in subset =2. Fit a Decision Tree for every subset-set of features of size 2 and plot their decision boundaries superimposed with the data. [20 marks]
4. Which of the above feature subsets represents the one that can be obtained by applying PCA(n_components =2)? Explain the difference in the accuracies between this subset and other subsets by running suitable experiments. [5 marks]