**Assignment Set - I**

**Problem - I**

Load the "california\_housing\_price" dataset from scikit-learn dataset and load it to the "raw-data" variable and perform PCA by following steps.

**Hints:** from sklearn.datasets import fetch\_california\_housing

raw\_data = fetch\_california\_housing()

* Show the keys (column names) of the DataFrame.
* Show the "target", "feature\_names", and "DESER" values.
* Store the 'data' key from the raw\_data dictionary as the data for your DataFrame and the 'feature\_names' key as the column names.
* Show the statistical description of your data frame.
* Also, show the top 5 rows of the data.
* Using a standard scaler, performed PCA with 2 Dimensions.
* Show the data matrix shape before and after employing PCA.
* Visualize the data (scatter plot) using the first two principal components.
* Determine explained variance and plot it against cumulative explained variance.
* Compute how many principal components are required to preserve 80% variance.

**Problem - II**

Perform the K-NN with the "Social\_Network\_Ads" dataset using the following steps.

The Dataset contains information about users on a Social Networking site and using that information as a feature for our ML model, the model predicts whether a particular user after clicking on an ad on the Social networking site goes on to buy a particular product or not. It is a CLASSIFICATION PROBLEM as the output says whether the user buys the product or not, so it’s either a yes or a no. Well, this particular Social Network has a Business client which let's assume is a car company that advertises itself by putting ads on the social networking site. Now the work of the social network here is to gather information as to whether the user bought the product or not.

* Load the data using pandas and split it 75 & 25 for the train and test set.
* Show the X\_train, X\_test , y\_train and y\_test.
* Scale it using "StandardScaler" and show the X\_train and X\_test.
* Perform the K-NN for five neighbours.
* Show the results on the test set in terms of confusion matrix and accuracy.
* Visualise the decision boundary on the Training set and Test set data separately.

**Hints:** To understand, how KNN works and how to implement KNN using Scikit-learn, please look at the documents shared in this folder.