**Question 1: Data Transformation**

The file SensorData question1.csv contains data obtained from a sensory system.

Library used: pandas

The input data is fetched from the input file using the pandas function read\_csv() and stored in a storage structure DataFrame named ‘df’.

***Q1a,*** two new columns named as Original Input3 and Original Input12 are generated which are the exact copies of the Input3 and Input12 columns respectively.

***Q1b,*** Z score value of the column Input3 is being calculated and is saved in the same column Input3. Z Score is the measure of deviation from the mean value of any data point. Z Score is calculated using the formulae: (df[col\_name]-df[col\_name].mean())/df[col\_name].std().

***Q1c,*** Input12 column is normalized to the range (0.0,1.0) and saved to the same column Input12. In case of huge differences between the values of dimensions, Normalization is required to standardize the input dimensions so as to get efficient output. It is calculated by: Normed\_df = (dataframe[column\_name] - dataframe.min()) / (dataframe.max() - dataframe.min())**.**

***Q1d,*** A new column named as ‘Average Input’ is calculated using the mean() function on all the columns except 2 columns ‘Original Input3’ and ‘Original Input12’ along all the rows ( axis = 1). At last, a new csv file is saved having all the data manipulations and transformations done so far using the pandas function to\_csv() with arguments: File path and indexing as false(to ignore the indexing in the file) named question1 out.csv.

**Question 2: Data Reduction and Discretisation**

The files DNAData question2 a.csv contains biological data arranged into mul-

tiple columns.

Library Used: sklearn.decomposition (PCA)

sklearn.preprocessing (Binning by KBinsDiscretiser)

The input data is fetched from the input file using the pandas function read\_csv() and stored in a storage structure DataFrame named ‘fileInput’.

***Q2a***, PCA is applied on the input data using the fit() method. One cannot visualize more than 3 dimensions , so if the data is of huge dimensions, its dimensionality is reduced using the PCA technique. PCA basically computes the co variance matrix, EigenVectors and Eigenvalues from it. Eigenvalue with highest value will comprise the n components. Then all the data points are projected on that EigenVector and transformed. A plot of Components Vs. Variance is displayed having pca.explained\_variance\_ratio\_.cumsum() which indicates the Cumulative sum of the variances. It returns a vector of variances for all the dimensions that can make 95% of the variance(22 dimensions variance ). Further the number of dimensions used for minimum 95% variance is returned using n\_components\_ property of pca object (pca.n\_components\_)

Thus the input dimensions gets reduced to 22 dimensions.

***Q2b,*** KBinsDiscretizer module is used to sort the input data into 10 bins of same width using KBinsDiscretizer(n\_bins, encode, strategy). encode is set to 'ordinal' for integer values of the bins, strategy is set to 'uniform' for equal width bins. Further Concatenating the new columns(pca\_ColumnNumber\_Width) with the original dataset.

***Q2c***, Similar way the data is binned into 10 buckets of same frequency(strategy = 'quantile'). Concatenating the new columns(pca\_ColumnNumber\_freq) with the original dataset is performed then.

***Q2d***, At last, a new csv file is saved having all the data manipulations and transformations done so far using the pandas function to\_csv() with arguments: File path and indexing as false(to ignore the indexing in the file) named question2 out.csv.