COMP40370 Practical 2: Data Preprocessing- Report

Rohit Jyotiba Chougule & 19200240

27/09/2019

Question 1: Data Transformation

The Sensor data from the csv was read using the read_Csv() function in pandas library.

df.insert((len(df.columns)), "Original Input3", df["Input3"].round(3), allow_duplicates=True)

1. The above code chunk was used to copy the column values and add a new column to the existing data frame.
2. The z-score transformation method can be evaluated using the formula for the z-score which requires the mean value of the column and the standard deviation and the same was used in calculating the zscore for Input3
3. The MinMax normalization was used to normalize the attribute Input12 in the dataset to normalize the data values in the range 0, 1
4. The average of the original input values was calculated using:

df["Average Input"] = df.iloc[:0:12].mean(axis=1)

5. The below code chunk was used to export the dataframe output values to the csv file.

df.to_csv('output/question1_out.csv', index=False, header=True)

Question 2

1. PCA or Principal Component Analysis is a dimensionality reduction technique. For a given set of dataset with M vectors in n dimensions, we find k<=n orthogonal vectors which are highly co-related to the M vectors in n dimensions to represnt the data. PCA works only for numeric data.

The below code chunk was implemented to perform PCA and ensure 95% of variance is explained.

pca = PCA(0.95)

test = pca.fit_transform(df2)

In this dataset, the dimension of the data was 88 * 59 we were able to reduce it to 88 * 22

2. Discretization of the attributes subset was done using the qcut function from the sklearn.decomposition library.

pca_test = pd.qcut(test[:, i], 10, labels=False)

The qcut() in pandasis Quantile based discretization, which discretizes the passed data into equal sized buckets depending on the rank.

In this example we have the data size to be 88 * 22.

Each of our 22 columns will pass through the qcut() and we will have the buckets according to the quantile membership of each row value.

The transformed data was in an n-d array, however the function qcut() accepts only 1-d array. Hence we have to move the transformed binned data into an 1-d array.

To add the data to the original data frame we have to concat the output from the qcut to the original data frame.

3. Discretization of the attribute subset with regards to frequency was done using the below function in the code:

pca_test2 = pd.cut(test[:, i], 10, labels=False)

The cut() in pandas discretizes the passed data into equal sized buckets depending on the parameter passed.

In this example we have the data size to be 88 * 22.

Each of our 22 columns will pass through the cut() and we will have the buckets according to the quantile membership of each row value.

Similar to qcut(), cut() also accepts only 1-d array, hence we have to follow the same process used in the above question.

The naming part of the question were done using the rename function in pandas using the below code:

_df2_original.rename(columns={i: 'pca' + str(i) + '*width'}, inplace=True)*

_df2_original.rename(columns={i: 'pca' + str(i) + '*freq'}, inplace=True)*

4. The csv files for the output were then exported using:

df2_original.to_csv('output/question2_out.csv', index=False, header=True)
