Data Transformation

Question 1. Generate a new attribute called Original Input3 which is a copy of the attribute Input3. Do the same with the attribute Input12 and copy it into Original Input12.

This question requires the function of .read_csv() in pandas, so I import the Pandas library, and read the path of SensorData_question1.csv in the folder.

I use .copy in DataFrame module to copy these two columns,respectively, and create two attributes named 'Original Input3' and 'Original Input12'.

Question 2. Normalise the attribute Input3 using the z-score transformation method.

I use the Z score method to normalise the Input3.

```
SensorData['Input3']=(SensorData['Input3']-SensorData['Input3'].mean())/SensorData['Input3'].std()
```

Question 3. Normalise the attribute Input12 in the range [0.0, 1.0].

I use the normal method to normalise the Input3.

```
SensorData['Input12']=((SensorData['Input12']-SensorData['Input12'].min())/(SensorData['Input12'].max()-SensorData['Input12'].min()).
```

Question 4. Generate a new attribute called Average Input, which is the average of all the attributes from Input1 to Input12. This average should include the normalised attributes values but not the copies that were made of these.

```
SensorData['Average Input']=np.mean(SensorData.iloc[:,:13],axis=1)
```

But there is a mistake that I was not able to correct which is this method make the negative number become positive when doing this ,I will research later.

Question 5.Save the newly generated dataset to ./output/question1 out.csv:

I used the to_csv method to do this.

```
SensorData.to_csv('./output/question1_out.csv')
```

Data Reduction and Discretisation

Question 1. Reduce the number of attributes using Principal Component Analysis (PCA), making sure at least 95% of all the variance is explained.

I Firstly scale the data to range(0,1):

```
scaler = MinMaxScaler()
data_rescaled = scaler.fit_transform(DNAData)
```

Then reduce the number of attributes using PCA:

```
pca = PCA(n_components = 0.95)
pca.fit(data_rescaled)
reduced = pca.transform(data_rescaled)
reduced=pd.DataFrame(reduced)
```

Question 2. Discretise the PCA-generated attribute subset into 10 bins, using bins of equal width. For each component X that you discretise, generate a new column in the original dataset named pcaX width. For example, the first discretised principal component will correspond to a new column called pca1 width.

```
	for i in range(reduced.shape[1]):
		DNAData['pca'+str(i)+'_width']=pd.cut(reduced[:,i],10)
```

Question 3. Discretise PCA-generated attribute subset into 10 bins, using bins of equal frequency (they should all contain the same number of points). For each component X that you discretise, generate a new column in the original dataset named pcaX freq. For example, the first discretised principal component will correspond to a new column called pca1 width.

```
	for i in range(reduced.shape[1]):
		DNAData['pca'+str(i)+'_freq']=pd.qcut(reduced[:,i],10)
```

Question 4. Save the generated dataset

```
	DNAData.to_csv('./output/question2_out.csv')
```