The program uses scipy library to calculate the distances and zscore normalization.

a) Here is the observation of the four most correlated and four least correlated.
   Order for max : 1>2>3>4 **(graph 1 denotes the pair which is maximum correlated)**
   Order for Min: 5>6>7>8**(graph 8 (last graph)denotes the pair which is least correlated)**

   The axes are labelled accordingly:

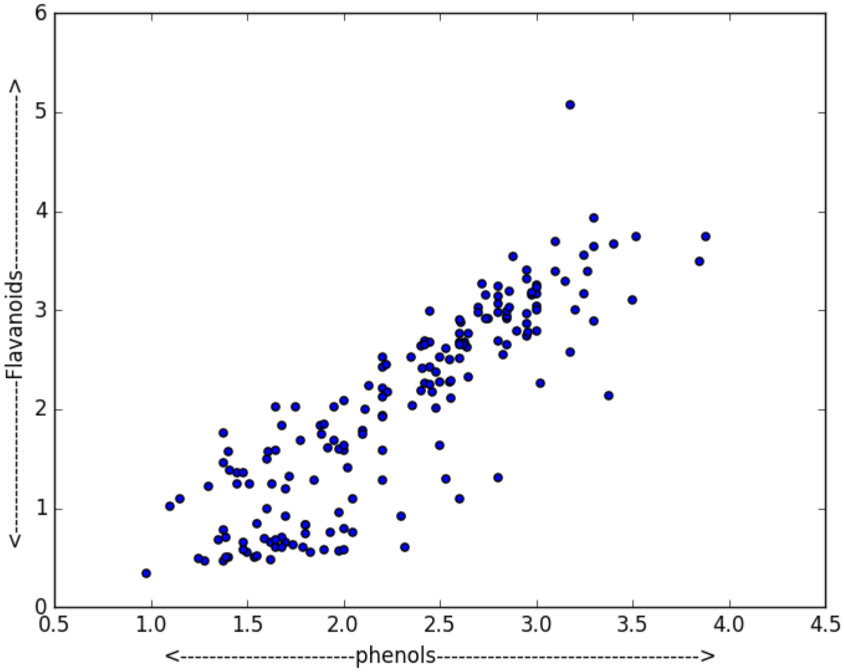
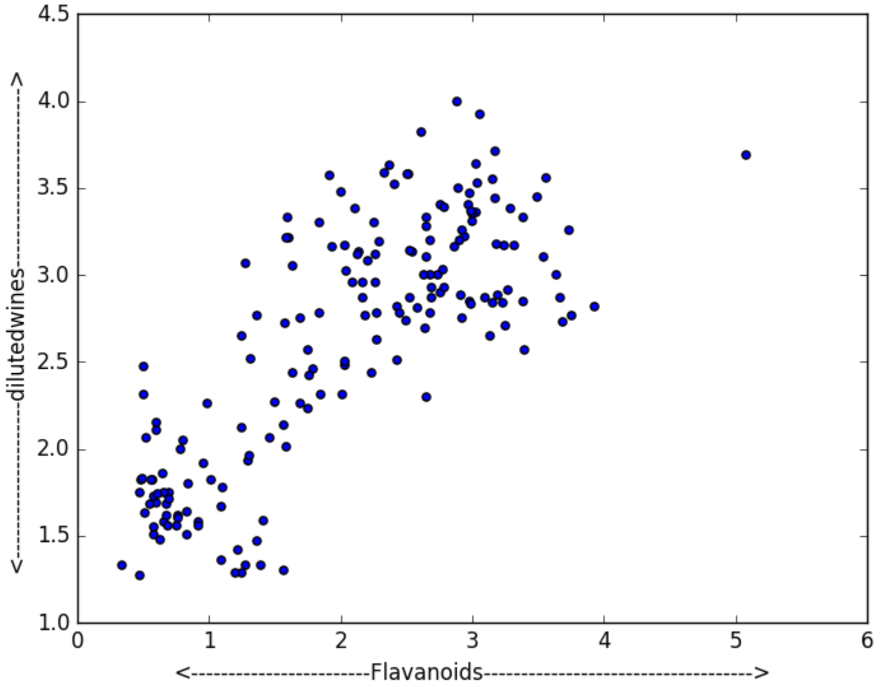
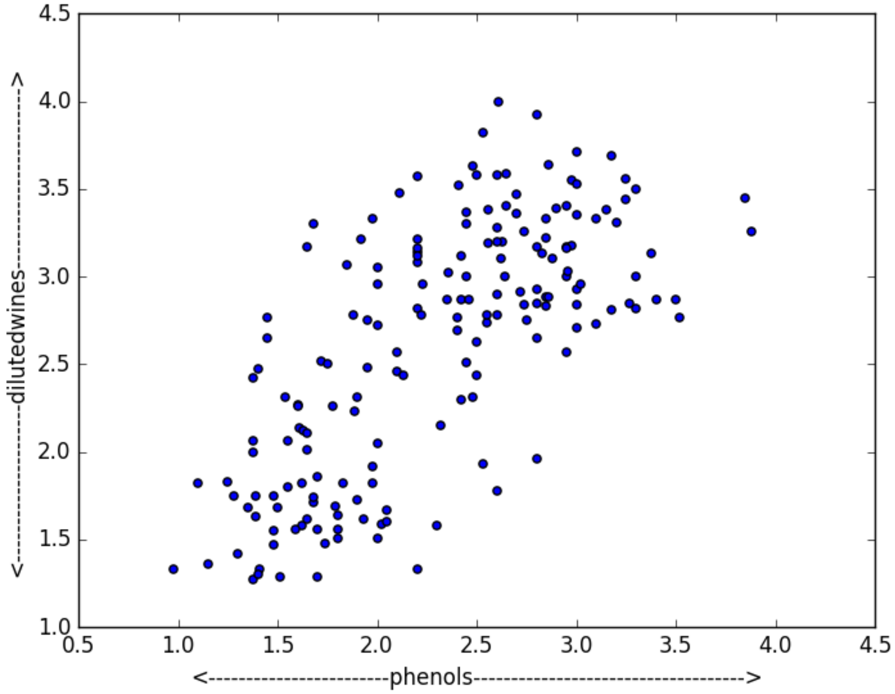
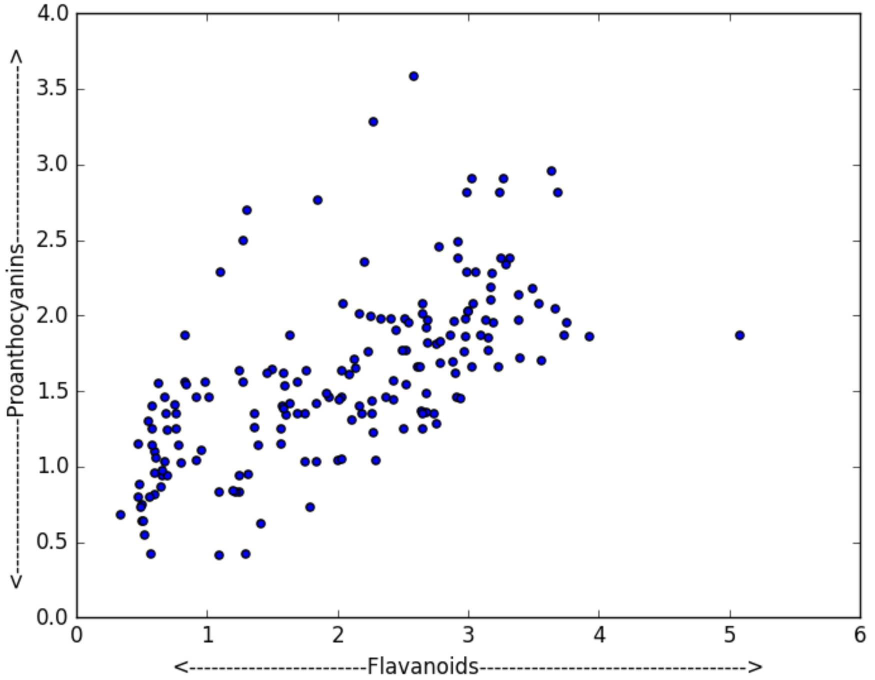
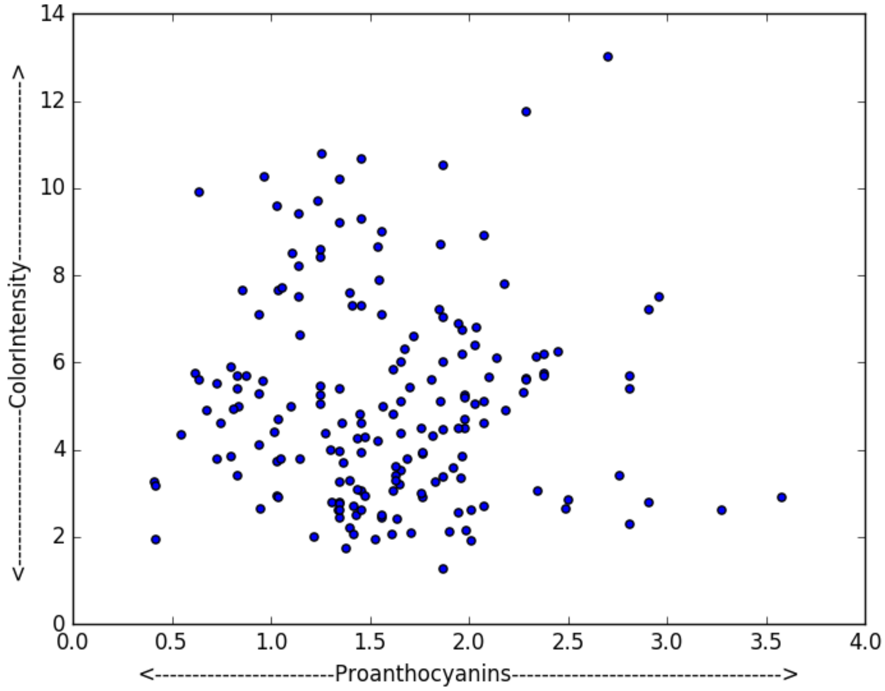


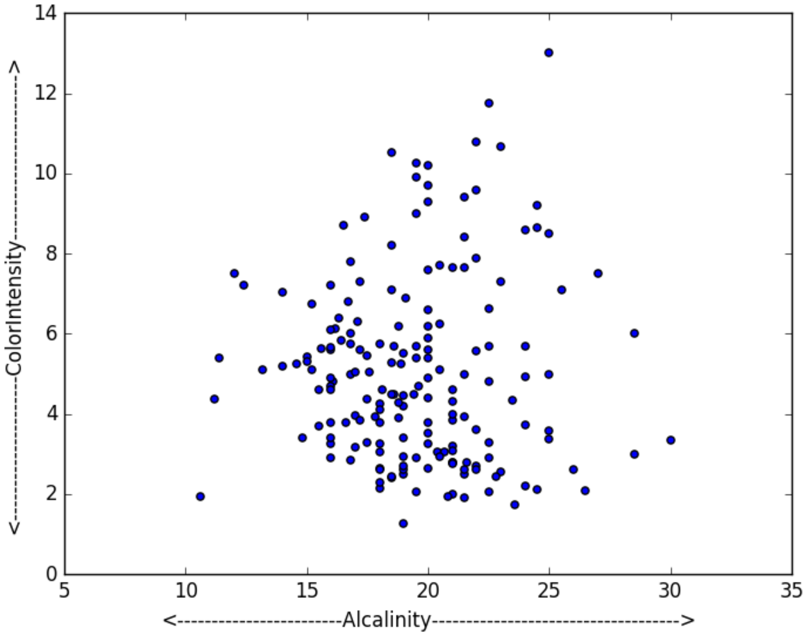
Figure 1 (Most correlated)

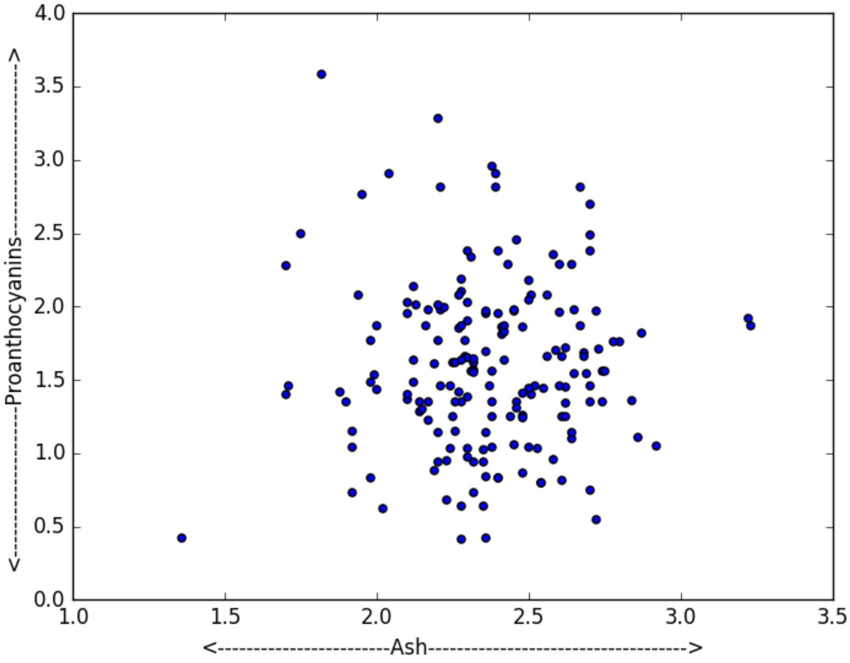

4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
<---------------dilutedwines--------------->
0
1
2
3
4
5
6
<--------------------Flavanoids------------------------------>


4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
<---------------dilutedwines--------------->
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
<--------------------phenols------------------------------>


4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
<------------------Proanthocyanins------------------>
0
1
2
3
4
5
6
<--------------------Flavanoids------------------------------->


14
12
10
8
6
4
2
0
<------------------ColorIntensity------------------>
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
<--------------------Proanthocyanins-------------------------------->


14
12
10
8
6
4
2
0
<-------------ColorIntensity------------->
5
10
15
20
25
30
35
<--------------------Alcalinity------------------------------>


4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
<-------------Proanthocyanins------------->
1.0
1.5
2.0
2.5
3.0
3.5
<-------------------Ash------------------------->

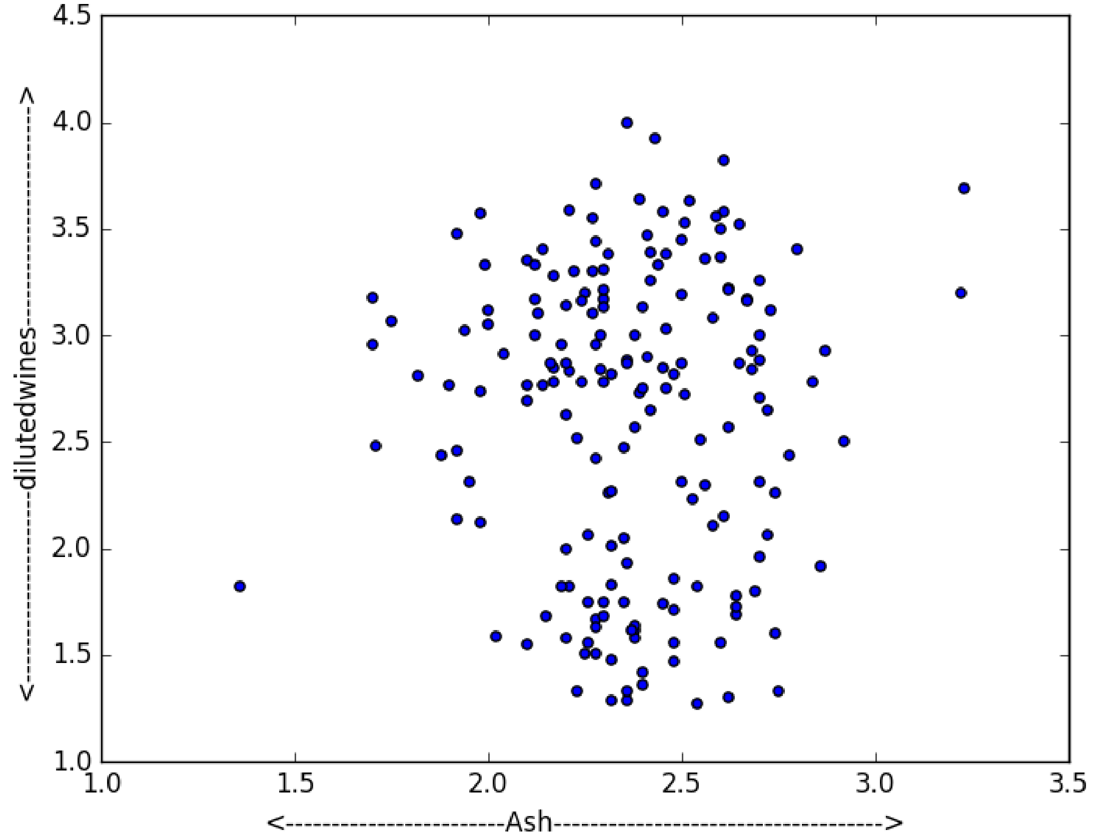
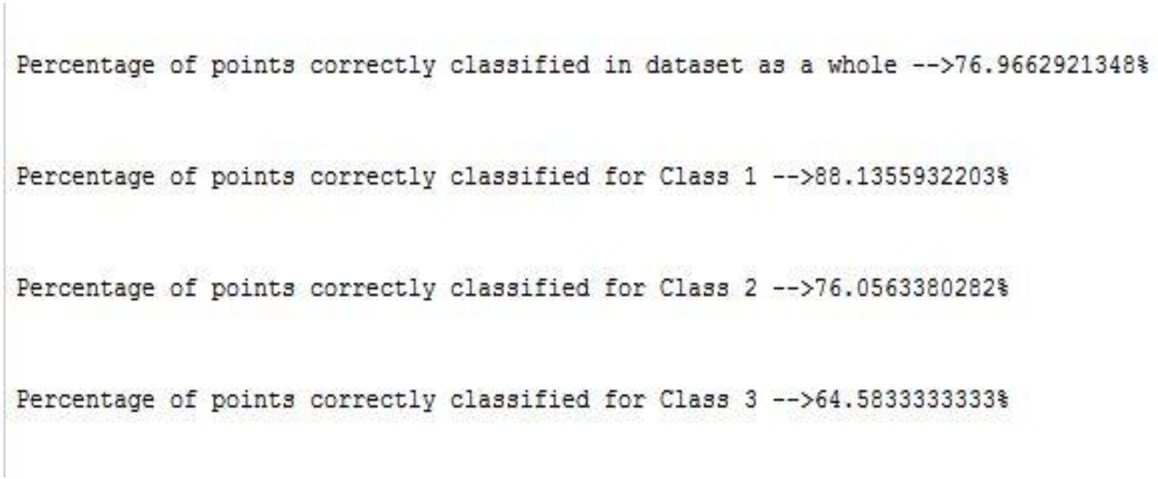


Figure 8(Least correlated)

Again, different criteria like color, dimensions, border were tried for plotting the scatter plots using matplotlib library of python and finally I zeroed on these ones which shows the scattering and spatial distances of the data point visibly and clearly.

b) After getting the most nearest neighbor applying Euclidean distance to each example in the dataset, here are the accuracy percentages observation for whole dataset and for each class.



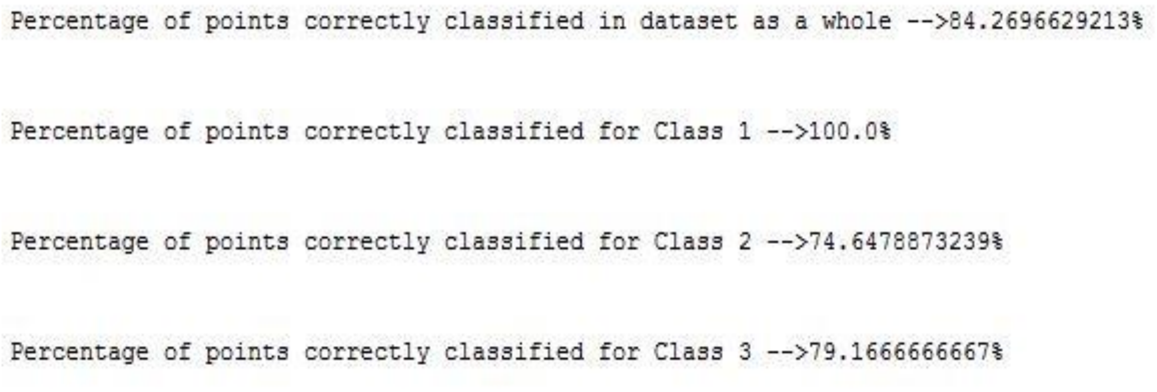
```
Percentage of points correctly classified in dataset as a whole -->76.9662921348%

Percentage of points correctly classified for Class 1 -->88.1355932203%

Percentage of points correctly classified for Class 2 -->76.0563380282%

Percentage of points correctly classified for Class 3 -->64.5833333333%
```

c) **Observation with 0-1 normalization:**



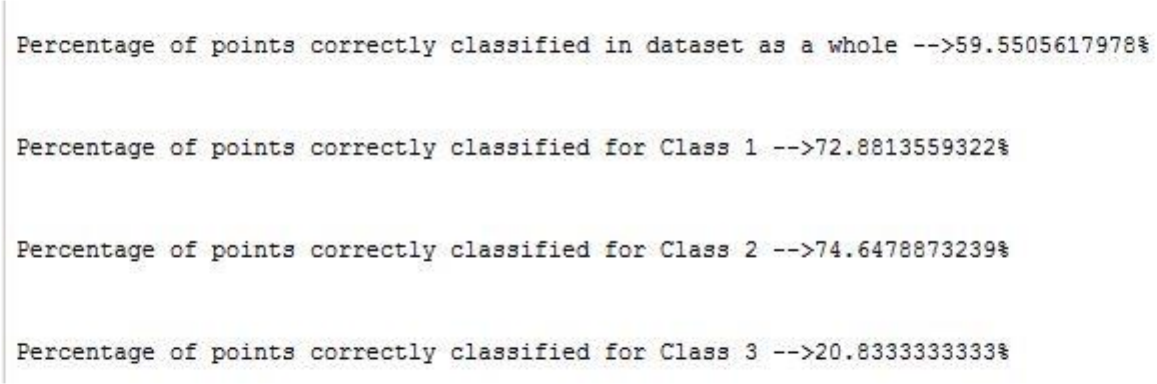
```
Percentage of points correctly classified in dataset as a whole -->84.2696629213%

Percentage of points correctly classified for Class 1 -->100.0%

Percentage of points correctly classified for Class 2 -->74.6478873239%

Percentage of points correctly classified for Class 3 -->79.1666666667%
```

Observation with z-score normalization:



```
Percentage of points correctly classified in dataset as a whole -->59.5505617978%

Percentage of points correctly classified for Class 1 -->72.8813559322%

Percentage of points correctly classified for Class 2 -->74.6478873239%

Percentage of points correctly classified for Class 3 -->20.8333333333%
```

Analysis:

The z-score linearly transforms the data in such a way, that the mean value of the transformed data equals 0 while their standard deviation equals 1. The transformed values themselves do not lie in a particular interval like [0,1] or so. But z-score might behave badly if the individual feature do not more or less look like standard normally distributed data like in the wine data set, the performance of z-score on class 3 is very less and when we carefully observe data for class 3, the data is not normally distributed.

But, In general z-score works best than the 0-1 normalization and other normalization techniques like min-max.