Table of Contents

1. ***Consider the wine datasets given at this LINK. One dataset at this site is for red wines and the other is for white wines. Data has eleven features and the twelfth column is the target attribute: "wine-quality". The values of wine-quality range between 0 and 10. The goal of this task is to learn a regression model that predicts "wine-quality" for a given set of features. Consider the white wine dataset and perform the following tasks with this dataset. Use Matlab or Python SKLearn libraries to perform these tasks.***
    a. ***(8) Do linear regression to learn the single-feature regression models, one model for each of the 11 features. Find the R^2 and AIC values for each of these models. Report these values for the models.***

Solution of 1(a):

Python Commands Used:

```
In [2]: from sklearn.metrics import roc_curve, auc
        # imports
        import pandas as pd
        import matplotlib as mpl
        import matplotlib.pyplot as plt
        import random
        from statsmodels.regression.linear_model import OLS
        from statsmodels.tools import add_constant
```

```
In [3]: # Load CSV using Pandas
        filename = 'winequality-white.csv'

        data1 = pd.read_csv(filename,';')
        print(data1.shape)
```

```
(4898, 12)
```

```
In [4]: data1.columns=['FA',
         'VA',
         'CA',
         'RSR',
         'CS',
         'FSD',
         'TSD',
         'DY',
         'PH',
         'SS',
         'AL',
         'quality']
```

Linear regression to learn the single-feature regression models:

```
import statsmodels.formula.api as smf

lmdataFA=smf.ols(formula='quality ~ FA', data=data1).fit()
lmdataVA=smf.ols(formula='quality ~ VA', data=data1).fit()
lmdataCA=smf.ols(formula='quality ~ CA', data=data1).fit()
lmdataRSR=smf.ols(formula='quality ~ RSR', data=data1).fit()
lmdataCS=smf.ols(formula='quality ~ CS', data=data1).fit()
lmdataFSD=smf.ols(formula='quality ~ FSD', data=data1).fit()
lmdataTSD=smf.ols(formula='quality ~ TSD', data=data1).fit()
lmdataDY=smf.ols(formula='quality ~ DY', data=data1).fit()
lmdataPH=smf.ols(formula='quality ~ PH', data=data1).fit()
lmdataSS=smf.ols(formula='quality ~ SS', data=data1).fit()
lmdataAL=smf.ols(formula='quality ~ AL', data=data1).fit()
```

R square values for each of the 11 models:

```
print(lmdataFA.rsquared)
print(lmdataVA.rsquared)
print(lmdataCA.rsquared)
print(lmdataRSR.rsquared)
print(lmdataCS.rsquared)
print(lmdataFSD.rsquared)
print(lmdataTSD.rsquared)
print(lmdataDY.rsquared)
print(lmdataPH.rsquared)
print(lmdataSS.rsquared)
print(lmdataAL.rsquared)
```

```
0.0129192390857
0.0379170346255
8.48073549092e-05
0.00952123753714
0.0440724568996
6.65540591903e-05
0.0305330952136
0.0943247292252
0.00988577719478
0.00288131449348
0.189725332749
```

AIC values for each of the 11 models:

```
print(lmdataFA.aic)
print(lmdataVA.aic)
print(lmdataCA.aic)
print(lmdataRSR.aic)
print(lmdataCS.aic)
print(lmdataFSD.aic)
print(lmdataTSD.aic)
print(lmdataDY.aic)
print(lmdataPH.aic)
print(lmdataSS.aic)
print(lmdataAL.aic)
```

```
12649.5426747
12523.9032382
12712.8180133
12666.3749649
12492.465077
12712.9074247
12561.351623
12227.9667222
12664.5719542
12699.1003683
11682.7824135
```

R square and AIC values for each model :

Model	R square value	AIC value
Fixed acidity	0.012919239	12649.54
Volatile Acidity	0.037917035	12523.9
Citric Acid	8.48074E-05	12712.82
Residual Sugar	0.009521238	12666.37
Chlorides	0.044072457	12492.47
Free Sulphur Dioxide	6.65541E-05	12712.91

Total Sulphur Dioxide	0.030533095	12561.35
Density	0.094324729	12227.97
pH	0.009885777	12664.57
Sulphates	0.002881314	12699.1
Alcohol	0.189725333	11682.78

Comments :

From the R square and AIC values for each of the models, we can clearly conclude that the single feature regression model with Alcohol Feature is the best regression model since this model has the highest R squared value and least AIC value. Hence, we choose alcohol as one of the features to form the bivariate regression model in 1(b).

b. (8) Select the model with the highest R^2 value, combine with its feature other features, one at a time, and thus generate all bivariate regression models (models containing two features). One of these two features is from the selected single-feature model and the other is from one of the remaining 10 features. Report the R^2 and AIC values for all the bivariate regression models.

Solution of 1(b):

Comments :

To generate the bivariate regression models, we select one of the features as alcohol(as this single variate model is having the highest R squared value) and combine it with remaining 10 features .

Python Commands Used for generating 10 Bivariate Regression models combining Alcohol feature and each of the remaining 10 features:

```
import statsmodels.formula.api as smf

bmdataFA=smf.ols(formula='quality ~ AL+FA', data=data1).fit()
bmdataVA=smf.ols(formula='quality ~ AL+VA', data=data1).fit()
bmdataCA=smf.ols(formula='quality ~ AL+CA', data=data1).fit()
bmdataRSR=smf.ols(formula='quality ~AL+ RSR', data=data1).fit()
bmdataCS=smf.ols(formula='quality ~ AL+CS', data=data1).fit()
bmdataFSD=smf.ols(formula='quality ~ AL+FSD', data=data1).fit()
bmdataTSD=smf.ols(formula='quality ~ AL+TSD', data=data1).fit()
bmdataDY=smf.ols(formula='quality ~ AL+DY', data=data1).fit()
bmdataPH=smf.ols(formula='quality ~ AL+PH', data=data1).fit()
bmdataSS=smf.ols(formula='quality ~ AL+SS', data=data1).fit()
```

R square values for each of the 10 bi variate models :

```
print(bmdataFA.rsquared)
print(bmdataVA.rsquared)
print(bmdataCA.rsquared)
print(bmdataRSR.rsquared)
print(bmdataCS.rsquared)
print(bmdataFSD.rsquared)
print(bmdataTSD.rsquared)
print(bmdataDY.rsquared)
print(bmdataPH.rsquared)
print(bmdataSS.rsquared)
```

```
0.193502758043
0.240231184753
0.190293912045
0.201951000816
0.192958610426
0.204351970052
0.190266569777
0.192454876622
0.191923201097
0.193480628029
```

AIC values for each of the 10 bi variate models :

```
print(bmdataFA.aic)
print(bmdataVA.aic)
print(bmdataCA.aic)
print(bmdataRSR.aic)
print(bmdataCS.aic)
print(bmdataFSD.aic)
print(bmdataTSD.aic)
print(bmdataDY.aic)
print(bmdataPH.aic)
print(bmdataSS.aic)
```

```
11661.8950011
11369.5515955
11681.3442227
11610.3167113
11665.1985912
11595.5586287
11681.5096162
11668.254839
11671.4785472
11662.0293987
```

R square and AIC values for 10 bivariate models :

Model		Values	
Feature1	Feature2	R square value	AIC value
Alcohol	Fixed acidity	0.193502758	11661.895
Alcohol	Volatile Acidity	0.240231185	11369.5516
Alcohol	Citric Acid	0.190293912	11681.34422
Alcohol	Residual Sugar	0.201951001	11610.31671
Alcohol	Chlorides	0.19295861	11665.19859
Alcohol	Free Sulphur Dioxide	0.20435197	11595.55863
Alcohol	Total Sulphur Dioxide	0.19026657	11681.50962
Alcohol	Density	0.192454877	11668.25484
Alcohol	pH	0.191923201	11671.47855
Alcohol	Sulphates	0.193480628	11662.0294

Comments :

From the R square and AIC values of all the bivariate models, the model with features Alcohol and Volatile Acidity is the best bivariate model since this model has the highest R square value and least AIC value.

c. ***(8) Select the bivariate model with the highest R^2 value as the Best model at this stage. Combine a third feature from the remaining nine features with this selected bivariate model to build (and then select the best) 3-feature regression models. Report the R^2 and AIC values of all these models.***

Solution of 1(c):

Comments :

We have got the bivariate model with features Alcohol and Volatile Acidity as the best bivariate regression model. We will combine this model with the remaining 9 features to get the best 3 feature regression model.

Python Commands Used for generating nine 3-feature Regression models combining Alcohol , volatile Acidity features and each of the remaining 9 features:

```
import statsmodels.formula.api as smf

tmdataFA=smf.ols(formula='quality ~ AL+VA+FA', data=data1).fit()
tmdataCA=smf.ols(formula='quality ~ AL+VA+CA', data=data1).fit()
tmdataRSR=smf.ols(formula='quality ~AL+VA+ RSR', data=data1).fit()
tmdataCS=smf.ols(formula='quality ~ AL+VA+CS', data=data1).fit()
tmdataFSD=smf.ols(formula='quality ~ AL+VA+FSD', data=data1).fit()
tmdataTSD=smf.ols(formula='quality ~ AL+VA+TSD', data=data1).fit()
tmdataDY=smf.ols(formula='quality ~ AL+VA+DY', data=data1).fit()
tmdataPH=smf.ols(formula='quality ~ AL+VA+PH', data=data1).fit()
tmdataSS=smf.ols(formula='quality ~ AL+VA+SS', data=data1).fit()
```

R square values for each of the nine 3-feature regression models :

```
print(tmdataFA.rsquared)
print(tmdataCA.rsquared)
print(tmdataRSR.rsquared)
print(tmdataCS.rsquared)
print(tmdataFSD.rsquared)
print(tmdataTSD.rsquared)
print(tmdataDY.rsquared)
print(tmdataPH.rsquared)
print(tmdataSS.rsquared)
```

```
0.244425120132
0.240309616241
0.258526158066
0.241395675749
0.250771237105
0.243138060511
0.246906356199
0.24166031226
0.243096588268
```

AIC values for each of the nine 3-feature regression models :

```
print(tmdataFA.aic)
print(tmdataCA.aic)
print(tmdataRSR.aic)
print(tmdataCS.aic)
print(tmdataFSD.aic)
print(tmdataTSD.aic)
print(tmdataDY.aic)
print(tmdataPH.aic)
print(tmdataSS.aic)
```

```
11344.4396631
11371.0459453
11252.1662114
11364.038715
11303.1273426
11352.7759049
11328.3286287
11362.3297664
11353.0442834
```

R square and AIC values for 9 3-variate models :

Model			Values	
Feature1	Feature2	Feature3	R square value	AIC value
Alcohol	Volatile Acidity	Fixed acidity	0.24442512	11344.43966
Alcohol	Volatile Acidity	Citric Acid	0.240309616	11371.04595
Alcohol	Volatile Acidity	Residual Sugar	0.258526158	11252.16621
Alcohol	Volatile Acidity	Chlorides	0.241395676	11364.03872

Alcohol	Volatile Acidity	Free Sulphur Dioxide	0.250771237	11303.12734
Alcohol	Volatile Acidity	Total Sulphur Dioxide	0.243138061	11352.7759
Alcohol	Volatile Acidity	Density	0.246906356	11328.32863
Alcohol	Volatile Acidity	pH	0.241660312	11362.32977
Alcohol	Volatile Acidity	Sulphates	0.243096588	11353.04428

Comments :

When combining the best Bivariate model (Alcohol+ Volatile Acidity) with the remaining 9 features we observe the R square and AIC values as stated above. When comparing the above R square values, we have got the 3- variate model {Alcohol+ Volatile Acidity+ Residual Sugar} with least AIC value and highest R square value. Therefore, this model is the best regression model with 3 features.

d. ***(14)Repeat the steps above to generate (k+1)-feature models from the k-feature models until the following situation arises: all the (k+1)-feature models have an AIC value higher than the AIC value of the k-feature model from which they are being generated. Stop the process and report the k-feature model found as being the best regression model for this data. Report the features included, their coefficients, and p-values for the coefficients. Comment on the magnitudes of the p-values.***

Solution of 1(d):

Explanation :

As illustrated in 1(a),1(b),1(c) we will repeat the above procedure to find the best 4 varaiate,5-variate----- till we get the (k+1)th model AIC higher than the AIC of the kth model, which means that kth model is better regression model than the (k+1)th model, we report the kth model as the best regression model.

R square and AIC values for 4-variate models(8) :

Model				Values	
Feature1	Feature2	Feature3	Feature 4	R square value	AIC value
Alcohol	Volatile Acidity	Residual Sugar	Fixed acidity	0.263473195	11221.37776
Alcohol	Volatile Acidity	Residual Sugar	Citric Acid	0.258939562	11251.4346
Alcohol	Volatile Acidity	Residual Sugar	Chlorides	0.258952792	11251.34716
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	0.263994221	11217.91164
Alcohol	Volatile Acidity	Residual Sugar	Total Sulphur Dioxide	0.259021191	11250.89505
Alcohol	Volatile Acidity	Residual Sugar	Density	0.263943223	11218.25101
Alcohol	Volatile Acidity	Residual Sugar	pH	0.262047902	11230.84699
Alcohol	Volatile Acidity	Residual Sugar	Sulphates	0.261928317	11231.64064
				0.263994221	11217.91164

From the 4 variate models, we can conclude that the model Alcohol, Volatile Acidity, Residual Sugar, Free Sulphur Dioxide is the best model. The AIC Values of the 4 variate models have reduced compared to the 3 variate models. Hence let's continue this process by Free Sulphur Dioxide as the 4th Feature.

R square and AIC values for 5-variate models(7) :

Model					Values	
Feature1	Feature2	Feature3	Feature4	Feature5	R square value	AIC value
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Fixed acidity	0.268028686	11192.98907
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Citric Acid	0.264569766	11216.07998
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Chlorides	0.264571597	11216.06779
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Total Sulphur Dioxide	0.264621569	11215.73496
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Density	0.268951565	11186.80971
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	pH	0.266980788	11199.99608
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Sulphates	0.266880907	11200.66344
					0.268951565	11186.80971

From the 5 variate models, we can conclude that the model Alcohol, Volatile Acidity, Residual Sugar, Free Sulphur Dioxide, Density is the best model. The AIC Values of the 5 variate models have reduced compared to the 4 variate models. Hence let's continue this process by taking Density as the 5th Feature.

R square and AIC values for 6-variate models(6) :

Model						Values	
Feature1	Feature2	Feature3	Feature4	Feature5	Feature6	R square value	AIC value
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Density	Fixed Acidity	0.270035604	11181.54129
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Density	Citric Acid	0.269063776	11188.05784
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Density	Chlorides	0.269319631	11186.34306
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Density	Total Sulphur Dioxide	0.268973505	11188.66271
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Density	pH	0.275182115	11146.88633
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Density	Sulphates	0.274737391	11149.89066
						0.275182115	11146.88633

From the 6 variate models, we can conclude that the model Alcohol, Volatile Acidity, Residual Sugar, Free Sulphur Dioxide, Density, pH is the best model. The AIC Values of the 6 variate models have reduced compared to the 5 variate models. Hence let's continue this process by taking pH as the 6th Feature.

R square and AIC values for 7-variate models(5) :

Model							Values	
Feature1	Feature2	Feature3	Feature4	Feature5	Feature6	Feature7	R square value	AIC value
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Density	pH	Fixed Acidity	0.275945413	11143.72558
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Density	pH	Citric Acid	0.275240233	11148.49358
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Density	pH	Chlorides	0.275327348	11147.90481
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Density	pH	Total Sulphur Dioxide	0.275207933	11148.71186
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Density	pH	Sulphates	0.280119583	11115.40694
							0.280119583	11115.40694

From the 7 variate models, we can conclude that the model Alcohol, Volatile Acidity, Residual Sugar, Free Sulphur Dioxide, Density, pH, Sulphates is the best model. The AIC Values of the 7 variate models have reduced compared to the 6 variate models. Hence let's continue this process by taking Sulphates as the 7th Feature.

R square and AIC values for 8-variate models(4) :

Model								Values	
Feature1	Feature2	Feature3	Feature4	Feature5	Feature6	Feature7	Feature8	R square value	AIC value
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Density	pH	Sulphates	Fixed Acidity	0.281751964	11106.28775
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Density	pH	Sulphates	Citric Acid	0.280156435	11117.15619
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Density	pH	Sulphates	Chlorides	0.280257853	11116.46607
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Density	pH	Sulphates	Total Sulphur Dioxide	0.280232262	11116.64022
								0.281751964	11106.28775

From the 8 variate models, we can conclude that the model Alcohol, Volatile Acidity, Residual Sugar, Free Sulphur Dioxide, Density, pH, Sulphates, Fixed Acidity is the best model. The AIC Values of the 8 variate models have reduced compared to the 7 variate models. Hence let's continue this process by taking Sulphates as the 8th Feature.

R square and AIC values for 9-variate models(3) :

Model									Values	
Feature1	Feature2	Feature3	Feature4	Feature5	Feature6	Feature7	Feature8	Feature9	R square value	AIC value
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Density	pH	Sulphates	Fixed Acidity	Citric Acid	0.281755343	11108.2647
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Density	pH	Sulphates	Fixed Acidity	Chlorides	0.281780515	11108.093
Alcohol	Volatile Acidity	Residual Sugar	Free Sulphur Dioxide	Density	pH	Sulphates	Fixed Acidity	Total Sulphur Dioxide	0.281835337	11107.7192
									0.281835337	11107.7192

From the AIC values of the 9 variate models, the AIC values of the 9 variate models have increased compared to the AIC value of the 8th optimal variate models. Hence 8 variate model is the best fit model for this dataset.

The value of K which is the best regression model for this dataset is the 8-feature model

Features included, their Coefficients and the p-values:

Feature ID	Features-included	Their Coefficients	p-values
1	Alcohol	0.193163	1.306643e-15
2	Volatile Acidity	-1.888140	1.020239e-64
3	Residual Sugar	0.082847	1.391738e-29
4	Free Sulphur Dioxide	0.003349	7.673309e-07
5	Density	-154.291277	5.275581e-17
6	pH	0.694213	2.066280e-11
7	Sulphates	0.628508	3.522028e-10
8	Fixed Acidity	0.068104	8.643880e-04
*	Intercept	154.106249	2.206827e-17

Comments for the magnitudes for the p-values:

The p values of the initial features selected are very lower than the p values of the latter features. This could be because of the p-values for each term tests the null hypothesis that the coefficient is equal to 0. Since the magnitudes of the p-values have declined it clearly signifies that the latter features are lesser fit(or there is greater chance of their coefficients being 0) to the regression model than the former features selected.

e. (7) Find the five wines that have the largest magnitudes of difference between the predicted and the actual wine-quality values. Look at the regression model, the rest of the data, and comment on why you think these wines are outliers.

Solution of 1(e):

Python Commands for finding the five wines with largest magnitudes of difference between predicted and actual wines:

In [56]:
```
datatestquality = temdataCA.predict(data1)
fivewines=abs(datatestquality-data1.quality)
fivewines.sort_values( ascending=False)
```

Out[56]:
```
253     3.483381
445     3.379958
3810    3.249026
740     3.230414
4745    3.184861
```

The five wines that have the largest magnitudes of difference between predicted and actual wines are

4745 -difference=3.82

3307 -difference=3.43

253 - difference=3.38

445 -difference=3.35

3810 -difference=3.23

```
data1.iloc[4745,:]
```

```
FA            6.10000
VA            0.26000
CA            0.25000
RSR           2.90000
CS            0.04700
FSD         289.00000
TSD         440.00000
DY            0.99314
PH            3.44000
SS            0.64000
AL           10.50000
quality       3.00000
Name: 4745, dtype: float64
```

```
data1.iloc[3307,:]
```

```
FA            9.40000
VA            0.24000
CA            0.29000
RSR           8.50000
CS            0.03700
FSD         124.00000
TSD         208.00000
DY            0.99395
PH            2.90000
SS            0.38000
AL           11.00000
quality       3.00000
Name: 3307, dtype: float64
```

```
data1.iloc[253,:]
```

```
FA            5.8000
VA            0.2400
CA            0.4400
RSR           3.5000
CS            0.0290
FSD           5.0000
TSD         109.0000
DY            0.9913
PH            3.5300
SS            0.4300
AL           11.7000
quality       3.0000
Name: 253, dtype: float64
```

```
data1.iloc[445,:]
```

```
FA            7.1000
VA            0.3200
CA            0.3200
RSR          11.0000
CS            0.0380
FSD          16.0000
TSD          66.0000
DY            0.9937
PH            3.2400
SS            0.4000
AL           11.5000
quality       3.0000
Name: 445, dtype: float64
```

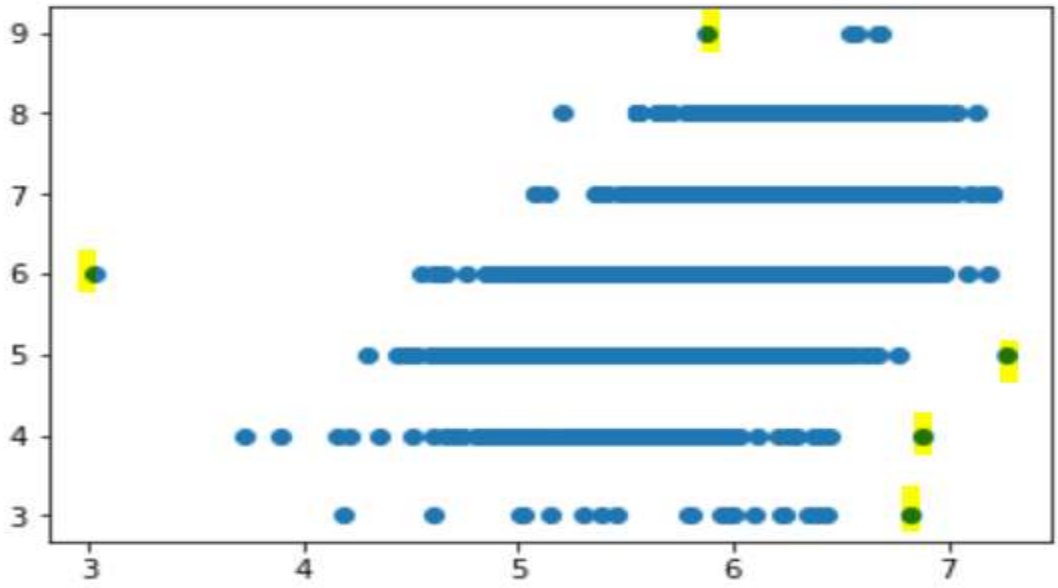
```
data1.iloc[3810,:]
```

```
FA            6.80000
VA            0.26000
CA            0.34000
RSR          15.10000
CS            0.06000
FSD          42.00000
TSD         162.00000
DY            0.99705
PH            3.24000
SS            0.52000
AL           10.50000
quality       3.00000
Name: 3810, dtype: float64
```

Comments on the five wines as outliers:

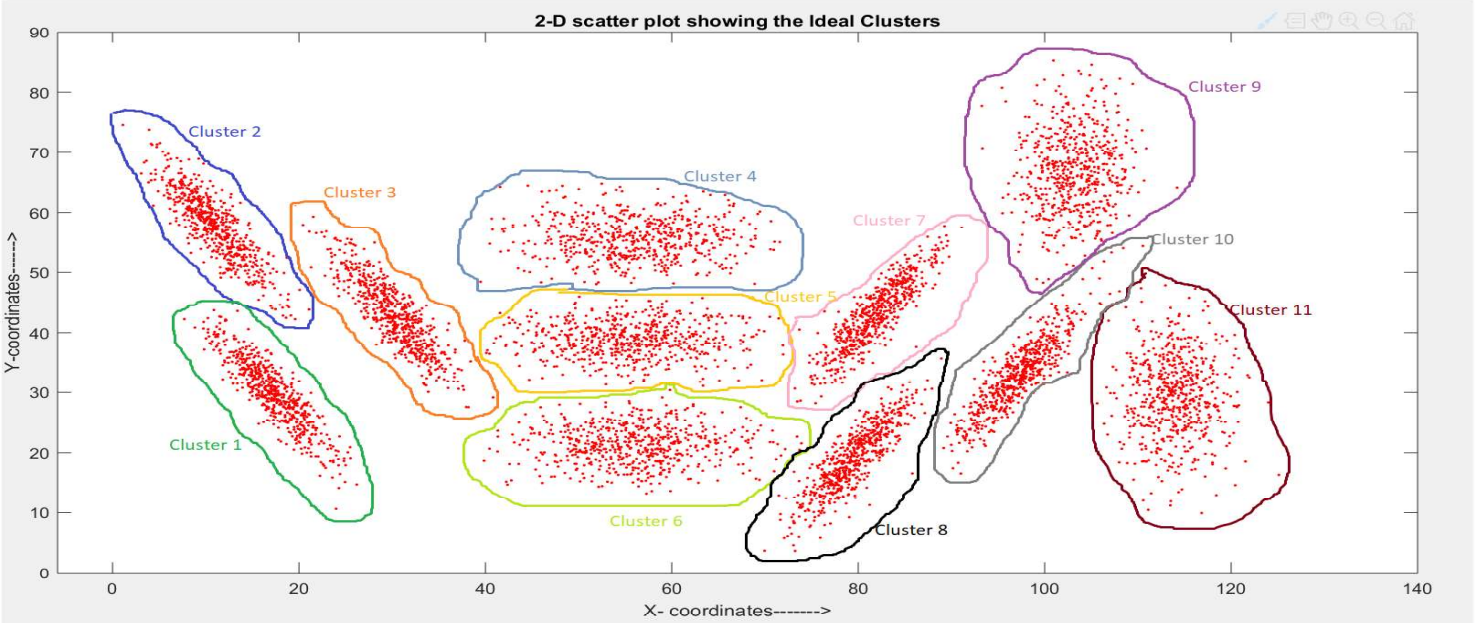
From the five wines, we can clearly see that all the outliers have quality 3 in which the difference between the actual and predicted values are highest. When we predict the value of quality using the regression model for index 4745, we get the quality of 6.82,which is the value that most of the data followed by regression. Hence these values are outliers, since these

Values are not following the model or the rest of the data. From the below graph between predicted and actual values, we can observe that these values are outliers in the data.



2. ***Consider the data-file attached with this homework containing 6600 data points on a 2-D plane. You will need to use the BIC metric to determine the quality of a clustering. This is computed here as: BIC = n*log(SSE/n) + log(n)*c*(d+1) where n is the number of data points, c is the number of clusters, and d is the number of features (dimensionality of the data). Remember to use the sum of SSEs for all the clusters in any clustering.***
    a. ***(3) Plot the data on a 2-D scatter plot and mark by hand the boundaries of the ideal clusters that you would like discovered in this dataset.***

Solution of 2(a):



The Ideal Cluster plot for the given datapoints is as shown in the above figure. As per ideal cluster intuition ,there are 11 clusters possible as shown in the figure indicating clusters from Cluster 1 to Cluster 11.

b. (12) Run the k-means algorithm for k = 3, 5, 7, 9, 11, 13, 15, 17 and 19. Plot the total SSE and BIC values for the above values of k. What is the best number of clusters for this dataset? How did you find the best number of clusters, briefly explain?

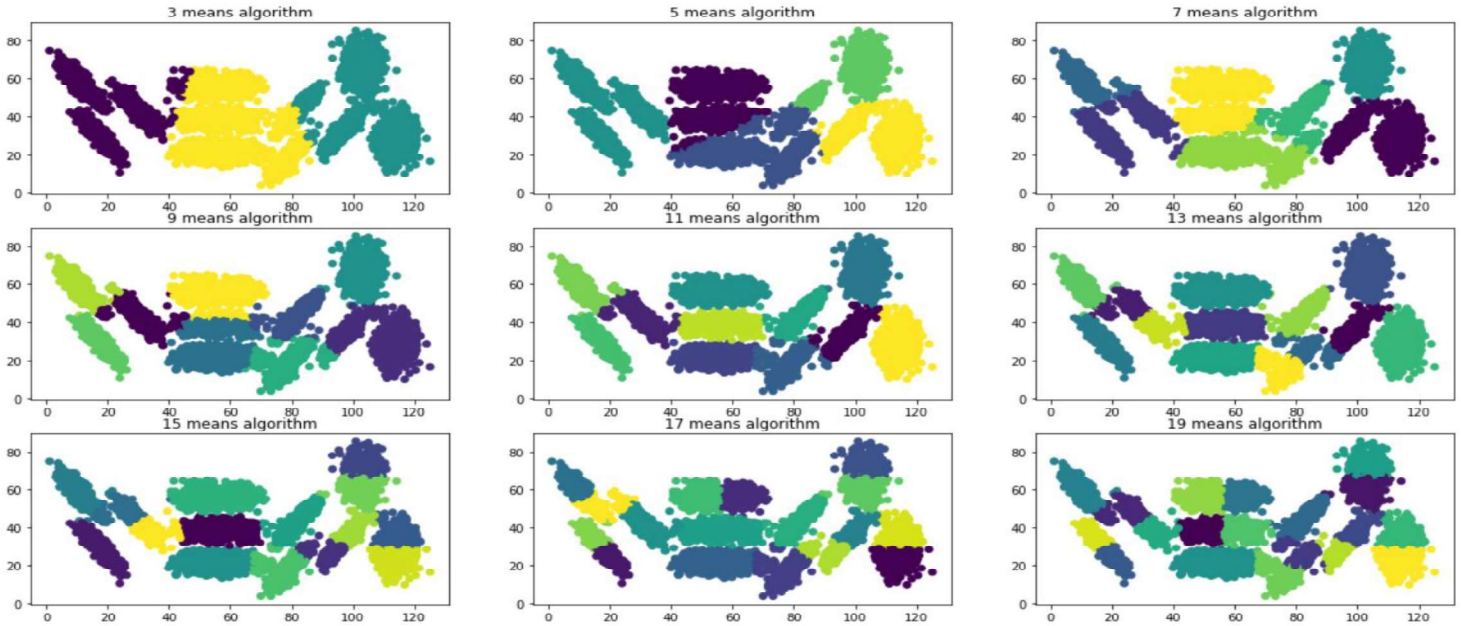
Solution of 2(b):

Python Commands for implementing k means algorithm on the dataset with different values of k:

```
from sklearn.cluster import KMeans
import numpy as np
y_pred_3 = KMeans(n_clusters=3, random_state=0).fit_predict(data2)
y_pred_5 = KMeans(n_clusters=5, random_state=0).fit_predict(data2)
y_pred_7 = KMeans(n_clusters=7, random_state=0).fit_predict(data2)
y_pred_9 = KMeans(n_clusters=9, random_state=0).fit_predict(data2)
y_pred_11 = KMeans(n_clusters=11, random_state=0).fit_predict(data2)
y_pred_13 = KMeans(n_clusters=13, random_state=0).fit_predict(data2)
y_pred_15 = KMeans(n_clusters=15, random_state=0).fit_predict(data2)
y_pred_17 = KMeans(n_clusters=17, random_state=0).fit_predict(data2)
y_pred_19 = KMeans(n_clusters=19, random_state=0).fit_predict(data2)
```

```
plt.figure(figsize=(20,10))
plt.subplot(331)
plt.scatter(data2.X,data2.Y,c=y_pred_3)
plt.title("3 means algorithm")

plt.subplot(332)
plt.scatter(data2.X,data2.Y,c=y_pred_5)
plt.title("5 means algorithm")

plt.subplot(333)
plt.scatter(data2.X,data2.Y,c=y_pred_7)
plt.title("7 means algorithm")

plt.subplot(334)
plt.scatter(data2.X,data2.Y,c=y_pred_9)
plt.title("9 means algorithm")

plt.subplot(335)
plt.scatter(data2.X,data2.Y,c=y_pred_11)
plt.title("11 means algorithm")
plt.subplot(336)
plt.scatter(data2.X,data2.Y,c=y_pred_13)
plt.title("13 means algorithm")

plt.subplot(337)
plt.scatter(data2.X,data2.Y,c=y_pred_15)
plt.title("15 means algorithm")
plt.subplot(338)
plt.scatter(data2.X,data2.Y,c=y_pred_17)
plt.title("17 means algorithm")
plt.subplot(339)
plt.scatter(data2.X,data2.Y,c=y_pred_19)
plt.title("19 means algorithm")
```

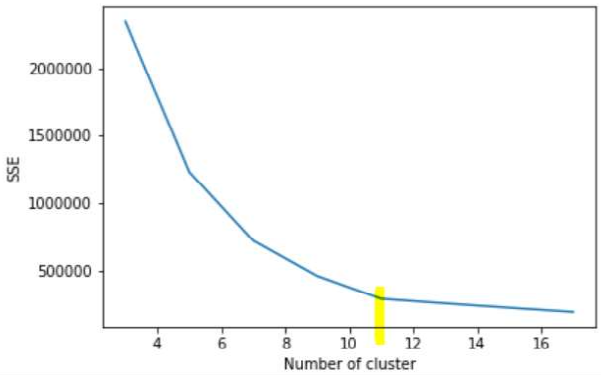
Plot for k means with different k values:



Python Commands for plotting SSE and BIC values for different values of k:

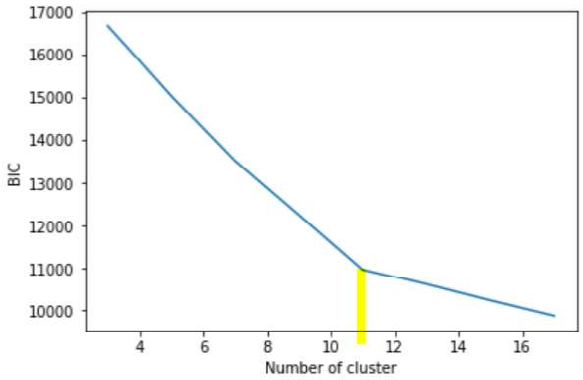
Plotting SSE :

```
sse = {}
for k in range(3,19,2):
    kmeans = KMeans(n_clusters=k, max_iter=1000).fit(data2)
    data2["clusters"] = kmeans.labels_
    #print(data["clusters"])
    sse[k] = kmeans.inertia_ # Inertia: Sum of distances of samples to their closest cluster center
plt.figure()
plt.plot(list(sse.keys()), list(sse.values()))
plt.xlabel("Number of cluster")
plt.ylabel("SSE")
plt.show()
```



Plotting BIC :

```
import math
bic = {}
n=6600
for k in range(3,19,2):
    kmeans = KMeans(n_clusters=k, max_iter=1000).fit(data2)
    data2["clusters"] = kmeans.labels_
    #print(data["clusters"])
    bic[k] = n*math.log10(kmeans.inertia_/n)+math.log10(n)*k*3 # Inertia: Sum of distances of samples to their closest cluster c
enter
plt.figure()
plt.plot(list(bic.keys()), list(bic.values()))
plt.xlabel("Number of cluster")
plt.ylabel("BIC")
plt.show()
```



Best Number of Clusters for this dataset is 11

Method for finding the best number of clusters:

The best number of clusters of a dataset is found using SSE and BIC plot of the dataset. The above figures depict the variation of SSE and BIC values with respect to the number of clusters in the dataset.

There is a distinct knee in the SSE and a distinct peak in the BIC value when the number of clusters is equal to 11. This clearly indicates that the value of SSE and BIC change their pattern and hence the optimum number of clusters suitable for this dataset is selected to be 11.

SSE method suggests that we can look at which point of the graph, further increasing k would stop yielding a substantial increase in SSE “elbow method”. Using SSE, we compute BIC by considering other parameters like no. of datapoints and find the knee point to get the optimal clusters.

c. ***(8) For the best number of clusters selected above, plot the scatter plot of the data showing the points of each cluster with a different color/symbol. Mark the points on the scatter plot that belong to clusters other than what your intuition says. Why did k-means algorithm place them in these different clusters – explain very briefly.***

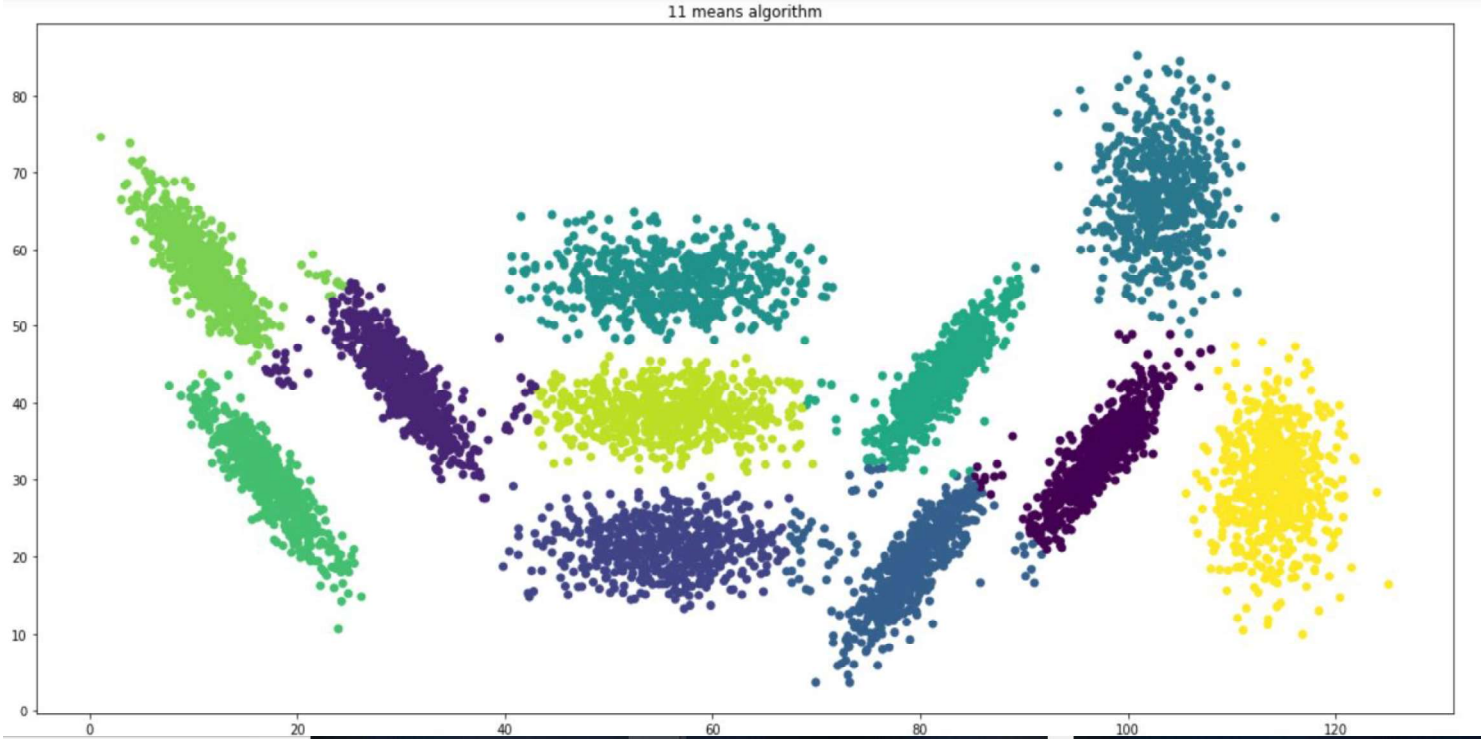
Solution of 2(c):

Scatter Plot of the data showing datapoints of each Cluster:

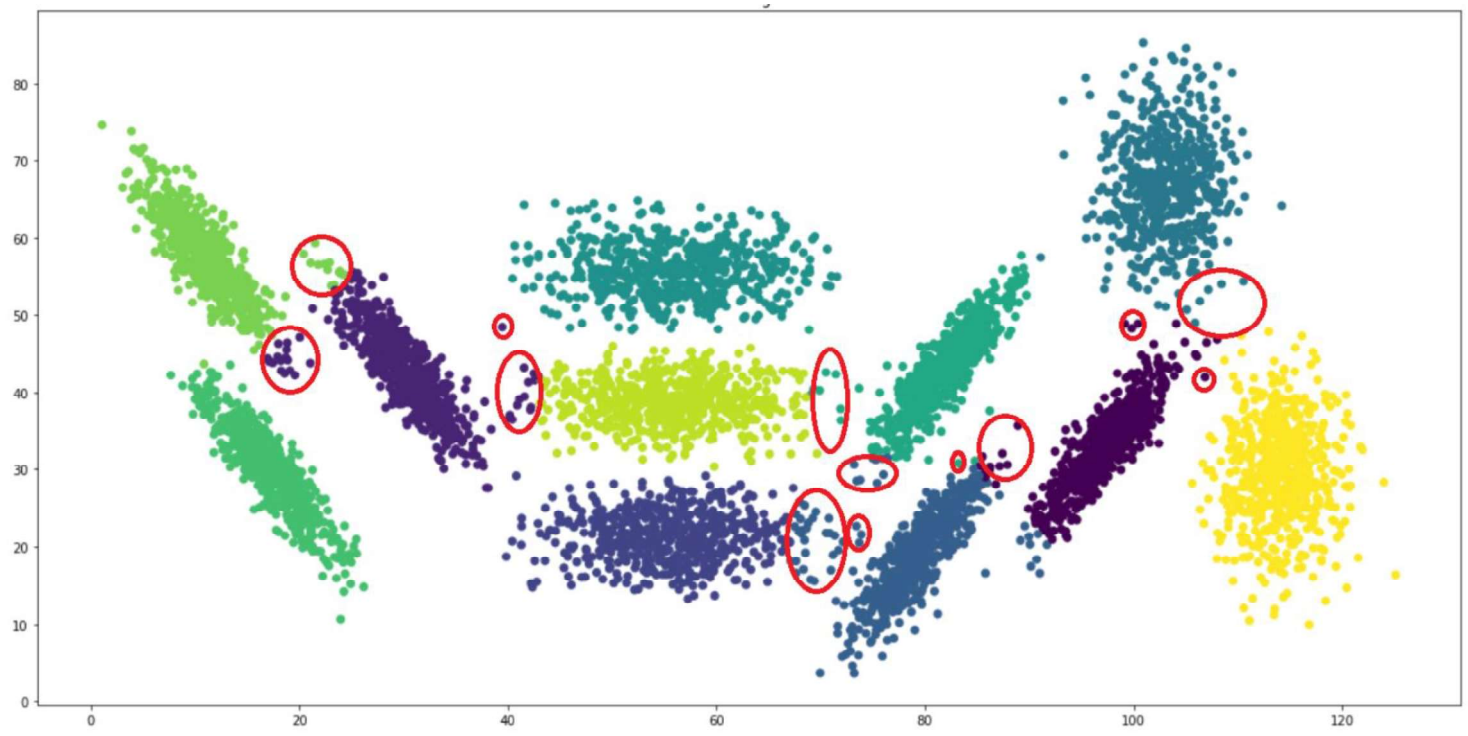
Python Commands for plotting the scatter:

```
plt.figure(figsize=(20,10))
plt.scatter(data2.X,data2.Y,c=y_pred_11)
plt.title("5 means algorithm")
plt.show()
```

Scatter Plot:



Plot indicating the points on the scatter that belong to a different cluster compared to intuition plot:



Brief Explanation of these misclassifications by k means:

1.K means has problems when clusters are of different sizes, densities and globular shapes. As per the dataset given the sizes of each intuitive clusters are different ,also the datasets have distinct densities and the shapes of the clusters are different.

Hence this problem of misclassification occurs in k means.

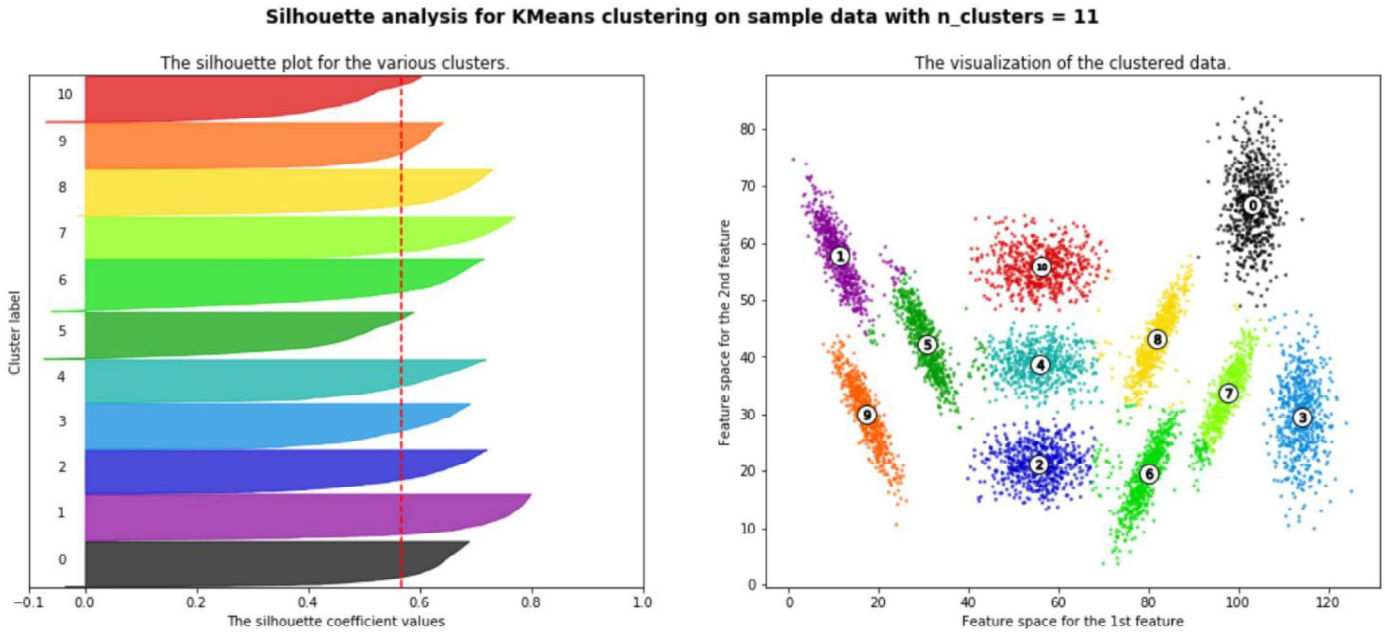
2.K means algorithm is based on selecting initial centroids and all the points are classified based on the initial centroids. Hence k means only classifies the points based on the distance from the initial centroid to each point but doesn't consider the structure or the density of the dataset. This would cause the problem of wrong classifications whenever a different dense cluster are present in the data. Density based clustering algorithms are best suitable for this kind of data.

d. (5) Plot the silhouette diagram for the best clustering you have selected. Comment on the characteristics of the silhouette diagram that you think are informative about this clustering. Comment using the cluster numbers and their plots on the silhouette diagram.

Solution of 2(d):

Python Commands for plotting the silhouette for k=11:

```
# Compute the silhouette scores for each sample
sample_silhouette_values = silhouette_samples(data2, cluster_labels)

y_lower = 10
for i in range(n_clusters):
    # Aggregate the silhouette scores for samples belonging to
    # cluster i, and sort them
    ith_cluster_silhouette_values = \
        sample_silhouette_values[cluster_labels == i]

    ith_cluster_silhouette_values.sort()

    size_cluster_i = ith_cluster_silhouette_values.shape[0]
    y_upper = y_lower + size_cluster_i

    color = cm.nipy_spectral(float(i) / n_clusters)
    ax1.fill_betweenx(np.arange(y_lower, y_upper),
                      0, ith_cluster_silhouette_values,
                      facecolor=color, edgecolor=color, alpha=0.7)
```



Comments on the characteristics of silhouette diagram:

1.By looking at the silhouette analysis of the dataset, we can see that all the silhouette values are greater than average silhouette .The silhouette coefficients of all the clusters are consistent except 1 and hence it is the best silhouette coefficient with the given dataset.

2.By looking at the thickness of the silhouette plot ,we observe that almost all the clusters are of similar size, which is the best sign for cluster selection and k=11 is the best number of clusters possible with the dataset.

Comments on the cluster numbers and plots on silhouette diagram:

Cluster 5 and Cluster 10 are having lesser silhouette values. A lesser value indicates that the sample is closer to nearest cluster compared to the remaining clusters which are having high silhouette values.

The highest silhouette value is observed for Cluster 1 which has a silhouette value of value of 0.8,which indicates that the cluster is far from the other clusters. As seen from the cluster plot, we can view that the 1st cluster which is violet in color is far from other clusters.

The thickness of the cluster 5 is the least, which indicates that the cluster is having lesser datapoints compared to the other clusters.

Silhouette values is negative for clusters 5 and 10 which indicates that some of the points should not belong to these clusters.

Thus, the silhouette plot indicates the following points.

1.The higher the silhouette coefficient, the cluster is far from the other clusters.

2. The thickness of the cluster is indicated by the thickness of the silhouette plot and uniform thickness indicate that the clusters are better and of same size.

3.Negative silhouette values indicate that some points do not belong to the mentioned cluster.

e. **(12) Perform single-linkage hierarchical clustering for this data and cut the dendrogram to obtain 11 clusters. There are options/parameters in most toolboxes to generate a given number of clusters. Plot the 2-D scatter plot of the dataset showing data points of each of the 11 clusters with different color/symbol.**

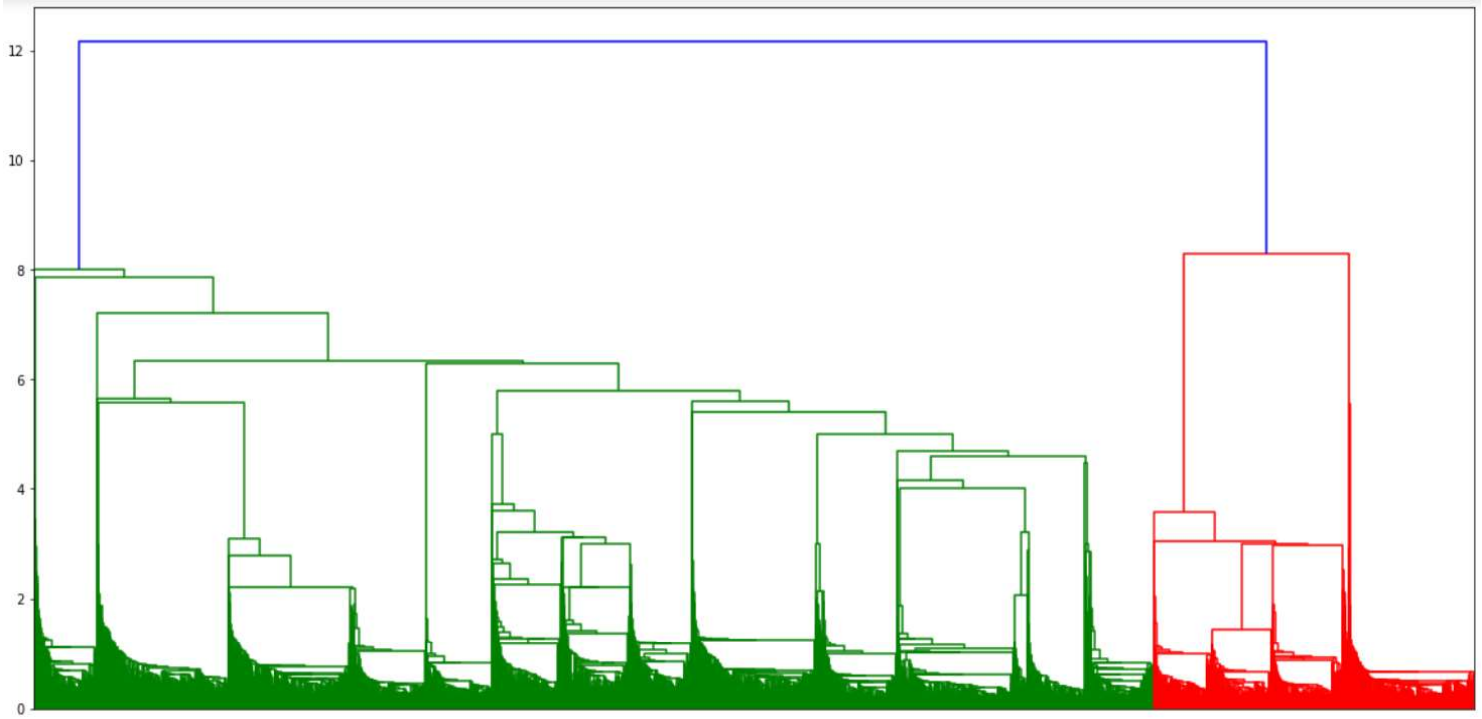
Solution of 2(e):

Python Commands for cutting the dendrogram and partition using hierarchical clustering:

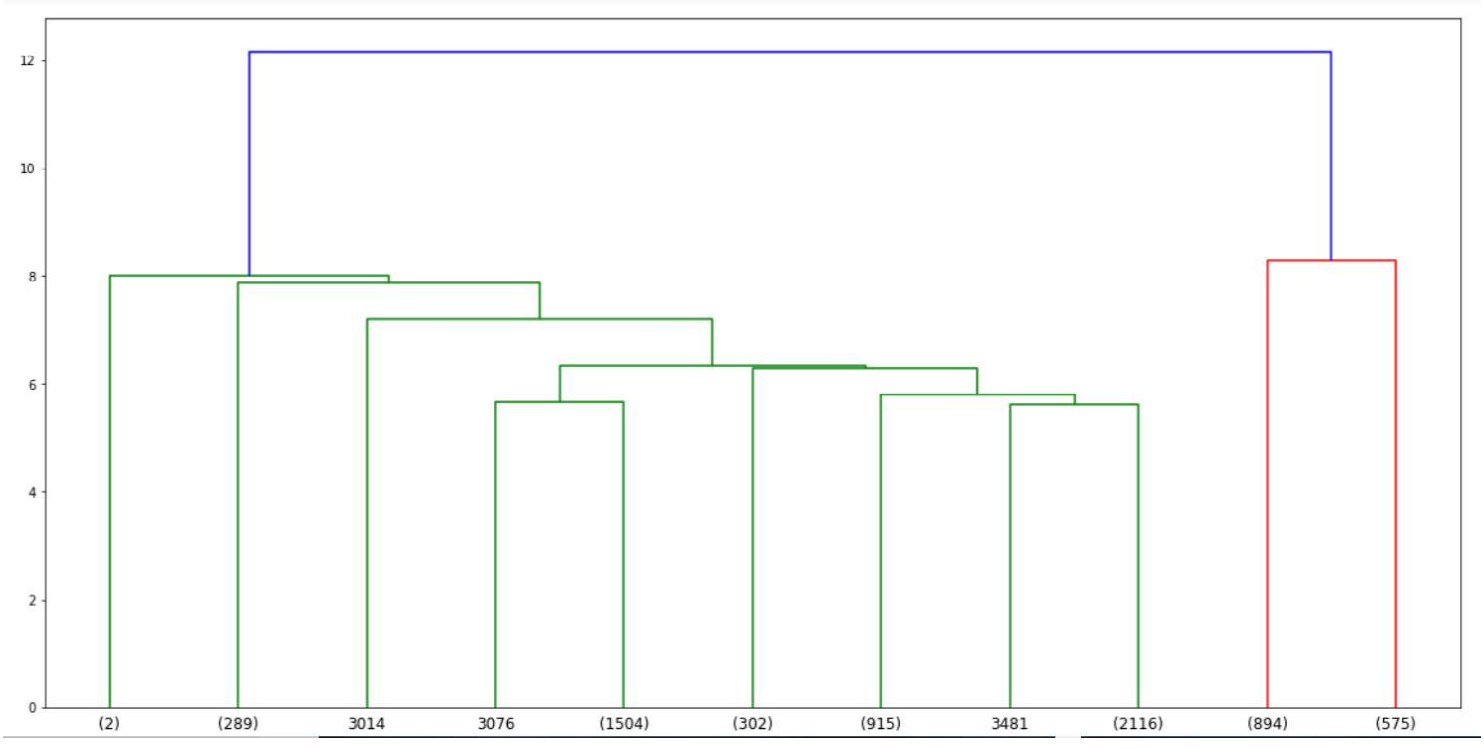
```
from scipy.cluster.hierarchy import dendrogram, linkage
from matplotlib import pyplot as plt

Z = linkage(data2, 'single')
plt.figure(figsize=(20, 10))
dendrogram(Z,
           orientation='top',
           distance_sort='ascending',
           show_leaf_counts=True)
plt.show()
```

```
import matplotlib.markers
plt.figure(figsize=(20, 10))
plt.scatter(data2.X, data2.Y, c=clusters,cmap='hsv')
plt.show()
```

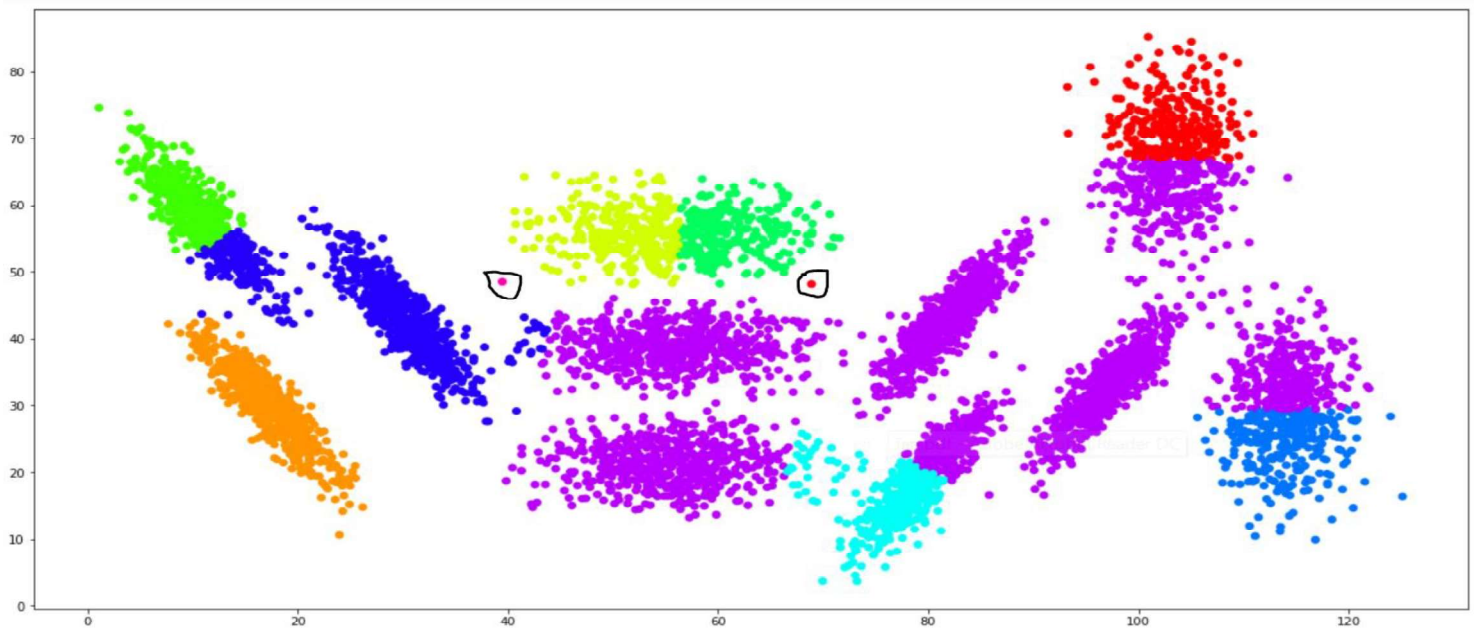
Dendrogram of all clusters:



Dendrogram of 11 clusters:



2-D Scatter plot of data points showing 11 clusters:

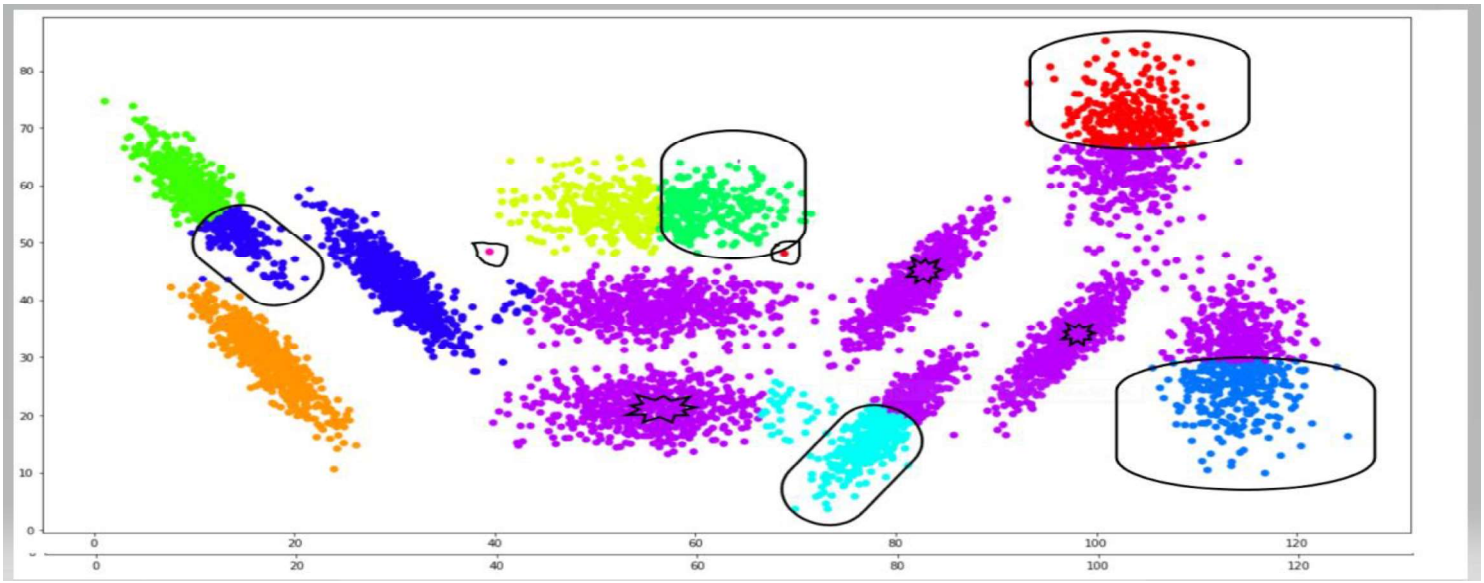


Comments on the Plots:

The plots indicate the hierarchical clustering using a single linkage method. The plot clearly suggests that there are several outliers in the data . Also, there are 2 1-point clusters formed which are circled. The reasons for this counter intuitive classifications will be mentioned in the next question.

f. (5) Mark any data points on this scatter plot that are clustered differently from your intuitive view of the correct clusters. Explain why Single-linkage clustering may have placed them in counter-intuitive clusters.

Solution of 2(f):



Reason for single linkage placing the points in counter intuitive way:

1.Single linkage is susceptible to outliers and noise. When we observe the dataset closely ,we can clearly view 2 outliers in the data, which are forming individual clusters ,due to which there is inconsistency in the data of 11 clusters.

2.Single linkage clustering is based on the minimum distance between the points. In case of the given dataset ,there are several places where the data has least distance between the intuitive clusters hence, they are formed as a single cluster. For example, consider the cluster in violet ,due to minimum distance criterion it has occupied many datapoints.

3.The third reason for counter-intuitive performance of single cluster is due to the number of clusters 11 mentioned in the code. As the number of clusters of single linkage is mentioned 11,it has remodified the cluster assignment itself to the above model and hence we see a lot of counter-intuitive performance.