Name: Shreyas Namjoshi

Assignment: ML UL Assignment 2

PGID: 12110103

Step 1: Download the Wine data from the UCI machine learning repository (Wine dataset- UCI Repository)

In [104…

```
import pandas as pd

## Read the dataset from online library
df_wine = pd.read_csv('https://archive.ics.uci.edu/ml/'
                      'machine-learning-databases/wine/wine.data',
                      header=None)

## Name the columns
df_wine.columns = ['Class label', 'Alcohol', 'Malic acid', 'Ash',
                   'Alcalinity of ash', 'Magnesium', 'Total phenols',
                   'Flavanoids', 'Nonflavanoid phenols', 'Proanthocyanins',
                   'Color intensity', 'Hue',
                   'OD280/OD315 of diluted wines', 'Proline']
```

In [105…

```
df_wine
```

Out[105…

	Class label	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols	Proanth
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	
...	...	...	...	...	...	...	...	...	...	
173	3	13.71	5.65	2.45	20.5	95	1.68	0.61	0.52	
174	3	13.40	3.91	2.48	23.0	102	1.80	0.75	0.43	
175	3	13.27	4.28	2.26	20.0	120	1.59	0.69	0.43	
176	3	13.17	2.59	2.37	20.0	120	1.65	0.68	0.53	
177	3	14.13	4.10	2.74	24.5	96	2.05	0.76	0.56	

178 rows × 14 columns

In [106…

```
#removing first column from the data
df_wine_new=df_wine.iloc[:,1:]
df_wine_new
```

Out[106…

	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols	Proanthocyani
0	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.
1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.

	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols	Proanthocyani
2	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.
3	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.
4	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.
...	...	...	...	...	...	...	...	...	
173	13.71	5.65	2.45	20.5	95	1.68	0.61	0.52	1.
174	13.40	3.91	2.48	23.0	102	1.80	0.75	0.43	1.
175	13.27	4.28	2.26	20.0	120	1.59	0.69	0.43	1.
176	13.17	2.59	2.37	20.0	120	1.65	0.68	0.53	1.
177	14.13	4.10	2.74	24.5	96	2.05	0.76	0.56	1.

178 rows × 13 columns

Step 2: Do a Principal Components Analysis (PCA) on the data. Please include (copy-paste) the relevant software outputs in your submission while answering the following questions.

a. Enumerate the insights you gathered during your PCA exercise. Please do not clutter your report with too many insignificant insights as it will dilute the value of your other significant findings.

In [107…

```
#Normalizing the data
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()
df_scaler=sc.fit_transform(df_wine_new)
df_scaler
```

Out[107…

```
array([[ 1.51861254, -0.5622498 ,  0.23205254, ...,  0.36217728,
         1.84791957,  1.01300893],
       [ 0.24628963, -0.49941338, -0.82799632, ...,  0.40605066,
         1.1134493 ,  0.96524152],
       [ 0.19687903,  0.02123125,  1.10933436, ...,  0.31830389,
         0.78858745,  1.39514818],
       ...,
       [ 0.33275817,  1.74474449, -0.38935541, ..., -1.61212515,
        -1.48544548,  0.28057537],
       [ 0.20923168,  0.22769377,  0.01273209, ..., -1.56825176,
        -1.40069891,  0.29649784],
       [ 1.39508604,  1.58316512,  1.36520822, ..., -1.52437837,
        -1.42894777, -0.59516041]])
```

In [108…

```
#Perform PCA on the data and print the scores
from sklearn.decomposition import PCA

pca1 = PCA() # creates an instance of PCA class
results = pca1.fit(df_scaler) # applies PCA on predictor variables
Z = results.transform(df_scaler) # create a new array of latent variables
print(results.score)
df_Z_PCA=pd.DataFrame(Z)
df_Z_PCA
```

```
<bound method PCA.score of PCA()>
```

Out[108…

	0	1	2	3	4	5	6	7	8

	0	1	2	3	4	5	6	7	8
0	3.316751	-1.443463	-0.165739	-0.215631	0.693043	-0.223880	0.596427	0.065139	0.641443
1	2.209465	0.333393	-2.026457	-0.291358	-0.257655	-0.927120	0.053776	1.024416	-0.308847
2	2.516740	-1.031151	0.982819	0.724902	-0.251033	0.549276	0.424205	-0.344216	-1.177834
3	3.757066	-2.756372	-0.176192	0.567983	-0.311842	0.114431	-0.383337	0.643593	0.052544
4	1.008908	-0.869831	2.026688	-0.409766	0.298458	-0.406520	0.444074	0.416700	0.326819
...	...	...	...	...	...	...	...	...	...
173	-3.370524	-2.216289	-0.342570	1.058527	-0.574164	-1.108788	0.958416	-0.146097	-0.022498
174	-2.601956	-1.757229	0.207581	0.349496	0.255063	-0.026465	0.146894	-0.552427	-0.097969
175	-2.677839	-2.760899	-0.940942	0.312035	1.271355	0.273068	0.679235	0.047024	0.001222
176	-2.387017	-2.297347	-0.550696	-0.688285	0.813955	1.178783	0.633975	0.390829	0.057448
177	-3.208758	-2.768920	1.013914	0.596903	-0.895193	0.296092	0.005741	-0.292914	0.741660

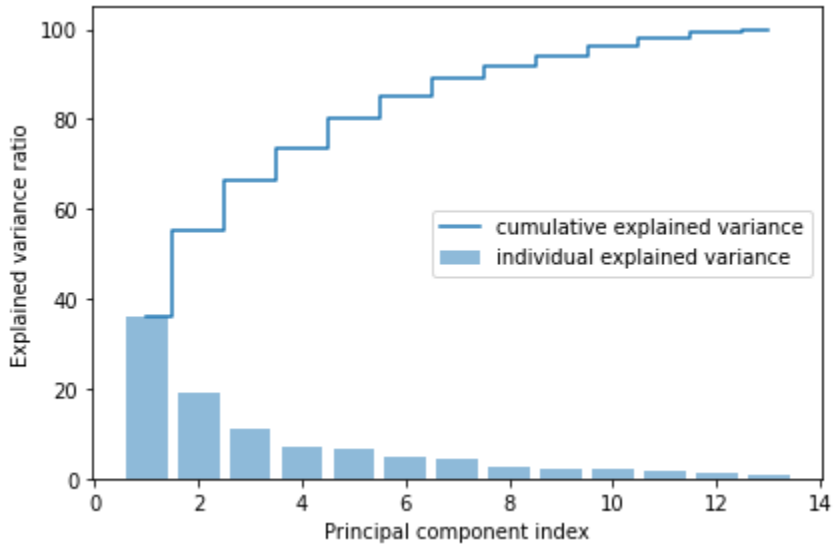
178 rows × 13 columns

In [109…

```
#Plot the explained variance for the PCA
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline

## Cumulative Variance
cum_var_exp = np.cumsum(results.explained_variance_ratio_*100)

plt.bar(range(1, 14),results.explained_variance_ratio_*100 , alpha=0.5, align='center',
        label='individual explained variance')
plt.step(range(1, 14), cum_var_exp, where='mid',
         label='cumulative explained variance')
plt.ylabel('Explained variance ratio')
plt.xlabel('Principal component index')
plt.legend(loc='best')
plt.tight_layout()
plt.show()
```



The above plot indicates that the first principal component alone accounts for approximately 36 percent of the total variance in the dataset. Also, we can see that the first two principal components combined explain almost 55 percent of the variance in the dataset From the plot

generated, we see that almost 85% of variance in data can be acquired from first 6 Components, So first 6 components can be choosen for the analysis.

b. What are the social and/or business values of those insights, and how the value of those insights can be harnessed—enumerate actionable recommendations for the identified stakeholder in this analysis?

In [110…

```
#Loading the components data
PCA_Data=pd.DataFrame(results.components_, columns=['Alcohol', 'Malic acid', 'Ash',
                    'Alcalinity of ash', 'Magnesium', 'Total phenols',
                    'Flavanoids', 'Nonflavanoid phenols', 'Proanthocyanins',
                    'Color intensity', 'Hue',
                    'OD280/OD315 of diluted wines', 'Proline'])
PCA_Data
```

Out[110…

	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols	Proa
0	0.144329	-0.245188	-0.002051	-0.239320	0.141992	0.394661	0.422934	-0.298533	
1	-0.483652	-0.224931	-0.316069	0.010591	-0.299634	-0.065040	0.003360	-0.028779	
2	-0.207383	0.089013	0.626224	0.612080	0.130757	0.146179	0.150682	0.170368	
3	-0.017856	0.536890	-0.214176	0.060859	-0.351797	0.198068	0.152295	-0.203301	
4	-0.265664	0.035214	-0.143025	0.066103	0.727049	-0.149318	-0.109026	-0.500703	
5	-0.213539	-0.536814	-0.154475	0.100825	-0.038144	0.084122	0.018920	0.258594	
6	-0.056396	0.420524	-0.149171	-0.286969	0.322883	-0.027925	-0.060685	0.595447	
7	-0.396139	-0.065827	0.170260	-0.427970	0.156361	0.405934	0.187245	0.233285	
8	0.508619	-0.075283	-0.307694	0.200449	0.271403	0.286035	0.049578	0.195501	
9	0.211605	-0.309080	-0.027125	0.052799	0.067870	-0.320131	-0.163151	0.215535	
10	-0.225917	0.076486	-0.498691	0.479314	0.071289	0.304341	-0.025694	0.116896	
11	-0.266286	0.121696	-0.049622	-0.055743	0.062220	-0.303882	-0.042899	0.042352	
12	0.014970	0.025964	-0.141218	0.091683	0.056774	-0.463908	0.832257	0.114040	

The above is the coefficient matrix. The first row is the coefficients that generated the first PC. In other words, the first PC was generated using the following formula:

PC1(index 0 above) = (Alcohol *0.144329) + (Malic acid* -0.245188) + … + (Proline * 0.286752)

and so on till

PC13(index 12 above)= (Alcohol *0.014970) + (Malic acid* 0.025964) + … + (Proline * 0.014447)

We know that the loading the correlation value above 0.5 is important.

Below are the observations with respect to components:

- Comp.3 is positively correlated with Ash, Alcainilty.
- Comp.4 is positively correlated with Malic Acid.
- Comp.5 is positively related with Mangnesium.
- Comp.6 is negatively correlated to Malic Acid.These component wines are less acidic in nature
- Comp.13 has positive correlation with respect to flavonides.
- ositive correlation with respect to color intensity and OD280/OD315 ratio.

Step 3: Do a cluster analysis—you may try different algorithms or approaches and go with the one that you find most appropriate— using (i) all chemical measurements (ii) using two most significant PC scores. Please include (copy- paste) the relevant software outputs in your submission while answering the following questions.

c. Any more insights you come across during the clustering exercise?

d) Are there clearly separable clusters of wines? How many clusters did you go with? How the clusters obtained in part (i) are different from or similar to clusters obtained in part (ii), qualitatively?

Here we will go by KMeans clustering as we have method like elbow curve that gives you proper number of clusters just by looking at the graph. Interpreting the dendogram for this many records is difficult though we can still make out clusters on the broader level.

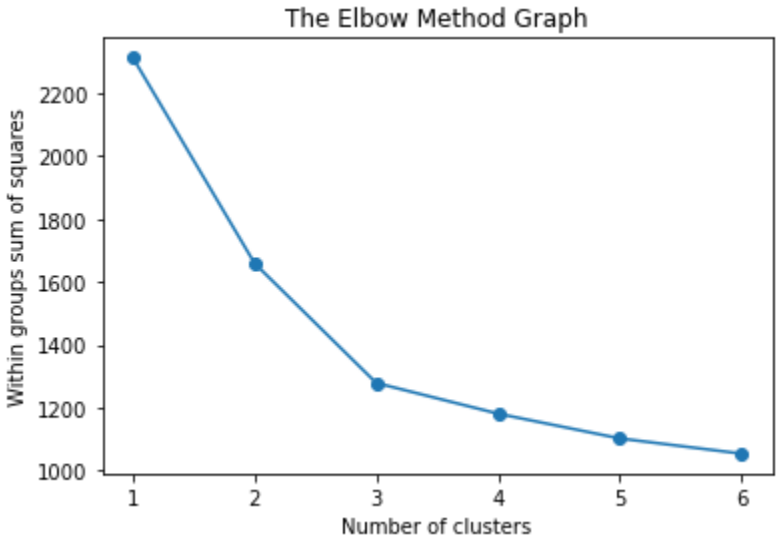
In [114…

```
from sklearn.cluster import KMeans
```

In [115…

```
#Clustering on the basis of all the fields
## Determine number of clusters
Cluster_Variability = []
for i in range(1,7):
    kmeans = KMeans(n_clusters=i).fit(df_scaler)
    Cluster_Variability.append(kmeans.inertia_)
#Plot the elbow graph
plt.plot(range(1,7),Cluster_Variability,marker='o')
plt.title('The Elbow Method Graph')
plt.xticks(range(1,7))
plt.xlabel('Number of clusters')
plt.ylabel('Within groups sum of squares')
plt.show()
```

```
D:\Users\snamjoshi\Anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:881: UserWarnin
g: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks t
han available threads. You can avoid it by setting the environment variable OMP_NUM_THREAD
S=1.
  warnings.warn(
```



From the elbow curve we see that number of clusters we can take as 3

In [116…

```
fit1 = KMeans(n_clusters=3, max_iter = 10, random_state=0).fit(df_scaler)
```

In [117…
```
df_wine_new['ClusterID']=fit1.labels_
df_wine_new
```

Out[117…

	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols	Proanthocyani
0	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.
1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.
2	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.
3	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.
4	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.
...	...	...	...	...	...	...	...	...	
173	13.71	5.65	2.45	20.5	95	1.68	0.61	0.52	1.
174	13.40	3.91	2.48	23.0	102	1.80	0.75	0.43	1.
175	13.27	4.28	2.26	20.0	120	1.59	0.69	0.43	1.
176	13.17	2.59	2.37	20.0	120	1.65	0.68	0.53	1.
177	14.13	4.10	2.74	24.5	96	2.05	0.76	0.56	1.

178 rows × 14 columns

In [118…
```
#Aggregating columns on K Means clusters
print("K-Clust")
df_wine_new.groupby('ClusterID').agg({'Alcohol':'mean','Malic acid':'mean','Ash':'mean','
                                      'Proanthocyanins':'mean','Color intensity':'mean','Hue':
```

K-Clust

Out[118…

ClusterID	0	1	2
Alcohol	13.134118	12.240455	13.711475
Malic acid	3.307255	1.899697	1.997049
Ash	2.417647	2.246364	2.453770
Alcalinity of ash	21.241176	20.190909	17.281967
Magnesium	98.666667	93.136364	107.786885
Total phenols	1.683922	2.261818	2.842131
Flavanoids	0.818824	2.095909	2.969180
Nonflavanoid phenols	0.451961	0.359394	0.289180
Proanthocyanins	1.145882	1.627879	1.922951
Color intensity	7.234706	3.018939	5.444590
Hue	0.691961	1.060697	1.067705
OD280/OD315 of diluted wines	1.696667	2.816818	3.154754
Proline	619.058824	509.484848	1110.639344

Cluster 1: This wine is has better score for Malic Acid hence its sweeter in taste.The color intensity refers to the degree of lightness of the color, and the greater the intensity, the darker the color. It reflects the nature of the grapes that make the wine.Thus this wine will be more darker.The lower absorbance ratio of OD280/OD315 indicates lower protein purity.This Cluster of wine have lower
n alkaline wine.Overall nutritional value is less.

Cluster 2: Proline is known to be an amino acid that regulates the flavor of the wine.Lower the Proline,lesser is the acid that regulates the flavour of wine.It has lesser Malic Acid and hence it will be bitter in taste.The color intensity is has smallest units, hence the wine is lighter in colour.Overall nutritional value is between cluster 1 and cluster 3.

Cluster 3: The type of wine has taste between cluster 1 and cluster 2. This has highest amount of magnesium,phenols and flavonoids and proline .This also has highest absorbance ratio.Thus this seems to be healtiest of the lot as they have highest phenols and flavonoids.The highest absorbance ratio of OD280/OD315 indicates higher protein purity.This cluster also has highest amount of Proanthocyanins hence the smell will be bitter as compared to cluster 1 and cluster 2.

In [119…

```
#Clustering using 2 most significants components.So taking the first 2 components

from sklearn.decomposition import PCA

pca2 = PCA(n_components=2) # creates an instance of PCA class
results = pca2.fit(df_scaler) # applies PCA on predictor variables
Z_PCA = results.transform(df_scaler) # create a new array of latent variables
Z_PCA

Z_PCA_DF= pd.DataFrame(Z_PCA,columns=['PComp1','PComp2'])
Z_PCA_DF
```

Out[119…

	PComp1	PComp2
0	3.316751	-1.443463
1	2.209465	0.333393
2	2.516740	-1.031151
3	3.757066	-2.756372
4	1.008908	-0.869831
...	...	...
173	-3.370524	-2.216289
174	-2.601956	-1.757229
175	-2.677839	-2.760899
176	-2.387017	-2.297347
177	-3.208758	-2.768920

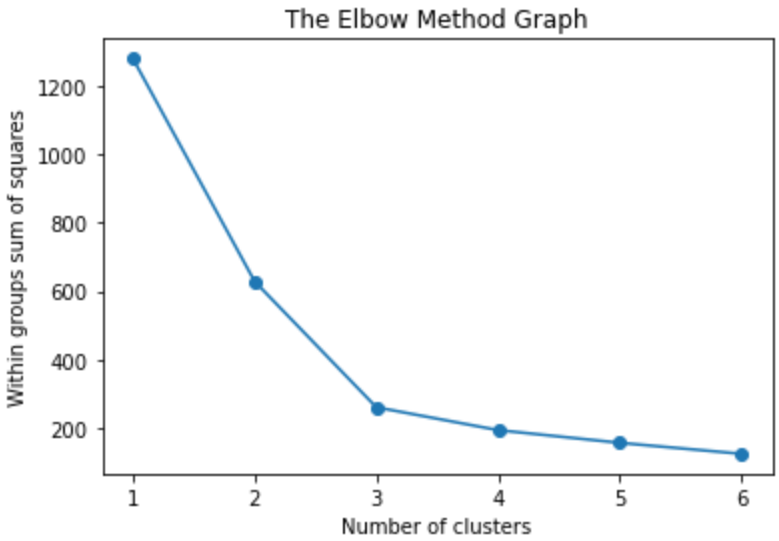
178 rows × 2 columns

In [120…

```
## Determine number of clusters
Cluster_Variability = []
for i in range(1,7):
    kmeans = KMeans(n_clusters=i).fit(Z_PCA_DF)
    Cluster_Variability.append(kmeans.inertia_)
#Plot the elbow graph
plt.plot(range(1,7),Cluster_Variability,marker='o')
plt.title('The Elbow Method Graph')
plt.xticks(range(1,7))
plt.xlabel('Number of clusters')
plt.ylabel('Within groups sum of squares')
plt.show()
```

```
hi\Anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:881: UserWarnin
```

```
g: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks t
han available threads. You can avoid it by setting the environment variable OMP_NUM_THREAD
S=1.
  warnings.warn(
```



From the elbow curve we see that number of clusters we can take as 3

In [121…
```
fit2 = KMeans(n_clusters=3, max_iter = 10, random_state=0).fit(Z_PCA_DF)
```

In [122…
```
Z_PCA_DF['ClusterIDWithPCA']=fit2.labels_
```

In [123…
```
Z_PCA_DF
```

Out[123…

	PComp1	PComp2	ClusterIDWithPCA
0	3.316751	-1.443463	1
1	2.209465	0.333393	1
2	2.516740	-1.031151	1
3	3.757066	-2.756372	1
4	1.008908	-0.869831	1
...	...	...	...
173	-3.370524	-2.216289	2
174	-2.601956	-1.757229	2
175	-2.677839	-2.760899	2
176	-2.387017	-2.297347	2
177	-3.208758	-2.768920	2

178 rows × 3 columns

In [124…
```
finalDf= pd.concat([df_wine_new, Z_PCA_DF], axis=1)
finalDf
```

Out[124…

	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols	Proanthocyani
	1	2.43	15.6	127	2.80	3.06	0.28		2.

	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols	Proanthocyani
1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.
2	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.
3	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.
4	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.
...	...	...	...	...	...	...	...	...	
173	13.71	5.65	2.45	20.5	95	1.68	0.61	0.52	1.
174	13.40	3.91	2.48	23.0	102	1.80	0.75	0.43	1.
175	13.27	4.28	2.26	20.0	120	1.59	0.69	0.43	1.
176	13.17	2.59	2.37	20.0	120	1.65	0.68	0.53	1.
177	14.13	4.10	2.74	24.5	96	2.05	0.76	0.56	1.

178 rows × 17 columns

In [126…

```
#Aggregating columns on K Means clusters
print("K-Clust")
finalDf.groupby('ClusterIDWithPCA').agg({'Alcohol':'mean','Malic acid':'mean','Ash':'mean
                                         'Proanthocyanins':'mean','Color intensity':'mean','Hue':
```

```
K-Clust
```

Out[126…

ClusterIDWithPCA	0	1	2
Alcohol	12.238308	13.659219	13.151633
Malic acid	1.931385	1.975781	3.344490
Ash	2.219385	2.463750	2.434694
Alcalinity of ash	19.898462	17.596875	21.438776
Magnesium	92.830769	107.312500	99.020408
Total phenols	2.204308	2.859688	1.678163
Flavanoids	1.989231	3.012656	0.797959
Nonflavanoid phenols	0.365538	0.290000	0.450816
Proanthocyanins	1.587692	1.921719	1.163061
Color intensity	2.992615	5.406250	7.343265
Hue	1.051631	1.069688	0.685918
OD280/OD315 of diluted wines	2.769231	3.157188	1.690204
Proline	506.353846	1082.562500	627.551020

Cluster 1: Proline is known to be an amino acid that regulates the flavor of the wine.Lower the Proline,lesser is the acid that regulates the flavour of wine.It has lesser Malic Acid and hence it will be bitter in taste.The color intensity is has smallest units, hence the wine is lighter in colour.The lower absorbance ratio of OD280/OD315 indicates lower protein purity.This Cluster of wine have lower protein purity

Cluster 2: This wine is has better score for Malic Acid hence its sweeter in taste than in cluster 1.The color intensity refers to the degree of lightness of the color, and the greater the intensity, r. It reflects the nature of the grapes that make the wine.Thus this wine will be

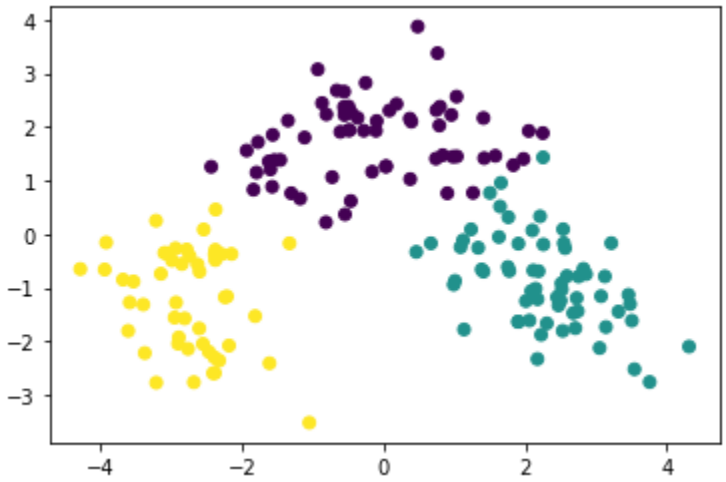
more darker than in cluster 1.The highest absorbance ratio of OD280/OD315 indicates higher protein purity.This cluster also has highest amount of Proanthocyanins hence the smell will be bitter as compared to cluster 1 and cluster 3.This has highest magnesium, amount of phenols and flavonoids and proline.This seems to be healthiest wine.

Cluster 3: The type of wine has highest color intensity hence it will be darker.It has lowest flavonoids,phenols hence this type is not as healthy as wines in cluster 1 and 2. It will be sweeter in taste but over all nutritional value is less.

plotting both the cluster analysis

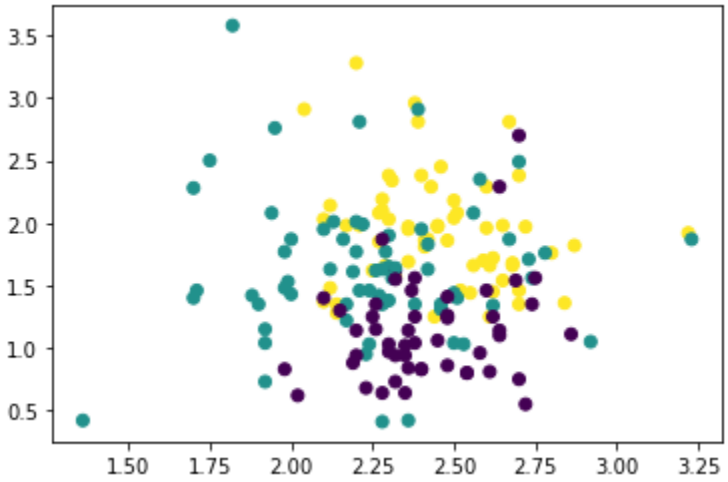
In [127…

```
#with PCA
labels2 = fit2.predict(Z_PCA_DF.iloc[:,:2])
plt.scatter(Z_PCA_DF['PComp1'], Z_PCA_DF['PComp2'], c=labels2)
plt.show()
```



In [128…

```
#Clustering with respect the normal chemical data
labels1 = fit1.predict(df_scaler)
plt.scatter(df_wine_new['Ash'], df_wine_new['Proanthocyanins'], c=labels1)
plt.show()
```



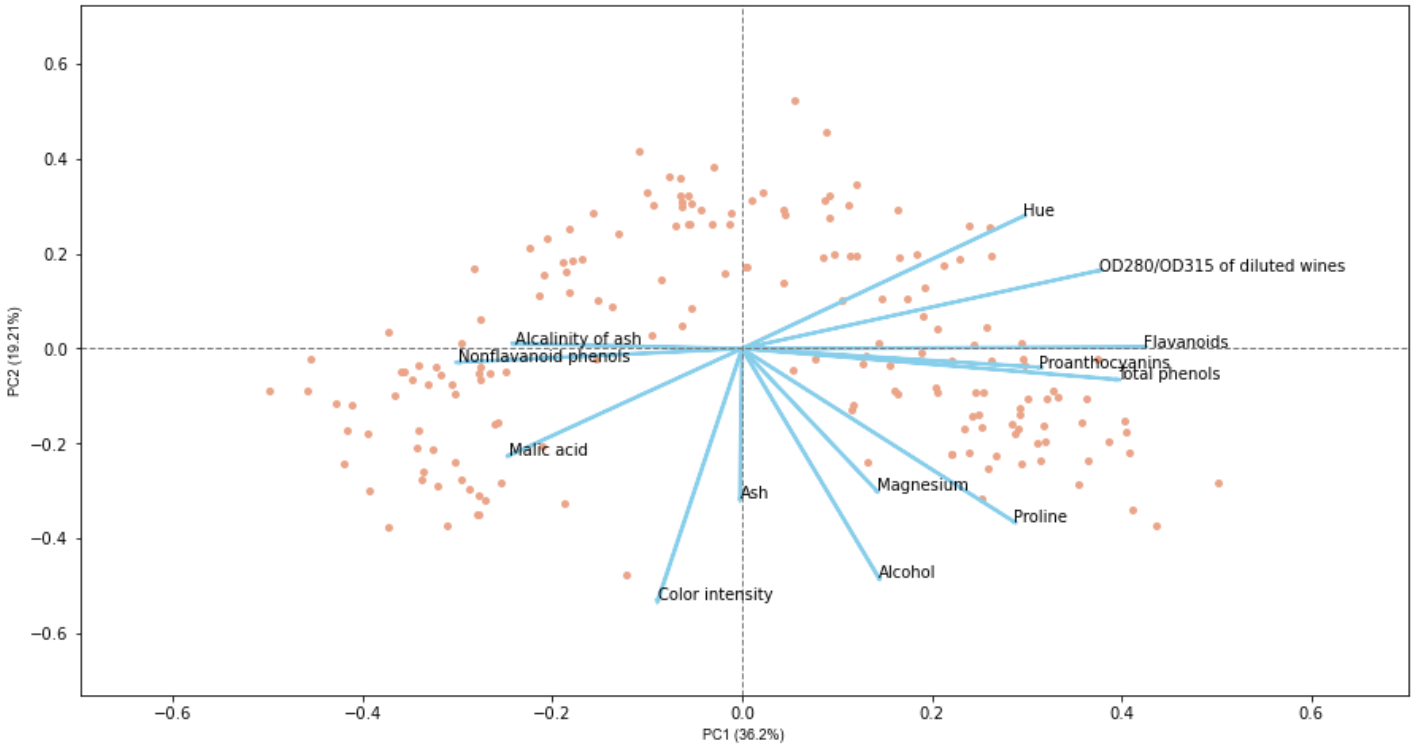
Though the cluster formation in both the cases is similar.PCA gives us more evident cluster graphically. If we plot the cluster using any two attributes from normal data, we see that there cluster are overlapped though there is demarkation while we see in the data.In PCA plotted clusters,we see that the clusters are identified seperately using two PCA's choosen.

e. Could you suggest a subset of the chemical measurements that can separate wines more

distinctly? How did you go about choosing that subset? How do the rest of the measurements that were not included while clustering, vary across those clusters?

In [129…
```
from bioinfokit.visuz import cluster
```

In [130…
```
# get 2D biplot
cluster.biplot(cscore=Z, loadings=results.components_,labels= PCA_Data.columns.values,var
    var2=round(results.explained_variance_ratio_[1]*100, 2),show=True,dim=(15,8),arrowlin
```



Based on PCA components we see that we can seperate wines based on healthy substances based on magnesium,flavonoids and phenols,wine's thickness based on color intensityand hue,its smell based on Proanthocyanins, taste which are based on malic acid and Protein purity which is based on absorbance ratio of OD280/OD315 and proline. We did not consider remaining factors that they showed lower variance in data as compared to above stated factors.

In [ ]:
```
```