Out[5]: Click here to toggle on/off the raw code.

Using abalone dataset: Assignment 1 revisited

Out[223]:

	Sex	Length	Diameter	Height	Whole_weight	Sucked_weight	Viscera_weight	Shell_weight
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055

```
            Length     Diameter       Height  Whole_weight  Sucked_weight  \
count  4177.000000  4177.000000  4177.000000   4177.000000    4177.000000
mean      0.523992     0.407881     0.139516      0.828742       0.359367
std       0.120093     0.099240     0.041827      0.490389       0.221963
min       0.075000     0.055000     0.000000      0.002000       0.001000
25%       0.450000     0.350000     0.115000      0.441500       0.186000
50%       0.545000     0.425000     0.140000      0.799500       0.336000
75%       0.615000     0.480000     0.165000      1.153000       0.502000
max       0.815000     0.650000     1.130000      2.825500       1.488000

       Viscera_weight  Shell_weight        Rings
count     4177.000000   4177.000000  4177.000000
mean         0.180594      0.238831     9.933684
std          0.109614      0.139203     3.224169
min          0.000500      0.001500     1.000000
25%          0.093500      0.130000     8.000000
50%          0.171000      0.234000     9.000000
75%          0.253000      0.329000    11.000000
max          0.760000      1.005000    29.000000
```

	mean	median	variance	skew	kurtosis
Length	0.523992	0.5450	0.014422	-0.639873	0.064621
Diameter	0.407881	0.4250	0.009849	-0.609198	-0.045476
Height	0.139516	0.1400	0.001750	3.128817	76.025509
Whole_weight	0.828742	0.7995	0.240481	0.530959	-0.023644
Sucked_weight	0.359367	0.3360	0.049268	0.719098	0.595124
Viscera_weight	0.180594	0.1710	0.012015	0.591852	0.084012
Shell_weight	0.238831	0.2340	0.019377	0.620927	0.531926
Rings	9.933684	9.0000	10.395266	1.114102	2.330687

```
Sex               0
Length            0
Diameter          0
Height            0
Whole_weight      0
Sucked_weight     0
Viscera_weight    0
Shell_weight      0
Rings             0
dtype: int64
```

Is there any missing data? Answer to this question can given by observing the output of isna(), we can see that for for all the columns we have got value 0, which means that there are no missing values for any of the columns in the abalone dataset.

```
there are no missing data
```

```
count     4177
unique       3
top          M
freq      1528
Name: Sex, dtype: object
Total Unique Sex:  ['M' 'F' 'I']
```

Out[227]:
```
M    1528
I    1342
F    1307
Name: Sex, dtype: int64
```

Finding correlations between 2 variables/columns in abalone dataset

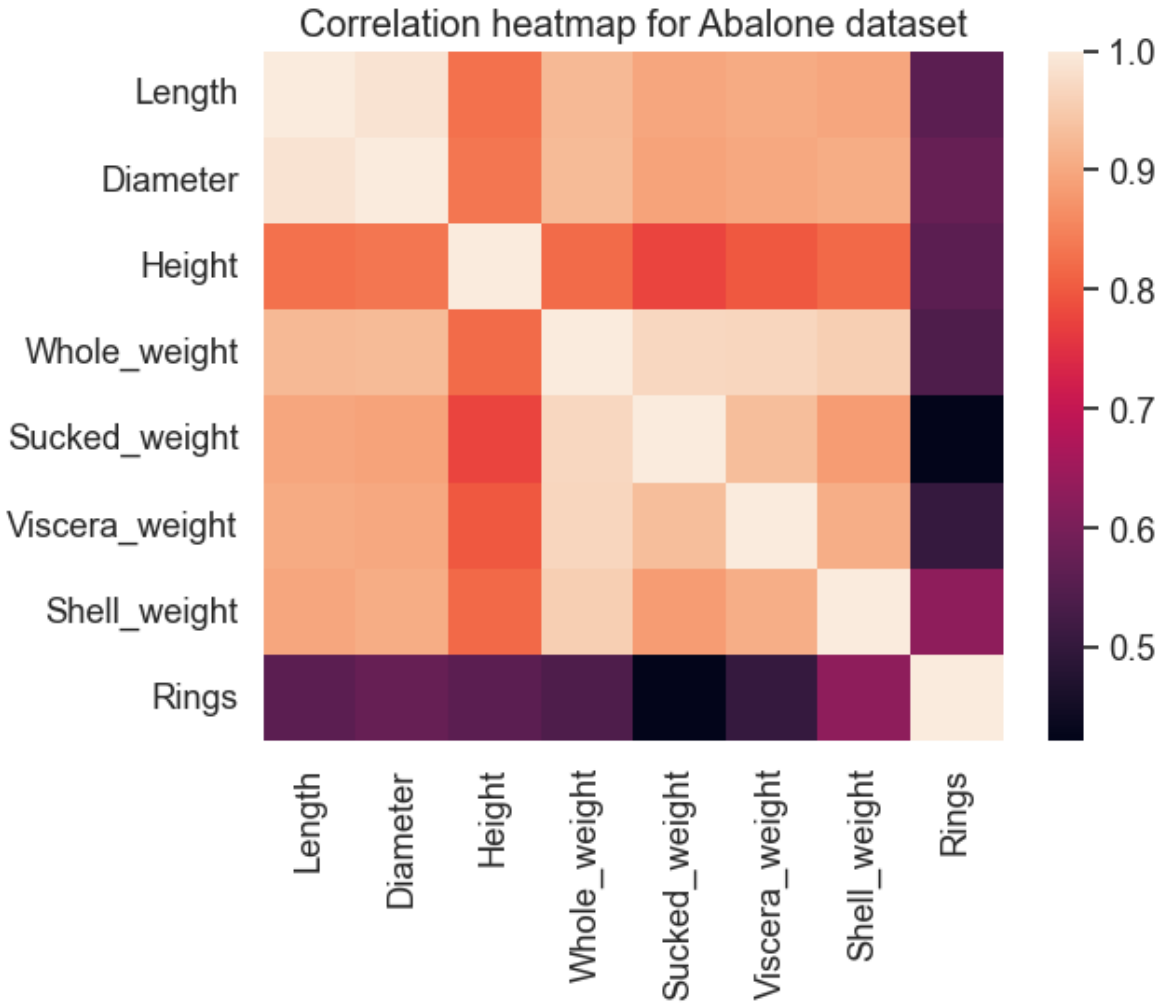
```
                  Length  Diameter    Height  Whole_weight  Sucked_weight  \
Length          1.000000  0.986812  0.827554      0.925261       0.897914
Diameter        0.986812  1.000000  0.833684      0.925452       0.893162
Height          0.827554  0.833684  1.000000      0.819221       0.774972
Whole_weight    0.925261  0.925452  0.819221      1.000000       0.969405
Sucked_weight   0.897914  0.893162  0.774972      0.969405       1.000000
Viscera_weight  0.903018  0.899724  0.798319      0.966375       0.931961
Shell_weight    0.897706  0.905330  0.817338      0.955355       0.882617
Rings           0.556720  0.574660  0.557467      0.540390       0.420884

                Viscera_weight  Shell_weight     Rings
Length                0.903018      0.897706  0.556720
Diameter              0.899724      0.905330  0.574660
Height                0.798319      0.817338  0.557467
Whole_weight          0.966375      0.955355  0.540390
Sucked_weight         0.931961      0.882617  0.420884
Viscera_weight        1.000000      0.907656  0.503819
Shell_weight          0.907656      1.000000  0.627574
Rings                 0.503819      0.627574  1.000000
```

finding the correlation w.r.t to rings

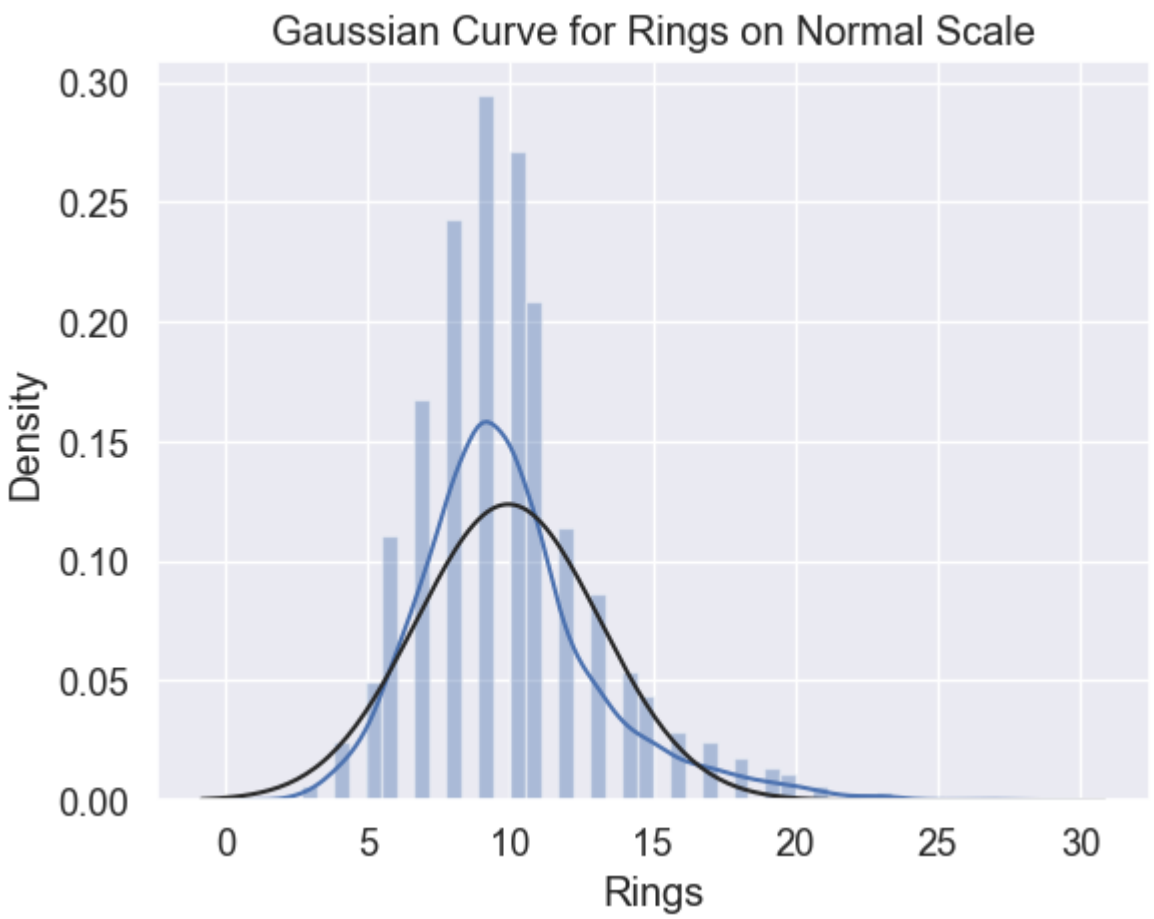
```
Length            0.556720
Diameter          0.574660
Height            0.557467
Whole_weight      0.540390
Sucked_weight     0.420884
Viscera_weight    0.503819
Shell_weight      0.627574
Rings             1.000000
Name: Rings, dtype: float64
```

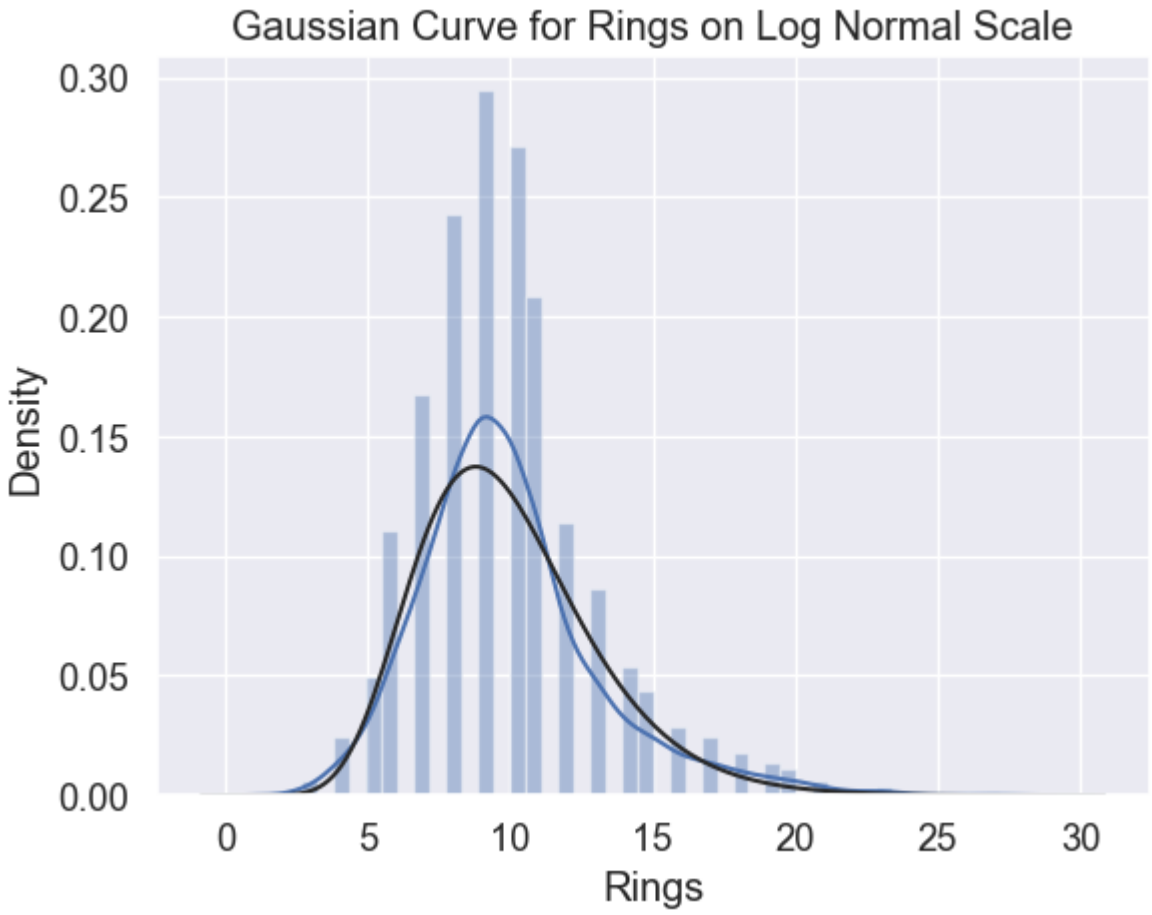
from the above block, we can see that only shell weight is somewhat correlated with rings columns. apart from this all other columns/parameters are not that much correlated with the rings. this would cause problem in the ring classification. 1 represents that the variables are highly correlated and 0 represents that variableas are not correlated.

Out[230]: <AxesSubplot:title={'center':'Correlation heatmap for Abalone dataset'}>

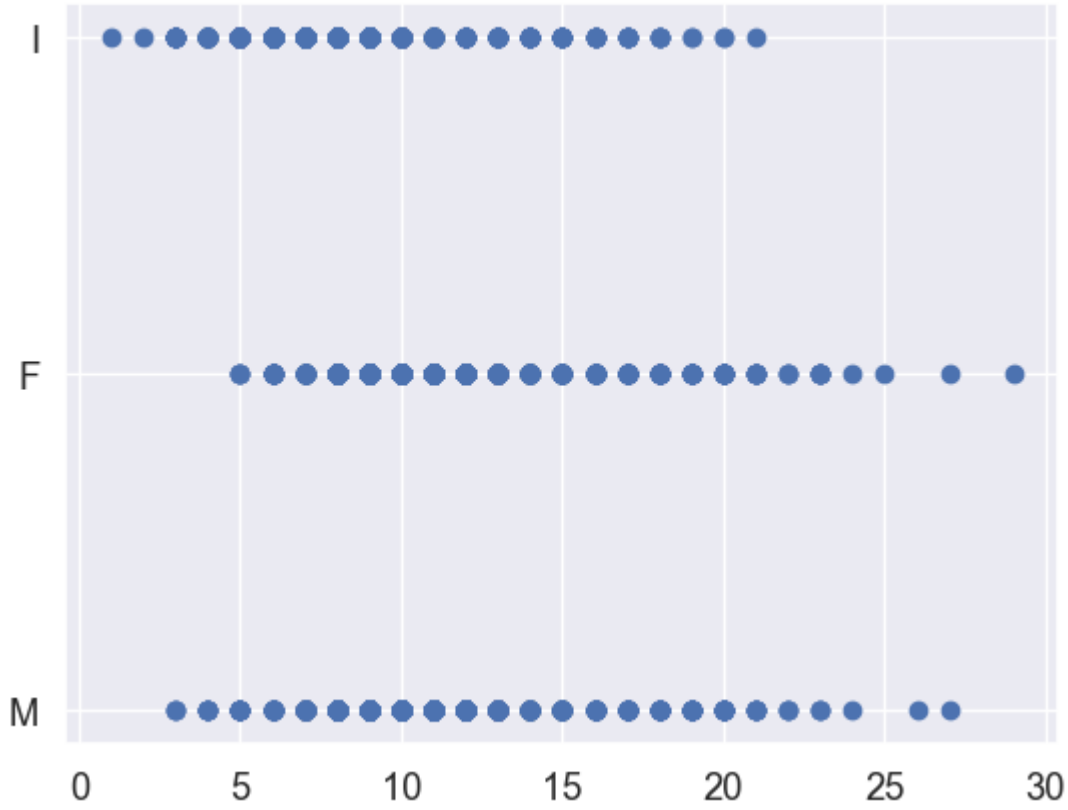


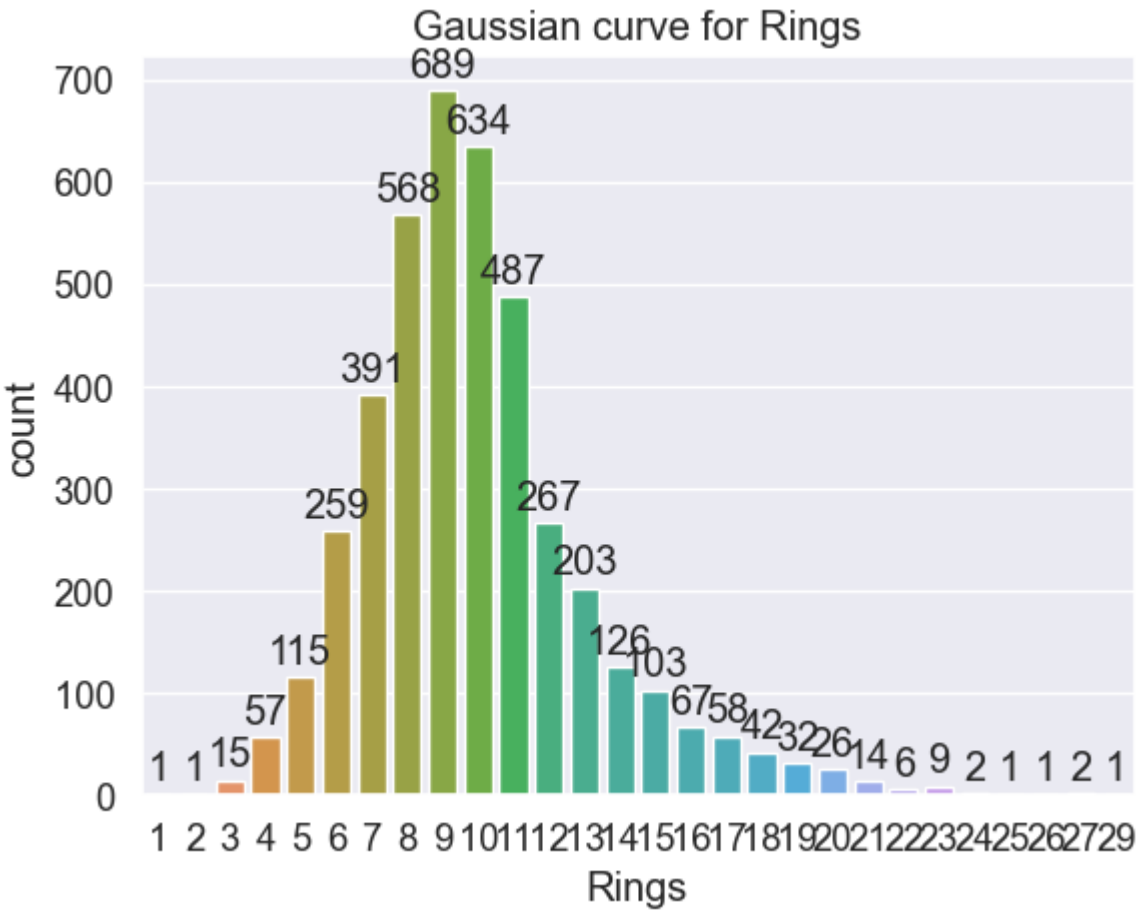
Out[231]: <AxesSubplot:title={'center':'Gaussian Curve for Rings on Log Normal Scale'}, xlabel='Rings', ylabel='Density'>

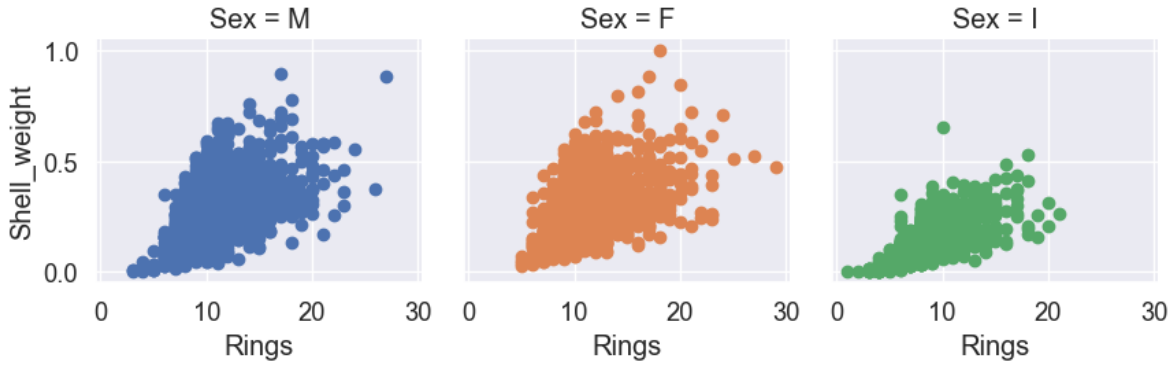




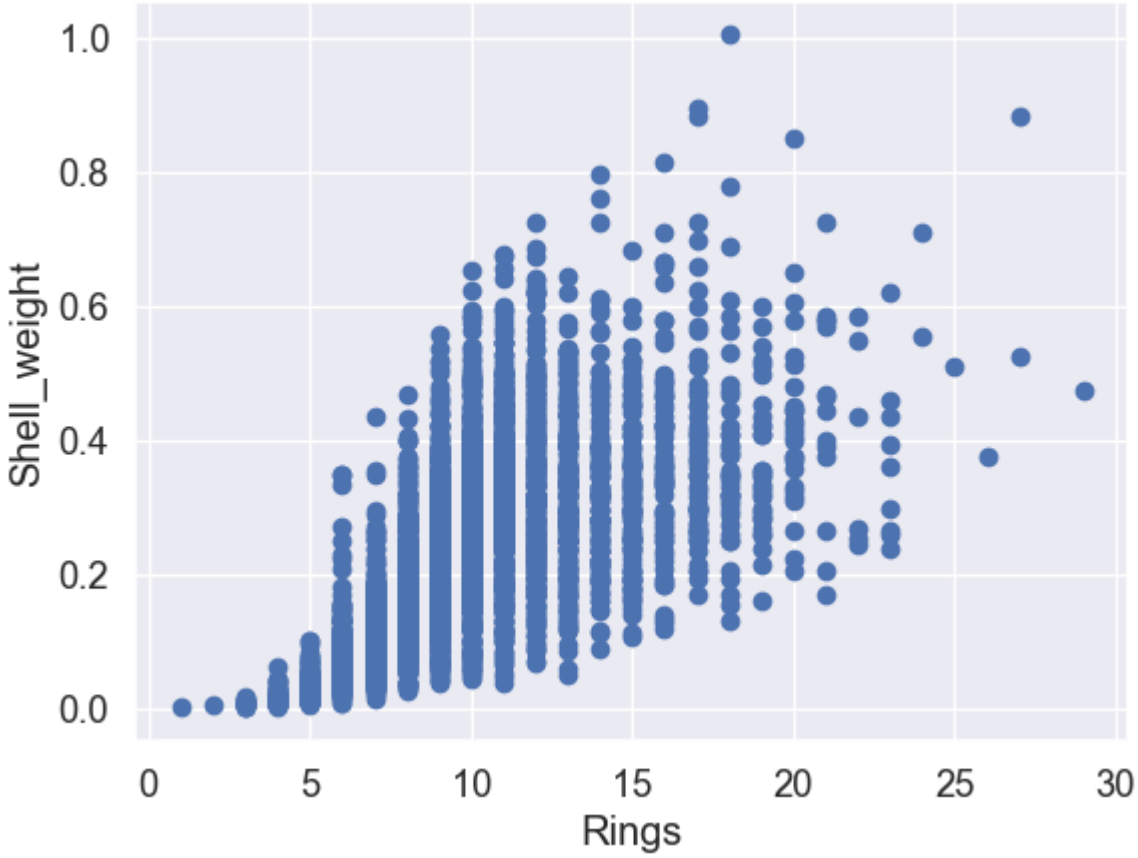
Out[232]: <matplotlib.collections.PathCollection at 0x1b253566310>



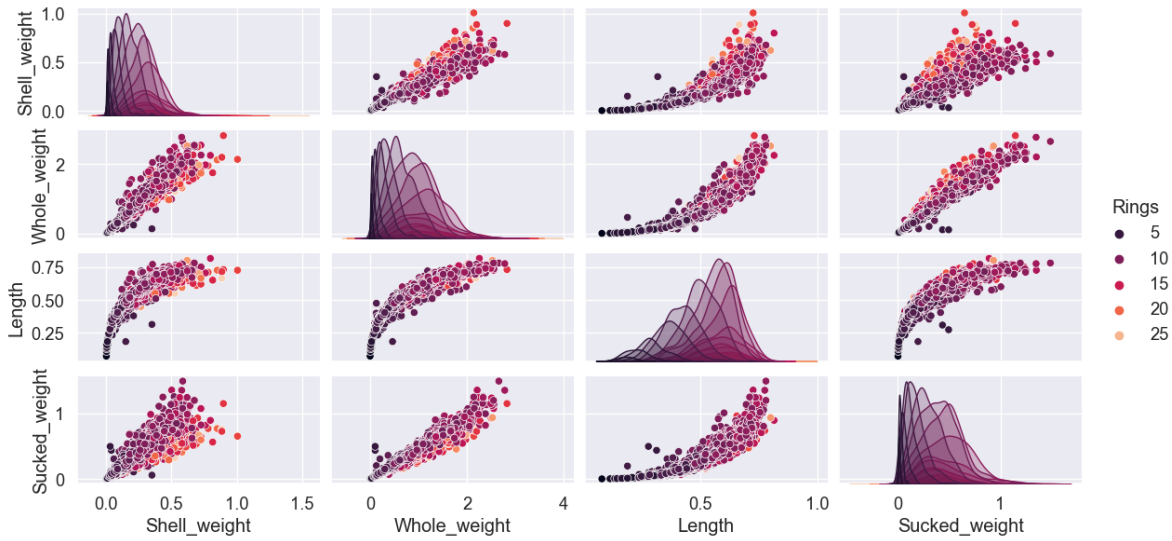

Gaussian curve for Rings
count
Rings
0
100
200
300
400
500
600
700
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 29
1 1 15 57 115 259 391 568 689 634 487 267 203 126 103 67 58 42 32 26 14 6 9 2 1 1 2 1


Sex = M
Sex = F
Sex = I
Shell_weight
0.0
0.5
1.0
0
10
20
30
Rings

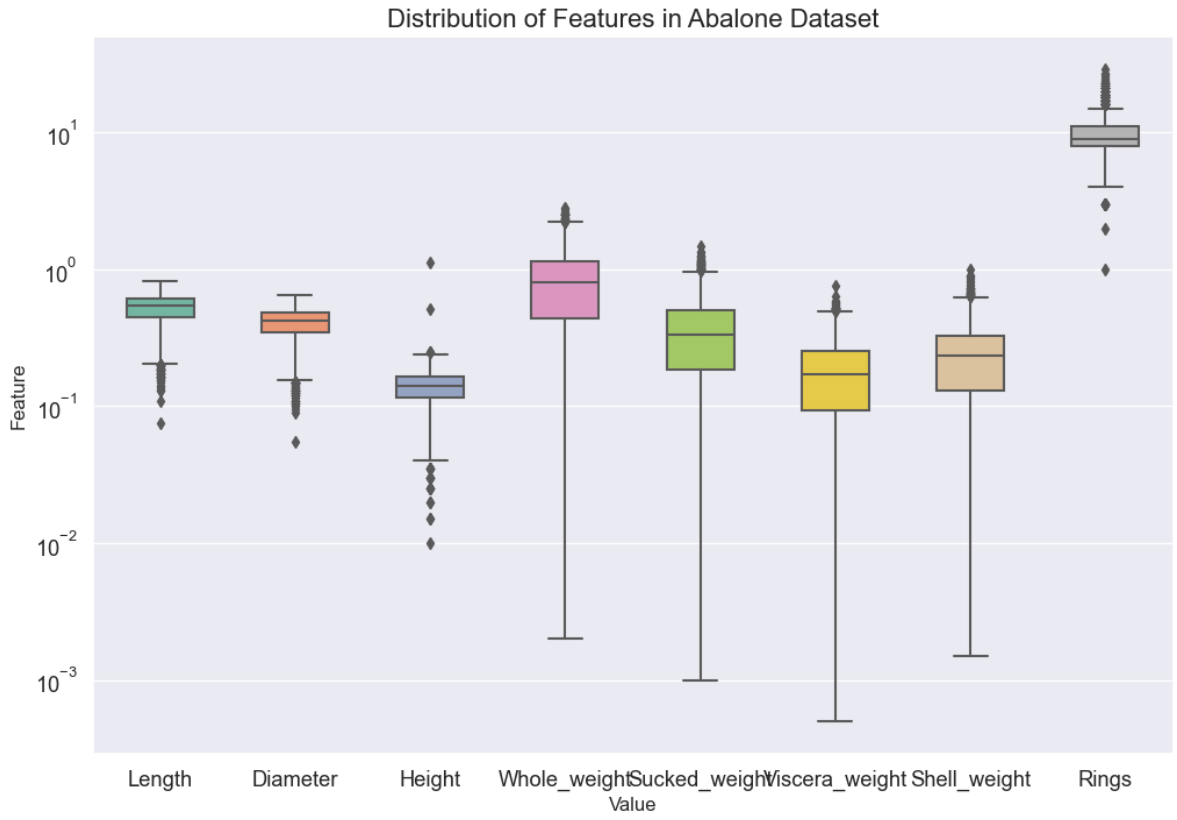
Out[235]: Text(0, 0.5, 'Shell_weight')



Out[236]: <seaborn.axisgrid.PairGrid at 0x1b2554e2640>



Out[237]: Text(0, 0.5, 'Feature')



Starting Normalization from here

	Sex	Length	Diameter	Height	Whole_weight	Sucked_weight	Viscera_weight	Shell_weight	Rings
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7

'Training Data Original'

	Length	Diameter	Height	Whole_weight	Sucked_weight	Viscera_weight	Shell_weight
1794	0.575	0.450	0.130	0.8145	0.4030	0.1715	0.2130
1466	0.515	0.425	0.145	0.9365	0.4970	0.1810	0.2185
2275	0.655	0.525	0.185	1.2590	0.4870	0.2215	0.4450
3929	0.650	0.515	0.215	1.4980	0.5640	0.3230	0.4250
1955	0.645	0.510	0.180	1.6195	0.7815	0.3220	0.4675

	Length	Diameter	Height	Whole_weight	Sucked_weight	Viscera_weight	Sl
count	3341.000000	3341.000000	3341.000000	3341.000000	3341.000000	3341.000000	3
mean	0.523590	0.407685	0.139397	0.829025	0.360037	0.180496	
std	0.120856	0.099806	0.042632	0.493738	0.223379	0.110090	
min	0.075000	0.055000	0.000000	0.002000	0.001000	0.000500	
25%	0.450000	0.350000	0.115000	0.438500	0.184500	0.092500	
50%	0.545000	0.425000	0.140000	0.797000	0.336000	0.170500	
75%	0.615000	0.480000	0.165000	1.153000	0.505500	0.253000	
max	0.815000	0.650000	1.130000	2.825500	1.488000	0.760000	

'Testing Data Original'

	Length	Diameter	Height	Whole_weight	Sucked_weight	Viscera_weight	Shell_weight
341	0.62	0.510	0.205	1.3475	0.4775	0.2565	0.480
3413	0.49	0.395	0.120	0.6740	0.3325	0.1235	0.185
1088	0.45	0.340	0.120	0.4925	0.2410	0.1075	0.120
98	0.47	0.370	0.130	0.5225	0.2010	0.1330	0.165
3661	0.55	0.415	0.150	0.7915	0.3535	0.1760	0.236

	Length	Diameter	Height	Whole_weight	Sucked_weight	Viscera_weight	Shell_
count	836.000000	836.000000	836.000000	836.000000	836.000000	836.000000	836
mean	0.525598	0.408666	0.139994	0.827610	0.356690	0.180982	0
std	0.117049	0.096998	0.038462	0.477058	0.216322	0.107758	0
min	0.160000	0.110000	0.015000	0.014500	0.005500	0.002500	0
25%	0.453750	0.350000	0.115000	0.451500	0.190500	0.095375	0
50%	0.545000	0.425000	0.140000	0.806000	0.336750	0.171000	0
75%	0.615000	0.481250	0.165000	1.157125	0.492625	0.252875	0
max	0.800000	0.630000	0.240000	2.526000	1.351000	0.590000	0

'Training Data Z Normalized'

	Length	Diameter	Height	Whole_weight	Sucked_weight	Viscera_weight	Shell_weigh
1794	0.424788	0.424464	-0.227545	-0.029046	0.196599	-0.082970	-0.18558
1466	-0.074885	0.172519	0.131117	0.219766	0.620144	0.003708	-0.14607
2275	1.091018	1.180300	1.087551	0.877486	0.575086	0.373230	1.48124
3929	1.049379	1.079522	1.804876	1.364912	0.922032	1.299315	1.33755
1955	1.007740	1.029133	0.967997	1.612704	1.902043	1.290191	1.64290

	Length	Diameter	Height	Whole_weight	Sucked_weight	Viscera_weight	Sl
count	3341.000000	3341.000000	3341.000000	3341.000000	3341.000000	3341.000000	3
mean	-0.003347	-0.001980	-0.002858	0.000578	0.003019	-0.000887	
std	1.006478	1.005828	1.019371	1.006950	1.006500	1.004457	
min	-3.739154	-3.556267	-3.335953	-1.686092	-1.614731	-1.643173	
25%	-0.616198	-0.583316	-0.586208	-0.795876	-0.787917	-0.803766	
50%	0.174951	0.172519	0.011563	-0.064736	-0.105289	-0.092094	
75%	0.757903	0.726798	0.609334	0.661305	0.658443	0.660635	
max	2.423480	2.440025	23.683287	4.072271	5.085388	5.286500	

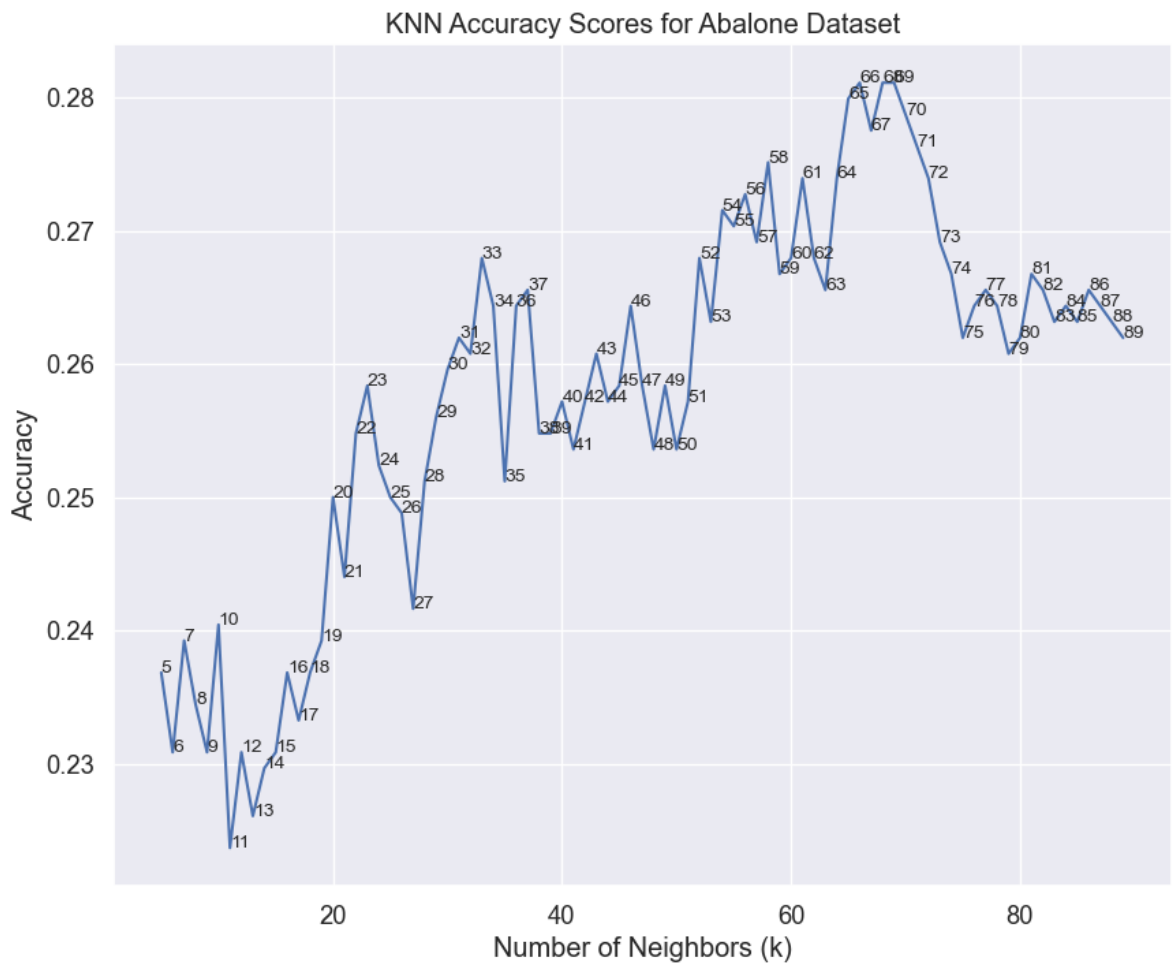
'Testing Data Z Normalized'

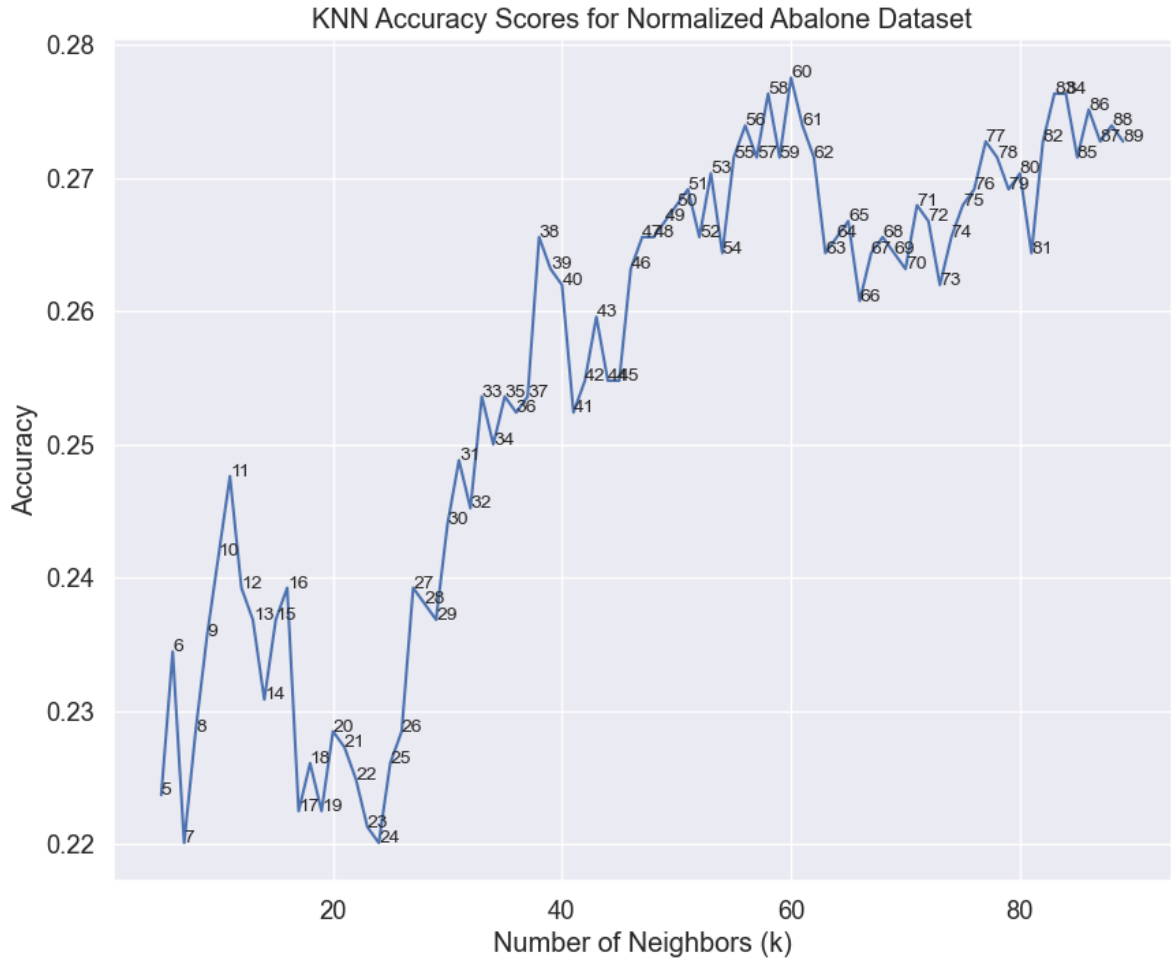
	Length	Diameter	Height	Whole_weight	Sucked_weight	Viscera_weight	Shell_weigl
341	0.799543	1.029133	1.565767	1.057976	0.532281	0.692569	1.73271
3413	-0.283082	-0.129815	-0.466653	-0.315588	-0.121059	-0.520922	-0.38675
1088	-0.616198	-0.684094	-0.466653	-0.685746	-0.533340	-0.666906	-0.85375
98	-0.449640	-0.381760	-0.227545	-0.624563	-0.713572	-0.434244	-0.53044
3661	0.216591	0.071741	0.250672	-0.075953	-0.026438	-0.041912	-0.02033

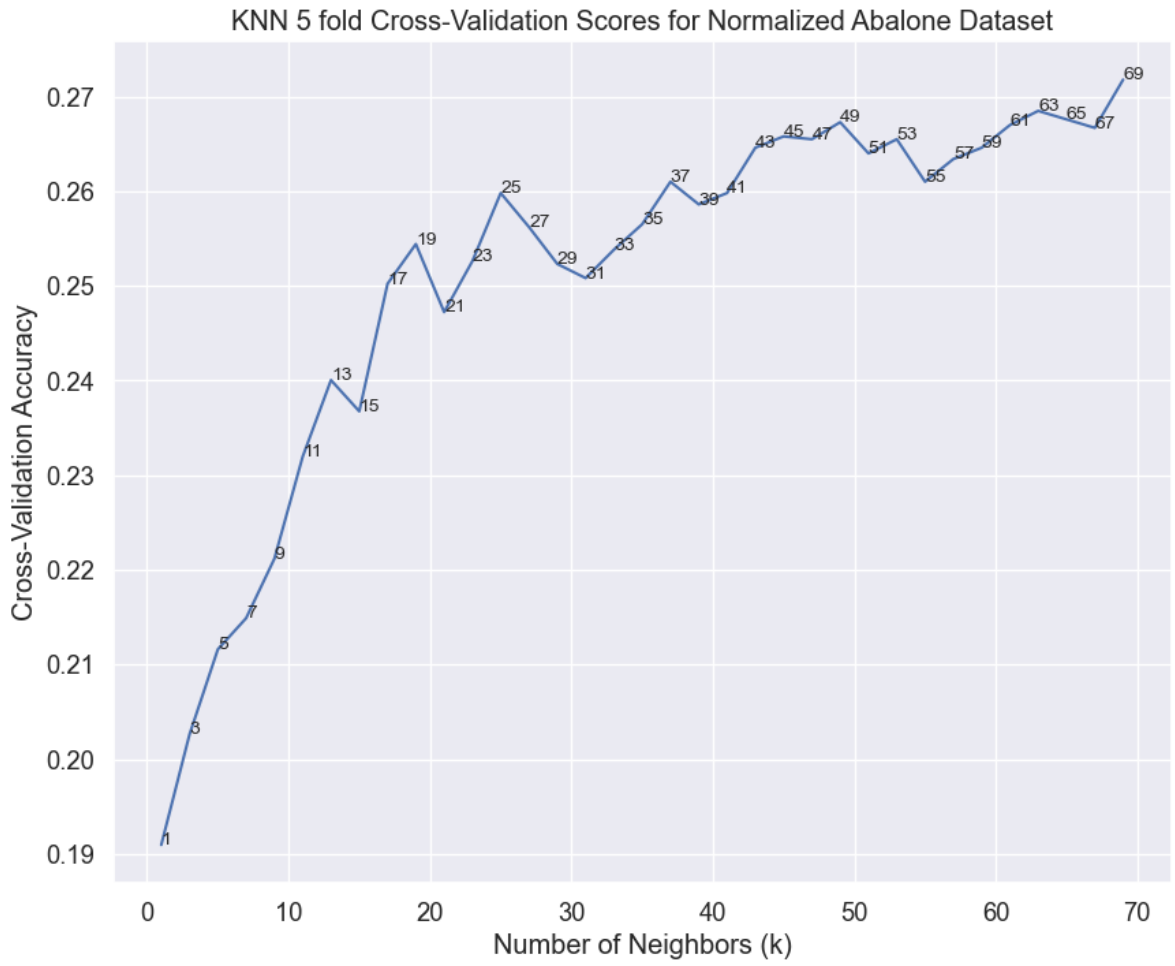
	Length	Diameter	Height	Whole_weight	Sucked_weight	Viscera_weight	Shell_
count	836.000000	836.000000	836.000000	836.000000	836.000000	836.000000	836
mean	0.013374	0.007911	0.011420	-0.002309	-0.012063	0.003544	0
std	0.974773	0.977526	0.919654	0.972932	0.974702	0.983182	0
min	-3.031284	-3.001988	-2.977291	-1.660599	-1.594455	-1.624925	-1
25%	-0.584968	-0.583316	-0.586208	-0.769363	-0.760882	-0.777534	-0
50%	0.174951	0.172519	0.011563	-0.046381	-0.101910	-0.087532	-0
75%	0.757903	0.739396	0.609334	0.669718	0.600431	0.659495	0
max	2.298562	2.238469	2.402646	3.461458	4.468094	3.735421	4

Out[241]: 0.20334928229665072

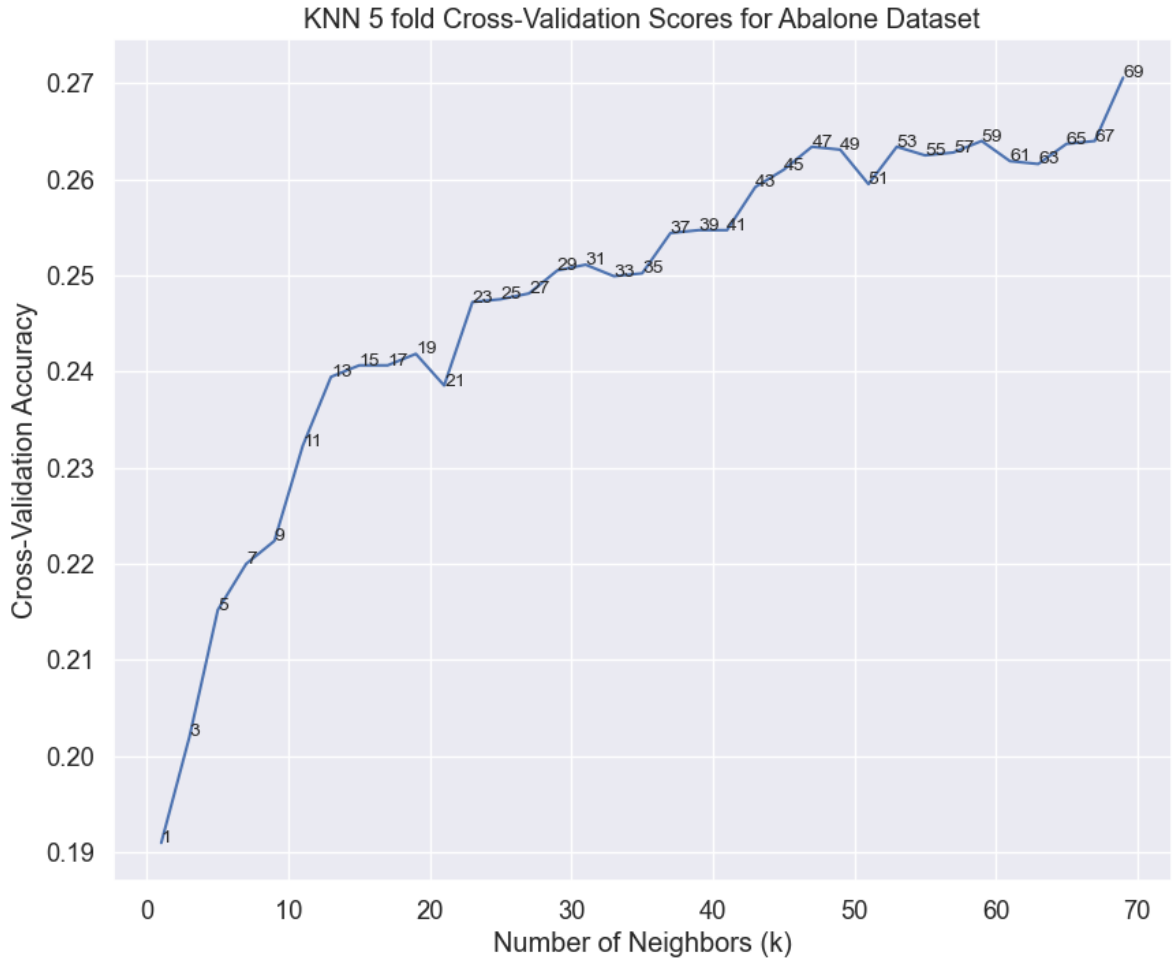
Out[242]: 0.215311004784689

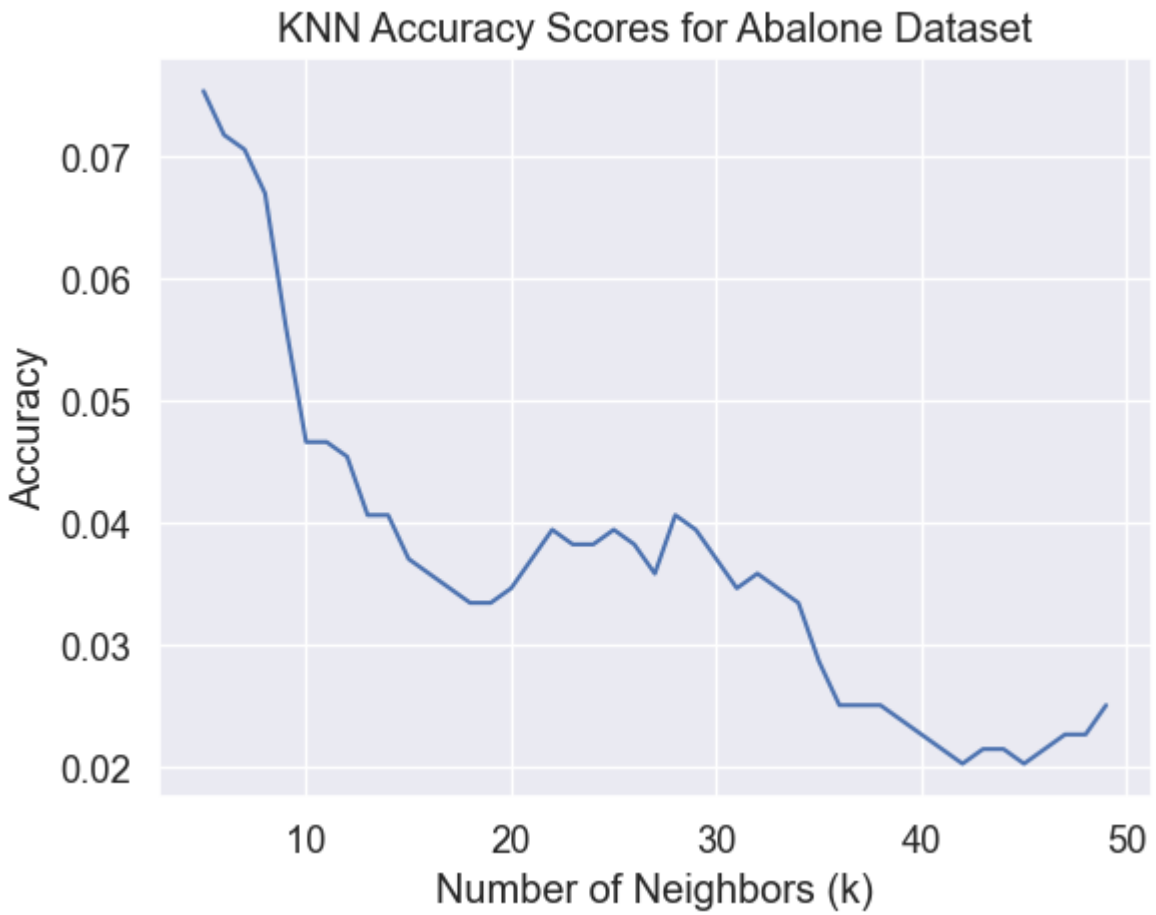


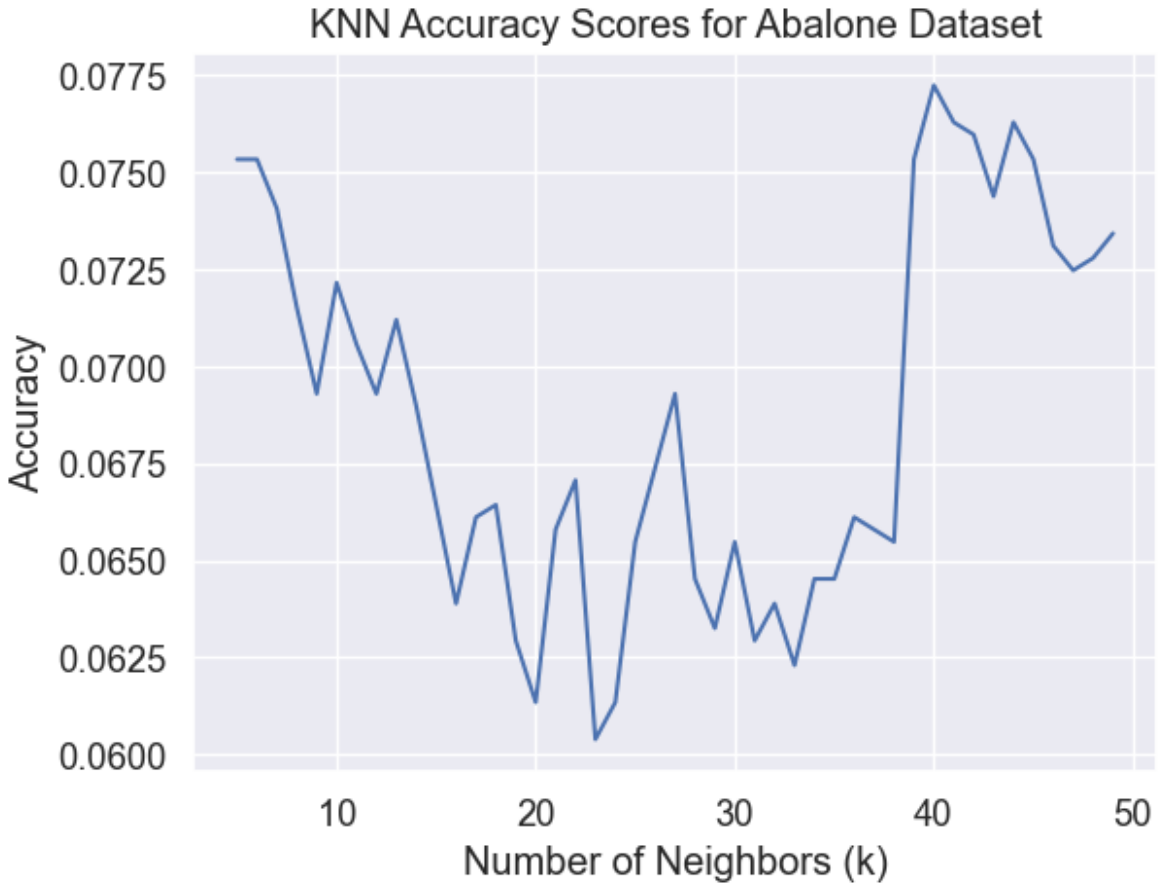

KNN Accuracy Scores for Normalized Abalone Dataset
Accuracy
Number of Neighbors (k)
0.28
0.27
0.26
0.25
0.24
0.23
0.22
20
40
60
80


KNN 5 fold Cross-Validation Scores for Normalized Abalone Dataset
Cross-Validation Accuracy
Number of Neighbors (k)
0.19
0.20
0.21
0.22
0.23
0.24
0.25
0.26
0.27
0
10
20
30
40
50
60
70

```
[0.19096604996285454, 0.20203807631374024, 0.2152068061187043, 0.219991854855
31178, 0.22238706443615014, 0.2322668564216858, 0.23945248516420076, 0.240649
64241919745, 0.2406478522775078, 0.2418432193908148, 0.2385533865005415, 0.24
723467862481313, 0.24753139460988338, 0.24813064454051537, 0.2505231689088191
5, 0.25112197130402875, 0.249924814049032, 0.25022421524663674, 0.25441314680
05693, 0.2547147856752862, 0.2547147856752862, 0.25920267089140103, 0.2609946
027228055, 0.2633902598390663, 0.2630922012477287, 0.25950117701816094, 0.263
3916024453335, 0.2624969791358986, 0.2627945901918137, 0.2639917474468104, 0.
2618968341344218, 0.2615983280076618, 0.26369324132005045, 0.2639926425176552
5, 0.2705758885815812]
```




KNN Accuracy Scores for Abalone Dataset
Accuracy
0.07
0.06
0.05
0.04
0.03
0.02
10
20
30
40
50
Number of Neighbors (k)


KNN Accuracy Scores for Abalone Dataset
Accuracy
0.0775
0.0750
0.0725
0.0700
0.0675
0.0650
0.0625
0.0600
10
20
30
40
50
Number of Neighbors (k)

Assignment 2 Question 1 implementation starts here

Question 1

Using Abalone Dataset

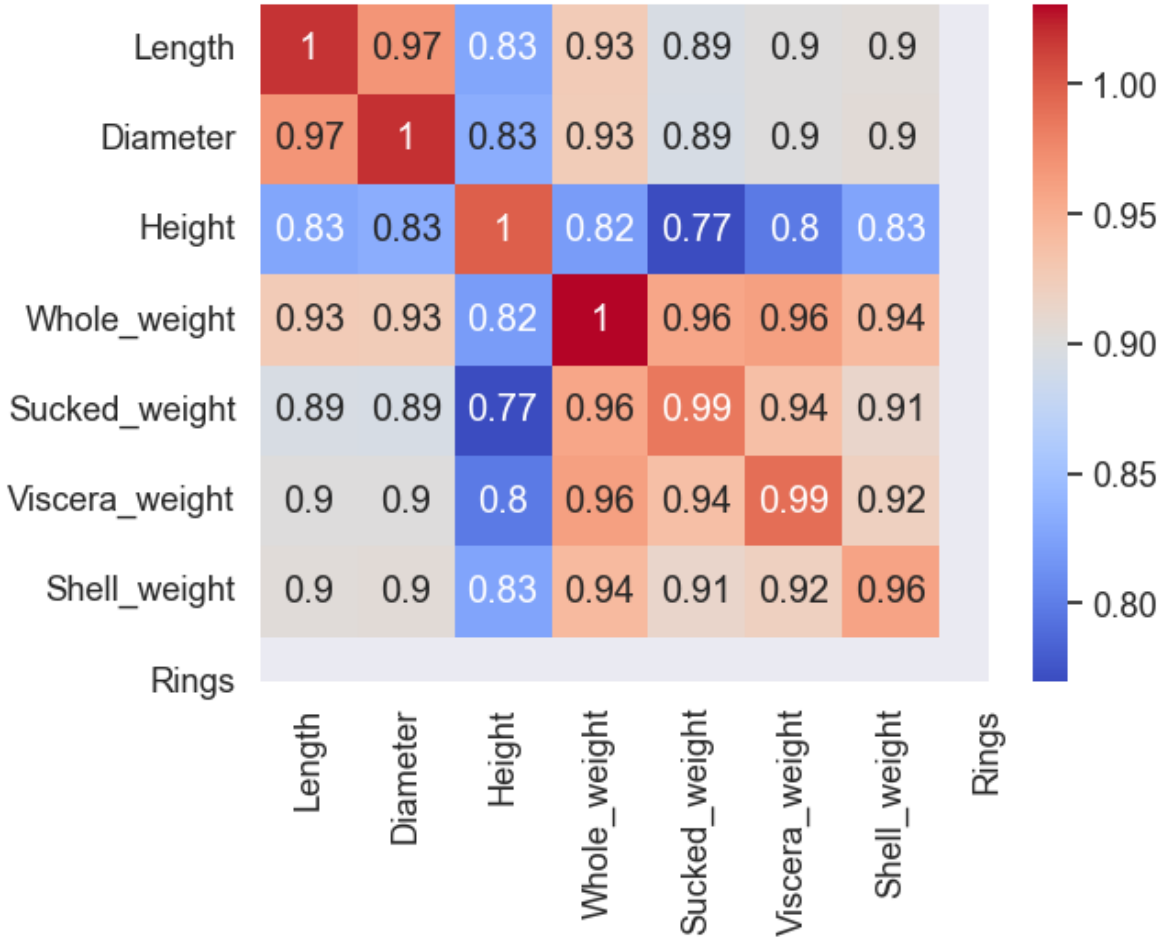
```
[15  7  9 ...  9 10 12]
Covariance Matrix Calculated using numpy:

1.0002  0.9870  0.8278  0.9255  0.8981  0.9032  0.8979
0.9870  1.0002  0.8339  0.9257  0.8934  0.8999  0.9055
0.8278  0.8339  1.0002  0.8194  0.7752  0.7985  0.8175
0.9255  0.9257  0.8194  1.0002  0.9696  0.9666  0.9556
0.8981  0.8934  0.7752  0.9696  1.0002  0.9322  0.8828
0.9032  0.8999  0.7985  0.9666  0.9322  1.0002  0.9079
0.8979  0.9055  0.8175  0.9556  0.8828  0.9079  1.0002
```

In the above we block, we calculated the covariance matrix in order to find the similarity between two feature variables.instead of using Direct PCA, we calculated it using numpy array to study the data. We can see that the feature points are quite correlated with each other.

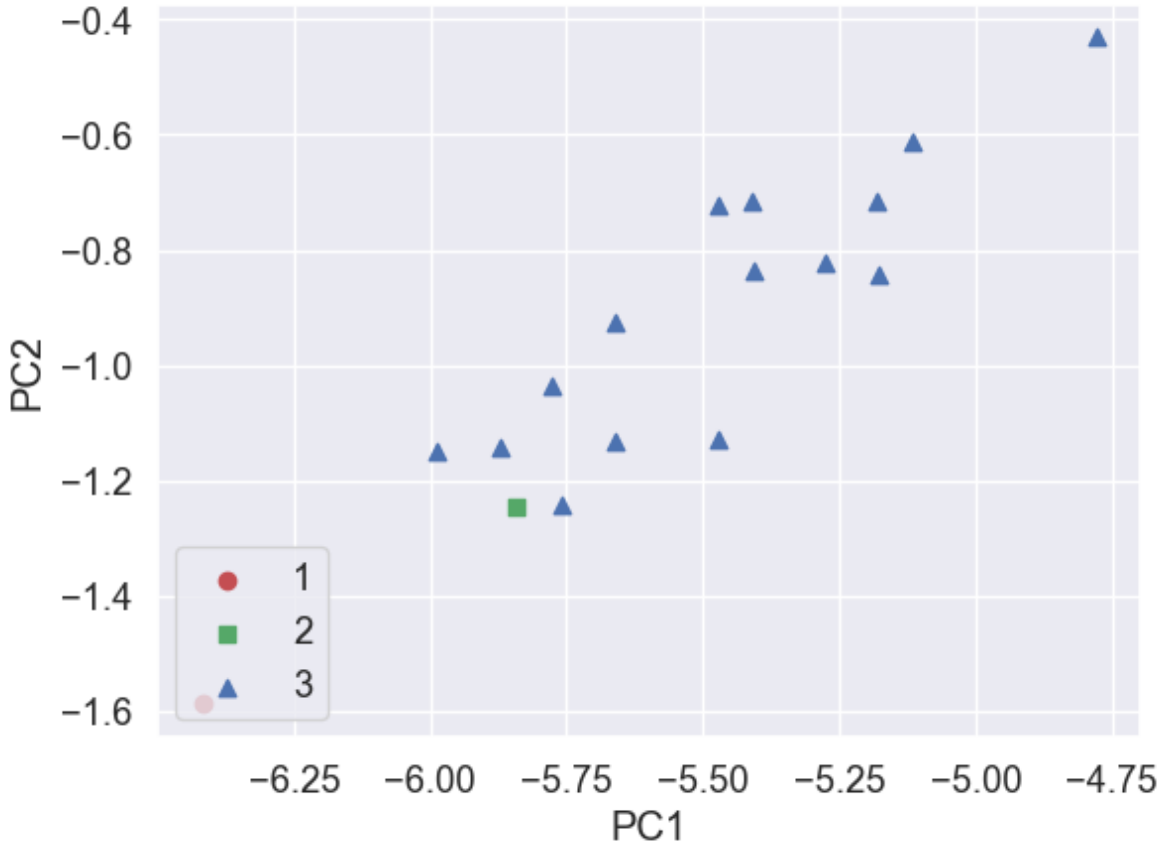
1. The diagonal elements are the variances of each variable, which represent the spread or variability of each variable around its mean.
2. The off-diagonal elements are the covariances between each pair of variables, which represent the degree to which the variables vary together.
3. The covariance matrix shows that the length and diameter of abalone have a high positive covariance, indicating that they tend to vary together. Similarly, the weight measurements have a positive covariance with each other.
4. The covariance matrix also shows that the height of abalone has a low covariance with the other variables, indicating that it is less related to the other variables.

Using PCA as a preprocessing step for abalone dataset

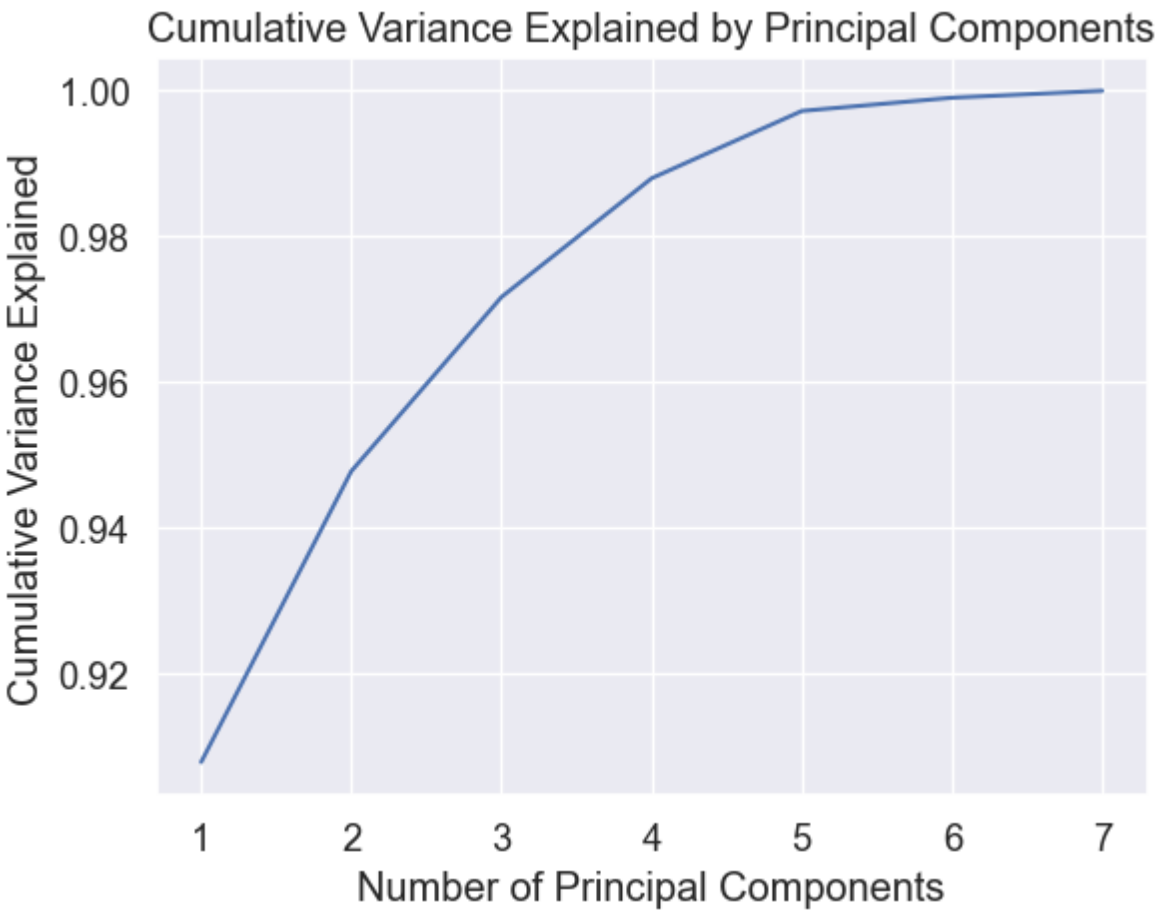
```
Variance explained by each principal component:
 [0.9078731478516083, 0.03991890899342265, 0.023906381975154992, 0.016295977883821613, 0.009236274060776192, 0.0018182993981407179, 0.0009510098370754434]
Cumulative variance explained by each principal component:
 [0.90787315 0.94779206 0.97169844 0.98799442 0.99723069 0.99904899
 1.        ]
Number of principal components needed to explain 95% of the variance: 3
```

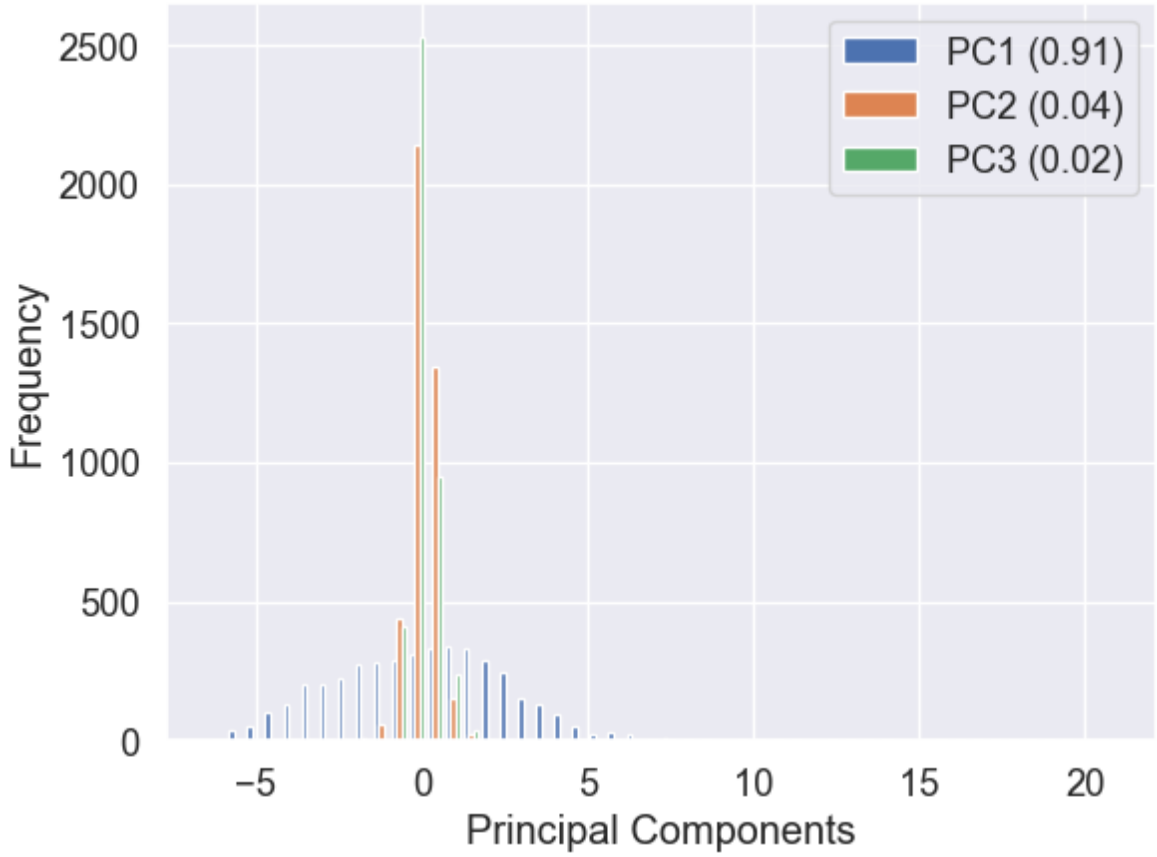


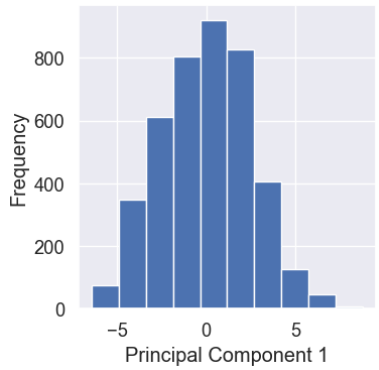
```
Covariance matrix of the principal components:
1.0177  0.9687  0.8271  0.9262  0.8949  0.9011  0.9041
0.9687  1.0189  0.8335  0.9257  0.8937  0.9005  0.9046
0.8271  0.8335  0.9985  0.8199  0.7696  0.7973  0.8264
0.9262  0.9257  0.8199  1.0310  0.9568  0.9610  0.9412
0.8949  0.8937  0.7696  0.9568  0.9858  0.9376  0.9129
0.9011  0.9005  0.7973  0.9610  0.9376  0.9909  0.9203
0.9041  0.9046  0.8264  0.9412  0.9129  0.9203  0.9590


Eigenvalues of the principal components:
6.3566
0.2795
0.1674


Eigenvectors of the principal components:
0.3833  0.3836  0.3481  0.3907  0.3782  0.3815  0.3789
0.0379  0.0653  0.8668  -0.2333 -0.3480 -0.2529 -0.0584
-0.5933 -0.5854 0.3149  0.2308  0.2316  0.2703  0.1621
```

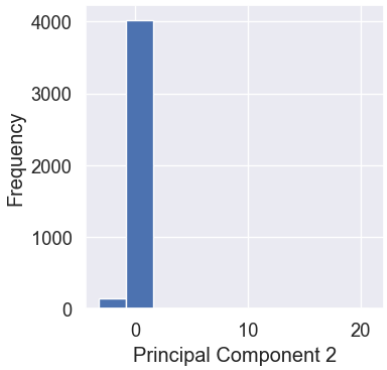


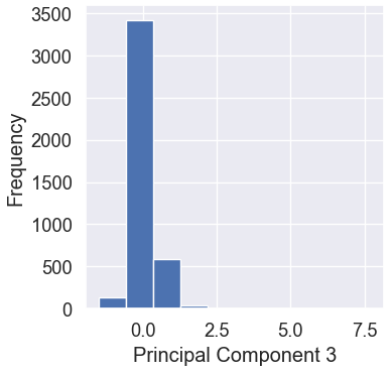
from the eigen values, we can see that the first principal components captures most variance.

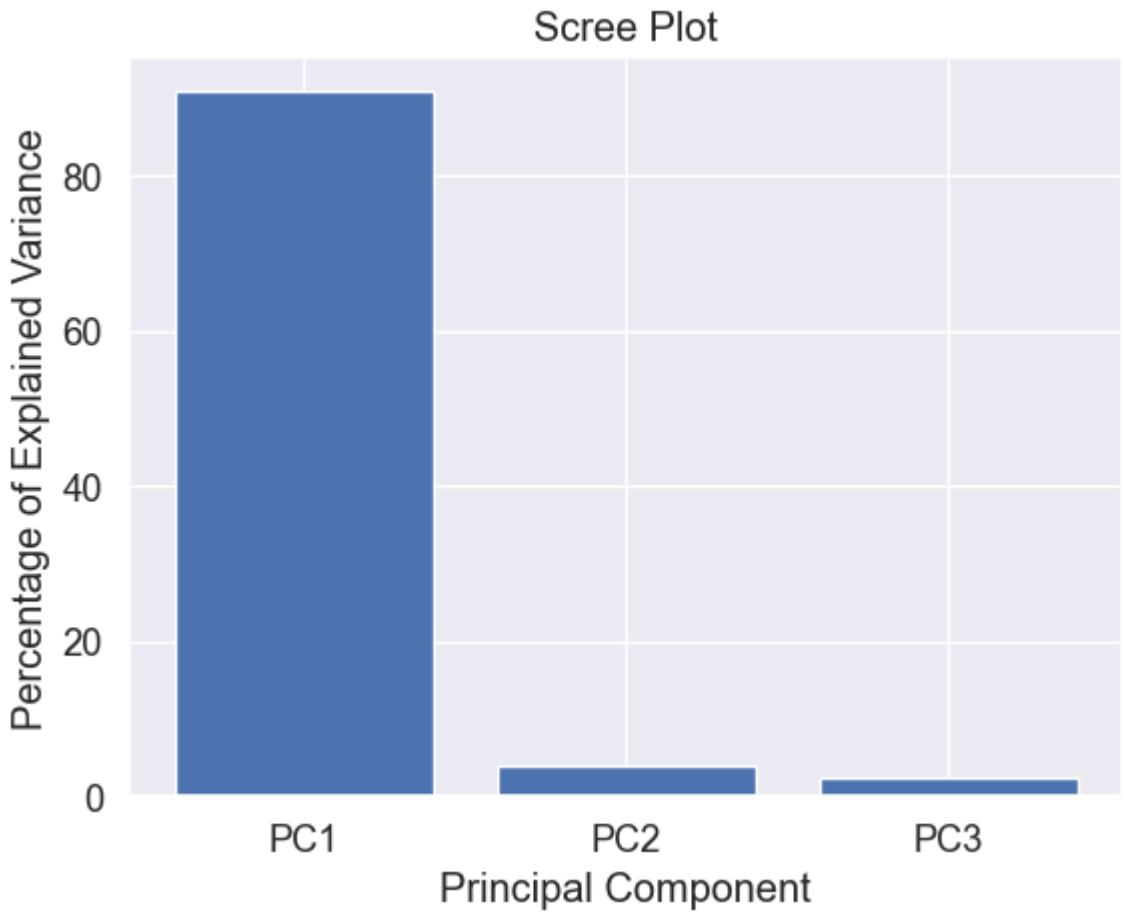

Cumulative Variance Explained by Principal Components
Cumulative Variance Explained
1.00
0.98
0.96
0.94
0.92
1
2
3
4
5
6
7
Number of Principal Components

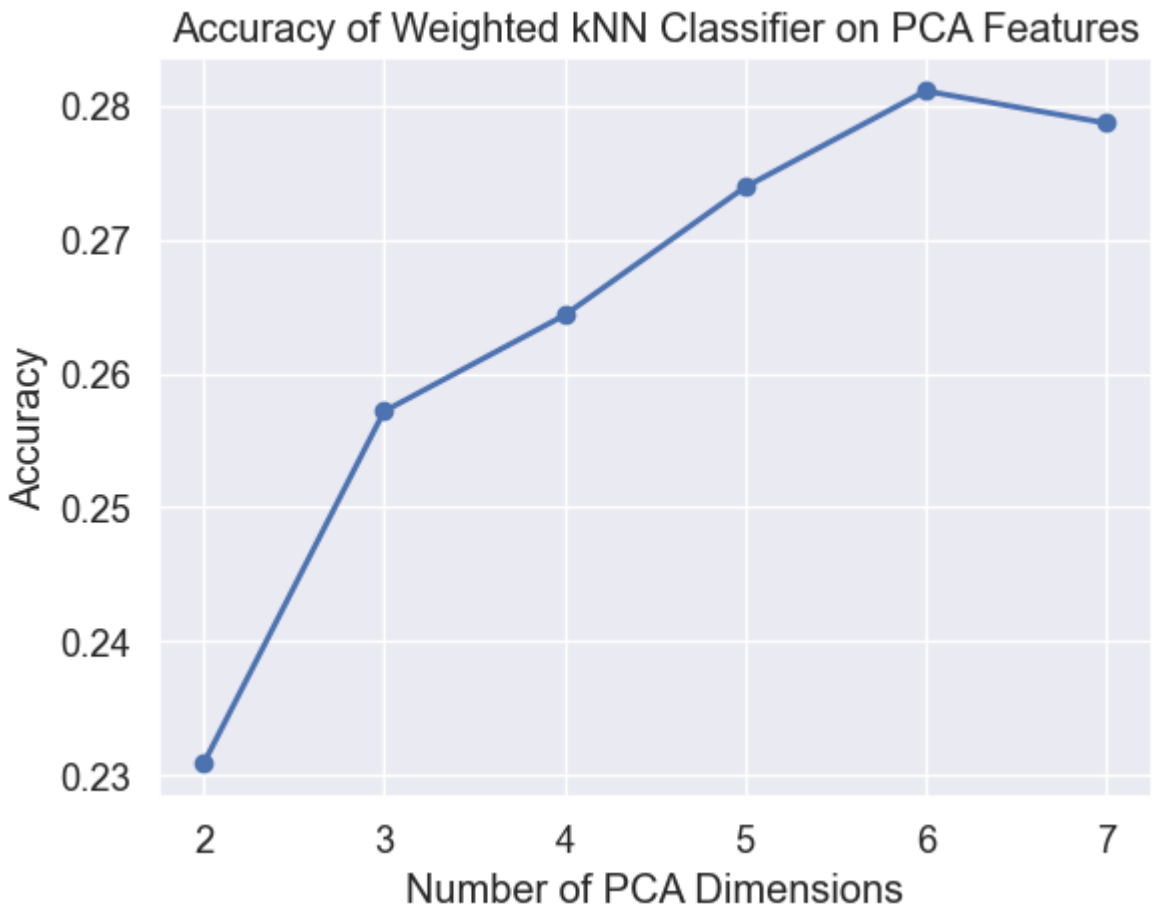

PC1 (0.91)
PC2 (0.04)
PC3 (0.02)
Frequency
2500
2000
1500
1000
500
0
−5
0
5
10
15
20
Principal Components


Frequency
800
600
400
200
0
−5
0
5
Principal Component 1


Frequency
4000
3000
2000
1000
0
0
10
20
Principal Component 2


Frequency
3500
3000
2500
2000
1500
1000
500
0
0.0
2.5
5.0
7.5
Principal Component 3

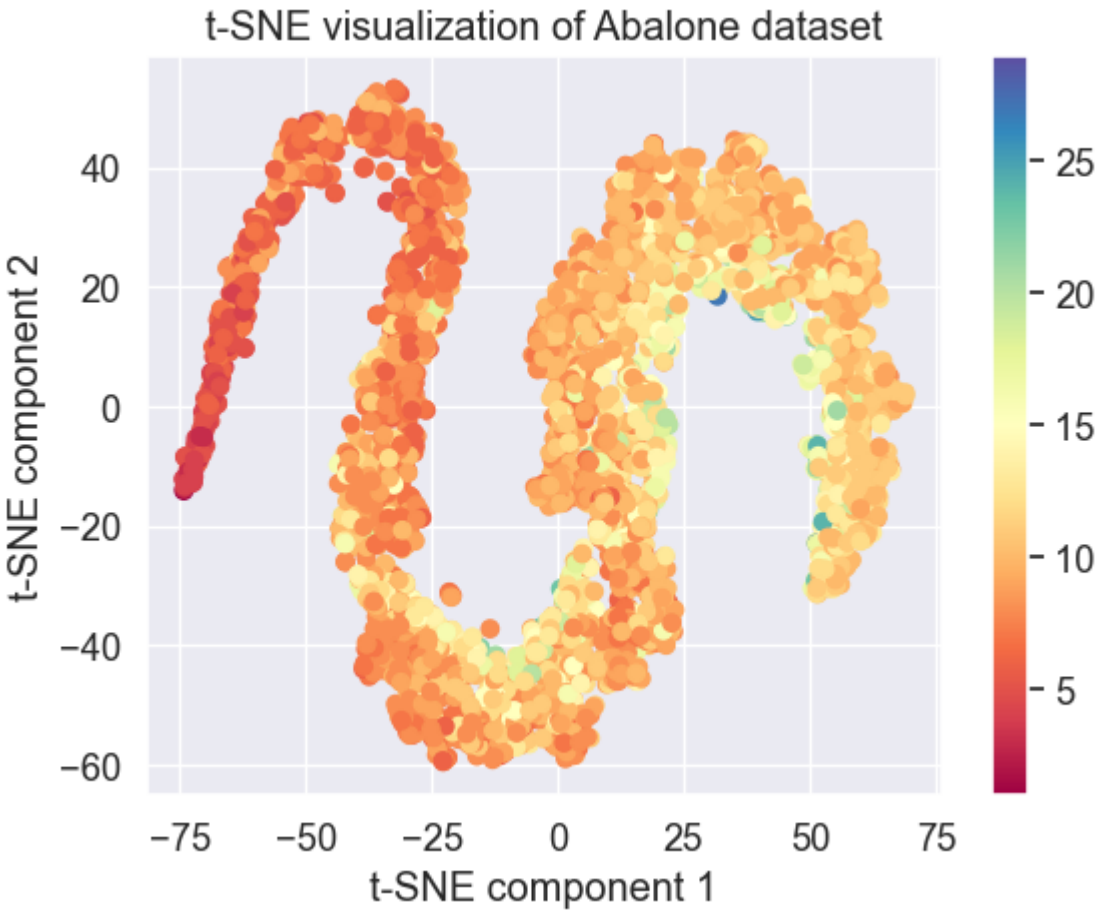

Scree Plot
Percentage of Explained Variance
80
60
40
20
0
PC1
PC2
PC3
Principal Component




```
Accuracy: 0.25717703349282295
```

Using LDA as a preprocessing step for abalone dataset

```
['old', 'young', 'middle-aged', 'middle-aged', 'young', ..., 'middle-aged',
'middle-aged', 'middle-aged', 'middle-aged', 'old']
Length: 4177
Categories (3, object): ['young' < 'middle-aged' < 'old']
Mean cv accuracy: 0.6692637147230204
Test Accuracy: 0.6961722488038278
CV Accuracy scores: [0.66367713 0.68263473 0.69011976 0.67215569 0.65568862]
CV Average accuracy: 0.6728551864880105
```



1. The t-SNE plot shows the abalone samples projected onto a two-dimensional space based on their similarity in the original high-dimensional space.
2. Each point in the plot represents an abalone sample, and the color of the point corresponds to the number of rings in the abalone (an indicator of age).
3. The t-SNE plot reveals that the abalone samples with similar numbers of rings tend to cluster together, indicating that age is an important factor in the variability of the data.
4. The plot also shows that the length and diameter measurements of the abalone are strongly correlated, as points that are close together in the plot tend to have similar values for these variables.
5. There is some overlap between the clusters corresponding to different numbers of rings, indicating that other variables in the dataset also contribute to the variability of the data.
6. Overall, the t-SNE visualization provides an intuitive way to explore the structure of the abalone dataset and can reveal interesting patterns and relationships between the variables.

Wine Dataset

Out[260]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alc
0	7.0	0.270	0.36	20.7	0.045	45.0	170.0	1.00100	3.00	0.45	
1	6.3	0.300	0.34	1.6	0.049	14.0	132.0	0.99400	3.30	0.49	
2	8.1	0.280	0.40	6.9	0.050	30.0	97.0	0.99510	3.26	0.44	
3	7.2	0.230	0.32	8.5	0.058	47.0	186.0	0.99560	3.19	0.40	
4	7.2	0.230	0.32	8.5	0.058	47.0	186.0	0.99560	3.19	0.40	
...	...	...	...	...	...	...	...	...	...	...	
6492	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	
6493	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	
6494	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	
6495	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	
6496	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	

6497 rows × 13 columns

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6497 entries, 0 to 6496
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         6497 non-null   float64
 1   volatile acidity      6497 non-null   float64
 2   citric acid           6497 non-null   float64
 3   residual sugar        6497 non-null   float64
 4   chlorides             6497 non-null   float64
 5   free sulfur dioxide   6497 non-null   float64
 6   total sulfur dioxide  6497 non-null   float64
 7   density               6497 non-null   float64
 8   pH                    6497 non-null   float64
 9   sulphates             6497 non-null   float64
 10  alcohol               6497 non-null   float64
 11  quality               6497 non-null   int64  
 12  colour                6497 non-null   int64  
dtypes: float64(11), int64(2)
memory usage: 660.0 KB
```

Out[262]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulf dioxi
count	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.0000
mean	7.215307	0.339666	0.318633	5.443235	0.056034	30.525319	115.7445
std	1.296434	0.164636	0.145318	4.757804	0.035034	17.749400	56.5218
min	3.800000	0.080000	0.000000	0.600000	0.009000	1.000000	6.0000
25%	6.400000	0.230000	0.250000	1.800000	0.038000	17.000000	77.0000
50%	7.000000	0.290000	0.310000	3.000000	0.047000	29.000000	118.0000
75%	7.700000	0.400000	0.390000	8.100000	0.065000	41.000000	156.0000
max	15.900000	1.580000	1.660000	65.800000	0.611000	289.000000	440.0000

There are no missing values, so now we can start with EDA. More samples of quality 5 or 6 have been observed in the dataset, which shows that it is not a balanced dataset. The standard deviation for most features vary over a range and hence, we require normalization of the features before applying PCA.

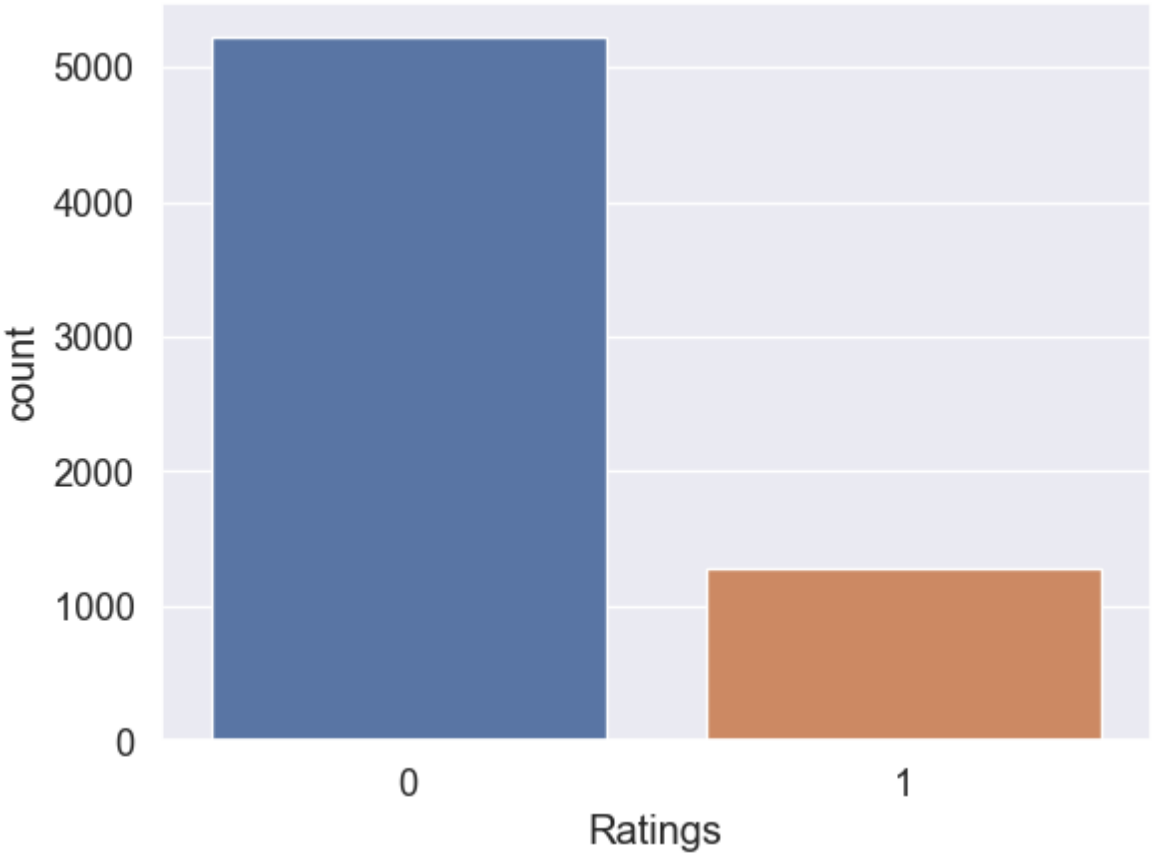
Out[263]: `(6497, 13)`

Most of the wines in this dataset has a quality score of 5 or 6. We will now add a feature called 'rating' depending on the quality score of each wine data point. If quality is <5, we assign them as 'Bad' (value of 0) and if quality is >=5, we assign it as 'Good' (value of 1).

Out[264]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alc
0	7.0	0.270	0.36	20.7	0.045	45.0	170.0	1.00100	3.00	0.45	
1	6.3	0.300	0.34	1.6	0.049	14.0	132.0	0.99400	3.30	0.49	
2	8.1	0.280	0.40	6.9	0.050	30.0	97.0	0.99510	3.26	0.44	
3	7.2	0.230	0.32	8.5	0.058	47.0	186.0	0.99560	3.19	0.40	
4	7.2	0.230	0.32	8.5	0.058	47.0	186.0	0.99560	3.19	0.40	
...	...	...	...	...	...	...	...	...	...	...	
6492	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	
6493	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	
6494	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	
6495	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	
6496	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	

6497 rows × 14 columns



Almost 5200 of the total number of wines seem to be "Bad" and the remaining 1297 wines "Good".

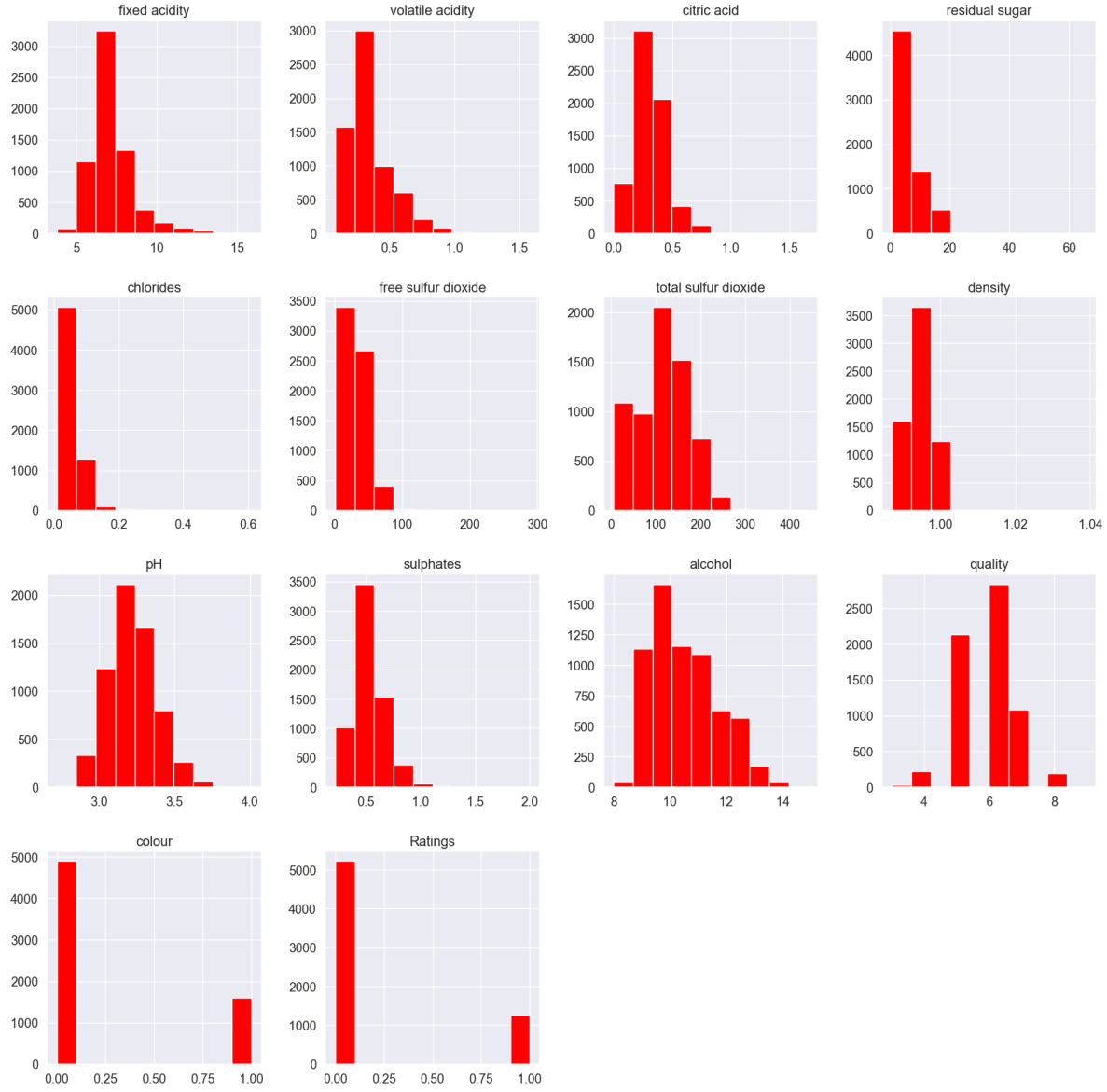
	fixed acidity	volatile acidity	citric acid	residual sugar
count	6497.000000	6497.000000	6497.000000	6497.000000
mean	7.215307	0.339666	0.318633	5.443235
std	1.296434	0.164636	0.145318	4.757804
min	3.800000	0.080000	0.000000	0.600000
25%	6.400000	0.230000	0.250000	1.800000
50%	7.000000	0.290000	0.310000	3.000000
75%	7.700000	0.400000	0.390000	8.100000
max	15.900000	1.580000	1.660000	65.800000

	chlorides	free sulfur dioxide	total sulfur dioxide	density
count	6497.000000	6497.000000	6497.000000	6497.000000
mean	0.056034	30.525319	115.744574	0.994697
std	0.035034	17.749400	56.521855	0.002999
min	0.009000	1.000000	6.000000	0.987110
25%	0.038000	17.000000	77.000000	0.992340
50%	0.047000	29.000000	118.000000	0.994890
75%	0.065000	41.000000	156.000000	0.996990
max	0.611000	289.000000	440.000000	1.038980

	pH	sulphates	alcohol	quality	colour
count	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000
mean	3.218501	0.531268	10.491801	5.818378	0.246114
std	0.160787	0.148806	1.192712	0.873255	0.430779
min	2.720000	0.220000	8.000000	3.000000	0.000000
25%	3.110000	0.430000	9.500000	5.000000	0.000000
50%	3.210000	0.510000	10.300000	6.000000	0.000000
75%	3.320000	0.600000	11.300000	6.000000	0.000000
max	4.010000	2.000000	14.900000	9.000000	1.000000

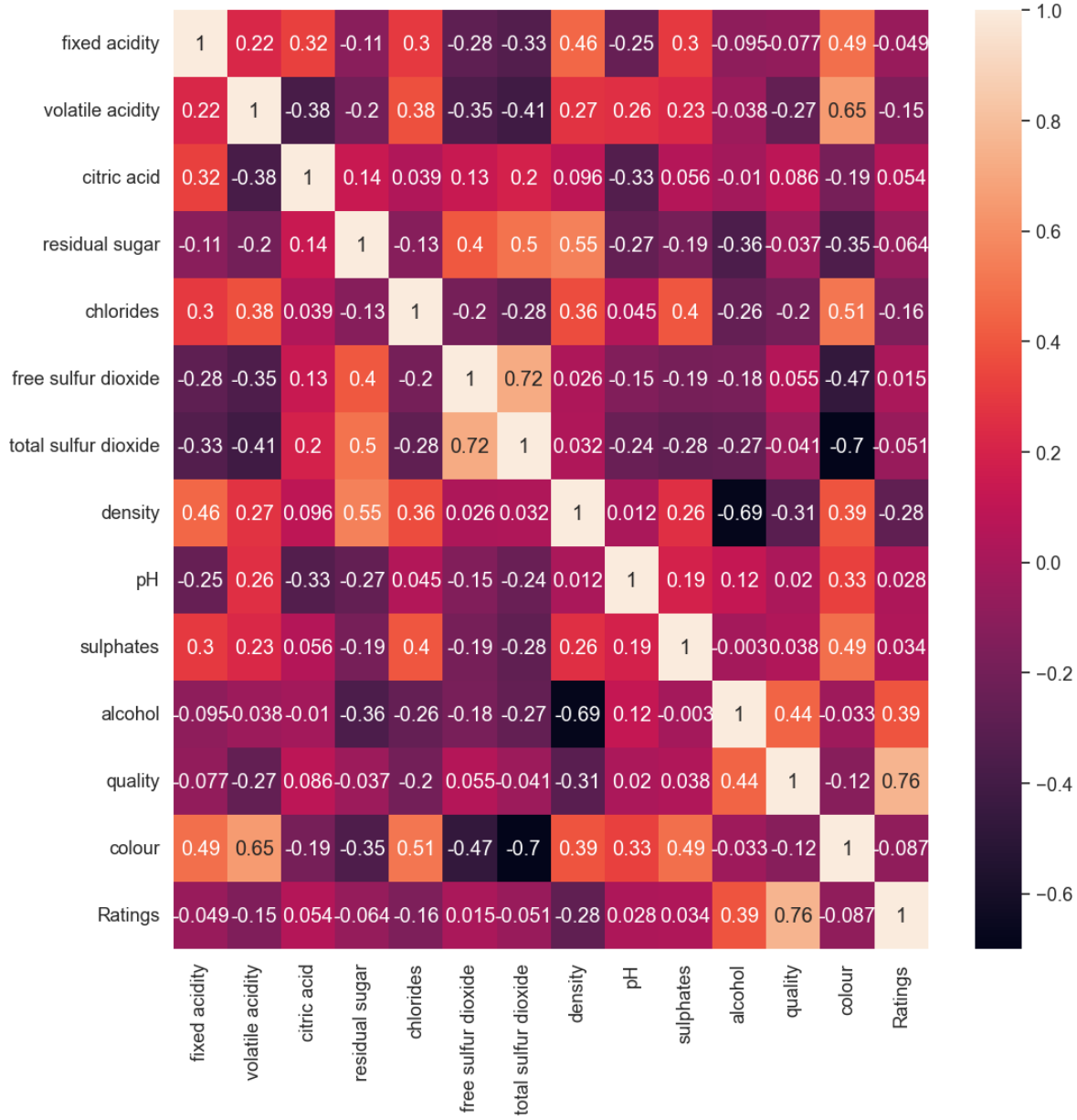
	Ratings
count	6497.000000
mean	0.196552
std	0.397421
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	1.000000

	mean	median	variance	skew	kurtosis
fixed acidity	7.215307	7.00000	1.680740	1.723290	5.061161
volatile acidity	0.339666	0.29000	0.027105	1.495097	2.825372
citric acid	0.318633	0.31000	0.021117	0.471731	2.397239
residual sugar	5.443235	3.00000	22.636696	1.435404	4.359272
chlorides	0.056034	0.04700	0.001227	5.399828	50.898051
free sulfur dioxide	30.525319	29.00000	315.041192	1.220066	7.906238
total sulfur dioxide	115.744574	118.00000	3194.720039	-0.001177	-0.371664
density	0.994697	0.99489	0.000009	0.503602	6.606067
pH	3.218501	3.21000	0.025853	0.386839	0.367657
sulphates	0.531268	0.51000	0.022143	1.797270	8.653699
alcohol	10.491801	10.30000	1.422561	0.565718	-0.531687
quality	5.818378	6.00000	0.762575	0.189623	0.232322
colour	0.246114	0.00000	0.185570	1.179095	-0.609922
Ratings	0.196552	0.00000	0.157944	1.527553	0.333522



Correlation Plot:

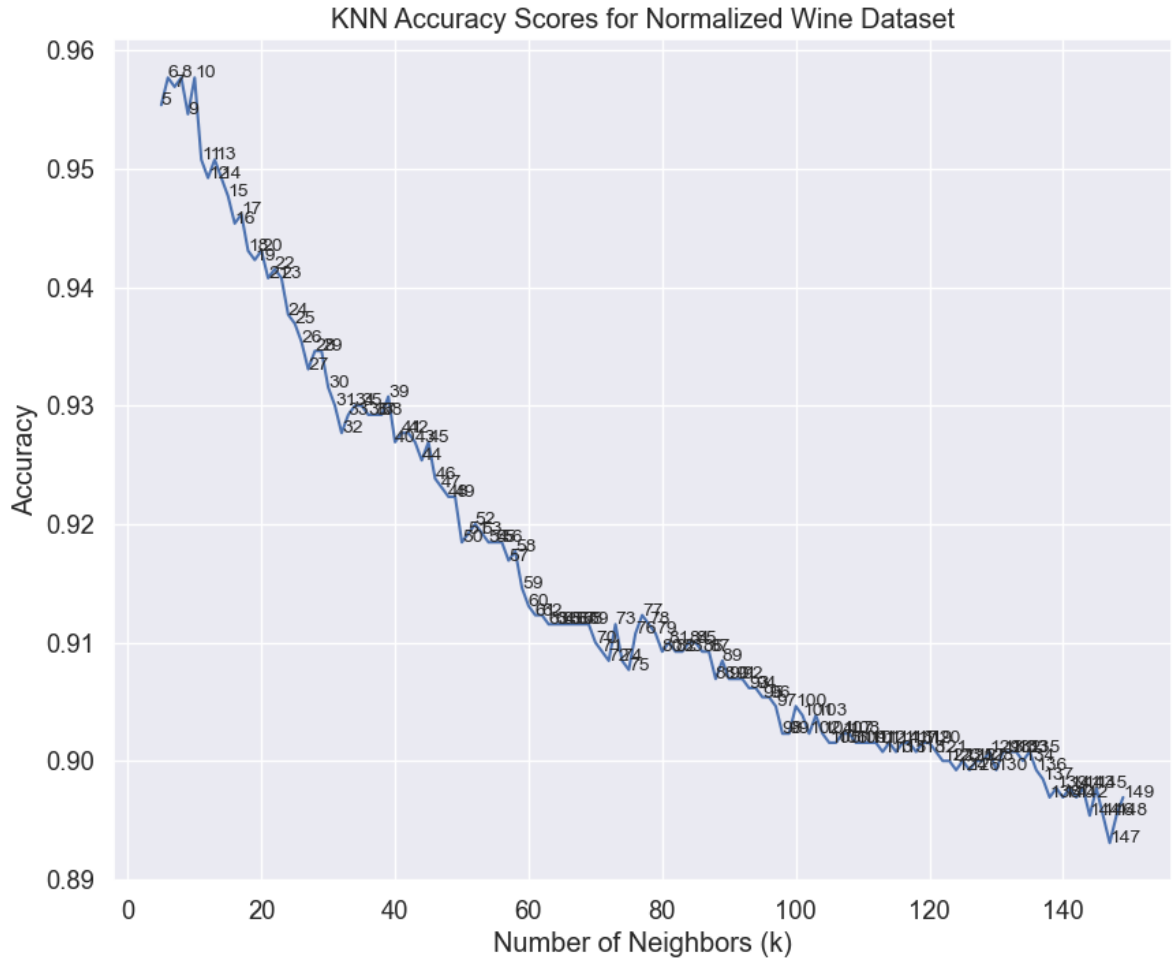
Out[268]: <AxesSubplot:>



Alcohol has the maximum correlation with quality followed by sulphates and citric acid and then fixed acidity. We can also observe that residual sugar has a significant positive correlation with density and total sulfur dioxide is strongly correlated with the type of wine.

Apply Z-score Normalization on wine dataset:

```
0       6
1       6
2       6
3       6
4       6
       ..
6492    5
6493    6
6494    6
6495    5
6496    6
Name: quality, Length: 6497, dtype: int64
The normalized dataset is:
 [[-0.16608919 -0.42318303  0.28468605 ... -1.41855821  0.20799905
  -0.57136659]
 [-0.70607349 -0.24094936  0.14704613 ... -0.83161516  0.20799905
  -0.57136659]
 [ 0.68245757 -0.36243847  0.55996589 ... -0.32852111  0.20799905
  -0.57136659]
 ...
 [-0.70607349  1.03468634 -1.29817304 ...  0.42611996  0.20799905
   1.75018984]
 [-1.01463595  1.85473786 -1.366993   ... -0.2446721  -0.93722961
   1.75018984]
 [-0.93749534 -0.1802048   1.04170561 ...  0.42611996  0.20799905
   1.75018984]]


Covariance Matrix Calculated using numpy:

1.0002  0.2190  0.3245  -0.1120 0.2982  -0.2828 -0.3291 0.4590  -0.2527 0.299
6       -0.0955 -0.0768 0.4868
0.2190  1.0002  -0.3780 -0.1960 0.3772  -0.3526 -0.4145 0.2713  0.2615  0.226
0       -0.0376 -0.2657 0.6531
0.3245  -0.3780 1.0002  0.1425  0.0390  0.1331  0.1953  0.0962  -0.3299 0.056
2       -0.0105 0.0855  -0.1874
-0.1120 -0.1960 0.1425  1.0002  -0.1290 0.4029  0.4956  0.5526  -0.2674 -0.18
60      -0.3595 -0.0370 -0.3489
0.2982  0.3772  0.0390  -0.1290 1.0002  -0.1951 -0.2797 0.3627  0.0447  0.395
7       -0.2570 -0.2007 0.5128
-0.2828 -0.3526 0.1331  0.4029  -0.1951 1.0002  0.7210  0.0257  -0.1459 -0.18
85      -0.1799 0.0555  -0.4717
-0.3291 -0.4145 0.1953  0.4956  -0.2797 0.7210  1.0002  0.0324  -0.2384 -0.27
58      -0.2658 -0.0414 -0.7005
0.4590  0.2713  0.0962  0.5526  0.3627  0.0257  0.0324  1.0002  0.0117  0.259
5       -0.6869 -0.3059 0.3907
-0.2527 0.2615  -0.3299 -0.2674 0.0447  -0.1459 -0.2384 0.0117  1.0002  0.192
2       0.1213  0.0195  0.3292
0.2996  0.2260  0.0562  -0.1860 0.3957  -0.1885 -0.2758 0.2595  0.1922  1.000
2       -0.0030 0.0385  0.4873
-0.0955 -0.0376 -0.0105 -0.3595 -0.2570 -0.1799 -0.2658 -0.6869 0.1213  -0.00
30      1.0002  0.4444  -0.0330
-0.0768 -0.2657 0.0855  -0.0370 -0.2007 0.0555  -0.0414 -0.3059 0.0195  0.038
5       0.4444  1.0002  -0.1193
0.4868  0.6531  -0.1874 -0.3489 0.5128  -0.4717 -0.7005 0.3907  0.3292  0.487
3       -0.0330 -0.1193 1.0002
```




```
Test Accuracy: 0.9176923076923077
CV Accuracy scores: [1. 1. 1. 1. 1.]
CV Average accuracy: 1.0
```

PCA preprocessing on Wine dataset

Out[273]:
```
array([[-2.06707183,  3.48606943, -0.12160483, ...,  0.355487  ,
         0.09523024,  0.03915981],
       [-0.27234588, -0.50787251, -0.41460196, ..., -0.14436514,
         0.04905848,  0.16266364],
       [-0.38931517,  0.29383078,  0.53340294, ..., -0.20948338,
        -0.65935401, -0.18165748],
       ...,
       [ 2.720703  , -0.90429614, -1.28925071, ...,  0.71079279,
         0.17217269,  0.25399957],
       [ 3.05387095, -0.52363091, -2.62264968, ...,  0.87445042,
        -0.15998844, -0.06881234],
       [ 1.79287529, -0.72259628,  0.31706237, ...,  1.30046828,
         0.76258543,  0.16620553]])
```

Out[274]:

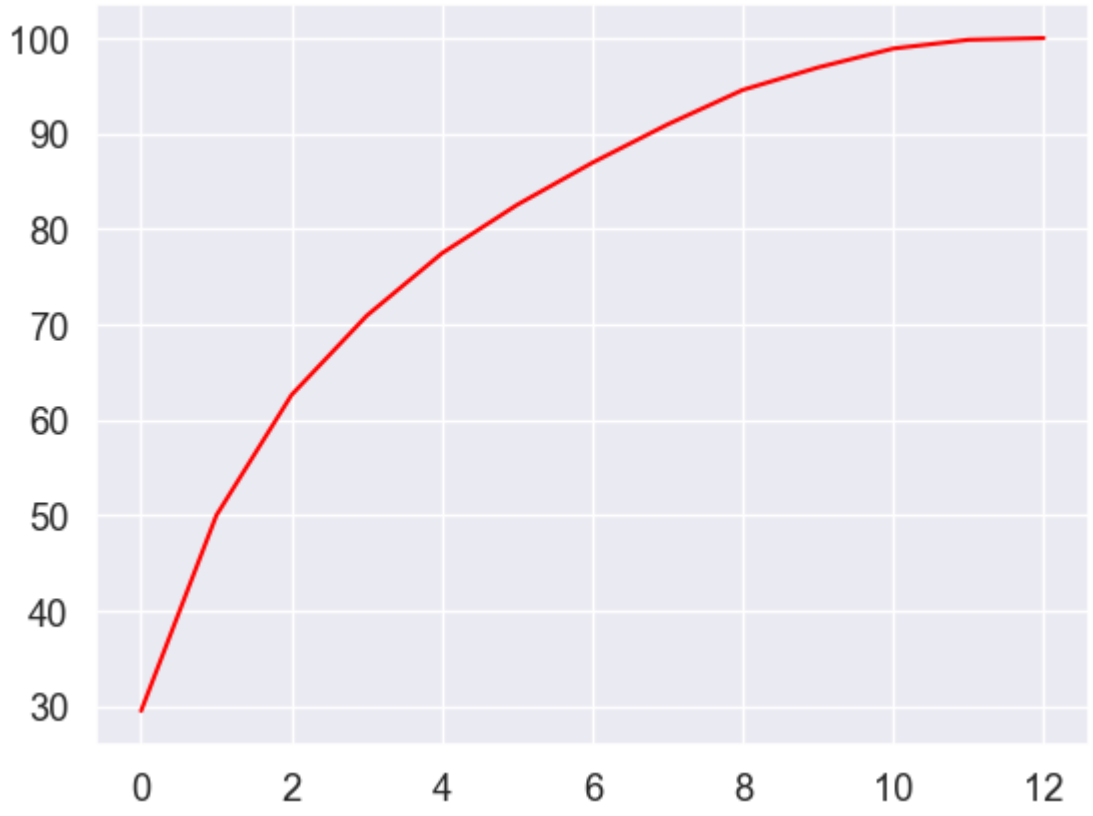
	component1	component2	component3	component4	component5	component6	component7
0	-2.067072	3.486069	-0.121605	0.225732	1.926477	0.354486	-0.672678
1	-0.272346	-0.507873	-0.414602	-0.251146	-0.448951	-0.855633	-0.837531
2	-0.389315	0.293831	0.533403	-0.253599	0.420879	-0.646033	-0.108228
3	-1.708927	0.968794	-0.219406	0.149823	0.055460	0.226626	-0.012470
4	-1.708927	0.968794	-0.219406	0.149823	0.055460	0.226626	-0.012470

Out[275]:
```
array([0.29492412, 0.20484445, 0.12627433, 0.08311596, 0.06536559,
       0.05083677, 0.04407412, 0.04007519, 0.03618948, 0.02352702,
       0.01975193, 0.00917425, 0.00184679])
```

Out[276]:
```
array([ 29.49,  49.97,  62.6 ,  70.91,  77.45,  82.53,  86.94,  90.95,
        94.57,  96.92,  98.9 ,  99.82, 100.  ])
```

Out[277]:
```
array([[ 0.26022761,  0.36378537, -0.11319392, -0.23277398,  0.30248902,
        -0.33871316, -0.40228457,  0.16134445,  0.17486612,  0.27953014,
        -0.00438771, -0.09658937,  0.46988304],
       [ 0.21697768,  0.04063327,  0.1652622 ,  0.38999091,  0.21461462,
         0.18038247,  0.21801564,  0.53387129, -0.18258784,  0.06996466,
        -0.49463822, -0.27584039,  0.04159581],
       [ 0.46915601, -0.27753553,  0.58755451, -0.07691544,  0.04901716,
        -0.10171768, -0.10349402, -0.05064624, -0.40644534,  0.17017062,
         0.21223488,  0.29407328, -0.00515413],
       [-0.15221794, -0.0988973 ,  0.05585934,  0.1409448 ,  0.11802731,
         0.33598581,  0.15119611,  0.14728961,  0.45593175,  0.54443786,
         0.0924771 ,  0.49999028,  0.09931403],
       [ 0.16420616,  0.13567635, -0.22703334,  0.50195548, -0.4279513 ,
        -0.21043504, -0.20327797,  0.30757457, -0.03611788, -0.25574119,
         0.12151437,  0.44307302,  0.09994023],
       [-0.02600784,  0.38249628, -0.35416504,  0.05274959,  0.41655005,
         0.30502379,  0.11684492, -0.16177157, -0.56167229,  0.00674719,
         0.1729653 ,  0.26868635,  0.02824745],
       [ 0.3868432 ,  0.40294671,  0.10595656, -0.13885955, -0.45217669,
         0.44036475,  0.24390134, -0.00370185,  0.11276566,  0.03066965,
         0.31551766, -0.25917623,  0.13860244],
       [ 0.03181395, -0.18196056, -0.4133581 , -0.03047916, -0.41017361,
        -0.13581435,  0.0184071 , -0.0299972 , -0.33985109,  0.66555974,
        -0.11850286, -0.15297009, -0.10122474],
       [-0.32809868,  0.27168218,  0.2906849 ,  0.47464037,  0.08487095,
        -0.29460432,  0.02474758, -0.03542723, -0.03627823,  0.24884103,
         0.46895273, -0.32552487, -0.15419297],
       [ 0.10889424, -0.51968453, -0.2497291 ,  0.23754043,  0.16676955,
         0.36129991, -0.37140759,  0.0430805 ,  0.01712679, -0.07384153,
         0.38220047, -0.3352888 ,  0.1920652 ],
       [-0.4722093 ,  0.12979518,  0.32485317, -0.00408122, -0.28055358,
         0.32649347, -0.46716453, -0.06333852, -0.25560562,  0.04477967,
        -0.24688386,  0.01512309,  0.34430043],
       [-0.21912105, -0.24108752, -0.00161111, -0.05294132, -0.05238527,
        -0.23414618,  0.53145939,  0.05878588, -0.15947249, -0.10382993,
         0.14265139, -0.01467474,  0.70057617],
       [-0.27041153, -0.01273268, -0.00846961, -0.45657506, -0.0166051 ,
         0.03258504, -0.05491676,  0.72967519, -0.1601302 , -0.04557912,
         0.31007441,  0.01434882, -0.23944217]])
```

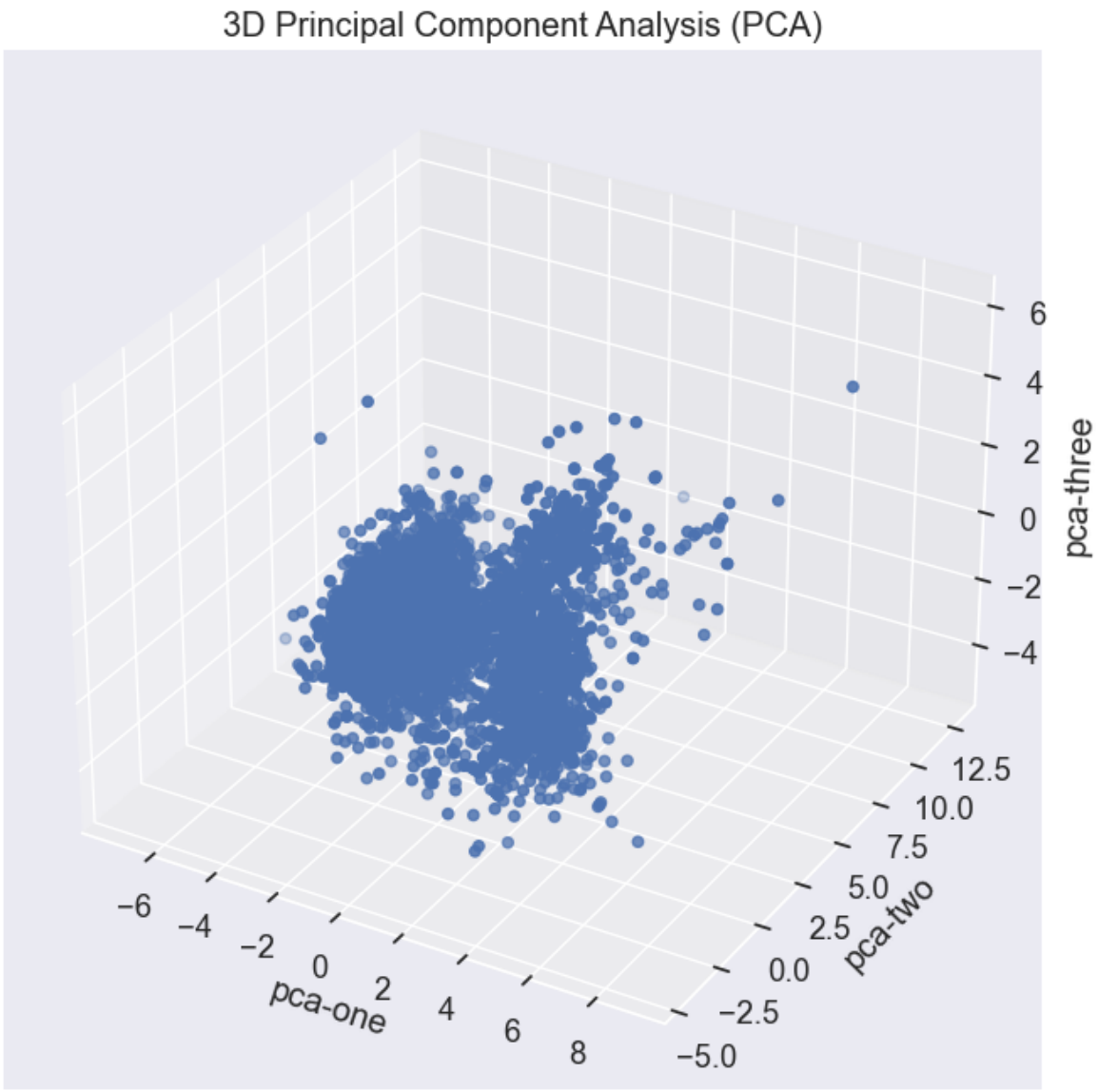
Out[278]: `[<matplotlib.lines.Line2D at 0x1b24d40d940>]`

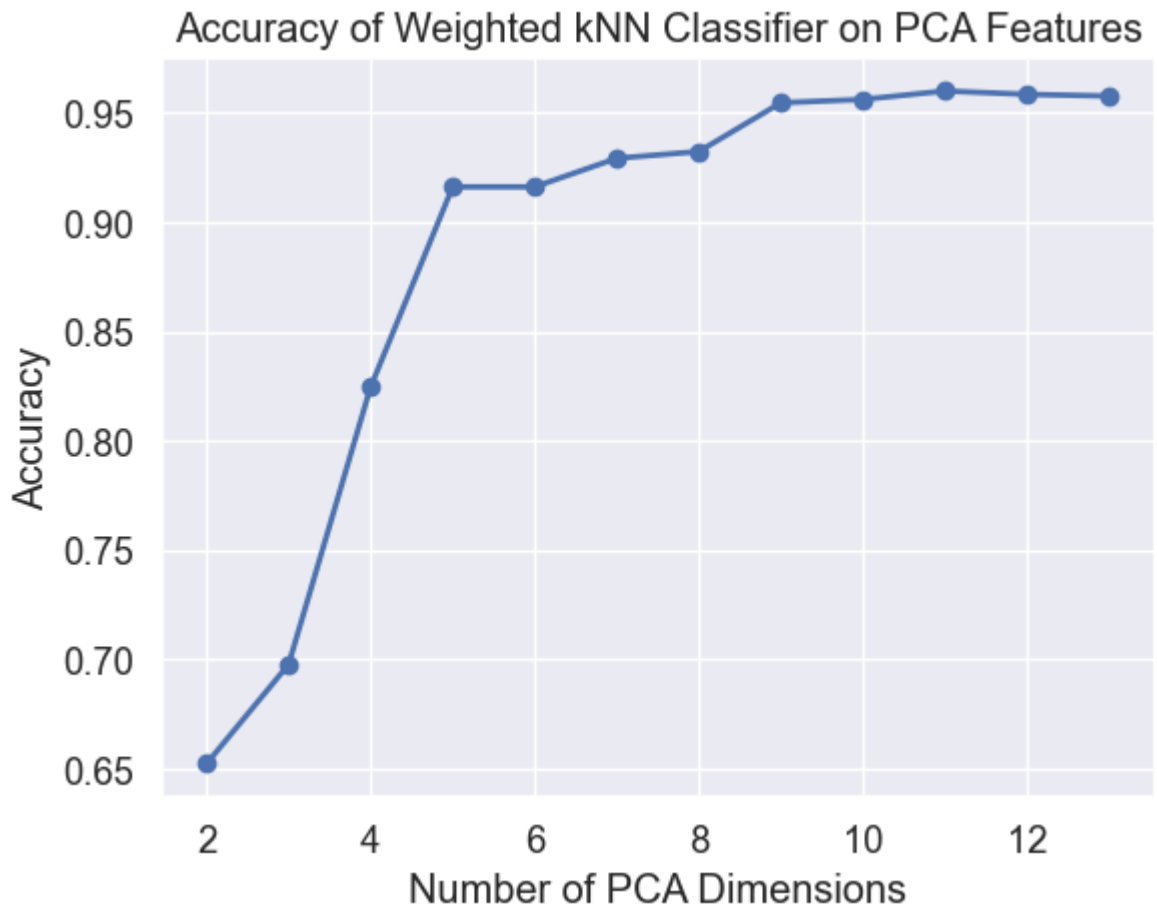


Our problem suggests to use the first 3 principal components.

Out[280]:

	component1	component2	component3
0	-2.067072	3.486069	-0.121605
1	-0.272346	-0.507873	-0.414602
2	-0.389315	0.293831	0.533403
3	-1.708927	0.968794	-0.219406
4	-1.708927	0.968794	-0.219406


3D Principal Component Analysis (PCA)
pca-one
pca-two
pca-three
−6
−4
−2
0
2
4
6
8
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
−4
−2
0
2
4
6




```
Test Accuracy: 0.7115384615384616
CV Accuracy scores: [0.67980769 0.70865385 0.67853705 0.68046198 0.70452358]
CV Average accuracy: 0.6903968312726734
```

LDA preprocessing on Wine dataset

```
Mean CV accuracy: 1.0
Test Accuracy: 1.0
CV Accuracy scores: [1. 1. 1. 1. 1.]
CV Average accuracy: 1.0
```

Conclusion for wine dataset after using PCA and LDA

The accuracy of KNN on the raw wine dataset is 98.85%, which is the highest accuracy among all settings. However, the accuracy drops significantly to 88.92% when using PCA as a preprocessing technique with three principal components. This drop in accuracy can be explained by the fact that PCA reduces the dimensionality of the dataset by projecting it onto a lower-dimensional space, which may cause some information loss.

On the other hand, when using LDA with three linear discriminants, the accuracy increases to 99.92%. This increase in accuracy can be explained by the fact that LDA seeks to find the linear combinations of features that maximize the separation between classes, leading to a better representation of the data for classification.

Conclusion for abalone dataset after using PCA and LDA

The results show that the accuracy of the KNN algorithm on the Abalone dataset is significantly improved after applying PCA and LDA. The accuracy of the KNN algorithm is 27.06% for the raw dataset, whereas it is increased to 62.68% and 69.62% after applying PCA and LDA, respectively.

The higher accuracy values obtained after applying PCA and LDA can be attributed to the fact that these techniques help to reduce the dimensionality of the dataset and select the most important features that contribute to the classification task. PCA identifies the directions of maximal variance in the data and projects the data onto a lower-dimensional subspace, while LDA finds the linear combinations of features that maximize the class separability. Both techniques help to reduce the noise in the data and make it easier for the KNN algorithm to find the correct class label for new data points.