CSE-619 Special Topics in Computer Engineering

Author: Uğur Ceylan

Final Project Report

This report include my works for wide known dataset Wine-Quality dataset in within the scope of the final project of CSE-619 Special Topics in Computer Engineering lecture. Basically, dataset include two wine samples which is red and white. Each dataset include 11 independent features which are "fixed acidity";"volatile acidity";"citric acid";"residual sugar"; "chlorides"; "free sulfur dioxide"; "total sulfur dioxide"; "density"; "pH"; "sulphates"; "alcohol"; and one dependent feature that is "quality" and specify wine quality. In this report we emphasize on data statics info such as Fisher distances, standart data distribution boxplots etc. and analysis each dataset with feature dimention reduction techniques such as principle component analysis(PCA) and self orginizing maps (SOM) and on the other hand analysis each dataset and compare their results with unsupervised learning algorithms which are K-Means, K-Center, DBSCAN and Farthest-First.

Introduction

Before dive into give data details, it is shown head 5 samples of each red and white wine datasets.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

Table 1: Red-wine data top 5 records. Data records is numeric, specifically quality is integer type and others are float type.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.0010	3.00	0.45	8.8	6
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.9940	3.30	0.49	9.5	6
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.9951	3.26	0.44	10.1	6
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6

Table 2: White-wine data top 5 records. Data records is numeric, specifically quality is integer type and others are float type.

For describe and look data information in this project we used pandas python library. So, let’s look at details of each data with dataframe decribe function.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	qu
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.00
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.63
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.80
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.00
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.00
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.00
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.00
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.00

Table 3: Red-wine data statisctic information.

Let's explain table 3 information:

- **count** -> show total count of feature samples (eg, count of records for fixed acidity is 1599 and also the other feature are the same number)

- **mean** -> mean of feature samples(eg, mean of fixed acidity is 8.319637)

- **std** -> std of feature samples(eg, std of fixed acidity is 1.741096)

- **min** -> minimum value of feature samples(eg, minimum value of fixed acidity is 4.6)

- **max** -> maximum value of feature samples(eg, maximum value of fixed acidity is 15.9)

- **25%** -> first quartile that is %25 percent of feature sample values smaller than this value(eg, %25 percentile of fixed acidity is 7.1)

- **50%** -> second quartile that is %50 percent of feature sample values smaller than this value(eg, %50 percentile of fixed acidity is 7.9)

- **75%** -> third quartile that is %75 percent of feature sample values smaller than this value(eg, %75 percentile of fixed acidity is 9.2)

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	qu
count	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.00
mean	6.854788	0.278241	0.334192	6.391415	0.045772	35.308085	138.360657	0.994027	3.188267	0.489847	10.514267	5.87
std	0.843868	0.100795	0.121020	5.072058	0.021848	17.007137	42.498065	0.002991	0.151001	0.114126	1.230621	0.88
min	3.800000	0.080000	0.000000	0.600000	0.009000	2.000000	9.000000	0.987110	2.720000	0.220000	8.000000	3.00
25%	6.300000	0.210000	0.270000	1.700000	0.036000	23.000000	108.000000	0.991723	3.090000	0.410000	9.500000	5.00
50%	6.800000	0.260000	0.320000	5.200000	0.043000	34.000000	134.000000	0.993740	3.180000	0.470000	10.400000	6.00
75%	7.300000	0.320000	0.390000	9.900000	0.050000	46.000000	167.000000	0.996100	3.280000	0.550000	11.400000	6.00
max	14.200000	1.100000	1.660000	65.800000	0.346000	289.000000	440.000000	1.038980	3.820000	1.080000	14.200000	9.00

Table 4: White-wine data statisctic information.

Let's explain table 4 information:

- **count** -> show total count of feature samples (eg, count of records for fixed acidity is 4898 and also the other feature are the same number)

- **mean** -> mean of feature samples(eg, mean of fixed acidity is 6.854788)

- **std** -> std of feature samples(eg, std of fixed acidity is 0.843868)

- **min** -> minimum value of feature samples(eg, minimum value of fixed acidity is 3.8)

- **max** -> maximum value of feature samples(eg, maximum value of fixed acidity is 14.2)

- **25%** -> first quartile that is %25 percent of feature sample values smaller than this value(eg, %25 percentile of fixed acidity is 6.3)

- **50%** -> second quartile that is %50 percent of feature sample values smaller than this value(eg, %50 percentile of fixed acidity is 6.8)

- **75%** -> third quartile that is %75 percent of feature sample values smaller than this value(eg, %75 percentile of fixed acidity is 7.3)

I. BOXPOTS

A boxplot is a conventional way of displaying the distribution of data based on a five number summary ("minimum", first quartile (Q1), median, third quartile (Q3), and "maximum"). It can tell us about outliers and what their values are. It can also tell us if data is symmetrical, how tightly data is grouped, and if and how data is skewed.

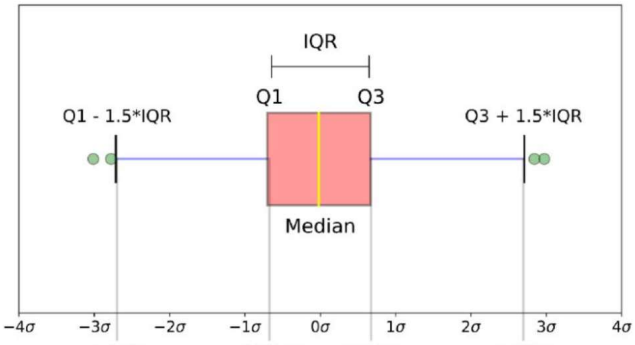

Figure 1: Box plot representation

- **median (Q2/50th Percentile)**: the middle vaue of the dataset.
- **first quartile (Q1/25th Percentile)**: the middle number between the smallest number (not the “minimum”) and the median of the dataset.
- **third quartile (Q3/75th Percentile)**: the middle value between the median and the highest value (not the “maximum”) of the dataset.
- **interquartile range (IQR)**: 25th to the 75th percentile.
- **whiskers** (shown in blue)
- **outliers** (shown as green circles)
- **"maximum"**: Q3 + 1.5*IQR
- **"minimum"**: Q1 -1.5*IQR

For visualization data we used matplotlib and seaborn python libraries.

We plot boxplots for red and white wines features individually and on the same plots to compare their results. It's explained each box plot for each feature separately via comparing their values

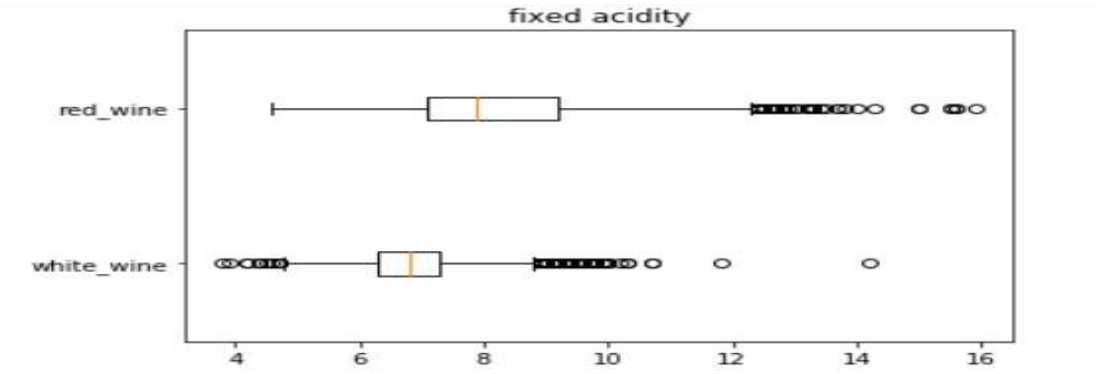

Figure 2: Box plot **fixed acidity** for red wine and white wine

Fixed acidity: For red win outliers only exceed max value whereas in white win values outliers distributed over more maximum value or less than min value. Also, for this feature, most of data in red wine distributed in in wide range while in white wine is in more narrow range. In red wine data skewed a bit on left, while for white is approximately normal.

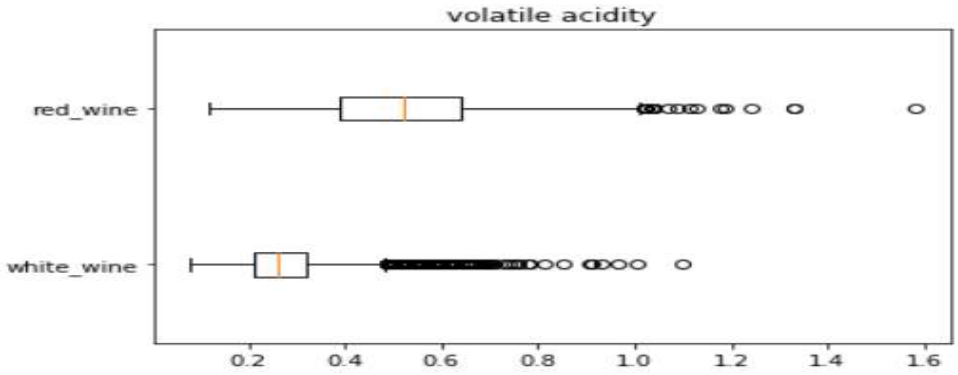

Figure 3: Box plot **volatile acidity** for red wine and white wine

Volatile acidity: For both wines outliers distributed over more maximum value, outliers are more in white data. Also, for this feature, most of data in red wine distributed in in wide range while in white wine is in more narrow range. For this feature in both data normal distributed.

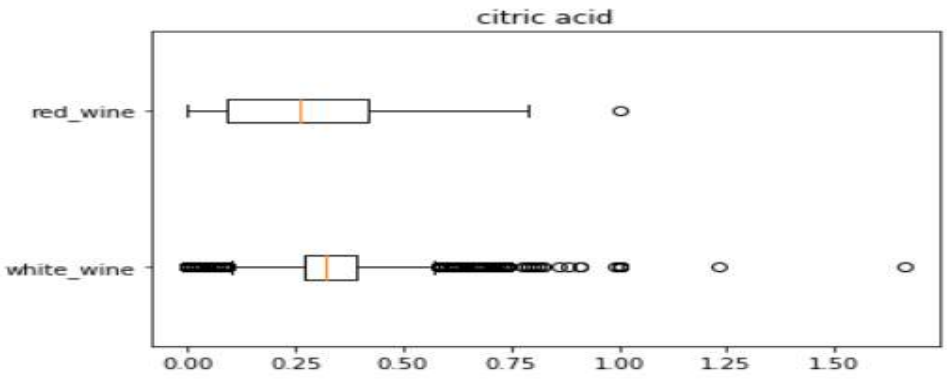

Figure 4: Box plot **citric acid** for red wine and white wine

Citric acid: In white-wine data this have more much outliers and left-skewed, in red-wine data more wide range distributed and has normal distributed.

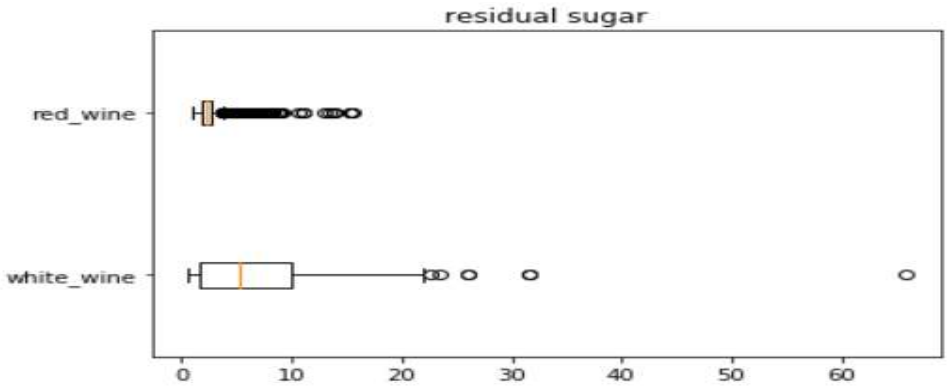

Figure 5: Box plot **residual sugar** for red wine and white wine

Residual sugar: In red-wine data has a lots of outliers and range is very narrow. On the other hand for white-wine data opposite.

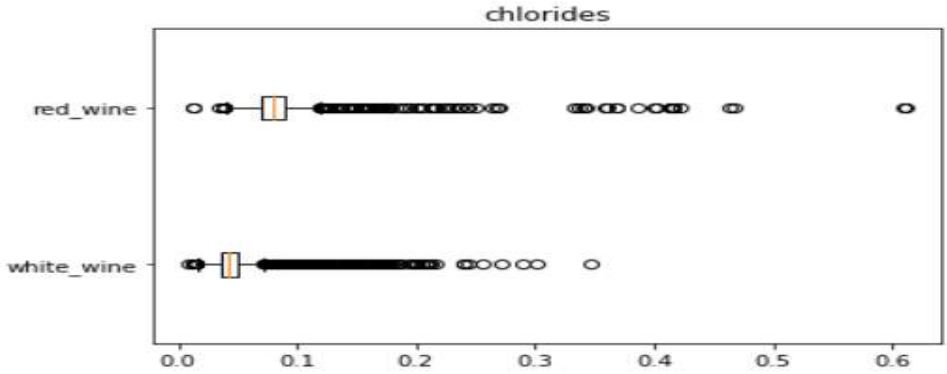

Figure 6: Box plot **chlorides** for red wine and white wine

Chlorides: Both data have lots of outliers and their distribution in narrow range with normal.

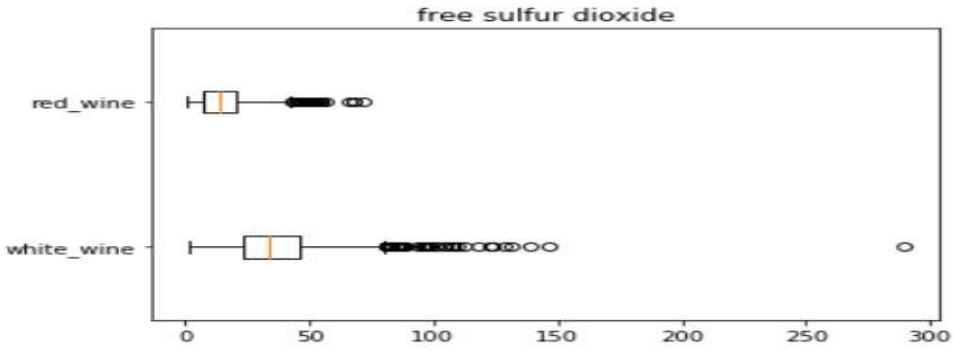

Figure 7: Box plot **free sulfur dioxide** for red wine and white wine

Free sulfur dioxide: Both have same shape of data with more wide range distributed in white-data.

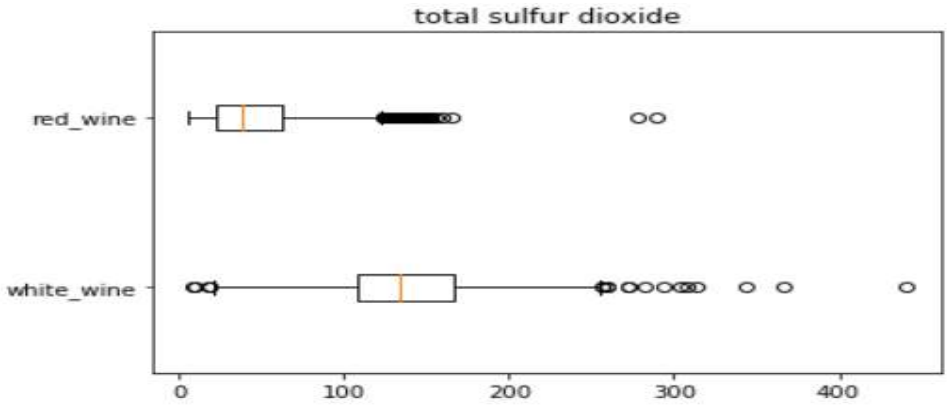

Figure 8: Box plot **total sulfur dioxide** for red wine and white wine

Total sulfur dioxide: In red-wine data outliers more much, one max side and data is left skewed, in white-data outliers in both side and data has normal distribution.

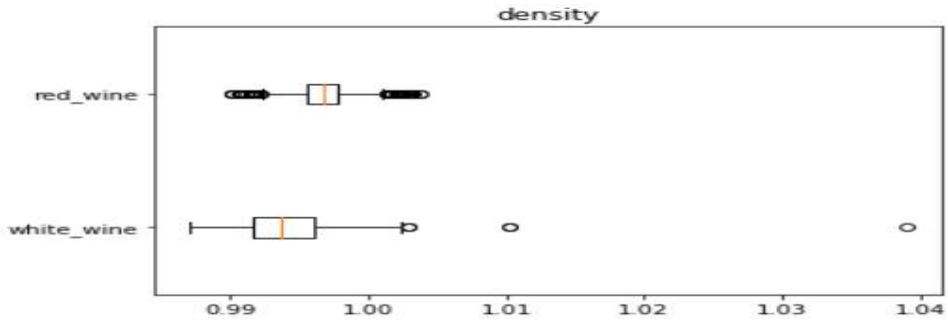

Figure 9: Box plot **density** for red wine and white wine

Density: In red-wine data this feature has outliers on both sides and normal distribution with narrow gap. In white-data outliers are less, wide range with normal distribution.

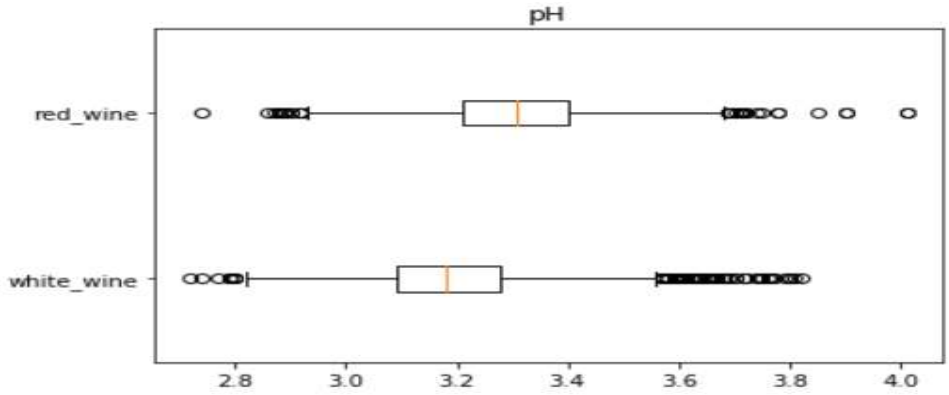

Figure 10: Box plot **pH** for red wine and white wine

PH: Both has similar distribution and have outliers, normal distribution (not skewed any side)

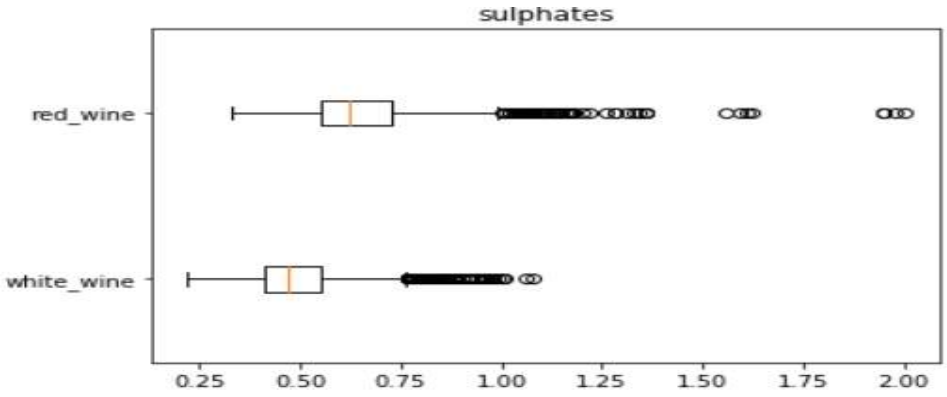

Figure 11: Box plot **sulphates** for red wine and white wine

Sulphates: In red-wine data outliers are more further from median value

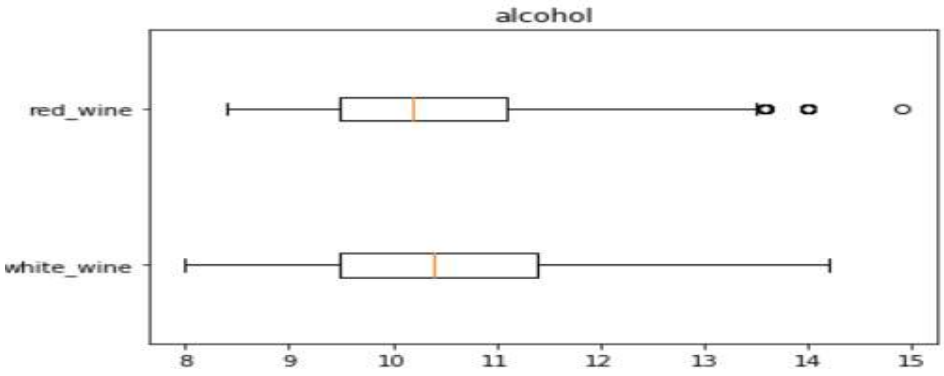

Figure 12: Box plot **alcohol** for red wine and white wine

Alcohol: In both has the near range

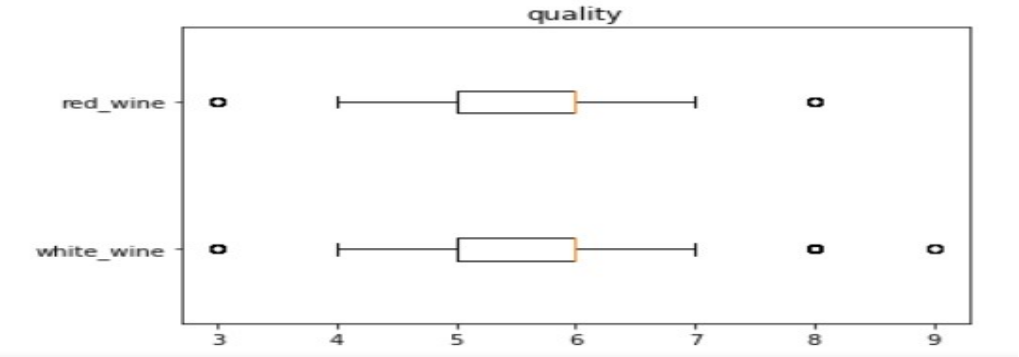

Figure 13: Box plot **quality** for red wine and white wine

Quality: In both has the similar distribution.

II. CORRELATION

Correlation give us between relation between two attributes via give a number between -1 and 1. If relationship between attributes is strong positive relationship then this value 1 or close to 1 and if this value close to -1 relation is strong negative relationship, other cases no relation.

Correlation calculated according to formula in figure 14. Formula implemented in numPy python numeric library from sratch and details codes in jupyter notebook.

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

r = correlation coefficient
x_i = values of the x-variable in a sample
$\bar{x}$ = mean of the values of the x-variable
y_i = values of the y-variable in a sample
$\bar{y}$ = mean of the values of the y-variable

Equation 1: Person-correlation formula

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
fixed acidity	1	-0.2561	0.6717	0.1148	0.0937	-0.1538	-0.1132	0.668	-0.683	0.183	-0.0617	0.1241
volatile acidity	-0.2561	1	-0.5525	0.0019	0.0613	-0.0105	0.0765	0.022	0.2349	-0.261	-0.2023	-0.3906
citric acid	0.6717	-0.5525	1	0.1436	0.2038	-0.061	0.0355	0.3649	-0.5419	0.3128	0.1099	0.2264
residual sugar	0.1148	0.0019	0.1436	1	0.0556	0.187	0.203	0.3553	-0.0857	0.0055	0.0421	0.0137
chlorides	0.0937	0.0613	0.2038	0.0556	1	0.0056	0.0474	0.2006	-0.265	0.3713	-0.2211	-0.1289
free sulfur dioxide	-0.1538	-0.0105	-0.061	0.187	0.0056	1	0.6677	-0.0219	0.0704	0.0517	-0.0694	-0.0507
total sulfur dioxide	-0.1132	0.0765	0.0355	0.203	0.0474	0.6677	1	0.0713	-0.0665	0.0429	-0.2057	-0.1851
density	0.668	0.022	0.3649	0.3553	0.2006	-0.0219	0.0713	1	-0.3417	0.1485	-0.4962	-0.1749
pH	-0.683	0.2349	-0.5419	-0.0857	-0.265	0.0704	-0.0665	-0.3417	1	-0.1966	0.2056	-0.0577
sulphates	0.183	-0.261	0.3128	0.0055	0.3713	0.0517	0.0429	0.1485	-0.1966	1	0.0936	0.2514
alcohol	-0.0617	-0.2023	0.1099	0.0421	-0.2211	-0.0694	-0.2057	-0.4962	0.2056	0.0936	1	0.4762
quality	0.1241	-0.3906	0.2264	0.0137	-0.1289	-0.0507	-0.1851	-0.1749	-0.0577	0.2514	0.4762	1

Figure 14: Person-correlation between attributes of red-wine dataset

There are some relations between some features and specified as following:

- **citric acid** and **fixed acidity** have **positive** correlation with 0.6717 value
- **total sulfur dioxide** and **free sulfur dioxide** have **positive** correlation with 0.6677 value
- **density** and **fixed acidity** have **positive** correlation with value 0.668
- **pH** and **fixed acidity** have **negative** correlation with -0.683 value
- **pH** and **citric acid** have **negative** correlation with -0.5419 value
- **citric acid** and **volatile acidity** have **negative** correlation with -0.5525
- **density** and **alcohol** have **negative** correlation with value -0.4962
- **quality** and **alcohol** have **positive** correlations with 0.4762 value

As we see relation between some features affect the other ones, for example relation between density and fixed acidity has positive and between fixed acidity and density negative relations.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
fixed acidity	1	-0.0227	0.2892	0.089	0.0231	-0.0494	0.0911	0.2653	-0.4259	-0.0171	-0.1209	-0.1137
volatile acidity	-0.0227	1	-0.1495	0.0643	0.0705	-0.097	0.0893	0.0271	-0.0319	-0.0357	0.0677	-0.1947
citric acid	0.2892	-0.1495	1	0.0942	0.1144	0.0941	0.1211	0.1495	-0.1637	0.0623	-0.0757	-0.0092
residual sugar	0.089	0.0643	0.0942	1	0.0887	0.2991	0.4014	0.839	-0.1941	-0.0267	-0.4506	-0.0976
chlorides	0.0231	0.0705	0.1144	0.0887	1	0.1014	0.1989	0.2572	-0.0904	0.0168	-0.3602	-0.2099
free sulfur dioxide	-0.0494	-0.097	0.0941	0.2991	0.1014	1	0.6155	0.2942	-0.0006	0.0592	-0.2501	0.0082
total sulfur dioxide	0.0911	0.0893	0.1211	0.4014	0.1989	0.6155	1	0.5299	0.0023	0.1346	-0.4489	-0.1747
density	0.2653	0.0271	0.1495	0.839	0.2572	0.2942	0.5299	1	-0.0936	0.0745	-0.7801	-0.3071
pH	-0.4259	-0.0319	-0.1637	-0.1941	-0.0904	-0.0006	0.0023	-0.0936	1	0.156	0.1214	0.0994
sulphates	-0.0171	-0.0357	0.0623	-0.0267	0.0168	0.0592	0.1346	0.0745	0.156	1	-0.0174	0.0537
alcohol	-0.1209	0.0677	-0.0757	-0.4506	-0.3602	-0.2501	-0.4489	-0.7801	0.1214	-0.0174	1	0.4356
quality	-0.1137	-0.1947	-0.0092	-0.0976	-0.2099	0.0082	-0.1747	-0.3071	0.0994	0.0537	0.4356	1

Figure 15: Person-correlation between attributes of white-wine dataset.

There are some relations between some features and specified as following:

- **residual sugar** and **density** have **positive** correlation with 0.839 value
- **free sulfur dioxide** and **total sulfur dioxide** have **positive** correlation with 0.6155 value
- **total sulfur dioxide** and **density** have **positive** correlation with 0.5299 value
- **alcohol** and **density** have **negative** correlations with -0.7801 value
- **alcohol** and **residual sugar** have **negative** correlations with -0.4506 value

As we see relation between some features affect the other ones, for example relation between alcohol and residual sugar has negative and between residual sugar and density positive relations.

III. FISHER DISTANCE

A *Fisher distance* calculated as absolute differences between means of features in red and white wines and divided by their total of squares standard values of two same features in red and white wines. Formula is given in equation 2 and coded in numpy library.

fisher_distance = (mean (x_feature_in_red_wine) + mean (x_feature_in_white_wine)) / (std (x_feature_in_red_wine) ^2 + std (x_feature_in_white_wine) ^2) (Equation 2)

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
pure_data	0.391515	5.914233	1.202588	0.139047	15.493974	0.048762	0.031828	217.483327	2.635242	4.03247	0.034459

Table 5: Fisher distance between each attributes in two wine datasets with orginal data.

Fisher distance calculated for each attributes between red and white wine datasets. Results are given in table 5. **Density** has maximum fisher distance with 217.48 value, followed by **chlorides** with values 15.48. So if I reduced dimensionality into two dimension, we can chose those two features.

IV. PCA and FISHER DISTANCE

Principle component analysis steps implemented as following in numpy library:

- Subtract the mean of each variable from the dataset (this is not mandatory)
- calculated covariance for each columns with numpy cov function
- obtained eigen vectors and eigen values with numpy linalg eigh function
- sorted eigen values according to their values descending order
- completed operation via apply dot product of eigen vectors and the data

Actually before applied data normalizer (MinMax, StandarScaler) can be implemented for PCA results for be better because interval of each feature are different range as we see in boxplots. But in here we omit it.

Additionally we used PCA function of sklearn library for comparing my PCA results and sklearn lib PCA results. And we saw my PCA and sklearn PCA results are the same values but for same columns have opponent sign, it's also normal of course. So we have used sklearn PCA results to rescue inconsistency result.

We also get fisher distance of results PCA data and compare this fisher distance with original data fisher distance. PCA fisher distance results are given in table 6.

	total sulfur dioxide	free sulfur dioxide	residual sugar	alcohol	fixed acidity	pH	citric acid	sulphates	volatile acidity	chlorides	density
pca_data	1.340822e-18	5.410794e-18	1.735083e-18	3.195989e-16	2.422000e-16	6.765779e-15	8.370187e-15	3.948588e-15	2.393932e-15	3.195311e-14	6.405426e-11

Table 6: Fisher distance between each attributes in two wine datasets with PCA data.

Density has maximum fisher distance with 6.405426e-11 value, followed by **chlorides** with values 3.195311e-14. So if I reduced dimensionality into two dimension, we can chose those two features.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
pure_data	3.915152e-01	5.914233e+00	1.202588e+00	1.390474e-01	1.549397e+01	4.876209e-02	3.182831e-02	2.174833e+02	2.635242e+00	4.032470e+00	3.445906e-02
pca_data	2.422000e-16	2.393932e-15	8.370187e-15	1.735083e-18	3.195311e-14	5.410794e-18	1.340822e-18	6.405426e-11	6.765779e-15	3.948588e-15	3.195989e-16

Table 7: Fisher distance between each attributes in two wine datasets with original and PCA data.

As we see before both maximum fisher distances (in original data and after PCA applied) are the same features.

We also calculated for each attributes eigen values on red and white wine data sets.
Red wine eigen values as following: {

```
'total sulfur dioxide': 1133.8070755113813,
 'free sulfur dioxide': 57.93541077477589,
 'fixed acidity': 3.101302284183163,
 'residual sugar': 1.8194153155263988,
 'alcohol': 1.0463403600411831,
 'volatile acidity': 0.04139672938211751,
 'sulphates': 0.02319265777853163,
 'pH': 0.01134646854369882,
 'citric acid': 0.010077984126299684,
 'chlorides': 0.0014549975485606906,
 'density': 5.614826673229536e-07
}
```

White wine eigen values as following: {

```
'total sulfur dioxide': 1931.5133157556152,
 'free sulfur dioxide': 168.452894944071,
 'residual sugar': 21.56099321438453,
 'alcohol': 1.0744203190040373,
 'fixed acidity': 0.6867086338130327,
 'pH': 0.018531883309064728,
 'citric acid': 0.014289805087294114,
 'sulphates': 0.011446104927402556,
 'volatile acidity': 0.00864204554993922,
 'chlorides': 0.0003960569839009528,
 'density': 3.168392615457504e-07
}
```

As we see fisher distance and eigen values are negative correlation. Eigen value of density is minimum and chlorides follow it.

V. SCATTER PLOT on PCA FEATURES

In this step, firstly I separated quality data as target data and concat both data sets, then shuffle data with target data. Secondly, apply PCA on this data with extracting the most impostant and two least important features and visualize their results with target data which is show which sample belong to true wine class.

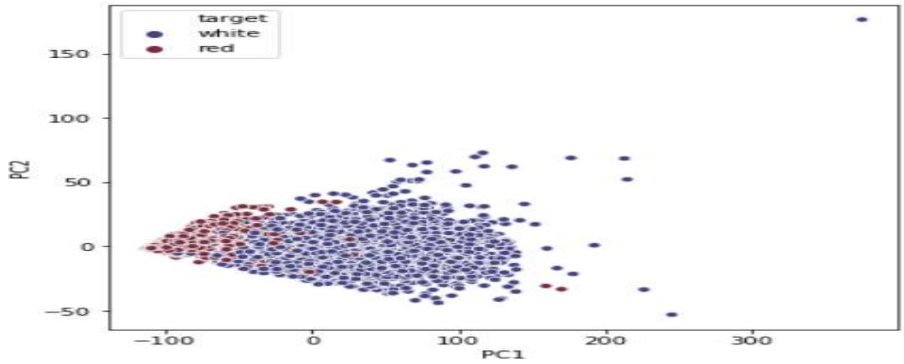

Figure 16: Two most important features that are PC1 and PC2 features scatter plots.

Using the most important features that are PC1 and PC2, we see in above figure, red and white wine samples are more separable with each other and we will obtain good result when it will be used for a classification task.

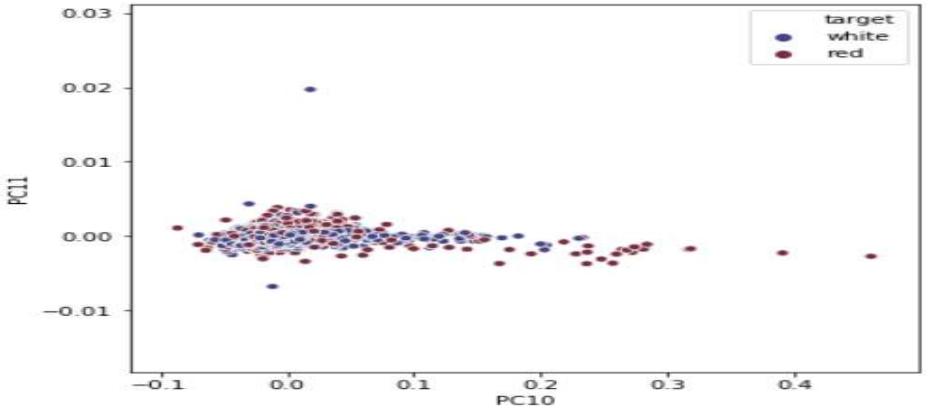

Figure 17: Two least important features that are PC10 and PC11 features scatter plots.

On the other hand, using the least important features that are PC10 and PC11, we see in above figure, red and white wine samples are intersection with each other and we will not good result when it will be used for a classification task.

VI. K-MEANS, K-CENCTERS, FARTHEST-FIRST, DBSCAN, SOM ALGORITHMS and THEIR RESULTS

In here, I first implemented algorithms on original data and PCA two most important features data, then compare results and visualize their results on two dimensional scatter plots.

A. K-MEANS

K-means algorithm not coded from stretch instead used sklearn K-Means algorithm. Quality column removed from both red-wine and white-wine data and using this target quality data for comparing algorithm scores.

K-means first applied on original data via number cluster selected maximum quality value.

Quality sample sizes for each cluster for actual red-wine data quality shown in following graph.

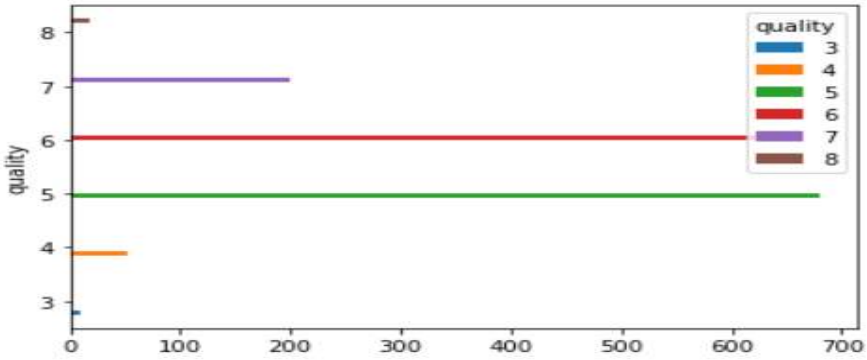

Figure 18: k-means results on original red-wine data with **actual** clusters with shown with bar plots.

Quality sample sizes for each cluster for prediction red-wine data quality shown in following graph.

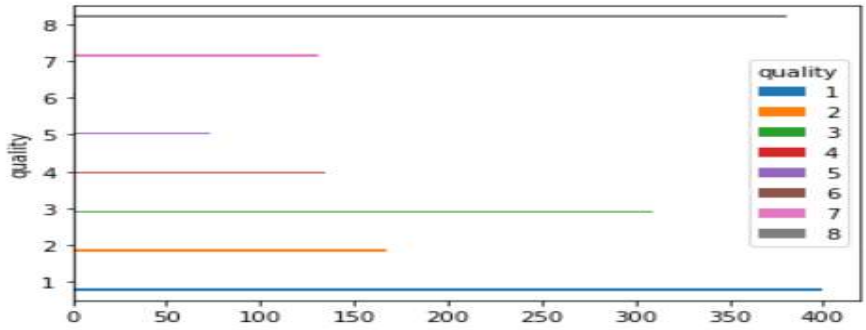

Figure 19: k-means results on original red-wine data with **predicted** clusters with shown with bar plots.

In red-wine data for each quality sample sizes are shown as above graphs and it has total 6 cluster while in our case we choose max quality number which is 8 as cluster numbers. So actually results are not good when we select max quality as cluster number.

So we choose the same operation via give 6 as cluster number. Quality sample sizes for each cluster for prediction in red-wine data quality shown in following graph.

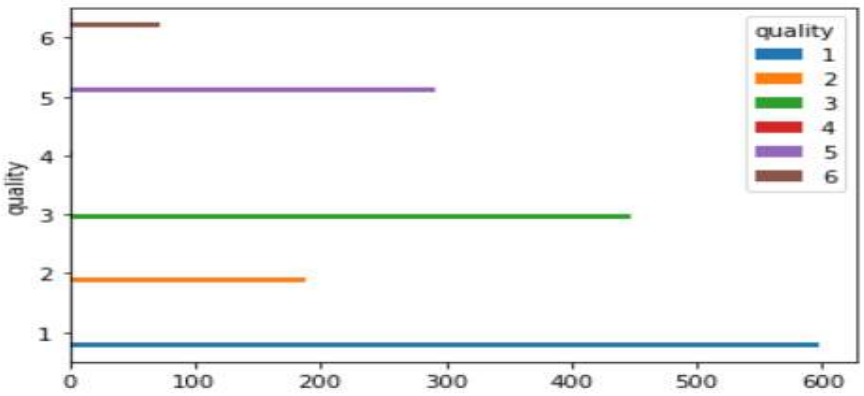

Figure 20: k-means results on original red-wine data with **predicted** clusters with shown with bar plots(number cluster the same with original data).

As we see, when we select number cluster with same actual cluster size our result are better. We also plot for each sample actual and predicted clusters.

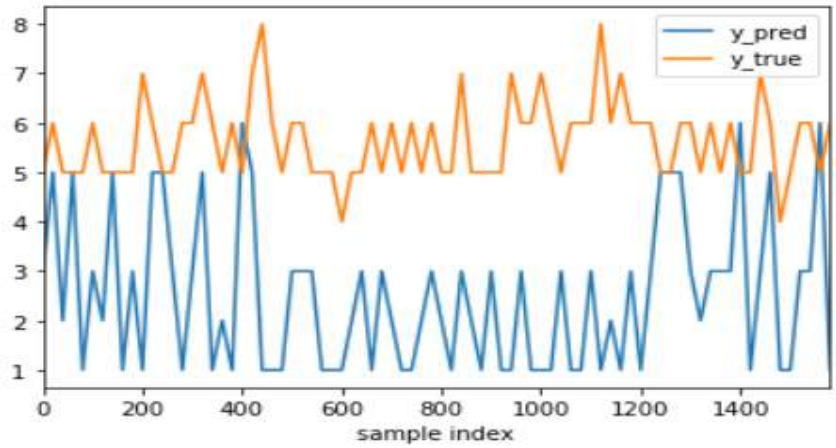

Figure 21: k-means results on original red wine data with **predicted** clusters **and actual** clusters (number cluster the same with original data).

I show every 20 point prediction cluster and real cluster value as graphically. Their values actual parallel, so result we cay cluster results are similar.

Note: cluster number don't importance for us, we pay attention on distribution on sample sizes on each clusters)

The same operation not implemented for white wine data set. Instead of this, other experiments applied on two important PCA results features to visualize results.

a. Red-wine data set results

It's applied k-means on two most important features data with cluster number as max quality number.

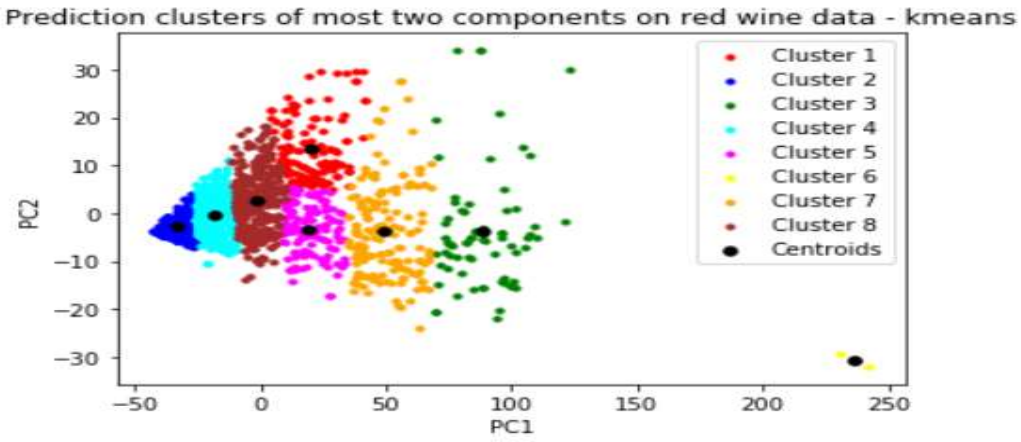

Figure 22: k-means results on two most important features in red wine data.

Each cluster data are within each other but although cluster results not bad. As we see in right botttom figure only two points seperated as different clusters, actually those are much more similar outliers, but kmeans assign them as different cluster. This is not good.

We repeat the same operation for slecting cluster number as 6 that is our real cluster numbers and compare visualize results.

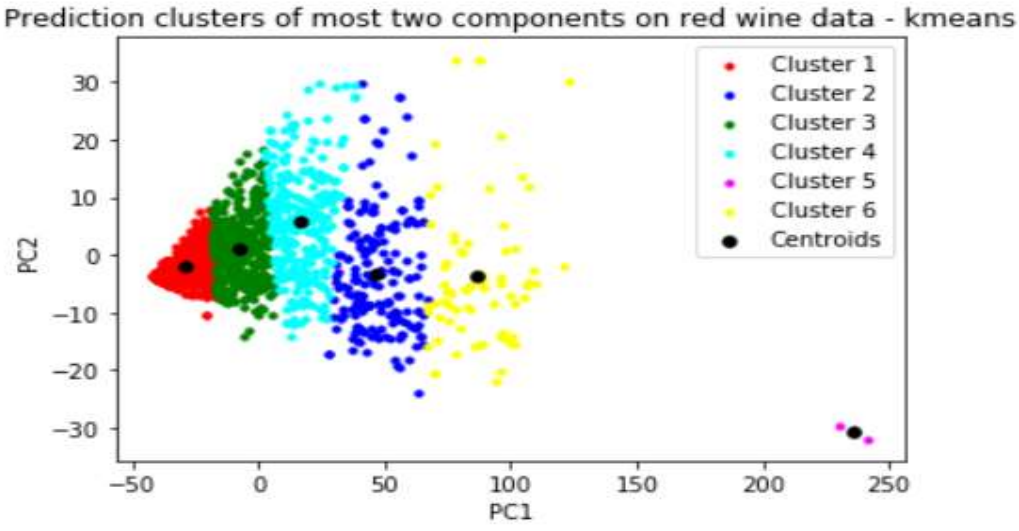

Figure 23: k-means **prediction** results on two most important features in red wine data(n cluster is the same with original data).

I visualized correct cluster scatter plots as following graph and compare our prediction results.

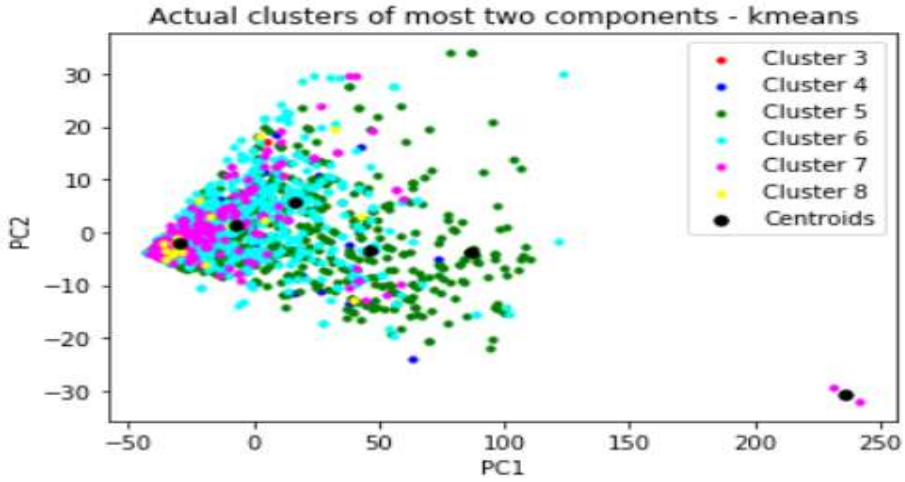

Figure 24: k-means **actual** results on two most important features in red wine data(n cluster is the same with original data).

As we see actual clusters samples and our predicted assigned results are different clusters for two most important features data.

The same operation applied on white-wine data set and results are given in sub section.

b. White-wine data set results

It's apply k-means on two most important features data with cluster number as max quality number.

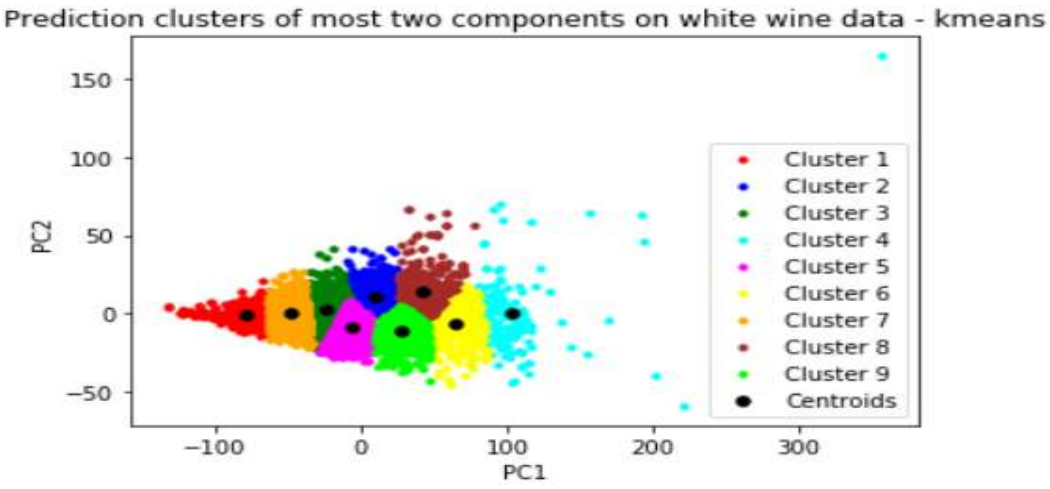

Figure 25: k-means results on two most important features in white wine data.

Each cluster data are within each other but in here outlier points not separated as different cluster. This is not good.

We repeat the same operation for selecting cluster number as 7 that is our real cluster numbers and compare visualize results.

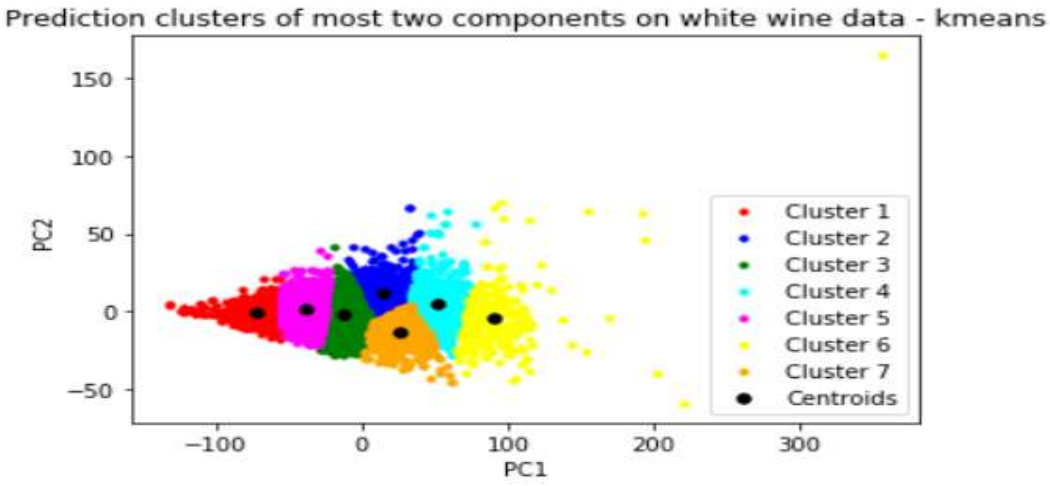

Figure 26: k-means **prediction** results on two most important features in white wine data(n cluster is the same with original data).

I visualized correct cluster scatter plots as following graph and compare our prediction results.

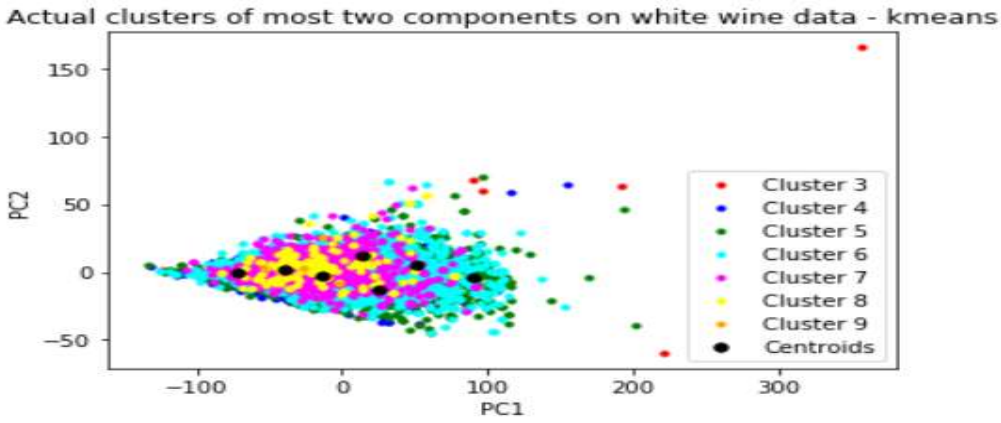

Figure 27: k-means **actual** results on two most important features in white wine data(n cluster is the same with original data).

As we see actual clusters samples and our predicted assigned results are different clusters for two most important features data.

So, using only two most features data is actually comparing with using original data features not better via only looking visualization. Using all data features give better results.

B. DB-SCAN

For DBSCAN algorithm, we also used sklearn library DBSCAN algorithm. Select epsilon as 3 and min sample point as 2.

a. Red-wine data set results

First I applied DBSCAN algorithm on PCA red wine data set and visualize the most two importance features data.

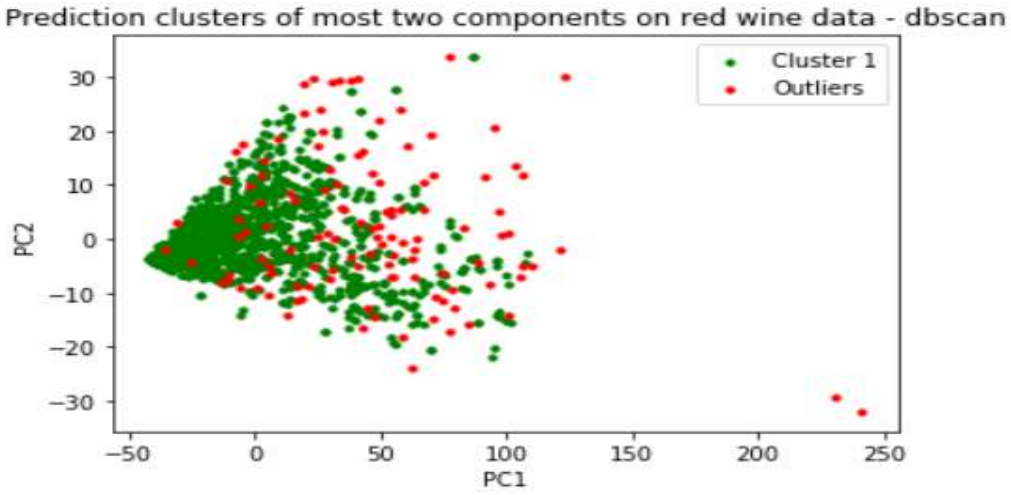

Figure 28: db-scan results on two most important features in red wine data

Using all pca data actually not not seperated outliers from data. Then, dbscan algorithm applied only two most important features data.

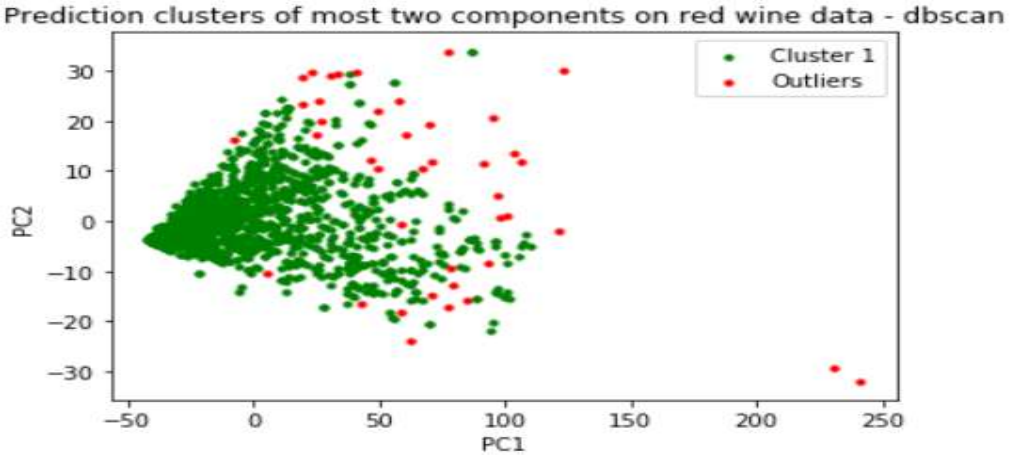

Figure 29: db-scan results on **only** two most important features in red wine data

Using two most important features data give more better results with separated outliers from data.

b. White-wine data set results

First it's applied DBSCAN algorithm on PCA white wine data set and visualize the most two importance features data.

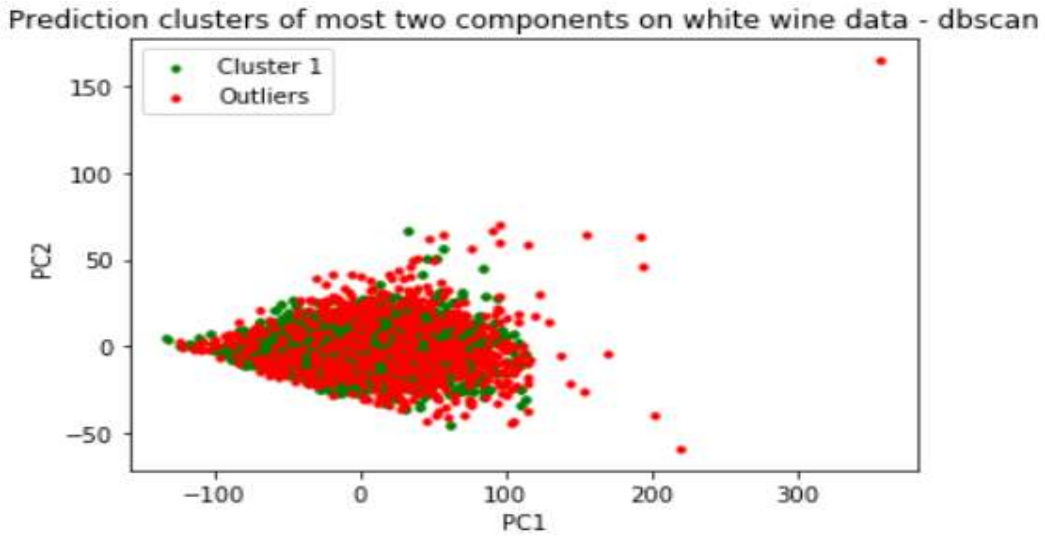

Figure 30: db-scan results on two most important features in white wine data

Using all PCA data actually not separated outliers from data. Then, DBSCAN algorithm applied only two most important features data.

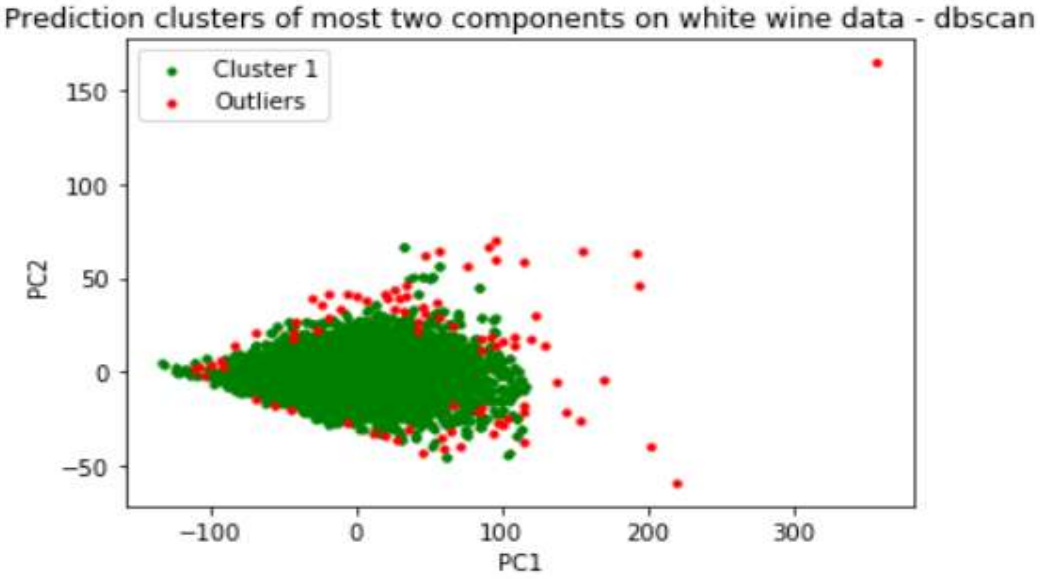

Figure 31: db-scan results on **only** two most important features in white wine data.

Using two most important features data give more better results with separated outliers from data.

C. K-CENTERS and FARTHEST-FIRST ALGORITHMS

K-centers and Farthest-first algorithms are coded from stretch and implemented according to pseudocode as specified following. Those algorithms are applied on two most important features data, not applied on original data because of take long time.

1. K-centers Algorithm

- X original data set, X={x1, x2,..xn}
- X' -> X.copy()
- w -> keep centers our data set
- r -> radious (e.g = 3)
- k -> number cluster(e.g: k= 5)
- while True:
    - X' = X.copy()
    - while X' is not empty:
        - pick a data in X'(randomly) and add it to w
        - delete all the data in X' that are within r from the center point we picked
    - if length(w) == k, break
    - elif length(w) > k, increase r
    - else length(w) < k, decrease r
- then, w attempt each sample to closest clusters

a. Red-wine data

Radious 20 selected and increased or decreased with with each step with 0.05. Cluster number also selected max quality value. Euclidian distance used. Optimal radius value find as 26.75 and total in 135 iteration.

	PC1	PC2
166	53.677901	-9.837014
37	-16.694337	0.791151
213	20.734959	7.762324
637	99.443307	-6.866180
57	69.489264	19.406115
584	41.044938	29.771530
1079	230.674270	-29.496431
1244	123.163740	30.060277

Table 8: Our centers samples are selected as above with 166.th, 37.th, .. 1244.th samples

After find ours centers as clusters, find closest cluster for each sample in data via using euclidian distance and visualize them.

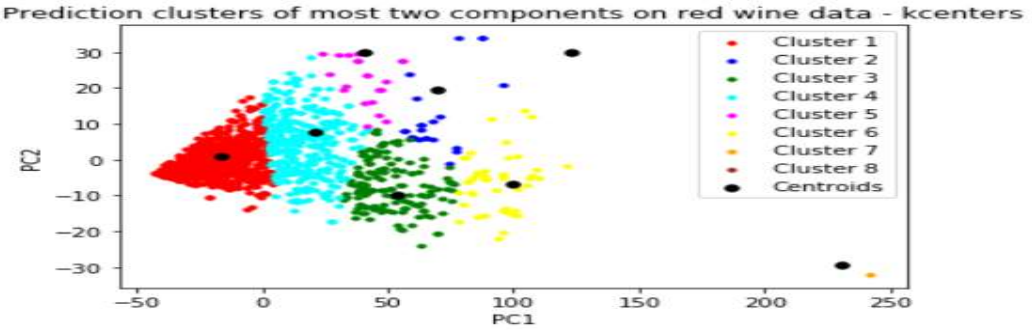

Figure 32: k-center results on **only** two most important features in red wine data.

Actually k-center not very good for some clusters but for four cluster are results not bad (Cluster 1, 4, 3, 6).

b. White-wine data

Radius 35 selected and increased or decreased with each step with 0.05. Cluster number also selected max quality value. Euclidian distance used. Optimal radius value find as 51.14 and total in 323 iteration.

	PC1	PC2
4353	26.050899	-8.597210
2679	-28.936518	1.361518
3620	47.521267	62.318509
909	78.932296	-10.265299
1739	-88.808231	-8.388744
387	129.582633	13.561049
2127	201.979678	-39.470989
4745	357.007255	165.435709
3050	155.621256	64.365738

Table 9: Our centers samples are selected as above with 4353.th, 2679.th, .. 3050.th samples

After find ours centers as clusters, find closest cluster for each sample in data via using euclidian distance and visualize them.

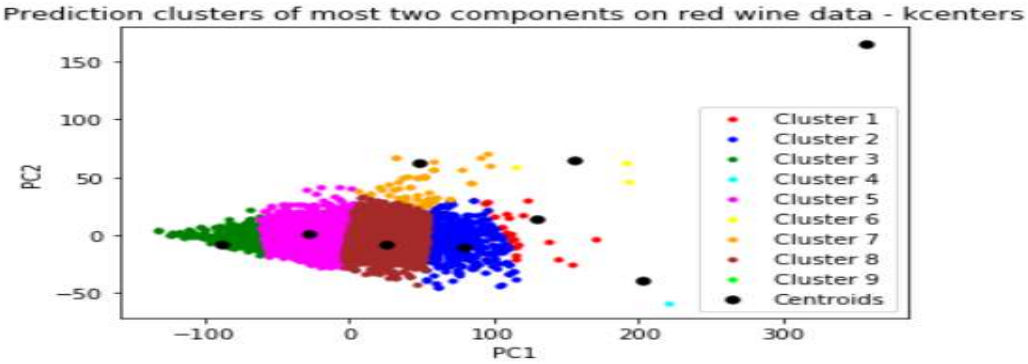

Figure 33: k-center results on **only** two most important features in white wine data.

Actually k-center good for almost all clusters, but as visually.

2. Farthest-First algorithm

Farthest-first algorithm implemented according to following pseudocode from stretch:

- pick any z€X and set w={z}
- while |w|<k:
  - z = argmax d(X, wi) - x€X, wi€we
  - w = w u {z}
- then, w attempt each sample to closest clusters

a. Red-wine data

Experiments implemented on two most important features data. Cluster number also selected max quality value. Euclidian distance used. In 8 iteration clusters found.

	PC1	PC2
1109	21.470626	6.514522
1081	241.406732	-31.904517
979	-42.334773	-3.804261
1079	230.674270	-29.496431
915	-42.315410	-3.686442
1244	123.163740	30.060277
984	-42.334773	-3.804261
354	121.032963	-1.870806
813	-41.108197	-2.897030

Table 10: Our centers samples are selected as above with 1109.th, 1081.th, .. 979.th samples

After find ours centers as clusters, find closest cluster for each sample in data via using euclidian distance and visualize them.

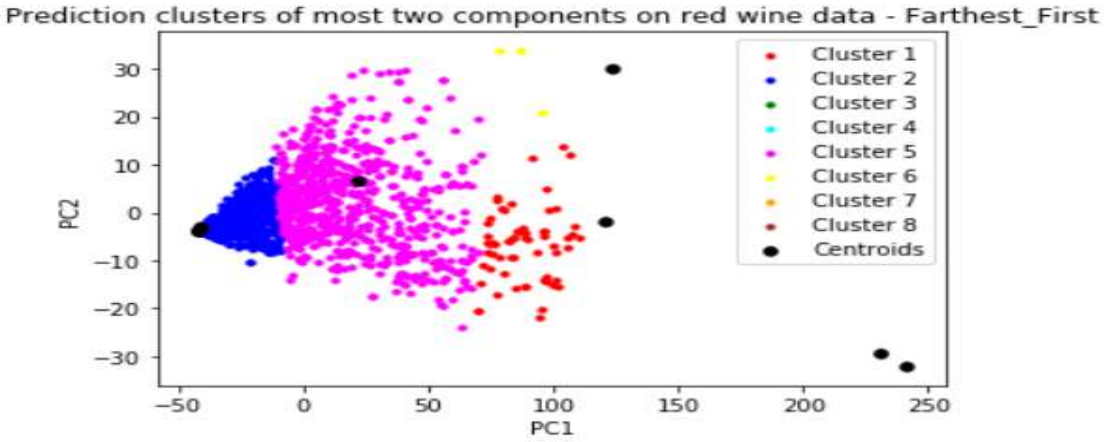

Figure 34: farthest-first results on **only** two most important features in red wine data.

Actually results not good for some clusters but for one is not bad (Cluster 5).

b. White-wine data

Experiments implemented on two most important features data. Cluster number also selected max quality value. Euclidian distance used.

	PC1	PC2
2762	16.359582	-32.894881
4745	357.007255	165.435709
3710	-132.919391	4.708558
1417	220.374280	-59.631938
3901	-132.211260	3.481095
1931	191.931279	62.761039
3094	-124.530983	0.404444
2127	201.979678	-39.470989
3095	-124.530983	0.404444

Table 11: Our centers samples are selected as above with 2762.th, 4745.th, .. 3095.th samples

After find ours centers as clusters, find closest cluster for each sample in data via using euclidian distance and visualize them.

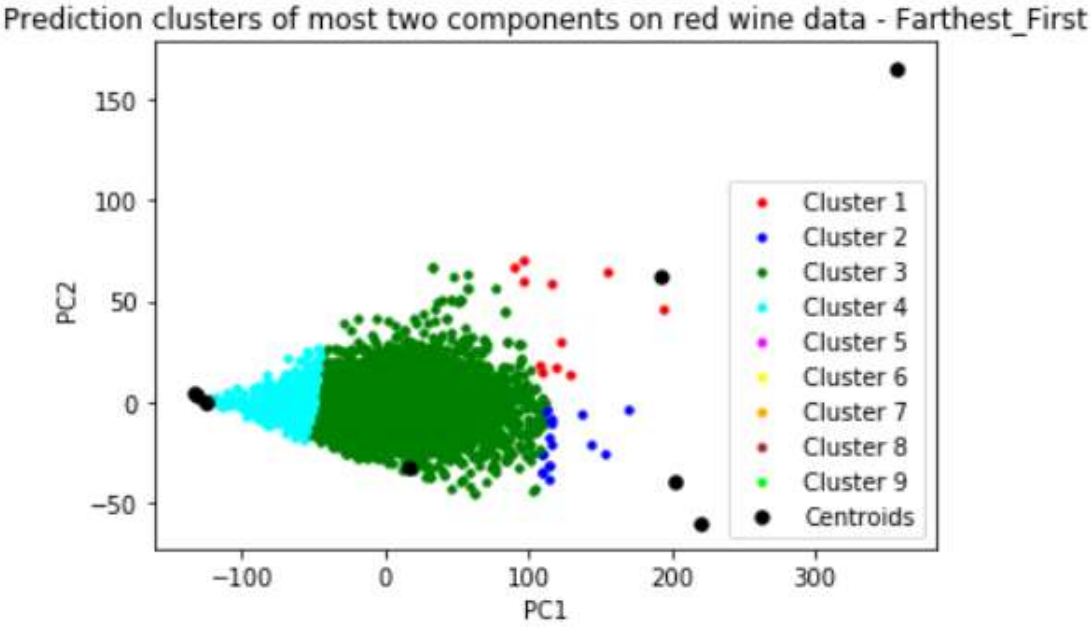

Figure 34: farthest-first results on **only** two most important features in white wine data.

Cluster center are not find close middle clusters centers, so results are not good.

D. SELF ORGANIZING MAP ALGORITHM

SOM algorithm implemented MiniSom package. This algorithm applied on original datasets.

a. Red-wine data

Learning rate taken as 0.5 and space size 30x30, number iteration is 10000. Last error rate is : 1.5

I visualize my results how to assign each sample data in new two dimension space and analysis it.

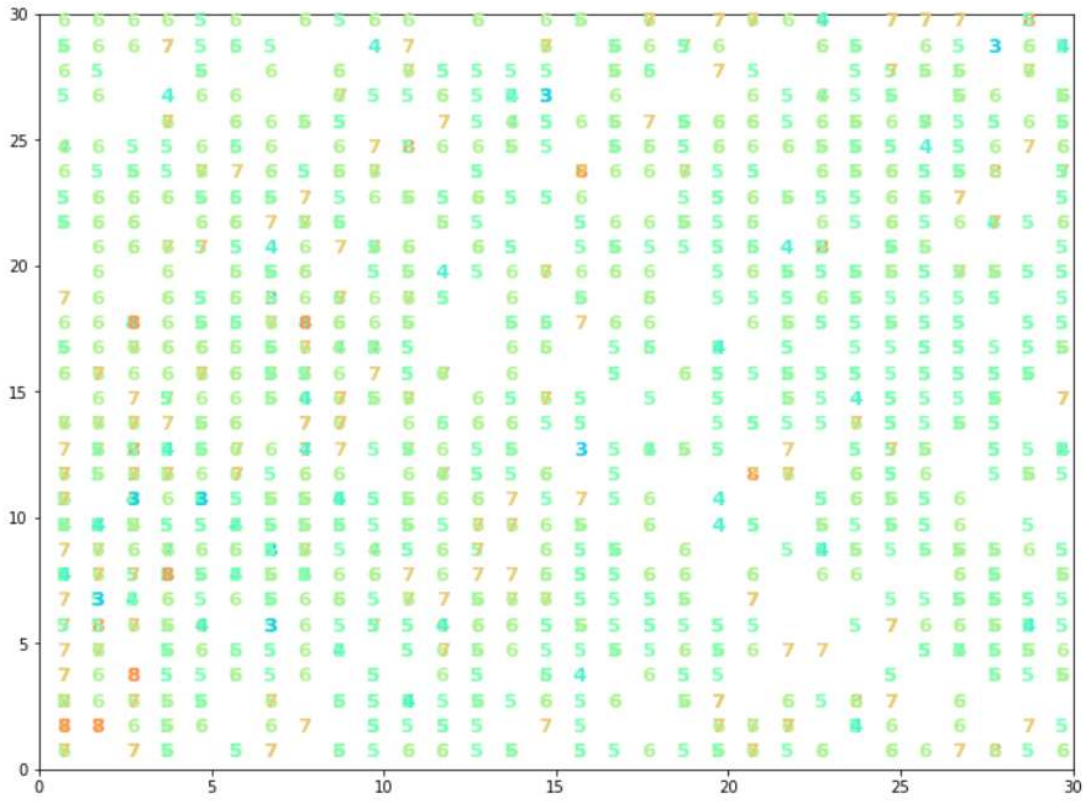

Figure 35: SOM results on red wine data.

As we see, 3, 4, 5 clusters distributed but for cluster 6 we have some good results via collected in same space.

b. White-wine data

Learning rate taken as 0.5 and space size 30x30, number iteration is 10000. Last error rate is: 3.08

I visualize my results how to assign each sample data in new two dimension space and analysis it.

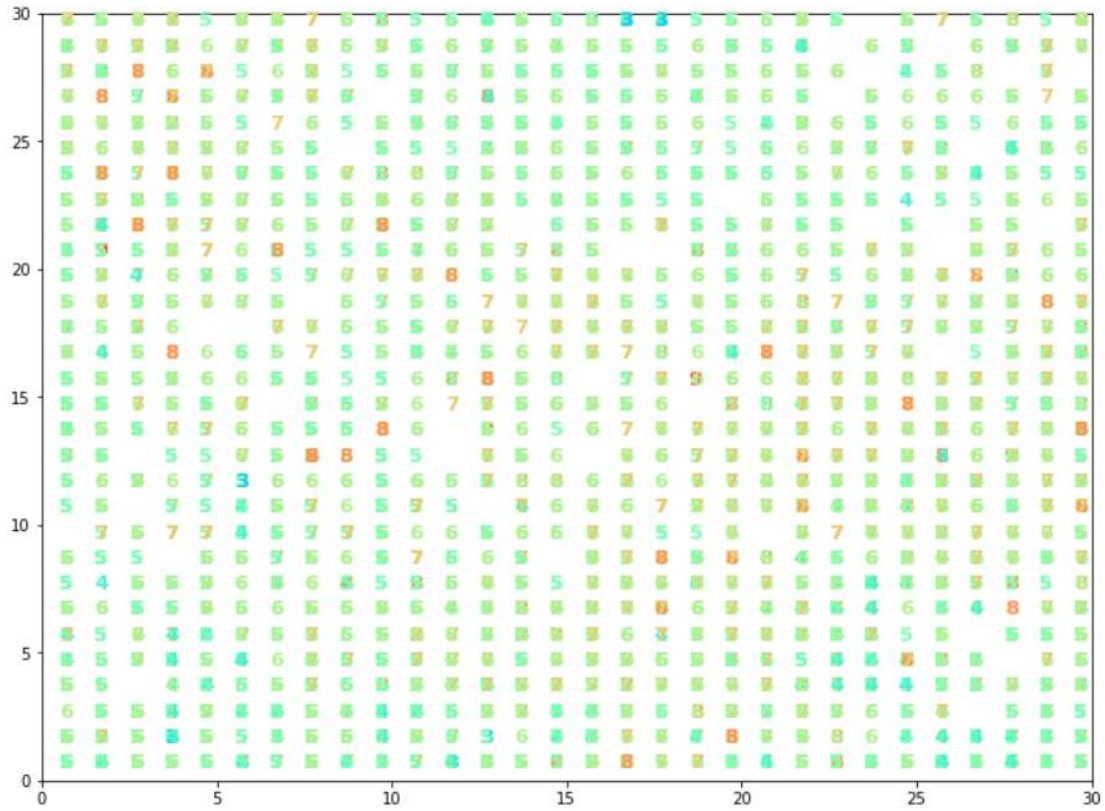

Figure 36: SOM results on white wine data.

As we see, 3, 7, 8 clusters distributed but for cluster 5 and 6 we have some good results via collected in same space.

REFERENCES

https://realpython.com/k-means-clustering-python/

https://medium.com/machine-learning-algorithms-from-scratch/k-means-clustering-from-scratch-in-python-1675d38eee42

https://github.com/JustGlowing/minisom/blob/master/examples/HandwrittenDigits.ipynb

https://stackoverflow.com/questions/22984335/recovering-features-names-of-explained-variance-ratio-in-pca-with-sklearn