



UNIVERSITÀ  
DEGLI STUDI  
DI MILANO

---

# WINE QUALITY PREDICTION USING SUPERVISED LEARNING TECHNIQUES

---

Statistical Learning, Deep Learning and  
Artificial Intelligence

Sai Spandana Adivishnu (03180A)

A.A. 2022-2023

# 1 Abstract

*In this Case Study, we will use the K-Means Clustering Algorithm to accomplish one of the most important applications of machine learning on wine categorization. R Studio will be used to implement wine classification. The main goal is to identify the various types of wines in the data set based on their chemical components. Then we will investigate the data on which our clustering model will be built. In addition, in this Statistics Learning project, we will look at descriptive data analysis and then develop multiple variations of the K-means algorithm. Furthermore, by collecting data, we can acquire a better understanding of wine clustering. Finally, we discover patterns and linkages, acquire insights into distinctive characteristics and flavors, and identify contamination issues that may have an impact on wine quality and authenticity.*

**Key Factors : Kmean Clustering Algorithm**

# Contents

<b>1</b>	<b>Abstract</b>	<b>2</b>
<b>2</b>	<b>Introduction</b>	<b>4</b>
<b>3</b>	<b>Data Exploration and Data Cleaning</b>	<b>5</b>
3.1	Data Insights . . . . .	5
<b>4</b>	<b>Exploratory Data Analysis and Feature Engineering</b>	<b>7</b>
4.1	Univariate EDA . . . . .	7
4.1.1	Visualization . . . . .	8
4.2	Data Scaling . . . . .	8
4.2.1	Data Summary . . . . .	9
4.2.2	Visualization . . . . .	10
4.3	Multivariate EDA . . . . .	12
4.3.1	Visualization . . . . .	12
<b>5</b>	<b>Model Building</b>	<b>13</b>
5.1	PCA and Feature Exploration . . . . .	13
5.1.1	Proportion of Variance Explained . . . . .	14
5.1.2	Cumulative proportion of variance explained . . . . .	15
5.2	K-means Clustering . . . . .	16
<b>6</b>	<b>Model Tuning</b>	<b>17</b>
6.1	The Elbow Method . . . . .	17
6.2	Average Silhouette Method . . . . .	18
6.3	Gap Statistic Method . . . . .	19
6.4	Majority rule . . . . .	20
<b>7</b>	<b>Final Model</b>	<b>22</b>
7.1	Aggregate table by cluster . . . . .	22
<b>8</b>	<b>Feature reduction</b>	<b>23</b>
8.1	Kmeans clustering based on Reduced Features . . . . .	24
8.2	Visualization . . . . .	24
8.3	Aggregate table . . . . .	24
<b>9</b>	<b>Conclusion</b>	<b>26</b>
<b>10</b>	<b>APPENDIX</b>	<b>27</b>
10.1	GitHub Link . . . . .	27
10.2	R- Code . . . . .	27
<b>11</b>	<b>References</b>	<b>32</b>

## 2 Introduction



Figure 1: Wine Classification

Our main objective is to perform clustering on a dataset of wines grown in the same region in Italy but derived from three different cultivars. The dataset includes the results of a chemical analysis that determined the quantities of 13 constituents in each type of wine. The attributes include alcohol, malic acid, ash, alcalinity of ash, magnesium, total phenols, flavanoids, nonflavanoid phenols, proanthocyanins, color intensity, hue, OD280/OD315 of diluted wines, and proline.

By clustering the data based on the chemical compounds present in each wine, we aim to identify the distinct types of wines present in the dataset. This analysis will help characterize and differentiate the wines based on their chemical composition. Different cultivars, regions, and winemaking techniques can result in variations in the concentration of certain compounds, which contribute to the unique characteristics and flavors of different wines.

Performing clustering on the dataset will provide insights into the relationships and similarities between the wines, allowing us to group them into clusters based on the levels of specific chemical compounds. This analysis can reveal patterns and associations that may not be immediately apparent and help us understand the composition and qualities of each wine type.

To achieve our objective, we propose utilizing K-means clustering, a popular algorithm for grouping similar data points together. By implementing K-means clustering in R, we can analyze the dataset and determine the number of distinct wine types present based on the levels of the 13 chemical constituents. This approach will provide a comprehensive understanding of the dataset and facilitate further analysis and interpretation of the wine samples.

### 3 Data Exploration and Data Cleaning

This data is collected from the kaggle. This data consists of 178 observations and 14 variables with the information about chemical compounds such as

1. Alcohol
2. Malic\_Acid
3. Ash
4. Ash\_Alcanity
5. Magnesium
6. Total\_Phenols
7. Flavanoids
8. Nonflavanoid\_Phenols
9. Proanthocyanins
10. Color\_Intensity
11. Hue
12. OD280
13. Proline

#### 3.1 Data Insights

Here is the imported wine dataset.

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	Hue	OD280	Proline
1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065
2	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050
3	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185
4	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480
5	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735

Figure 2: Data Insights

We observed that

- Every variable is of numeric(double) data types.
- There are 178 observations of 13 variables.

Here is the observation is neither infinity, NA, Null or NaN values in any variable in the whole dataset.

```

> str(wine_clustering)
'data.frame': 178 obs. of 13 variables:
 $ Alcohol      : num 14.2 13.2 13.2 14.4 13.2 ...
 $ Malic_Acid   : num 1.71 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 ...
 $ Ash          : num 2.43 2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 ...
 $ Ash_Alcanity : num 15.6 11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 ...
 $ Magnesium    : int 127 100 101 113 118 112 96 121 97 98 ...
 $ Total_Phenols : num 2.8 2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 ...
 $ Flavanoids   : num 3.06 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 ...
 $ Nonflavanoid_Phenols: num 0.28 0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 ...
 $ Proanthocyanins : num 2.29 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 ...
 $ Color_Intensity : num 5.64 4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 ...
 $ Hue          : num 1.04 1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 ...
 $ OD280        : num 3.92 3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 ...
 $ Proline      : int 1065 1050 1185 1480 735 1450 1290 1295 1045 1045 ...
> #Checking for the null values
> apply(wine_clustering, 2, function(x)sum(is.na(x)))
      Alcohol      Malic_Acid      Ash      Ash_Alcanity      Magnesium
      0          0          0          0          0
Total_Phenols      Flavanoids Nonflavanoid_Phenols      Proanthocyanins      Color_Intensity
      0          0          0          0          0
      Hue      OD280      Proline
      0          0          0

```

Figure 3: Structure of Data

The dataset consists of wine characteristics, including various attributes such as alcohol percentage, Malic Acid, Ash, Ash Alcanity, Magnesium, Total Phenols, Flavanoids, Nonflavanoid Phenols, Proanthocyanins, Color Intensity, Hue, OD280, and Proline. The alcohol percentage ranges from 11% to 14%, while Malic Acid varies from 0.7 to 5.8. Ash content ranges between 1.3 and 3.2, and Ash Alcanity ranges from 10.6 to 30. Magnesium levels vary from 70 to 162. Total Phenols range from 0.9 to 3.8, Flavanoids from 0.3 to 5.8, Nonflavanoid Phenols from 0.1 to 3.5, Proanthocyanins from 0.4 to 3.5, Color Intensity from 1.2 to 13, Hue from 0.4 to 1.7, OD280 from 1.2 to 4, and Proline from 278 to 1680. Notably, the dataset contains no null values, indicating that all attribute values are present.

```

> summary(wine_clustering)
  Alcohol      Malic_Acid      Ash      Ash_Alcanity      Magnesium      Total_Phenols
Min.   :11.03   Min.   :0.740   Min.   :1.360   Min.   :10.60   Min.   : 70.00   Min.   :0.980
1st Qu.:12.36   1st Qu.:1.603   1st Qu.:2.210   1st Qu.:17.20   1st Qu.: 88.00   1st Qu.:1.742
Median :13.05   Median :1.865   Median :2.360   Median :19.50   Median : 98.00   Median :2.355
Mean   :13.00   Mean   :2.336   Mean   :2.367   Mean   :19.49   Mean   : 99.74   Mean   :2.295
3rd Qu.:13.68   3rd Qu.:3.083   3rd Qu.:2.558   3rd Qu.:21.50   3rd Qu.:107.00   3rd Qu.:2.800
Max.   :14.83   Max.   :5.800   Max.   :3.230   Max.   :30.00   Max.   :162.00   Max.   :3.880
  Flavanoids Nonflavanoid_Phenols Proanthocyanins Color_Intensity      Hue      OD280
Min.   :0.340   Min.   :0.1300   Min.   :0.410   Min.   : 1.280   Min.   :0.4800   Min.   :1.270
1st Qu.:1.205   1st Qu.:0.2700   1st Qu.:1.250   1st Qu.: 3.220   1st Qu.:0.7825   1st Qu.:1.938
Median :2.135   Median :0.3400   Median :1.555   Median : 4.690   Median :0.9650   Median :2.780
Mean   :2.029   Mean   :0.3619   Mean   :1.591   Mean   : 5.058   Mean   :0.9574   Mean   :2.612
3rd Qu.:2.875   3rd Qu.:0.4375   3rd Qu.:1.950   3rd Qu.: 6.200   3rd Qu.:1.1200   3rd Qu.:3.170
Max.   :5.080   Max.   :0.6600   Max.   :3.580   Max.   :13.000   Max.   :1.7100   Max.   :4.000
  Proline
Min.   : 278.0
1st Qu.: 500.5
Median : 673.5
Mean   : 746.9
3rd Qu.: 985.0
Max.   :1680.0

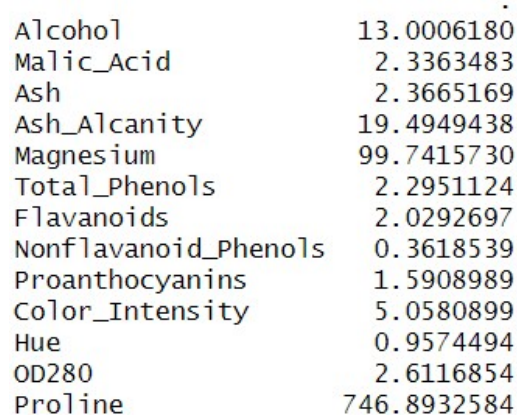
```

Figure 4: Summary of Data

## 4 Exploratory Data Analysis and Feature Engineering

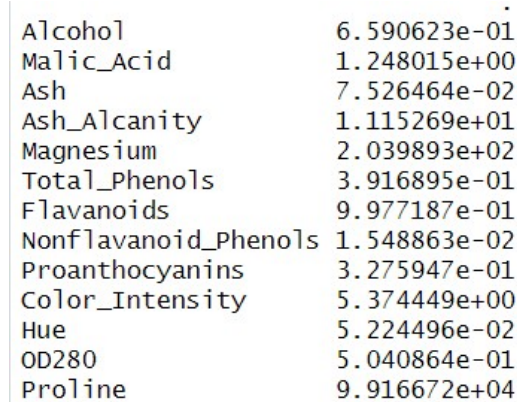
### 4.1 Univariate EDA

In the univariate exploratory data analysis (EDA) phase, we begin by summarizing the dataset and examining the mean and variance of each feature. This provides us with an initial understanding of the distribution and variability within each variable.



Alcohol	13.0006180
Malic_Acid	2.3363483
Ash	2.3665169
Ash_Alcanity	19.4949438
Magnesium	99.7415730
Total_Phenols	2.2951124
Flavanoids	2.0292697
Nonflavanoid_Phenols	0.3618539
Proanthocyanins	1.5908989
Color_Intensity	5.0580899
Hue	0.9574494
OD280	2.6116854
Proline	746.8932584

Figure 5: Mean



Alcohol	6.590623e-01
Malic_Acid	1.248015e+00
Ash	7.526464e-02
Ash_Alcanity	1.115269e+01
Magnesium	2.039893e+02
Total_Phenols	3.916895e-01
Flavanoids	9.977187e-01
Nonflavanoid_Phenols	1.548863e-02
Proanthocyanins	3.275947e-01
Color_Intensity	5.374449e+00
Hue	5.224496e-02
OD280	5.040864e-01
Proline	9.916672e+04

Figure 6: Variance

### 4.1.1 Visualization

To further analyze the dataset, we boxplot the features. This visual representation allows us to identify any outliers, investigate the spread of the data, and assess if the features are uniformly scaled or not.

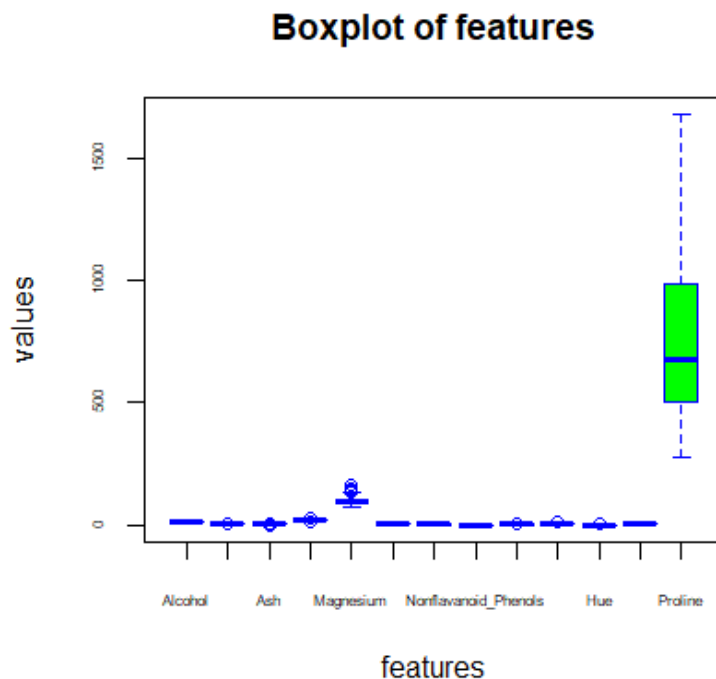


Figure 7: BoxPlot of Original Dataset

We observe that the features are not well scaled to apply clustering algorithm. Hence we scale the features such that mean of each features becomes 0 and standard deviation becomes 1.

## 4.2 Data Scaling

Upon observation, we found that the features are not well-scaled or have varying ranges, it is necessary to apply feature scaling techniques to ensure fair comparison and accurate clustering results. One common scaling method is standardization, where we scale the features such that the mean of each feature becomes 0 and the standard deviation becomes 1. This transformation helps in aligning the scales of different features, making them more comparable and suitable for clustering algorithms.

$$Z = \frac{(x - \mu)}{\sigma}$$



#### 4.2.1 Data Summary

Now the features are well scaled and ready to compare to be used for further analysis. Hence we scaled the features such that mean of each features becomes 0 and the standard deviation becomes 1.

Alcohol	1.625085e-17
Malic_Acid	-5.171888e-18
Ash	-9.152372e-18
Ash_Alcanity	-1.783025e-17
Magnesium	-2.450343e-17
Total_Phenols	-1.085040e-17
Flavanoids	1.493519e-17
Nonflavanoid_Phenols	-2.892383e-17
Proanthocyanins	6.377911e-18
Color_Intensity	1.899790e-17
Hue	-5.271172e-18
OD280	2.783354e-17
Proline	-3.003940e-17

Figure 8: Mean Scaled

Alcohol	1
Malic_Acid	1
Ash	1
Ash_Alcanity	1
Magnesium	1
Total_Phenols	1
Flavanoids	1
Nonflavanoid_Phenols	1
Proanthocyanins	1
Color_Intensity	1
Hue	1
OD280	1
Proline	1

Figure 9: Variance Scaled

#### 4.2.2 Visualization

The resulting boxplot provides insights into the distribution, spread and presence of outliers in the scaled data, Each box represents the interquartile range(IQR)

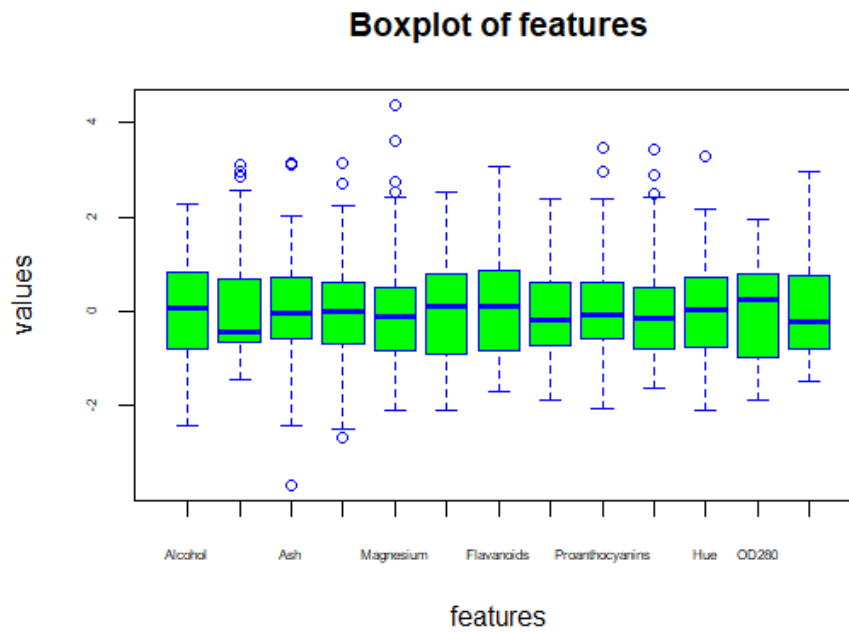


Figure 10: Boxplot Scaled

Now the features are well scaled and ready to be compared and to be used whenever required.

**Histogram with Density Plots** This analysis reveals that several attributes can be clustered into two distinct groups (in our Keyfindings it is classified into 2 or 3 clusters). The clustering helps in understanding the potential differentiation among wines based on these attributes, providing insights into their chemical composition and properties. By observing the plot we can identify the following

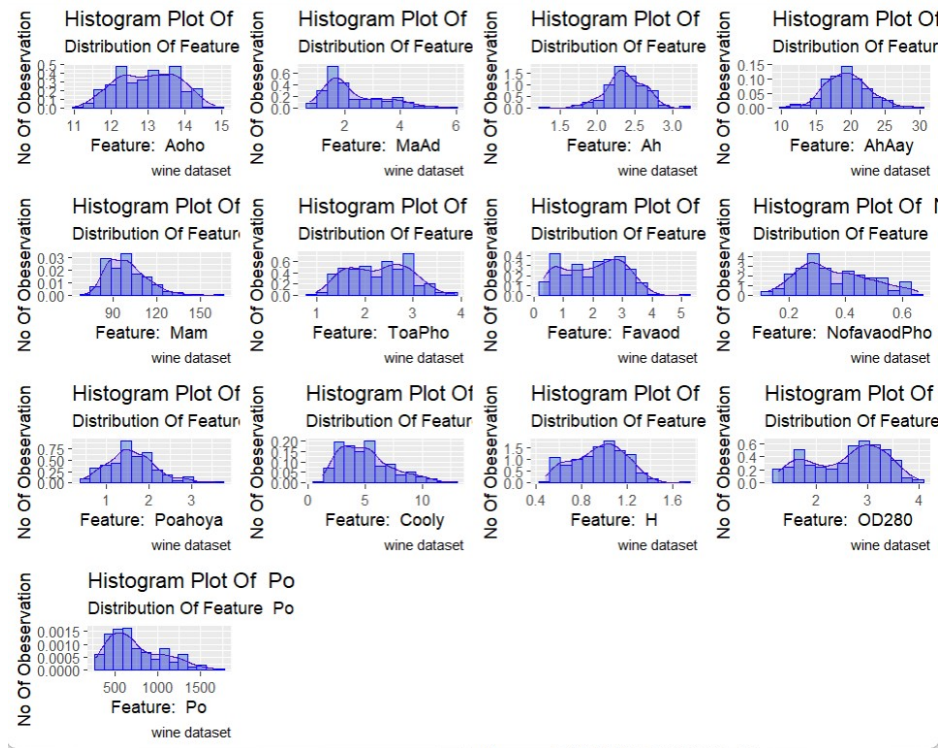


Figure 11: Histogram with density plots

#### KEYFINDINGS:

- Alcohol, Total Phenol, Hue, OD280 seems to be clustered in 2 clusters.
- Flavanoids, Nonflavonoids phenols, Poranthocyanins, Colour Intensity, Magnesium, Proline seems to be clustered into 3 clusters.
- Alcalinity of ash seems to be normally distributed.

### 4.3 Multivariate EDA

Performing multivariate analysis before applying models is crucial as it helps reveal relationships between variables, account for confounding factors, assess variable importance, identify interactions, validate assumptions, and reduce dimensionality. By comprehending these aspects of the data, models can be constructed with accuracy and reliability, effectively capturing the underlying dynamics of the system being studied.

I want to start multivariate EDA with correlation and the table look like this.

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	Hue	OD280	Proline
Alcohol	1.000000	0.0943969	0.2115446	-0.3102331	0.2707982	0.2891011	0.2368149	-0.1559295	0.1366979	0.5463642	-0.0717472	0.0723432	0.6437200
Malic_Acid	0.0943969	1.000000	0.160455	0.2685004	-0.0545751	-0.3351670	-0.4110066	0.2929771	-0.2207462	0.2489853	-0.5612957	-0.3687104	-0.1920106
Ash	0.2115446	0.160455	1.000000	0.4433672	0.2865867	0.1289795	0.1150773	0.1862304	0.0096519	0.2588873	-0.0746669	0.0039112	0.2236263
Ash_Alcanity	-0.3102331	0.2685004	0.4433672	1.000000	-0.0833331	-0.3211133	-0.3513699	0.1619217	-0.1973268	0.0187320	-0.2739552	-0.2767685	-0.4405969
Magnesium	0.2707982	-0.0545751	0.2865867	-0.0833331	1.000000	0.2144012	0.1957838	-0.2562940	0.2364406	0.1999500	0.0553982	0.0660039	0.3933508
Total_Phenols	0.2891011	-0.3351670	0.1289795	-0.3211133	0.2144012	1.000000	0.8645635	-0.4499353	0.6124131	0.0551364	0.4336813	0.6994944	0.4981149
Flavanoids	0.2368149	-0.4110066	0.1150773	-0.3513699	0.1957838	0.8645635	1.000000	-0.5378996	0.6526918	-0.1723794	0.3434786	0.7871939	0.4941931
Nonflavanoid_Phenols	-0.1559295	0.2929771	0.1862304	0.1619217	-0.2562940	-0.4499353	-0.5378996	1.000000	-0.3658451	0.1390570	-0.2626396	-0.5032696	-0.3113852
Proanthocyanins	0.1366979	-0.2207462	0.0096519	-0.1973268	0.2364406	0.6124131	0.6526918	-0.3658451	1.000000	-0.0252499	0.2955443	0.5190671	0.3304167
Color_Intensity	0.5463642	0.2489853	0.2588873	0.0187320	0.1999500	-0.0551364	-0.1723794	0.1390570	-0.0252499	1.000000	-0.5218132	0.1000000	0.1616001
Hue	-0.0717472	-0.5612957	-0.0746669	-0.2739552	0.0553982	0.4336813	0.3434786	-0.2626396	0.2955443	-0.5218132	1.000000	0.5654683	0.2361834
OD280	0.0723432	-0.3687104	0.0039112	-0.2767685	0.0660039	0.6994944	0.7871939	-0.5032696	0.5190671	-0.4288149	0.5654683	1.000000	0.3127611
Proline	0.6437200	-0.1920106	0.2236263	-0.4405969	0.3933508	0.4981149	0.4941931	-0.3113852	0.3304167	0.1616001	0.2361834	0.3127611	1.000000

Figure 12: Correlation Plot

#### 4.3.1 Visualization

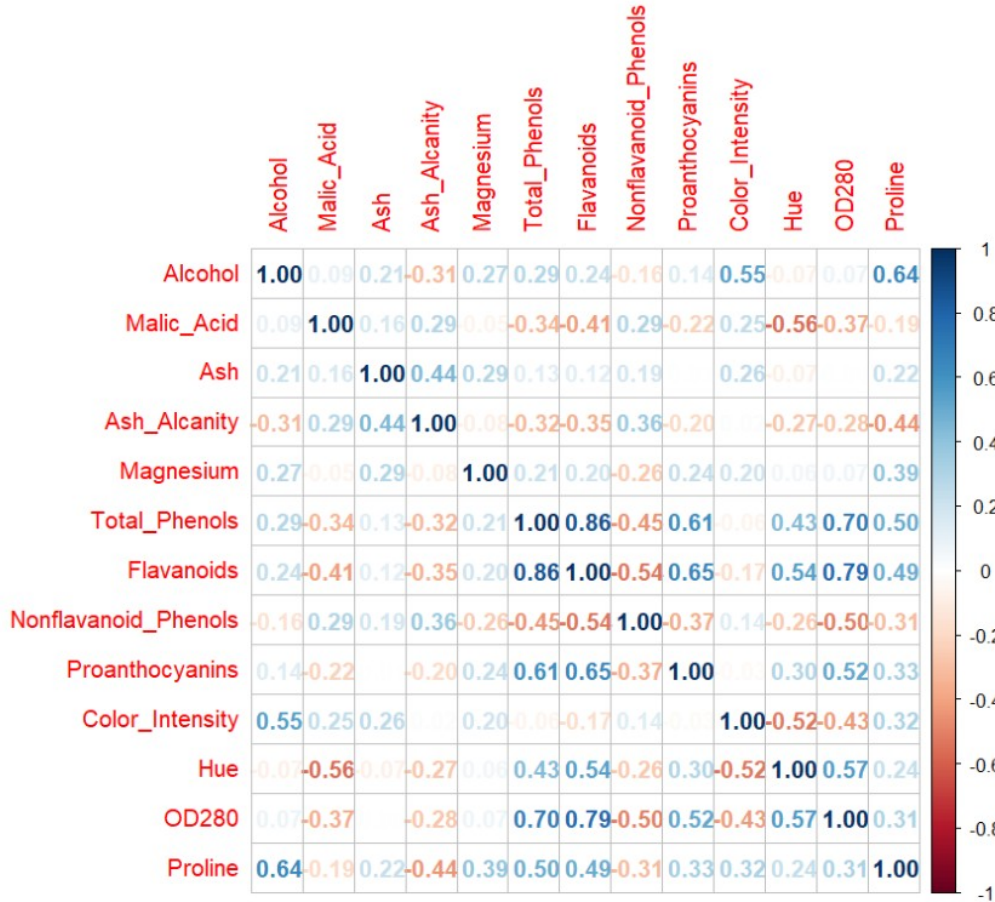


Figure 13: Correlation Plot

## KEYFINDINGS:

Based on the table and plots, several observations can be made:

- There appears to be a positive correlation between Alcohol and Proline.
- Alcohol and Color Intensity also show a positive correlation.
- Malic Acid and Hue exhibit a negative correlation.
- There are many more correlated features which gives overall idea of how the features are correlated with each other

These findings highlight the relationships between the variables, indicating whether they tend to increase or decrease together.

## 5 Model Building

### 5.1 PCA and Feature Exploration

Principal Component Analysis (PCA) is a widely used technique in multivariate analysis for reducing the dimensionality of data while retaining most of the relevant information. It accomplishes this by transforming the original variables into a new set of uncorrelated variables called principal components. PCA identifies the directions of maximum variance in the data and projects the data onto these directions. The first principal component captures the most significant variation in the data, followed by subsequent components in descending order of importance.

$$\text{Explained Variance Ratio} = (\text{Eigenvalue} / \Sigma \text{Eigenvalues}) * 100$$

To perform PCA, the data is first standardized to eliminate any scaling issues. Next, the covariance matrix of the standardized data is computed, and its eigenvectors and eigenvalues are calculated. The eigenvectors represent the directions along which the data varies the most, while the eigenvalues indicate the amount of variance explained by each eigenvector. By selecting a subset of the eigenvectors with the highest eigenvalues, one can obtain the principal components. The transformed data is then obtained by projecting the original data onto these principal components.

PCA offers several advantages, including dimensionality reduction, interpretation of complex data, and noise reduction. It enables the visualization and analysis of high-dimensional data by condensing the information into a lower-dimensional space. It not only simplifies the subsequent modeling process but also aids in identifying the most relevant variables driving the patterns in the data. Additionally, PCA can help in noise reduction by focusing on the principal components that capture the most significant variation and filtering out the noise present in the data.

Importance of components:													
	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13
Standard deviation	2.169	1.5802	1.2025	0.95863	0.92370	0.80103	0.74231	0.59034	0.53748	0.5009	0.47517	0.41082	0.32152
Proportion of Variance	0.362	0.1921	0.1112	0.07069	0.06563	0.04936	0.04239	0.02681	0.02222	0.0193	0.01737	0.01298	0.00795
Cumulative Proportion	0.362	0.5541	0.6653	0.73599	0.80162	0.85098	0.89337	0.92018	0.94240	0.9617	0.97907	0.99205	1.00000

Figure 14: Summary of PCA

## KEYFINDINGS:

- The standard deviation measures the amount of variation or spread captured by each PC. Larger standard deviations indicate that the corresponding PC explains more variability in the original data. In this model, PC1 has the highest standard deviation of 2.169, followed by PC2 with 1.5802, and so on.
- The proportion of variance signifies the relative importance of each PC in capturing the variability in the dataset. It represents the proportion of total variance explained by each PC. For instance, PC1 explains 36.2% of the total variance, PC2 explains 19.2%, PC3 explains 11.1%, and so on.
- The cumulative proportion, on the other hand, gives the accumulated proportion of variance explained by each PC, up to that specific component. It helps determine how much of the total variance is captured when considering multiple components. In this example, PC1 alone explains 36.2% of the variance, PC1 and PC2 combined explain 55.41%, PC1 to PC3 explain 66.53%, and so on.

### 5.1.1 Proportion of Variance Explained

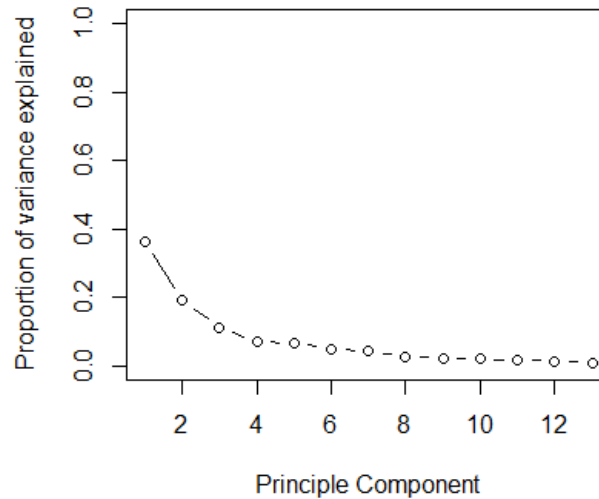


Figure 15: Proportion of variance explained

By sorting the eigenvalues in descending order and dividing each eigenvalue by the total sum of eigenvalues, the proportion of variance explained by each component can be determined. This proportion, ranging from 0 to 1, indicates the importance of each component

in capturing the dataset's variability. The cumulative proportion of variance explained can also be calculated by summing up the proportions, giving insight into the overall amount of variance captured by a certain number of principal components.

### 5.1.2 Cumulative proportion of variance explained

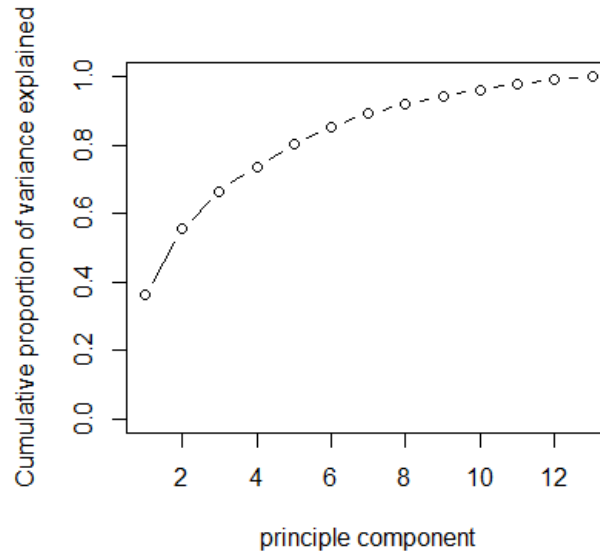


Figure 16: Cumulative proportion of variance explained

Cumulative proportion of variance explained shows the accumulated amount of variance captured by the included principal components. It helps determine how many principal components are needed to represent a desired level of variability in the dataset. Typically, a higher cumulative proportion indicates that fewer principal components are required to explain a significant portion of the variance, while a lower cumulative proportion may suggest the need for more components to capture a satisfactory level of variability.

#### KEYFINDINGS:

We can observe the cumulative proportion of variability explained by each principle components. PC1 to PC9 explains the 0.9424 proportion of variability of the data.

Now we will see the clustering tendency by visualising the principle components.

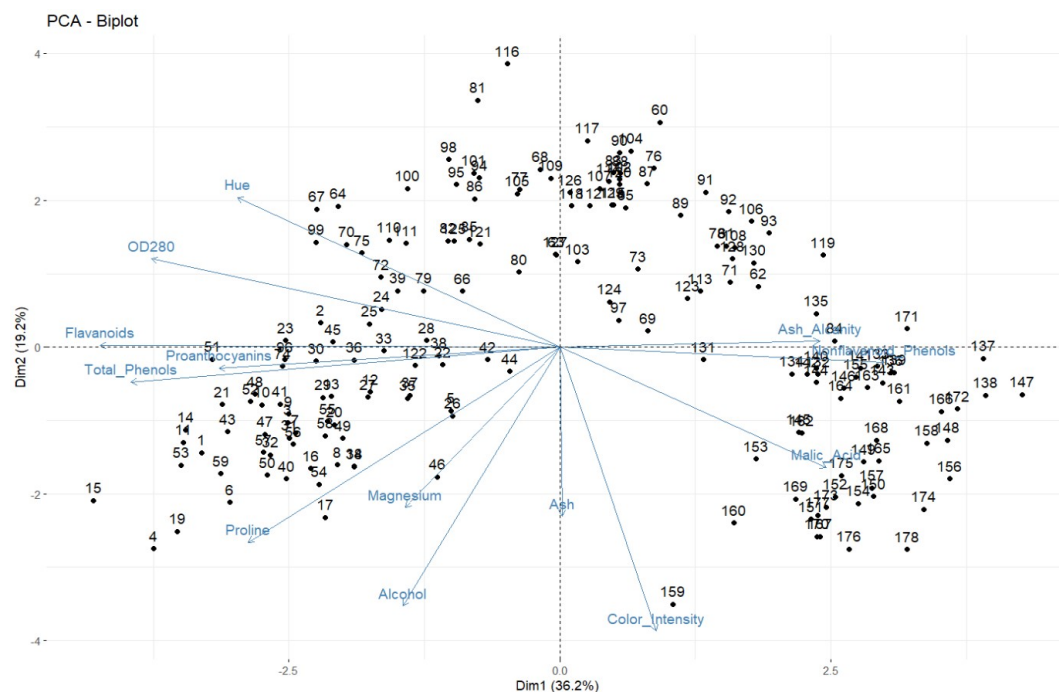


Figure 17: PCA- Biplot

## KEYFINDINGS:

- Alcohol, Color intensity, Proline, Magnesium, and Ash seem to contribute to one cluster.
- while Hue, Total phenols, Flavanoids, and OD280/OD315 of diluted wines contribute to another cluster.
- Additionally, Malic acid, Nonflavanoid phenols and Alcalinity of ash contribute to yet another cluster.

## 5.2 K-means Clustering

K-means clustering is an iterative algorithm used for partitioning a dataset into K distinct and non-overlapping clusters. The algorithm aims to minimize the sum of squared distances between data points and their respective cluster centroids. It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid is at the minimum. The less variation we have within clusters, the more homogeneous the data points are within the same cluster.



The diagram shows the objective function for K-means clustering:  $J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2$ . Annotations include:
 

- An arrow from "number of clusters" to the upper limit  $k$  of the first summation.
- An arrow from "number of cases" to the upper limit  $n$  of the second summation.
- An arrow from "case  $i$ " to the term  $x_i^{(j)}$ .
- An arrow from "centroid for cluster  $j$ " to the term  $c_j$ .
- An arrow from "Distance function" to the norm  $\|x_i^{(j)} - c_j\|^2$ .
- An arrow from "objective function" to the variable  $J$ .

Figure 18: Kmeans Clustering

The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.
- Hence each cluster has datapoints with some commonalities, and it is away from other clusters.

## 6 Model Tuning

### 6.1 The Elbow Method

The elbow method is a technique used to determine the optimal number of clusters (K) in K-means clustering. It involves plotting the sum of squared distances (SSD) between data points and their cluster centroids for different values of K and identifying the "elbow" point in the plot. The elbow point represents a significant drop in SSD, indicating the number of clusters that provides a good balance between compactness within clusters and separation between clusters.

$$WCSS = \sum_{i \in n} (X_i - Y_i)^2$$

Figure 19: Elbow Method

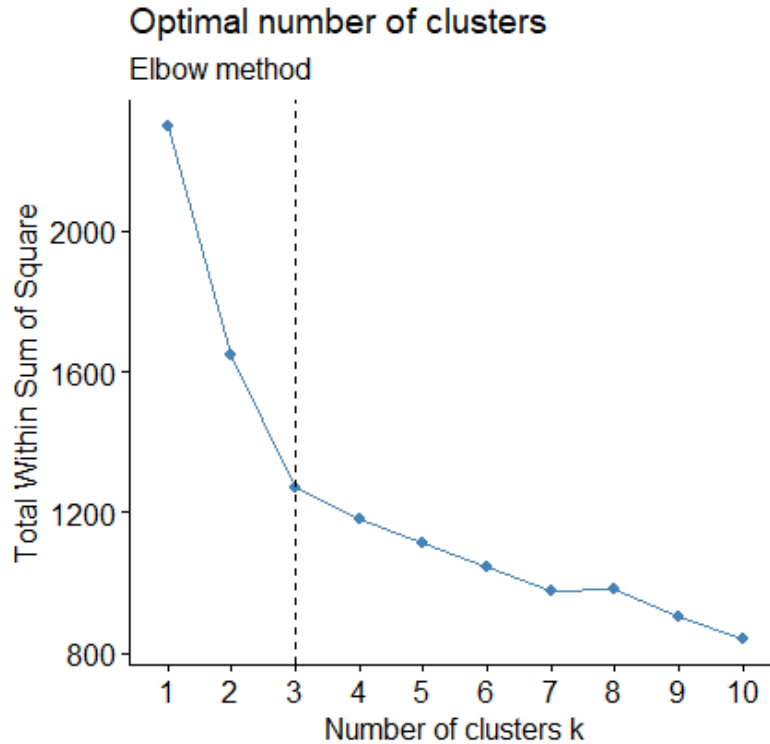


Figure 20: Elbow Method

Given the above plot, one could say that clusters 3, so I decided to use 3 clusters, as this number would provide the most compact clusters, and thus the most detailed information about the wines. Choosing to have fewer clusters would result in looser associations between datapoints, minimizing the amount of meaningful takeaways that can be deduced.

**Observation:** We can see elbow at  $k = 3$ .

## 6.2 Average Silhouette Method

The Average Silhouette Method is a technique used to evaluate the quality of clustering in K-means clustering. It provides a measure of how well data points fit within their assigned clusters and helps determine the optimal number of clusters (K). The method calculates the average silhouette coefficient for each K value and identifies the K with the highest average silhouette score as the optimal number of clusters.

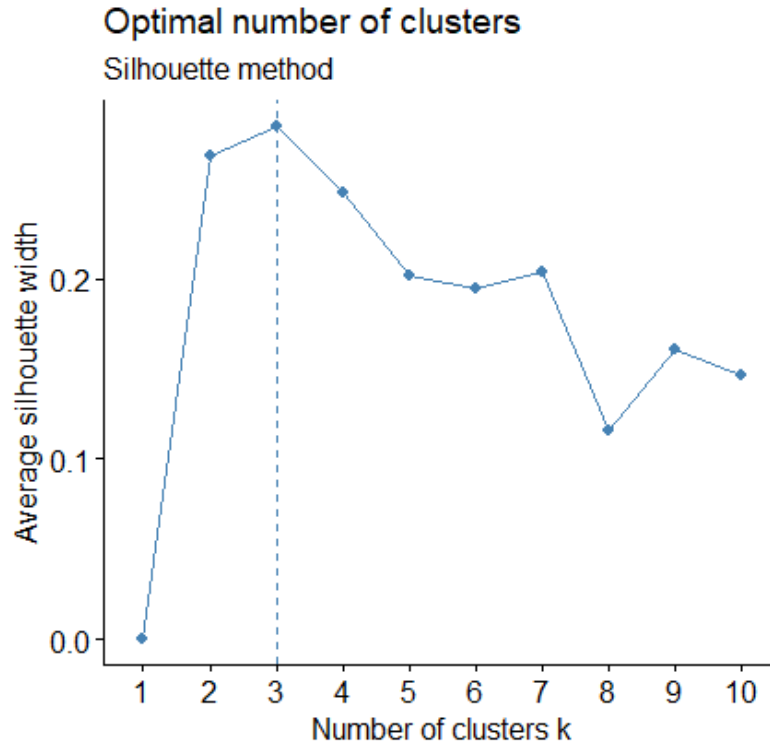


Figure 21: Silhouette Method

Silhouette analysis allows you to calculate how similar each observation is with the cluster it is assigned relative to other clusters. This metric ranges from -1 to 1 for each observation in your data and can be interpreted as a poor fit (-1), a loose fit that is borderline between clusters (0), and a great fit (1). Maximizing the silhouette metric is the goal, and should yield the optimal amount of clusters.

### 6.3 Gap Statistic Method

The Gap Statistic method is a technique used to estimate the optimal number of clusters (K) in K-means clustering by comparing the within-cluster dispersion of the data with that of randomly generated reference datasets. The method helps identify the K value that maximizes the gap between the expected dispersion in the reference datasets and the observed dispersion in the actual data.

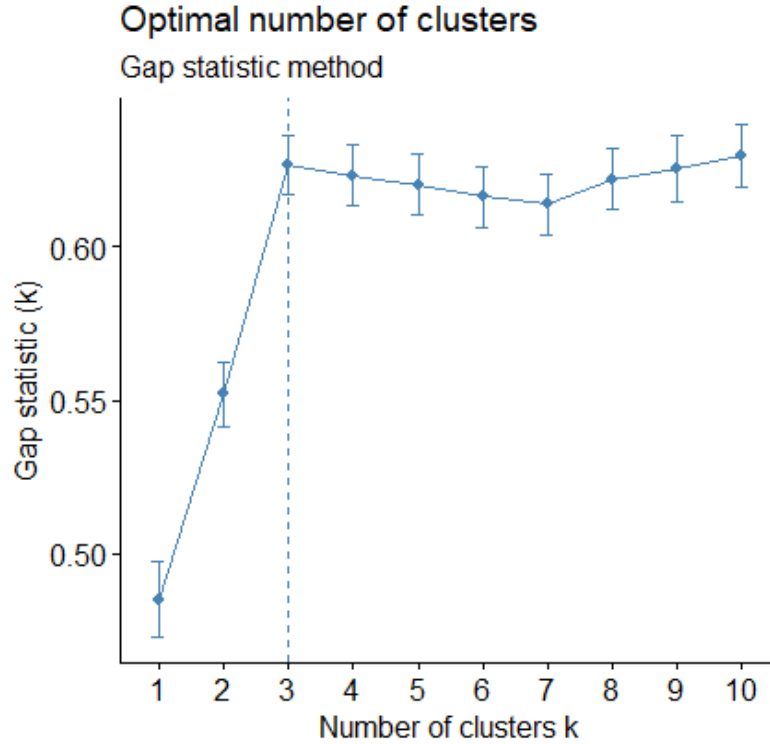


Figure 22: Gap statistic Method

For computing the gap statistics method we can utilize the `clusGap` function for providing gap statistic as well as standard error for a given output.

## 6.4 Majority rule

The concept of "Majority rule" is applied within the `NbClust()` method. It involves assessing multiple clustering indices to identify the number of clusters that achieves the greatest consensus among them. By considering several criteria simultaneously, the Majority rule provides a more robust and objective means of selecting the optimal number of clusters in a dataset.

The `NbClust()` uses two statistics to decide on the optimum number of cluster:-

The Hubert index:- It is a graphical method of determining the number of clusters. In the plot of Hubert index, we seek a significant knee that corresponds to a significant increase of the value of the measure i.e the significant peak in Hubert index second differences plot.

The D index:- It is a graphical method of determining the number of clusters. In the plot of D index, we seek a significant knee (the significant peak in Dindex second differences plot) that corresponds to a significant increase of the value of the measure.

```

*****
* Among all indices:
* 2 proposed 2 as the best number of clusters
* 19 proposed 3 as the best number of clusters
* 2 proposed 10 as the best number of clusters

***** Conclusion *****

* According to the majority rule, the best number of clusters is 3

*****

```

Figure 23: Majority Rule by using Nbclust()

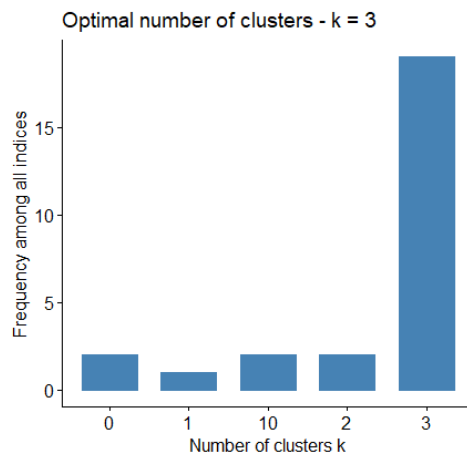


Figure 24: Number of Clusters

### KEYFINDINGS:

We saw different method to decide the optimum number of cluster for kmeans and the maximum algorithm gave k=3 as optimum number of cluster for k-means clustering.

## 7 Final Model

After the deep analysis we decided to do the k-means clustering at k=3 and the results as follows:

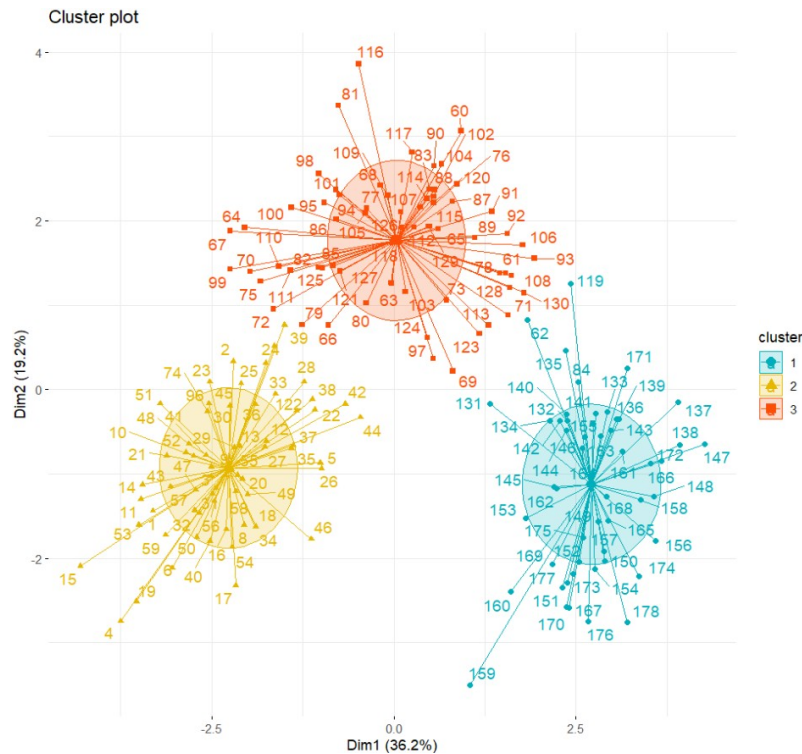


Figure 25: Clusters - 3

Observation: It looks similar to the clusters obtained by clustering with all the features. That's a great point indicating our reduced features are good.

### 7.1 Aggregate table by cluster

we use the aggregate() function to group the data by the cluster column and calculate the mean for each column in the table. The resulting aggregated\_table will have the mean values for each variable, separated by clusters.

```
> aggar.kmeans%>%kable()
```

Group.1	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	Hue	od280	Proline
1	13.13412	3.307255	2.417647	21.24118	98.66667	1.683922	0.8188235	0.4519608	1.145882	7.234706	0.6919608	1.696667	619.0588
2	13.67677	1.997903	2.466290	17.46290	107.96774	2.847581	3.0032258	0.2920968	1.922097	5.453548	1.0654839	3.163387	1100.2258
3	12.25092	1.897385	2.231231	20.06308	92.73846	2.247692	2.0500000	0.3576923	1.624154	2.973077	1.0627077	2.803385	510.1692

Figure 26: Aggregate Table

## KEYFINDINGS:

Alcohol, Malic\_Acid, Ash, etc., represent the average values of those variables within each cluster. The table provides insights into how the variables differ across the clusters.

For instance, considering the Alcohol column, the average alcohol content is 13.13412 for Cluster 1, 13.67677 for Cluster 2, and 12.25092 for Cluster 3. This indicates that alcohol content varies among the clusters, allowing us to observe the characteristic differences between them.

Similarly, you can analyze the other variables in the table. For instance, Color\_Intensity, Flavanoids, and Proline show the average values of these attributes for each cluster. This information can help identify patterns and characteristics that differentiate the clusters from each other.

## 8 Feature reduction

Feature reduction techniques can still be applied to enhance the interpretability of the clustering results. While K-means is an unsupervised learning algorithm that doesn't directly involve a target variable, feature reduction can help simplify the representation of data and facilitate the understanding of the clusters.

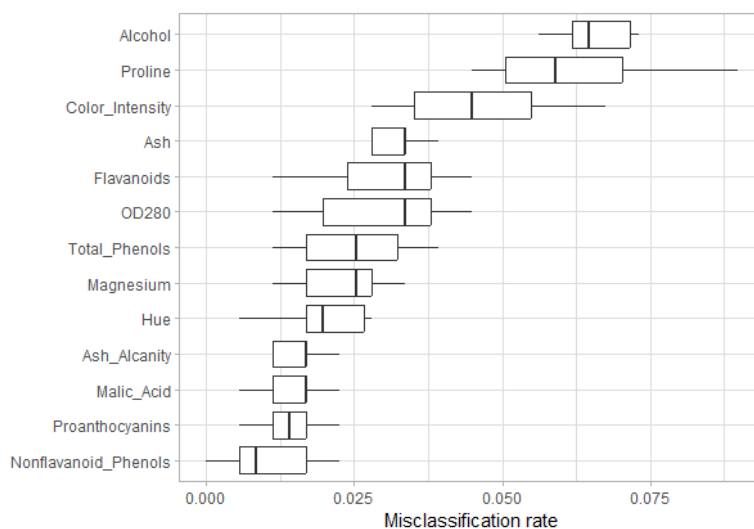


Figure 27: Misclassification of Data

## KEYFINDINGS:

We can observe highest misclassification rate is of “Alcohol” followed by “proline” and “Colour Intensity”. Hence we can finally select these features as primary features for clustering. Now lets do kmeans clustering based on these three features and compare it with kmeans model for all the features to see if its working good in terms of clustering or not.

## 8.1 Kmeans clustering based on Reduced Features

## 8.2 Visualization



Figure 28: Kmeans cluster with reduced feature

## KEYFINDINGS:

It looks similar to the clusters obtained by clustering with all the features. That's a great point indicating our reduced features are good. Lets explore it more.

## 8.3 Aggregate table

```
> aggar.reduced.kmeans %>% kable()
```

Group.1	Alcohol	Proline	Color_Intensity
1	13.29375	637.6562	8.634063
2	13.76895	1132.9298	5.585263
3	12.40315	538.9326	3.434719

Figure 29: Aggregate table with reduced feature

- Cluster 1: This cluster (labeled as Group.1 = 1) has an average alcohol content of 13.29375, an average proline value of 637.6562, and an average color intensity of 8.634063. The cluster is characterized by moderate alcohol content, medium proline values, and relatively high color intensity.



- Cluster 2: This cluster (Group.1 = 2) has an average alcohol content of 13.76895, an average proline value of 1132.9298, and an average color intensity of 5.585263. It is distinguished by higher alcohol content, higher proline values, and moderate color intensity.
- Cluster 3: This cluster (Group.1 = 3) exhibits an average alcohol content of 12.40315, an average proline value of 538.9326, and an average color intensity of 3.434719. It is characterized by lower alcohol content, lower proline values, and relatively low color intensity.

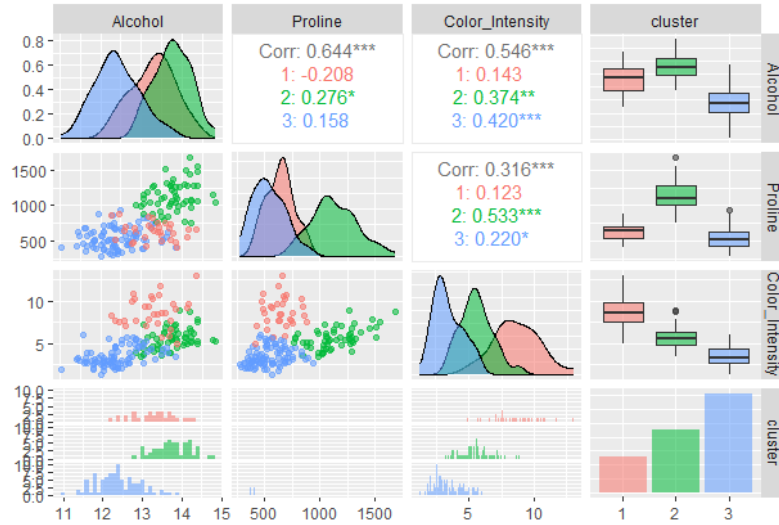


Figure 30: visualization with reduced feature

#### KEYFINDINGS:

- Cluster 1: High Alcohol, Highest Colour intensity and Low Proline.
- Cluster 2: Highest Alcohol, highest Proline and medium Colour Intensity.
- Cluster 3: Everything is lowest.

## 9 Conclusion

I explored rigorously the clustering algorithm (PCA and kmeans) for clustering the wine data set. From beginning, while doing multivariate analysis, there seemed to be three cluster in the data set and lastly we confirmed that by doing in-depth analysis. At last I selected the k-means algorithm as the best clustering algorithm for this data set with reduced features namely “Alcohol”, “Proline” and “Colour intensity”.

So, there are three clusters in the given data set which we can identify by below table:

```
> aggar.reduced.kmeans %>% kable()
```

Group.1	Alcohol	Proline	Color_Intensity
1	13.29375	637.6562	8.634063
2	13.76895	1132.9298	5.585263
3	12.40315	538.9326	3.434719

Figure 31: visualization with reduced feature

## 10 APPENDIX

### 10.1 GitHub Link

You can find R files on given GitHub link

<https://github.com/Spandanaadivishnu/Unsupervised-Learning-Project.git>

### 10.2 R- Code

```
#setting the working directory
getwd()
setwd("C:\\Users\\SaiSpandana\\OneDrive\\Desktop\\Unsupervised_Learning_Project")

#Loding packages
library(readr) #Loading the data
library(dplyr) # library for data manipulation
library(tidyr)
library(GGally)
library(gridExtra)
library(factoextra)
library(FactoMineR)
library(plotly)
library(stringr)
library(corrplot)
library(RColorBrewer)
library(knitr)
library(NbClust)
library(FeatureImpCluster)
library(flexclust)
#devtools::install_github("o1iv3r/FeatureImpCluster")

#Loding the Data
wine_clustering <- read.csv("wine_clustering.csv")
View(wine_clustering)

#Dataset Insights
colnames(wine_clustering)
#Feature of data
head(wine_clustering)

#Dimension of data
dim(wine_clustering)

#Structure of data
str(wine_clustering)
```

```

#Summary of data
summary(wine_clustering)

#Checking for the null values
apply(wine_clustering, 2, function(x)sum(is.na(x)))

#EDA (Exploratory Data Analysis) – Univariate
# mean of the features
apply(wine_clustering,2,mean) %>% as.data.frame()
# variance of the features
apply(wine_clustering,2,var)%>%as.data.frame()
#Boxplot of the data
# boxplot of features
boxplot(wine_clustering, xlab="features",ylab="values",
        main ="Boxplot_of_features", col = "green",border = "blue",cex.axis=.5)

###Standardizing the Variables (Scaling)
wine_standardized <- select(wine_clustering, c(1:13))
wine_scaled <- as.data.frame(scale(wine_standardized))
head(wine_scaled)
str(wine_scaled)
# mean of the scaled data set
apply(wine_scaled,2,mean)%>%as.data.frame()
# variance of the scaled dataset
apply(wine_scaled,2,var)%>%as.data.frame()
# boxplot of scaled features
boxplot(wine_scaled, xlab="features",ylab="values", main ="Boxplot_of_features",
        col = "green",border = "blue",cex.axis=.5)

# function to create histogram and density plot
histf<-function(z){
  feature=str_replace_all(deparse(substitute(z)),"[wine_clustering$]","")
  ggplot(wine_clustering) +
    aes(x = z) +
    geom_histogram(aes(y=..density..), position="identity",
                   alpha=0.5,bins = 14L, fill = "#497AD2", colour = "blue") +
    geom_density(alpha=0.2, fill = "#4411D2", colour = "#4411D2")+
    labs(x = paste("Feature:",feature),y = "No_Of_Observation",
         title = paste("Histogram_Plot_Of_",feature),
         subtitle = paste("Distribution_Of_Feature_",feature),
         caption = "wine_dataset") +
    theme_grey()
}

# Create a list to store the plots

```

```

plots <- list()

# calling function for different features
plots[[1]] <- histf(wine_clustering$Alcohol)
plots[[2]] <- histf(wine_clustering$Malic_Acid)
plots[[3]] <- histf(wine_clustering$Ash)
plots[[4]] <- histf(wine_clustering$Ash_Alcanity)
plots[[5]] <- histf(wine_clustering$Magnesium)
plots[[6]] <- histf(wine_clustering$Total_Phenols)
plots[[7]] <- histf(wine_clustering$Flavanoids)
plots[[8]] <- histf(wine_clustering$Nonflavanoid_Phenols)
plots[[9]] <- histf(wine_clustering$Proanthocyanins)
plots[[10]] <- histf(wine_clustering$Color_Intensity)
plots[[11]] <- histf(wine_clustering$Hue)
plots[[12]] <- histf(wine_clustering$OD280)
plots[[13]] <- histf(wine_clustering$Proline)

# Arrange and display all the plots in a single plot
grid.arrange(grobs = plots, ncol = 4)

### Deeper Exploration – Relationships between the Variables
#Multivariate EDA
#Correlation matrix
ggcorr(wine_clustering, low = "navy", high = "darkred")
Correlation <- cor(wine_clustering)
corrplot(Correlation, order = "hclust", col = brewer.pal(n=8, name = "RdBu"))
corrplot(Correlation, method = "number")
cor(wine_clustering) %>% kable()

#Scatterplot for all features
ggpairs(wine_clustering)

###PCA AND FEATURE EXPLORATION
#principle component
pca.out<-prcomp(wine_scaled)
summary(pca.out)

#Proportion of Variance Explained (PVE) by each principle components
pr_var <- pca.out$sdev^2
pve <- pr_var/sum(pr_var)
summary(pve)
# plot of PVE explained by each principle component
plot(pve, xlab="Principle_Component", ylab = "Proportion_of_variance_explained",
      ylim=c(0,1), type="b")

```

```

# Cumulative proportion of variance explained
plot(cumsum(pve), xlab="principle_component",
      ylab="Cumulative_proportion_of_variance_explained", ylim=c(0,1), type="b")

# Biplot
fviz_pca_biplot(pca.out)

###Kmeans clustering
# deciding optimal number of cluster
# Elbow method
fviz_nbclust(wine_scaled, kmeans, method = "wss")+
  geom_vline(xintercept = 3, linetype = 2)+labs(subtitle = "Elbow_method")

# Silhouette method
fviz_nbclust(wine_scaled, kmeans, method = "silhouette")+
  labs(subtitle = "Silhouette_method")

# Gap statistic
# nboot = 50 to keep the function speedy.
# recommended value: nboot= 500 for your analysis.
# Use verbose = FALSE to hide computing progression.
set.seed(123)
fviz_nbclust(wine_scaled, kmeans, nstart = 25, method = "gap_stat",
             nboot = 50)+labs(subtitle = "Gap_statistic_method")

#Optimum number of cluster by using NbClust() method
nb_clust <- NbClust(wine_scaled, distance = "euclidean", min.nc = 2,
                  max.nc = 10, method = "kmeans")
fviz_nbclust(nb_clust)

# The best number of clusters is 3
#Now fit the kmeans with 3 clusters
set.seed(123)
km <- kmeans(wine_scaled, 3, nstart = 25)

# visualising k-means result
fviz_cluster(km, data = wine_scaled,
             palette = c( "#00AFBB", "#E7B800", "#FC4E07" ),
             ellipse.type = "euclid", # Concentration ellipse
             star.plot = TRUE, # Add segments from centroids to items
             repel = TRUE, # Avoid label overplotting (slow)
             ggtheme = theme_minimal())

#Aggregate table by the cluster
# making cluster as factor

```

```

km$cluster <- as.factor(km$cluster)
# assining cluster to the original wine data set
data.clust.kmeans <- cbind(wine_clustering, cluster = km$cluster)
# aggregating the feature by cluster
aggar.kmeans <- aggregate(data.clust.kmeans[,1:13],
                           by=list(data.clust.kmeans$cluster), mean)
                           %>% as.data.frame()

aggar.kmeans%>%kable()

#visuvalizing the clustered data
ggpairs(data.clust.kmeans, aes(color=cluster, alpha=0.5),
        lower = list(combo = wrap("facethist", binwidth = 0.1)))
#Feature Reduction for better interpretability of the model
set.seed(10)
res <- kcca(wine_scaled, k=3)
FeatureImp_res <- FeatureImpCluster(res, as.data.table(wine_scaled))
plot(FeatureImp_res)

#Kmeans clustering based on reduced features
# making new data framed of reduced features
data.scaled <- as.data.frame(wine_scaled)
# data (with reduced features) containing unscaled values of feature
data.reduced <- wine_clustering[c("Alcohol", "Proline", "Color_Intensity")]
# data (with reduced feature) containing scaled fratures
data.scaled.reduced <- data.scaled[c("Alcohol", "Proline", "Color_Intensity")]
# Compute k-means for reduced features with k = 3
set.seed(123)
km_reduced <- kmeans(data.scaled.reduced, 3, nstart = 25)

#Visuvalizing the clusters of reduced features
# visualising k-means result
suppressWarnings(
  fviz_cluster(km_reduced, data = data.scaled.reduced,
               palette = c("#00AFBB", "#E7B800", "#FC4E07"),
               ellipse.type = "euclid", # Concentration ellipse
               star.plot = TRUE, # Add segments from centroids to items
               repel = TRUE, # Avoid label overplotting (slow)
               ggtheme = theme_minimal())
)

# Aggrigratee table by cluster
# making cluster as factor
km_reduced$cluster <- as.factor(km_reduced$cluster)
# assining cluster to the original wine data set
data.clust.reduced.kmeans <- cbind(data.reduced, cluster = km_reduced$cluster)

```

```

# Aggregating the clustered data (reduced feature) by cluster
aggar.reduced.kmeans <- aggregate(data.clust.reduced.kmeans[,1:3],
                                by=list(data.clust.reduced.kmeans$cluster), mean)
                                %>% as.data.frame()

aggar.reduced.kmeans %>% kable()

#Visuvalising the clustered data
suppressMessages( ggpairs(data.clust.reduced.kmeans,
                          aes(color=cluster, alpha=0.5),
                          lower = list(combo = wrap("facethist", binwidth = 0.1))))

Cluster <- c("Cluster_1", "Cluster_2", "Cluster_3")
Alcohol <- c("High", "Highest", "Lowest")
Proline <- c("Low", "Highest", "Lowest")
Colour.intensity <- c("Highest", "Medium", "Lowest")
df<-data.frame(Cluster, Alcohol, Proline, Colour.intensity)
df %>% kable()

```

## 11 References

- <http://www.sthda.com/english/articles/25-clusteranalysis-in-r-practical-guide/>
- <https://archive-beta.ics.uci.edu/dataset/109/wine>
- [https://cran.r-project.org/web/packages/ggfortify/vignettes/plot\\_pca.html](https://cran.r-project.org/web/packages/ggfortify/vignettes/plot_pca.html)
- [https://uc-r.github.io/kmeans\\_clustering](https://uc-r.github.io/kmeans_clustering)