Wine Quality Analysis Report

Introduction

This report presents an analysis of a white wine dataset using unsupervised learning techniques in R. The dataset contains 2700 white wine samples from Portugal, with 11 physicochemical attributes measured for each sample. The goal is to perform clustering analysis to group similar wines together based on these attributes, without using the quality rating information. Both the full attribute space and a reduced PCA space will be used for clustering to compare, and learn the effects dimensionality reduction on k means analysis.

Data Preprocessing

First, let's load the required libraries and the wine dataset:

```
library(NbClust)
library(cluster)
library(factoextra)
library(tidyverse)
library(readxl)
library(ggplot2)
library(dplyr)
library(fpc)

wine_data = read_excel("Data/Whitewine_v6.xlsx")
view(wine_data) # reading excel file into application using readxl
```

The dataset appears to be ordered by the quality, so first we need to randomise the data, then remove the outliers.

```
# randomising data set as it has been grouped and ordered by quality
wine_random <- wine_data[sample(1:nrow(wine_data)), ]
view(wine_random)

# Remove outliers using Z score method
z_scores = as.data.frame(scale(wine_random))
no_outliers <- z_scores[!rowSums(abs(z_scores)>3.8), ]


#show the dimensions of the dataframes to see what has been removed
dim(wine_random)
dim(no_outliers)
boxplot(no_outliers)
```

After adjusting the upper range to remove the z scores, I settled on 3.8 as it removed a healthy number of outliers without massively reducing the dataset.

Determining Number of Clusters

To determine the optimal clusters, we'll use the four methods outlined in the brief:

```
set.seed(26)
```

```
##NOW PLOTTING THE ELBOW CURVE
fviz_nbclust(no_outliers, kmeans, method = 'wss', k.max = max(k))
#Elbow Curve says 2

fviz_nbclust(no_outliers, kmeans, method = 'silhouette', k.max = max(k))
#Silhouette says 2

fviz_nbclust(no_outliers, kmeans, method = 'gap_stat')
#gap_stat says 2

clusterNo=NbClust(no_outliers,distance="euclidean",
min.nc=2,max.nc=10,method="kmeans",index="all")

*******************************************************************
* Among all indices:
* 10 proposed 2 as the best number of clusters
* 7 proposed 3 as the best number of clusters
* 1 proposed 5 as the best number of clusters
* 4 proposed 9 as the best number of clusters
* 1 proposed 10 as the best number of clusters
```

Every method that we've used has recommended 2 as the optimal number of clusters so we'll proceed with k as 2.

K-means Clustering

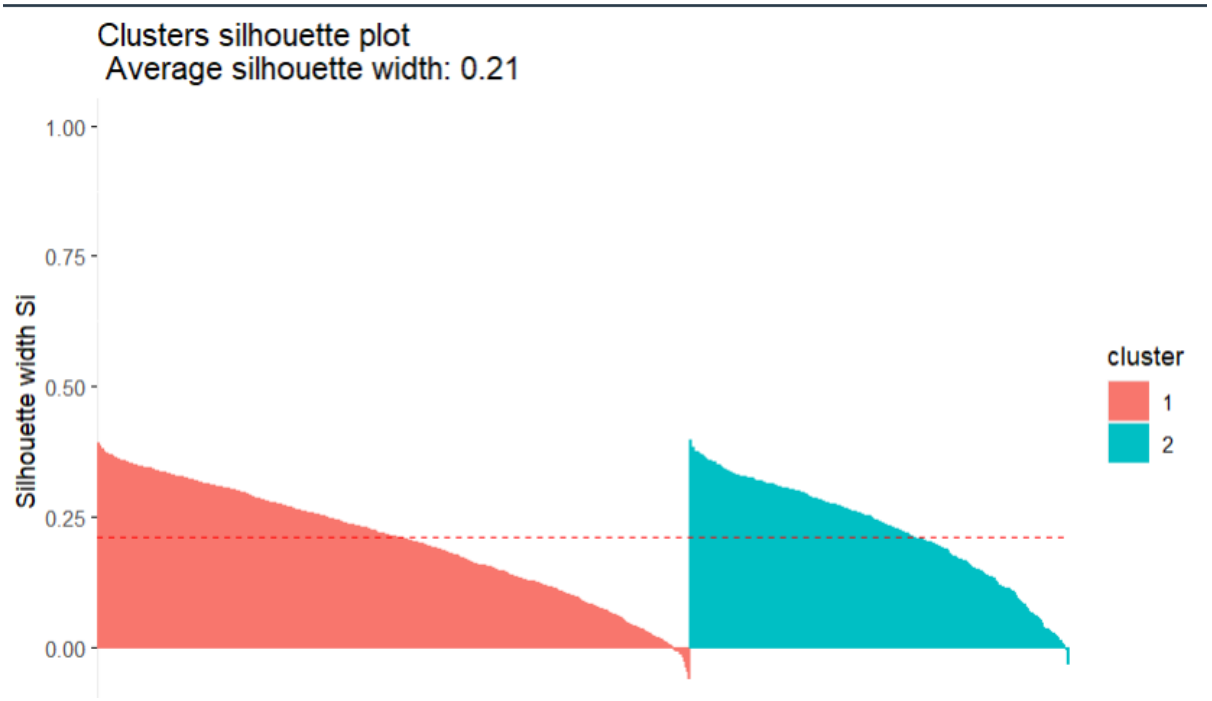
Now let's perform k-means clustering with k=2 on the dataset, note that we will remove all the results from the dataset as clustering is unsupervised; maintaining the results in then kmeans will negatively effect the clustering results.

```
#*Time to test using 2 as cluster number
x=no_outliers[, -length(no_outliers)]
y=no_outliers$quality
view(y)

kc = kmeans(x, 2)
kc

Within cluster sum of squares by cluster:
[1] 12201.664  7628.497
 (between_SS / total_SS =  23.9 %)

wss = kc$tot.withinss
bss = kc$betweenss
tss = kc$totss
bss/tss
#0.239
```

The BSS/TSS ratio of 0.239 indicates that a large portion of the variance is within the same clusters. We want the variance between the clusters themselves to be much higher generally. Because however, there are only 2 clusters, it makes sense that there is a lot of variance within the clusters because the dataset still is only split into 2, and splitting a dataset with still 7 dimensions into 2 is difficult to get the variance to be mostly between clusters.

Let's visualize the clusters with a silhouette plot:

```
sil <- silhouette(kmeans_no$cluster, dist(no_outliers))
fviz_silhouette(sil)
```



The average silhouette width of 0.21 suggests the clusters are not well separated and are not that reliable. There is overlap between the clustering that is showing they are not well defined.

PCA and Clustering

To reduce dimensionality, we'll apply PCA and select PCs that explain at least 85% of variance:

```
pca_wine = prcomp(x, center = TRUE, scale = FALSE)
summary(pca_wine)
```

```
#                           PC1    PC2    PC3     PC4     PC5     PC6    PC7
PC8     PC9    PC10
# Standard deviation     1.8060 1.2419 1.0501 0.96988 0.93854 0.80110
0.7532 0.70942 0.52948 0.48700
# Proportion of Variance 0.3271 0.1547 0.1106 0.09434 0.08834 0.06436
0.0569 0.05047 0.02812 0.02378
# Cumulative Proportion  0.3271 0.4818 0.5924 0.68670 0.77504 0.83940
0.8963 0.94677 0.97488 0.99867
# PC11
# Standard deviation     0.11516
# Proportion of Variance 0.00133
# Cumulative Proportion  1.00000
```

The first 7 PCs explain 89.6% of total variance, so we'll use those for clustering. -----

```
wine_transform = as.data.frame(-pca_wine$x[,1:7])
head(wine_transform)
```

Repeating the cluster number analysis on the PCA dataset:

```
fviz_nbclust(wine_transform, kmeans, method = 'gap_stat') #says 2

fviz_nbclust(wine_transform, kmeans, method = 'silhouette') #says 2

fviz_nbclust(wine_transform, kmeans, method = 'wss') #2

clusterNo=NbClust(wine_transform,distance="euclidean",
min.nc=2,max.nc=10,method="kmeans",index="all")
clusterNo

# * Among all indices:
#   * 12 proposed 2 as the best number of clusters
# * 3 proposed 3 as the best number of clusters
# * 1 proposed 4 as the best number of clusters
# * 2 proposed 6 as the best number of clusters
# * 4 proposed 7 as the best number of clusters
# * 1 proposed 8 as the best number of clusters
# * 1 proposed 10 as the best number of clusters
```

All of the methods are suggesting 2 clusters as a large majority, so we'll stick with k=2.

K-means clustering on the PCA dataset:

```
k = 2
kmeans_pca = kmeans(wine_transform, centers = k, nstart = 10)
kmeans_pca


wss = kmeans_pca$tot.withinss
bss = kmeans_pca$betweenss
tss = kmeans_pca$totss
bss/tss
```

The BSS/TSS ratio increased slightly to 0.267 which is better but still quite low. We want to minimise the WSS while maximising the BSS. With 0.239, and 0.267, the clustering explains a small portion of the variance and although there is an improvement, it is still not optimal, as over 70% of the variance is unaccounted for.

```
###silhouette plot
sil <- silhouette(kmeans_pca$cluster, dist(wine_transform))
fviz_silhouette(sil)


# cluster size ave.sil.width

# 1       1 1021          0.27

# 2       2 1593          0.24

#Average width 0.25
```

The average silhouette width increased to 0.25, indicating a slightly better formed cluster after PCA. Still, although an improvement, the silhouette width is showing that the clusters are not well separated overall, as we want the value to be as close to 1 as possible, which

would indicate that its more similar to its own cluster in comparison to the other cluster – 0.25 are still not highly separated.

Finally, let's calculate the Calinski-Harabasz index for the PCA clustering:

```
ch_index <- calinhara(wine_transform, kmeans_pca$cluster)#PCA
print(ch_index)
#949.2846

# to compare ...

ch_index2 <- calinhara(x, kmeans_no$cluster, cn=max(kmeans_no$cluste))
print(ch_index2)
#820
```

The CH index of 949 is a marginal improvement to the original dataset of 820, confirming the clusters are better defined and seperated. This does indicate that the PCA improved the structures of the clusters as the higher CH index suggests better definition and separation of clusters.

Conclusion

In this analysis, we performed k-means clustering on a white wine dataset, both before and after applying PCA for dimensionality reduction. Using multiple methods, we determined the optimal number of clusters is 2. The clusters formed using the full attribute space were moderately separated, and the PCA approach led to a slight improvement in cluster definition based on silhouette analysis and the CH index. The identified clusters could be useful for understanding patterns and similarities among the white wine samples based on their physicochemical properties.

Appendix: Full R Code

```
#install.packages("readxl")

#install.packages("NbClust")

#install.packages("cluster")

#install.packages("factoextra")

#install.packages("fpc")


library(NbClust)

library(cluster)

library(factoextra)
```

```
library(tidyverse)

library(readxl)

library(ggplot2)

library(dplyr)

library(fpc)



################################################

#READING THE EXCEL SHEET

wine_data = read_excel("Data/Whitewine_v6.xlsx")

view(wine_data) # reading excel file into application using readxl


wine_random <- wine_data[sample(1:nrow(wine_data)), ]

view(wine_random) # randomising data set as it has been grouped and ordered
by quality


boxplot(wine_random["pH"]) # to analyse data and view outliers in box plot


################################################


#After doing research I could find 2 ways to remove outliers

#1 was via IQR -> Outlier = Observations > Q3 + 1.5*IQR  or < Q1 - 1.5*IQR

#IQR itself is the measure of the spread of values in interquartile range
(50%)


#2 was via Z score by measuring the standard deviations it deviates from
the mean

# z = (X - μ) / σ  ->  Outlier = values with z-scores > 3 or < -3

#Although I enjoy IQRs elegance, I'd like to use Z-score as we were taught
about it

#and we can use that to normalise the data too


################################################


no_result <- subset(wine_random, select = -quality)

view(no_result) #taking quality out of the data frame
```

```
#z_scores <- as.data.frame(sapply(wine_random, function(wine_random)
((wine_random - mean(wine_random)) / sd(wine_random))))

z_scores = as.data.frame(scale(wine_random))


view(z_scores)

boxplot(z_scores)

dim(z_scores)


####################################

####################################

#################################### Calculated the z-score separately for
result


no_outliers <- z_scores[!rowSums(abs(z_scores)>3.8), ]

#show the dimensions of the dataframes to see what has been removed

dim(wine_random)

dim(no_outliers)

boxplot(no_outliers)


####################################

####################################

#Next is to use automated systems to estimate the number of clusters


##ELBOW METHOD##


k = 2:12

set.seed(42)

WSS = sapply(k, function(k) {kmeans(no_outliers, centers=k)$tot.withinss})

#Ecountered error due to NA entries


#checking...

any_na <- any(is.na(no_outliers))

print(any_na)

#found na going to compare to orignal data
```

```
any_na1 <- any(is.na(wine_random))

print(any_na1)

#No NA in the original data set !!

##The removal of z score over 3 was wrong had to change the line of code


##NOW PLOTTING THE ELBOW CURVE

fviz_nbclust(no_outliers, kmeans, method = 'wss', k.max = max(k))

#getting clearer answer this way looking like 2 clusters


plot(k, WSS, type = "b", pch = 19, frame = FALSE,

     xlab = "Number of Clusters (k)", ylab = "Total Within-Cluster Sum of
Squares (WSS)",

     main = "Elbow Method for Optimal Number of Clusters")


#########NOW USING NBcluster

#set.seed(26)

clusterNo=NbClust(no_outliers,distance="euclidean",
min.nc=2,max.nc=10,method="kmeans",index="all")

clusterNo=NbClust(no_outliers,distance="manhattan",
min.nc=2,max.nc=10,method="kmeans",index="all")

clusterNo=NbClust(no_outliers,distance="maximum",
min.nc=2,max.nc=10,method="kmeans",index="all")


print(clusterNo)

# *****************************************************************

#   * Among all indices:

#   * 10 proposed 2 as the best number of clusters

# * 7 proposed 3 as the best number of clusters

# * 1 proposed 5 as the best number of clusters

# * 4 proposed 9 as the best number of clusters

# * 1 proposed 10 as the best number of clusters


#silhouette says 2

fviz_nbclust(no_outliers, kmeans, method = 'silhouette', k.max = max(k))

#gap_stat says 10

fviz_nbclust(no_outliers, kmeans, method = 'gap_stat')
```

```
################################################
#*Time to test using 2 as cluster number
x=no_outliers[, -length(no_outliers)]
y=no_outliers$quality
view(y)
view(x)

kc = kmeans(x, 2)
kc

wss = kc$tot.withinss
bss = kc$betweenss
tss = kc$totss
bss/tss

table(y,kc$cluster)

plot(x, col=kc$cluster)
points(kc$centers, col=1:3, pch=23, cex=3)

k = 2
kmeans_no = kmeans(x, centers = k, nstart = 10)
kmeans_no
fviz_cluster(kmeans_no, data = no_outliers)

###silhouette plot
sil <- silhouette(kmeans_no$cluster, dist(no_outliers))
fviz_silhouette(sil)
# cluster size ave.sil.width
# 1       1 1592          0.20
# 2       2 1022          0.23

####NOW WITH PCA
```

```
pca_wine = prcomp(x, center = TRUE, scale = FALSE)

summary(pca_wine)


#                           PC1    PC2    PC3     PC4     PC5     PC6    PC7
PC8     PC9    PC10

# Standard deviation     1.8060 1.2419 1.0501 0.96988 0.93854 0.80110
0.7532 0.70942 0.52948 0.48700

# Proportion of Variance 0.3271 0.1547 0.1106 0.09434 0.08834 0.06436
0.0569 0.05047 0.02812 0.02378

# Cumulative Proportion  0.3271 0.4818 0.5924 0.68670 0.77504 0.83940
0.8963 0.94677 0.97488 0.99867

# PC11

# Standard deviation     0.11516

# Proportion of Variance 0.00133

# Cumulative Proportion  1.00000


#PC7 is over 85%



##################################################EXTRACT UP TO PCA 7

wine_transform = as.data.frame(-pca_wine$x[,1:7])

head(wine_transform)




##Now to analyse the clusters for this data set



clusterNo=NbClust(wine_transform,distance="euclidean",
min.nc=2,max.nc=10,method="kmeans",index="all")

clusterNo


# * Among all indices:

#   * 12 proposed 2 as the best number of clusters
```

```
# * 3 proposed 3 as the best number of clusters
# * 1 proposed 4 as the best number of clusters
# * 2 proposed 6 as the best number of clusters
# * 4 proposed 7 as the best number of clusters
# * 1 proposed 8 as the best number of clusters
# * 1 proposed 10 as the best number of clusters

fviz_nbclust(wine_transform, kmeans, method = 'gap_stat') #says 2

fviz_nbclust(wine_transform, kmeans, method = 'silhouette') #says 2

fviz_nbclust(wine_transform, kmeans, method = 'wss') #2
#Every method is saying 2 so lets do it
k = 2
kmeans_pca = kmeans(wine_transform, centers = k, nstart = 10)
kmeans_pca

fviz_cluster(kmeans_pca, data = wine_transform)

wss = kmeans_pca$tot.withinss
bss = kmeans_pca$betweenss
tss = kmeans_pca$totss
bss/tss


###silhouette plot
sil <- silhouette(kmeans_pca$cluster, dist(wine_transform))
fviz_silhouette(sil)

# cluster size ave.sil.width
# 1       1 1021          0.27
# 2       2 1593          0.24
#Average width 0.25
```

```
ch_index <- calinhara(wine_transform, kmeans_pca$cluster,
cn=max(kmeans_pca$cluster))

print(ch_index)



# to compare ...


ch_index2 <- calinhara(x, kmeans_no$cluster, cn=max(kmeans_no$cluste))

print(ch_index2)
```