Table of Contents

1.1 Introduction to dataset 3
2.1 Key Challenges and Problems 4
3.1 Exploratory Data Analysis 5
3.1.1 Statstical Inferences and Outlier detection 5
3.1.2 Class Imbalance 7
3.1.3 Correlation 8
3.2 Feature scaling 9
4.1 Supervised Learning 9
4.4 Random Forest 10
4.5 Confusion Matrix 11
4.6 Evaluation 11
5.1 Unsupervised Learning – Clustering 14
5.1.1 PCA – Dimensionality Reduction 14
5.1.2 PCA Performance Measure 15
5.2 K-Means Clustering 16
5.2.1 K-Means without PCA 16
5.2.2 Model evaluation K-means without PCA 17
5.2.3 K-means with PCA 18
5.2.4 Model evaluation K-means without PCA 19
6.1 Reflection 20
6.2 BIBLIOGRAPHY 20

List of figures

a. Fig 1.1: Distribution of the data..... 5

b. Fig 1.2: Boxplot of the Dataset.....5

c. Fig 1.3: Outlier calculation using Inter Quartile method 6

d. Fig 1.4 : Distribution of wine type 7

e. Fig1.5 : Quality before and after balancing the dataset 7

f. Fig 1.6 : Correlation Plot..... 8

g. Fig 1.7 : Explained Variance Ratio 14

h. Fig 1.8 : Elbow method for optimal K 17

i. Fig 1.9 : K-Means without PCA visualisation 18

j. Fig 1.10 : K-Means with PCA visualisation 19

Chapter 1

1.1 Introduction to dataset

Consumers and manufacturers value wine's quality. Product quality certification boosts sales if wine manufacturers maintains the quality. Worldwide, wine is a popular beverage, and enterprises employ quality certification to improvise their market value. Previously, product quality testing was done at the end of production, which was time-consuming and expensive due to the necessity for human experts to analyse product quality. Every human has their own take on the test, so categorizing wine quality based on humans is difficult. There are various wine quality predictors, however not all are relevant. The project tries to determine what wine features are crucial to produce a positive result utilising supervised machine learning classification methods such as Logistic Regression, Naïve Bayes and some tree based and ensemble methods on a wine quality dataset. The aim also to use some unsupervised methods like K-means and Agglomerative clustering. The UCI machine learning repository has the wine dataset (Cortez et al., 2009). The collection includes red and white "Vinho Verde" wine variations. It contains machine learning datasets.

The dataset consists of 6497 entries and 12 columns. Red wine has 1593 instances and white wine has 4870. Both files provide 11 input/output features. Input characteristics are based on tested lab results, and output variables are based on sensory data (0-very bad to 10-very good). For unsupervised learning method the input data is used as a whole. The data contains some outliers. It would be interesting to use some outlier detection techniques to understand the relevancy of the outliers. By having initial look at the data is appears we would have to use some feature engineering techniques to prepare the data well for the model. Accuracy score will help us to evaluate our model's performance. (Wine Quality, 2022)

Dataset - https://www.kaggle.com/datasets/rajyellow46/wine-quality

Chapter 2

2.1 Key Challenges and Problems

The main issues which we will be dealing in this data with the missing values and data cleaning without losing any important instances, such as by removing of columns that provide valuable information to the model. To improve the accuracy and retain the variability in model, the outliers in the dataset must be carefully inspected. Before the model can be trained, the target variable need to have balanced classes.

The model has super-vised methods and un-supervised methods. The supervised analysis includes logistic regression, Decision tree and Random Forest algorithms that analyses the quality of wine based on the different characteristics of wine. The outliers present in the data needs to evaluated on the basis of whether they need to be removed for the data to be statistically significant of they have to be retained as they contain valuable information.

In the unsupervised analysis, similar unlabelled data are grouped together with each other such that they create a cluster. After this the model groups similar attributes and find patterns in the dataset. The result from the unsupervised model could be wrong because of the lack of training data and no prior knowledge about the data such as label.

Chapter 3

3.1 Exploratory Data Analysis

3.1.1 Statstical Inferences and Outlier detection

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

Table 2.1: First five rows of the data

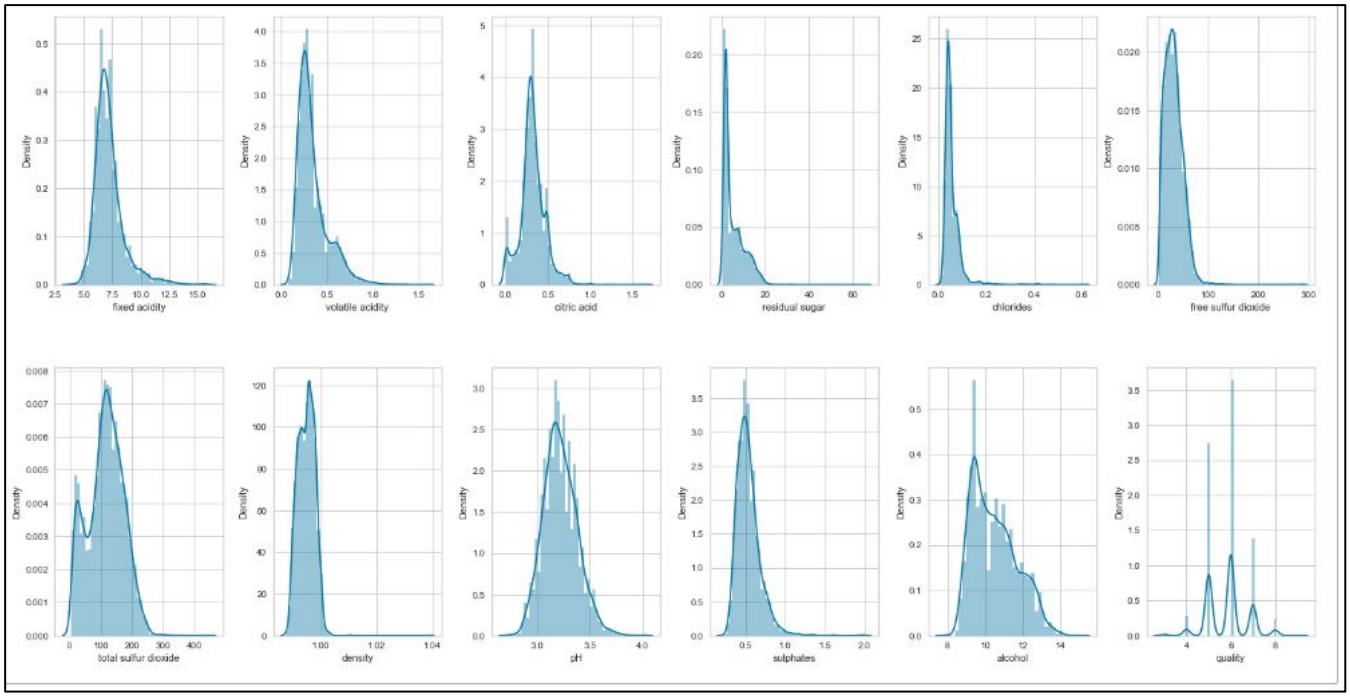

Fig 1.1: Distribution of the data

Based on the distribution charts, we may conclude that the data is almost normally distributed. Some columns, such as free sulphur dioxide, residual, sulphates, and alcohol, are slightly right skewed. However, we can correct this by either normalising or standardising the skewed features.

```
#create box plots
fig, ax = plt.subplots(ncols=6, nrows=2, figsize=(20,10))
index = 0
ax = ax.flatten()

for col, value in df.items():
    if col != 'type':
        sns.boxplot(y=col, data=df, ax=ax[index])
        index += 1
plt.tight_layout(pad=0.5, w_pad=0.7, h_pad=5.0)
```

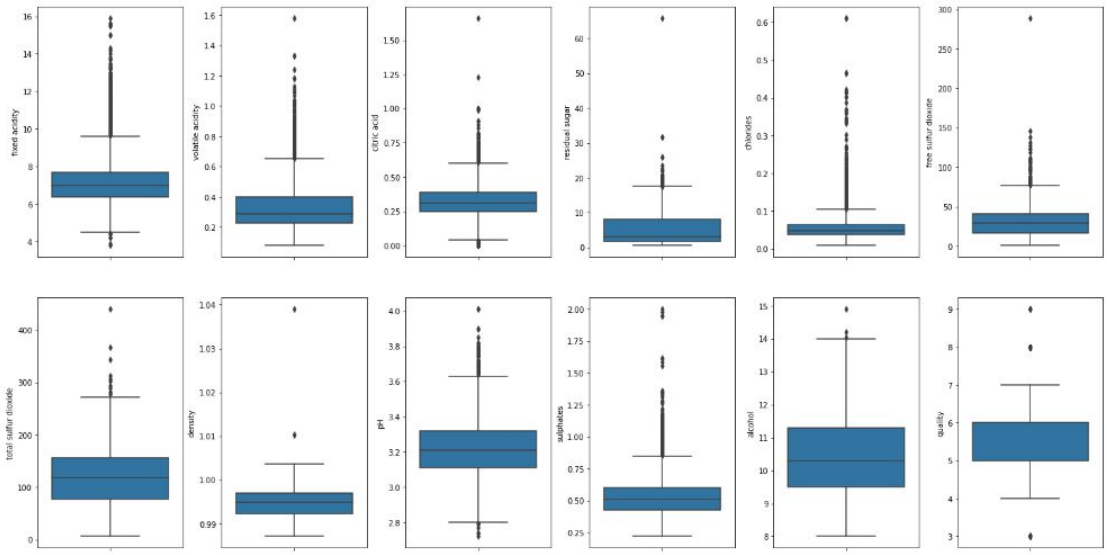

Fig 1.2: Boxplot of the Dataset

CALCULATING OUTLIERS

```
def remove_outlier(final_data):
    q1 = final_data.quantile(0.25)
    q3 = final_data.quantile(0.75)
    iqr = q3-q1

    final_data = final_data[~((final_data < (q1 - 1.5 * iqr)) |(final_data > (q3 + 1.5 * iqr))).any(axis=1)]

    return final_data
df2=df.copy()
```

```
df.shape
```

```
(6463, 13)
```

```
df2 = remove_outlier(df2) #outlier removal
df2.shape
```

```
C:\Users\Dell\Anaconda3\lib\site-packages\ipykernel_launcher.py:6: FutureWarning: Automatic reindexing on DataFrame vs Series comparisons is deprecated and will raise ValueError in a future version.  Do `left, right = left.align(right, axis=1, copy=False)` before e.g. `left == right`
```

```
(4815, 13)
```

Fig 1.3: Outlier calculation using Inter Quartile method

From fig 1.3 we can say that there are some outliers in almost every column. A function was implemented to calculate the outliers and it was found out that there are in total 1648 values which lie out of the IOR region as shown in figure 1.4.

There is a total of 25% of data that is excluded from the IOR. If we remove the outlier, we will lose crucial information about the few high-quality wines. The majority of wines fit into the 5, 6 and 7 classes, which explains why there are only a few outstanding grade wines that are mistaken as outliers. We cannot just eliminate these values to obtain statistical significance

because they exist. Furthermore, we cannot make the data appear less variable than it is. As a result, we would not eliminate the outlier in order to retain the variability. (Frost, 2022)

3.1.2 Class Imbalance

Figure 1.2 shows that the quality class of the red wine and white wine dataset shows that its distribution and we can see that most of the values are 5, 6, and 7, and all of the class values are between 3 and 9 in the quality feature. This means that the data is not balanced, and the model built from the data would be a biased model that ignores the classes that come up less often.

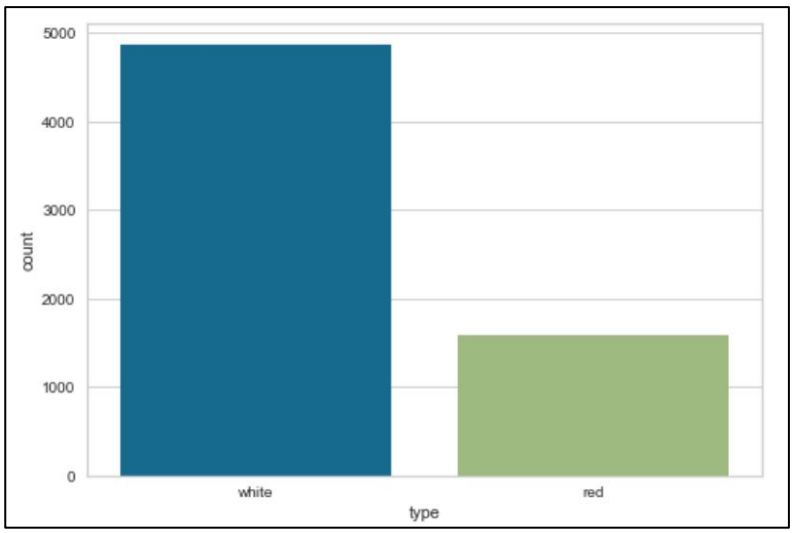

Fig 1.4 : Distribution of wine type

The middle class is rarely compared to the highest quality ratings. By resampling this problem, examples from the underrepresented class are added unnaturally (over-sampling) or deleted (under-sampling). Unless you have an abundance of data, it is often preferable to use a sample size greater than 15 samples. Although over-sampling has advantages, it has some drawbacks as well. Because it increases the dataset's instances, the model's processing time increases. Extreme cases of over-sampling can lead to over fitting Resampling is therefore chosen.

The SMOTE (Synthetic Minority Oversampling Technique) method works by picking a point at random from the minority class and finding its k-nearest neighbours. The synthetic points are added between the chosen point and the points around it. The synthetic point is placed anywhere on the line joining the point under consideration and its chosen neighbour. It then repeats the steps until data is balanced. (Imbalanced Classification | Handling Imbalanced Data using Python, 2022)

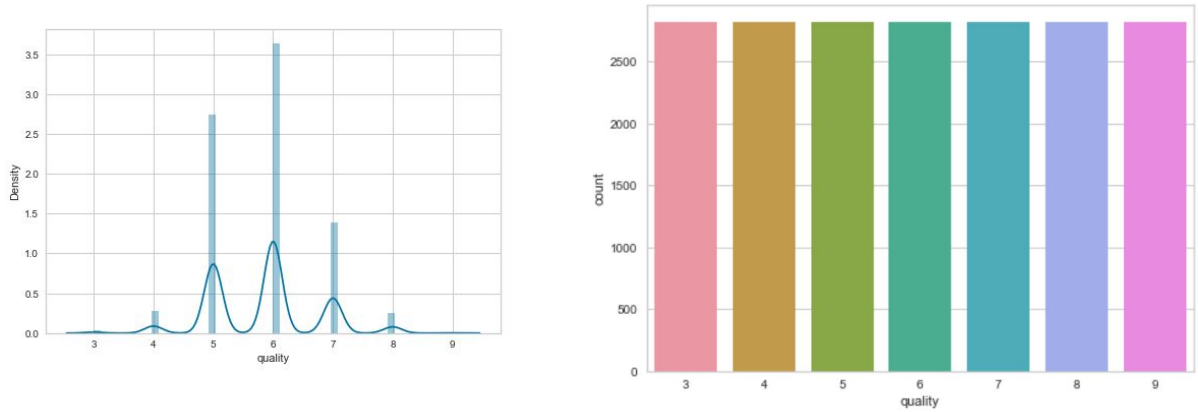

Fig1.5 : Quality before and after balancing the dataset

3.1.3 Correlation

The correlation shows how much two variables are related to each other. Positive correlation values range from -1 to 1, whereas negative correlation values range from 0 to 1. Each square in a heatmap displays correlation between two variables. When one variable rises, so does the other. When one variable rises, the other falls.

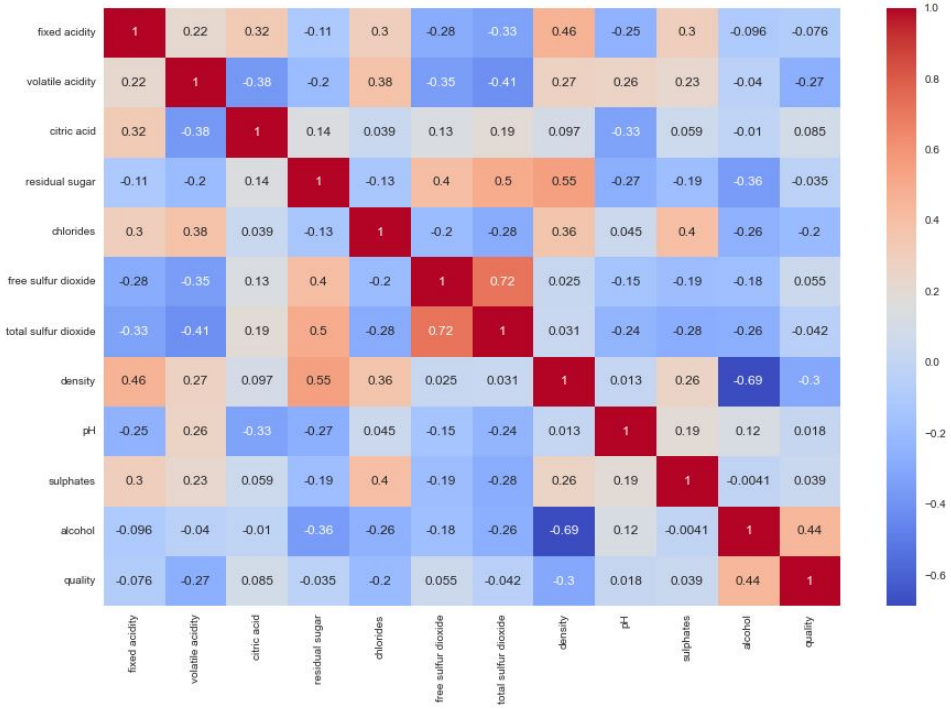

Fig 1.6 : Correlation Plot

A correlation matrix has been utilized to figure features that are highly related to the quality of red wine. From the heatmap, we can see that most of the features have a weak relationship with the quality of the wine, except for alcohol (0.48), which has a medium relationship.

Volatile acidity (-0.27), chlorides (-0.2), and density (-0.3) all have a negative relationship with the quality of wine. As these variables go down, the quality of wine will go up and vice versa.

On the other hand, fixed acidity (0.12), citric acid (0.085), and alcohol (0.44) are all positively related to wine quality. As these variables go up, so does the quality of the wine. The column Total Sulfur Dioxide was dropped as it was colliear with Free Sulphur Dioxide.

3.2 Feature scaling

The data standardisation technique can scale the features between 0 and 1. This will help you learn the model by applying it to all the numeric features and then separating the data by standard derivation. So, we use this method to make sure the data are all the same.

The formula for standardisation is:

$$zi = xi – u / \sigma$$

where σ is the standard derivation, xi is each value, and u is the mean value of the array x.

I used Python's sklearn.preprocessing module's StandardScaler class to do standard scaling. StandardScaler class's fit transform() method was called and the dataframe was fed into it. This was done to to bring the features in same scale to reduce the effect of outliers. Also, standardizatioin is a safe approach after implementing SMOTE.

Chapter 4

4.1 Supervised Learning

In supervised learning, input variables (x) and output variables (Y) are used in conjunction with an algorithm to learn the mapping function from input to output. The objective is to approximate the mapping function so well that, given new input data (x), the output variables (Y) can be predicted.

Multiple algorithms were used for supervised learning with logistic regression as the baseline model with Decision Tree and Random Forest as the best performing model. (Brownlee, 2022)

4.2 Logistic Regression

The logistic function, often known as the sigmoid function, is the name of the function used in logistic regression. The S-shaped function transfers any real number to 1 or 0. This method predicts binary classification using real-valued inputs.

If the input's likelihood of belonging to one class is less than 0.5, the prediction is for the other class. The approach estimates coefficient values using stochastic gradient descent based on the training set. Sigmoid function converts algorithm output to probability..
(sklearn.linear_model.LogisticRegression, 2022)

4.3 Decision Tree

Decision trees are a very strong and versatile machine learning technique that can suit composite datasets and can also be utilised for regression and classification applications. "Using a decision tree to make predictions is pretty straightforward. The tree is fed new input, and each input is assessed at the tree's root node. The decision tree divides the space by branching out into rectangles or hyperrectangles. The model predicts the output value of fresh input by filtering through the tree and settling in one of the rectangles; the model's prediction is the value of that associated rectangle."

The first step is to generate a decision tree classifier object, which is then trained using the.fit function on the training set of variables X and y. The model is used to predict the response for the test dataset after it has been trained. (Brownlee, 2022)

4.4 Random Forest

Like its name suggests, a random forest contains many independent decision trees that work together. Our model's prediction is based on the class predictions made by each individual tree in the random forest.

A random subset of size max features or all input features are used to find the best split for each node during tree construction. Details on how to fine-tune parameters can be found in the parameter tuning recommendations).

The forest estimator's variance can be reduced by using these two randomness generators. Individual decision trees, on the other hand, have a tendency to overfit and demonstrate a high degree of variability. Decision trees with decoupled prediction errors are the result of randomization being injected into forests. Some mistakes can be eliminated by averaging the projections. By merging trees from different species, random forests can minimise variation, but this can result in an increase in bias. Since the reduction in variance is often large, the entire model is improved. (Understanding Random Forest, 2022)

4.5 Confusion Matrix

The confusion matrix consists of four basic characteristics (numbers) that are used to define the measurement metrics of the classifier. These four numbers are:

1) **True positive (TP) –**

   A test result indicating the presence of a condition or characteristic.

2) **True negative (TN) –**

   A test result indicating the absence of a condition or trait.

3) **False positive (FP) -**

   A test result that incorrectly implies the presence of a specific condition or feature.

4) **False negative (FN) -**

   A test result that incorrectly shows the absence of a specific condition or attribute.

The evaluation results were achieved from each implementation of the classification algorithm calculated. As mentioned in the Evaluation sub-section. (Confusion matrix - Wikipedia, 2022)

4.6 Evaluation

Logistic Regression	True Positive	True Negative	False Positive	False Negative
3	3925	391	296	323
4	3833	353	367	382
5	3811	272	442	410
6	3825	170	546	392
7	3846	221	482	386
8	3769	363	318	485
9	4150	714	0	71

Table 2.2: Confusion Matrix of Logistic Regression

Decision tree	True Positive	True Negative	False Positive	False Negative
3	4914	662	25	54
4	4113	593	127	102
5	4003	467	247	218
6	3921	397	319	298
7	4052	527	176	180
8	4156	624	57	98
9	4219	713	1	2

Table 2.3 : Confusion Matrix of Decision Tree

Random Forest	True Positive	True Negative	False Positive	False Negative
3	4220	684	3	28
4	4122	677	43	93
5	4036	513	201	185
6	4020	410	306	199
7	4085	563	140	147
8	4091	660	21	63
9	4221	713	1	0

Table 2.4 : Confusion matrix of Random Forest

The evaluation from the classification methods is shown in the tables. shows the unbalanced classes and the performance of the prediction model, in terms of Accuracy, precision, recall, and F1 score is examined, as expressed in Table 2.2, 2.3, 2.4

	LR			DT			RF		
Class	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
3	0.55	0.57	0.56	0.92	0.96	0.94	0.96	1.00	0.98
4	0.48	0.49	0.49	0.85	0.82	0.84	0.88	0.94	0.91
5	0.40	0.38	0.39	0.68	0.65	0.67	0.73	0.72	0.73
6	0.30	0.24	0.27	0.57	0.55	0.56	0.67	0.57	0.62
7	0.36	0.31	0.34	0.75	0.75	0.75	0.79	0.80	0.80
8	0.43	0.53	0.47	0.86	0.92	0.89	0.91	0.97	0.94
9	0.91	1.00	0.95	1.00	1.00	1.00	1.00	1.00	1.00
Accuracy		0.50			0.81			0.85	

Table 2.5 : Performance evaluation for Logistic regression, Decision Tree and Random Forest

As we can see The Random Forest has the best accuracy. Its interesting to see that the precision values for extremely high and lower classes is 1 due to the reason that we have balanced the dataset. So there has to be another way to correctly predict wine class. We will try some unsupervised methods and Dimensionality reduction technuiques to obtain a better results.

Chapter 5

5.1 Unsupervised Learning – Clustering

In the unsupervised method, just input data and no output variable are given to the model. No proper solutions exist; algorithms must find patterns in data on their own. We'll cluster this data using K-means clusturing.

5.1.1 PCA – Dimensionality Reduction

Principal Component Analysis, or PCA, is a method for lower the number of dimensions in a large data set. It does this by changing a large set of variables into a smaller set that still has most of the same information.

When the number of variables in a data set, we lose some accuracy. However, the key to dimensionality reduction is to give up a little accuracy for ease of use. Smaller data sets are easier to explore and see, and they make it easier and faster for machine learning algorithms to analyse data because they don't have to deal with as many variables.

Principal components are the new variables that are constructed as linear combination of initial variable.These combinations are done in a way to stuff as much that 10 combinations are made of 10 variables but the algorithms tries to squeeze in as much variables as possible in the first components, followed by remaing information in second component until the graph

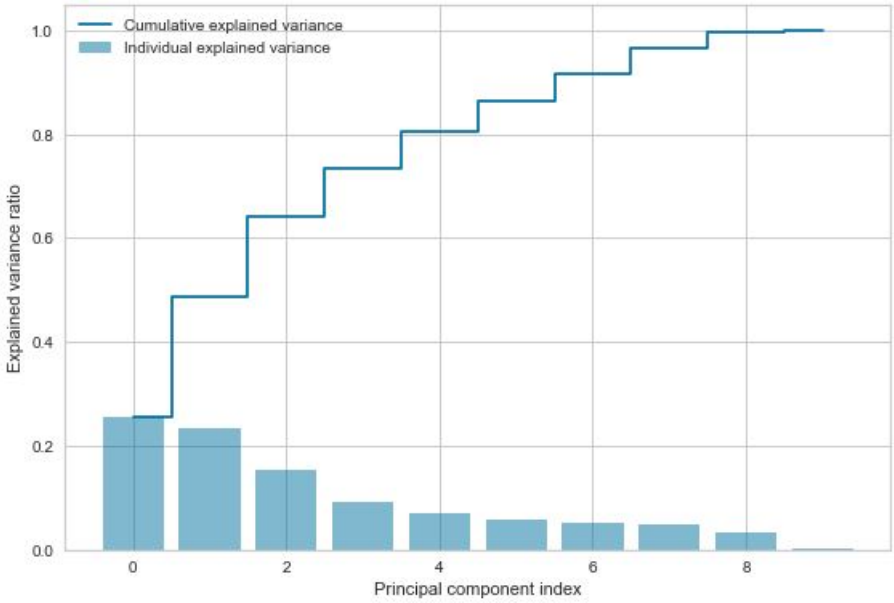

Fig 1.7 : Explained Variance Ratio

looks like below; principal component represent the directions of the data that explains maximum amount of variance.

Steps to carry out pca:-

1) **Standardization** –

A normal feature scaling to bring all the obsevartions in the same scale.

2) **Covariance Matrix Computation** –

The goal of this step is to figure out how the variables in the input data set differ from the mean in relation to each other, or if there is any link between them. The covariance matrix is a symmetric p x p matrix (where p is the number of dimensions) with entries for all possible pairs of initial variables and their covariances. (A Step-by-Step Explanation of Principal Component Analysis (PCA), 2022) For example, the covariance matrix for a 3-dimensional data set with three variables (x, y, and z) is a 33 matrix of: (A Step-by-Step Explanation of Principal Component Analysis (PCA), 2022)

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

3) **Computation of Eigen Values and Eigen Vectors to Identify the Principal Component** –

The directions of the axes where there is the most variance (most information) are called Principal Components and are the eigenvectors of the covariance matrix. And eigenvalues are just the coefficients that are attached to eigenvectors. These coefficients show how much variation each Principal Component carries. (A Step-by-Step Explanation of Principal Component Analysis (PCA), 2022)

5.1.2 PCA Performance Measure

We are taking n_components = 6. There will be 6 principal components which means the data of 11 dimensions is reduced to 6 dimension. To evaluate the performance, we will check the overall explained variance ratio. Also we will check the explained variance ratio of each component.

Explained variance represents the information explained using different principal components (eigenvectors) (A Step-by-Step Explanation of Principal Component Analysis (PCA), 2022)

Explained variance is calculated as ratio of eigenvalue of a particular principal component (eigenvector) with total eigenvalues.

Explained variance can be calculated as the attribute explained_variance_ratio_ of PCA instance created using sklearn.decomposition PCA class.

The Overal Explained variance ratio is 86.5% which means the components are explaining 86.5% variability in the data.

The explained variance ratio of individual components are displayed in the table below:

Component	1	2	3	4	5	6
Ratio(%)	25.51	23.32	15.32	9.29	7.03	6

Table 2.6 : explained variance ratio of individual components

We will check the performance of the unsupervised model both with and without PCA.

5.2 K-Means Clustering

This unsupervised technique groups similar data points by k. Euclidean distance measures similarity. As a first step, begin by creating k(mean) points and then categorising data points to their nearest mean and updating the mean coordinates, which are the average of the data points in that mean.

Two approaches can be used to determine k. Use the elbow method or the silhouette method. We used elbow to find k. When using the elbow method, the dataset is clustered for a range of k values (from 1 to 10) and distortion and inertia are calculated for each k value, where distortion is the average of the squared distances between the various cluster centres and inertia is the sum of the squared distances between samples and the closest cluster centre. The optimal value of k can be determined by plotting these measurements.The elbow graph shows appropriate cluster size. (Understanding K-means Clustering in Machine Learning, 2022)

5.2.1 K-Means without PCA

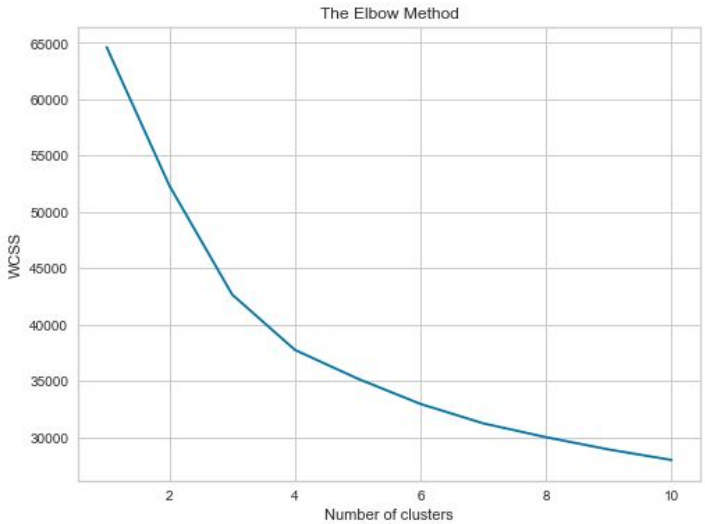

Fig 1.8 : Elbow method for optimal K

The elbow point is at k=2, hence the optimal number of clusters is 2 in the given picture.

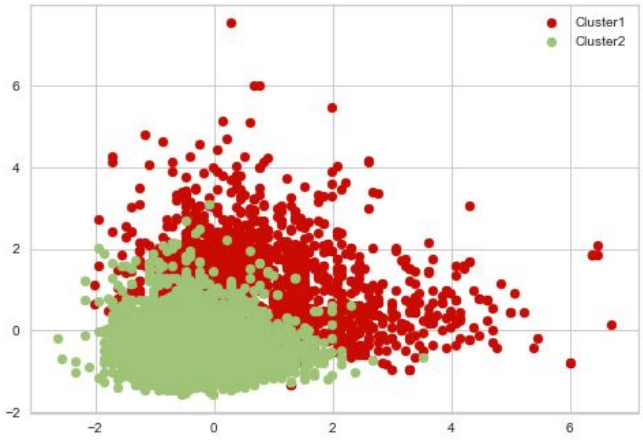

Fig 1.9 : K-Means without PCA visualisation

In figure, we can see that the cluster are not properly separated which means some of the data points from cluster 1 is present in cluster 2 and vice versa. We will analyse the performance with evaluation metrics such as Silhoutte Score, Homogenity score and Inertia score

5.2.2 Model evaluation K-means without PCA

1) **Silhouette Score** –

When comparing data points in one cluster to those in other clusters, the metric is known as the cluster-to-cluster comparison. Clustering occurs in the range -1 to 1, with 1 indicating full clustering.

2) **Homogenity score-**

It checks the homogeneity of cluster where each cluster has information that directs a place toward a similar class label. Homogeneity portrays the closeness of the clustering algorithm to this (homogeneity_score) perfection. A score between 0 to 1 means the data is homogeneous.

3) **Inertia Score-**

It checks the the K-Means clustering performance. It is determined by calculating the separation between each data point and its centroid, squaring this separation, and adding these squares for each cluster. (Learn the Basics of Machine Learning: Clustering: K-Means Cheatsheet | Codecademy, 2022)

Mectrics	Score
Sillhoutte score	25.6
Homogenity score	0.008
Inertia score	52331.71

Table 2.7 : Performance evaluation for k-Means without PCA

Based on the comparison of the clusters with the average silhouette score, the ideal silhouette score of the clusters is determined, which is the suboptimal cluster will have a silhouette score higher than the average. After applying PCA, we'll try using k-means to analyse the data, which has a lower score than the Silhouette score of 26. The inertia score also is high which means we are using a lot more features than required. On that note we will try to evaluate the performance of K-means with PCA.

5.2.3 K-means with PCA

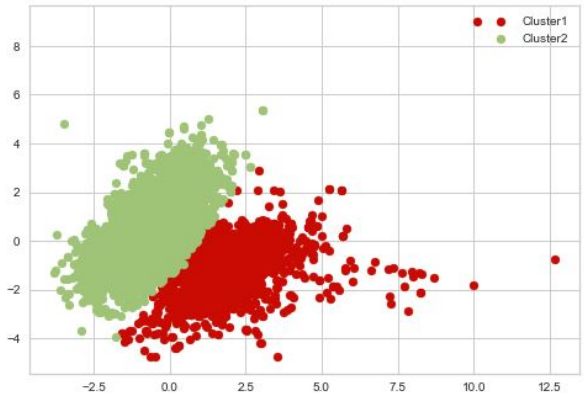

Fig 1.9 : K-Means with pca visualisation

The figure shows that there is a better separation of data points that without pca. The evaluation is done by using the same methods in the table below;

5.2.4 Model evaluation K-means without PCA

Metrics	Score
Silhouette score	28.5
Homogeneity score	0.008
Inertia score	43611

Table 2.7 : Performance evaluation for k-Means with PCA

The value of inertia is High.

K-means clustering fails with multidimensional data. Even with PCA, silhouette score is still low. It has a high inertia even though it's better with PCA. Ideal inertia values are low. The model doesn't suit the data properly.

CHAPTER 6

6.1 Reflection

The supervised learning method performed better than unsupervised learning method. The dataset has several data points, which helped the model train and predict new values. The random forest model achieved 85% of test accuracy and it is believed that with hyper parameter tuning we can achieve better results. However, this was observed in the confusion matrix where false negative and false positive are nearly 0 for the classes which are extreme and have been balanced by SMOTE.

As an unsupervised model, K-means was weak. Even if grouping wine by quality was the primary purpose, there may be no logical connection between the characteristics that were selected. Because the model is sensitive to outliers, or because the data overlapping makes it difficult for the algorithm to discern that there are two groups, this could be the most likely explanation. The PCA was able to explain 86.5 percent of the variance in the 11 features down to just six. Additionally, the model's sillhoutte score improved to 0.27 after PCA was applied, but it was still below the threshold required to match the data points with their respective clusters. Even after pca, my inertia score remained high, indicating that I was still extracting more features than was necessary. As a result, K-means cannot be used as a model for this situation.

6.2 Bibliography

1) Analytics Vidhya. 2022. *Imbalanced Classification | Handling Imbalanced Data using Python*. [online] Available at: <https://www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/#:~:text=One%20way%20to%20fight%20imbalance,of%20the%20currently%20available%20samples.> [Accessed 9 July 2022].
2) Brownlee, J., 2022. *Supervised and Unsupervised Machine Learning Algorithms*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/> [Accessed 8 July 2022].
3) Medium. 2022. *Understanding K-means Clustering in Machine Learning*. [online] Available at: <https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1> [Accessed 8 July 2022].
4) Frost, J., 2022. *Guidelines for Removing and Handling Outliers in Data*. [online] Statistics By Jim. Available at: <https://statisticsbyjim.com/basics/remove-outliers/> [Accessed 8 July 2022].
5) Kaggle.com. 2022. *Wine Quality*. [online] Available at: <https://www.kaggle.com/datasets/rajyellow46/wine-quality> [Accessed 8 July 2022].
6) Codecademy. 2022. *Learn the Basics of Machine Learning: Clustering: K-Means Cheatsheet | Codecademy*. [online] Available at: <https://www.codecademy.com/learn/machine-learning/modules/dspath-

clustering/cheatsheet#:~:text=K%2DMeans%3A%20Inertia,these%20squares%20across%20one%20cluster.> [Accessed 8 July 2022].
7) Built In. 2022. *A Step-by-Step Explanation of Principal Component Analysis (PCA)*. [online] Available at: <https://builtin.com/data-science/step-step-explanation-principal-component-analysis> [Accessed 8 July 2022].
8) Medium. 2022. *Understanding Random Forest*. [online] Available at: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2> [Accessed 8 July 2022].
9) scikit-learn. 2022. *sklearn.tree.DecisionTreeClassifier*. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html> [Accessed 9 July 2022].
10) scikit-learn. 2022. *sklearn.linear_model.LogisticRegression*. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html> [Accessed 8 July 2022].
11) Brownlee, J., 2022. *Classification And Regression Trees for Machine Learning*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/> [Accessed 9 July 2022].
12) En.wikipedia.org. 2022. *Confusion matrix - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Confusion_matrix> [Accessed 8 July 2022].