Assignment 2: Data Modelling

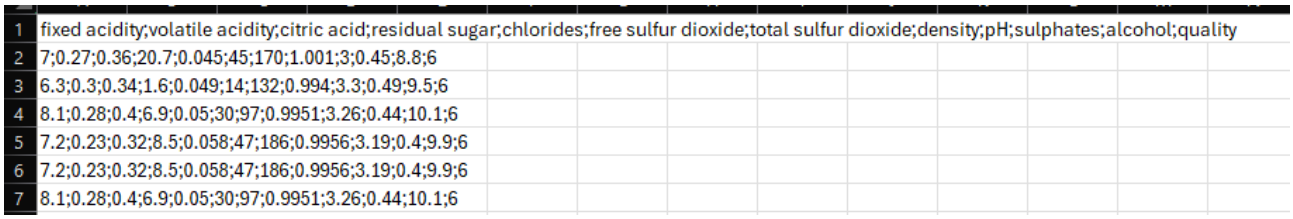
Ridge Tagala (s3934367)

In this assignment, we are given a data set based off the quality of wine production, with 12 columns: *Fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol* and *quality.* We are to perform a bunch of statistical analysis models on the data set based on the questions given.

Task 1: Regression

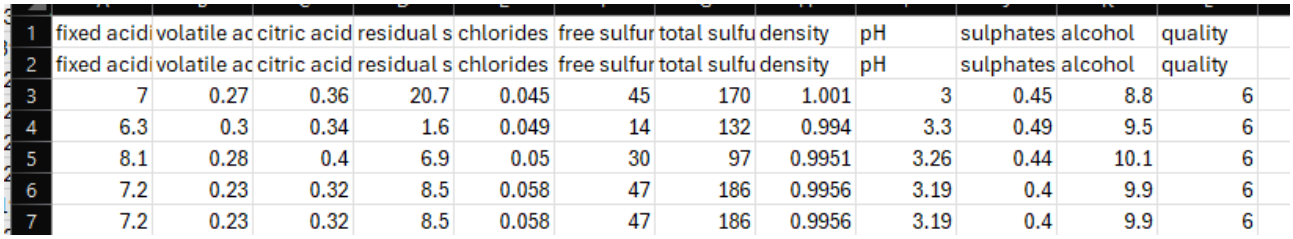
Data Preparation

Similar to the previous assignment, I had to prepare the data for analysis. Going through the initial spreadsheet, we could see that the data needed to be formatted (image below)



1	fixed acidity;volatile acidity;citric acid;residual sugar;chlorides;free sulfur dioxide;total sulfur dioxide;density;pH;sulphates;alcohol;quality
2	7;0.27;0.36;20.7;0.045;45;170;1.001;3;0.45;8.8;6
3	6.3;0.3;0.34;1.6;0.049;14;132;0.994;3.3;0.49;9.5;6
4	8.1;0.28;0.4;6.9;0.05;30;97;0.9951;3.26;0.44;10.1;6
5	7.2;0.23;0.32;8.5;0.058;47;186;0.9956;3.19;0.4;9.9;6
6	7.2;0.23;0.32;8.5;0.058;47;186;0.9956;3.19;0.4;9.9;6
7	8.1;0.28;0.4;6.9;0.05;30;97;0.9951;3.26;0.44;10.1;6

This was done by reading the .csv file using pandas, and then using the keyword *sep=','* to separate values, along with listing the columns using *names* to correctly format the data. This was saved to a spreadsheet called *s3934367-formatted-A2data.csv* The end result is this:



1	fixed acidi	volatile ac	citric acid	residual s	chlorides	free sulfur	total sulfu	density	pH	sulphates	alcohol	quality
2	fixed acidi	volatile ac	citric acid	residual s	chlorides	free sulfur	total sulfu	density	pH	sulphates	alcohol	quality
3	7	0.27	0.36	20.7	0.045	45	170	1.001	3	0.45	8.8	6
4	6.3	0.3	0.34	1.6	0.049	14	132	0.994	3.3	0.49	9.5	6
5	8.1	0.28	0.4	6.9	0.05	30	97	0.9951	3.26	0.44	10.1	6
6	7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4	9.9	6
7	7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4	9.9	6

Next, we needed to further clean the data, as looking through the spreadsheet there were some columns of data that were missing or empty (also known as NULL or NA) which can affect the accuracy of our analysis. To fix this, we used the *dropna()* function which gets rid of all the invalid data.

The end result of this data cleaning process is a .csv file which can provide the most accurate results from the analysis, as the data contains no empty columns and no NULL values.

Box plot analysis

This question stated to compare the relationship between the two variables **alcohol** and **density,** to find out if there is a relationship between the two. The graph chosen to display this relationship is a **box plot,** as we have multiple data ranges, which can be considered as 'groups'.
Firstly, i created a list containing all the values for 'density' and 'alcohol'.

```
Alcohol = [
    9.2, 11.4, 8.7, 9.1, 11.3, 9.8, 9.5, 9, 9, 10.4, 11.2, 10.8, 11.4, 9, 9.7, 10, 9, 8.9, 11.3, 9.1,
    10.1, 10.2, 12.2, 10.4, 11, 12, 12, 9.4, 12.5, 9.5, 10.5, 12.5, 10.5, 8.8, 12.3, 10.5, 12.5, 10.2,
    13, 10, 9.2, 12.4, 10.8, 11.2, 9.5, 8.8, 9.8, 9.7, 11.8, 13.8, 9.4, 9.1, 9.2, 12.2, 10, 9.4, 9,
    8.6, 8.7, 9.3, 11.9, 9.2, 8.7, 8.6, 10.5, 9.2, 11.7, 10.7, 9, 11.3, 12.7, 10, 13, 10, 9.8, 11, 8.9,
    10.8, 11.4, 9.2, 10.5, 8.7, 9.5, 10.2, 12.2, 9, 12.5, 10.5, 9.1, 11.3, 9.2, 11.4, 12.1, 10.5, 10.2,
    10.8, 11.5, 12.4, 11.5, 10.1, 9.2, 10.2, 8.9, 11.5, 11, 9.5, 11.4, 10.1, 9.5, 10.4, 13.9, 9.7, 8.9,
    9.6, 12.6, 11.9, 9.9, 10.6, 11, 9.9, 12, 13.7, 12.4, 9, 9.4, 9, 11, 11, 9.2, 9.4, 10.1, 10.6, 9.2,
    11, 10.2, 11.5, 12.8, 12.1, 10.6, 9.4, 10.4, 9.4, 10.1, 10.2, 9, 10.1, 10.9, 10.5, 9.5, 9.9, 11.5,
    11.3, 9.6, 9.8, 9.5, 10, 11.8, 9, 9.3, 10.6, 9.4, 11.05, 11, 9.5, 11.5, 9.1, 9.4, 11.9, 9.9, 8.9,
    10, 9.5, 11, 10.9, 12, 13, 10.9, 12.3, 9.2, 10.9, 11.2, 9.2, 10.1, 9, 11.1, 9.8, 9.9, 11.6, 9.8,
```

```
Density = [
    0.994, 0.9936, 0.99786, 0.9936, 0.9904, 0.99699, 0.9958, 0.99773, 0.99772, 0.9926, 0.9904, 0.99365,
    0.99044, 0.9989, 0.9946, 0.9932, 0.9978, 0.9979, 0.9918, 0.99336, 0.9944, 0.99348, 0.99122, 0.99454,
    0.9914, 0.99193, 0.9924, 0.9948, 0.9888, 0.997, 0.99508, 0.9902, 0.99382, 0.99807, 0.99024, 0.99523,
    0.99151, 0.9928, 0.99, 0.99388, 1.00051, 0.9895, 0.9923, 0.99082, 0.9952, 0.99908, 0.99438, 0.9966,
    0.9923, 0.98889, 0.9954, 0.9983, 0.99804, 0.99021, 0.99243, 0.9993, 0.99792, 0.9999, 0.9976, 0.99545,
    0.9936, 0.9968, 1.0002, 0.9966, 0.99144, 0.99318, 0.98949, 0.9962, 0.9959, 0.99146, 0.9898, 0.9933,
    0.98919, 0.99383, 0.99232, 0.99096, 0.9961, 0.99402, 0.9914, 0.99344, 0.9918, 0.9936, 0.99784, 0.99166,
    0.98944, 0.9951, 0.9916, 0.99184, 0.99571, 0.99135, 0.9964, 0.9925, 0.99258, 0.99546, 0.995, 0.99299,
    0.9928, 0.98962, 0.99335, 0.9934, 0.9956, 0.99289, 0.99862, 0.99084, 0.9922, 0.9968, 0.99047, 0.9948,
    0.99551, 0.99587, 0.98936, 0.9952, 0.9986, 0.99767, 0.9896, 0.99004, 0.9968, 0.99246, 0.9966, 0.9939,
    0.9944, 0.9907, 0.98914, 0.99449, 0.9963, 0.99758, 0.9939, 0.99096, 0.9952, 0.99548, 0.99555, 0.9927,
    0.9989, 1.0001, 0.9942, 0.9928, 0.9895, 0.99254, 1.00295, 0.99552, 0.992, 0.9965, 0.9944, 0.99532,
    0.99625, 0.99352, 0.994, 0.9928, 0.99784, 0.9937, 0.9924, 0.99165, 0.9948, 0.99438, 0.99784, 0.99725,
    0.99035, 0.9951, 0.99658, 0.99397, 0.9933, 0.99043, 0.9924, 0.9969, 0.9947, 0.9986, 0.9974, 0.99093,
    0.9976, 0.9958, 0.992, 0.99402, 0.99268, 0.99406, 0.991, 0.98912, 0.9926, 0.98952, 0.9956, 0.9911,
    0.992, 0.99804, 0.99155, 0.9958, 0.99127, 0.9927, 0.9958, 0.9911, 0.99434, 0.9954, 0.98964, 0.9956,
    0.9912, 0.99542, 0.99924, 0.99206, 0.99174, 0.99088, 0.98926, 0.99726
```

As you can see here, there are more values in density than there are in alcohol.

This was then made into a data frame and converted. I then used the boxplot() function to generate a boxplot, denoted as A2_sample_data.boxplot(), with the functions column = alcohol and by = density. (group by density). This is the end result:

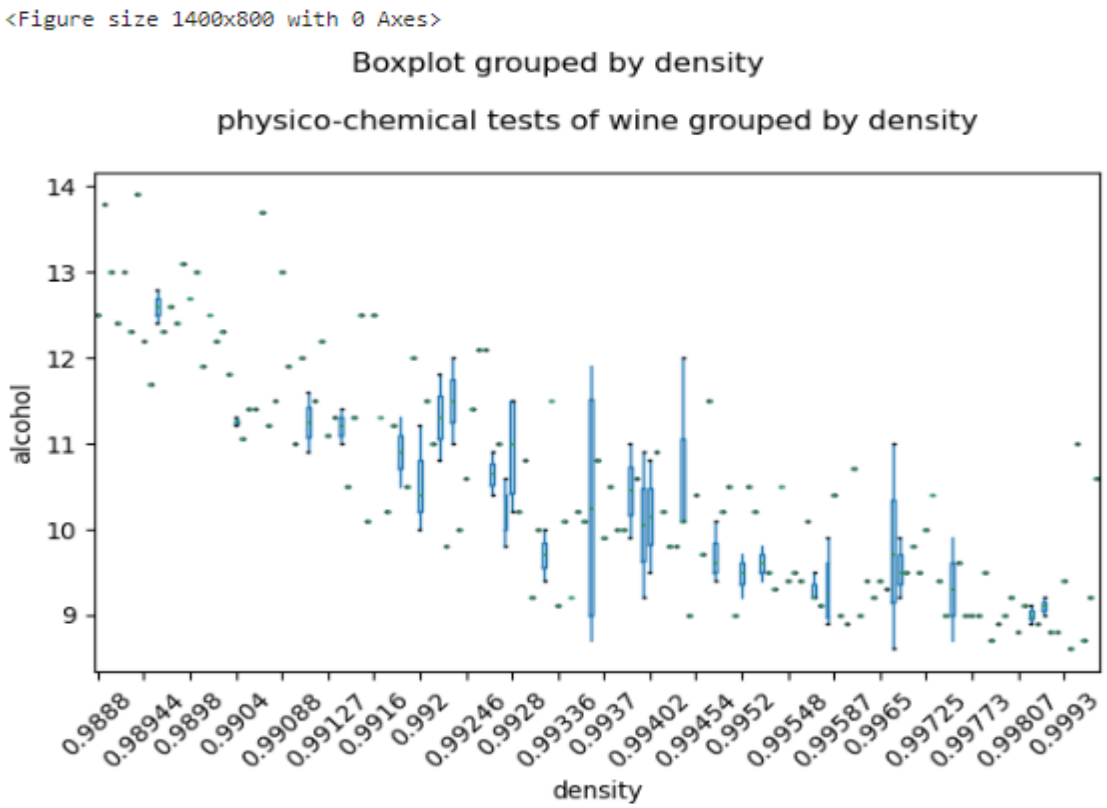


Figure 1: boxplot of alcohol grouped by density. left is lower, right is higher

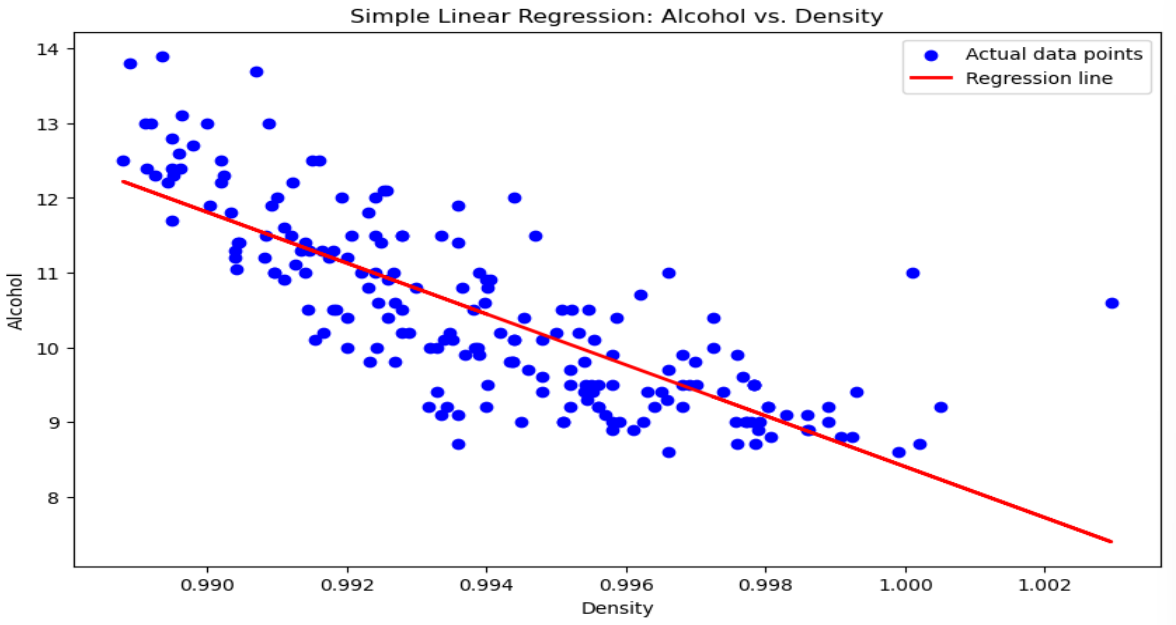
Graph reading: ← lower, higher →

As denoted by the graph above, there is a **negative relation** between the two variables, meaning that as density decreases, the alcohol content tends to increase. This is shown in the downward trend in the values across the density groups. The spread of alcohol content indicates that at lower densities (around 0.988), alcohol content tends to be higher, around the 12-13% mark, while the wines with a density greater than 0.995 contain a lower content of alcohol. This may be because denser liquids contain more dissolved sugars, which can indicate lower fermentation in the wine process.

Simple Linear Model

For this task, the question asks to create a linear model for the two variables. Alcohol being the dependent variable (y) and density as the independent variable. The linear model chosen is a **simple linear regression** model of the two variables.

I used the list and dataframe created in the previous question and assigned it to 2 variables, called X_new, y_new, which would perform the regression model analysis using model_new.fit(X_new, y_new).
After implementing the steps to building a linear regression model (taken from week 5 slides, page 22), this is the end result:



Similar to the boxplot, we find a **negative correlation** between the two variables, and as density increases, the alcohol content tends to decrease. The red line, representing a **line of best fit** for the relationship, indicates that the regression equation is:

Alcohol = 349.37 + (-340.97) x Density.

This equation indicates that for each unit increase in density, the alcohol content decreases by around 340.97 units.

The MSE (Mean Squared Error) of 0.58 is the average of the squared differences between the actual and predicted values, with generally a lower value being better.
The RMSE (Root Mean Squared Error equal to 0.76 represents the average difference between predicted and actual alcohol content. With it being fairly low, it indicates a reasonable fit.

Looking at the graph, there are a couple of dots which are plotted far from the rest of the data – indicating some potential outliers. Looking at it with the regression line, there are some variations in the distribution of the dots. This can potentially indicate that the relationship between alcohol and density may not fully explain the variation in alcohol content.

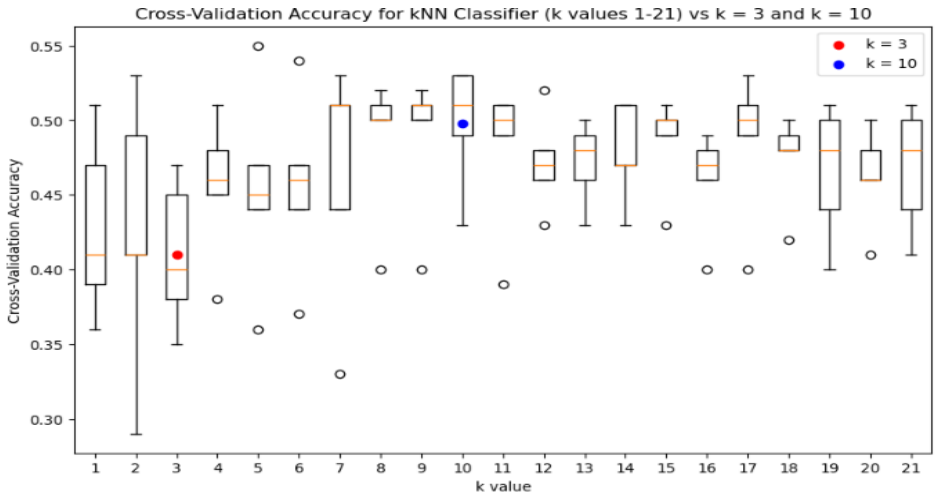
Task 2: Classification

kNN

after the implementation of the kNN algorithm, the chosen k value is 3. This is a small and reasonable starting point as its common to begin with a lower k-value before further doing research and determining the most effective value through optimisation methods such as cross validation.

K = 3 was chosen as it allows a more accurate representation of local data variations by focusing on closer neighbours. With it being an odd number, it helps avoid ties when determining majority class in classification problems. This factor helps mitigate any potential bias which could appear from starting with a higher k-value.

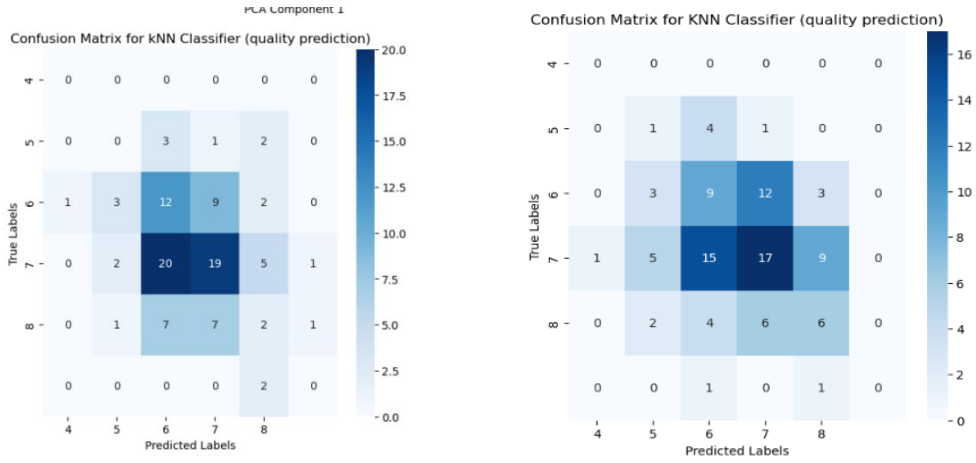
starting with a larger value, such as k = 15 can lead to underfitting and increase in potential bias, which can result in a simplistic model. A larger value be less sensitive to individual outliers as it averages more across data points.



As seen by this boxplot of a cross-validation accuracy between values of 1-21, we can see that the lower values have underfitting. This can be seen with the variance, as it is wider compared to the higher values. There is also overfitting seen in the data, as the model focuses on individual points too often. There is also a lower accuracy (excluding the couple of outliers) compared to the higher values. Later in the report, I will discuss the best k-value for this data set.

modified kNN

for this section, I decided to use the kNN algorithm, but in conjunction with PCA (Principle Component Analysis)
applying PCA reduces the dimensionality of the data set, identifying which directions the data varies the most. Reducing this value to 2 ensures that the algorithm focuses on the most important patterns of variance in the data, which ignores noise or dimensions which are of less importance.
Doing so also improves computational efficiency – making the kNN distance calculations faster, also improving the efficiency and performance of the kNN algorithm.



As you can see with two of the confusion matrices, **left** being the one with the modified kNN including PCA and **right** being the one without kNN.

The left image shows that the algorithm performs well in predicting the true label 7, getting 20 and 19 correct predictions respectively. There are a couple misclassifications present, such as when the model predicted 7 for a true label of 6, indicating that the model isn't perfect and some improvements can be made to further increase the accuracy of the algorithm.

The right image has a similar performance, where it shows 17 correct predictions for true label 7, however this is slightly lower than the enhanced version, indicating that PCA-modified algorithm has slightly better performance. Like the left image, it also misclassifies 9 instances as label 7, with misclassifications between label 6 and label 8. This model seems to have less high-confidence predictions which are correct compared to the algorithm enhanced with PCA.

Decision Tree (and Comparison)

In this question, I implemented a decision tree using the sklearn.tree module, with a split of 5.

While both models have a similar accuracy level of 0.33, the kNN classifier has a higher micro avg. precision by 0.04 (0.20), but the decision tree performs better for class 6, in relation to recall and f1-score. The kNN classifier performs better in terms of precision for class 7.

Classification Report:

	precision	recall	f1-score	support
3	0.00	0.00	0.00	0
4	0.09	0.17	0.12	6
5	0.27	0.33	0.30	27
6	0.47	0.36	0.41	47
7	0.32	0.33	0.32	18
8	0.00	0.00	0.00	2
accuracy			0.33	100
macro avg	0.19	0.20	0.19	100
weighted avg	0.36	0.33	0.34	100

Classification Report:

	precision	recall	f1-score	support
3	0.00	0.00	0.00	0
4	0.00	0.00	0.00	6
5	0.29	0.44	0.35	27
6	0.53	0.40	0.46	47
7	0.15	0.11	0.13	18
8	0.00	0.00	0.00	2
accuracy			0.33	100
macro avg	0.16	0.16	0.16	100
weighted avg	0.35	0.33	0.33	100

As you can see by the classification reports above, where left is kNN (not modified) and right is the Decision Tree Classifier, the right classification report hints slightly better in some key areas, such as class 6, but the kNN performs consistently across the different classes compared to the Decision Tree. This is shown in class 4 and 7, where the Decision Tree struggled to gain a classification compared to kNN, which managed to somewhat classify that class.

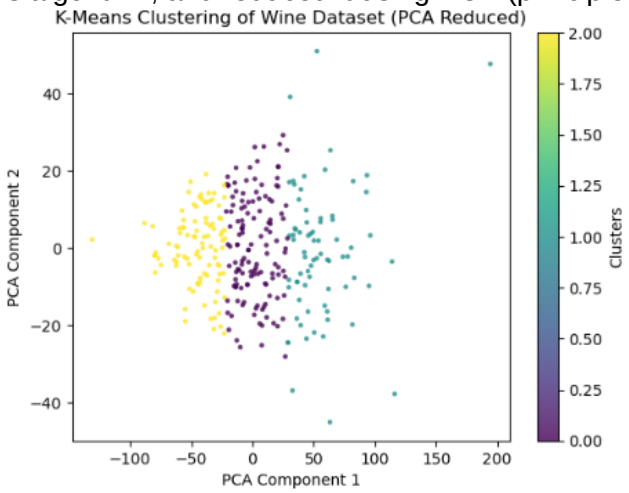
Exceptional performance in class 6 indicates that this path is more interpretable compared to others, which is a strength of using a Decision Tree. While poor performance in class 7 indicates overfitting, as they do not handle noisy data well unless pruned.

Still, neither model is performing well, thus more modifications/tuning may still be required to both algorithms. Methods such as cross-validation or grid search can help find the optimal k value, and techniques such as pruning or hyperparameter tuning to further improve performance of the decision tree.

Task 3: Clustering

k-means and PCA

I implemented the k-means algorithm, and reduced it using PCA (principle component analysis).



As seen in the algorithm above, we have 3 main clusters:

- Cluster 0: purple
- Cluster 1: yellow
- Cluster 2: green

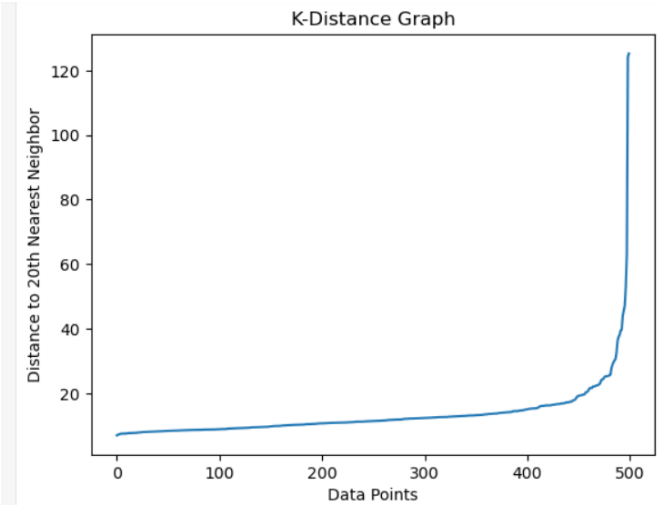
With the graph above showing the distribution of wine quality, within each of the three clusters.

Cluster 0 is dominated by wines at quality 6, containing 61 samples, with quality 5 not far behind at 32 samples, and quality 7 at 29 samples.

Cluster 1 also has 33 samples of quality 6 wines, as well as quality 5 and 7 wines at 29 and 26 samples respectively, containing fewer samples of quality 4 and 8.

Finally, Cluster 2 contains a large portion of quality 6 wines, slightly behind Cluster 0 at 50 samples, with quality 5 at 23 and quality 7 at 24 samples.

From the graph above, we can observe that quality 6 is dominant across all three clusters, indicating that quality 6 wines share similar features which lead to them being assigned into different clusters. The overlap of qualities 5, 6 and 7 wines indicate that the clusters aren't fully defined by wine quality, thus other factors such as chemical and physical characteristics (pH, acidity, etc) are more likely considering the outcome of the cluster than wine quality itself. The frequent overlap in the quality ratings across the clusters implies that it has separated the wines based on certain chemical properties, thus it determines that further analysis may be required to understand what drives the separation in the data which has been PCA reduced.



The k-distance graph above shows the most optimal value for the eps value of the DBSCAN algorithm (next question/prompt in report). a common way to determine this is to choose the 'elbow' point, that is where the graph decides to turn, which is at approximately 450 (45) data points.

DBSCAN (eps, minpts)

Implementation of the DBSCAN Algorithm helps identify clusters in the dataset based on the features of Alcohol, with quality as the target value. The EPS value is 6, as a value that is too small may classify too many points, while a value that is extremely large may may result in fewer neighbours, This is often a good point to start off with little information. This is discussed later in this section, but it is determined that an eps value of 11 is a better choice due to the positive silhouette score, and more better defined clusters.

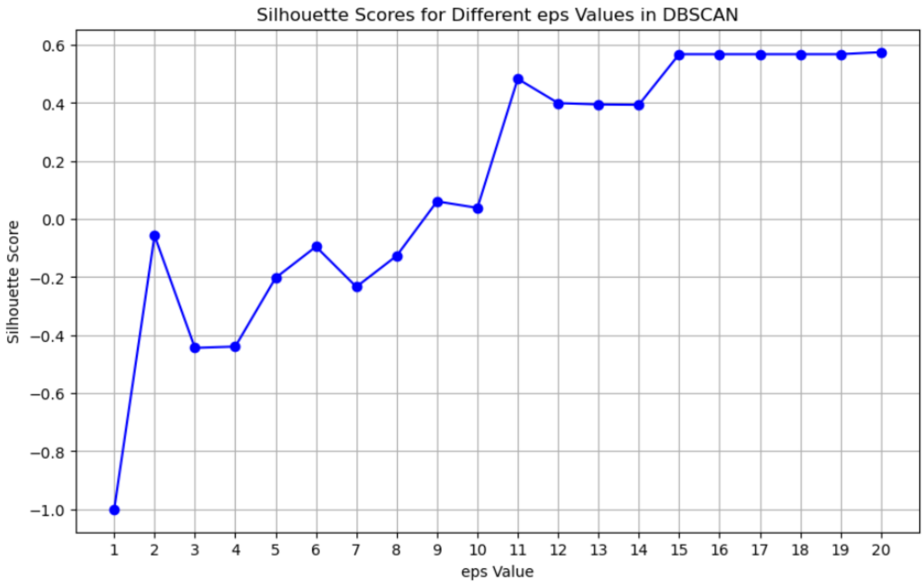
```
[ 0  1  4  2  3 14 -1 -1  4  3 -1  5  6  4  6  3 -1  7  3  3 -1  4 -1  4
 -1 -1 -1  8  9 -1  0  4 -1 -1 -1  3  3  3 -1  4  3  4  4  6 10 -1 15 10
  4  3  3  3  3  4  3  3  3  4 11 -1  4 12  3  3  3 11  5 -1 -1  4 10 -1
  4  5 13  3 -1  3 16  3  6  3 10 -1  4 -1  4 -1 -1 -1 10 12  4  4  4 -1
  4 -1 10 -1 14 -1 -1  3  7  6 -1  3 13  3 -1 10  8 -1 -1  7 -1  4  3 -1
  7  4  4 11 11  4  4  3  8  7  3  4 -1  4 -1 -1  3  3 -1  4 11 -1  1 -1
  3  3 -1 -1  4  3  3 -1 -1 -1  7 10 -1  3 -1 -1  4  3  3 -1 -1 11 -1 11
  2 -1  3 15 15 11 -1  4 10  3  0 17 10 -1 16  6 16 -1 -1  2  4  7 -1  3
 -1 -1  3  3 -1 -1  3  3  6 -1  3 15 -1  3  4  3  4 17 -1 10  3 10  6 17
 10 -1 -1 11  7  3 -1  3  3 -1 -1 -1  4 -1  1 -1  4 14 -1 -1 -1 -1 -1  9
 11 -1  3 -1  9 -1 -1 11  4 -1  4  3 -1 -1  4  6  1  9 -1 -1 -1  7 -1 12
  6 -1 -1  3 -1 -1  6 -1 -1  9 -1 -1  5  3  8  4 16 -1  3 13 -1 -1  6  3
  4 12  3 -1 -1 -1 -1 -1  4 -1 -1  4]
```

As seen by the array generated above, there is plenty of noise present, thus they don't belong to any specified cluster and indicates that DBSCAN has a significant portion of the dataset as noise. A factor contributing to this could be the value of eps or min_sample. (discussed later)

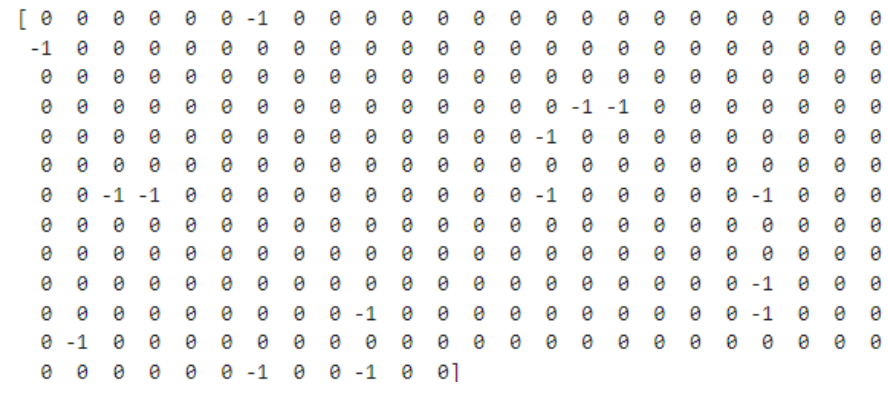
The other points which are denoted as 0, 1, 3, 4, etc are values which represent actual clusters formed by the algorithm. This algorithm doesn't find many clear, defined clusters, which suggests that the eps value might not be optimal for this dataset.

```
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0]
```

However, as seen with the graph where eps = 45, (above) we have almost all the data sets clustered, with one anomaly present in the data. This suggests that a eps too high has a single cluster, while too low has a lot of noise.



I generated a graph which displays eps values and its accuracy with a graph, shown above. This graph shows an optimal performance at eps = 11. Compared to eps = 6, eps 11 shows more a more accurate silhouette score which is important in created better clusters, making the DBSCAN algorithm more effective. This makes the data structure clearer to the user and leads to fewer outliers in the data compared to eps = 6. (image below)



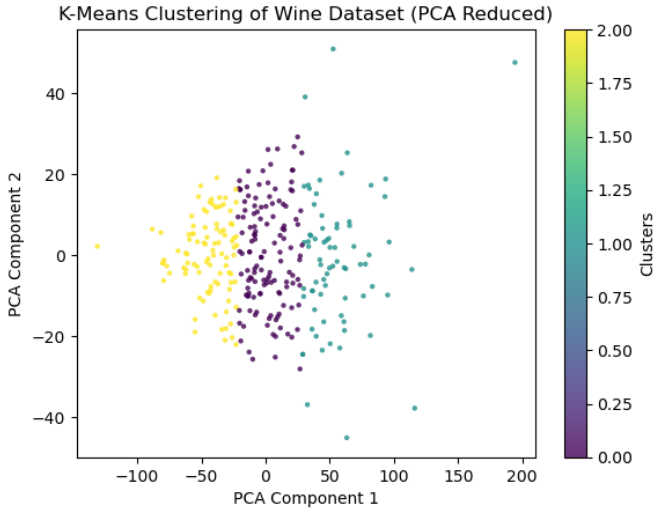
The min_samples (minPts) value is 3, which allows the formation of clusters with just 3 neighbours. A lower value makes the criteria less strict, thus this increases the chances of identifying clusters in areas where the density of points is low, which can be beneficial for this data set as the values are quite spread out. This makes the DBSCAN algorithm more sensitive to smaller patters / local structures in the data, which is ignored with higher values. Furthermore, an EPS value of 1 or 2 can increase over-clustering. Setting it to 3 ensures balance by reducing over-clustering and capturing important patterns.

Comparison between the k-means and DBSCAN algorithm.

The performance between k-means and DBSCAN is similar, however it depends on your goal.

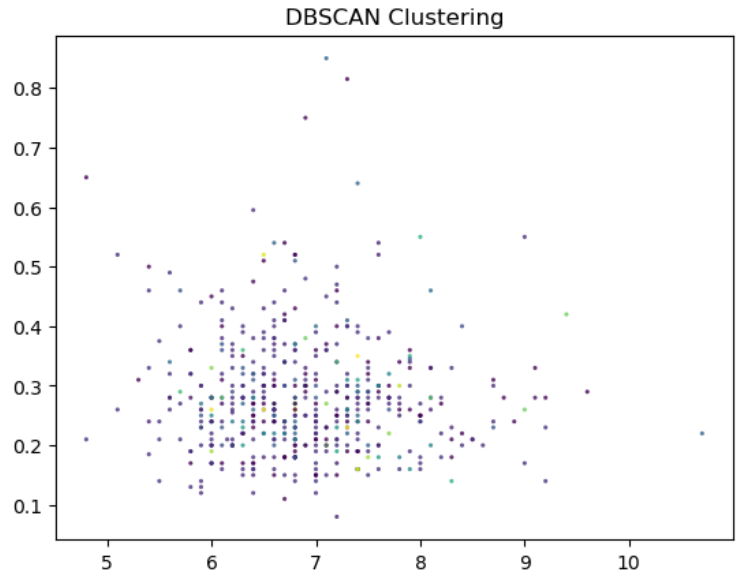
While DBSCAN identifies multiple points as noise (-1) and forms new clusters (0), while k-means partitions into exactly 3 clusters, regardless of density or distance. DBSCAN identifies noise points which aren't dense enough to be apart of a cluster, making it more robust to outliers., while k-means doesn't deal with noise or outliers as it assigns every point to one of the 3 clusters, regardless of whether they belong or not.

DBSCAN is better when there are clear areas of dense points, however with this dataset it seems to capture a lot of the data in one large cluster (0), with several classified as noise (-1)
The K-Means algorithm clearly partitions the data into 3 clusters which is more appropriate for this dataset given the lack of strong density differences.
Thus for the wine quality data set, it would be best to use k-means, as we do need a more structured separation.




```
[2 0 2 0 0 2 1 1 2 0 2 0 0 2 1 0 1 1 2 0 0 2 2 2 1 1 0 0 0 0 2 2 1 1 0 2 0
 0 2 2 0 2 2 0 2 1 0 2 2 0 0 0 0 2 0 0 0 2 0 1 2 0 2 0 2 0 0 0 1 2 2 1 2 0
 2 0 1 0 1 0 1 0 2 0 2 1 2 1 1 0 2 0 2 2 2 1 2 1 2 1 2 0 1 0 1 1 0 0 2 0 1
 2 0 2 0 1 2 2 0 1 1 2 2 0 0 2 2 0 0 1 2 2 0 2 2 1 0 0 0 2 0 2 0 1 0 0 2 1
 2 0 2 2 1 0 1 2 0 0 1 0 2 0 0 2 1 0 2 0 1 1 2 0 0 0 1 2 2 0 2 0 2 0 1 0 1
 0 0 1 2 1 2 0 0 2 0 0 0 2 0 0 0 0 0 0 1 0 2 0 2 0 2 2 2 2 1 0 2 2 0 0 1 0
 1 0 0 1 1 2 2 0 0 0 2 2 2 0 1 1 0 0 0 1 0 0 0 1 1 0 2 1 2 0 1 1 2 1 0 0 0
 1 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 1 0 0 0 2 1 0 0 2 0 2 0 0 2 0 2 2 1 0 1 0
 2 1 2 2]
```

(k-means clustering: separated into 3 distinguishable clusters, with minimal noise or outliers.)



(DBSCAN: less clusters but shows outliers clearly and identifies noise in the dataset.)

References:

In addition to my Practical Teacher, **Chengyao Xie,** and resources provided from RMIT via the modules on Canvas, I also used these sources:

External References (eg. Websites, official documentation, youtube explained videos, etc):

scikit-learn. (n.d.). *sklearn.model_selection.train_test_split*. scikit-learn. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

GeeksforGeeks. (n.d.). *How to randomly select rows from pandas DataFrame?*. Retrieved from https://www.geeksforgeeks.org/how-to-randomly-select-rows-from-pandas-dataframe/

Stack Exchange. (n.d.). *Train test split error: Found input variables with inconsistent numbers of samples*. Data Science Stack Exchange. Retrieved from https://datascience.stackexchange.com/questions/20199/train-test-split-error-found-input-variables-with-inconsistent-numbers-of-sam

Medium. (n.d.). *Simple linear regression*. Geek Culture on Medium. Retrieved from https://medium.com/geekculture/simple-linear-regression-bd4348e1ee62

statsmodels. (n.d.). *Statsmodels: Statistics in Python*. statsmodels. Retrieved from https://www.statsmodels.org/stable/index.html

scikit-learn. (n.d.). *Linear models* (Documentation). scikit-learn. Retrieved from https://scikit-learn.org/stable/modules/linear_model.html

scikit-learn. (n.d.). *sklearn.linear_model.LinearRegression*. scikit-learn. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

StatQuest. (n.d.). *Clustering with DBSCAN, clearly explained!!!!* [Video]. YouTube. Retrieved from https://www.youtube.com/watch?v=RDZUdRSDOok

StatQuest. (n.d.). *Supervised and unsupervised learning, clearly explained!!!!* [Video]. YouTube. Retrieved from https://www.youtube.com/watch?v=v5CcxPiYSlA

Aksakal, K. (n.d.). *Confusion matrix with KNN algorithm*. Kaggle. Retrieved from https://www.kaggle.com/code/kadriyeaksakal/confusion-matrix-with-knn-algorithm

Stack Overflow. (n.d.). *How to include a confusion matrix for a KNN in Python?*. Retrieved from https://stackoverflow.com/questions/60748497/how-to-include-a-confusion-matrix-for-a-knn-in-python

scikit-learn. (n.d.). *sklearn.model_selection.GridSearchCV*. scikit-learn. Retrieved from https://scikit-learn.org/dev/modules/generated/sklearn.model_selection.GridSearchCV.html

scikit-learn. (n.d.). *sklearn.model_selection.GridSearchCV (v0.18)*. scikit-learn. Retrieved from https://scikit-learn.org/0.18/modules/generated/sklearn.model_selection.GridSearchCV.html

Great Learning. (n.d.). *GridSearchCV: A guide on how to fine-tune your machine learning models*. Great Learning. Retrieved from https://www.mygreatlearning.com/blog/gridsearchcv/

IBM. (n.d.). *What is k-nearest neighbors (kNN)?* IBM. Retrieved from https://www.ibm.com/topics/knn#:~:text=The%20choice%20of%20k%20will,optimal%20k%20for%20your%20dataset.

GeeksforGeeks. (n.d.). *How to find the optimal value of k in kNN?* Retrieved from https://www.geeksforgeeks.org/how-to-find-the-optimal-value-of-k-in-knn/

Codecademy. (2024). *K-nearest neighbors cheatsheet*. Codecademy. Retrieved from https://www.codecademy.com/learn/introduction-to-supervised-learning-skill-path-2024/modules/k-nearest-neighbors-skill-path/cheatsheet

IBM. (n.d.). *K-nearest neighbors (KNN) in machine learning*. IBM. Retrieved from https://www.ibm.com/topics/knn#:~:text=Lower%20values%20of%20k%20can%20overfit%20the%20data%2C%20whereas%20higher,it%20can%20underfit%20the%20data

Reddit. (2022). In k nearest neighbor models, why does a smaller k tend to lead to overfitting? *r/AskStatistics*. Retrieved from

https://www.reddit.com/r/AskStatistics/comments/yvlfti/in_k_nearest_neighbor_models_why_does_a_smaller_k/

Internal References (eg. Modules, Labs, etc):

RMIT University. (n.d.). *Document available from RMIT Instructure Course 125162, file 41079936*. Retrieved from https://rmit.instructure.com/courses/125162/files/41079936?module_item_id=6649699
RMIT University. (n.d.). *Document available from RMIT Instructure Course 125162, file 40958914*. Retrieved from https://rmit.instructure.com/courses/125162/files/40958914?module_item_id=6638409
RMIT University. (n.d.). *Document available from RMIT Instructure Course 125162, file 40605855*. Retrieved from https://rmit.instructure.com/courses/125162/files/40605855?module_item_id=6602561
RMIT University. (n.d.). *Document available from RMIT Instructure Course 125162, file 40725423*. Retrieved from https://rmit.instructure.com/courses/125162/files/40725423?module_item_id=6610405