Data Preparation and Analysis

Task 1.1

Import Part

I am importing
import pandas as pd - This line imports Pandas library and and assigns it the alias 'pd'.
Import numpy as np - This line imports numpy library and and assigns it the alias 'np'.
seaborn as sns - This line imports seaborn library and assigns it the alias ‘sns’ .
matplotlib.pyplot as plt - This line imports matplotlib.pyplot library and and assigns it the alias 'plt'.
statsmodels.api as sm - This line imports statsmodels.api library and and assigns it the alias 'sm'.

I am using pd.read_csv function to read the data from a CSV file named 'A2data.csv' and stores it in a DataFrame called 'wine_data'.

wine_data.head() - By default this shows first 6 columns of wine_data

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.0	0.27	0.36	20.7	0.045	45	170	1.001	3.00	0.45	8.8	6
1	6.3	0.30	0.34	1.6	0.049	14	132	0.994	3.30	0.49	9.5	6
2	8.1	0.28	0.40	6.9	0.050	30	97	0.9951	3.26	0.44	10.1	6
3	7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.40	9.9	6
4	7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.40	9.9	6

duplicated() - I am checking duplicate count in the dataframe and then printing total duplicates in the dataframe

drop.duplicates() - I am dropping all the duplicate rows from the dataframe.

dtypes - This is used to obtain data types of each column in DataFrame named wine_data

pd.to_numeric - This is used to convert object datatype to float64 datatype.We are converting residual sugar, free sulfur dioxide, total sulfur dioxide, density variables which are object based datatype to float64 datatype. We are viewing the converted datatype using dtypes .

missing_data = wine_data.isnull().sum() - isnull() function is used to check null values present in each column of the dataframe.I could see 9 columns having null values.

dropna() - This is used to drop null values from all the 9 columns and I am storing the entire dataframe to data_cleaned dataframe. I am using .isnull() function again to view the presence of null values and all the null values present in the column are removed now.

I am importing stats module from scipy library. We are calculating z-scores for each data point in data_cleaned dataframe. Stats.zscore is used to prevent the data to have a mean of 0 and SD of 1. np.abs is used to ensure all z-score are positive.If z-score for each row in df is greater than 3. If any z-score in a row exceeds 3, corresponding row is marked as outlier and we will remove from dataframe. I am showing the output through Boxplots.

I am taking 600 random observations from the cleaned dataframe data_cleaned using a random seed 44 and storing them in a new dataframe named random_sample.

I am saving the cleaned random_sample dataframe to a CSV file named A2RandomSample.csv in the current working directory.The index=False argument is used to prevent writing the DataFrame's index as a separate column in the CSV file.

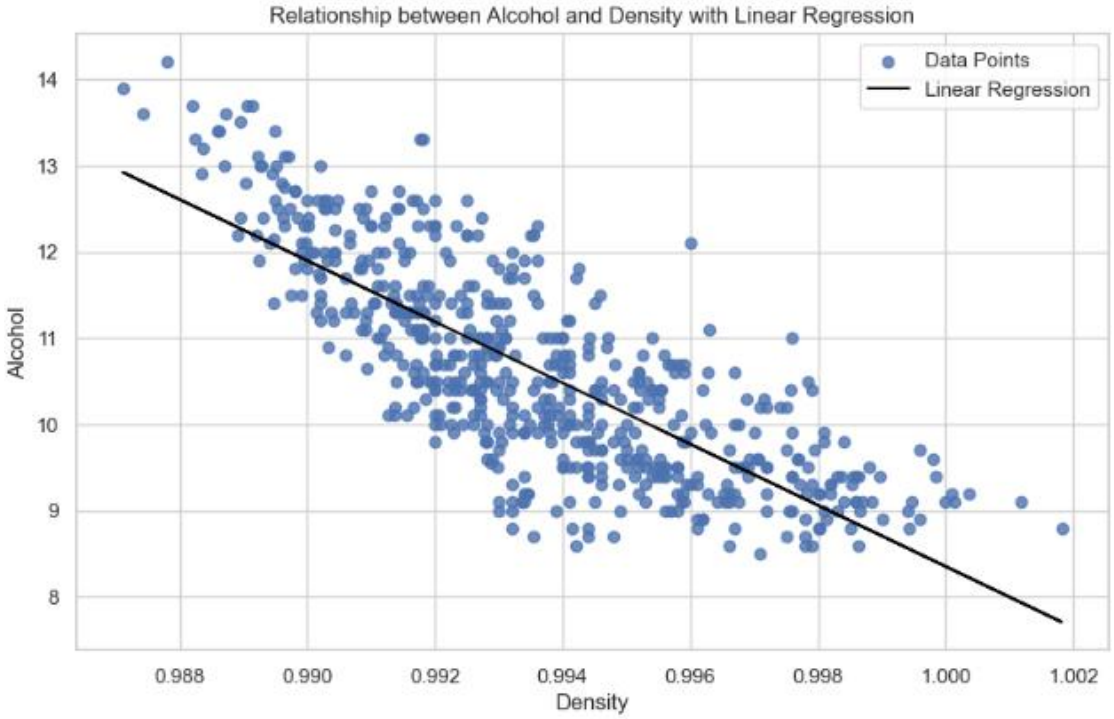
I am reading the CSV file A2RandomSample.csv and storing them in a new Dataframe named sample. I am viewing the count of rows and columns present in sample.

I am using value_counts() function to count the unique values in the sample DataFrame.

Task 1.2

I am using scatterplot to plot the difference between density and alcohol. The data is collected from sample dataframe with xlabel being Density with Independent Variable and ylabel being Alcohol with dependent Variable and the title is Scatter Plot of Alcohol vs Density.We are using matplotlib library to show the scatterplot.

X_data is an array of density from sample dataframe and y_data is an array of alcohol from sample dataframe.
I am importing LinearRegression class from sklearn library.
We are creating a linear regression model object and we are fitting it to our data. Fit function trains the linear regression model to find best-fitting line.
intercept = model_cleaned.intercept_ - We are retrieving intercept term of linear regression model.
slope = model_cleaned.coef_[0] - We are retrieving the coefficients for the features used in the linear regression model.
After running the above lines of code, I have trained a linear regression model (model_cleaned) on my data, and I have extracted intercept and slope (coefficient) of the linear regression equation.

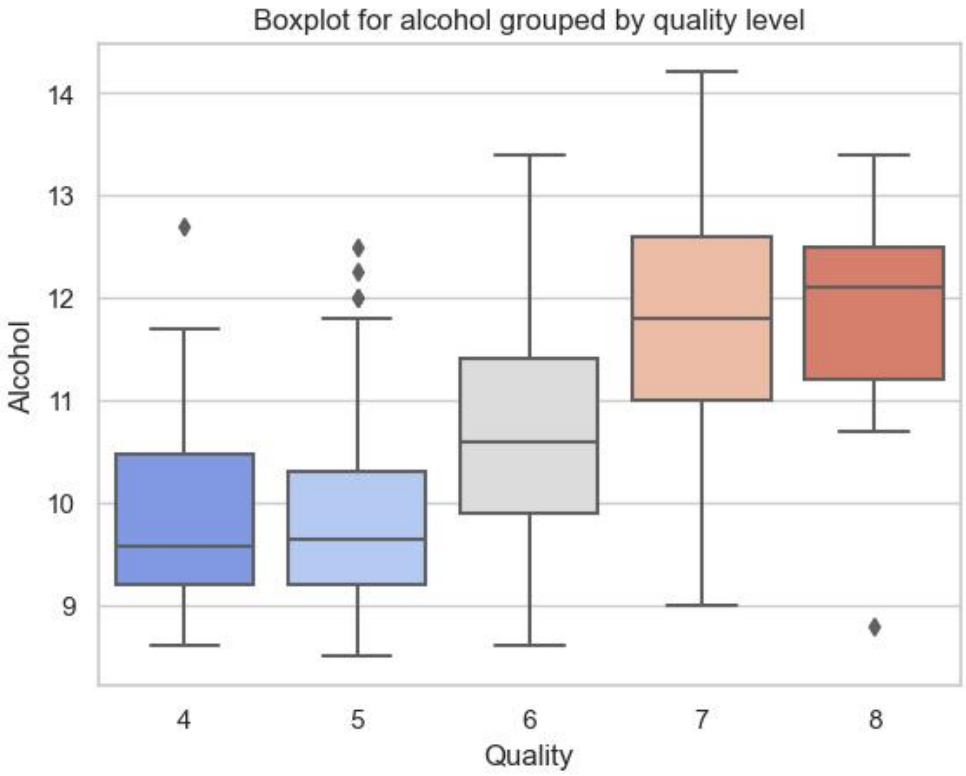


plt.plot(sample['density'], intercept + slope * sample['density'], color='black', label='Linear Regression') - This line augments the plot with a linear regression line. The intercept and slope figures from the linear regression model are used. The x-axis is made up of sample['density'] values, and for each x-value, a simple linear regression model's equation is utilized to determine the associated y-value. A label and the legend are added, and the line is made in black.

A scatter of data points for "Density" and "Alcohol" are displayed in the plot, along with a straight line showing how well the linear regression model fits the data. We could see a lot of scattered dots in the middle part of the regression line.

Task 1.3

We are creating a boxplot for alcohol grouped by quality level.
Median Trend: Wines with quality rating of 4 and 5 have lower Median alcohol content compared to other wine quality. Wine with quality rating of 8 has highest median range. This suggests more variability in the alcohol content for wines within these quality categories.
Outliers: Wine with quality rating of 4, 5 and 8 have some outliers on the higher end and lower end of the alcohol content spectrum.
In conclusion, there are exceptions and differences within each grade level. This implies that although alcohol content may affect wine quality, it's probably not the primary factor. The wine's quality rating is also influenced by additional elements and qualities.
We could see that Quality 6 and 7 have equal amount of alcohol content and quality based on median range and IQR.



Task 2: Classification

Task 2.1

I am importing necessary libraries and I am suppressing UserWarnings. I am loading dataset from CSV file A2RandomSample.csv into sample dataframe.I am removing quality column from sample dataframe as it is not the same datatype. I am splitting the dataset into features and target variables. I am scaling the features using StandardScaler function to have a mean of 0 and SD of 1. I am splitting the data into training and testing sets where 70% of data is used for training and 30% for testing. I am training a K-NN model with n_neighbors parameter=5 and fitting it to training data. I am using trained KNN model to predict labels for test data. I am calculating and printing the accuracy, precision and recall on models's prediction. I am computing a confusion matrix and developing a classification report which provides summary of various classification metrics for each class, including precision, recall, and F1-score. By doing this I am getting the accuracy and recall value to be 0.5056 and precision value to be 0.4714.

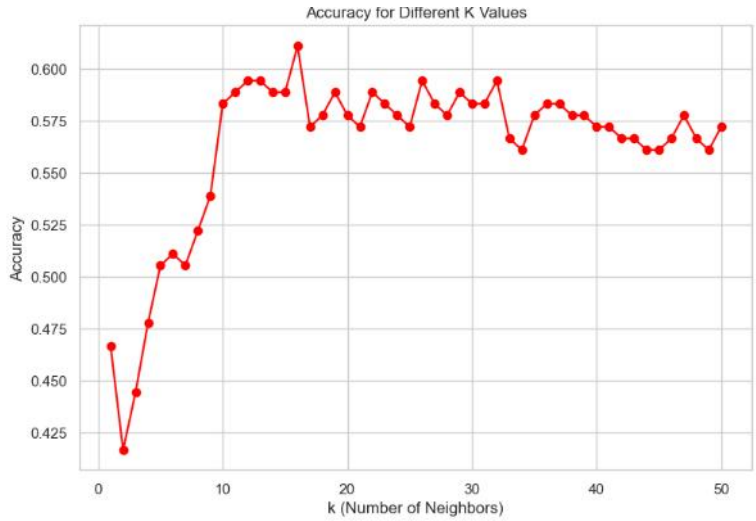
```
accuracy = 0.5056
precision = 0.4714
recall = 0.5056
confusion matrix:
[[ 0  6  2  0  0]
 [ 0 26 13  2  0]
 [ 1 25 56  6  1]
 [ 0  5 21  9  0]
 [ 0  0  4  3  0]]
              precision    recall  f1-score   support

           4       0.00      0.00      0.00         8
           5       0.42      0.63      0.50        41
           6       0.58      0.63      0.61        89
           7       0.45      0.26      0.33        35
           8       0.00      0.00      0.00         7

    accuracy                           0.51       180
   macro avg       0.29      0.30      0.29       180
weighted avg       0.47      0.51      0.48       180
```

Task 2.2

I am choosing a k_range with values from 1 to 51. I am creating an empty list named accuracy which will store all the accuracy score of different k values. I am using a for loop to iterate through k range of values,training and evaluating KNN classifier.I am using n_neighbors parameter here. For each k value in range , it trains a KNN with that k value, makes predictions and calculates accuracy. Accuracy scores are then appended to the list.Highest accuracy score in the list corresponds to Best_k value. I am plotting a line graph to visualize the accuracy of different k values. Finally, I am printing out the best k value based on accuracy. From the graph, we can clearly see that the data point between 10 and 20 has the highest accuracy and this justifies my code.

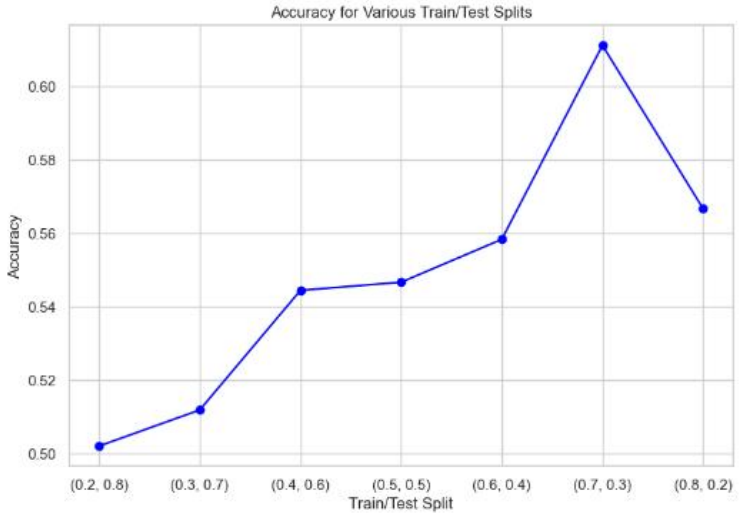



```
Best k value based on accuracy is: 16
```

Task 2.3

I am creating a tuple that contains diff_split list, each specifying different train/test split. I am initializing an empty list split_accuracy to store accuracy of KNN classifier for each train/test split. I am using for loop to iterate through various train/test splits.

I am splitting scaled data into training and testing sets according to various splits. I am initializing KNN classifier with best_k and I am using n-neighbors parameter. I am calculating the accuracy of the classifier on the test data and appends it to the split_accuracy list. I am plotting the accuracy for each train/test split using Matplotlib, showing the accuracy for each split on the y-axis and the split ratios on the x-axis. Determines the best train/test split based on the highest accuracy and prints the best split ratio.After plotting the train/test split we can see that the best split based on highest accuracy is (0.7, 0.3).



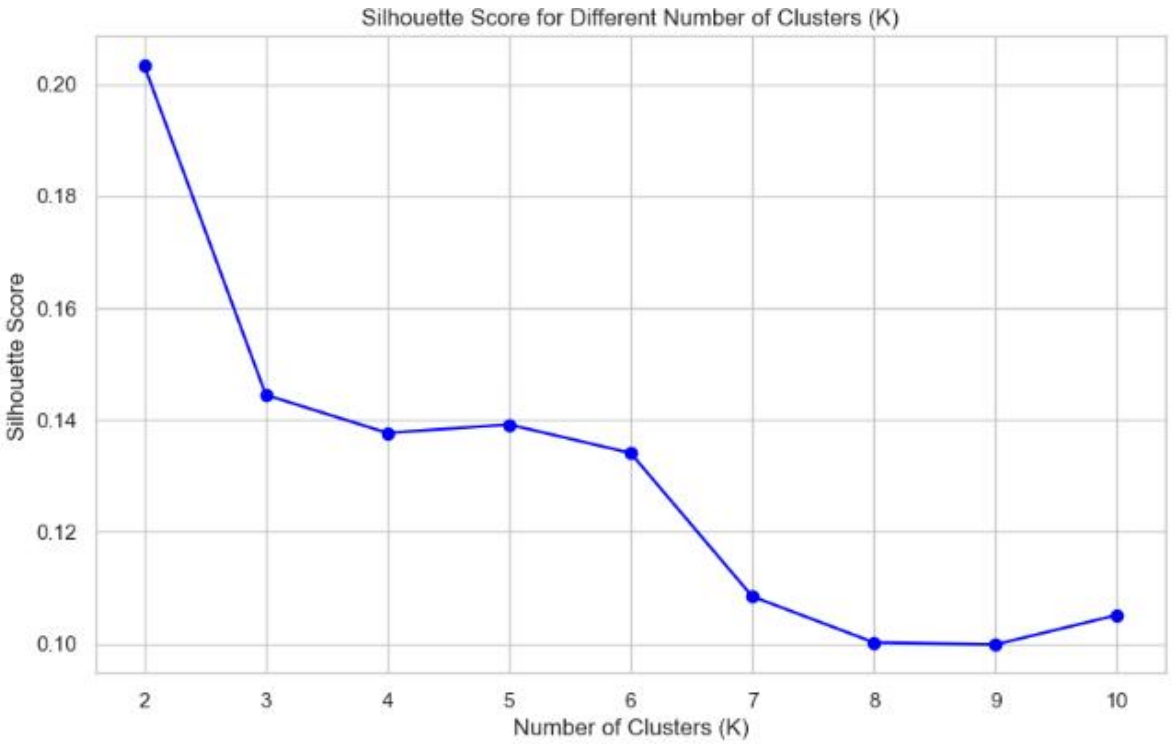

```
Best split based on highest accuracy is: (0.7, 0.3)
```

Task 3: Clustering

Task 3.1

I am importing necessary libraries for K-Means clustering, for calculating Silhouette score and for data preprocessing. I am loading the dataset from a CSV file. I am dropping Quality column from sample dataframe as mentioned in the condition. I am standardizing the data to have 0 mean and Unit variance. I am defining a range of cluster numbers from 2 to 11. I am creating a empty list named silhouette_scores. For each value of K in the k_range, the code will create K-Means clustering model with the current K value. I am fitting the model to standardized data. Calculating the Silhouette score, which measures the quality of clustering, based on how well data points are grouped together. I am plotting the Silhouette score in a line graph for different k values. From the highest Silhouette score, I am selecting best k value . With the best k value, I am building a final k-means model and I am fitting it to standardized data for final clustering result.I am finally printing the best k value.

Silhouette AvgScore is : 0.10510550585896741
The chosen number of clusters is 2.



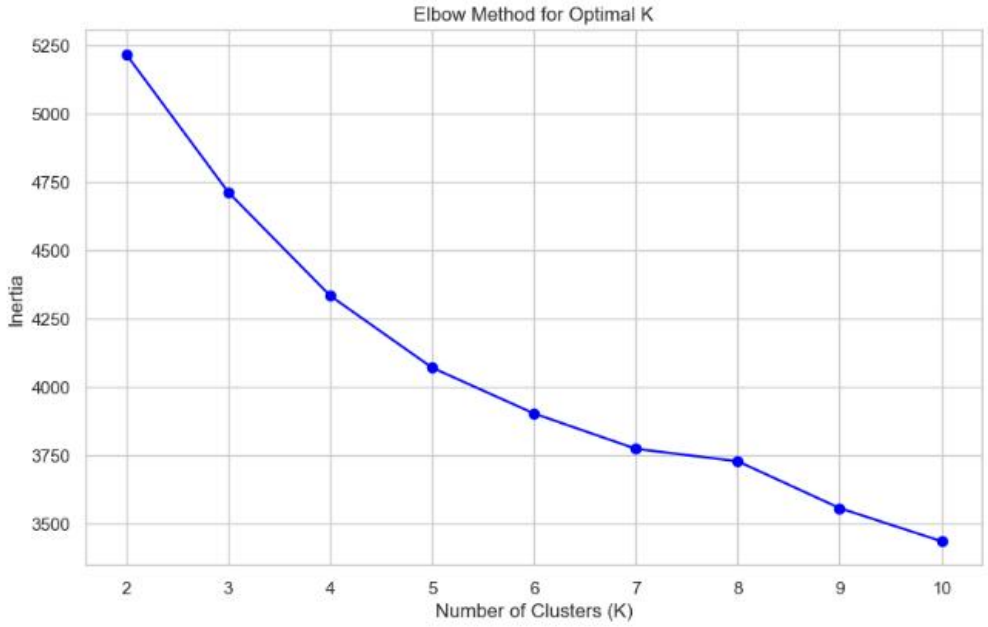
Task 3.2

I am using Elbow Method to determine the optimal number of clusters (K) for a K-Means clustering algorithm. The Elbow Method is a technique to find the point on the plot of the sum of squared distances (inertia) where the rate of decrease starts to slow down, indicating a good K value. I am importing necessary libraries. I am setting an empty list named inertia_values. I am creating a list named k_values with range from 2 to 11. For each K value, I am creating a K-Means clustering model with the current K value. I am fitting the model to the standardized data. Calculating the inertia, which represents the sum of squared distances of data points to their nearest cluster centers. I am plotting Elbow Method Curve using a line graph. This code generates a plot that shows the inertia values for different K values. The point at which the rate of decrease in inertia slows down indicates the optimal K value. We can visually see the plot and identify elbow point where inertia curve begins to bend. I can clearly see that the bend starts at 2. Best K value using the Elbow method is found using the formula

best_k_elbow = np.argmin(np.diff(inertia_values)) + 2

np.argmin find index of min diff
I am adding 2 to the result because our range starts from k=2
I am calculating differences between consecutive inertia.

I am finally printing out the best k value using Elbow method which is 2.




```
The best K value using the Elbow Method is: 2
```

Task 3.3

I am importing all the necessary libraries and loading sample dataframe. I am extracting features by selecting all columns except first column. I am setting optimal number of clusters to 2. The K-Means clustering algorithm is applied with specified number of clusters (optimal_k). The predict method assigns each data point to one of the K clusters based on the fitted K-Means model. I am extracting true quality levels from 'quality' column from sample dataframe. I am using the confusion_matrix function to create confusion matrix that compares true quality levels (labels) with the cluster labels assigned by K-Means and finally I am printing out the confusion matrix. Row 1 shows first true quality level . Column 1 represents first cluster label.

```
confusion matrix:
[[  0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0]
 [ 17   9   0   0   0   0   0]
 [ 88  90   0   0   0   0   0]
 [155 115   0   0   0   0   0]
 [ 85  20   0   0   0   0   0]
 [ 14   7   0   0   0   0   0]]
```

Ref:-

1.“pandas.DataFrame.dropna”
URL https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.dropna.html
2. “pandas.DataFrame.duplicated”
URL
https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.duplicated.html
3. URL https://www.educative.io/answers/how-to-identify-outliers-in-a-dataset-using-scipy-in-python

4. Jake VanderPlas(November 2016)'Python Data Science Handbook' URL https://jakevdp.github.io/PythonDataScienceHandbook/03.04-missing-values.html
5. https://rmit.instructure.com/courses/107387/files/33468051?module_item_id=5521696
6. https://rmit.instructure.com/courses/107387/files/33596556?module_item_id=5538966
7. https://rmit.instructure.com/courses/107387/files/33822932?module_item_id=5568719
8. https://rmit.instructure.com/courses/107387/files/33941011?module_item_id=5584237