Project 1

a. Data Preprocessing:

In the *winequality-white-v3* dataset, there are 7 variables: Alcohol, density, chlorides, total sulfur dioxide, citric acid, residual sugar, and quality(class variable). Some data preprocesses were implemented which are as follows:

- There were missing values in chlorides column, and I replaced the missing values by mean that is *0.046286*. Rest all columns remained the same.
- Now after replacing the missing values another data issue came which was additional rows at the end of the file. I used .isna() and .drop() functions to identify the empty rows in alcohol column and droped all the rows with na values(at the end of file).
- Also I had to transform the class variables from continuous to discreet values for further processing(from '5' and '7' to 0 and 1) using user defined function *transformQuality*.

When I tried to check the missing values in the columns in data set I found that it can be difficult for a data analyst to manually see each and every column before applying data cleaning. Like for instance I had to look for missing values in *winequality-white-v3* dataset and found chlorides have missing values. But what if some other columns too has a missing values but may be only a few values? So best solution can be to sort the data in a column and all the missing values will be sorted down to a group and then can be eliminted. Or we can include all the columns in the condition checking missing columns(*df0["chlorides"].where(~ missingchlorides, meanchlorides, inplace=True)*).

b. Wine quality prediction:

i. In this part I have defined all the variables: the features and class variable (quality). I have split the data into training (80%) and testing data (20%). I have used SklTreeLearner to create a decision tree. Also created a external prediction function having decision tree and input.

ii. In this part I have compared the performance of the tree by comparing with actual class. I have used accuracy, scoring.CA(),scoring.AUC() evaluation methods to comptue the performance of the tree.

I tried to use confusion matrix but it shows an error.

I ran the decision tree many times to see how much accuacy is obtained with the current model. The results vary every time we run as it shuffles data but model show an approximate accuracy of Accuracy = 0.824

Accuracy: 0.838

AUC: 0.828

Here scoring.CA method shows a higher accuracy. However all are very close.

Experimental part

I tried changing the variables we are considering as features for our model. I found an insteresting observation that some features(alcohol, density, chlorides) are more effective in predictng wine quality. I tried testing model by taking variables from 1 to 6. There could be 6! Combinations to test but I choose to take only simple ones. Here are the results I found.

1. feature_vars = list(Filtered_data.domain.variables[1:2])
   Accuracy = 0.802, Accuracy: 0.803, AUC: 0.857
2. feature_vars = list(Filtered_data.domain.variables[1:3])
   Accuracy = 0.840, Accuracy: 0.827, AUC: 0.829
3. feature_vars = list(Filtered_data.domain.variables[1:4])
   Accuracy = 0.848, Accuracy: 0.832, AUC: 0.823
4. feature_vars = list(Filtered_data.domain.variables[1:5])
   Accuracy = 0.842, Accuracy: 0.828, AUC: 0.819
5. feature_vars = list(Filtered_data.domain.variables[1:6])
   Accuracy = 0.834, Accuracy: 0.835, AUC: 0.829

6. feature_vars = list(Filtered_data.domain.variables[5:6])
   Accuracy = 0.666, Accuracy: 0.662, AUC: 0.694
7. feature_vars = list(Filtered_data.domain.variables[4:6])
   Accuracy = 0.725, Accuracy: 0.737, AUC: 0.742
8. feature_vars = list(Filtered_data.domain.variables[3:6])
   Accuracy = 0.711, Accuracy: 0.762, AUC: 0.750
9. feature_vars = list(Filtered_data.domain.variables[2:6])
   Accuracy = 0.816, Accuracy: 0.810, AUC: 0.800
10. feature_vars = list(Filtered_data.domain.variables[1:6])
    Accuracy = 0.789, Accuracy: 0.827, AUC: 0.820

    This experiment clearly shows that variables from 3:6 are not as important as variables 1 and 2 in predicting the quality of wine. So we can just use ‘alcohol’ and ‘density’ to predict the quality of the wine.