

## Warmup Exercise Report

### Sanjeeb Humagain

Imported libraries to use in the code for linear regression and classification. Following libraries are used.

---

1. Pandas
2. OS
3. Numpy
4. Seaborn
5. Sklearn
6. Matplotlib.pyplot
7. Different modules of sklearn

```
# import the libraries
import pandas as pd
# import the OS module
import os
import numpy as np
# import torch as torch
import seaborn as sb
import sklearn as sk
import matplotlib.pyplot as plt

# modules used for Linear Regression model
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score as r2 # used for LR
from sklearn.metrics import mean_squared_error as mse # used for LR
from sklearn.metrics import explained_variance_score as evs # used for LR
from sklearn import preprocessing
from sklearn import svm
from sklearn.metrics import accuracy_score as accuracy # for Classification
from sklearn.metrics import confusion_matrix as cm # for Classification
from sklearn.metrics import ConfusionMatrixDisplay as showMatrix # for Classification

# modules used for Classification
import seaborn as sns
```

Importing the dataset from the excel file.

---

```
# import the dataset for red wine from excel file
filepath = 'winequality-red.csv'
dataset = pd.read_csv(filepath, sep = ';')
```

```
dataset.info()
# describe is a descriptive statistics and using the following way makes it
easier to compare
dataset.describe().T
```

**Splitting the training and test data using `train_test_split` function from sklearn library.**

---

```
# lets split the data into training and test by using train_test_split
x_train, x_test, y_train, y_test = train_test_split(dataset, qual_label,
train_size=.75, shuffle=True, random_state=0)
x_train.shape, x_test.shape #dimension of the array will be displayed
```

**Linear Regression model was created using `LinearRegression()` module from sklearn library.**

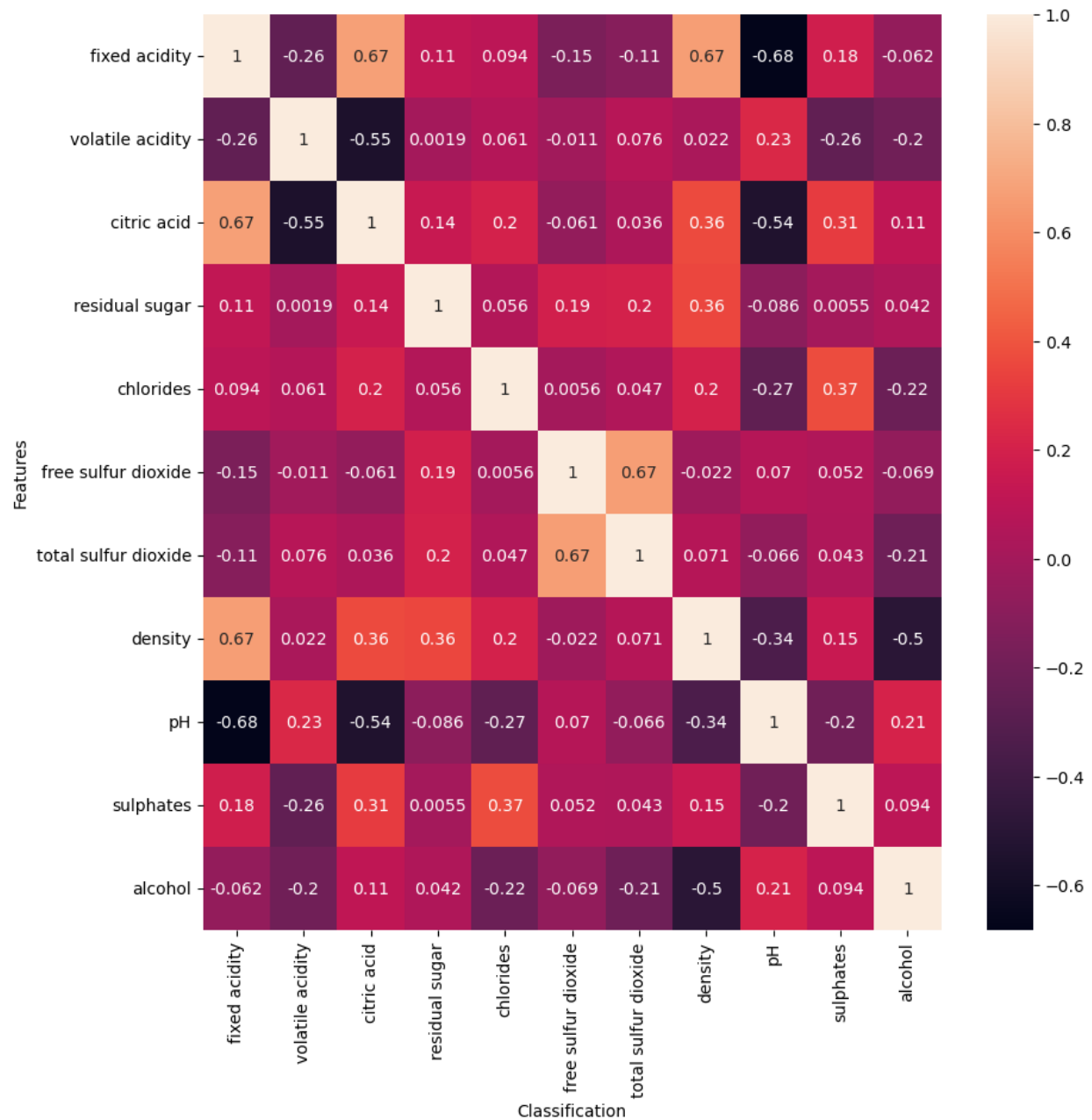
---

```
y_test = y_test.to_numpy() # before converting to numpy, the x range of plot
was extending up to 1600
lin_reg = LinearRegression().fit(x_train,y_train) # the lin_reg contains the
trained model
lin_predict = lin_reg.predict(x_test) # predicted value will be stored in
lin_predict
lin_modelscore = lin_reg.score(x_train,y_train) # Return the coefficient of
determination of the prediction
lin_R2 = r2(y_test,lin_predict) # R2 score of the model
lin_explained_var = evs(y_test,lin_predict) # explained variance ratio score
lin_mse = mse(y_test,lin_predict) # mse between the predicted value and the
true value
```

### Accuracy and Errors;

```
The Training Accuracy Score of the Regression Model: 0.3632493675603261
The R2 Score of the Regression Model: 0.34542438124562913
The Explained Variance Score of the Regression Model: 0.35006835112199897
The MSE of the Regression Model: 0.40007252708505425
```

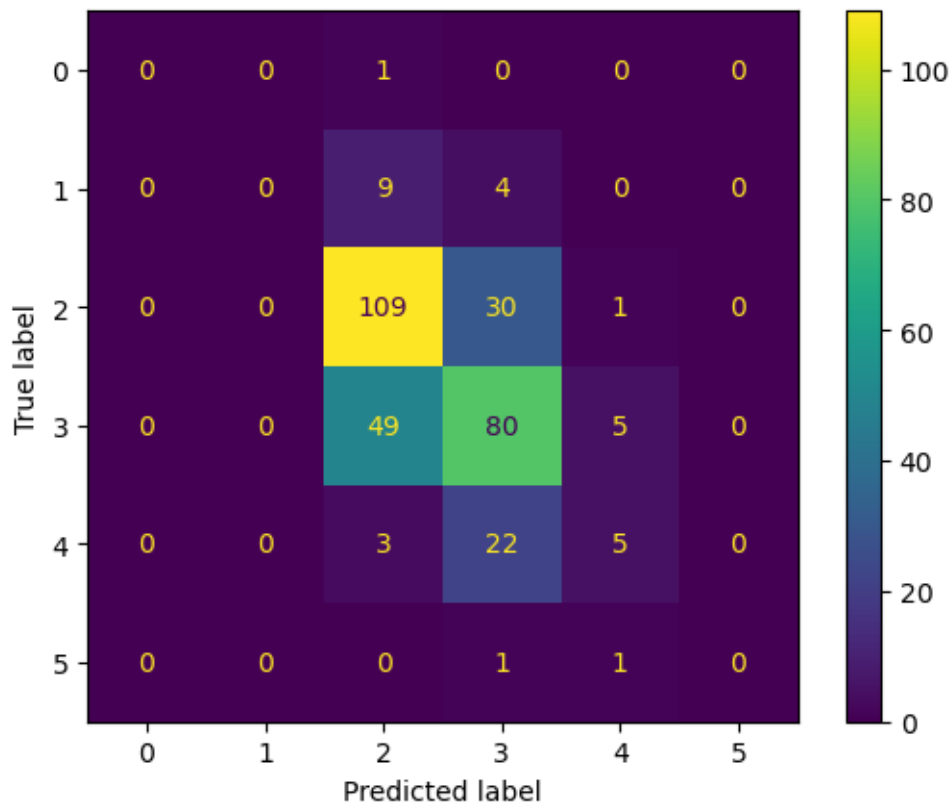
Heatmap from sns was used to see how it works. The plot shows a nice correlation diagram with different features.



While predicting the wine quality as a class,

1. Data divided into test and training data in the same way as in Regression exercise.
2. Preprocessing was done for both test and training data using `standardscaler()`.
3. Classification model was created using `svm`.
4. Wine Quality was predicted using the model.

5. The accuracy of prediction was evaluated, and the results of the prediction was predicted in matrix form.



From the above diagram, we can see those 109 wines of quality 2 was predicted correctly and 80 wines of quality 3 was predicted correctly.

### **Conclusion:**

It was a nice learning opportunity about linear regression. Importing data from excel file, extracting the required data from the dataset, using plots to for visual inspection of data, creating LR model, evaluating the model performance parameters are very useful step which are applicable for other ML applications as well.

Additionally, classification exercise helped to understand how to classify wines based on its quality.