Linear Regression using Wine Quality Dataset

Verzeo July,2021(Minor Project,Ritvik Chawla)

=>THE PROBLEM: The dataset is related to the red variant of the Portuguese "Vinho Verde" wine. For more details, consult the reference [Cortez et al., 2009]. These datasets can be viewed as regression tasks. The classes are ordered and not balanced (e.g. there are much more normal wines than excellent or poor ones). Apply Regression and find the quality of Wine.

=>TOOLS TO BE USED: The tools I have used in this projects are:

- Python programming language
- Google Collaboratory
- Pandas Software Library
- Numpy Package
- Matplotlib plotting library
- Sklearn Library
- Excel sheet data in CSV(Comma Separated Values) format

=>WHY LINEAR REGRESSION IS USED?: Linear Regression has been used in making of this project because it could be noticed that the input and output had a linear relation between them and so linear regression method has been used.

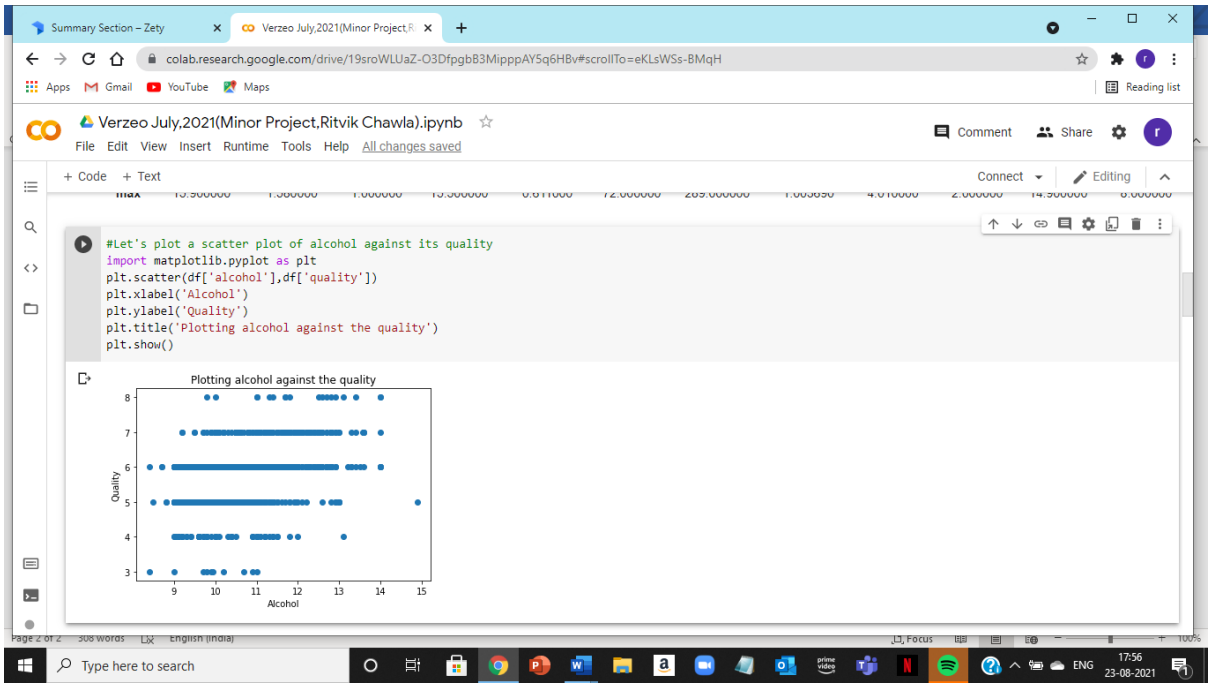
=>EXPLAINING THE ALGORITHM: In order to do the project, an algorithm was used which has been explained in the following few lines-->

1)First, NumPy was imported for linear algebra and then pandas library was imported for data processing of files(CSV in this case).

2)A data frame in pandas called df was created to read the excel sheet data in CSV format.

3) 5 rows of the data frame were printed using df.head(5) and then the computing summary statistics for the whole columns using df.describe().

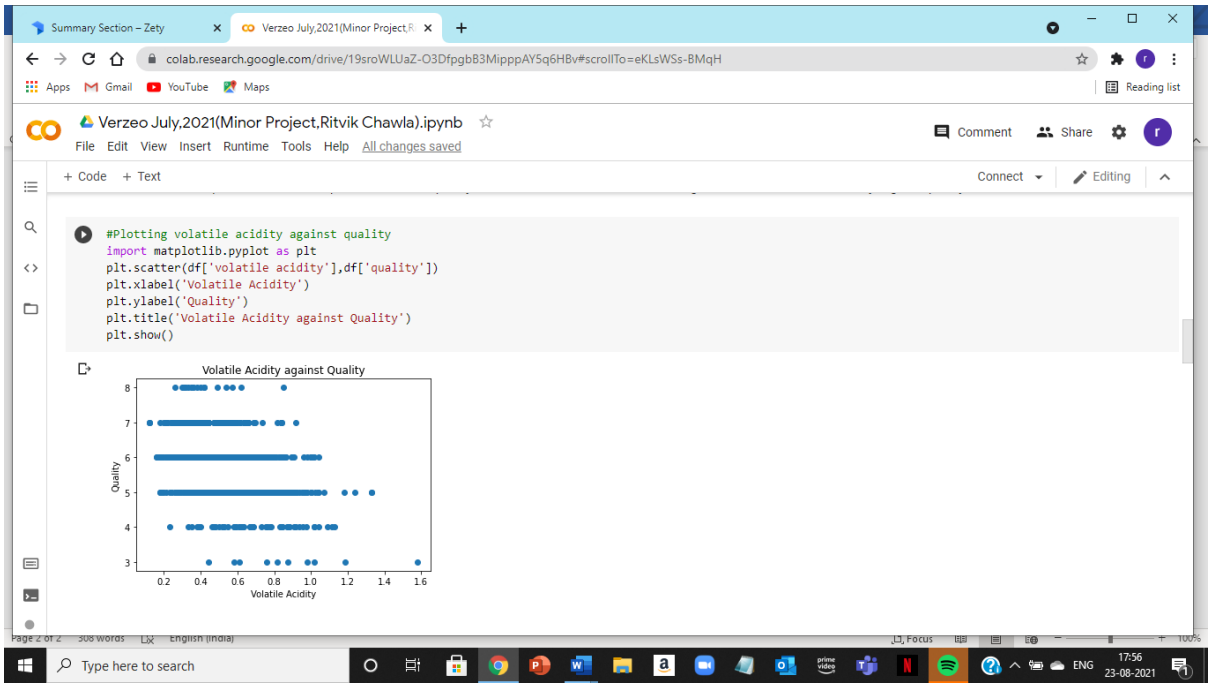
4)A scatter plot was plotted by help of matplotlib library which was imported. This scatter plot was to show relation between the alcohol against its quality whose values were used from the data frame given. From that scatter plot, there is a weak positive relationship between wine quality and alcohol content. Wines with high alcohol content are mostly high in quality.



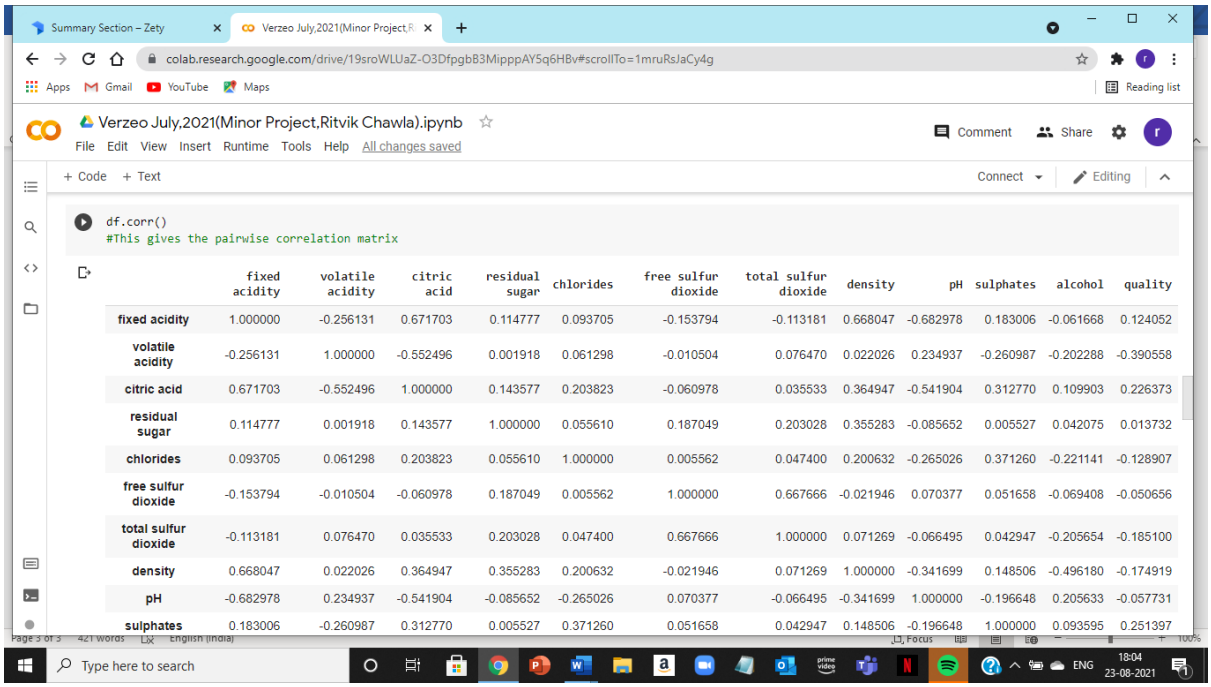
A similar scatter plot was also plotted for volatile acidity against quality from the data frame. There is a negative relationship between volatile acidity and quality.

Our quality is the response variable, and we need to find the features due to which the response variable gets affected. For e.g., we see a positive relationship between quality and alcohol but a negative

relationship between volatile acidity and quality So to determine that we have to print the coefficient matrix.

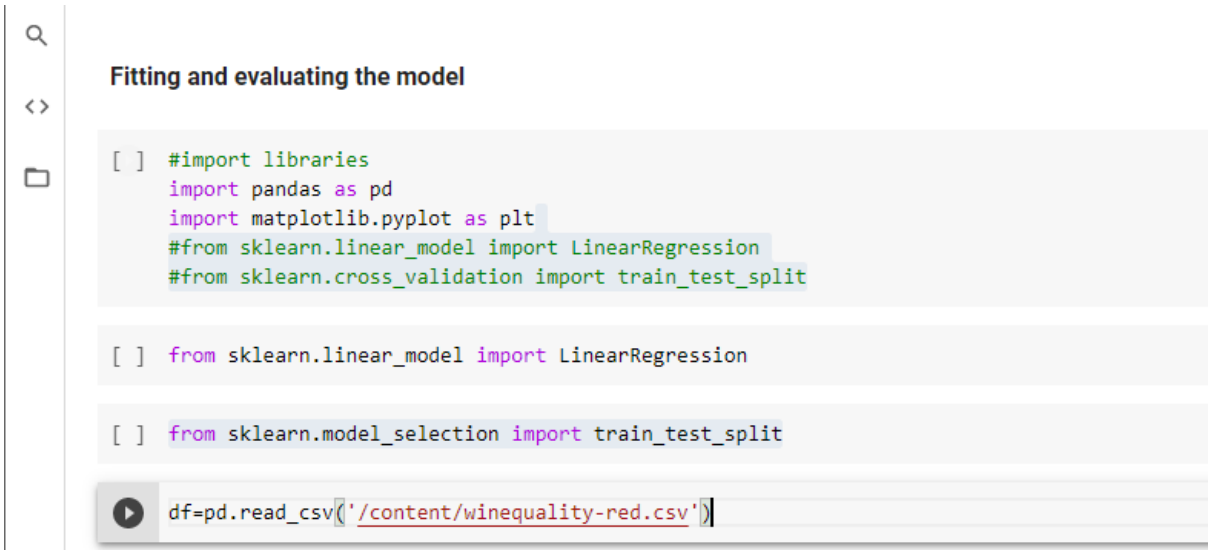


5)Next, we find the corelation of all the columns of the data frame with each other using df.corr().It helps in giving a pairwise correlation matrix. From the correlation table, it is seen that quality has positive relation with many features, but the quality has the strongest correlation with alcohol among all.

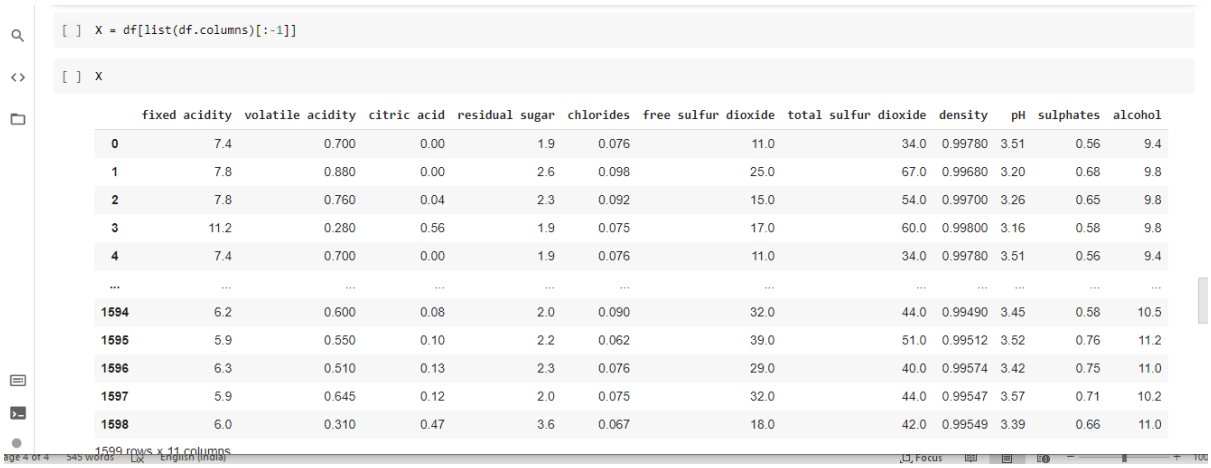


	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
fixed acidity	1.000000	-0.256131	0.671703	0.114777	0.093705	-0.153794	-0.113181	0.668047	-0.682978	0.183006	-0.061668	0.124052
volatile acidity	-0.256131	1.000000	-0.552496	0.001918	0.061298	-0.010504	0.076470	0.022026	0.234937	-0.260987	-0.202288	-0.390558
citric acid	0.671703	-0.552496	1.000000	0.143577	0.203823	-0.060978	0.035533	0.364947	-0.541904	0.312770	0.109903	0.226373
residual sugar	0.114777	0.001918	0.143577	1.000000	0.055610	0.187049	0.203028	0.355283	-0.085652	0.005527	0.042075	0.013732
chlorides	0.093705	0.061298	0.203823	0.055610	1.000000	0.005562	0.047400	0.200632	-0.265026	0.371260	-0.221141	-0.128907
free sulfur dioxide	-0.153794	-0.010504	-0.060978	0.187049	0.005562	1.000000	0.667666	-0.021946	0.070377	0.051658	-0.069408	-0.050656
total sulfur dioxide	-0.113181	0.076470	0.035533	0.203028	0.047400	0.667666	1.000000	0.071269	-0.066495	0.042947	-0.205654	-0.185100
density	0.668047	0.022026	0.364947	0.355283	0.200632	-0.021946	0.071269	1.000000	-0.341699	0.148506	-0.496180	-0.174919
pH	-0.682978	0.234937	-0.541904	-0.085652	-0.265026	0.070377	-0.066495	-0.341699	1.000000	-0.196648	0.205633	-0.057731
sulphates	0.183006	-0.260987	0.312770	0.005527	0.371260	0.051658	0.042947	0.148506	-0.196648	1.000000	0.093595	0.251397

6)Now, we must fit and evaluate the model which is to be done by importing libraries like sklearn. Sklearn library is used to import functions related to regression like LinearRegression, train_test_split,etc. Train_test_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default, Sklearn train_test_split will make random partitions for the two subsets.



Now, the data is divided into two variables X and y, in which X has all the data of the features of the columns except quality column in an ordered manner and y has the elements of the quality column of the data frame.



	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4
...	...	...	...	...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0

Both, X and y are divided into testing and training set by assigning them variables as X_train,X_test,y_train and y_test. Then a linear regression model is created to fit the data in it and a prediction variable is created using the test cases of X.

```
[19] #divide the data into training and testing set
     X_train, X_test,y_train,y_test=train_test_split(X,y, test_size=0.2, random_state=0)

[20] X_test.shape

     (320, 11)

[21] #Creating the linear regression model and fitting the data to it
     regressor=LinearRegression()
     regressor.fit(X_train,y_train)
     y_prediction=regressor.predict(X_test)
```

7) The R-score(regressor score) is calculated and printed with the testing cases of both X and y.

```
[22] #Printing the r score value
     print('R-score is %s'%regressor.score(X_test,y_test))

     R-score is 0.32838876395814165
```

8) Accuracy_score and confusion_matrix are imported from the metrics function of sklearn library. The y prediction and y test values are then printed.

```
[23] from sklearn.metrics import accuracy_score, confusion_matrix

[24] y_prediction
```

```
5.03391406, 5.18721293, 6.48783659, 5.33301372, 6.30458308,
6.03452293, 5.82849303, 5.58503524, 5.28512701, 5.75656128,
6.15273792, 5.14134479, 5.48228021, 6.38983359, 5.7901819 ,
5.42936928, 6.04810414, 6.70159678, 6.60160183, 5.97253803,
4.78890763, 5.49565005, 6.01881894, 5.48895041, 6.10824243,
5.3126164 , 5.28492351, 5.76043226, 6.36081498, 5.7195388 ,
5.26971928, 5.08498332, 5.31050693, 6.50895816, 5.44323413,
5.16377401, 5.76427324, 6.0703837 , 6.28933679, 5.07315141,
6.11874641, 5.35681067, 6.01343647, 6.12184268, 5.98547673,
5.35327968, 5.4367731 , 5.06017324, 5.58939889, 5.62593935,
6.25651326, 5.37875073, 5.31024191, 5.72008022, 6.30275434,
5.71093253, 5.27783285, 5.89582077, 6.0876033 , 5.50616533,
5.37465715, 6.2437351 , 5.07332478, 5.32665801, 4.97398339,
5.15975198, 5.3222433 , 5.80802642, 5.41018532, 6.20608506,
5.35327968, 6.29102872, 5.3563169 , 5.44615319, 5.91167448,
6.8638794 , 5.95652806, 5.78388971, 6.29371091, 5.59599754,
4.95365311, 5.59599754, 5.42146389, 6.01303846, 5.27638601,
5.88672069, 5.12760247, 6.2448062 , 5.02250118, 5.61019652,
5.78465194, 5.67109764, 6.40669755, 5.80411913, 5.64203971,
5.1425518 , 5.90518858, 5.40535036, 5.08051616, 6.21318148,
6.31555382, 5.31479331, 5.43479319, 6.00371415, 6.44580043,
5.42123137, 5.21176713, 6.0115925 , 5.82456823, 6.30161065,
```

0s completed at 9:40 AM

+ Code + Text

RAM Disk

```
[24]
```

```
5.04430938, 5.90506617, 5.06816998, 5.87105926, 6.25707687,
5.63526985, 6.1124061 , 6.58032257, 6.51304421, 6.12637713,
5.34985341, 4.91834558, 5.02633902, 5.10209868, 5.93751622,
4.82726361, 5.58643272, 5.25007252, 4.87286412, 6.34221756,
4.87320824, 5.26969753, 4.89400677, 5.33869588, 5.52716577,
5.53897238, 5.12721593, 5.06001131, 5.13911179, 5.59628108,
5.34085607, 6.96723853, 5.31357668, 5.6468939 , 5.24038769,
4.69060078, 5.33061878, 4.97930332, 5.60209709, 6.49226284,
5.72883634, 5.7319909 , 5.63538976, 6.086924  , 5.92577835,
5.85174439, 6.37461452, 5.18497794, 6.56584684, 6.14312626,
5.37501664, 6.91651759, 5.7319909 , 6.34086207, 5.78421572,
5.18439125, 5.54950551, 5.33360868, 6.269919  , 5.88265344,
5.68163948, 5.42086654, 6.12538899, 5.31406745, 5.19528191,
5.70178575, 5.5675469 , 5.14670499, 5.15556398, 5.24007613,
5.69892321, 5.06382643, 5.35893169, 5.69605721, 6.10381152,
5.84992365, 5.33722415, 5.05158628, 5.59207818, 5.30872494,
5.91032222, 5.92390941, 5.3494868 , 5.3563169 , 5.10159886,
6.55975683, 6.01757571, 6.25129053, 5.16392559, 6.36720517])
```

```
[25] y_test
```

```
1109    6
1032    5
1002    7
487     6
979     5
       ..
```

0s completed at 9:40 AM

```
[25] y_test

1109    6
1032    5
1002    7
487     6
979     5
       ..
794     6
813     4
1322    5
704     4
1023    6
Name: quality, Length: 320, dtype: int64
```

9)We now import cross_val_score from the model_selection function of sklearn library to find the cross-validation score using 5-fold cross validation method.CV is used to determine the folds i.e., 5.

```
[35] from sklearn.model_selection import cross_val_score

regressor=LinearRegression()
#Computing score using 5 fold cross validation method. cv is used to determine the folds ie 5
scores=cross_val_score(regressor,X,y,cv=5)
```

10) The mean of the cross-validation scores as well as the cross-validation scores are printed.

```
[28] print(scores.mean())

     0.2900416288433719

[29] print(scores)

     [0.13200871 0.31858135 0.34955348 0.369145   0.2809196 ]
```

=>CONCLUSION: From the above lines, we can conclude that while analysing the data frame there comes out to be errors in regression which can be minute and certain scores can be interpreted from the data like R-score(Regression Score), mean_value_score and accuracy_score which need to be found out by making test and train samples of the data given.