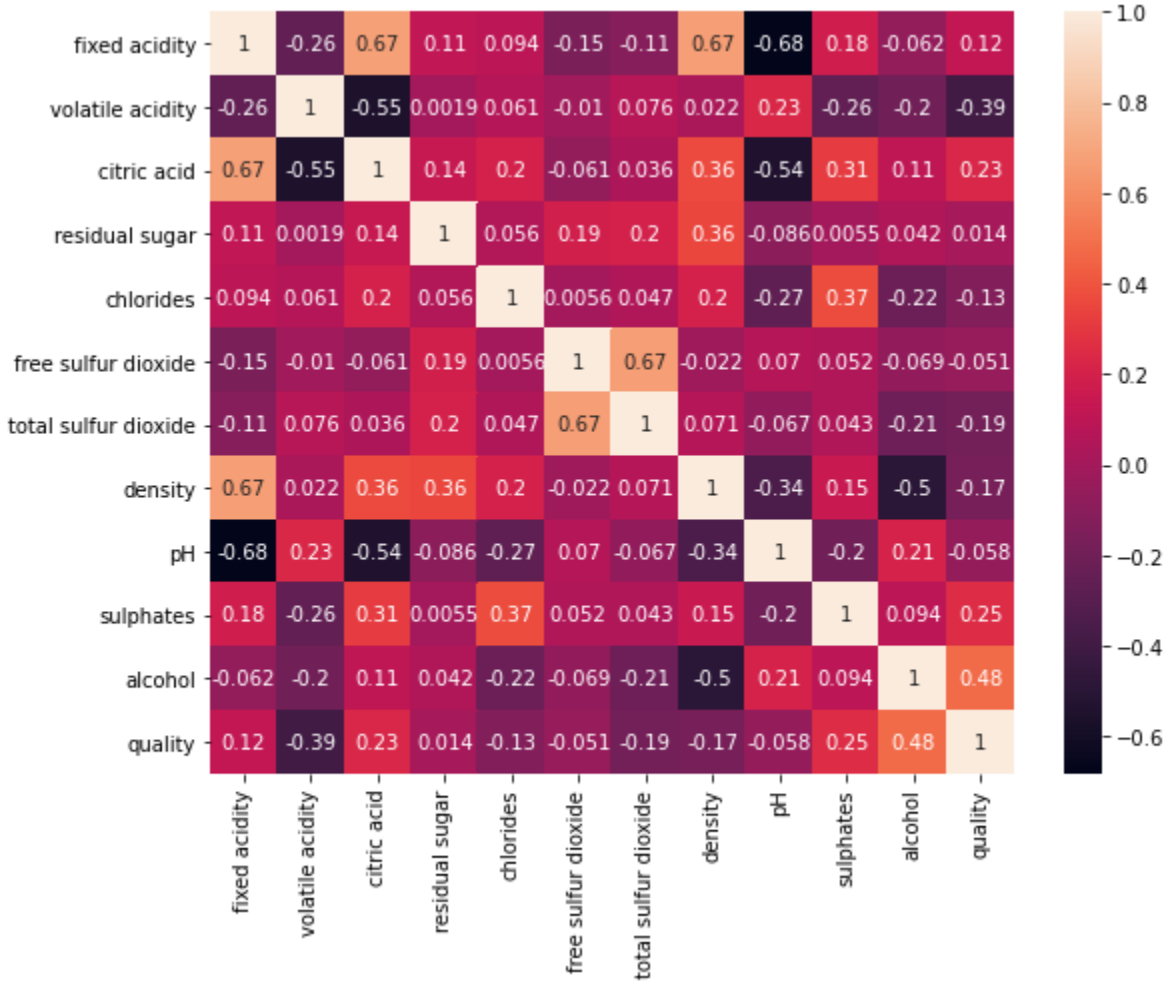
CPC DATA SCIENCE ASSIGNMENT- 4

REPORT BY 111119066

Task-1

Wine data set consists of 1599 rows and 12 columns. Results and observations after performing **Exploratory Data Analysis (EDA)** are as follows.

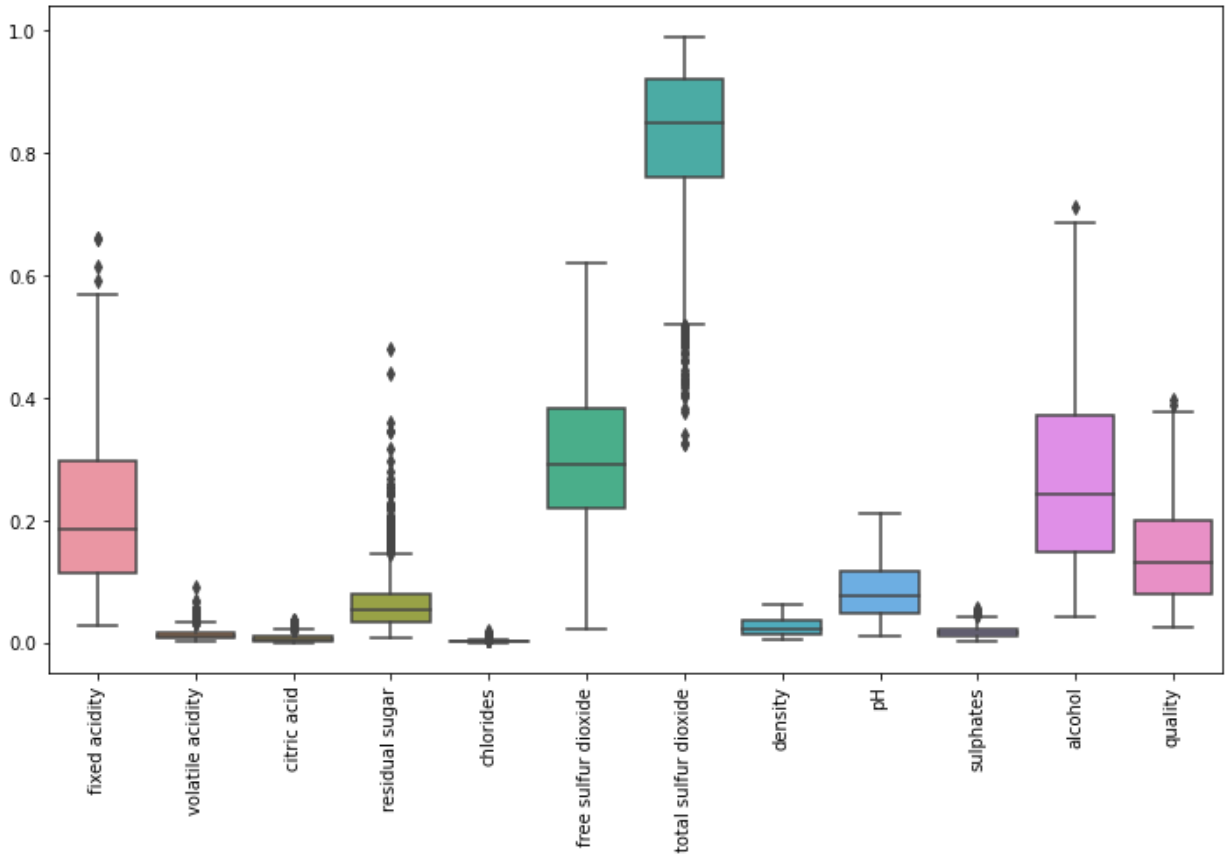
- Heatmap correlations of the wine dataset :



Talking about the heatmap, the numbers inside each box depicts the correlation between variables. Higher the number, stronger the relation between the variables.

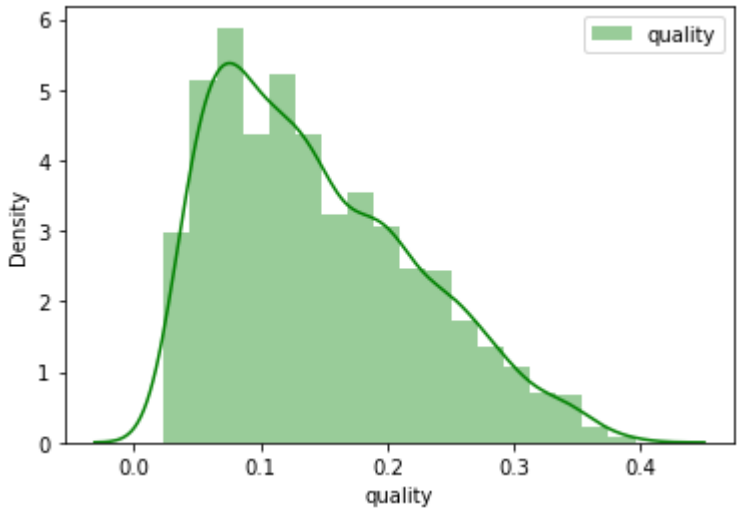
- Quality has strong positive correlation with the alcohol and negative correlation with volatile acidity.
- Density has strong positive correlation with fixed acidity.
- pH has strong negative correlation with the fixed acidity.

- Box plot:
  First we normalize (mean=0, variance=1) the data that gives a better visualization to the box plot.



From the box plot we can see median, mean, quartiles and the outliers for each attribute.
From the above plot we can see that total so_2, residual sugar and volatile acidity have many outliers, which means there's lot of variance in the data.

- Distribution Plots :

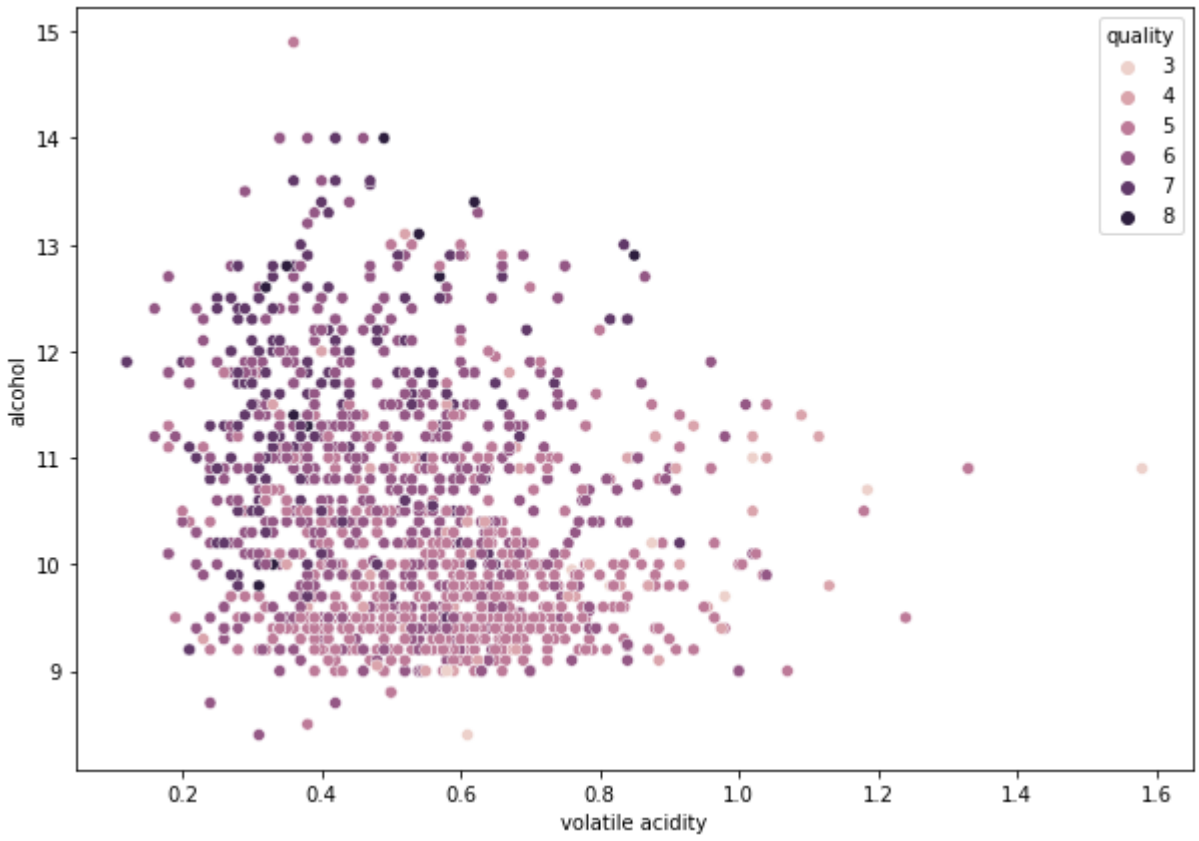


This is the dist. plot of quality, which depicts the value distributions of that attribute.

- Pair plot:
  This is a series of plots that shows every possible relation between 2 attributes of the datasets. We can observe patterns and act accordingly.

(plot is too large to include here)

- Alcohol content and acidity affecting the quality:



It's a scatter plot between acidity and alcohol, hue as quality. Darker the colour, better the quality. From the graph we can deduct that darker dots are in the higher alcohol range and lower acidity range. That means low acidity and higher alcohol composition results in better quality.

Task- 2

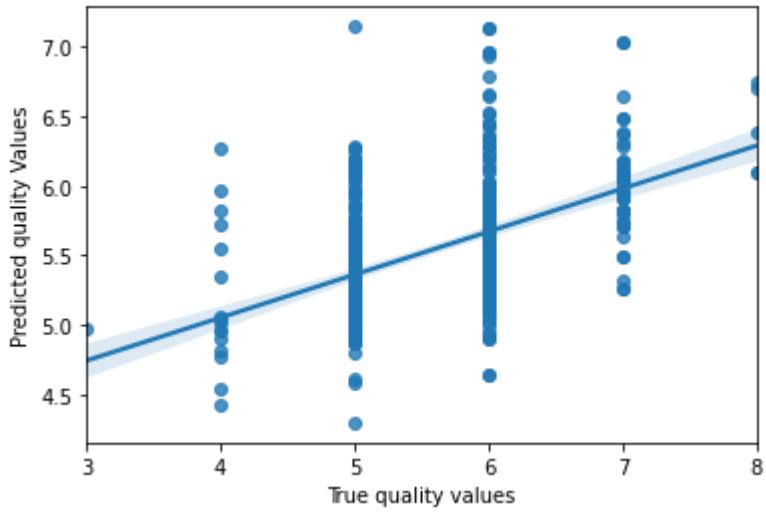
Linear Regression model to detect the quality of wine. Here we use 2 data sets, one to train the model with and other to test the model. After defining features and targets, we use linear regression.

After getting the predicted quality values and comparing to true values, we got match score of 24.75%.

```
: # Accuracy of model, intercept and respective coefficients
  print('Accuracy of model is up to:',reg.score(x_test,y_test)*100,'%')
  print('intercept:',reg.intercept_)
  pd.DataFrame({'coefficients':reg.coef_,'features':x_test.columns})

  Accuracy of model is up to: 24.572583176357522 %
  intercept: 31.70008609312341
```

Fitting line over training data:



RMSE (Root mean square error) value is 0.4463; it tells us the standard deviation of the residual errors.

Based on the above accuracy score, **it is very low and we can't use linear regression in order to get a better model**.

Task- 3

Creating binary classification from quality attribute. I've taken Boolean values because we get value error while calculating accuracy and precision scores. So

Bad – 0, Good – 1.

- Logistic Regression: we get,

```
Accuracy of the model is 0.8729166666666667
Precision of the model is 0.14814814814814814
Recall Score of the model is 0.34782608695652173
F1 Score of the model is 0.20779220779220778

Specificity is 0.899343544857768
Sensitivity is 0.34782608695652173
```

- Linear Regression as a classifier:
  Firstly, we can't use regression in classification problems because values are discrete here. For linear regression we need continuous values.
  Here's a glimpse of the predicted values array,

```
array([-1.24843063e-01, -4.48198225e-02, -2.59373494e-02,  1.72222612e-01,
       -1.24843063e-01, -1.22188729e-01, -7.65587210e-02,  1.53257966e-02,
       -1.06288933e-02,  2.25154254e-01, -8.34462049e-02,  2.25154254e-01,
       -3.50580355e-02,  3.67805573e-01, -4.69769819e-02, -1.63959082e-02,
```

Values should be either 1 or 0, but most of these were out of range and hence we can't use this.

- SVM: we get,

```
Accuracy of the model is 0.88125
Precision of the model is 0.2222222222222222
Recall Score of the model is 0.4444444444444444
F1 Score of the model is 0.2962962962962963
Specificity is 0.9072847682119205
Sensitivity is 0.4444444444444444
```

- Naive Bayesian model: we get,

```
Accuracy of the model is 0.8354166666666667
Precision of the model is 0.5
Recall Score of the model is 0.34177215189873417
F1 Score of the model is 0.40601503759398494
Specificity is 0.9326683291770573
Sensitivity is 0.34177215189873417
```

By looking at these observations we can surely eliminate linear regression, doesn't work for classification problems. Now for judging a model we use all these above mentioned evaluation metrics. Higher accuracy, precision and sensitivity define a good model. SVM and logistic regression have high accuracy, Naive and SVM have high precision whereas, SVM have higher sensitivity.

Hence based on the above conclusions we can infer that SVM is good ML model for predicting the quality of wine dataset.

Task- 4

Now applying Principal component analysis on the wine datasets before building a model. Now applying PCA with 5 components, now it reduces the 11 attributes to 5 feature variables.

- Logistic Regression: we get,

```
Accuracy of the model is 0.8833333333333333
Precision of the model is 0.018518518518518517
Recall Score of the model is 0.25
F1 Score of the model is 0.034482758620689655
Specificity is 0.8886554621848739
Sensitivity is 0.25
```

- Linear Regression as classifier:
  Again, we can't use regression in classification problems because values are discrete here. For linear regression we need continuous values.

Here's a glimpse of the predicted values array,

```
array([ 0.22877752,  0.04278823,  0.15835235,  0.08798446,  0.22877752,
        0.19819484,  0.13540751,  0.26548136,  0.28447901, -0.11360683,
        0.13309718, -0.11360683,  0.16586391,  0.36384672, -0.40230161,
       -0.4026051 , -0.13010241,  0.32628868,  0.19153403,  0.26617142,
       -0.0114241 ,  0.01325893,  0.2780436 ,  0.14827097,  0.11697066,
        0.27265554,  0.34795544,  0.2780436 ,  0.20594856,  0.31365789,
        0.06663109,  0.14272639,  0.00787564, -0.45637785,  0.17714197,
```

Values should be either 1 or 0, but most of these were out of range and hence we can't use this.

- SVM: we get,

```
Accuracy of the model is 0.8854166666666666
Precision of the model is 0.018518518518518517
Recall Score of the model is 0.3333333333333333
F1 Score of the model is 0.03508771929824561

Specificity is 0.8888888888888888
Sensitivity is 0.3333333333333333
```

- Naive Bayesian: we get,

```
Accuracy of the model is 0.8875
Precision of the model is 0.018518518518518517
Recall Score of the model is 0.5
F1 Score of the model is 0.03571428571428571

Specificity is 0.8891213389121339
Sensitivity is 0.5
```

Again, we can rule out the linear regression model here.

PCA is majorly used to make the model faster and efficient. We can clearly see from the previous task that accuracy has increased. Here accuracy, precision and f1 score is almost same for all 3 models. Sensitivity is higher for naives.

Since accuracy and precision are almost similar for 3 models and only sensitivity varies, based on sensitivity Naïve Bayesian model might perform better here.