# Team 1 - Project Summary Report

Topic: Analyses of White Wine Data

Author: Suhas Buravalla, Steven Chen, Siva Gogineni, Varun Shah

DS 6103 - Final Project
George Washington University

Table of Contents

Overview

Exploratory Data Analysis

Machine Learning Algorithms

- Logistic Regression
- K-Nearest Neighbors
- Support Vector Machines
- Random Forest

Predictions

Conclusion

# Overview:

Our team decided to use a large dataset on white wine (originally collected by Cortez et al.) to understand what impacts the quality of a wine. Is it just personal taste, or are there measurable features that can make a difference in the quality of a wine from low grade to a high grade? What can various physicochemical features tell us about the quality, and does it give an accurate insight on its quality?

Our data consists of 4898 white wine samples with 12 different features, including the subjective quality of the wine on a scale from 3 to 9. The characteristics include the levels of *fixed acidity and volatile acidity, the content of citric acid, residual sugar, chlorides, free sulphur dioxide, total sulphur dioxide and sulphates, the wine's density, pH level, and its alcohol content*.

Our Exploratory Data Analyses gave us some clarity on the distribution of the quality of the wine samples. There was  a significant imbalance in the distribution of data across the quality levels: there were a lot more average wine samples than very good or very bad wine samples. To simplify and bring a balance to our data, we reduced the number of categories from 7 to 3.

Cortez et al. originally used this dataset to train several models, including SVM, multiple regression, and a neural network to try to predict subjective quality. This group treated the quality as a continuous variable, and achieved a 65% accuracy using SVMs when allowing for the predicted quality to be within +/-0.5 of the true quality, and a ~87% accuracy when allowing for the predicted quality to be within +/-1.0 of the true quality. They concluded that the SVM model had the best performance at this task.

We posed the below questions looking at the data, to better understand what makes the most impact to the quality of a wine.

Questions:

- Are the density, fixed acidity, residual sugar, and other features correlated to each other?
- Which variables among fixed acidity, residual sugar, and total sulfur dioxide are most correlated with alcohol content?
- Which characteristics impact the quality of the wine the most?
- Are any characteristics less significant than the other?

- Which Machine Learning algorithm gives us the best results to predict the quality of the wine?

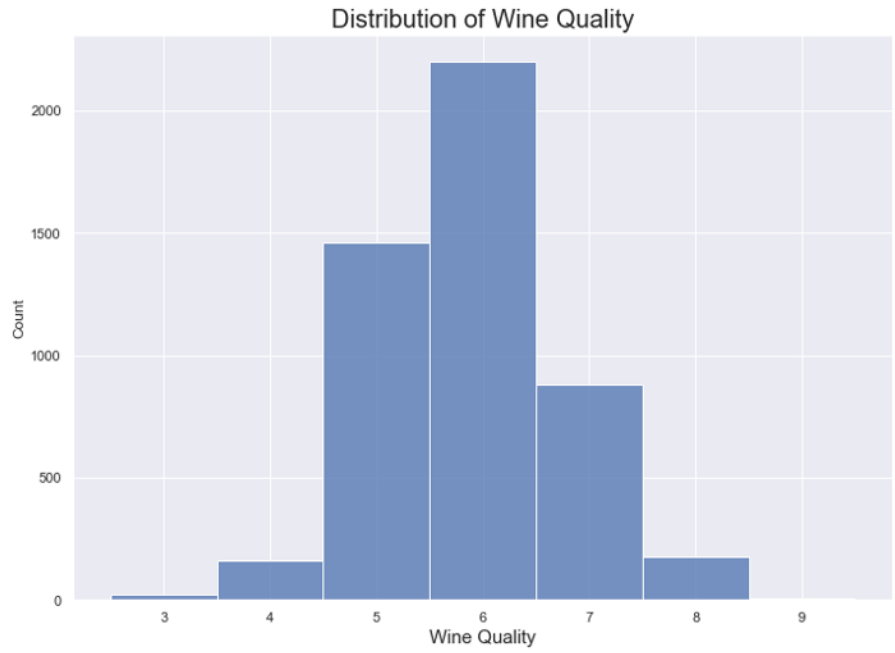
Our analysis tests four different models for predicting wine quality based on physicochemical information (Multinomial Logistic Regression, K-Nearest Neighbours, Support Vector Machines and Random Forests).

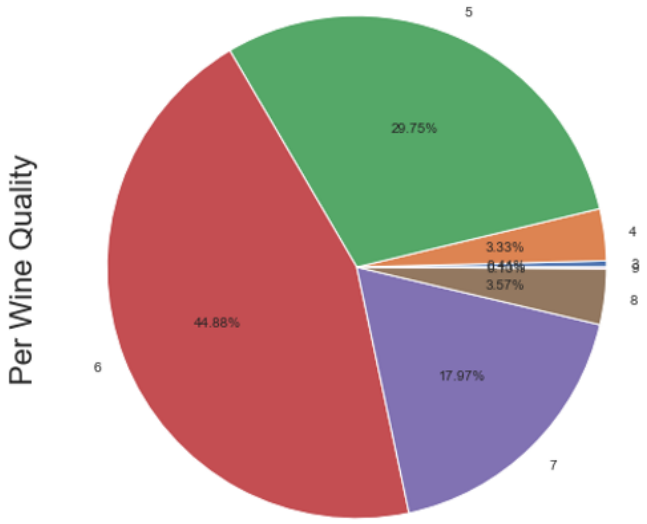
# Exploratory Data Analysis:

The below shows the overview of our data, that includes 11 different features + the subjective quality of the wine.

| | 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 | 45.0 | 170.0 | 1.0010 | 3.00 | 0.45 | 8.8 | 6 |
| 1 | 6.3 | 0.30 | 0.34 | 1.6 | 0.049 | 14.0 | 132.0 | 0.9940 | 3.30 | 0.49 | 9.5 | 6 |
| 2 | 8.1 | 0.28 | 0.40 | 6.9 | 0.050 | 30.0 | 97.0 | 0.9951 | 3.26 | 0.44 | 10.1 | 6 |
| 3 | 7.2 | 0.23 | 0.32 | 8.5 | 0.058 | 47.0 | 186.0 | 0.9956 | 3.19 | 0.40 | 9.9 | 6 |
| 4 | 7.2 | 0.23 | 0.32 | 8.5 | 0.058 | 47.0 | 186.0 | 0.9956 | 3.19 | 0.40 | 9.9 | 6 |
| 5 | 8.1 | 0.28 | 0.40 | 6.9 | 0.050 | 30.0 | 97.0 | 0.9951 | 3.26 | 0.44 | 10.1 | 6 |
| 6 | 6.2 | 0.32 | 0.16 | 7.0 | 0.045 | 30.0 | 136.0 | 0.9949 | 3.18 | 0.47 | 9.6 | 6 |
| 7 | 7.0 | 0.27 | 0.36 | 20.7 | 0.045 | 45.0 | 170.0 | 1.0010 | 3.00 | 0.45 | 8.8 | 6 |
| 8 | 6.3 | 0.30 | 0.34 | 1.6 | 0.049 | 14.0 | 132.0 | 0.9940 | 3.30 | 0.49 | 9.5 | 6 |
| 9 | 8.1 | 0.22 | 0.43 | 1.5 | 0.044 | 28.0 | 129.0 | 0.9938 | 3.22 | 0.45 | 11.0 | 6 |

We first look at the distribution of Quality across the data, to see if there is any imbalance. Our analyses show us that the Quality class is roughly normally distributed, with the below distribution:

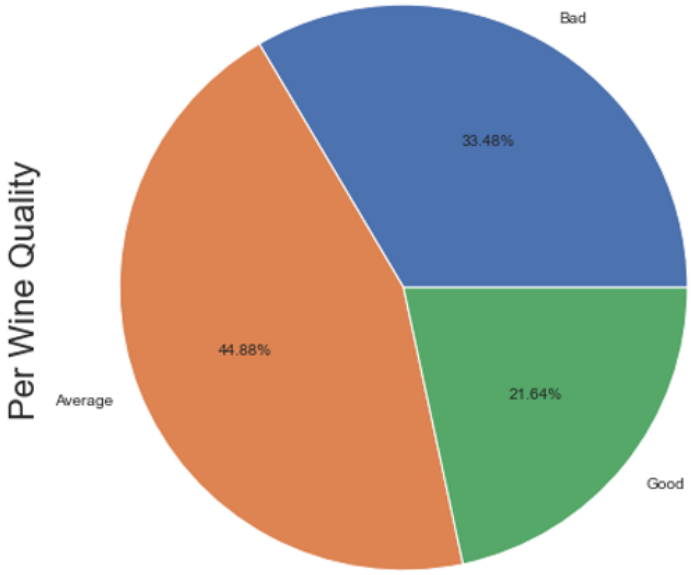



| Quality | Count |
|---|---|
| 3 | 20 |
| 4 | 163 |
| 5 | 1457 |
| 6 | 2198 |
| 7 | 880 |
| 8 | 175 |
| 9 | 5 |

As we see here, the majority of the wines fall in the Quality class 6 (about 48%), 29% in Quality class 5 followed by others, where the least being the Quality 9, being less than 1%. To bring some balance to our data, and also to improve the performance of our models, we group the different qualities of wine and reduce it to only 3 categories, Bad, Average and Good.

After grouping the data, we get the below result:




| Quality | Grades | Count |
|---|---|---|
| 3 | Bad | 1640 |
| 4 | Bad | |
| 5 | Bad | |
| 6 | Average | 2198 |
| 7 | Good | 1060 |
| 8 | Good | |
| 9 | Good | |

To further understand the relationship among the different features, we plot a correlation matrix and get a better understanding of the characteristics:

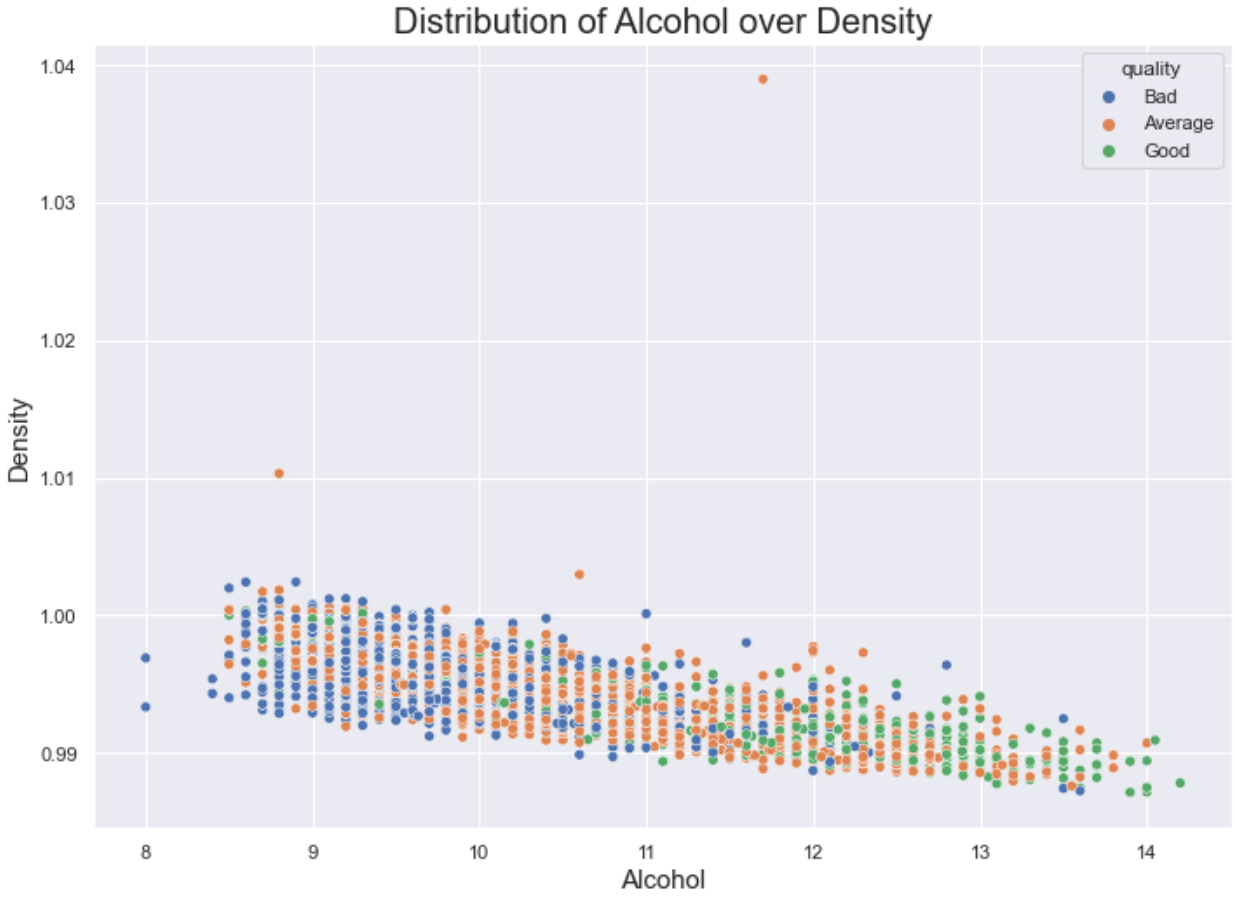
## Correlation Matrix across different features:

| | fixed acidity | volatile acidity | citric acid | residual sugar | chlorides | free sulfur dioxide | total sulfur dioxide | density | pH | sulphates | alcohol | quality |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| fixed acidity | 1.00 | -0.02 | 0.29 | 0.09 | 0.02 | -0.05 | 0.09 | 0.27 | -0.43 | -0.02 | -0.12 | -0.11 |
| volatile acidity | -0.02 | 1.00 | -0.15 | 0.06 | 0.07 | -0.10 | 0.09 | 0.03 | -0.03 | -0.04 | 0.07 | -0.19 |
| citric acid | 0.29 | -0.15 | 1.00 | 0.09 | 0.11 | 0.09 | 0.12 | 0.15 | -0.16 | 0.06 | -0.08 | -0.01 |
| residual sugar | 0.09 | 0.06 | 0.09 | 1.00 | 0.09 | 0.30 | 0.40 | 0.84 | -0.19 | -0.03 | -0.45 | -0.10 |
| chlorides | 0.02 | 0.07 | 0.11 | 0.09 | 1.00 | 0.10 | 0.20 | 0.26 | -0.09 | 0.02 | -0.36 | -0.21 |
| free sulfur dioxide | -0.05 | -0.10 | 0.09 | 0.30 | 0.10 | 1.00 | 0.62 | 0.29 | -0.00 | 0.06 | -0.25 | 0.01 |
| total sulfur dioxide | 0.09 | 0.09 | 0.12 | 0.40 | 0.20 | 0.62 | 1.00 | 0.53 | 0.00 | 0.13 | -0.45 | -0.17 |
| density | 0.27 | 0.03 | 0.15 | 0.84 | 0.26 | 0.29 | 0.53 | 1.00 | -0.09 | 0.07 | -0.78 | -0.31 |
| pH | -0.43 | -0.03 | -0.16 | -0.19 | -0.09 | -0.00 | 0.00 | -0.09 | 1.00 | 0.16 | 0.12 | 0.10 |
| sulphates | -0.02 | -0.04 | 0.06 | -0.03 | 0.02 | 0.06 | 0.13 | 0.07 | 0.16 | 1.00 | -0.02 | 0.05 |
| alcohol | -0.12 | 0.07 | -0.08 | -0.45 | -0.36 | -0.25 | -0.45 | -0.78 | 0.12 | -0.02 | 1.00 | 0.44 |
| quality | -0.11 | -0.19 | -0.01 | -0.10 | -0.21 | 0.01 | -0.17 | -0.31 | 0.10 | 0.05 | 0.44 | 1.00 |

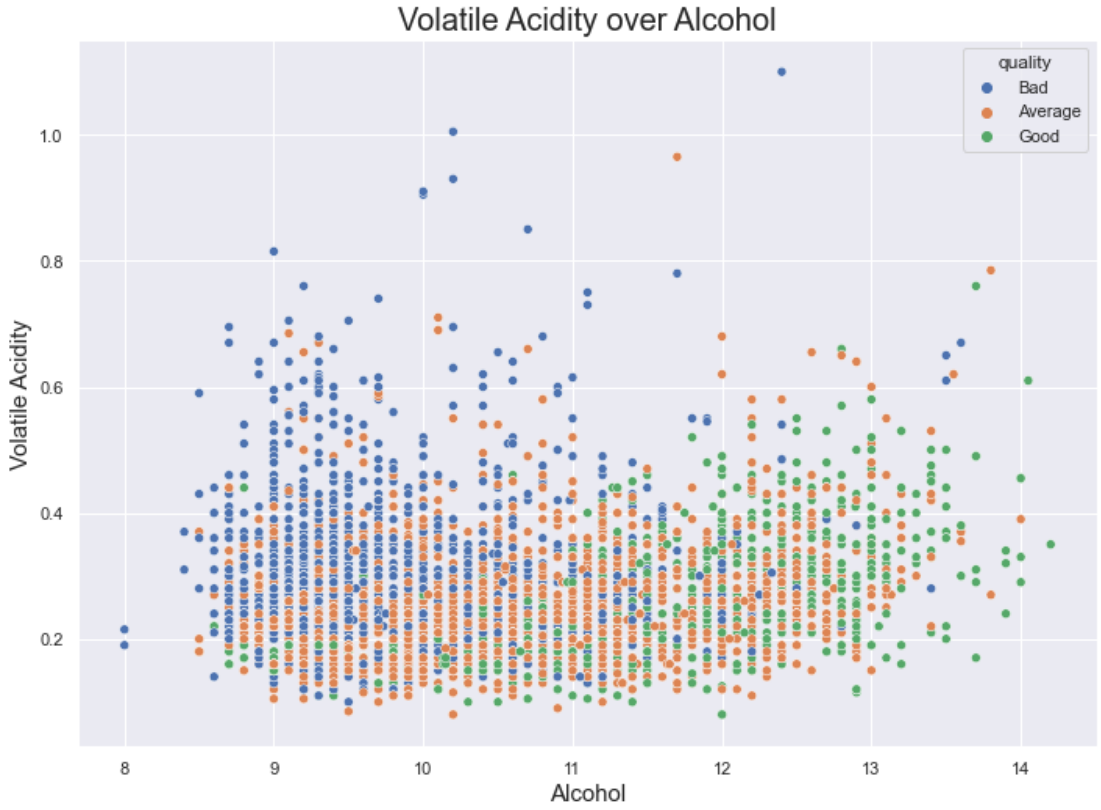
Here, we see Alcohol having the strongest correlation with Quality, followed by density and volatile acidity.

We also see Alcohol and Density are highly correlated, where the density reduces by a factor of 0.78 with the increase of alcohol content. We also see the Total Sulphur Dioxide reducing by 45% with the increase in alcohol. Chlorides, Residual Sugar are also strongly correlated with alcohol.

The below plot shows the distribution of different grades of wine over its alcohol content and Density. As we see below, the Good wines (in Green) fall under higher alcohol content and low density compared to the average and the low grade ones.

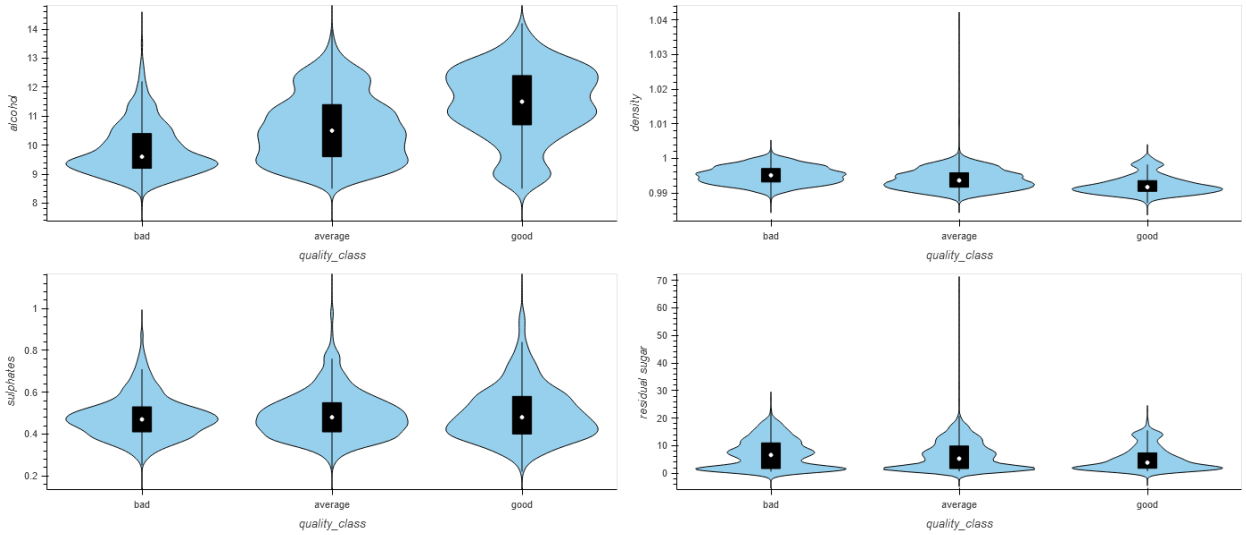


The below shows the Volatile Acidity over Alcohol content, showing us the distribution of different quality wines. The good ones have a reduced volatile acidity and higher alcohol content compared to the others.




Violin plots showing the distribution of some chemical concentrations across bad, average, and good wines is shown below. In good wine, the distribution of alcohol is visibly skewed towards higher concentrations, and the opposite is true for bad wine. Good wine generally has low density and residual sugar concentrations, but theres a

small number of good wines with high density and residual sugar.




# Machine Learning Algorithms

*Model Preparation:*

The data is first split into training and test set with 4:1 ratio.

We set the Predictors X as: *Fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulphur dioxide, total sulphur dioxide, sulphates, density, pH level and alcohol.*

We set the target y as: *Quality*

**Our Best Models:**

## Multinomial Logistic Regression:

The first model we considered is the Multinomial logistic regression model.

Multinomial Logistic regression is an extension of logistic regression that supports classifications where multiple classes are involved.

Logistic regression is the process of modeling the probability of a discrete outcome, given one or more input variables. Despite its name, it is a classification model and not a regression model. This is a classification model that is very easy to understand and it achieves fairly good performance with linearly separable classes.

Logistic regression is generally limited to two class classification problems. If we wish to use it in multi-class classification problems, we can perhaps try one vs one or one vs rest  logistic regression where a separate model is trained for each class predicted.

$$Logistic\ function = \frac{1}{1+e^{-x}}$$

One-vs-One Logistic regression is a heuristic method for using the binary classification algorithms for multi-class problems. One-vs-One splits the entire dataset into one dataset for each class vs every other class.

(NumClasses * (NumClasses – 1)) / 2

One-vs-Many Logistic Regression involves splitting the multi-class dataset into multiple binary classification problems. The binary classifier is trained on each binary classification problem and predictions are made from the model that is most accurate. The number of classes of the dependent variable determines the number of one-vs-rest binary classification models required.

These strategies are not as favorable because they are adapted strategies to deal with multi-class classification problems and the dataset is learned approximately. Instead of creating multiple binary classifiers and choosing the one with highest confidence, we can use the Multinomial Logistic regression function in Sklearn or Statsmodels which learns all the classes directly, parameters are learned interdependently and it is better against outliers.

*Scores and Model evaluation:*

We initially run a Multinomial Logistic Regression model with all 11 variables for a frame of reference.

We need to build a more efficient model by using fewer features while maintaining or increasing the accuracy level.

*Train set evaluation:*

```
Classification Report
              precision    recall  f1-score   support

     average       0.53      0.67      0.59      1630
         bad       0.63      0.59      0.61      1238
        good       0.59      0.36      0.44       805

    accuracy                           0.57      3673
   macro avg       0.59      0.54      0.55      3673
weighted avg       0.58      0.57      0.57      3673
```

*Test set evaluation:*

```
Classification Report
              precision    recall  f1-score   support

     average       0.56      0.69      0.62       568
         bad       0.64      0.59      0.61       402
        good       0.62      0.38      0.47       255

    accuracy                           0.59      1225
   macro avg       0.61      0.55      0.57      1225
weighted avg       0.60      0.59      0.59      1225
```

Using just Volatile acidity, density, chlorides, alcohol and total sulfur dioxide as the features or variables, we can maintain similar accuracy, precision and recall scores.

*Train set evaluation:*

```
Classification Report
              precision    recall  f1-score   support

     average       0.52      0.66      0.58      1630
         bad       0.64      0.60      0.62      1238
        good       0.55      0.31      0.40       805

    accuracy                           0.56      3673
   macro avg       0.57      0.52      0.53      3673
weighted avg       0.57      0.56      0.55      3673
```

*Test set evaluation:*

```
Classification Report
              precision    recall  f1-score   support

     average       0.53      0.64      0.58       568
         bad       0.63      0.59      0.61       402
        good       0.52      0.34      0.41       255

    accuracy                           0.56      1225
   macro avg       0.56      0.52      0.53      1225
weighted avg       0.56      0.56      0.55      1225
```

*Results:*

We can see that the accuracy, precision and recall scores aren't very good. This is because of the dataset where amongst the 3 qualities of wines, the average quality occurs the most number of times and due to this most of the cases are predicted as average. We can see the higher recall rate for the average and bad wines whereas the recall rate of good wines is comparatively lower. The classifier finds it difficult to predict good wine as good which might be because of having fewer data points for good wine. The model has the best precision and recall rate for the bad wines and the average wines. When a wine is predicted as average or bad, there is a fairly decent chance (50-60%) that the wine is actually average or bad respectively.
Multinomial Logistic Regression might not be the best suited model for this project as the majority of wines are predicted as average(Quality 6). The model isn't very good for predicting good wines (Quality 7, 8 and 9), the accuracy and other scores aren't the most ideal scores.

## K-Nearest Neighbour:

K-Nearest Neighbour is one of the models we used to test the quality of the wine data. KNN will use the similarity between data points and use that info to predict the predictor values. K value is the Key parameter in this KNN algorithm. In most cases, the high number of K values ,where we consider the number of neighbour data points, will help to predict the predictor more accurately. Few iterations are needed to pick the correct K value. Besides K value, we use few other algorithms such as brute,kde etc in this KNN model.

*Scores and Model Evaluation:*

Following are the scores we achieved on the training and test data set evaluations. Based on these scores,we can definitely consider the model is performing really well on the training set of data. One observation we made is that the Model is giving importance to all of the input features as it is performing better with all of the input features than dropping any of the features based on the Multicollinearity or co relationship.

*Train set evaluation:*

```
Accuracy score train Test:              0.9997277429893819
Confusion Matrix for train Set :
                                        [[ 795    0    0]
                                        [   0 1634    0]
                                        [   0    1 1243]]
Classification Report for train Set:
                              precision    recall  f1-score   support

                         1.0       1.00      1.00      1.00       795
                         2.0       1.00      1.00      1.00      1634
                         3.0       1.00      1.00      1.00      1244

                    accuracy                           1.00      3673
                   macro avg       1.00      1.00      1.00      3673
                weighted avg       1.00      1.00      1.00      3673
```

*Test set evaluation:*

```
Accuracy score test Test:                0.7020408163265306


Confusion Matrix for Test Set:

                                        [[400  92  72]
                                        [101 280  15]
                                        [ 74  11 180]]

Classification Report for Test Set:

                              precision    recall  f1-score   support

                     average       0.70      0.71      0.70       564
                         bad       0.73      0.71      0.72       396
```

| | | | | |
|---|---|---|---|---|
| **good** | **0.67** | **0.68** | **0.68** | **265** |
| **accuracy** | | | **0.70** | **1225** |
| **macro avg** | **0.70** | **0.70** | **0.70** | **1225** |
| **weighted avg** | **0.70** | **0.70** | **0.70** | **1225** |

*Results:*

Above posted results are achieved with all of the input features (fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density,pH, sulphates, alcohol, quality_class). Model reveals that quality of the wine is achieved by considering all of these input features.

## Support Vector Machine:

The next model we looked into is Support Vector Machine, this is a supervised Machine Learning algorithm that is mostly used in classification challenges. The objective of the SVM is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier.

For our data, we used the Kernel trick, Radial Basis Function (rbf) and used GridSearch to find the right parameter levels for C and gamma levels.

Through testing different combinations of C and gamma levels, the best parameters had C at 1000, and gamma , 1.

```
grid.best_params_ :
{C: 1000, gamma: 1}

grid.best_score_ :
0.56166
```

*Scores:*

```
SVM model accuracy (with the train set): 0.92
Classification Report for Training set:
              precision    recall  f1-score   support

     Average       0.92      0.92      0.92      1630
         Bad       0.93      0.92      0.92      1238
        Good       0.92      0.93      0.92       805

    accuracy                           0.92      3673
   macro avg       0.92      0.92      0.92      3673
weighted avg       0.92      0.92      0.92      3673


SVM model accuracy (with the test set): 0.602
Classification Report for Test Set:
              precision    recall  f1-score   support

     Average       0.62      0.63      0.63       568
         Bad       0.60      0.61      0.61       402
        Good       0.55      0.53      0.54       255

    accuracy                           0.60      1225
   macro avg       0.59      0.59      0.59      1225
weighted avg       0.60      0.60      0.60      1225
```

*Results:*

The SVM gives us Test Scores of 60% Accuracy, Precision, Recall and F1-scores, while the training scores were 92% for accuracy, precision (weighted average), recall (weighted average) and F1-scores (weighted average).

This was achieved by selecting the most correlated features with *Quality*, which was *Alcohol, Density, Chlorides, Volatile Acidity and Total Sulphur Dioxide.*

*Model Evaluation:*

We began by first selecting all the features as predictors, for our Quality as the target. With all 11 features included, the model gave us a Training Set accuracy, precision, recall and F1-scores of 95% and Test Score accuracy, precision and recall scores of 60.8%.

We also considered multicollinearity for our feature selection, with VIF check, we were able to reduce the features to *Volatile Acidity, Citric Acid, Residual Sugar, Chlorides and Free Sulphur Dioxide.* This again gave us the training score at 95%, however the test scores reduced to 58%.

For SVM, our final model selected were the features *Alcohol, Density, Chlorides, Volatile Acidity and Total Sulphur Dioxide,* which we achieved using the Correlation Matrix giving us the best results.

## Random Forest:

The next model we considered is the random forest model. The random forest is an ensemble of decision trees, each of which predicts a class based on a part of the training dataset, using a portion of the input features. The overall forest then predicts the class based on the average prediction probability distributions of the individual trees. In this way, a random forest model avoids some issues common to individual decision trees, including a tendency to overfit the data.

The various hyperparameters of the random forest classifier were varied to produce the best classification results. Important hyperparameters are:

**n_estimaters , 300, number of trees in forest**

**min_samples_leaf , 2, minimum number of samples required to be at a leaf node before splitting decision tree is possible**

**max_depth , None, trees go as deep as possible**

**min_samples_split , 2, minimum number of samples required to split a node**

*Scores and Model Evaluation:*

```
Random Forest model accuracy (with the test set): 0.707
Random Forest model accuracy (with the train set): 0.996
Classification Report on test set
```

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| average | 0.64 | 0.76 | 0.69 | 517 |
| bad | 0.79 | 0.72 | 0.75 | 431 |
| good | 0.77 | 0.59 | 0.67 | 277 |
| accuracy | | | 0.71 | 1225 |
| macro avg | 0.73 | 0.69 | 0.70 | 1225 |
| weighted avg | 0.72 | 0.71 | 0.71 | 1225 |

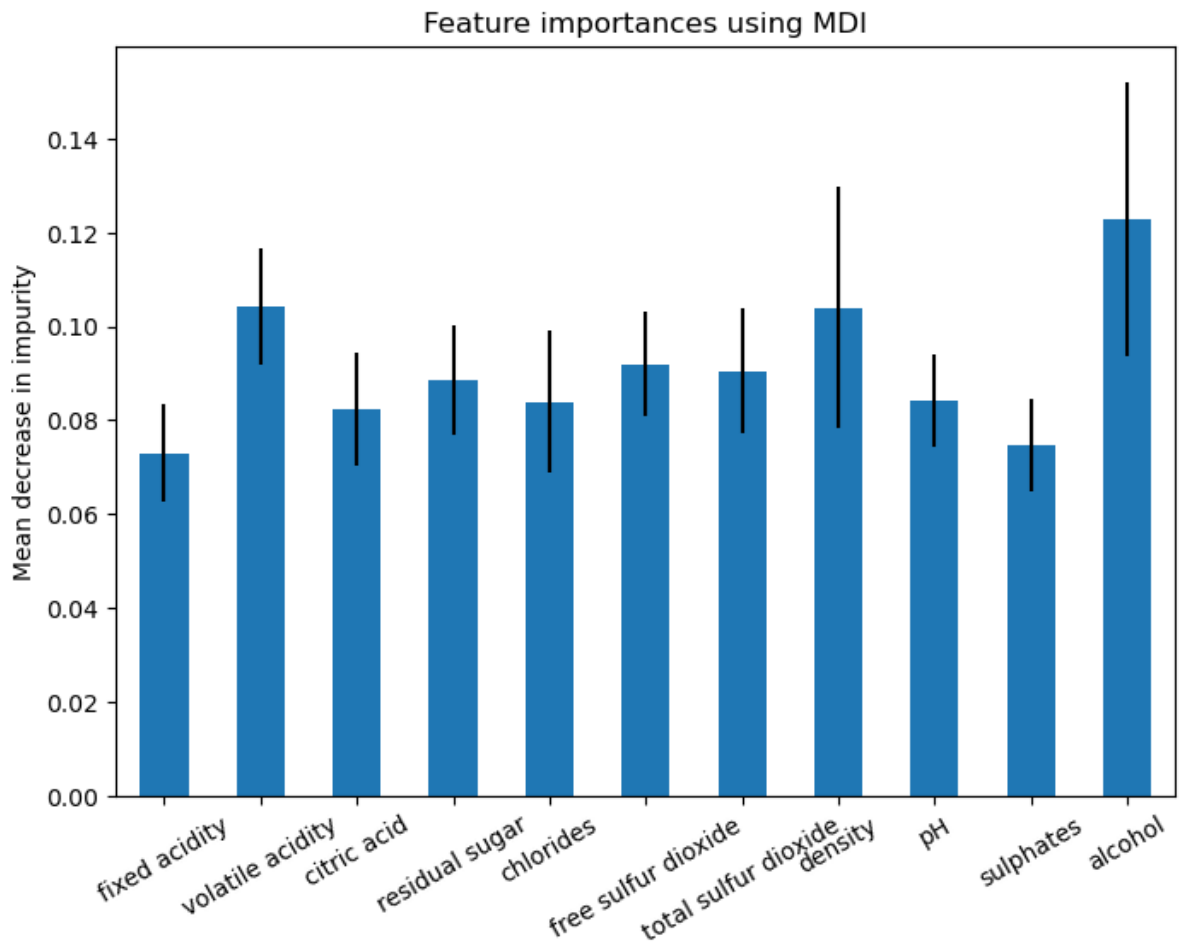
Classification Report on training set

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| average | 0.99 | 1.00 | 1.00 | 1681 |
| bad | 1.00 | 1.00 | 1.00 | 1209 |
| good | 1.00 | 0.99 | 0.99 | 783 |
| accuracy | | | 1.00 | 3673 |
| macro avg | 1.00 | 0.99 | 1.00 | 3673 |
| weighted avg | 1.00 | 1.00 | 1.00 | 3673 |

*Results:*

The random forest classifier achieves around 70% accuracy, precision, recall, and f1-scores after hyperparameter optimization in predicting the quality class of the test wine samples. This is among the best prediction scores of our models. The recall rate of good wine is significantly lower than average or bad wine, indicating that the classifier is worse at predicting good wine to be good, although if the classifier does predict a wine to be good, its highly likely that it is good (78% precision). The opposite is true for average wine, where if a wine is predicted to be average, theres a 37% chance that it is not average.

The feature importance plot is shown below.




Alcohol content, density, and volatile acidity are the 3 most important features according to the random forest model, although all features are somewhat important. These important features are not the most independent features according to the VIF check, but they agree with the features that correlate most with quality according to the correlation matrix above.

If the raw quality scores (3-9) are used instead of the reduced quality class, performance is lowered about 2% across the various metrics.

If only the top 5 important features (alcohol, density, volatile acidity, free sulfur dioxide, total sulfur dioxide) are used, performance is lowered about 1.5% across the various metrics.

If only the top 5 features correlated with quality (alcohol, density, volatile acidity, chlorides, total sulfur dioxide) are used,  performance is lowered about 1.5% across the various metrics.

The top 5 important features and the top 5 features correlated with quality share 4 features, and decreases performance slightly, indicating that either can be used instead of the full feature set if run time of the code is important.

# Predictions we can make from our Models:

Of the 4 models we used, 2 of the models(KNN,Random Forest) are doing well in predicting the wine quality.

Assuming the test set of white wine samples is representative of white wine, we expect to be able to predict if a white wine sample is, generally speaking, good, bad, or average, with 70% confidence.

**Following are some of the examples we tested through the model:**

*Example: 1*

```
fixed acidity, 7
volatile acidity,.27
citric acid,1
residual sugar,20.7
Chlorides,.045
free sulfur dioxide,70
total sulfur dioxide,200
density,1.001
pH,6
sulphates, .45
Alcohol,8.8
```

Wine Quality Prediction: ‘Bad’

Our first example has the above features, key indicators alcohol at 8.8%, Residual Sugar at 20.7 and volatile acidity at 0.27, and our model predicts the wine to be of ‘bad’ quality.

*Example 2:*

```
fixed acidity, 1
volatile acidity,.27
```

```
citric acid,1
residual sugar,2.7
Chlorides,.045
free sulfur dioxide,70
total sulfur dioxide,200
density,1.001
pH,1
sulphates, .45
Alcohol,8.8
```

Wine Quality Prediction: `'average'`

The second scenario, has all the same features as the first one, except the residual sugar in this wine has reduced to 2.7, which improves the quality of the wine to an ‘average’ one.

*Example 3:*

```
fixed acidity, 1
volatile acidity,.27
citric acid,0.1
residual sugar,2.7
Chlorides,.045
free sulfur dioxide,7
total sulfur dioxide,100
density,1.001
pH,1
sulphates, .45
Alcohol,18.8
```

Wine Quality Prediction: `'good'`

Our final scenario has very similar characteristics as the two above, except the alcohol level in this wine has been increased to 18.8% while keeping the same residual sugar as the average win at 2.7. This improves the quality of the wine further and is predicted as a ‘good’ wine.

# Conclusion

From our analyses, it was found that volatile acid, citric acid, residual sugar, chlorides and free sulphur dioxide are least correlated to each other. The features with highest correlation to alcohol are density, total sulphur dioxide and residual sugar. The subjective quality of the wine seems to be most impacted by Alcohol, Density, Chlorides, Volatile Acidity & Total Sulphur Dioxide. The features which seemed to be most significant among all the other features were sulphates and pH. KNN and Random Forest turned out to be the most accurate models for predicting the subjective quality of white wines.

# Citations

Brownlee, J. (2016). *Machine Learning Mastery with Python: Understand Your Data, Create Accurate Models and Work Projects End-to-end*.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, *47*(4), 547–553. https://doi.org/10.1016/j.dss.2009.05.016

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *The Journal of Machine Learning Research*, *12*(null), 2825–2830.

Wilimitis, D. (2019, February 21). *The Kernel Trick*. Medium. https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f