|  |  |
| --- | --- |
| https://imagesvc.meredithcorp.io/v3/mm/image?url=https%3A%2F%2Fcdn-image.foodandwine.com%2Fsites%2Fdefault%2Ffiles%2F.%2Ffwx-best-red-wines-for-white-wine-drinkers.jpg&w=400&c=sc&poi=face&q=85  data scientist’s guide to quality wine | term project report   SCS 3251\_017 Team 6   Ali Mortazavi,  Neda Iranmanesh,  Nisarg Patel,  Olga Senko,  Vaishali Bhat, Vinay Pandit |

**Contents**

[SCS 3251\_017 Team 6 0](file:///C:\Education\Continuing%20Education\UFT%20-%203251%20-%20Statistics\Assignments\3251%20-%20Group%206%20-%20Project\Submission\3251_017_Team6_Wine_ProjectReport.docx#_Toc26446189)

[Introduction 2](#_Toc26446190)

[Business Objective 3](#_Toc26446191)

[1. Hypothesis Testing 4](#_Toc26446192)

[1.1. Hypothesis 1 4](#_Toc26446193)

[1.2. Hypothesis 2 6](#_Toc26446194)

[1.3. Hypothesis 3 7](#_Toc26446195)

[1.4. Conclusion - Hypothesis Testing 9](#_Toc26446196)

[2. Linear Regression 10](#_Toc26446197)

[2.1. Abstract 10](#_Toc26446198)

[2.2. Introduction to Classification and Regression Approaches for Predictive Modeling 10](#_Toc26446199)

[2.3. Predictive Modeling using Linear Regression 11](#_Toc26446200)

[2.4. Predictive Modeling using Ridge Regression with Regularization by scikit learn 17](#_Toc26446201)

[2.5. Conclusion – Linear Regression 17](#_Toc26446202)

[3. Logistic Regression 18](#_Toc26446203)

[3.1. Logistic Regression Classification 18](#_Toc26446204)

[3.2. Checking Logistic Regression Conditions 18](#_Toc26446205)

[3.3. Prepare Test/Train Data Set and Fit Model 19](#_Toc26446206)

[3.4. Tune Threshold value based on Training Set before evaluating model using Test data 21](#_Toc26446207)

[3.5. Model Performance with Tune Threshold using Test data 22](#_Toc26446208)

[3.6. Conclusion - Logistic Regression 23](#_Toc26446209)

[4. Conclusions 24](#_Toc26446210)

[4.1. Summary 24](#_Toc26446211)

[4.2. Results using Predictive Modeling methods 24](#_Toc26446212)

[5. Appendix 25](#_Toc26446213)

[5.1. Source of Data 25](#_Toc26446214)

[5.2. Data collection: 25](#_Toc26446215)

[5.3. Data preparation 25](#_Toc26446216)

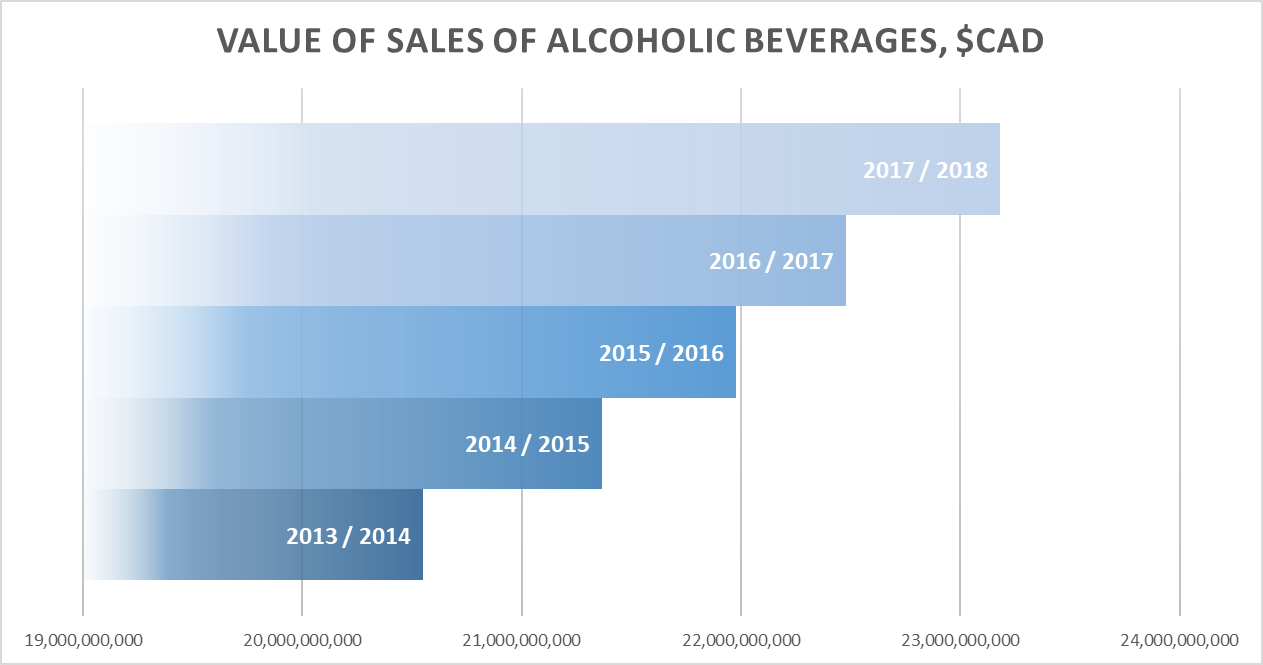
[5.4. Description of attributes: 26](#_Toc26446217)

[5.5. Data Analysis 26](#_Toc26446218)

[5.6. SVR 32](#_Toc26446219)

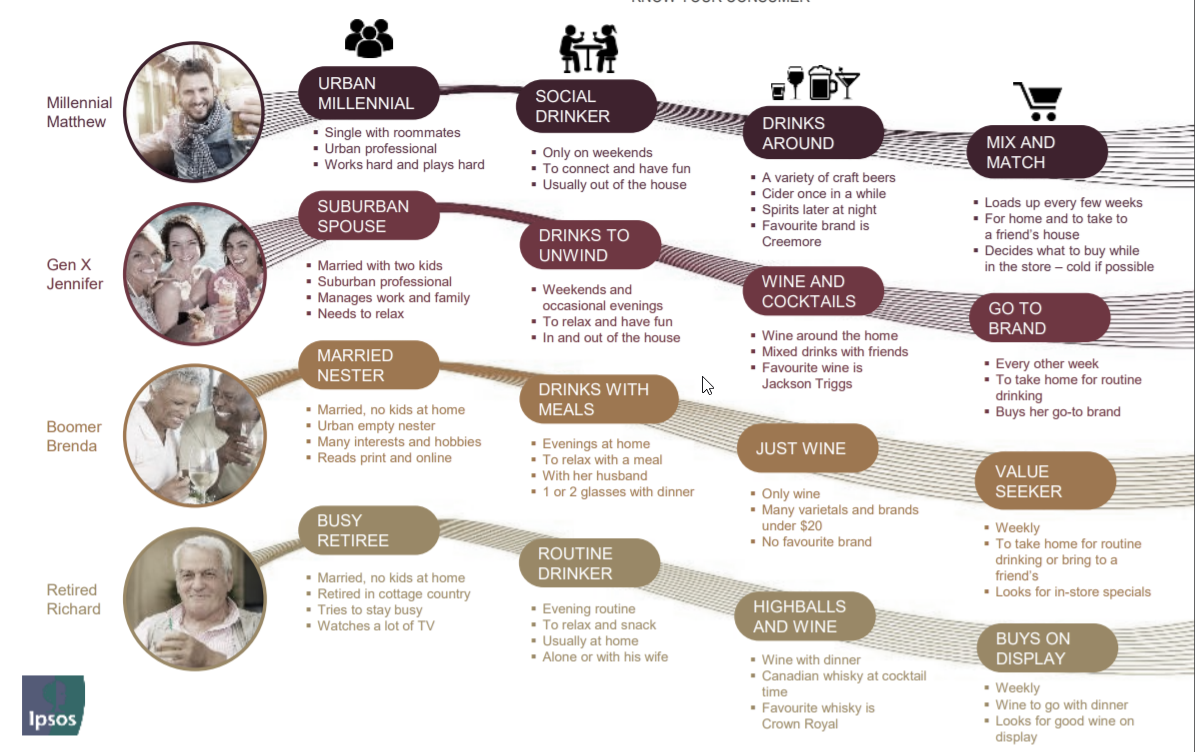
# Introduction

According to a 2016 report from World Health Organization, Canadians drink 3.6 liters more alcohol per capita than the worldwide average. Knowing that, we shouldn’t be surprised to learn that Canadians spent more than $23B on alcohol in 2018. On average, the industry’s sales have been growing by 3% year-over-year for the last 5 years.

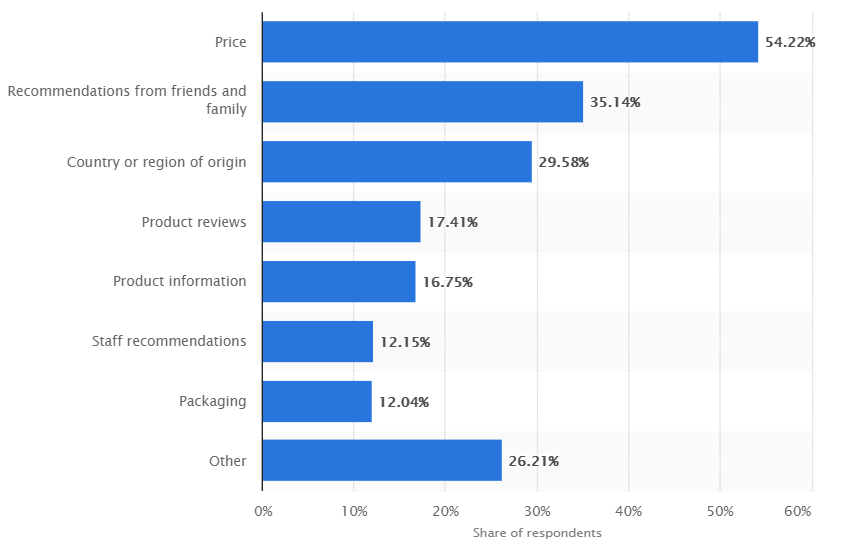


We start wondering what drives that growth. Based on the recent report by StatsCanada, Canada is still primarily a beer market, but the latter’s consumption has been declining over the last decade. The second biggest sector, wines, is showing constant growth, with a 27% increase in consumption between 2008/2009 and 2017/2018.

Given the huge demand for the beverage alcohol industry overall and for wine specifically, we decided to deep dive into consumption, to uncover who is the main alcohol consumer, how they consume and what drives their purchase decision. Following, is a snapshot of available data answering the questions at hand.



Canadian purchase criteria when buying wine 2019



Reviewing the bar chart above, we were surprised that wine quality wasn’t listed as one of the most important purchasing criteria.

# Business Objective

Based on the information we gathered through the initial stages of our analysis, we identified opportunities for growth in the market. Whoever switches consumers’ mindset from price centric to quality centric can become the new Starbucks of the wine industry and teach Canada to drink wine. To win it all, we should identify and simplify quality assessment for consumers and make it part of our marketing campaign, potentially considering going after competitors who do not meet the same quality standards.

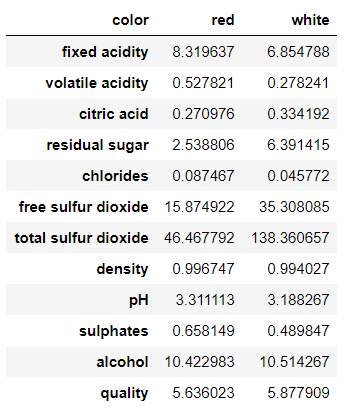
# Hypothesis Testing

To further investigate wine quality, we analyzed the UCI's wine dataset which comprises data related to Red and White Vinho Verde wines, from the north of Portugal. The goal was to model wine quality based on physicochemical tests. While many hypotheses can be tested using the dataset, we tested the 2 following types of hypotheses that we found most relevant based on the analysis we performed last semester.

* Drawing inferences by comparing features between Red and White wines
* Drawing inferences by treating a subset of our data as a sample and the rest of the data as the population

Many other hypotheses were tested, but not reported given the expected length of the assignment.

Our ultimate goals were to uncover relevant insights regarding our dataset as well as to use multiple hypothesis testing methods. Below are the means for each attribute for Red and White wines. For the first and second hypotheses, we tested whether there is significant difference between the means of both wines for some of these attributes.



### Hypothesis 1

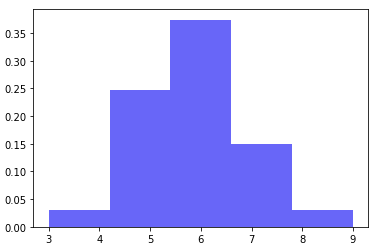
We were interested to understand whether there is a difference in quality between Red and White wines or not. Given that we want to find out if the two populations have the same mean quality score, we are using the **two-sample hypothesis test**.

* H0: 𝜇r = 𝜇w
* H𝑎: 𝜇r ≠ 𝜇w

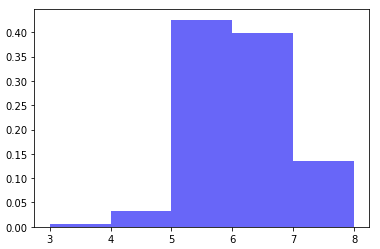
Where µr is the mean score for the quality of Red wines and µw is the mean score for the quality of White wines. We are looking at whether there is a difference between the averages for quality scores of Red and White wines or not.

* + 1. Verifying normality assumptions

Normal distribution histogram showing White wine quality data



Normal distribution histogram showing Red wine quality data



The quality scores range between 3 and 9 (on a 10-point scale) and the bar chart for White wines is shaped in a bell curve, which suggests that there are no obvious departures from the normal model. The quality scores range between 3 and 8 (on a 10-point scale) and the bar chart for Red wines is skewed to the left, which isn't an issue given that the sample size is of 1599. Because the normality conditions are reasonably satisfied, we can apply the t-distribution to this setting.

* + 1. Testing the hypothesis

Using the T-Test statistics, we calculated the p-value for this hypothesis.

|  |
| --- |
| Output:  Ttest\_indResult(statistic = -10.149363059143164, Pvalue = 8.168348870049682e-24 |

We reject the null hypothesis since p < 0.05 and hence can conclude that the mean quality score of White wines is significantly higher (5.88) than of Red wines (5.64).

### Hypothesis 2

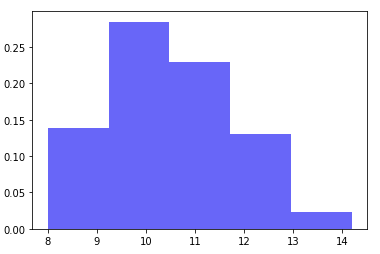
After finding out that there was in fact a significant difference between quality levels based on the color of the wine, we investigated whether there was a difference between ABV% between Red and White wines – because we suspected that alcohol levels may affect preferences. Given that we want to find out if the two populations have the same mean ABV%, we are using the **two-sample hypothesis test**.

* H0: 𝜇r = 𝜇w
* H𝑎: 𝜇r ≠ 𝜇w

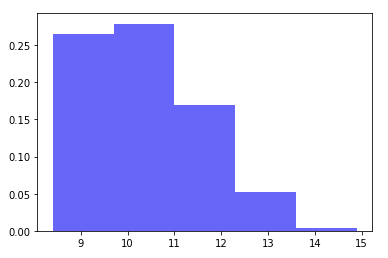
Where µr is the mean ABV% for Red wines and µw is the mean ABV% for White wines. We are looking at whether there is a difference between the averages for alcohol content of Red and White wines or not.

* + 1. Verifying normality assumptions

Normal distribution histogram showing White wine alcohol levels data



Normal distribution histogram showing Red wine alcohol levels data



Both bar charts are skewed to the right, which isn't an issue given that the sample sizes are of 4898 for White wines and of 1599 for Red wines.

Because the normality conditions are reasonably satisfied, we can apply the t-distribution to this setting.

* + 1. Testing the hypothesis

Using the T-Test statistics, we calculated the p-value for this hypothesis.

|  |
| --- |
| Output:  Ttest\_indResult(statistic = -2.8590287839639124, Pvalue = 0.004277779864993429) |

We reject the null hypothesis since p < 0.05 and hence can conclude that White wines have a higher mean alcohol level of 10.51% compared to Red wines which have a mean alcohol level of 10.42%.

### Hypothesis 3

So far, we have uncovered that White wines are preferred to Red wines and that White wines have higher alcohol content. These 2 findings prompted us to investigate whether there is a significant difference in quality levels between wines with high and low alcohol content or not.

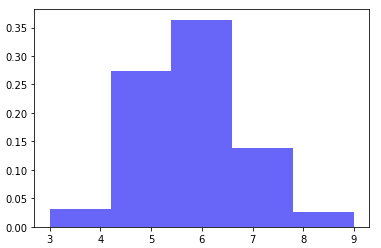
We are using a **one-sided z-test** to evaluate whether wine quality is better when wines have high alcohol content.

* H0: quallowABV = qualhighABV
* H𝑎: quallowABV < qualhighABV

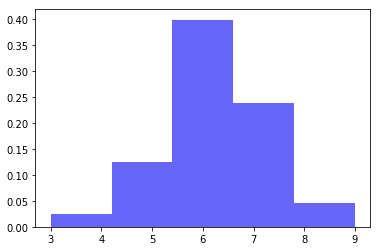
We are interested in finding out if quality is significantly different for wines with higher alcohol content compared to the population of wines.

1.3.1. Verifying normality assumptions

Normal distribution histogram showing quality data for all wines

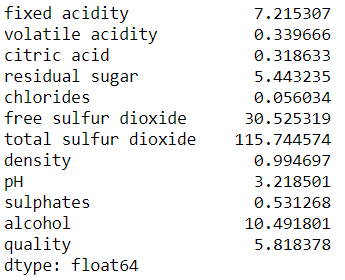


Normal distribution histogram showing quality data for wines with high alcohol content (> 10.49%)



The above histograms look normally distributed. There is a slight skew to the right for the original dataset and a slight skew to the left in the high ABV% dataset, which isn't an issue given that the sample sizes are of 6497 and 3006 respectively.

1.3.2. Testing the hypothesis



The mean of ABV% is 10.49, we will categorize wines with higher alcohol content as any wine that has ABV% higher than 10.49%. Our original sample will become the population and we will create a sample using wines with high ABV%.

The graphs already suggest that wines with higher ABV% have higher quality scores, and we confirmed this by calculating the z-score which is 22.93.

We reject the null hypothesis that wines with above average ABV% have the same quality as all other wines - meaning that the quality of wines with higher alcohol content is significantly higher.

### Conclusion - Hypothesis Testing

We tested the following types of hypotheses

* + We drew inferences by comparing features between Red and White wines
  + We drew inferences by treating a subset of our data as a sample and the rest of the data as the population

The mean quality value was higher for White wine as compared to Red wine. This came to us as a surprise based on popular knowledge but given that White wines are typically sweeter and are made to highlight their floral aromas and acidity, the slight preference for White wine makes sense. Portugal has the advantage of numerous indigenous grape varieties that can keep their acidity in hot climates and including these in a blend provides balancing freshness for rich White wines. The region is known for its White wines (86%) and is the main source of a unique style of White wine.

We did not initially expect White wines to have significantly higher ABV% than Red wines, but the data uncovered a significant difference between the ABV% of Red and White wines. This highlights the importance of hypothesis testing and how data-driven conclusions are much more robust than our own expectations. Given the climate in the North-West corner of Portugal, the grape varieties that grow in that region make wines with high ABV%.

As we expected, alcohol level impacts positively wine likability. According to experts, the alcohol content of wine has spiked considerably in recent years. “There’s pressure on winemakers from critics for intense flavors” - this increase in demand for wines with higher alcohol content could explain why, in our sample, the quality scores were significantly higher for wines with above average alcohol content. Thus, we now know that wine quality increases with alcohol content for both Red and White wine.

In conclusion, if you are ever interested to try a Vinho Verde wine and are not a wine connoisseur, your best bet is to get a White wine with high alcohol content. If you prefer Red wines, your best chances of getting a good quality wine is to purchase the one with the highest alcohol content.

# Linear Regression

### Abstract

We analyzed the UCI's wine dataset and demonstrated how various Predictive models can be built to predict the quality of wine as a continuous variable. Understanding which attributes play a key role in making a “quality” wine is essential in driving revenue growth. This opens many opportunities for wine producers to increase wine sales revenue in the global market.

Though wine quality is ordinal categorical in the dataset, we chose to investigate using a model for continuous response knowing that if we use the average ratings from multiple experts for the same wine, we are likely to get real numbers while each individual rating is an integer. For example, expert 1 ranks a specific wine as 6 and another expert ranks it as 7 thus the average for the wine is 6.5. However, one short coming that remains is that the rankings are ordinal so 6 is higher than 3 but unlike continuous numbers a score of ordinal 6 is not twice as high as an ordinal ranking of 3.

A predictive model that used all the independent variables as predictors was built first. A detailed study of the model summary was done to analyze its complexity. To improve our understanding of the data and to boost  we performed the following 3 analyses.

1. Modify the Loss function of Linear Regression: We chose the "Ridge Regression with Regularization" technique to penalize the coefficients that were extreme.
2. Data Transformation: We did Data Normalization using Standard Scaler and Log transformation of skewed variables.
3. Use a higher order Regression such as Polynomial or Support Vector: We opted for Support Vector Regressor on Standardized data whose parameters were selected using GridSearch.

We finally concluded that models that used Regularization with Ridge Regression and SVR may perform better on unseen data as these are simpler than other linear models. We have also shown that a carefully selected GridSearch using Support Vector Regression on Standardized data improved the from 0.32 to 0.44, which is better than publicly available Kaggle kernels.

### Introduction to Classification and Regression Approaches for Predictive Modeling

There are two types of Machine Learning methods: Classification and Regression.

##### Classification:

Predictive modeling using classification is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y). The output variables are often called labels. The mapping function predicts the class or category for a given observation. For example, an email of text can be classified as belonging to one of two classes: "spam" and "not spam".

##### Regression:

Predictive modeling using Regression is the task of approximating a mapping function (f) from input variables (X) to a continuous output variable (y). A continuous output variable is a real-value, such as an integer or floating-point value. These are often quantities, such as amounts and sizes. For example, a house may be predicted to sell for a specific dollar value, perhaps in the range of $100,000 to $200,000.

In this study, we will mainly investigate the following types of regression approaches:

###### Linear Regression:

Linear regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, linear regression is estimating the parameters of a linear model. A linear model has a continuous dependent variable.

Given a dataset of n observations {𝑦i, 𝑥i}where *i=1, 2, …* ***n***, each observation 𝑖 includes a scalar response 𝑦i and a column vector 𝑥𝑖 of values of 𝑝 predictors 𝑥𝑖,𝑗 for 𝑗=1, ..., 𝑝. In a linear regression model the response variable 𝑦𝑖 is a linear function of the predictors:

𝑦𝑖=𝛽1𝑥𝑖1 + 𝛽2𝑥𝑖2 + ⋯ + 𝛽𝑝𝑥𝑖𝑝 + 𝜀

So, when we extrapolate the above linear equation to the entire dataset, this yields us an **overdetermined system**  *, i=1, 2, …* ***n*** of n equations with 𝑝 unknown coefficients 𝛽1, 𝛽2, …, 𝛽𝑝 with 𝑛>𝑝.

Such a system has no exact solution, so the goal is instead to find the coefficients 𝛽 which "best" fit the equations, in the sense of solving the quadratic minimization problem

= where denotes a **loss function** that searches for the estimates of β.

The most used loss function is **ordinary least squares (OLS)** which chooses the coefficients 𝛽 that minimizes the sum of the squared residuals. Mathematically, the loss function for OLS is

###### Support Vector Regression:

Support Vector Regression is a statistical modelling technique that is very similar to Linear regression. The loss function of this technique is the same as for Linear Regression but, ignores the errors which are situated within the certain distance of the true value. This type of function is often called – epsilon intensive – loss function. The cost of the errors is zero for all points that are inside the band. More details are covered in [the appendix](#_SVR).

### Predictive Modeling using Linear Regression

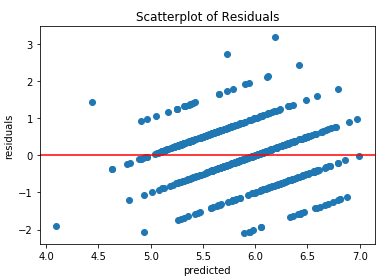
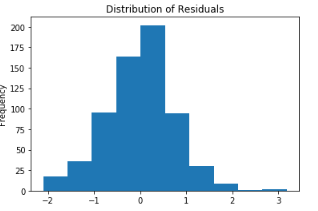
When fitting a least squares line, we generally require the following:

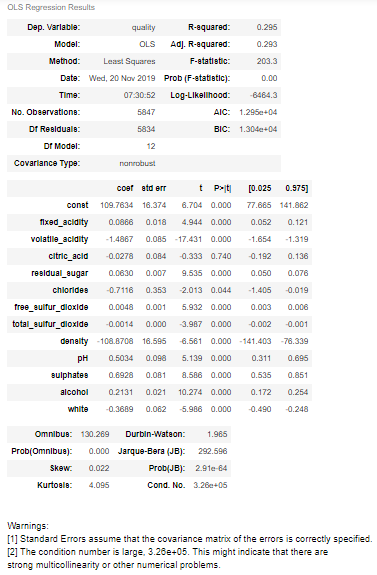
1. Linearity. The data should show a linear trend. If there is a nonlinear trend, an advanced regression method should be applied.
2. Nearly normal residuals. Generally, the residuals must be nearly normal. When this condition is found to be unreasonable, it is usually because of outliers or concerns about influential points.
3. Constant variability. The variability of points around the least squares line remains roughly constant.
4. Independent observations.

In this section, our task is to check if the wine dataset meets the above 4 requirements.

Wine data will be split into two groups - train and test. 10% of data will be randomly selected as test data, while the remaining 90% will be selected as training data. The training data will then be fitted under the Linear Regression model.

Now, the model is trained using Linear Regression, let us plot the distribution of Residuals and the scatter plot of predicted quality vs. actual quality.





Next, we are printing the R2 of the model on Test. Model Score on Test Set: 0.31

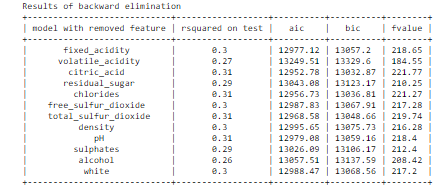
Our observations about the dataset are as follows:

1. There is a weak Linear trend amongst response and independent variables.
2. Our dataset is small and has only 6497 samples. Hence, we can safely assume that observations are independent.
3. The distribution of the residuals is normal.
4. The scatterplot of the fitted values vs residuals does not show constant variability because residuals are not "bouncing randomly" around the 0 line. This confirms our assumption that linear relationship is not reasonable.

However, the model has an R2 of 0.3, which is poor. This is further corroborated by very high AIC and BIC scores and very low Log-Likelihood. But the good news is that the Prob (F-statistic) tells us that the model is significant and the p-values of all the coefficients except citric\_acid are statistically significant at the 95% confidence Level. Our next goal is to try to improve R2 by building different models.

##### Fine Tuning by backward elimination

The first technique we are employing is called 'Backward Elimination'. Backward elimination starts with the model that includes all potential predictor variables. Variables are eliminated one-at-a-time from the model until we cannot improve the adjusted R2.



The above code builds several models each by removing one independent variable and training using the rest of the variables and lists the model's R2 for the Test set. From the table above, we see that none of the models have an R2that is considerably better than the first model. Hence, the backward elimination has NOT helped us much.

Our next technique is to use a different method to improve the model.

##### Fine Tuning by backward elimination using p-value

This is our second way to build a better model than the previous two. Here we use the p-value of the basic model (also called as "largest model" because this model is trained using all independent variables).

The p-value may be used as an alternative to adjusted R2 for model selection. In backward elimination, we would identify the predictor corresponding to the largest p-value. If the p-value is above the significance level, usually α = 0.05, then we would drop that variable, refit the model, and repeat the process. If the largest p-value is less than α = 0.05, then we would not eliminate any predictors and the current model would be our best-fitting model.  
  
In our case, we will keep the OLS summary of the largest model as reference, we see that most of the variables have a p-value of 0. But, **citric\_acid and chlorides have p-values of**   0.625 and 0.152 respectively, which are both larger than 0.05. Hence, we will be training a new model without these 2 variables.

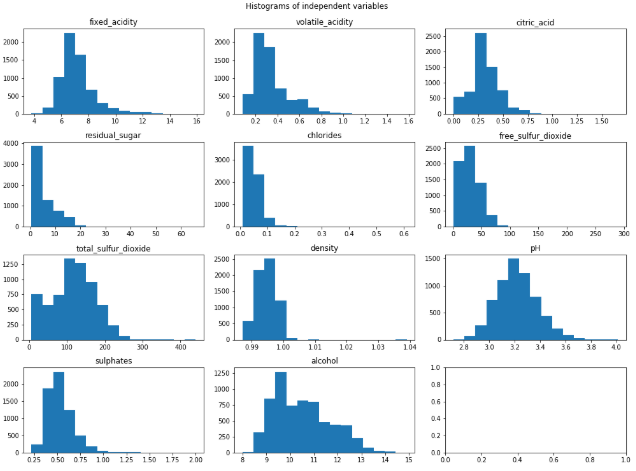


The above code returned R2 on the Test set of the model that was trained without citric\_acid and chlorides. Notice that the R2 of the new model is NOT much better than any of the models obtained through the backward elimination technique. This proves that backward elimination using p-value has NOT helped us much.

##### Fine Tuning by transforming independent variable

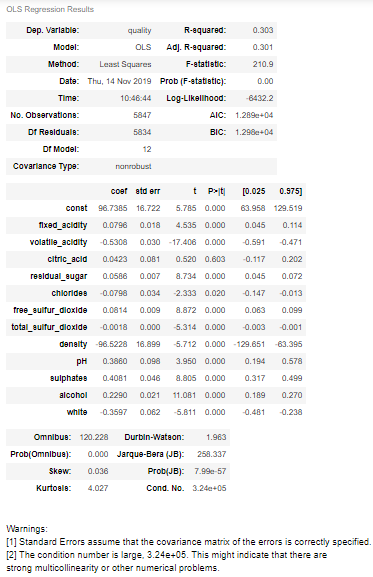
Symmetric distribution is preferred over skewed distribution as it is easier to interpret and generate inferences. Some modeling techniques require normal distribution of variables. So, whenever we have a skewed distribution, we can use transformations which reduce skewness. For right skewed distributions, we take the square / cube root or logarithm of variables and for left skewed, we take the square / cube or exponential of variables. These transformations may improve the prediction and Log transformation is one of the commonly used transformation techniques used in these situations.

To visualize the distributions of variables, we will be plotting their histograms first. Then, do the necessary transformations and train a model using these transformed variables.



The above figure contains histograms of all independent variables. The variables that need to be transformed are chlorides, free\_sulfur\_dioxide, volatile\_acidity and sulphates. The free\_sulfur\_dioxide variable was transformed using np.sqrt and the other three variables were log-transformed. Let us display their histograms to confirm.

Next, we are training a model using these transformed variables.



Model Score on Test Set: 0.33

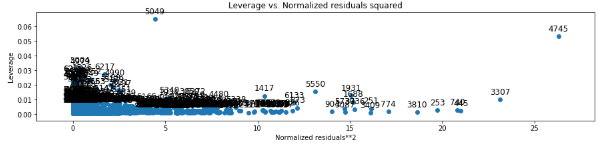
##### Model Selection using OLS Summary

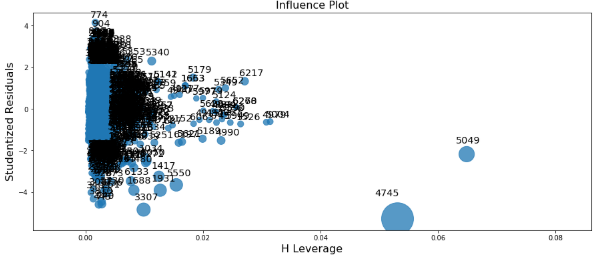
In this section, our goal is to compare the statistics of the largest model with the model obtained using transformed variables and decide which model is better. We see that the largest model has a lesser 𝑅2 than the other. Even though the AIC and BIC of both models are nearly identical, the model with transformed variables has more statistically significant coefficients than the largest model because, almost all of the coefficients have p-value<0.05 and the p-values of citric\_acid and chlorides are much better.

Hence, our claim is that the model that used transformed variables may perform better on unseen data.

##### Deciding Further Steps

The model that was trained using these transformed variables has an R2 of -0.32 on Test which is not a big improvement. Hence, we need to find yet another technique to try to improve the model.  
Before proceeding to our next technique let us first analyze the Leverage plot and Influence plot to check the influence of outliers.





From the above Influence and residual leverage plots we feel that if we treat the outliers by Normalizing data, we may be able to improve R2.

Also, let us first confirm that the features are having different value ranges because, when a dataset has variables that have different value ranges, often, the loss function tends to give 'less importance' to columns with smaller value ranges. Data normalization is a technique to alleviate these undesirable effects and may help us improve R2.



We can say that 'chlorides', 'density', 'citric\_acid', 'volatile\_acidity', 'sulphates' have low dynamic range, 'pH', 'alcohol', 'fixed acidity', 'residual sugar' have medium range and 'free sulfur dioxide', 'total sulfur dioxide' have very high ranges.

There are many ways one can normalize this data. We are going for most frequently used StandardScaler() which standardizes features by removing the mean and scaling to unit variance and then train.

### Predictive Modeling using Ridge Regression with Regularization by scikit learn

Linear regression works by estimating coefficients for each independent variable that minimizes a loss function. However, if the coefficients are too large, the model starts modelling intricate relations to estimate the output and ends up overfitting to the training data. Such a model will not generalize well on the unseen data. In our case, the intercept and density variables have coefficients that are too high and low respectively. So, putting a constraint on the magnitude of coefficients can be a good idea to reduce model complexity. We do this using "Regularization" which penalizes large coefficients.

Ridge regression is an extension of linear regression where the loss function is modified to minimize the complexity of the model. In Ridge Regression, the OLS loss function is augmented in such a way that we not only minimize the sum of squared residuals, but also penalize the size of parameter estimates, in order to shrink them towards zero. This is mathematically written as

=

Notice that the penalty parameter is the square of the magnitude of the coefficients and α is the parameter we need to select. So, by changing the values of alpha, we are basically controlling the penalty term. The higher the values of α, the bigger the penalty and therefore the magnitude of coefficients are reduced.

In scikit-learn, a ridge regression model is constructed by using the Ridge class. We will be instantiating the Ridge Regression model with an alpha value of 0.01 after we transform/standardize the data.

Model Score on Test Set: 0.31

printing the ridge model coefficients

[ 0.1122537 -0.24524274 -0.0040585 0.29663793 -0.02491713 0.0846123

-0.07655952 -0.32036124 0.08086912 0.1031732 0.25394175 -0.15911866]

The displayed R2 of the model is on the Test set and is not an improvement. However, it is interesting to notice that all the model coefficients have shrunk towards 0 and none of them are extremely negative or extremely positive like in the case of conventional OLS, thereby reducing the complexity of the model.

We also noticed a warning message - " A large Cond. No. indicates strong multicollinearity or other numerical problems." in the OLS Summary of both the largest model and the model that transformed variables. Hence, we will be fitting a more sophisticated algorithm such as Support Vector Machine to improve the R2. This is covered in the next section.

### Conclusion – Linear Regression

In this section we demonstrated how Regression can be used to build a machine learning model to predict the wine quality. We analyzed UCI's Wine Quality Dataset to predict wine quality as a continuous value.

10% of the data was randomly picked as test data and 90% of the data was randomly picked as train data. Predictive models were built using the "Linear Regression", "Ridge Regression with Regularization", and "Support Vector Regression" approaches. A detailed study of the model summary revealed that the model is slightly complex. So, we looked at ways to normalize our data and impose regularization which gave a boost to R2. We believe that models that used Regularization with Ridge Regression and SVR may perform better on unseen data as these are simpler than other linear models.

We have also shown that a carefully selected GridSearch using Support Vector Regression and Standardized data improved the R2 from 0.32 to 0.44.

# Logistic Regression

### Logistic Regression Classification

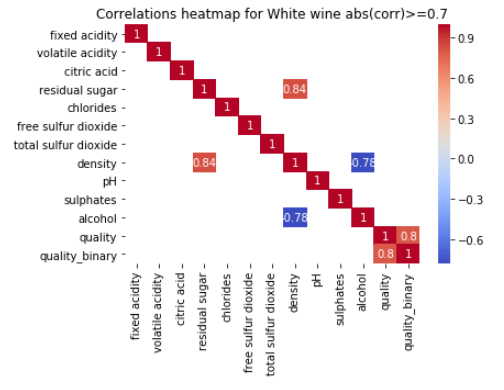
A Binary Logistic regression model was fitted to the White wine data and optimized to predict wine quality based on significant predictors. Classification models, such as logistics regression are sensitive to unbalance data, meaning the number of positive instances should be comparable to the number of negative instances. In order to use a binary (0 or 1) model and obtain a more balanced data, White wines were classified based on their wine quality ranking (1 to 10) into a binary class of premium wine (1) and economy wine (0):

* Quality score <= 5: Economy Wine
* Quality score >= 6: Premium Wine

### Checking Logistic Regression Conditions

There are two key conditions for fitting a logistic regression model:

1. Logistic regression requires the observations to be independent of each other (same as its underlying Binomial distribution). Given the relatively large number of samples this is a reasonable assumption in our case. We will also revisit this condition for the Training data set in a later section of the report.
2. Logistic regression also requires there to be little or no multicollinearity among the independent variables or predictors. To test this, we look at correlation values (shown on a heat map) among various pairs of predictors. To decide whether (the absolute value of) a correlation is "too high", the cut-off value was set to abs(r)>= 0.7 which is consistent to values used in the multiple-linear regression model and the cut-off value for Variance Inflation Factor VIF=1/(1-r2). A VIF value between 2 to 5, indicates significant collinearity between two predictors. In such case, one of the two predictors should be removed from the model.



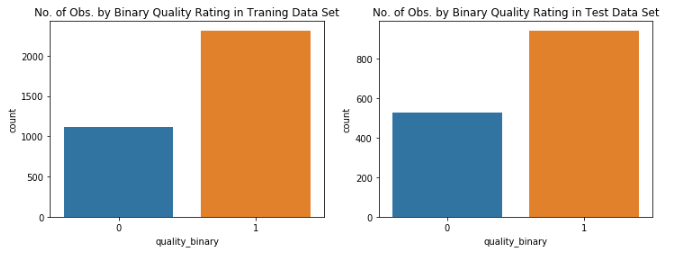
As seen from the correlation heat map, density has a correlation of +0.84 with residual sugar and of -0.78 with alcohol. This is consistent with physical laws of mixtures for density where residual sugar (heavier than water) increases density and alcohol that is lighter than water decreases density. Hence Density is a redundant predictor and it is removed from the model.

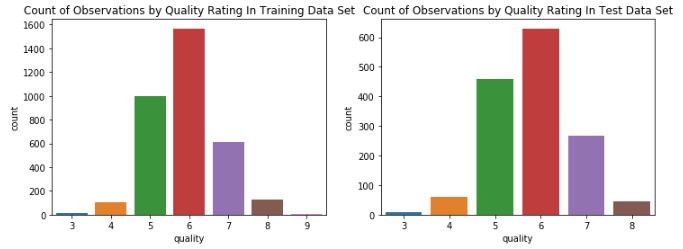
### Prepare Test/Train Data Set and Fit Model

The distribution and histogram for chlorides, free sulfur dioxide, and residual sugar (for White wine) appeared to be skewed (see graphs in section 2.3.3) and hence were candidate for log transformation to obtain a distribution that is more normal to improve model performance.

The data set was split between Train and Test sets using a split ration of 0.3. The train data has approximately twice as many observations with positive value (2315 records for 1 /premium wines) as those with negative target value (1113 records for 0/economy wines). Strongly unbalanced training datasets, may be improved by:

* Reducing the overrepresented class, by shuffling its rows and selecting n rows so that the two datasets - positive and negative classes will have comparable sizes.
* Inflating the underrepresented class, by duplicating entries randomly so that the final dataset has an approximately equal number of positive and negative instances.
* Another approach is to stratify the dataset that is to fetch randomly selected observations from each subclass and compose a new smaller dataset where both classes are equally represented.



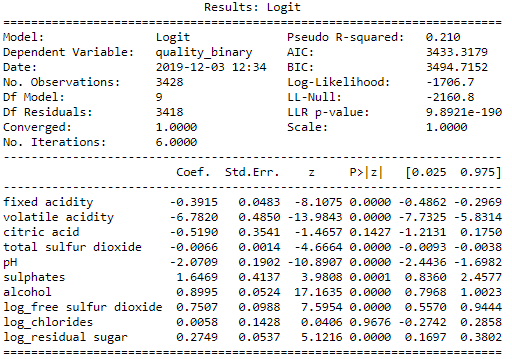


However, in our case, this difference is not too high, so we can proceed with the unmodified dataset. Also, the quality ranking profiles of premium and economy wines are similar between training and test data sets.

The two conditions for fitting a logistic regression model are assumed to hold in the training data set:

* Each predictor (xi) is linearly related to logit(pi) if all other predictors are held constant. While we can plot each predictor xi against logit(pi), we do not have a sufficiently large data set to ensure that all other predictors are held constant. We assume this condition holds.
* Each outcome Yi is independent of the other outcomes. For logistics regression, similarly to its underlying Binomial distribution, trials must be independent, given relatively large number of samples in each group of training set (1,113 and 2,315) this is a reasonable assumption.

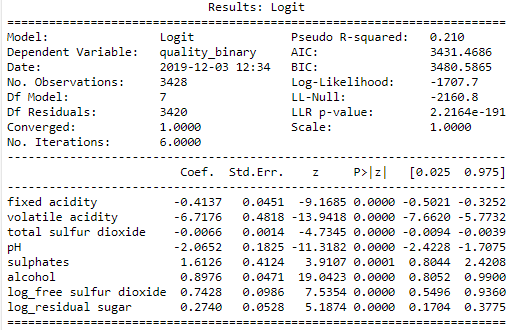
The Initial logistics model was fitted, the table below show the Summary Output Table



Interpreting the Regression Summary Output Table for the Initial model

* Converged indicates the solver (default is Newton method) converged successfully.
* It took 6 iterations for the solver to find the solution.
* Pseudo R-squared is one of the model quality measurement, is a modest 21%, relatively a low value.
* P-value for log-likelihood of the null model (LLR P-value) is nearly zero (less than 0.05); hence we reject the null hypothesis: model without predictors, where logit(𝑝)=𝛽0. Hence, we consider the model significant.
* P > | z | is the p-value for z-statistic associated with the given parameter, i.e. the probability that the true value of the given regression parameter is equal to the null value zero, given the set of observations. Except for citric acid and log\_chloride, p-values for remaining predictors are less than 0.05; hence they are statistically significant.

In the Second iteration of the model we removed predictors (citric acid and chloride) that were not statistically significant in the first iteration:



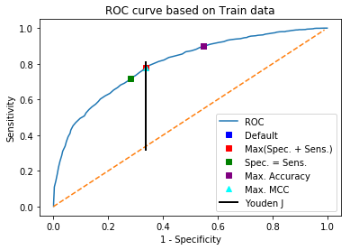
All p-values for the remaining predictors are less than 0.05; and hence are statistically significant, without much change to overall model performance based on Pseudo R-squared or AIC & BIC values.

### Tune Threshold value based on Training Set before evaluating model using Test data

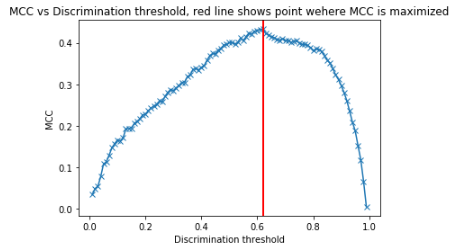
Next step was to tune the Threshold value based on the Training set before evaluating the model using Test data. Performance of the model using the Training data set with default threshold value of 0.5 is:

* Accuracy: 0.7538
* Mathews correlation coefficient (more robust against unbalanced binary response) 0.4002
* Precision, TP/ (TP+FP), fraction of correct estimates of all Predicted: 0.7737
* Recall/sensitivity, TP/ (TP+FN), fraction of correct estimates of True positives: 0.8981

The Receiver Operating Characteristic (RoC) curve was used to select the optimum Threshold value. As seen from the RoC curve point of maximum MCC (identical to maximum of the sum of specificity and sensitivity) is closest to the “error-free point” at (0,1). The corresponding threshold value (0.62) is the “optimal threshold” value.

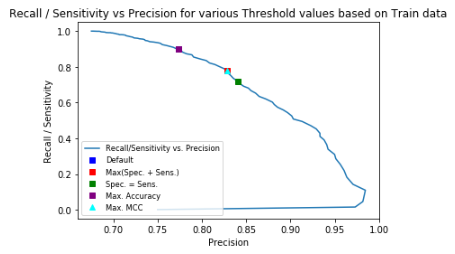


The corresponding threshold value, for maximum MCC shown in the graph below (0.62) is the “optimal threshold” value.



The wine maker has to balance the risk of a false positive (wine predicted as premium but is in truth an economy wine), with higher precision in order to avoid disappointing customers and possibly loss of brand equity with the risk of a false negative (wine predicted as economy but is in truth a premium wine), with higher recall/sensitivity in order to minimize loss of revenue and profit (may lead to higher volume and more customers in the economy segment).

As seen from graphs below model's precision, fraction of correct estimates of all predicted positives, increases with threshold value while recall/sensitivity, fraction of correct estimates of true positives, decreases. The Max (Spec. + Sens.) Youden's J criteria on the ROC curve seems to optimize the mentioned trade-off. Hence value of threshold value based on this criterion is used as final (replacing default value of 0.5) for the model and its performance valuation using test data.



Performance of the model using the Training dataset with final threshold value of 0.62 is:

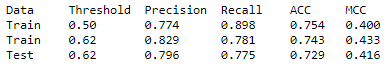
* Accuracy: 0.7430 (was 0.7538)
* Mathews correlation coefficient: 0.4332 (was 0.4002)
* Precision: 0.8286 (was 0.7737)
* Recall/sensitivity: 0.7810 (was 0.8981)

### Model Performance with Tune Threshold using Test data

The model was used to predict wine quality of the Test dataset (with density, citric acid, and residual sugar removed and threshold value tuned to 0.62):

* Accuracy: 0.7286
* Mathews correlation coefficient: 0.4163
* Precision: 0.7963
* Recall/sensitivity: 0.7752

Table below summarizes/compares the metrics for the scenarios:



Recall wine makers are most concern with the risk of a false positive (FP). As seen from chart below. False Positives, FP, are minimized by the model when evaluated against Test data.



### Conclusion - Logistic Regression

* In order to use Binary Logistics Model, wine quality rankings (1 to 10) were reclassified into a binary class of premium wines (1) and economy wines (0)
* Logistic regression condition that observations are independent of each other is assumed to be reasonable.
* Logistic regression condition that there is little or no multicollinearity among the independent variables was not initially met. Based on the analysis of correlations among various predictors, Density was removed from the list of predictors to satisfy the multicollinearity condition.
* Chlorides, free sulfur dioxide, and residual sugar were transformed using log transformation to obtain a more normal distribution and in turn improve model performance. Data was split between Training and Test sets using 30% as the split ratio.
* Balance of data points with positive and negative responses in the Training dataset was deemed adequate, 3 options to improve balance for future studies were recommended.
* First iteration of the Binary Logistics Regression model was fit to Training data. Two predictors, citric acid and chlorides, with p-values greater than 0.05 (statistically not significant) were removed from list of predictors.
* Second iteration of the model was fit to Training data with the remaining predictors. All p-values were less than 0.05; hence predictors were statistically significant, without much change to overall model performance based on Pseudo R-squared or AIC & BIC values (compared to the first iteration).
* The default threshold value (0.5) was tuned using the Training dataset and ROC curve (before evaluating the model using Test data).The maximal MCC, identical to maximum of the sum of specificity and sensitivity, was the closest point to the \*\*error-free point\*\* (0.1) and its corresponding threshold value was selected as the \*\*optimal threshold\*\* (0.62).
* The selected optimal threshold value also strikes a balance between the risks of false positive (wine predicted as premium but is in truth an economy wine. This could lead to a disappointed customers and loss of brand equity) and risk of a false negative (wine predicted as economy but is in truth a premium wine. This could lead to loss of revenue and profit).
* Model performance, with optimal threshold value, was evaluated using the Test dataset. Metrics, including Precision, Recall, ACC and MCC, were comparable to those from the Train dataset (also with optimal threshold value).
* While wine maker can explore more advanced models (with possibly higher performance), the final Binary Logistics Model is "adequate" to predict wine classification: premium versus economy wine, based on selected predictors.

# Conclusions

### Summary

Wine quality rating increased with more alcohol content for both Red and White wine. For Red wine, higher volatile acid, which turns wine to vinegar, lowers the quality of wine. Density has positive correlations with acidity and residual sugar (wine becomes "heavier than water") while it has a negative correlation with alcohol (ethanol is lighter than water). This negative correlation with alcohol level might suggest the negative correlation between density and qualities of wine.

There is more SO2 in White wine, added as a preservative to extend shelf life, higher amount of SO2 leads to lower quality rating ("rotten egg smell") in both Red and White wine.

Most of the Red wines in the dataset are dry (low residual sugar). Enriching the data with sweet Red wine will improve data analysis and modeling, if sweet wines are produced in this region. For White wine, residual sugar and alcohol content are negatively correlated. Residual sugar is the natural grape sugars leftover in a wine after the fermentation. During the fermentation process, yeast eats the sugar to produce alcohol. A dry wine is when yeast consumes most or all the sugar (often leads to higher levels of alcohol) while a sweet wine has left over sugar, or residual sugar.

### Results using Predictive Modeling methods

4898 data points of various wine quality records were analyzed in order to predict wine quality on a scale of 1-Good, 2-Medium, 3-Poor using various characteristics of wine such as fixed acidity, density, pH etc. Two predictive models were built using the "Random Forest Classifier" approach and the "Logistic Regression Classifier" approach respectively. 30% of the data was randomly picked as test data and 70% of the data was randomly picked as train data. The Predictive Model built using Random Forest Classifier resulted in 82% accuracy without scaled data and 80% accuracy with scaled data. The model was then tuned using the cross-validation test method but did not result in a higher prediction rate. The Predictive Model built using Logistic Regression Classification approach resulted in 71% accuracy.

We also demonstrated how Support Vector Machine can be used to build a machine learning model to predict the wine quality. We analyzed the dataset and found that there is a significant class imbalance amongst the 7 quality classes. Knowing that we would need a non-linear separating boundary, we set up a Grid Search for SVM parameters. We searched over several combinations of kernels, C and Gamma. However, we found that we could go only as far as 60% accuracy. Detailed study of model predictions and errors revealed that the model is able to predict classes 5 and 6. We decided to go further in that direction and decided to build a Binary Classifier which can predict whether the wine quality is less than 6 or not. This time we got encouraging 74% model accuracy. We also, looked at ways of normalizing our data and that gave another 5% accuracy boost. Our final binary wine prediction model had promising 79% accuracy on the validation set and 78% on the hold-out test set.

Based on our analysis of the Wine dataset, the Random Forest Classifier method yielded higher prediction accuracy.

# Appendix

### Source of Data

Source of Data: <https://archive.ics.uci.edu/ml/datasets/Wine+Quality>

### Data collection:

The dataset used in this analysis comes from the University of Minho’s website under Department of Information Systems. The datasets reside in two separate CSV files, broken down as White and Red wine. Each of the samples in the datasets have a unique index number. In total, twelve attributes of each sample make up the datasets. Of those attributes, quality is our main subject and the other eleven attributes are factors that potentially affect the perceptions of quality. Descriptions to the attributes are provided as per below.

* Two CSV files (delimiter by semicolon ";") one for Red Wine (with 1599 rows) and one for White Wine (with 4898 rows), we may choose both or just one/White wine for our project.
* Each file includes a header row for columns titles.
* There are 12 attributes/columns (including one for quality) plus Type (Red or White) if the files are combined.
* There are no missing values; data types are real, integer and string.

The dataset used in this analysis is comprised of 1599 instances of Red wine samples and 4898 instances of White wine samples. The independent variables (inputs) include physicochemical tests and the dependent variable (output) is based on sensory data (median of at least three evaluations made by wine stewards). Each wine steward graded the wine quality between 0 (very bad) and 10 (very excellent). The wine samples are Portuguese “Vinho Verde”.

### Data preparation

The first step of the analysis was to examine the datasets for null value or other potential issues. Pulling the dataframe info in Python allows quick identification of null value in the datasets. As the result, the White wines dataset showed all 4898 records with non-null numerical values for all twelve columns. The Red wine datasets also showed same results for all its 1599 samples. This confirmed that there were no missing data points, nor attributes that needs to be converted to numerical value for further analysis. Due to the completeness of the data, no records need to be modified or dropped at the preparation stage.

The second step was to combine the White wine and the Red wine datasets into one dataframe. This allows the analysis between the two different wine groups, especially when normalized with the same scale (next step).

The third step was to create a normalized version of the same data frames (scale to global max and min). Normalization scheme does not modify shape of original distribution. The process simply adjusted the scales of each attributes to between 0 and 1.

Normalized values where calculated by subtracting the global minimum value from each individual value, and divided by the range for the attribute (global maximum minus global minimum): **(X-Min)/ (Max-Min)**

### Description of attributes:

* **Fixed Acidity** (tartaric acid - g / dm3)
  + fixed or non-volatile (do not evaporate readily) acids in wine
* **Volatile Acidity** (acetic acid - g / dm3)
  + the amount of acetic acid in wine, high levels can lead to a vinegar taste
* **Citric Acid** (g / dm3)
  + in small quantities, citric acid can add ‘freshness’ and flavor to wines
* **Residual Sugar** (g / dm3)
  + amount of sugar remaining after fermentation, wines with greater than 45 grams/liter are considered sweet
* **Chlorides** (sodium chloride - g / dm3)
  + the amount of salt in the wine
* **Free Sulphur Dioxide** (mg / dm3)
  + prevents microbial growth and the oxidation of wine, above 50 ppm, SO2 becomes evident in aroma and taste of wine
* **Total Sulphur Dioxide** (mg / dm3)
* **Density** (g / cm3)
  + the density of wine, will differ from water (1? /) depending on the alcohol and sugar content
* **pH**
  + describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic); most wines are between 3-4 on the pH scale
* **Sulphates** (potassium sulphate - g / dm3)
  + additive which can contribute to sulfur, acts as an antimicrobial and antioxidant
* **Alcohol**
  + the percent alcohol content by volume
* **Output variable** (based on sensory data)
  + quality (score between 0 and 10)

### Data Analysis

##### The First Analysis

The first analysis was counting of quality scores by the color of wine. The result indicates that most of the quality rankings for both Red and White wine are concentrated toward the middle. Hence there are fewer low scored (e.g. 3 or 4) or high scored (e.g. 8 or 9) wines comparing to those with average / middle scores (e.g. wine with quality rating of 5 and 6).

Another observation is that none of the Red wines scores 9, while five White wines get the highest score. However, since there are over 3 times the records of White wine comparing to Red wine, at this stage we cannot draw any conclusions to the distribution of quality by color.

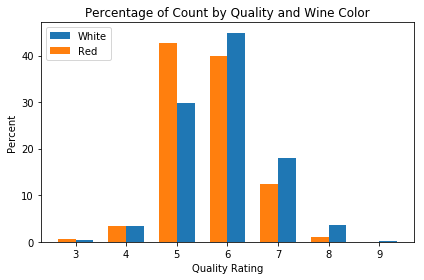
| **Quality** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Color** |  |  |  |  |  |  |  |
| **Red wine** | 10 | 53 | 681 | 638 | 199 | 18 | 0 |
| **White wine** | 20 | 163 | 1457 | 2198 | 880 | 175 | 5 |

##### The Second Analysis

The second analysis is to calculate the percentage of wine by color and by quality. The count in the above table was divided by the total count by wine color in this calculation.

| **Quality** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Color** |  |  |  |  |  |  |  |
| **Red wine** | 0.6% | 3.3% | 42.6% | 39.9% | 12.4% | 1.1% | 0.0% |
| **White wine** | 0.4% | 3.3% | 29.7% | 44.9% | 18.0% | 3.6% | 0.1% |

Based on the illustration below, White wine has higher proportion in all quality ratings above 5, while Red wine has a much larger concentration at quality rating 5.



##### The Third Analysis

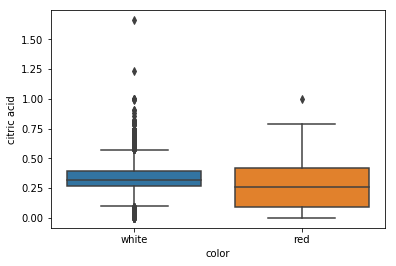
Observing a difference in quality between Red wine and White wine, the Third step for the analysis was to investigate the difference between the other attributes of wines by color. The difference between Red and White wine can help explaining what contributes to the overall quality of wines.

The average of each of the twelve attributes was calculated by color. The averages of the White wine attributes were then divided by the average of the same attributes of the Red wine. Among the twelve attributes, White wine has significantly higher residual sugar, free and total sulfur dioxide. Red wine on the other hand, has significantly higher volatile acidity, and much higher chlorides comparing to White wine.

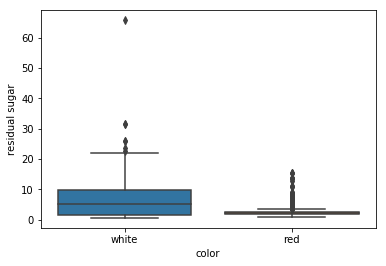
|  |  |  |  |
| --- | --- | --- | --- |
| **color** | **red** | **white** | **white/red** |
| **fixed acidity** | 0.37 | 0.25 | 0.82 |
| **volatile acidity** | 0.30 | 0.13 | 0.53 |
| **citric acid** | 0.16 | 0.20 | 1.23 |
| **residual sugar** | 0.03 | 0.09 | 2.52 |
| **chlorides** | 0.13 | 0.06 | 0.52 |
| **free sulfur dioxide** | 0.05 | 0.12 | 2.22 |
| **total sulfur dioxide** | 0.09 | 0.30 | 2.98 |
| **density** | 0.19 | 0.13 | 1.00 |
| **pH** | 0.46 | 0.36 | 0.96 |
| **sulphates** | 0.25 | 0.15 | 0.74 |
| **alcohol** | 0.35 | 0.36 | 1.01 |
| **quality** | 0.44 | 0.48 | 1.04 |

Another part of the analysis was to review the distribution of the attributes by wine color. There are several interesting observations:

1. Red and White wine have similar citric acid on average, but Red wine has a much wider range of distribution.



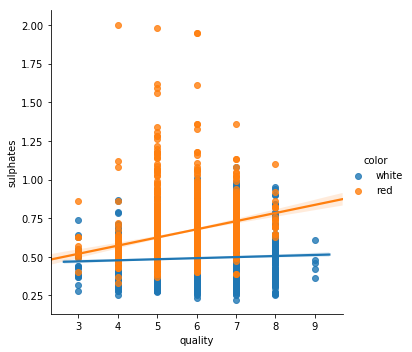
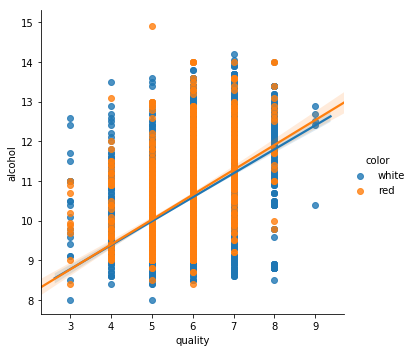
1. Red wine not only has much lower residual sugar comparing to White wine, the distribution is also highly concentrated at the average, which is only 2.54 grams. The sample appears to contain more dry Red wines while whites have both dry and sweet, suggest augmenting future data set to improve predictive model.



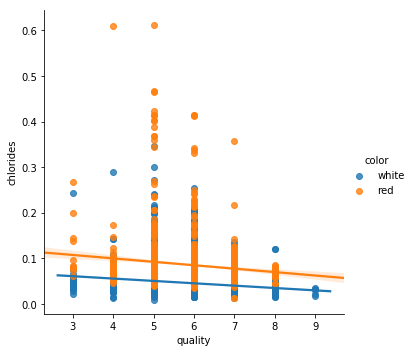
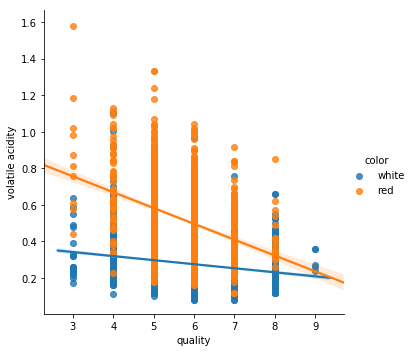
##### The Fourth Analysis

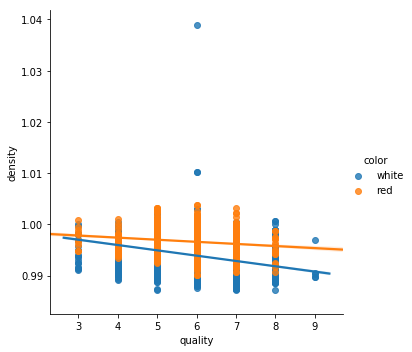
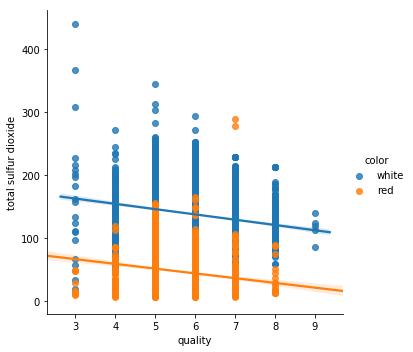
The fourth analysis is to compare the correlation between each of the attributes and quality, by wine color.

The first observation in this analysis indicated clear positive correlation between the level of alcohol and wine quality. The level of sulfates also seems positively related to wine quality, more noticeable within the Red wine group.

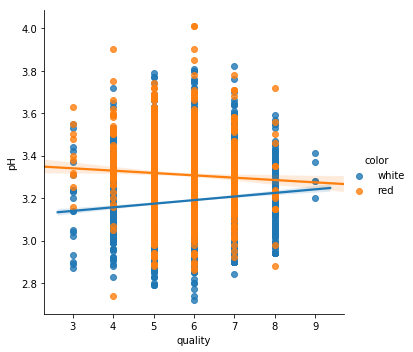
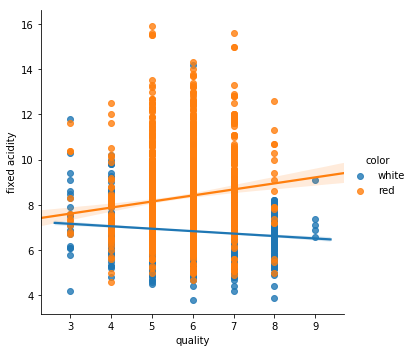


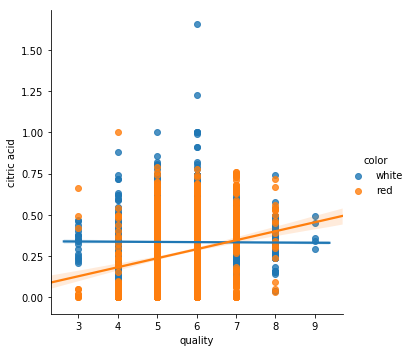
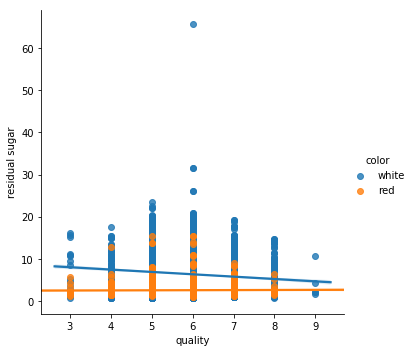
The second observation identified a group of attributes that might have negative impact on the qualities of wines. Those included volatility acidity, chlorides, total Sulphur dioxide, and density.





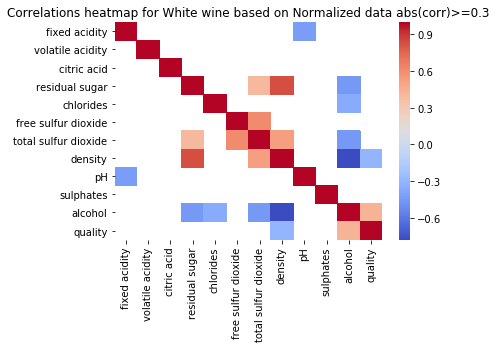
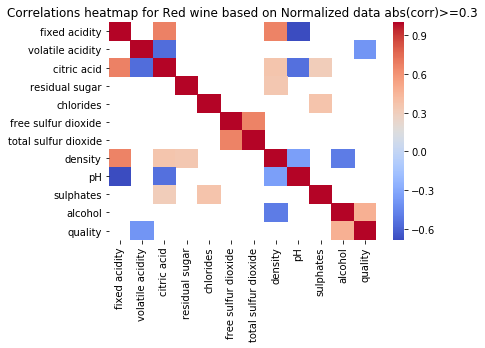
The third observation is to capture those attributes that seems less obvious in the correlation with the quality score of wines. Fixed acidity seems positively related to the Red wine quality, but negatively related to the White wine quality. Similarly, PH and residual sugar seem to show mixed correlation with Red and White wine. Citric acid on the other hand, does not seem to affect the White wine quality, but adds to the Red wine quality.





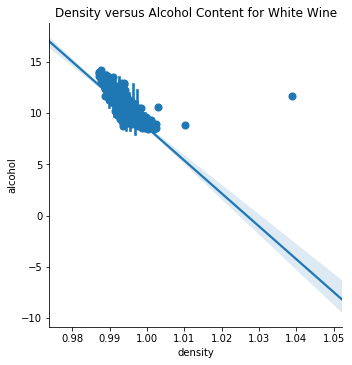
##### The Fifth Analysis

Now we have identified the relationship between each attribute and the quality of Red and White wine. The fifth analysis is to identify which of these correlations are statistically significant. In other words, which of these attributes matter in our model and which do not? Below is a group of heatmaps that show correlation between each attributes of Red wine with absolute correlation of over 0.3. This chart captures any positive or negative correlation when significant enough.



As demonstrated in the heat maps, alcohol level is positively correlated to the quality score of both Red and White wine, confirming our previous analysis. Based on the second observation of the Fourth Analysis, volatile acidity, as expected, has a negative impact on Red wine quality. It however has less impact on White wine quality.

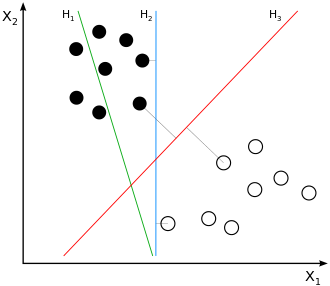
Density of White wine, based on the heat map on the right, also has a negative correlation with the quality of the wine. The relationships between these few selected attributes are also considered. Density has positive correlations with residual sugar while it has a negative correlation with alcohol (ethanol is lighter than water). Therefore, the reason why density is negatively correlated to the quality of wine might be due to its negative correlation with alcohol level.



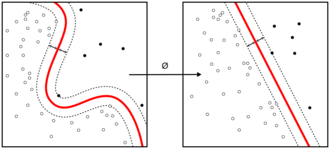
Volatile acidity level in Red wine is strongly negatively correlated to the citric acid level in Red wine. Based on previous analysis, that may explain the positive relationship between the citric acid level and the Red wine quality, although not significant enough to show in the heat maps.

### SVR

Support vector machine (SVM) is a popular machine learning technique well suited for both classification and regression tasks. Consider the dataset in following image:



In this image, we have got few samples belonging to two classes: one of them is represented in white circles and other in dark circles. X1 and X2 are two features which are being studied for their accuracy of predictions of the two classes of interest. The task of the classifier is to find a 'boundary' between samples of the dataset such that samples get separated as well as possible. The classifiers carryout a statistical analysis on the (training) dataset and find such boundaries automatically. While finding an optimal boundary a classifier may evaluate many boundary candidates, like H1, H2 and H3 shown in the figure and decide on a best candidate. Note that, H1 does NOT separates all samples but H2 and H3 do. Though both H2 and H3 are valid boundaries, H3 is considered better since it leaves out 'maximum' space for 'unseen' samples (samples not in training set). SVM is formulated to efficiently find the 'maximal margin hyperplane' (the red line) for a given dataset. In other words, SVM is guaranteed to find a maximal margin hyperplane if one exists (i.e. data is linearly separable). That brings us to the question, what happens when data is NOT linearly separable. Now consider below dataset (left part of the image):

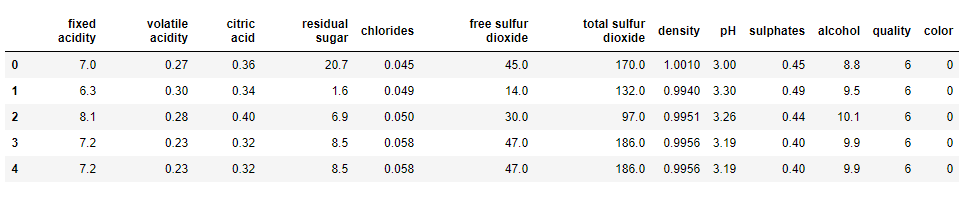


Like previous dataset, this dataset too has samples belonging to two classes and there two predictors are in x and y axis. Just that, there is NO nice line separating them anymore. What makes SVMs popular is their ability to work well even in this type of datasets. SVM uses a notion of 'kernel' to map the original dataset to a very high dimensional space, where the samples become linearly separable (please see right part of the image). SVMs do this automatically for us. This kernel mapping is based on a strong mathematical concept which makes mapping to very large dimensional space computationally feasible.

In this work, we will be using SVM implementation provided by python's Scikit learn library. The library has several in-built kernels including Radial Basis Function (RBF).

##### Load Data

Let us start with loading the data and take a quick peek at it for sanity check. The dataset contains details of White and Red wine in separate csv files. We load both files and combine them in a single data frame.



The dataset appears to have been loaded properly. We have 6497 samples in our dataset. The column 'quality' is our target column and remaining 11 columns provide various physicochemical properties of wine samples. We have added one additional column 'color' which is set 0 for Red wine and 1 for White.

Next, let us look at the distribution of our target column 'quality'.

##### Train and Test Set

Process of building a machine learning model requires significant amount of parameter tuning before we arrive at a good model. In this process, however, we might end up overfitting to our data. Hence, at the end of this process we need to 'confirm' that we have not overfit and our model indeed works well on unseen data. All that we must do is to keep few samples out of the training phase, call that a test set and use them only one time at the end of the training phase. Usually, performance of the model is reported on such test set. We have 5847 samples in train set and remaining about 10% in test set.

##### Data Normalization

When dataset columns have different value ranges, often, the classifiers tend to give 'less importance' to columns with smaller value ranges. Data normalization is a technique to alleviate this undesirable effect. Below table lists the dynamic range of various data attributes in our dataset.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **chlorides** | **density** | **citric acid** | **volatile acidity** | **sulphates** | **pH** | **alcohol** | **fixed acidity** | **residual sugar** | **free sulfur dioxide** | **total sulfur dioxide** |
| **Min** | 0.009 | 0.9871 | 0.00 | 0.08 | 0.22 | 2.72 | 8.0 | 3.8 | 0.6 | 1.0 | 6.0 |
| **Max** | 0.611 | 1.0389 | 1.66 | 1.58 | 2.00 | 4.01 | 14.9 | 15.9 | 65.8 | 289.0 | 440.0 |

We can say that, 'chlorides', 'density', 'citric acid', 'volatile acidity', 'sulfates' have low dynamic range. 'pH', 'alcohol', 'fixed acidity', 'residual sugar' have medium dynamic range and 'free sulfur dioxide', 'total sulfur dioxide' have very high value ranges.

There are many ways one can normalize this data. We are going for most frequently used StandardScaler() which standardizes features by removing the mean and scaling to unit variance. After normalizing the data, we reran the step of Grid-Search for the best model. On this standardized dataset, we got 5% improvement in accuracy. So, accuracy of our final model is 79% on validation set.

Our final SVM Model has 'RBF' as its kernel, penalty term 'C' is set at 10 and RBF Gamma is 0.4. And we need data normalization.

##### Build and test SVR Model

We can tune three parameters of SVM namely,

* + Kernel Linear or non-linear (for example: RBF)
  + C Penalty term to prevent overfitting
  + Gamma Kernel coefficient

Parameters 'C' and 'gamma' can be set to any arbitrary positive floating-point numbers implying that we can try really large number of different models on the same train set. Hence, we need a mechanism to 'select' the best one of these models. That is where cross-validation comes into picture. Idea is that we divide our train set into k subsets, train the model with 'one particular combination of above parameters' on k-1 subsets and validate on the remaining subset. We do this k times leaving out different subset every time and fixing a parameter combination. At the end of kth step, we average out the performance across k steps. That will be the performance for that parameter combination. Now we change the parameter combination and repeat above k steps until we have tried all the parameter combination of interest. That is lot of computation. Fortunately, python utility GridSearchCV() can do that automatically for us.

It is common practice to report the model accuracy on a hold-out test set.

0.4477171108966258

We tried both 'linear' kernel and 'rbf'. For linear kernel, we tried 4 different penalty values 1, 10, 100 and 100. Additionally, for RBF kernel we experimented with 9 different gamma values ranging from 0.1 to 0.9. To assess a combination, we utilized cross validation with k = 5.

Despite such exhaustive Grid Search, the best model 𝑅2 is just 0.44. Though this number is still small, we have achieved nearly 12% improvement when we switched from Linear Regression algorithm to Support Vector Regression which is encouraging. Maybe we can do better if we use XGBoost or Random Forest with a more refined GridSearch.