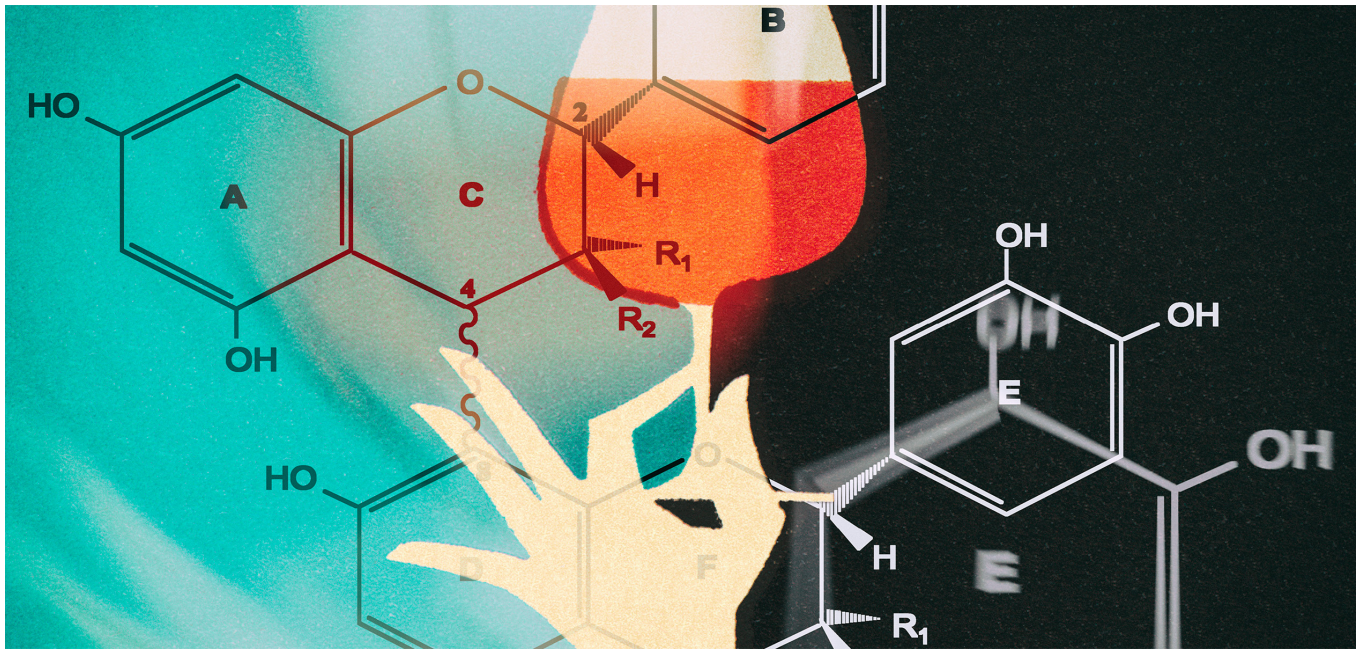
Predicting Wine Quality by comparing Linear Regression with Machine Learning Techniques.

Joshua Paul Barnard

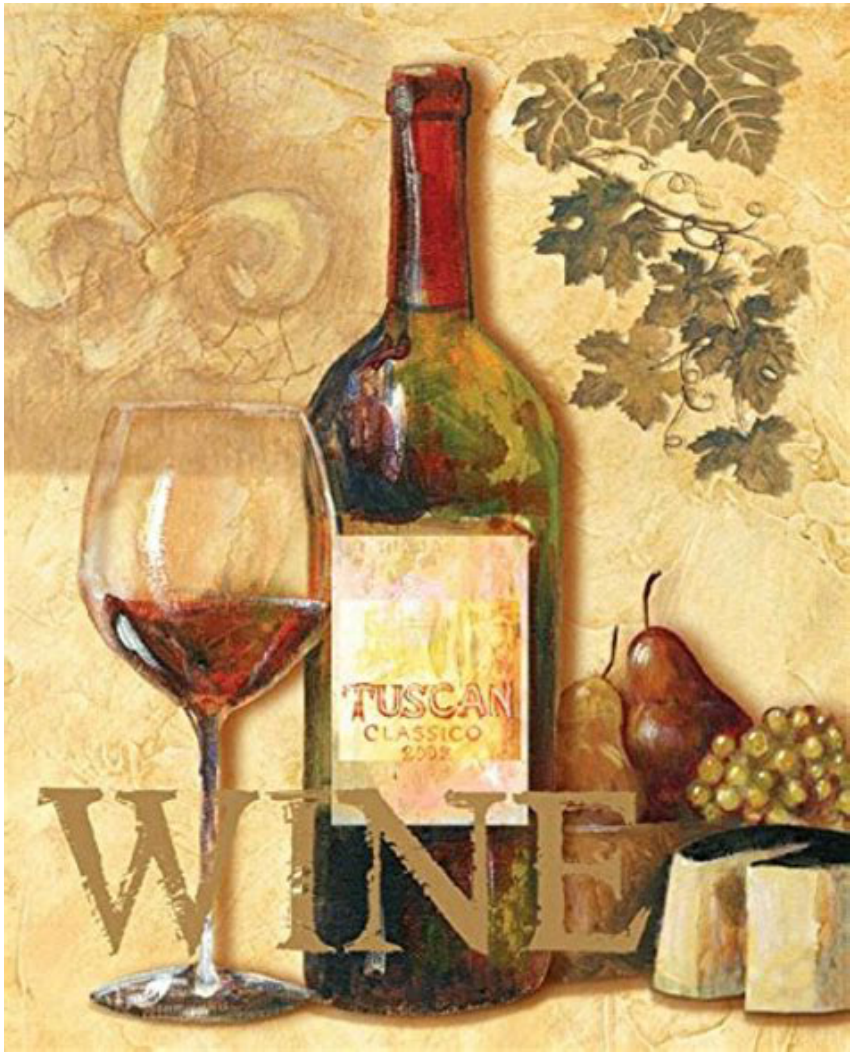


What is Wine Quality?

- The quality of wine is complex and is considered to be difficult to understand, with its quazi-aesthetic character and relationship to personal taste, it is peculiarly hard to pinpoint.
- Some suggest that quality is more objective, with a crucial determinant in the quality of wines to be an absence of faults. While others suggest that a wines quality is perceived, and is based on its 'fit for a purpose', which brings up more dimensions as to what a purpose even is.
- Due to the concept of wine quality being hard to define precisely, we will focus on the more objective physicochemical properties to determine the quality of wine.

Motivation and Reasons

- A winery is a business, and just like any business they need to know which wines are worth distributing and marketing. Wine quality can be used as a metric to help determine the marketability of one wine to another.
- Understanding which chemical are responsible for a wine's quality allows for winemakers to add and remove wanted and unwanted chemicals from their wine wines while maintaining quality.



The Source of our Data

Paulo Cortez
Department of Information Systems — School of Engineering — University of Minho, Portugal

P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.

- Two datasets are related to red and white variants of the "Vinho Verde" wine from the north of Portugal.
- Vinho verde is a unique product from the Minho (northwest) region of Portugal. Medium in alcohol, is it particularly appreciated due to its freshness (specially in the summer).
  More details can be found at: http://www.vinhoverde.pt/en/

The source data

- Our data came from two separate datasets, one for white wines and one for red wines.
- Both were saved as .csv, yet the delimiter was a semicolon and not a comma.

```
red_wine = np.genfromtxt(wine_quality_data["red wine data"], delimiter=';', skip_header=1)
white_wine = np.genfromtxt(wine_quality_data["white wine data"], delimiter=';', skip_header=1)
```

- The White Wine Dataset contains 4898 samples.
- The Red Wine Dataset contains 1599 samples.
- Combined we have 6497 total samples.

Merging the datasets and creating indicator variables

- We then created 3 new variables in the two separate datasets: wine_type - is a string variable containing: 'Red Wine' or 'White Wine'
- Red_Wine - is an indicator (dummy) variable to allow us to use the category in our regression analysis. 1 indicates Red Wine, and 0 indicates White Wine.
- White_Wine - is an indicator (dummy) variable to allow us to use the category in our regression analysis. 1 indicates White Wine, and 0 indicates Red Wine.

Red_Wine	White_Wine	wine_type
1	0	Red Wine
1	0	Red Wine
1	0	Red Wine
1	0	Red Wine
1	0	Red Wine

What transformations were tried

During the modeling stage we noticed some issues with the residuals for the Quality and Alcohol variables.

We tried various transformations, including:

- Log Transformations (base: 2, e, and 10)
- Power Transformations (2, 3, and 4)
- Root Transformations (2, 3, and 4)
- Inverse Transformations
- Reciprocal Transformations
- Centering and Scaling Transformations (standardization)

Various transformations were attempted, but they did not help.

SQL Database

- We created the architecture for our SQL database.
- The database includes all of our original variables, plus our new categorical variable and its 2 indicator variables.
- We then loaded our dataset into our new database using sqlite3.

```
1   DROP TABLE IF EXISTS wine_quality;
2   CREATE TABLE IF NOT EXISTS wine_quali
3       wine_ID INTEGER PRIMARY KEY,
4       quality INTEGER,
5       Red_Wine INTEGER,
6       White_Wine INTEGER,
7       wine_type STRING,
8       fixed_acidity DECIMAL,
9       volatile_acidity DECIMAL,
10      citric_acid DECIMAL,
11      residual_sugar DECIMAL,
12      chlorides DECIMAL,
13      free_sulfur_dioxide DECIMAL,
14      total_sulfur_dioxide DECIMAL,
15      density DECIMAL,
16      pH DECIMAL,
17      sulphates DECIMAL,
18      alcohol DECIMAL
19  );
```

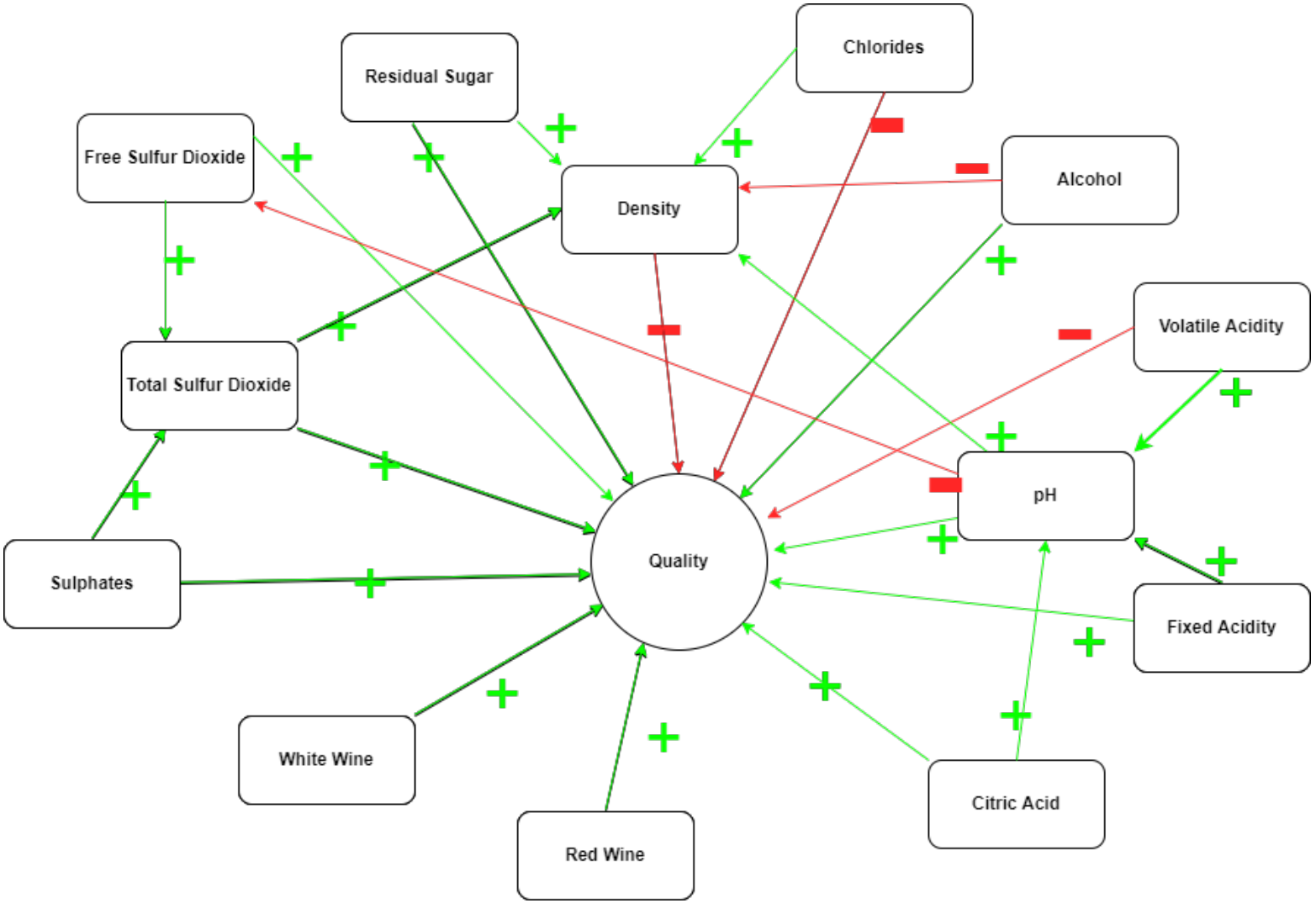
About the Variables

- **Quality:** Is the median of at least 3 evaluations made by wine experts, with each expert grading the wine quality from 0 to 10, with 0 representing 'very bad' and 10 indicating 'very excellent'
- **Wine Type:** People have preferences for different the types/colours of wine, which can influence a judge's expectations and perception of quality.
- **Fixed Acidity:** Usually refers to the amount of non-acetic acids in a wine. In this dataset is refers to the concentration of Tartaric Acid.
- **Volatile Acidity:** The amount of acetic acid in a wine. VA is vinegar, and if levels are too high it can lead to an unpleasant, vinegar taste. A measure of volatile acidity is used routinely as a indicator of wine spoilage. A taster's sensitivity to acetic acid will vary, but most people can detect excessive amounts at around 600 mg/L.
- **Citric Acid:** Can add 'freshness' and flavor to wines. One of the three primary acids found in wine grapes, along with tartaric, and malic.
- **Residual Sugar:** A mixture of glucose and fructose, it is the amount of sugar remaining after fermentation stops, wines are typically between 1 gram/liter and 45 grams/liter.

About the Variables

- **Chlorides:** Sodium Chloride, the amount of salt in the wine. People typically do not enjoy salty wines.
- **Free Sulfur Dioxide:** Free form of SO2, helps prevent microbial growth and the oxidation (spoilage) of wine. Free SO2 is unnoticeable until 50ppm when it imparts a chemically aroma and taste in the wine.
- **Total Sulfur Dioxide:** The amount of free and bound forms of S02. High concentrations can be an indication of wine faults.
- **Sulphates:** Potassium Sulphate, a wine additive which can contribute to sulfur dioxide (S02) levels. Has a bitter, salty taste.
- **Density:** The density of wine is close to that of water depending, with dense wines being thick and syrupy.
- **pH:** A scale from 0 (very acidic) to 14 (very basic); most wines are between 3-4 on the pH scale. White Wines are usually more acidic from 3.0 and 3.4. Red Wines are usually less acidic, from 3.3 and 3.6.
- **Alcohol:** The percent alcohol content of the wine. Alcohol helps bring out the aromas and tastes in a wine, so it will be more apparent if a wine is of poor quality if it has a higher alcohol content. Conversely, a good quality wine will be more pleasant and enjoyable with higher alcohol.

Causal Loop Diagram



Checking for Unreasonably Low or High Values

All of the variables look appropriate, with none of them have a minimum or maximum values outside their expected ranges.

	quality	fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH	sulphates	alcohol
count	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000
mean	5.818378	7.215307	0.339666	0.318633	5.443235	0.056034	30.525319	115.744574	0.994697	3.218501	0.531268	10.491801
std	0.873255	1.296434	0.164636	0.145318	4.757804	0.035034	17.749400	56.521855	0.002999	0.160787	0.148806	1.192712
min	3.000000	3.800000	0.080000	0.000000	0.600000	0.009000	1.000000	6.000000	0.987110	2.720000	0.220000	8.000000
25%	5.000000	6.400000	0.230000	0.250000	1.800000	0.038000	17.000000	77.000000	0.992340	3.110000	0.430000	9.500000
50%	6.000000	7.000000	0.290000	0.310000	3.000000	0.047000	29.000000	118.000000	0.994890	3.210000	0.510000	10.300000
75%	6.000000	7.700000	0.400000	0.390000	8.100000	0.065000	41.000000	156.000000	0.996990	3.320000	0.600000	11.300000
max	9.000000	15.900000	1.580000	1.660000	65.800000	0.611000	289.000000	440.000000	1.038980	4.010000	2.000000	14.900000

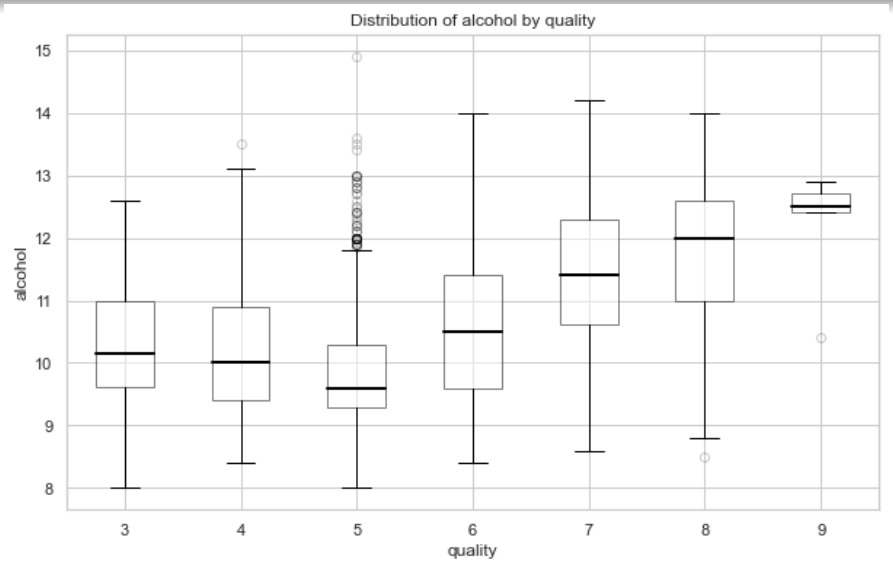
Analyzing our Variables

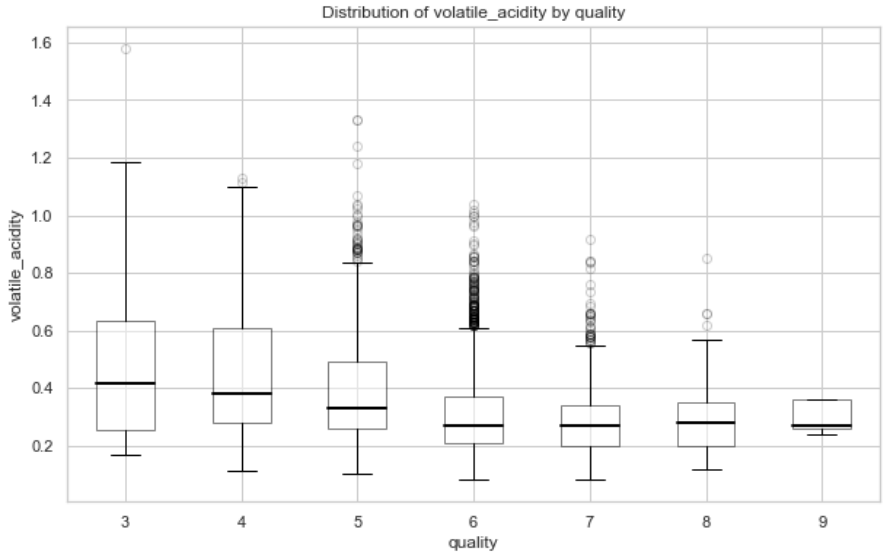
Highest Correlated Variables with Quality:

- Alcohol: 0.45
- Density: -0.32
- Chlorides: -0.30
- Volatile Acidity: -0.26

Lowest Correlated Variables with Quality:

- Total Sulfur Dioxide: -0.055
- pH: 0.032
- Sulphates: 0.030
- Residual Sugar: -0.017





Check for Multicollinearity

Based on our Domain Knowledge there might be multicollinearity between the following variables:

- Total Sulfur Dioxide with Free Sulfur Dioxide and Sulphate.
- pH with fixed acidity, volatile acidity, citric acid, total sulfur dioxide, free sulfur dioxide, and sulphates.
- Density with every other variable except quality, Red Wine and White Wine.

The following variables had correlations above 0.5 (or less than -0.5):

- Total Sulfur Dioxide and Free Sulfur Dioxide: 0.72
- Density and Alcohol: -0.69
- Density and Residual Sugar: 0.55

These correlations make sense based on our domain knowledge, and are not high enough to prevent us from getting good estimates of our coefficients.

Total Sulfur Dioxide:

	feature	r	rho
0	free_sulfur_dioxide	0.720934	0.741438
1	sulphates	-0.275727	-0.256745

pH:

	feature	r	rho
0	fixed_acidity	-0.252700	-0.250044
1	volatile_acidity	0.261454	0.194876
2	citric_acid	-0.329808	-0.285905
3	total_sulfur_dioxide	-0.238413	-0.242719
4	free_sulfur_dioxide	-0.145854	-0.164699
5	sulphates	0.192123	0.254263

Density:

	feature	r	rho
0	pH	0.011686	0.011777
1	fixed_acidity	0.458910	0.434056
2	volatile_acidity	0.271296	0.261437
3	citric_acid	0.096154	0.065690
4	total_sulfur_dioxide	0.032395	0.061540
5	free_sulfur_dioxide	0.025717	0.005841
6	sulphates	0.259478	0.274792
7	alcohol	-0.686745	-0.699442
8	residual_sugar	0.552517	0.526664
9	chlorides	0.362615	0.590729

Check for Outliers

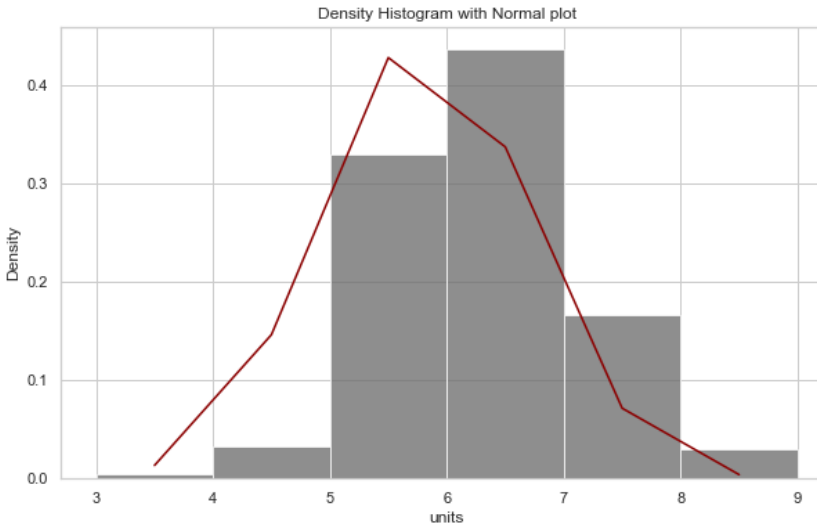
To check for outliers we will examine points which are below Q1 - 1.5 x IQR and above Q3 + 1.5 x IQR.
Based on our EDA we expect potential outliers to exist for: volatile acidity, citric acid, chlorides, free sulfur dioxide, total sulfur dioxide, sulphates, density.

```
509 potential outliers for Citric Acid
377 potential outliers for Volatile Acidity
286 potential outliers for Chlorides
191 potential outliers for Sulphates
62 potential outliers for Free Sulfur Dioxide
10 potential outliers for Total Sulfur Dioxide
3 potential outliers for Density
```

We kept all of the outliers as they were within reason, and carelessly ignoring outliers can lead to fragile models which lack robustness.

The Null Model

- Quality appears normally distributed.
- Quality is integers from 0 to 10, which is ordinal data with each grade being a different rank. As quality is not a concrete concept, we will treat quality as if it were continuous and interpret the results as more of a percent from 0 to 100.
- The Null Model:
  - Mean: 5.82 (58.2%)
  - Standard Deviation: 0.87
  - Which gives us a 95 percent chance for a score to fall between 4.1 and 7.5
- When comparing the null model with sampled data, our prediction is off by 0.97 on average.



What is Linear Regression?

- Linear Regression can be described as:

$$y = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n \ \epsilon \ ,$$
$$\epsilon \sim N(0, \sigma)$$

- Bootstrapping is where we resample from our data, for each sample, and calculate new values for the outcomes and parameters, estimate their distribution and get a better idea of the uncertainty associated with our model. This allows us to create 95% Bayesian Credible Intervals and satisfy the OLS assumption of normally distributed errors.
- N-Fold Cross Validation borrows from the idea of bootstrapping to divide the data into N folds (or sections) and loop over each fold to use it as the test set against each training set. This will yield N-estimates of MSE.

Our Linear Regression Models

Our best models based upon our Domain Knowledge:

- All-in Model: "quality ~ Red_Wine + White_Wine + fixed_acidity + volatile_acidity + citric_acid + residual_sugar + chlorides + free_sulfur_dioxide + total_sulfur_dioxide + density + pH + sulphates + alcohol"
- Simplest Model: "quality ~ volatile_acidity + alcohol"
- Interactions Model: "quality ~ Red_Wine + White_Wine + fixed_acidity + volatile_acidity + citric_acid + residual_sugar + chlorides + free_sulfur_dioxide + total_sulfur_dioxide + density + pH + sulphates + alcohol + total_sulfur_dioxide:sulphates + total_sulfur_dioxide:free_sulfur_dioxide + total_sulfur_dioxide:pH + pH:free_sulfur_dioxide + pH:sulphates + pH:citric_acid + pH:fixed_acidity + pH:volatile_acidity + pH:alcohol + pH:chlorides + pH:residual_sugar + density:pH + density:sulphates + density:total_sulfur_dioxide + density:free_sulfur_dioxide + density:residual_sugar + density:chlorides + density:alcohol + density:volatile_acidity + density:fixed_acidity + density:citric_acid"

Comparing our models

- Our metrics for comparing models will be error (σ^2) and the coefficient of determination (R^2), with error being the most important.
- Our goal is to construct a model with the lowest σ^2, highest R^2, and beta coefficients which match our CLD.
- All three models performed better than the null model, but only slightly better. The model with interaction terms performed the best.

Table: Comparing Error and R^2

Model	σ^2	$\mathbf{R^2}$
Null	0.88	
Simple	0.75	0.26
All-In	0.73	0.30
Interactions	0.71	0.33

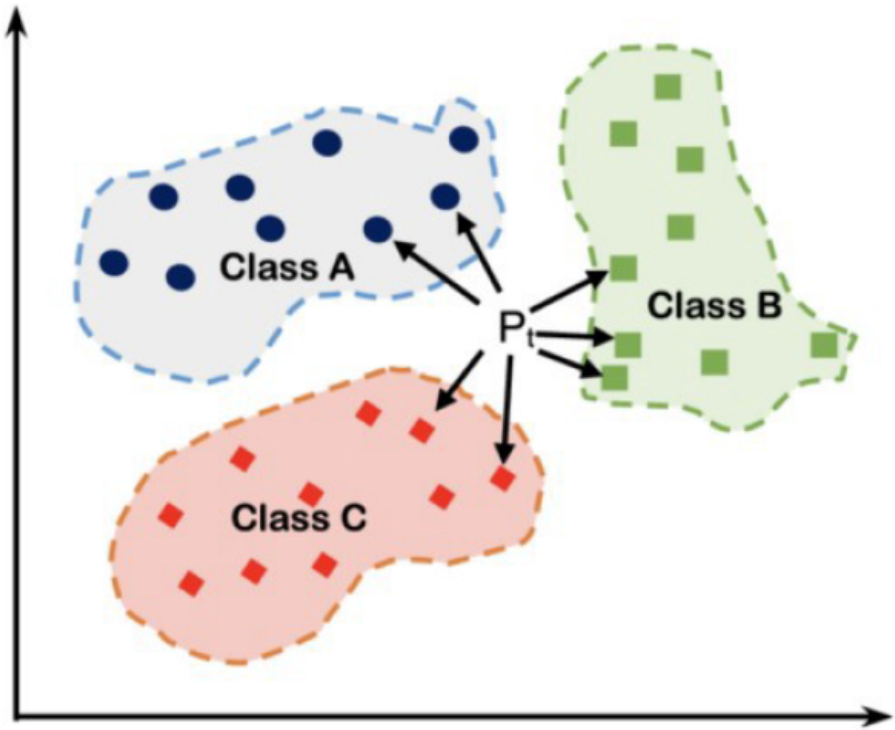
- We decided to use the "All-In" model as its beta coefficients match our CLD, and the differences in error were not significant.

		95% BCI		
Coefficients		**Mean**	**Lo**	**Hi**
	β_0	-4.61	-7.43	-2.58
fixed_acidity	β_1	-0.01	-0.01	-0.00
volatile_acidity	β_2	0.17	0.15	0.19
citric_acid	β_3	0.01	-0.01	0.02
density	β_4	8.55	5.46	12.83
alcohol	β_5	-0.02	-0.02	-0.01
Red_Wine	β_6	-2.33	-3.74	-1.31
White_Wine	β_7	-2.28	-3.69	-1.27
residual_sugar	β_8	-0.01	-0.01	-0.00
chlorides	β_9	0.09	0.01	0.19
free_sulfur_dioxide	β_{10}	-0.00	-0.00	-0.00
total_sulfur_dioxide	β_{11}	0.00	0.00	0.00
pH	β_{12}	-0.03	-0.05	-0.02
sulphates	β_{13}	-0.07	-0.08	-0.05

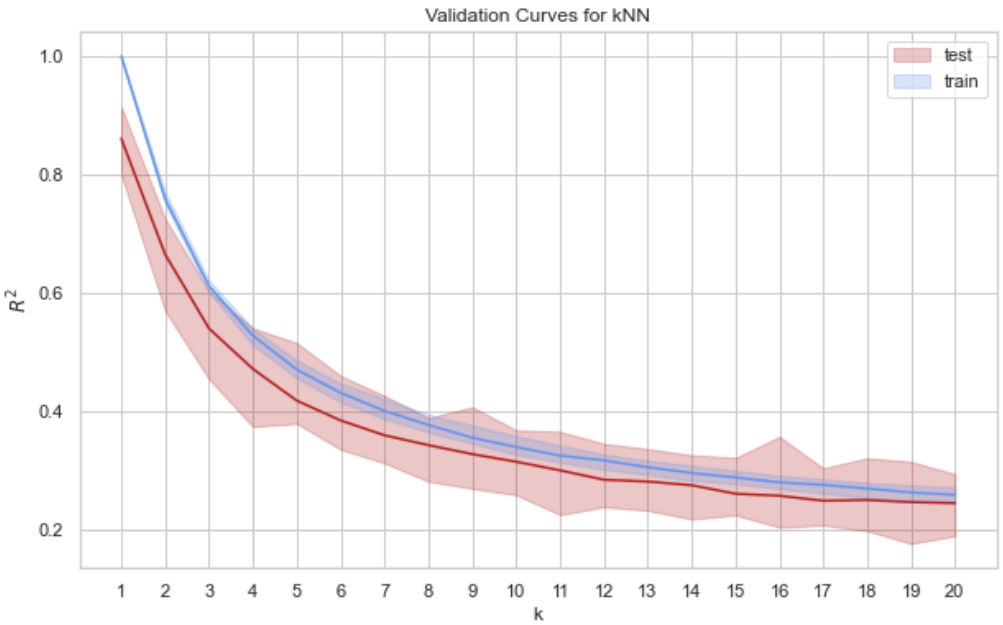
Metrics	**Mean**	**Lo**	**Hi**
σ	0.07	0.07	0.08
R^2	0.26	0.24	0.29

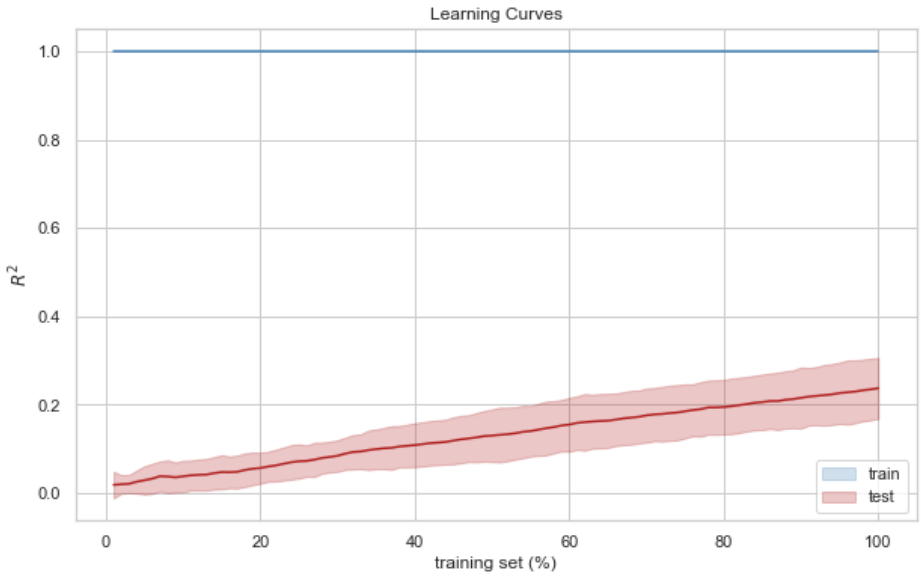
What is kNN?

- k-Nearest Neighbors is a non-parametric method which can be used for both regression and classification, making it a very useful Machine Learning Algorithm.
- The kNN algorithm utilizes "feature similarity" where a new point is assigned a value based on how closely it resembles points in the training set to predict values for new data points.
- kNN regression works by approximating the association between regressors and the continuous outcome by averaging the observations within the same neighborhood.



Choosing the number of neighbors





k = 1 is the best value for k in this model.

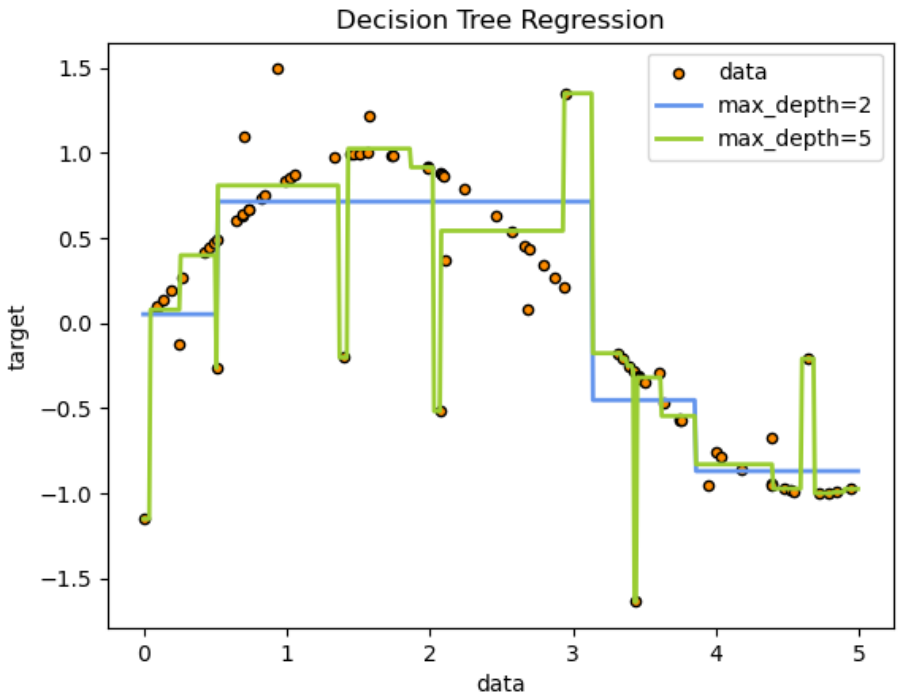
It looks like we are overfitting, and have not converged.
Looks like we have high variance.

kNN results

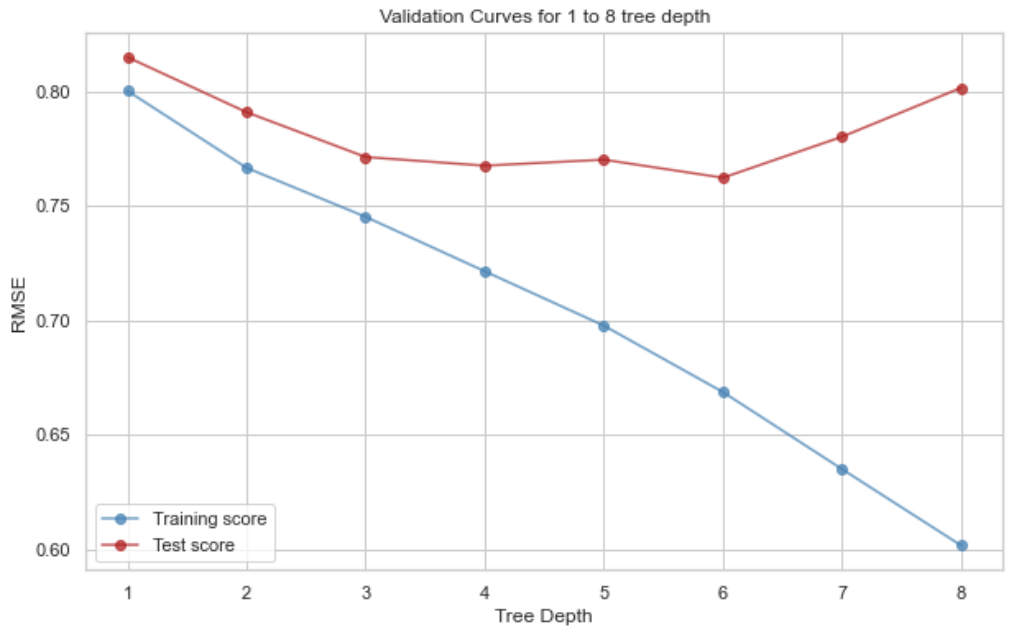
- Now we will perform 3 x 10 fold cross validation to help us understand the generalization error of our final model by looking at the credible interval from the posterior distribution of R^2 and RMSE.
- The kNN regression RMSE is 0.32, with a 95% Credible Interval from [0.25, 0.41].
  This is over a half the error from our linear regression model.
- The kNN regression R^2 is 0.86, with a 95% Credible Interval from [0.79, 0.91].
  This is over twice the R^2 from our linear regression model.

What is a Decision Tree?

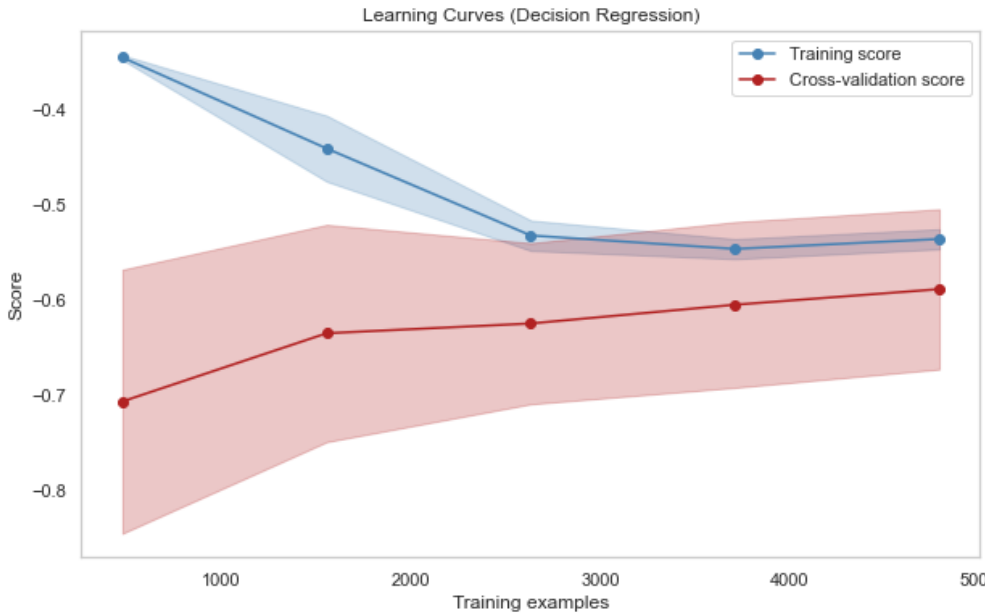
- Decision Trees break down a dataset into increasingly smaller and smaller subsets, while the associated decision tree is developed incrementally.
- A tree is the final result with decision and leaf nodes.
- Decision nodes have two or more branches, representing values for the attribute tested.
- Leaf nodes represents a decision on the numerical target.
- Decision trees can handle both categorical and numerical data.



Decision Tree modeling



A tree depth of 4 looks to be the best for bias/variance trade off, as we increase tree depth we tend to overfit the model.



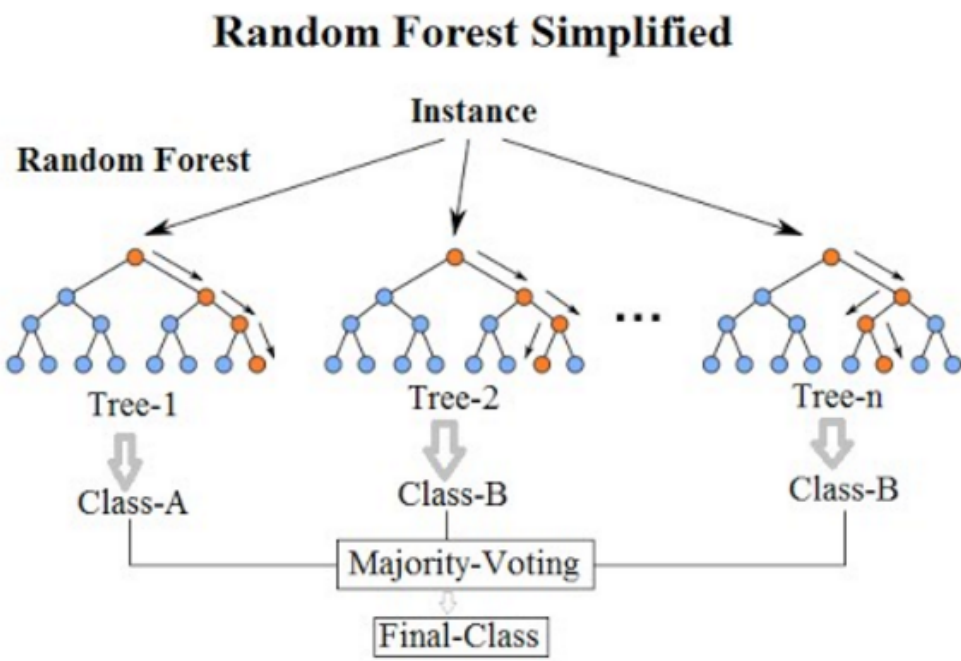
This model suffers from high bias.

Decision Tree results

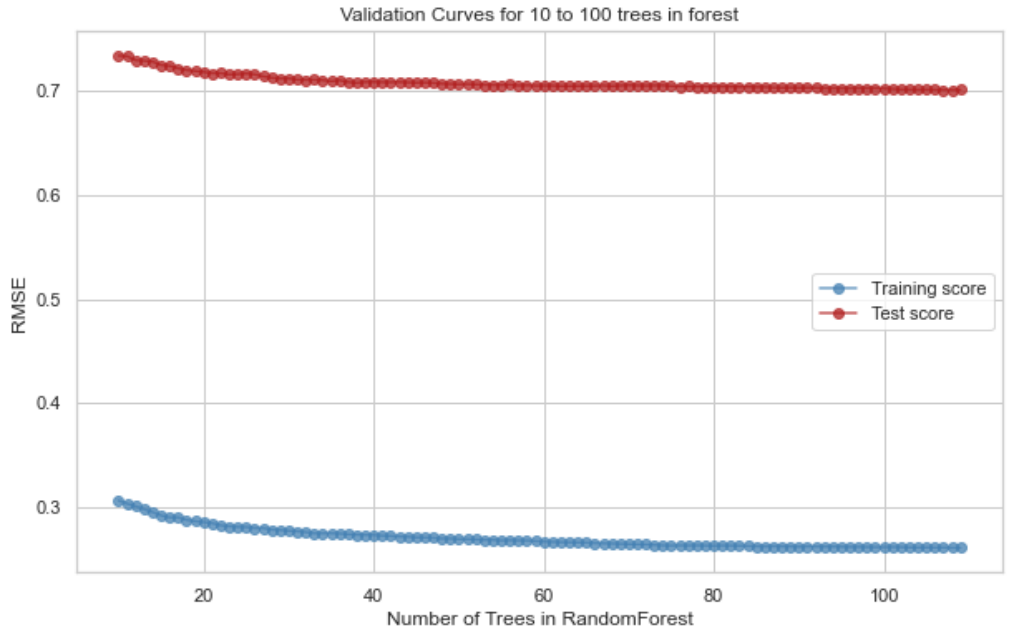
- Now we will perform 3 x 10 fold cross validation to help us understand the generalization error of our final model by looking at the credible interval from the posterior distribution of R^2.
- The Decision Tree regression RMSE is 0.97.
  The error is higher than with our null model.
- The Decision Tree regression R^2 is 0.26, with a 95% Credible Interval from [0.20, 0.30].
  This is similar to our simplest linear regression model.

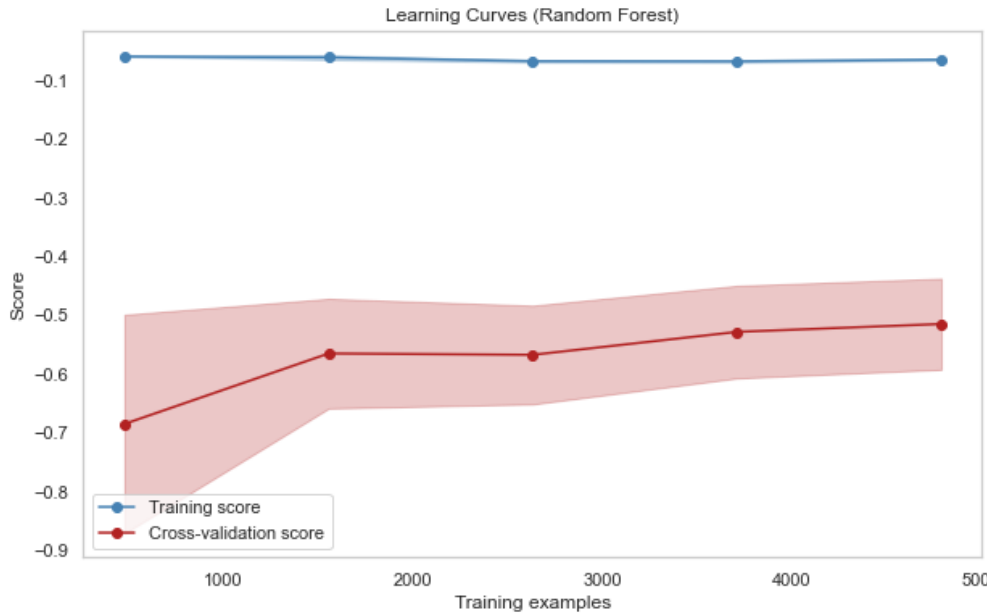
What is Random Forest?

- Random Forests construct a multitude of decision trees at training time.
- For classification the output is the class selected by the most trees.
- For regression the average prediction of the individual trees is returned.
- Random Forests correct for the decision trees' habit of overfitting to their training set.
- Overfitting is when we have low bias but high variance trade off.



Random Forest modeling





We will choose 15 trees for our random forest.

This model suffers from high variance.

Random Forest results

- Now we will perform 3 x 10 fold cross validation to help us understand the generalization error of our final model by looking at the credible interval from the posterior distribution of R^2.
- The Random Forest regression RMSE is 0.69.
  This error is similar to our linear regression model, not much of an improvement.
- The Random Forest regression R^2 is 0.39, with a 95% Credible Interval from [0.32, 0.45].
  This is alittle higher than our "all-in" linear regression model.

Model Evaluation

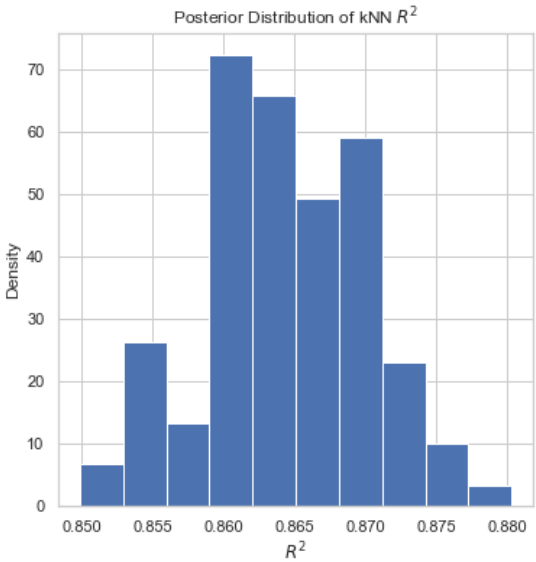
The All-In model was used in all models, except for the null model.

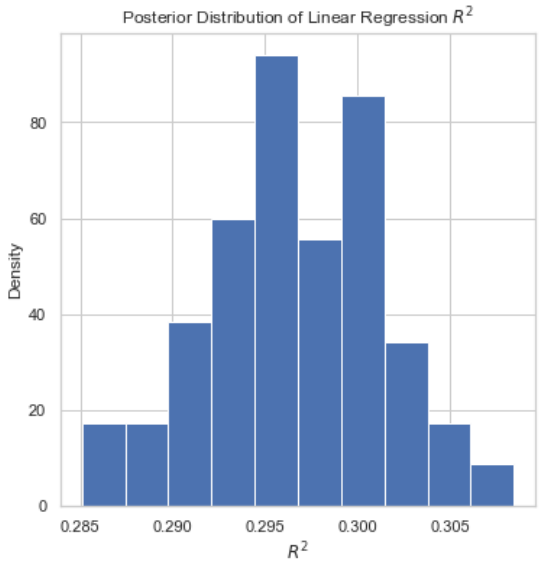
Table: Comparing Error and R^2

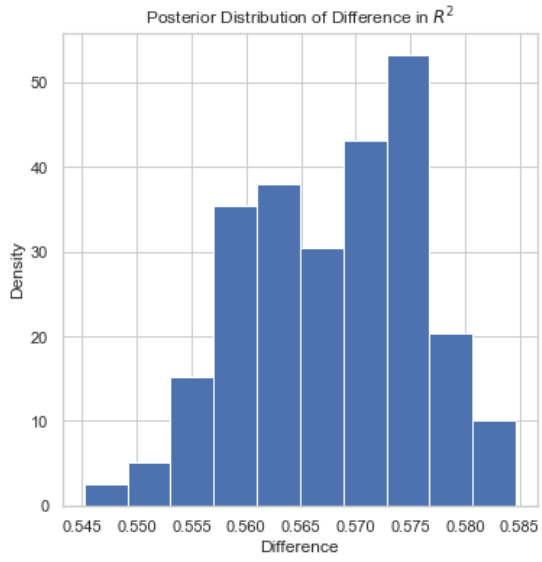
Model	**MSE**	**R^2**
Null	0.87	
LR	0.74	0.30
kNN	0.33	0.86
DT	0.97	0.25
RF	0.69	0.39

Model Evaluation

It looks like Linear Regression did worse than k-Nearest Neighbors.







Let us examine the probabilities that the models mean R^2 are better than the others.

```
print("P(kNN >= LR)", np.mean(difference_knn >= 0))
print("P(LR > kNN)", np.mean(difference_knn < 0))
```

```
P(kNN >= LR) 1.0
P(LR > kNN) 0.0
```

There is a 100% probability, based on the given evidence, that the k-Nearest Neighbors is a better model than the Linear Regression model (with these features).

```
print("P(DT >= LR)", np.mean(difference_dt >= 0))
print("P(LR > DT)", np.mean(difference_dt < 0))
```

```
P(DT >= LR) 0.0
P(LR > DT) 1.0
```

There is a 100% probability, based on the given evidence, that the Linear Regression is a better model than the Decision Tree model (with these features).

```
print("P(RF >= LR)", np.mean(difference_rf >= 0))
print("P(LR > RF)", np.mean(difference_rf < 0))
```

```
P(RF >= LR) 1.0
P(LR > RF) 0.0
```

There is a 100% probability, based on the given evidence, that the Random Forest is a better model than the Linear Regression model (with these features).

```
print("P(kNN >= RF)", np.mean(difference_knn_rf >= 0
print("P(RF > kNN)", np.mean(difference_knn_rf < 0))
```

```
P(kNN >= RF) 1.0
P(RF > kNN) 0.0
```

There is a 100% probability, base on the given evidence, that the k-Nearest Neighbors is a better model than the Random Forest model (with these features).

Final Models

- In the end our best fit was the "All-In" model:
- All-in Model: "quality ~ Red_Wine + White_Wine + fixed_acidity + volatile_acidity + citric_acid + residual_sugar + chlorides + free_sulfur_dioxide + total_sulfur_dioxide + density + pH + sulphates + alcohol"
- Using k-Nearest Neighbors Regression.

Predictions

We shall use k-Nearest Neighbors to predict wine quality from physicochemical properties.

- A red wine with moderate alcohol content, low residual sugar.
  prediction1 = [1, 0, 8.2, 0.25, 0.1, 2, 0.05, 20, 50, 1.0, 3.4, 0.7, 11]
  Predicted quality: 7
- A white wine with alot of free sulfur dioxide, high alcohol content, and very sweet.
  prediction2 = [0, 1, 6.5, 0.1, 0.2, 70, 0.1, 75, 200, 0.99, 3.1, 0.3, 14]
  Predicted Quality: 5
- A white wine with low alcohol, dry, citric acid, low pH, salty, high volatile acidity, acidic. This wine has faults.
  prediction3 = [0, 1, 10, 1.1, 0.6, 1.5, 0.15, 35, 126, 1.01, 3.0, 1.1, 7]
  Predicted Quality: 5

The third prediction was a surprise, as we expected it to have a lower quality.

Limitations

- All of this data is collected from wineries in the Minho region of Northern Portugal. So we would expect it to perform well when used to predict the quality of wines from that region.
- The ability to generalize this model to wines from other regions has not been established, and should be done carefully.

Conclusions

- Linear Regression performed better than the null model and Decision Tree Regression (in this situation).
- kNN regression was the had the best performance by far.
- The simplest and complex models are not always the best.
- Sometimes just the "normal" model can yield the best results.

References

- P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009. Available at: <http://www3.dsi.uminho.pt/pcortez/wine/>
- Charters, S. and Pettigrew, S., 2007. The dimensions of wine quality. Food Quality and Preference, [online] 18(7), pp.997-1007. Available at: <https://doi.org/10.1016/j.foodqual.2007.04.003>

End Card

- Joshua Paul Barnard
- May, 2022
- Over 2 years of experience in the Wine Industry.
- The Presentation was made in LaTeX, with the code in Python.