Modeling Wine Quality Using Classification and Regression

Mario Wijaya

MGT 8803

November 28, 2017

Motivation

Quality

- How to assess it?
- What makes a good quality wine?

Good or Bad Wine?

- Subjective?
- Wine taster

Who cares?

- Consumer
- Wine industry

Data Science

- Classification
- Regression

Goal

- Predict quality of a given wine
- Classify whether a wine is good or bad

Dataset

Consists of

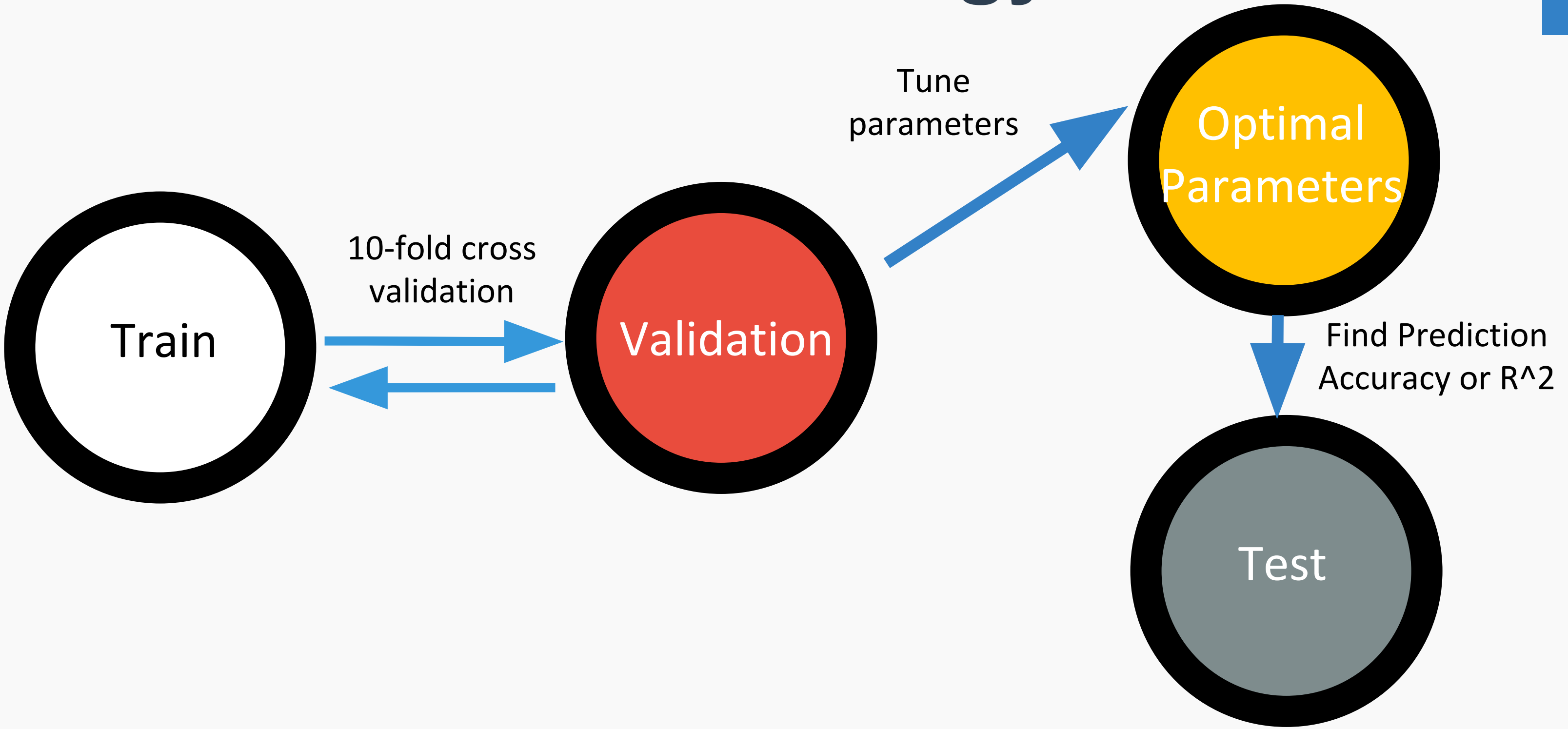
- White wine: 4898 samples
- Red wine: 1599 samples
- Variables:
  - Fixed acidity
  - Volatile acidity
  - Quality
  - GoodBad
    - Quality > 5: Class 1
    - Quality <=5: Class 0
  - etc
- Potential problem?
  - Class imbalance
  - Bias
  - High variance

Solution

- Oversampling underrepresented class
- Downsampling overrepresented class
- Overweight underrepresented classes in loss function
- Normalization for classification and regression (SGD)

Source of Dataset: UCI (https://archive.ics.uci.edu/ml/datasets/Wine+Quality)

General Strategy



Tools: Python3 with Scikit-learn package, Matplotlib & Seaborn (Plot & Visualization)

Models & Challenges

Regression

- Multi linear regression
- Stochastic Gradient Descent
- Ridge Regression
- Lasso Regression
- Decision Tree Regression

Classification

- SVM
- K-Nearest Neighbor
- Decision Tree Classification
- Used PCA to do dimension reduction
  - 11 variables mapped to 2 dimension

Challenges

- Find optimal parameters
  - SVM: C, gamma
  - Etc
- Find model that can be generalized
- Prevent overfitting
  - K-fold cross validation

Quick Lecture

Stochastic Gradient Descent

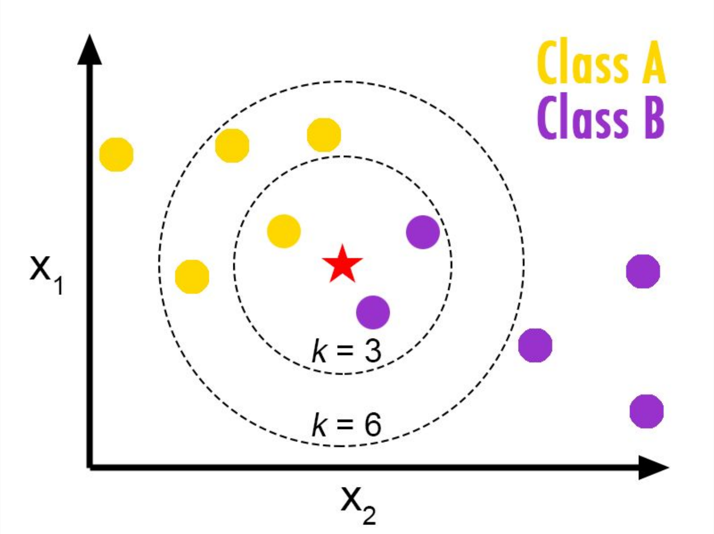
The standard gradient descent algorithm updates the parameters θ of the objective $J(\theta)$ as,

$$\theta = \theta - \alpha \nabla_\theta E[J(\theta)]$$

where the expectation in the above equation is approximated by evaluating the cost and gradient over the full training set. Stochastic Gradient Descent (SGD) simply does away with the expectation in the update and computes the gradient of the parameters using only a single or a few training examples. The new update is given by,

$$\theta = \theta - \alpha \nabla_\theta J(\theta; x^{(i)}, y^{(i)})$$

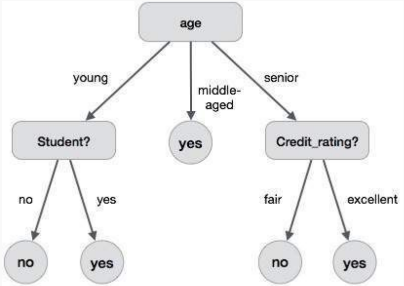
Classification - KNN



Regression

- Ridge Regression
    - L-2 penalty
- Lasso
    - L-1 Penalty
- Decision Tree

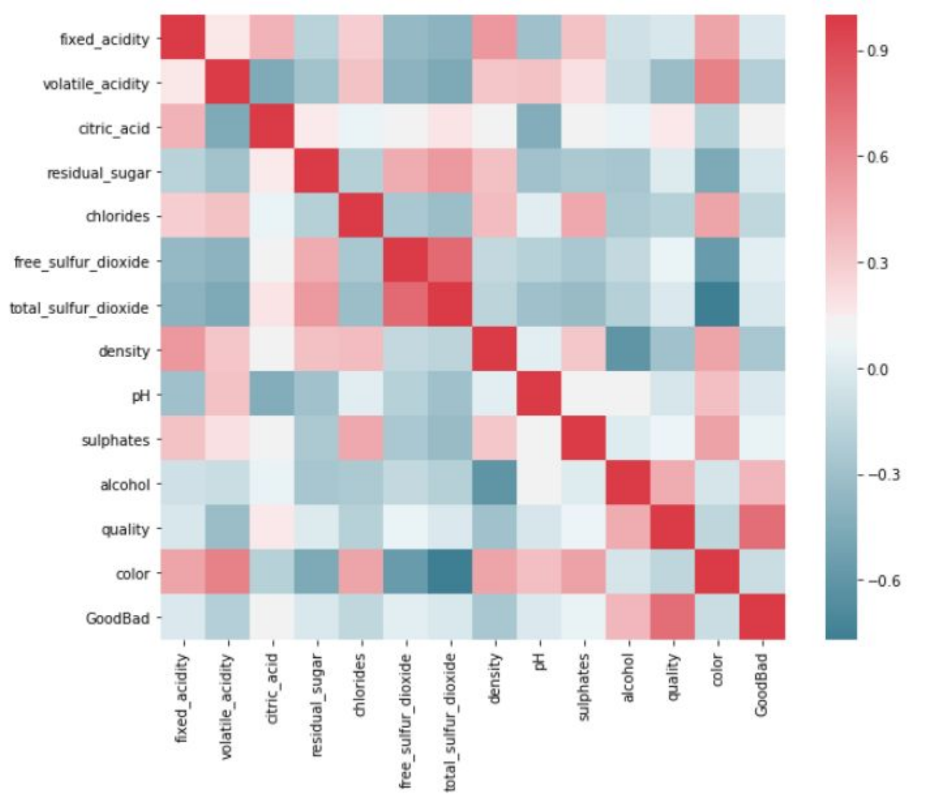
$$\|x\|_p = \left(\sum_{i \in N} |x_i|^p \right)^{1/p}$$



Regression

Regression

- Correlation Matrix
  - Look at possible high correlation feature



Regression

Regression

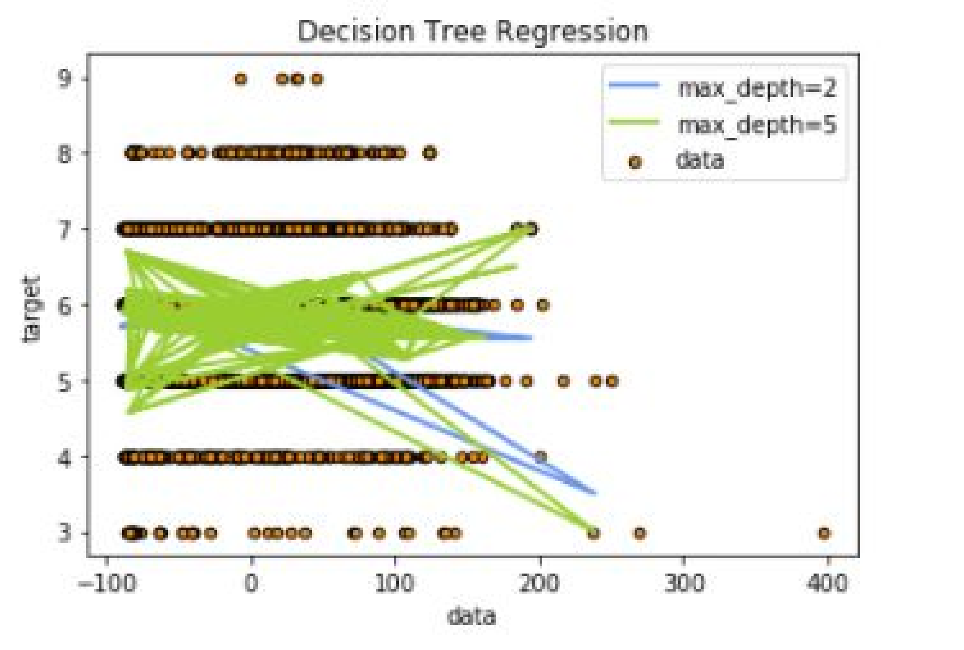
- Correlation Matrix
  - Look at possible high correlation feature
- Multiple Linear Regression
  - Y = X1beta1 + X2beta2 +... XnbetaN + E
  - R^2 = 0.325
  - Pretty bad!
- SGD - R^2: 0.323
- Lasso and Ridge equally bad
- Used interaction terms and remove high p-value -> bad
- Forward selection -> not good either

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                quality   R-squared:                       0.325
Model:                            OLS   Adj. R-squared:                  0.324
Method:                 Least Squares   F-statistic:                     274.0
Date:                Thu, 23 Nov 2017   Prob (F-statistic):               0.00
Time:                        14:08:44   Log-Likelihood:                -6677.9
No. Observations:                6268   AIC:                         1.338e+04
Df Residuals:                    6256   BIC:                         1.346e+04
Df Model:                          11
Covariance Type:            nonrobust
========================================================================================
                           coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------
Intercept               49.9334     10.625      4.700      0.000      29.105      70.762
fixed_acidity            0.0566      0.014      4.014      0.000       0.029       0.084
volatile_acidity        -1.2555      0.068    -18.378      0.000      -1.389      -1.122
citric_acid             -0.0784      0.077     -1.021      0.307      -0.229       0.072
residual_sugar           0.0376      0.005      7.623      0.000       0.028       0.047
chlorides               -1.1647      0.283     -4.118      0.000      -1.719      -0.610
free_sulfur_dioxide      0.0053      0.001      6.510      0.000       0.004       0.007
total_sulfur_dioxide    -0.0027      0.000     -9.472      0.000      -0.003      -0.002
density                -48.4174     10.867     -4.456      0.000     -69.720     -27.115
pH                       0.2880      0.091      3.151      0.002       0.109       0.467
sulphates                0.8255      0.068     12.181      0.000       0.693       0.958
alcohol                  0.2607      0.015     17.448      0.000       0.231       0.290
==============================================================================
Omnibus:                      155.360   Durbin-Watson:                   2.002
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              356.307
Skew:                          -0.071   Prob(JB):                     4.25e-78
Kurtosis:                       4.159   Cond. No.                     1.96e+05
==============================================================================
```

Regression

Regression

- Correlation Matrix
  - Look at possible high correlation feature
- Multiple Linear Regression
  - Y = X1beta1 + X2beta2 +... XnbetaN + E
  - $R^2 = 0.325$
  - Pretty bad!
- SGD - R^2: 0.323
- Lasso and Ridge equally bad
- Used interaction terms and remove high p-value -> bad
- Forward selection -> not good either

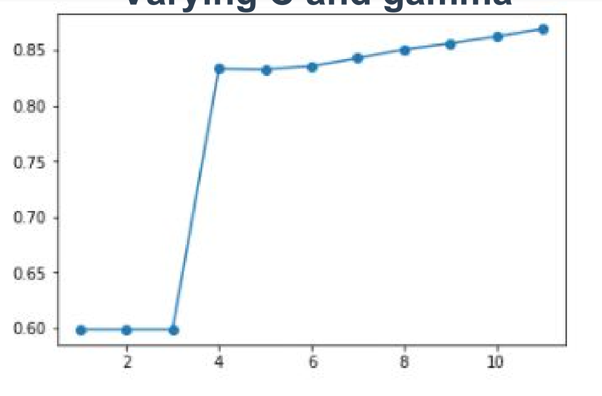


Classification - SVM

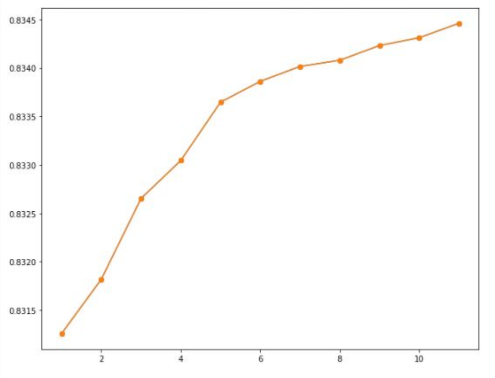
Classification

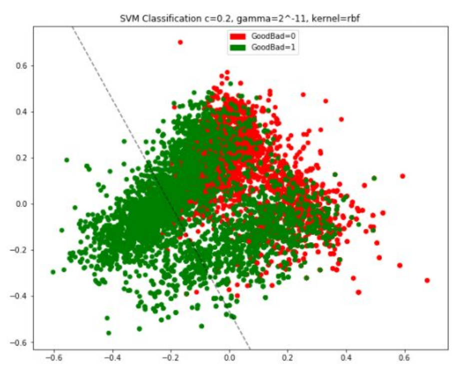
- Normalize data (0,1)
- Varies parameter of C and gamma
  - 10-fold cross validation
  - Find best model that gives lowest error rate or highest accuracy rate
- ~83% prediction accuracy but clearly linear kernel is better in this case from support vector drawn
- How do you draw 11 dimensions into 2 dimensions?
  - PCA

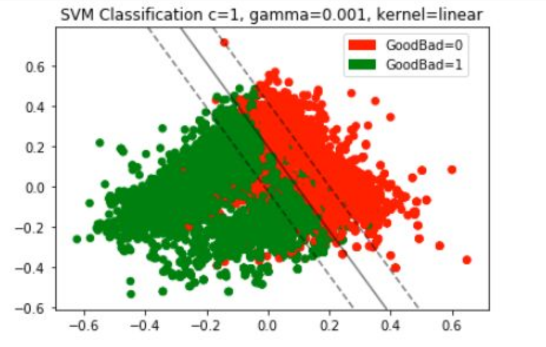
Prediction Accuracy - RBF Kernel Varying C and gamma



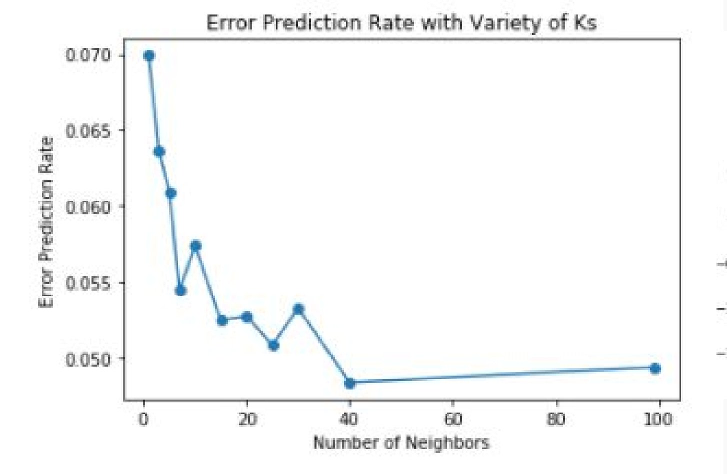
Prediction Accuracy - Linear Kernel Varying C

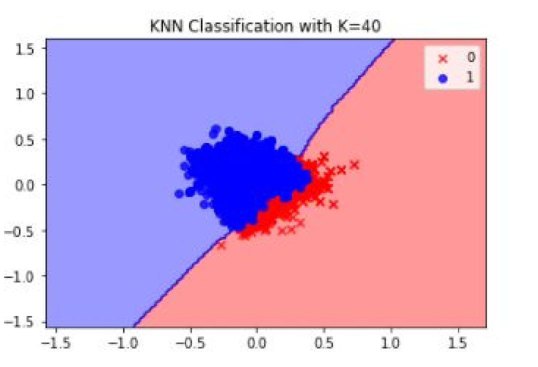






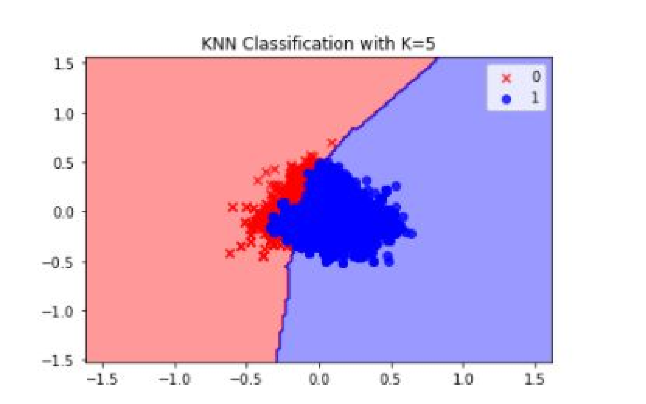
Classification - KNN

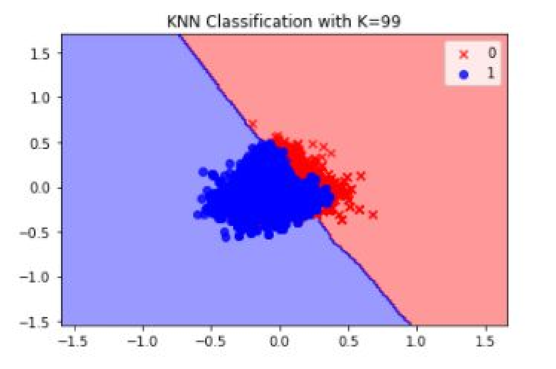




Classification

- Ad-hoc knowledge:
  - K = 1/sqrt(# of samples) = ~99
- Use 10-fold CV
  - Determine error rate
  - Use it to find best K
  - K = 40 -> K = 100
    - Not much different
- Higher K -> smoother curves
- Relatively good for classification
  - Easily overfitting
  - Careful!





Classification - Decision Tree

Classification

- Recursively find label
- Used Gini Index for splitting
    - Other methods: Information Gain (Entropy)
- 88% prediction accuracy
    - Also tried with testing data
- Need to set depth, otherwise we will have overfitting

```
digraph Tree {
node [shape=box, style="filled, rounded", color="black", fontname=helvetica] ;
edge [fontname=helvetica] ;
0 [label=<alcohol &le; 10.25<br/>gini = 0.48<br/>samples = 7836<br/>value = [3129, 4707]<br/>class = o>, fillcolor="#399de555"] ;
1 [label=<volatile_acidity &le; 0.275<br/>gini = 0.484<br/>samples = 3960<br/>value = [2332, 1628]<br/>class = G>, fillcolor="#e581394d"] ;
0 -> 1 [labeldistance=2.5, labelangle=45, headlabel="True"] ;
2 [label=<volatile_acidity &le; 0.227<br/>gini = 0.453<br/>samples = 1103<br/>value = [383, 720]<br/>class = o>, fillcolor="#399de577"] ;
1 -> 2 ;
3 [label=<density &le; 0.992<br/>gini = 0.376<br/>samples = 558<br/>value = [140, 418]<br/>class = o>, fillcolor="#399de5aa"] ;
2 -> 3 ;
4 [label=<chlorides &le; 0.035<br/>gini = 0.426<br/>samples = 13<br/>value = [9, 4]<br/>class = G>, fillcolor="#e581398e"] ;
3 -> 4 ;
5 [label=<gini = 0.0<br/>samples = 2<br/>value = [0, 2]<br/>class = o>, fillcolor="#399de5ff"] ;
4 -> 5 ;
6 [label=<free_sulfur_dioxide &le; 18.0<br/>gini = 0.298<br/>samples = 11<br/>value = [9, 2]<br/>class = G>, fillcolor="#e58139c6"] ;
4 -> 6 ;
7 [label=<total_sulfur_dioxide &le; 68.0<br/>gini = 0.444<br/>samples = 3<br/>value = [1, 2]<br/>class = o>, fillcolor="#399de57f"] ;
6 -> 7 ;
8 [label=<gini = 0.0<br/>samples = 1<br/>value = [1, 0]<br/>class = G>, fillcolor="#e58139ff"] ;
7 -> 8 ;
9 [label=<gini = 0.0<br/>samples = 2<br/>value = [0, 2]<br/>class = o>, fillcolor="#399de5ff"] ;
7 -> 9 ;
10 [label=<gini = 0.0<br/>samples = 8<br/>value = [8, 0]<br/>class = G>, fillcolor="#e58139ff"] ;
6 -> 10 ;
11 [label=<density &le; 0.997<br/>gini = 0.365<br/>samples = 545<br/>value = [131, 414]<br/>class = o>, fillcolor="#399de5ae"] ;
3 -> 11 ;
12 [label=<free_sulfur_dioxide &le; 19.5<br/>gini = 0.412<br/>samples = 321<br/>value = [93, 228]<br/>class = o>, fillcolor="#399de597"] ;
11 -> 12 ;
13 [label=<fixed_acidity &le; 6.95<br/>gini = 0.497<br/>samples = 56<br/>value = [30, 26]<br/>class = G>, fillcolor="#e5813922"] ;
12 -> 13 ;
14 [label=<citric_acid &le; 0.24<br/>gini = 0.444<br/>samples = 27<br/>value = [9, 18]<br/>class = o>, fillcolor="#399de57f"] ;
13 -> 14 ;
15 [label=<gini = 0.0<br/>samples = 3<br/>value = [3, 0]<br/>class = G>, fillcolor="#e58139ff"] ;
14 -> 15 ;
16 [label=<free_sulfur_dioxide &le; 17.5<br/>gini = 0.375<br/>samples = 24<br/>value = [6, 18]<br/>class = o>, fillcolor="#399de5aa"] ;
14 -> 16 ;
17 [label=<free_sulfur_dioxide &le; 12.0<br/>gini = 0.266<br/>samples = 19<br/>value = [3, 16]<br/>class = o>, fillcolor="#399de5cf"] ;
16 -> 17 ;
18 [label=<fixed_acidity &le; 5.6<br/>gini = 0.5<br/>samples = 6<br/>value = [3, 3]<br/>class = G>, fillcolor="#e5813900"] ;
17 -> 18 ;
19 [label=<gini = 0.0<br/>samples = 1<br/>value = [1, 0]<br/>class = G>, fillcolor="#e58139ff"] ;
18 -> 19 ;
20 [label=<volatile_acidity &le; 0.195<br/>gini = 0.48<br/>samples = 5<br/>value = [2, 3]<br/>class = o>, fillcolor="#399de555"] ;
18 -> 20 ;
21 [label=<gini = 0.0<br/>samples = 2<br/>value = [0, 2]<br/>class = o>, fillcolor="#399de5ff"] ;
20 -> 21 ;
22 [label=<fixed_acidity &le; 6.85<br/>gini = 0.444<br/>samples = 3<br/>value = [2, 1]<br/>class = G>, fillcolor="#e581397f"] ;
20 -> 22 ;
23 [label=<gini = 0.0<br/>samples = 2<br/>value = [2, 0]<br/>class = G>, fillcolor="#e58139ff"] ;
22 -> 23 ;
```

Conclusion & Discussion

Conclusion

- Several clustering algorithm works well with the dataset
- Bad performance with regression
  - Possibly need more work in determining which features to keep
- Combat subjective result from wine taster when we can use Data Science to answer the question

Discussion

- If good regression model can be found then a Python based application can be build for interactivity
- Need to understand dataset well and find optimal parameters

Modeling Wine Quality

- ★ Ran several algorithm on multiple linear regression
  - ○ Ordinary Least Square (Linear Regression)
  - ○ Ridge Regression
  - ○ Lasso Regression
  - ○ Stochastic Gradient Descent
  - ○ Forward Selection
  - ○ Decision Tree Regression
- ★ Created several classification models to predict whether the quality of a given wine is good or bad
  - ○ K-Nearest Neighbors
  - ○ SVM
  - ○ Decision Tree Classification
  - ○ Used PCA for dimensionality reduction

Mario Wijaya

Georgia Tech