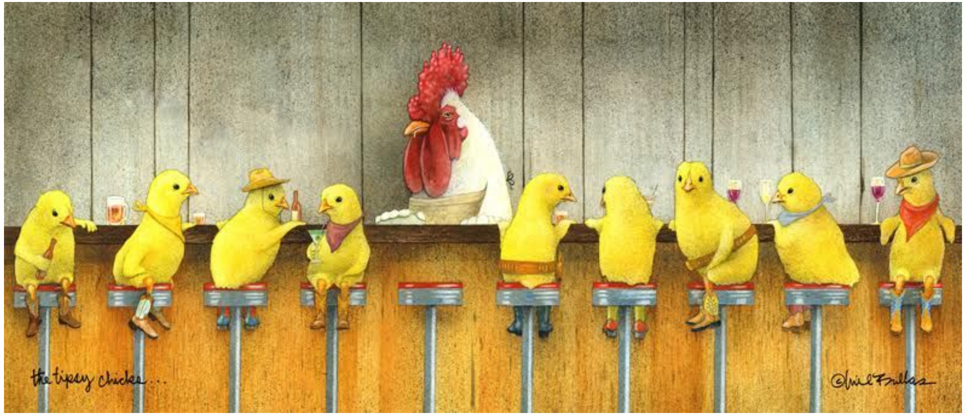
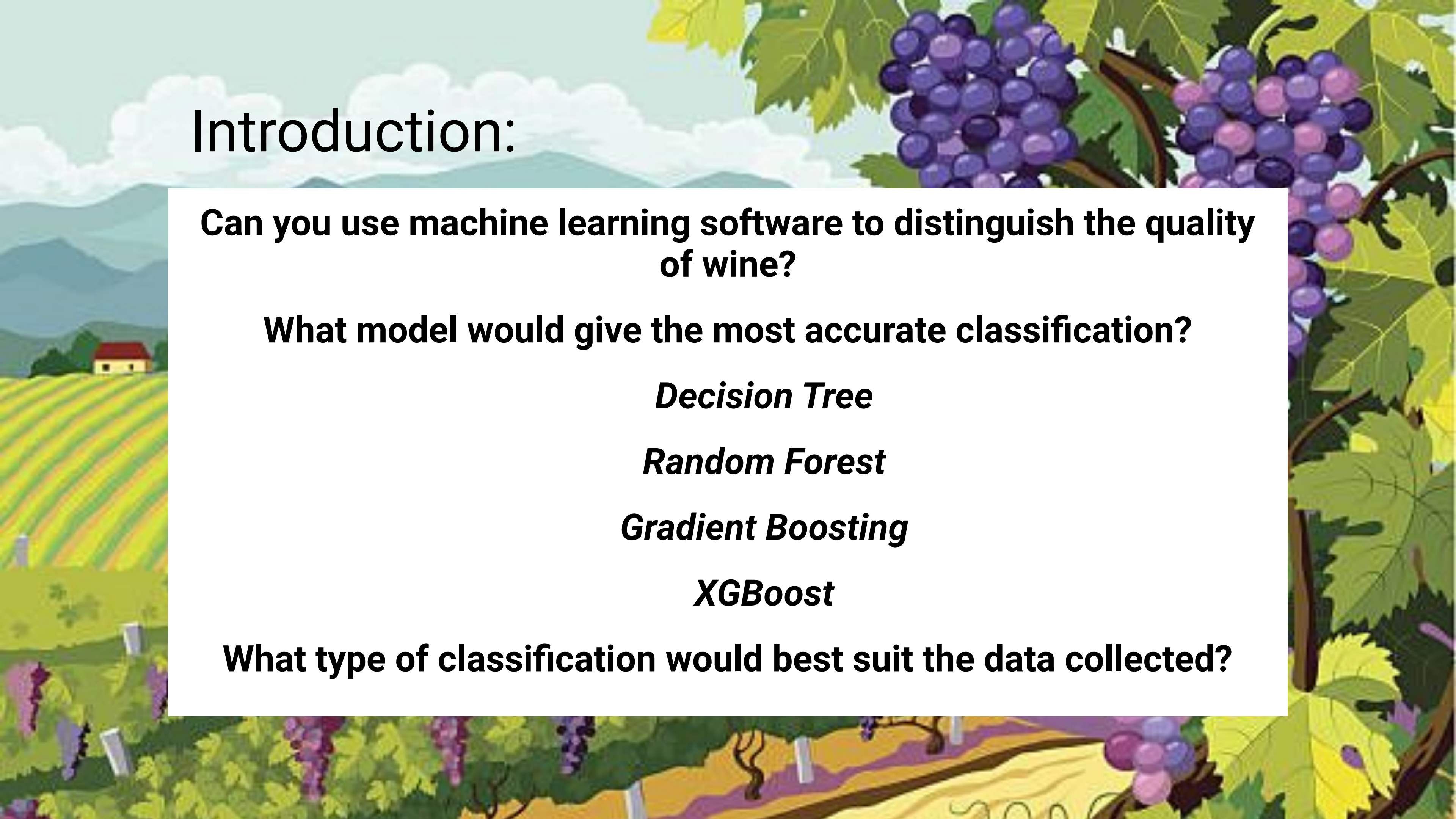
Predicting Wine Quality





Introduction:

Can you use machine learning software to distinguish the quality of wine?

What model would give the most accurate classification?

Decision Tree

Random Forest

Gradient Boosting

XGBoost

What type of classification would best suit the data collected?

Playground Prediction Competition

Ordinal Regression with a Tabular Wine Quality Dataset

Playground Series - Season 3, Episode 5

Kaggle · 901 teams · 6 months ago

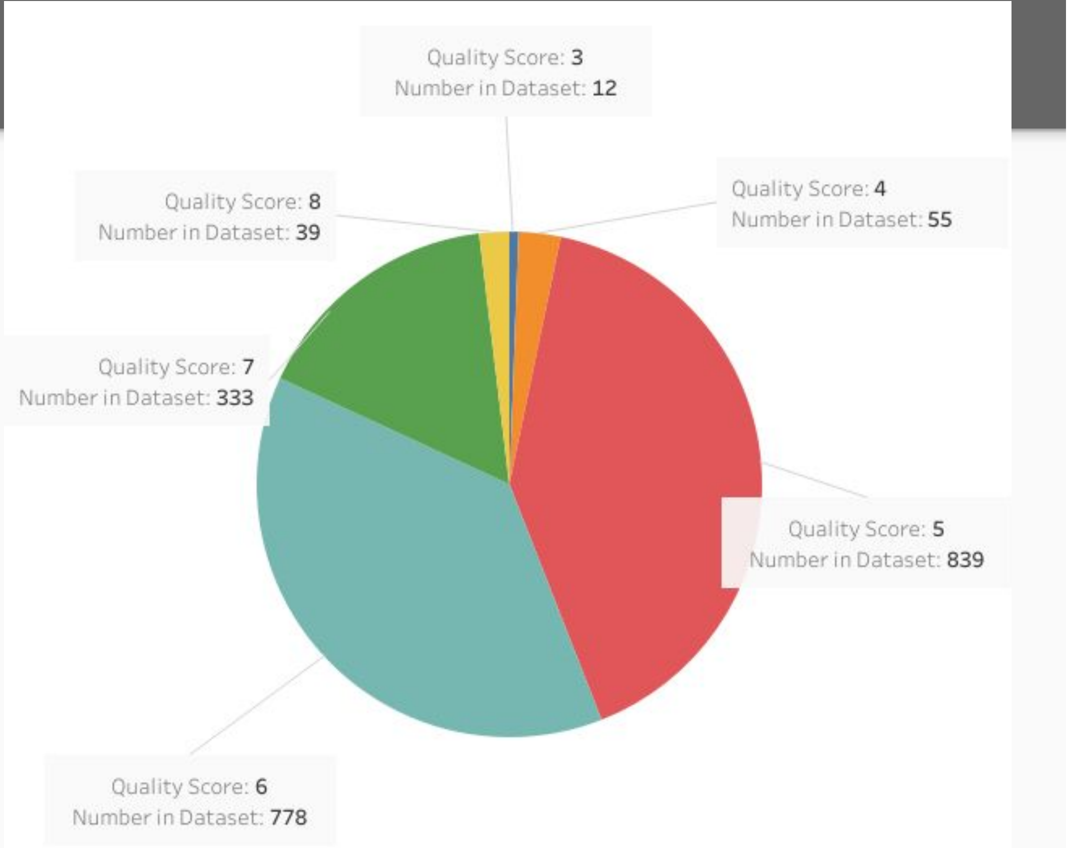
Data Collected:

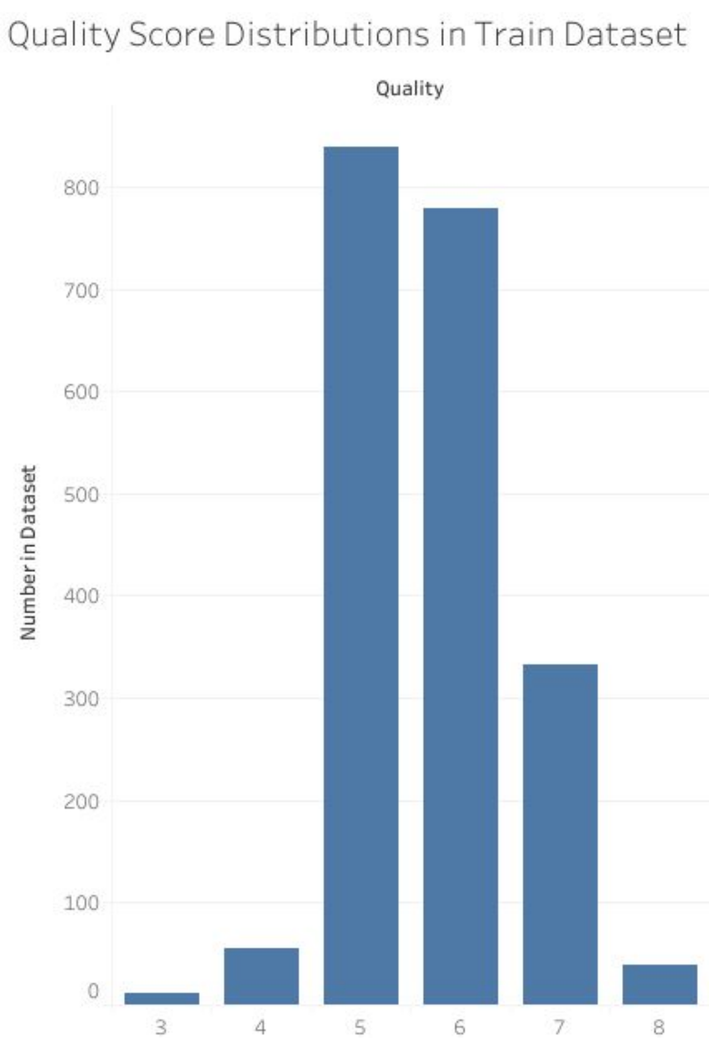
- **Features** included profiles on alcohol content, chlorides, citric acid, acidity, sulfur dioxide, residual sugar, density, and pH for each wine
- **Target** = quality score (discrete value between 1 and 10)

- Train and Test dataset supplied by Kaggle
- 2056 wines included in training data

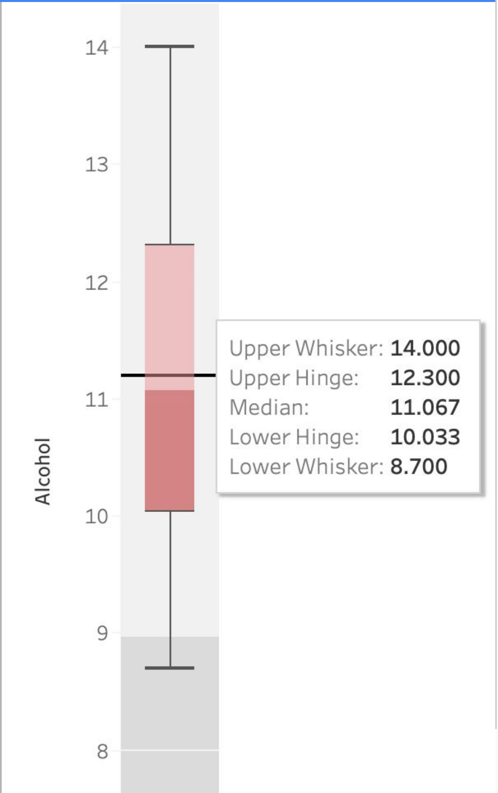
- We discovered our dataset was generated from a larger dataset that separately described red and white wines. The Kaggle competition dataset we used does not differentiate between reds and whites for the quality scoring.

Test Data Quality Score Distribution

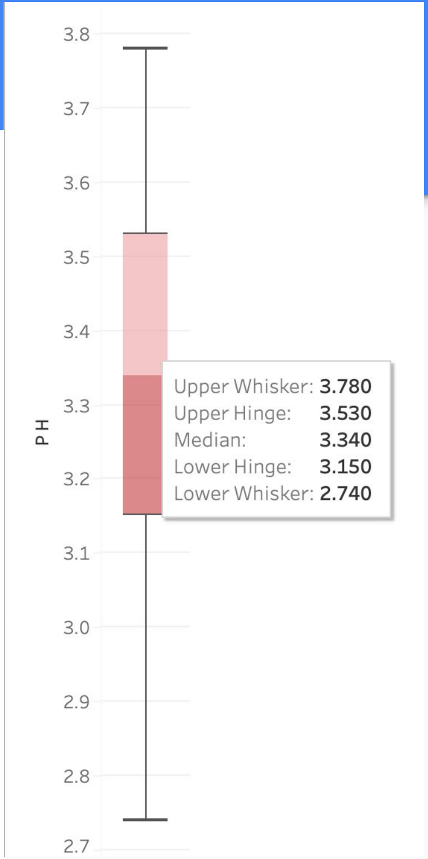




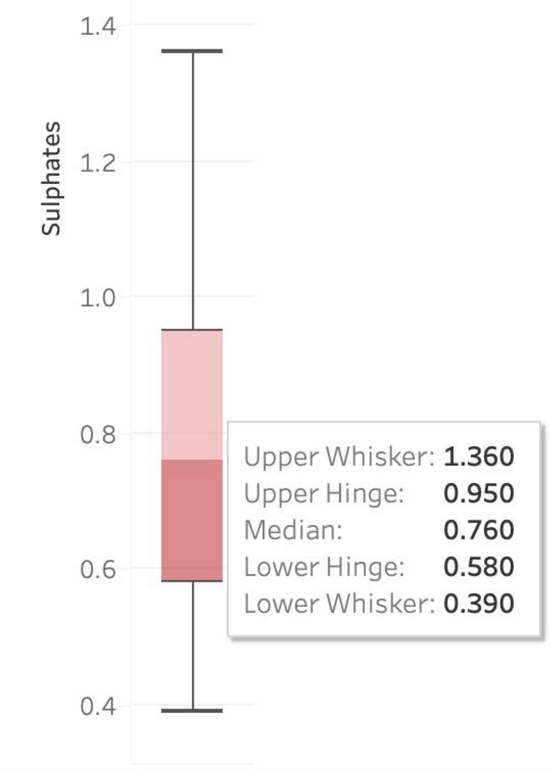
Exploring Our Data



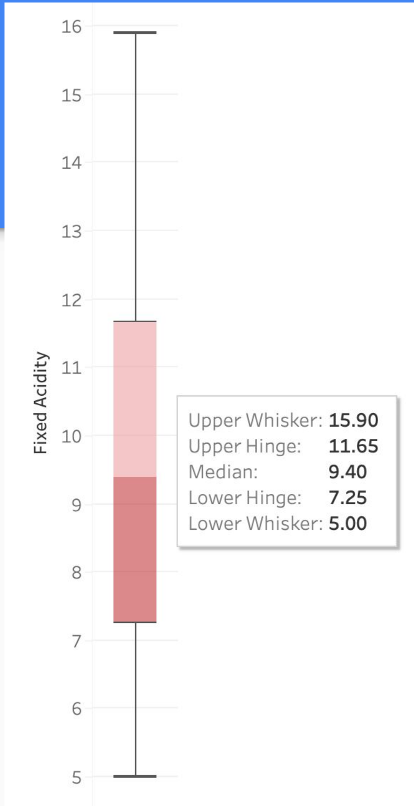
Upper Whisker:	14.000
Upper Hinge:	12.300
Median:	11.067
Lower Hinge:	10.033
Lower Whisker:	8.700



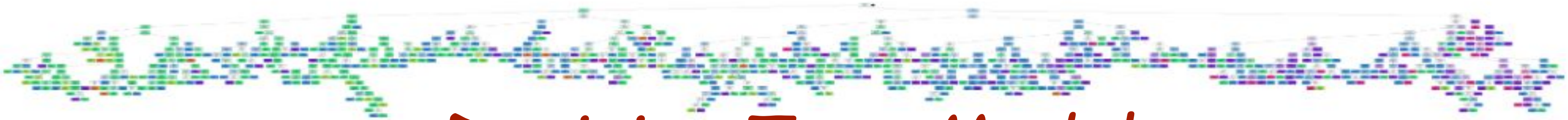
Upper Whisker:	3.780
Upper Hinge:	3.530
Median:	3.340
Lower Hinge:	3.150
Lower Whisker:	2.740



Upper Whisker:	1.360
Upper Hinge:	0.950
Median:	0.760
Lower Hinge:	0.580
Lower Whisker:	0.390



Upper Whisker:	15.90
Upper Hinge:	11.65
Median:	9.40
Lower Hinge:	7.25
Lower Whisker:	5.00

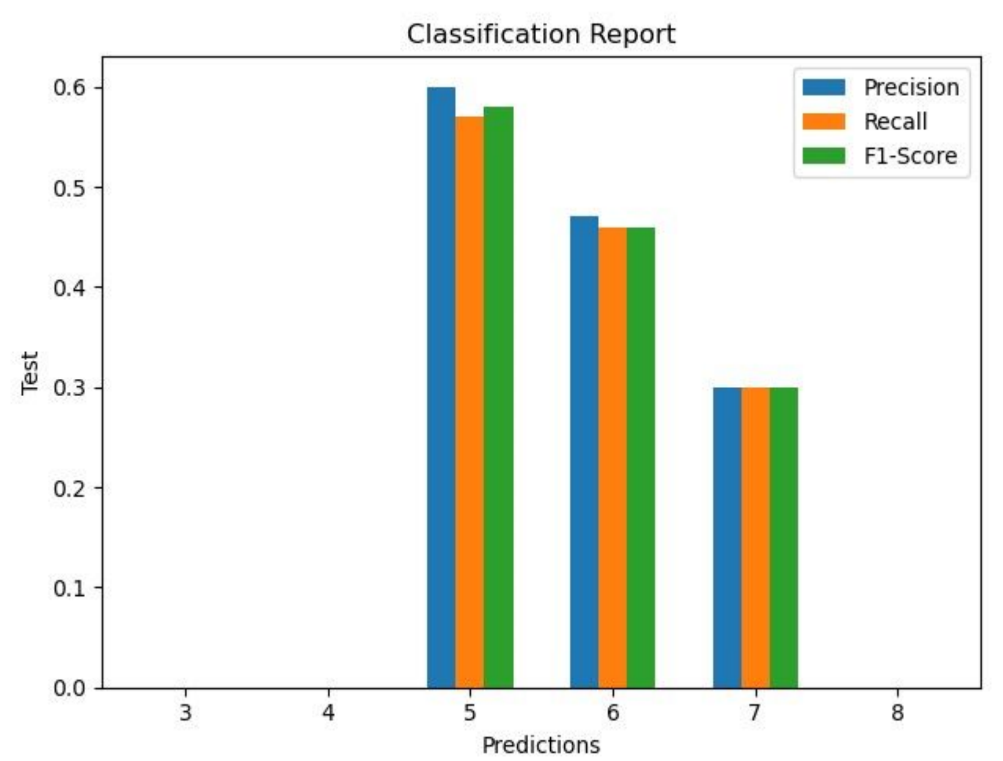


Decision Tree Model:

Kappa Score : 0.2883 :(

Classification Report:

```
              precision    recall  f1-score   support
        3       0.00       0.00      0.00        2
        4       0.00       0.00      0.00       8
        5       0.60       0.57      0.58       169
        6       0.47       0.46      0.46       158
        7       0.30       0.30      0.30       69
        8       0.00       0.00      0.00       6
accuracy                             0.46      412
macro avg    0.23       0.22      0.23          412
weighted avg  0.48    0.46      0.47       412
```



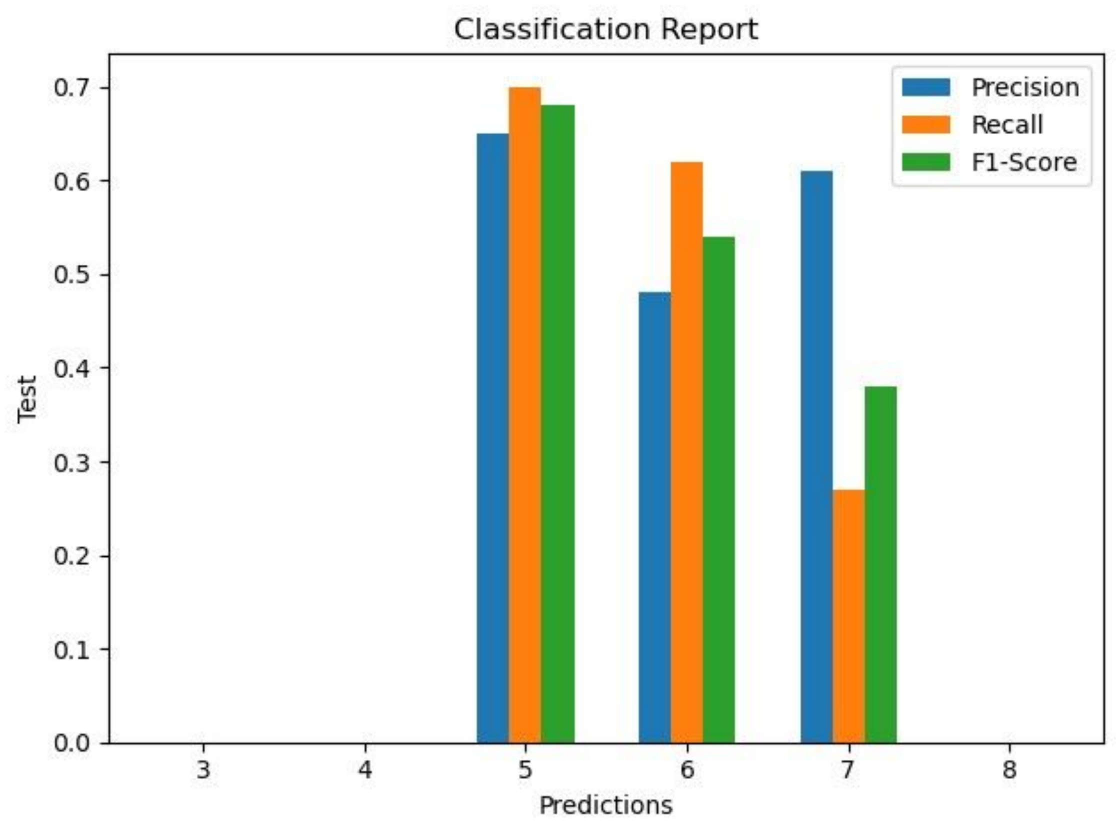
Random Forest Model

Kappa Score : 0.4428
Classification Report:

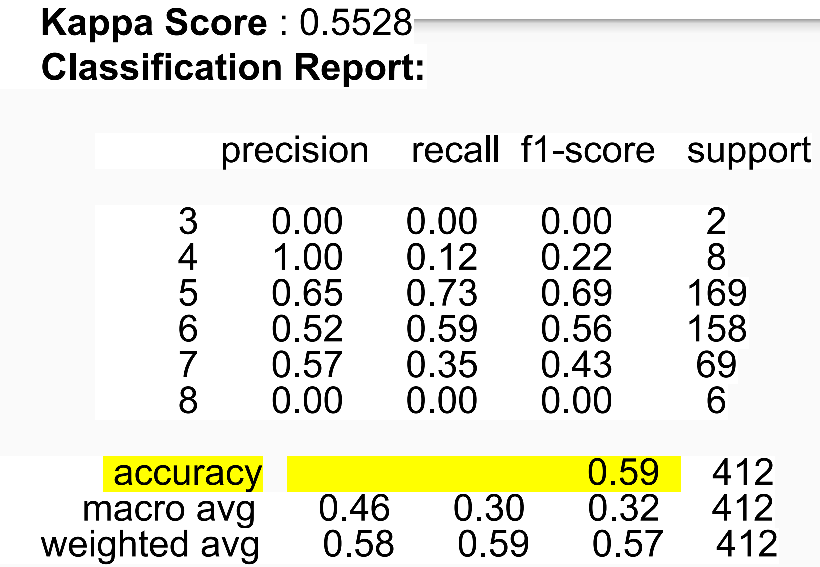
```
          precision    recall  f1-score   support

       3       0.00      0.00      0.00         2
       4       0.00      0.00      0.00        11
       5       0.65      0.70      0.68       216
       6       0.48      0.62      0.54       183
       7       0.61      0.27      0.38        91
       8       0.00      0.00      0.00        11

 accuracy                          0.57       514
 macro avg     0.29      0.27      0.27       514
weighted avg 0.55      0.57      0.54       514
```



Gradient Boosting Model:



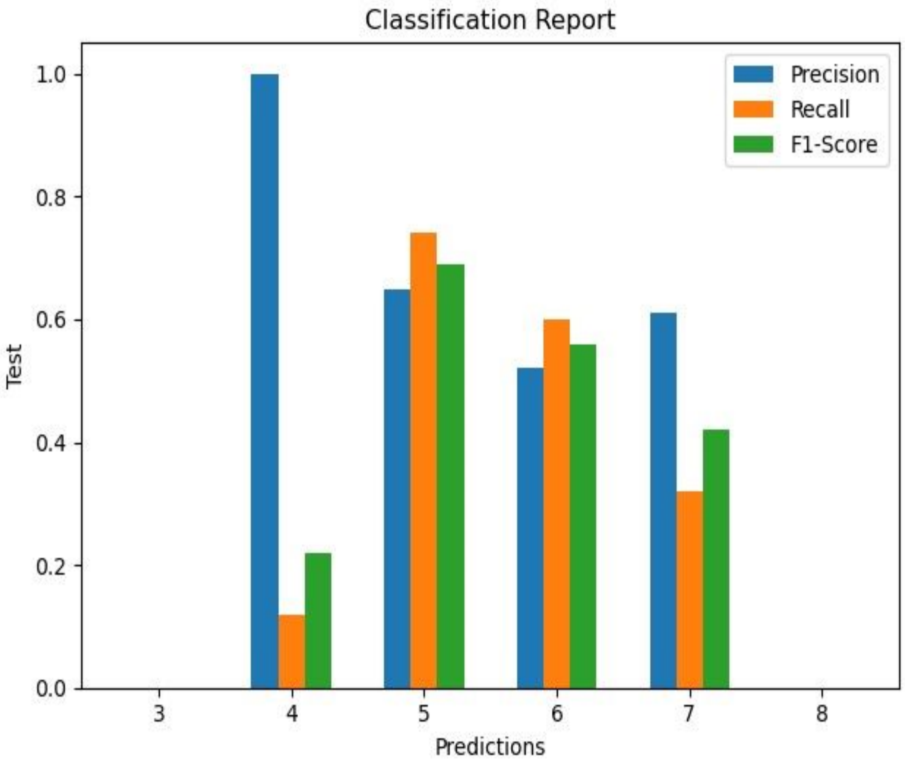
Kappa Score : 0.5528

Classification Report:

```
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         2
           4       1.00      0.12      0.22         8
           5       0.65      0.73      0.69       169
           6       0.52      0.59      0.56       158
           7       0.57      0.35      0.43        69
           8       0.00      0.00      0.00         6

    accuracy                           0.59       412
   macro avg       0.46      0.30      0.32       412
weighted avg       0.58      0.59      0.57       412
```



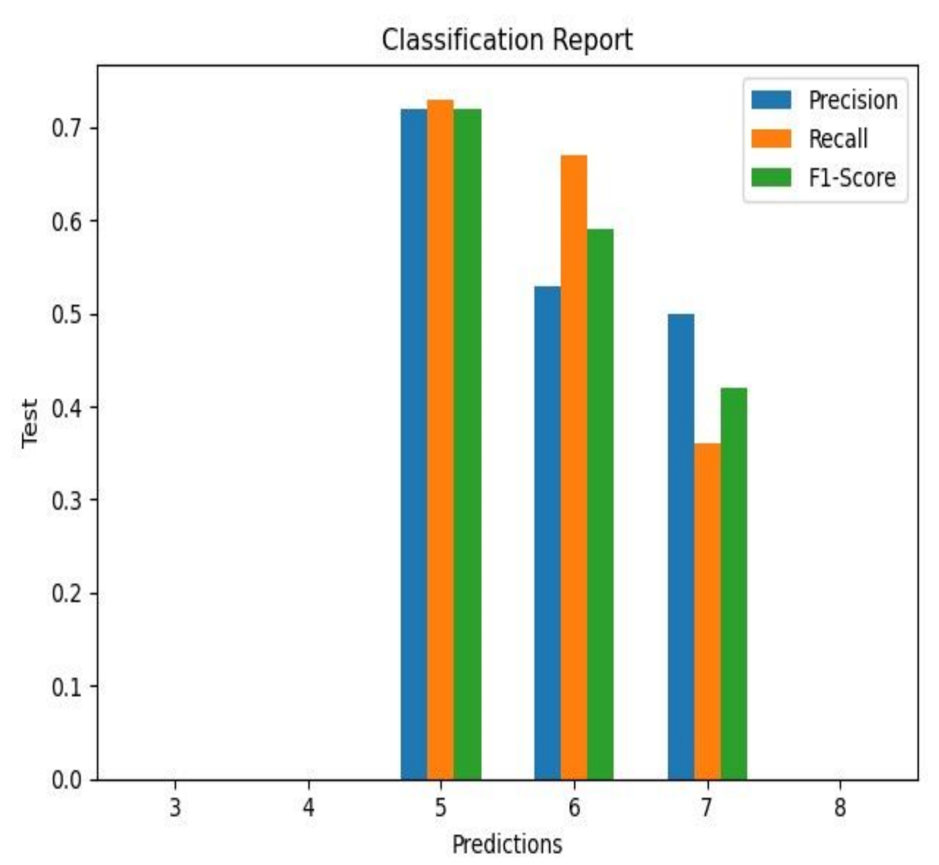
XGBoost Matrix Model

Kappa Score : 0.5329
Classification Report:

```
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         2
           4       0.00      0.00      0.00        11
           5       0.72      0.73      0.72       168
           6       0.53      0.67      0.59       156
           7       0.50      0.36      0.42        67
           8       0.00      0.00      0.00         8

    accuracy                           0.61       412
   macro avg       0.29      0.39      0.29       412
weighted avg       0.58      0.61      0.59       412
```

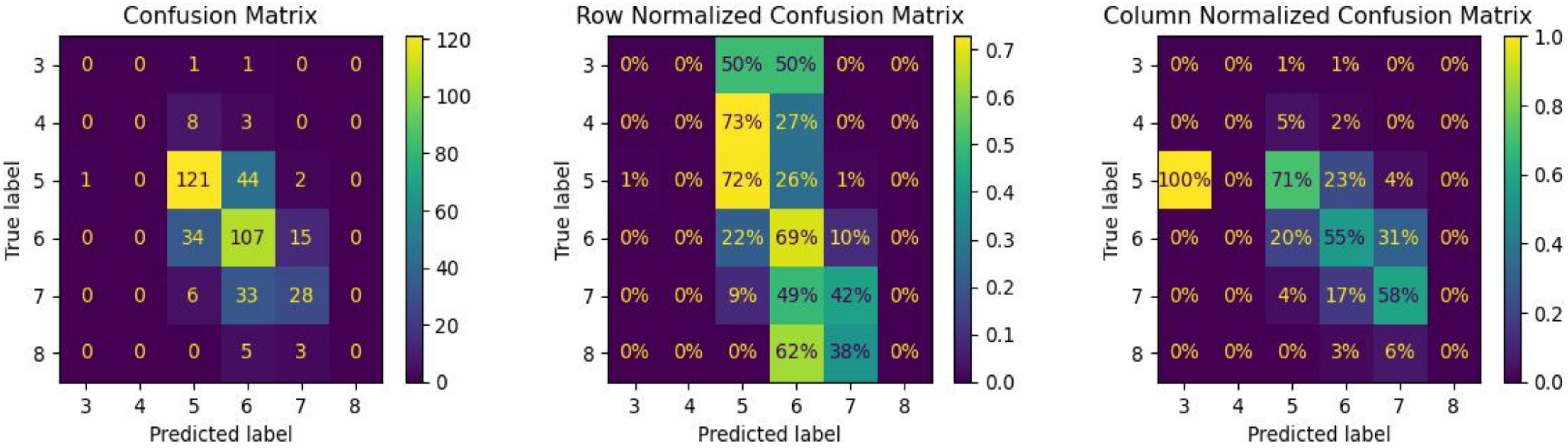


Gradient_Boosting Classifier Model:

Confusion Matrix from Fourth Gradient Boosted Search

Learning Rate: 0.05, Tree Depth: 2, Number of Trees: 30

Quadratic Kappa Score: 0.5528, Model Compute Time: 0.42 sec



Confusion Matrix

	3	4	5	6	7	8
3	0	0	1	1	0	0
4	0	0	8	3	0	0
5	1	0	121	44	2	0
6	0	0	34	107	15	0
7	0	0	6	33	28	0
8	0	0	0	5	3	0

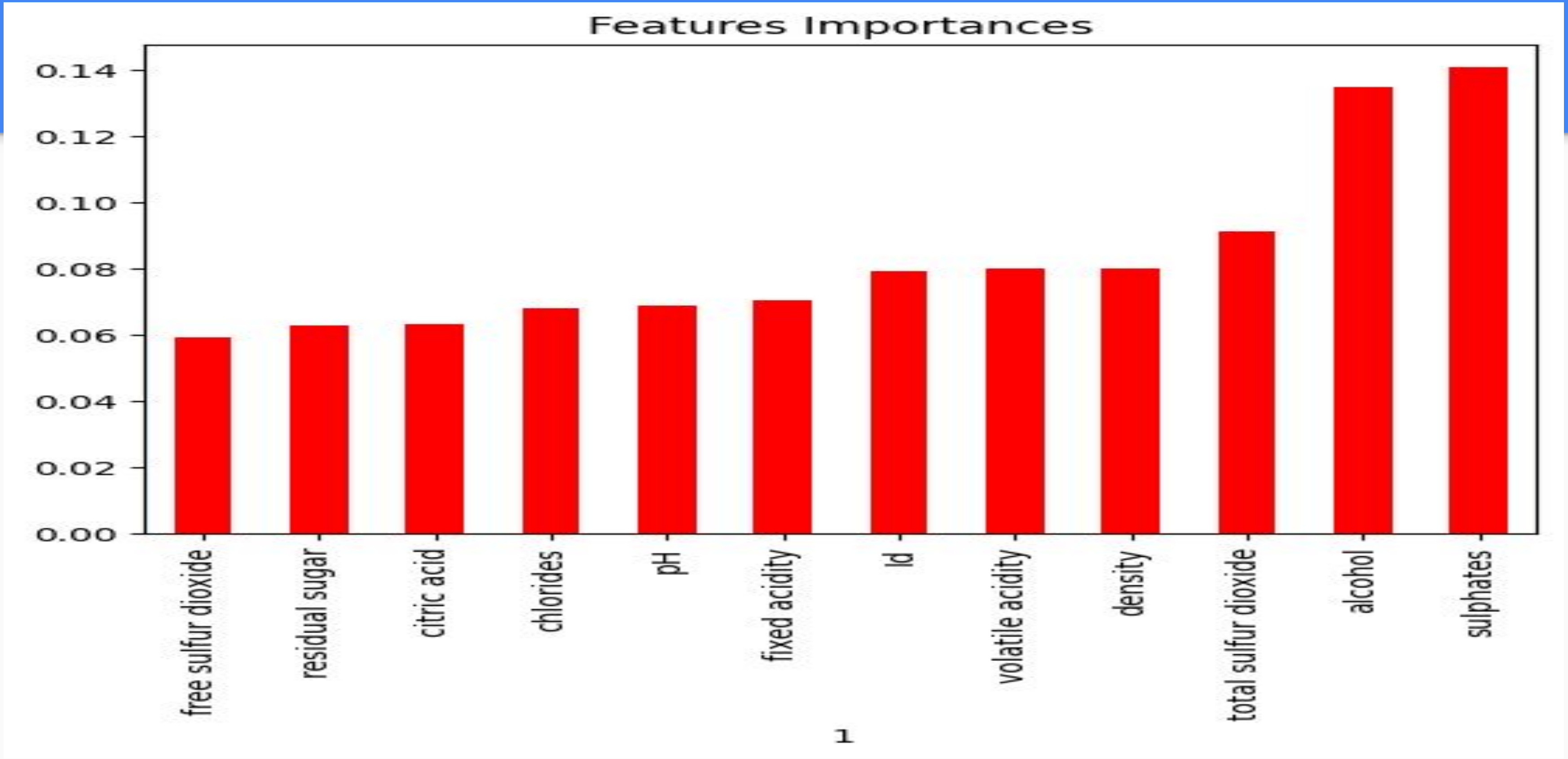
Row Normalized Confusion Matrix

	3	4	5	6	7	8
3	0%	0%	50%	50%	0%	0%
4	0%	0%	73%	27%	0%	0%
5	1%	0%	72%	26%	1%	0%
6	0%	0%	22%	69%	10%	0%
7	0%	0%	9%	49%	42%	0%
8	0%	0%	0%	62%	38%	0%

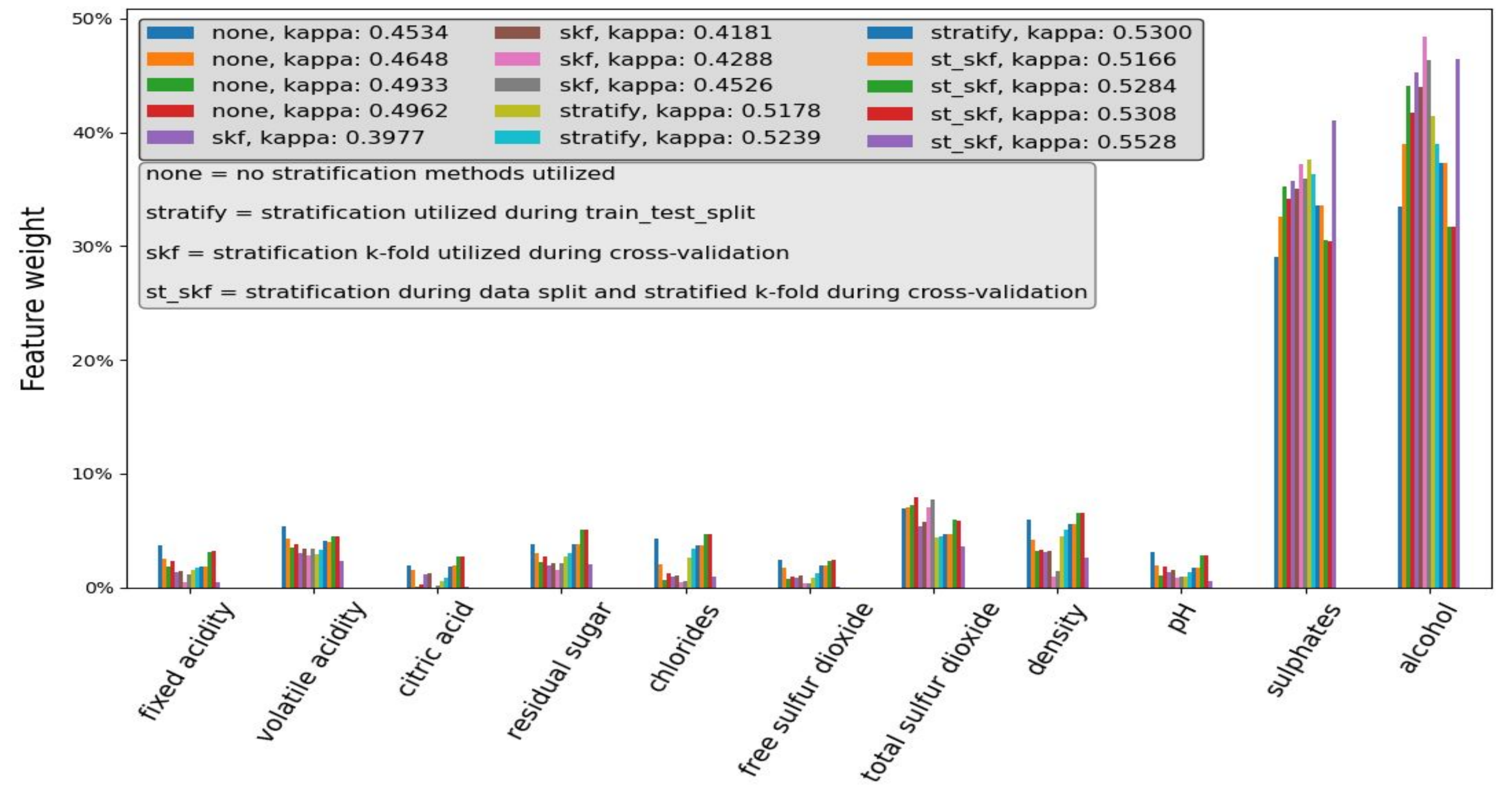
Column Normalized Confusion Matrix

	3	4	5	6	7	8
3	0%	0%	1%	1%	0%	0%
4	0%	0%	5%	2%	0%	0%
5	100%	0%	71%	23%	4%	0%
6	0%	0%	20%	55%	31%	0%
7	0%	0%	4%	17%	58%	0%
8	0%	0%	0%	3%	6%	0%

Random_Forest Model



Gradient Boosting

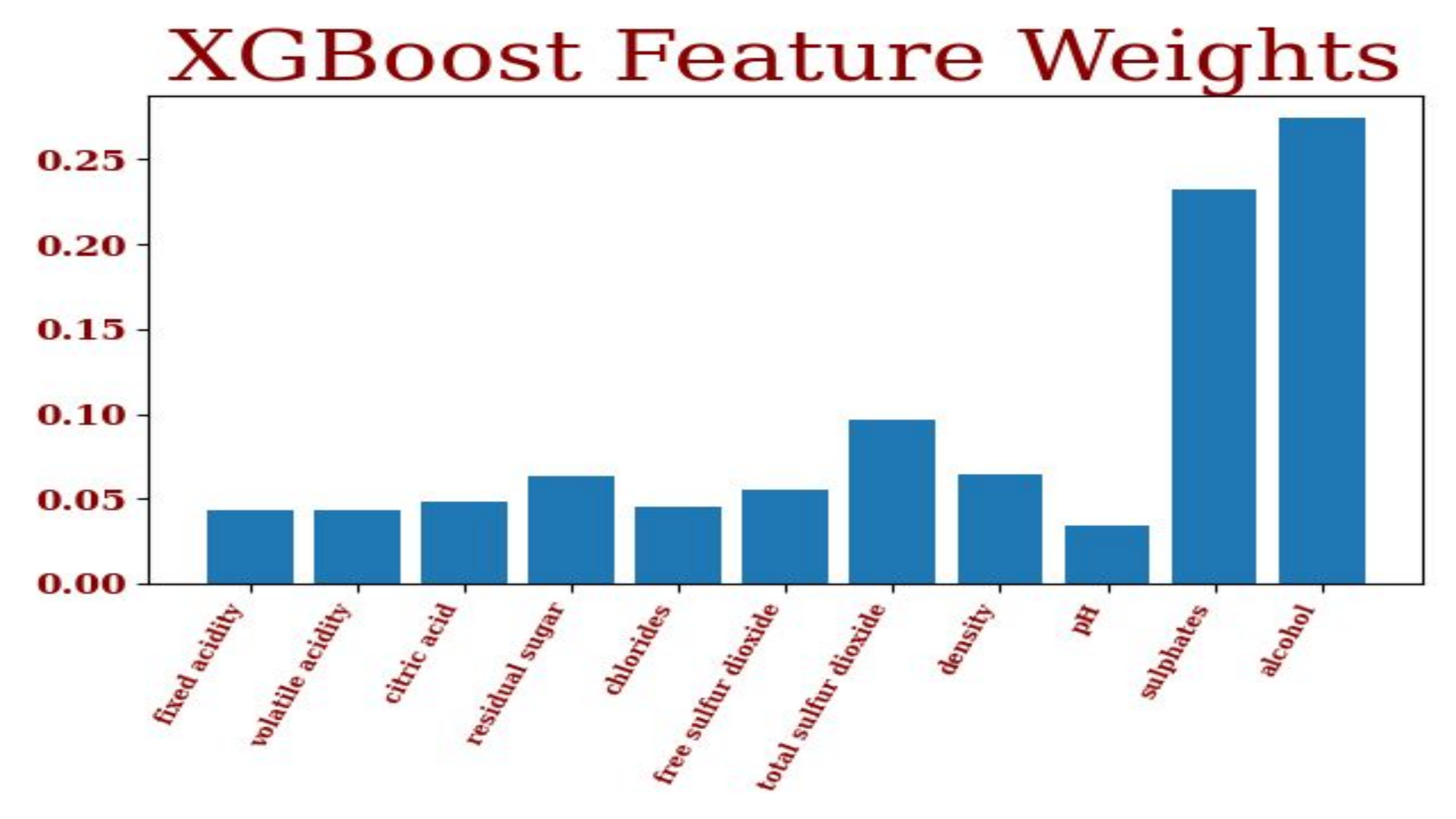


none = no stratification methods utilized

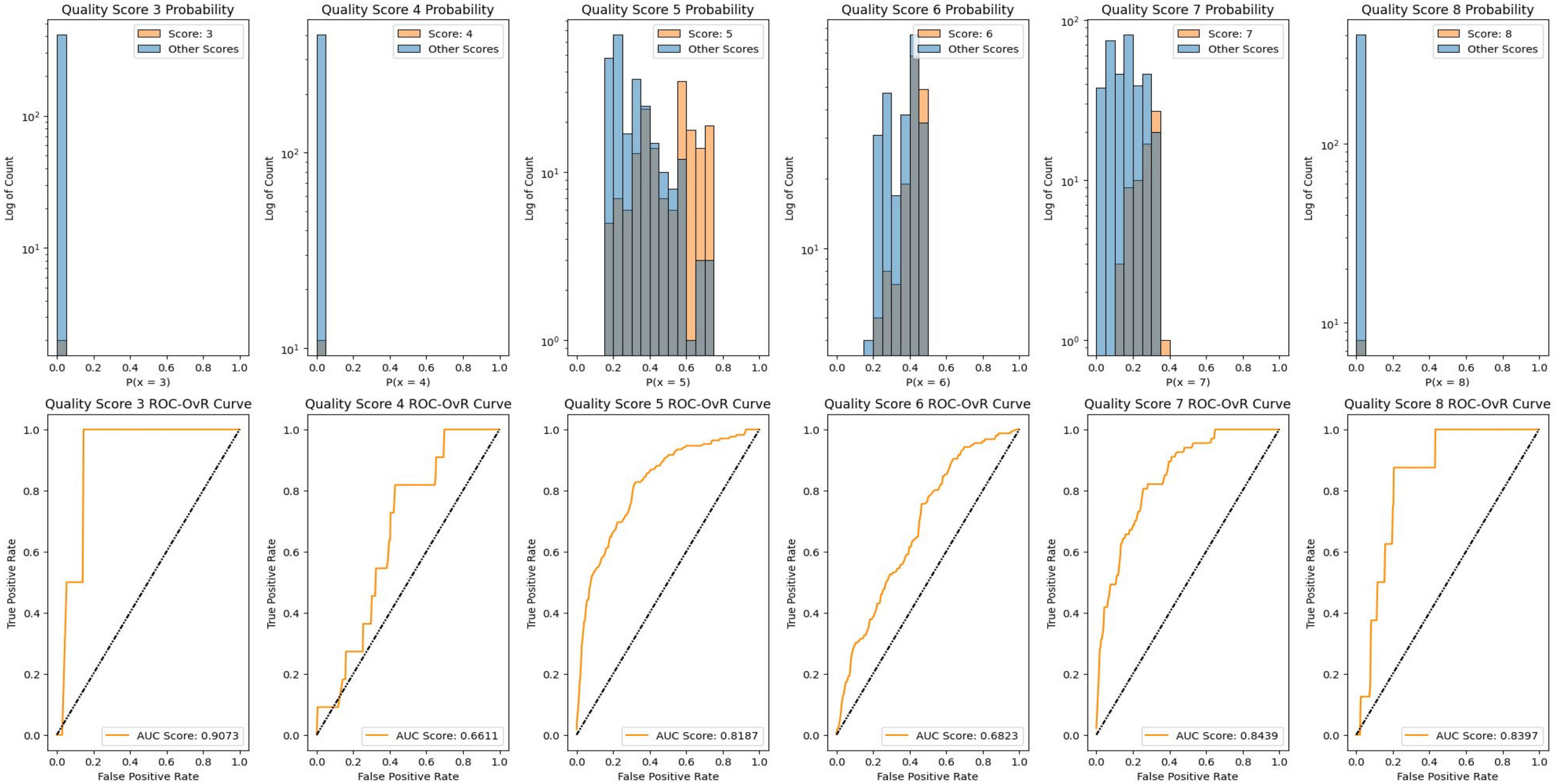
stratify = stratification utilized during train_test_split

skf = stratification k-fold utilized during cross-validation

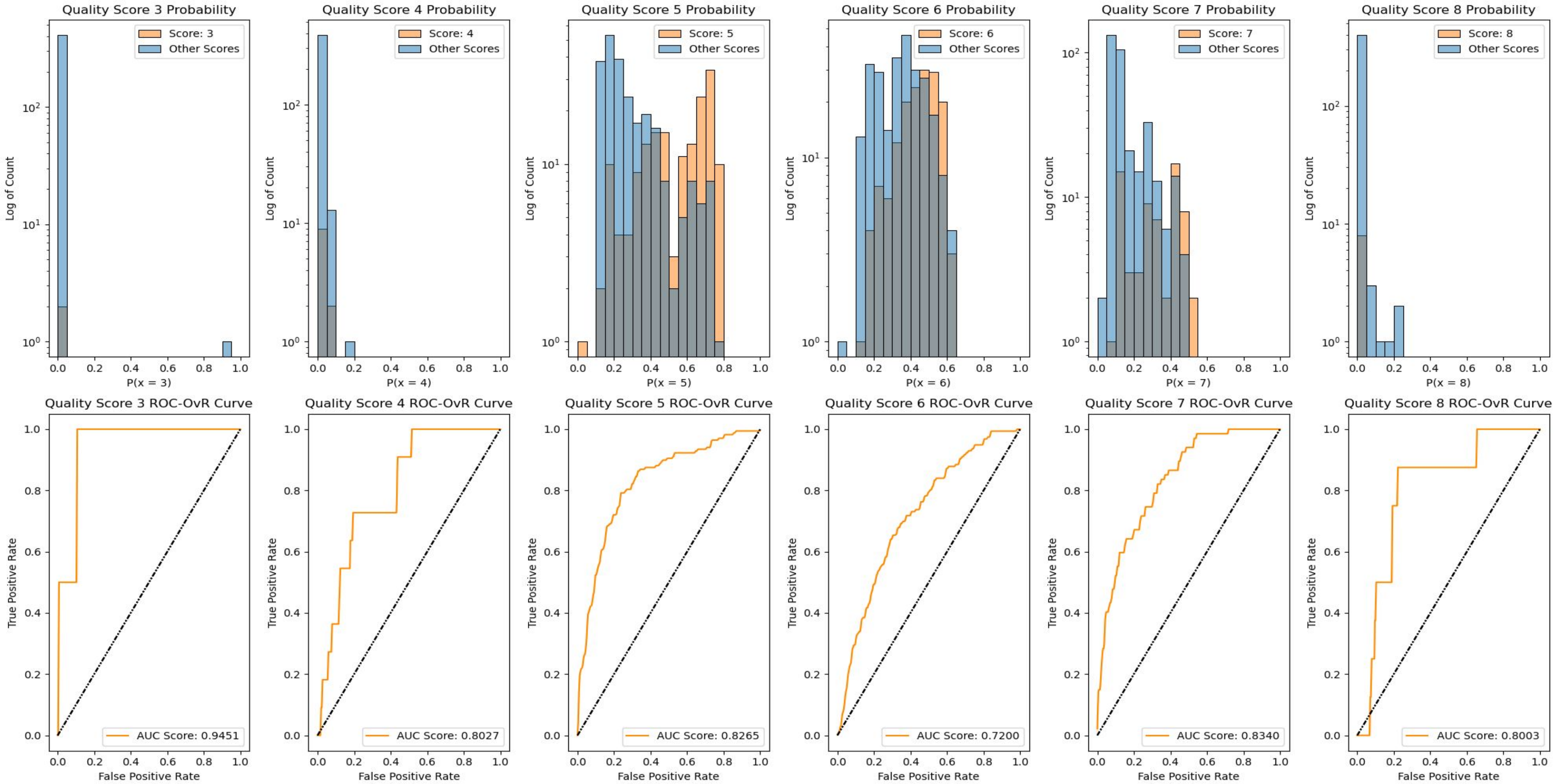
st_skf = stratification during data split and stratified k-fold during cross-validation


XGBoost Feature Weights
0.25
0.20
0.15
0.10
0.05
0.00
fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol

Random_Forest ROC plot



Gradient Boosting ROC-OvR plot



Recommended Model to Use

- **XGBoost** model showed the most promise for our dataset
- Second Highest kappa score
- Best runtime (seconds vs. 30+ minutes)
- Process of elimination with our multiclass dataset - knowing what would or wouldn’t be realistic

Parameters of model

Best Estimators: *30*

Best Learning Rate: *0.1*

Best Max Depth: *3*

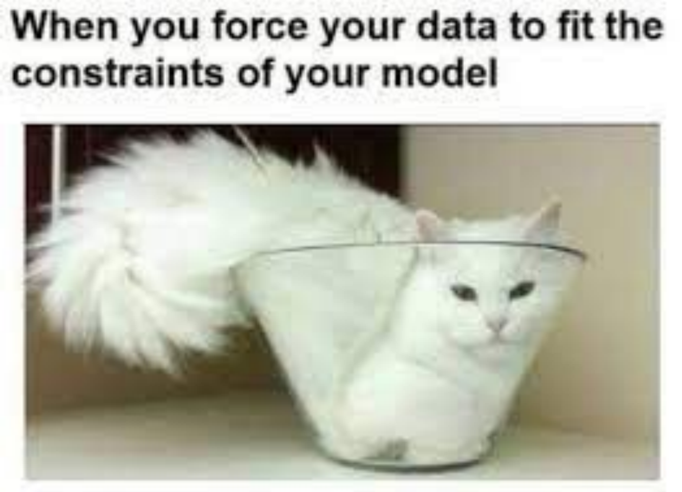
Accuracy: *0.61*

F1: *0.59*

Precision: *0.58*

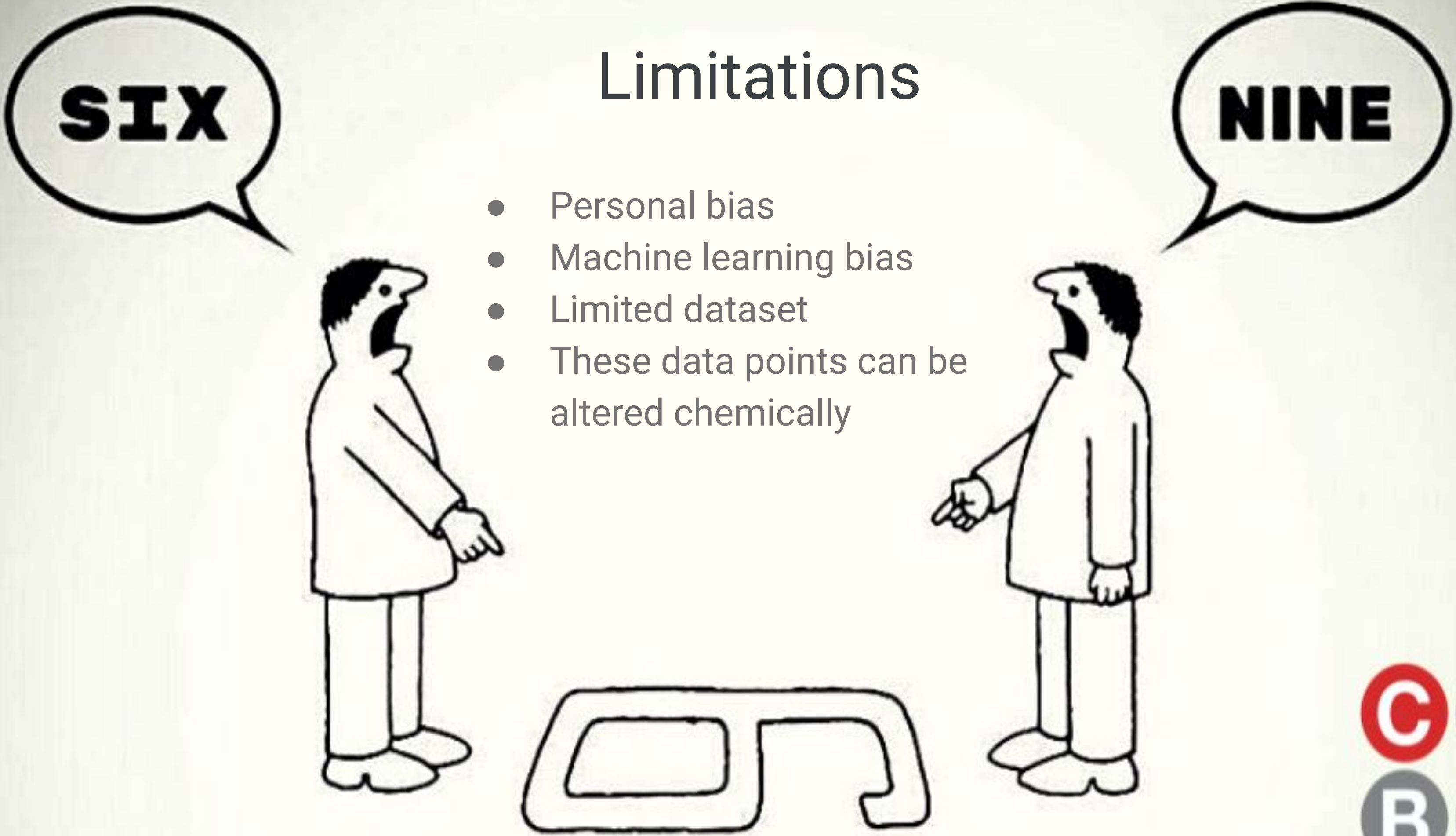
Recall: *0.61*

Runtime: *39.2 Seconds*


When you force your data to fit the constraints of your model

Limitations

- Personal bias
- Machine learning bias
- Limited dataset
- These data points can be altered chemically



Great advice.

Future Plans:

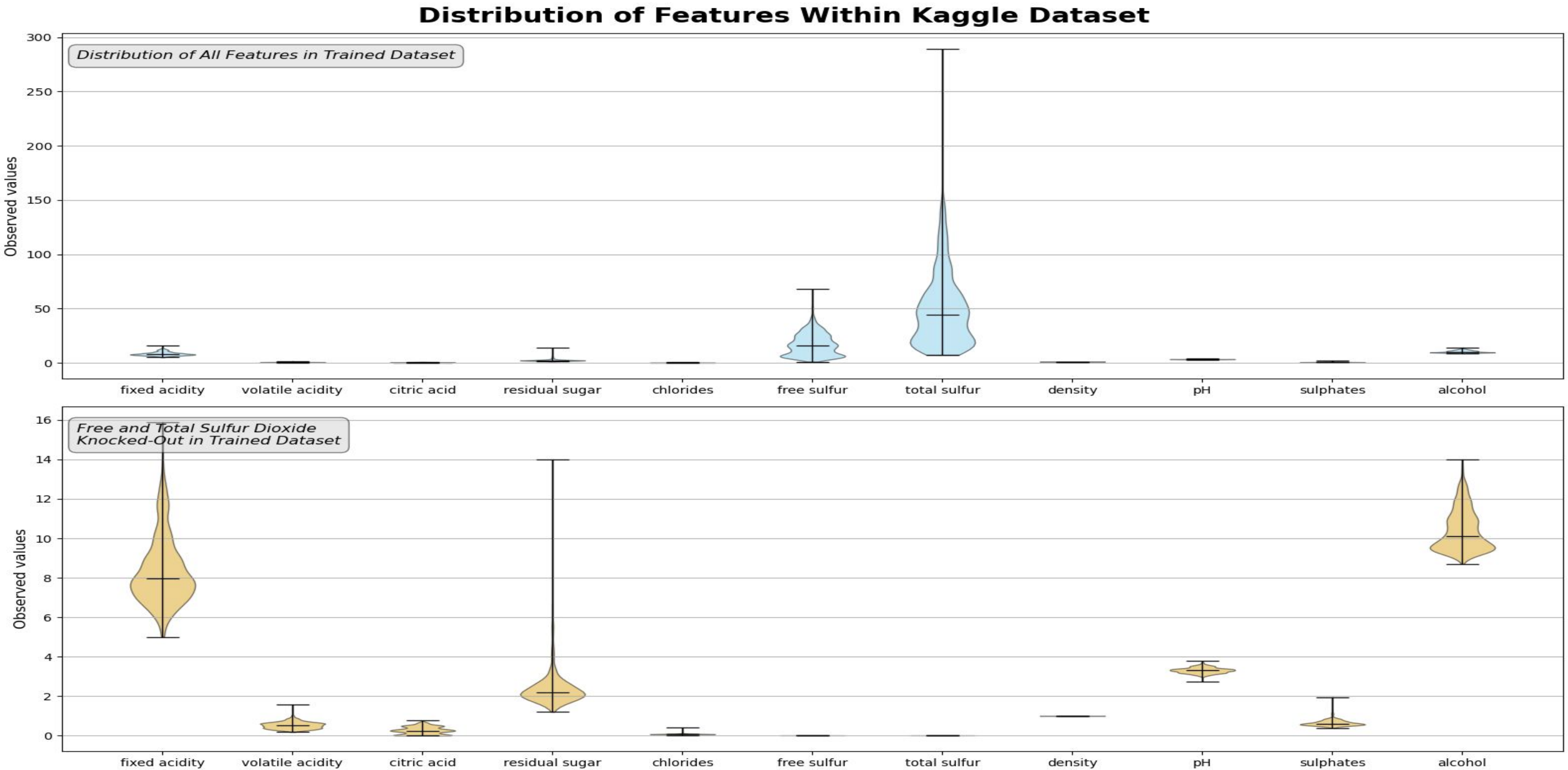
- *Using a wider variety and collection of wine data*
- *Look into building a system granularity depending on wine color*
- *Look for other classifiers to enhance the machine learning algorithm*

Thank You for Our Time Together

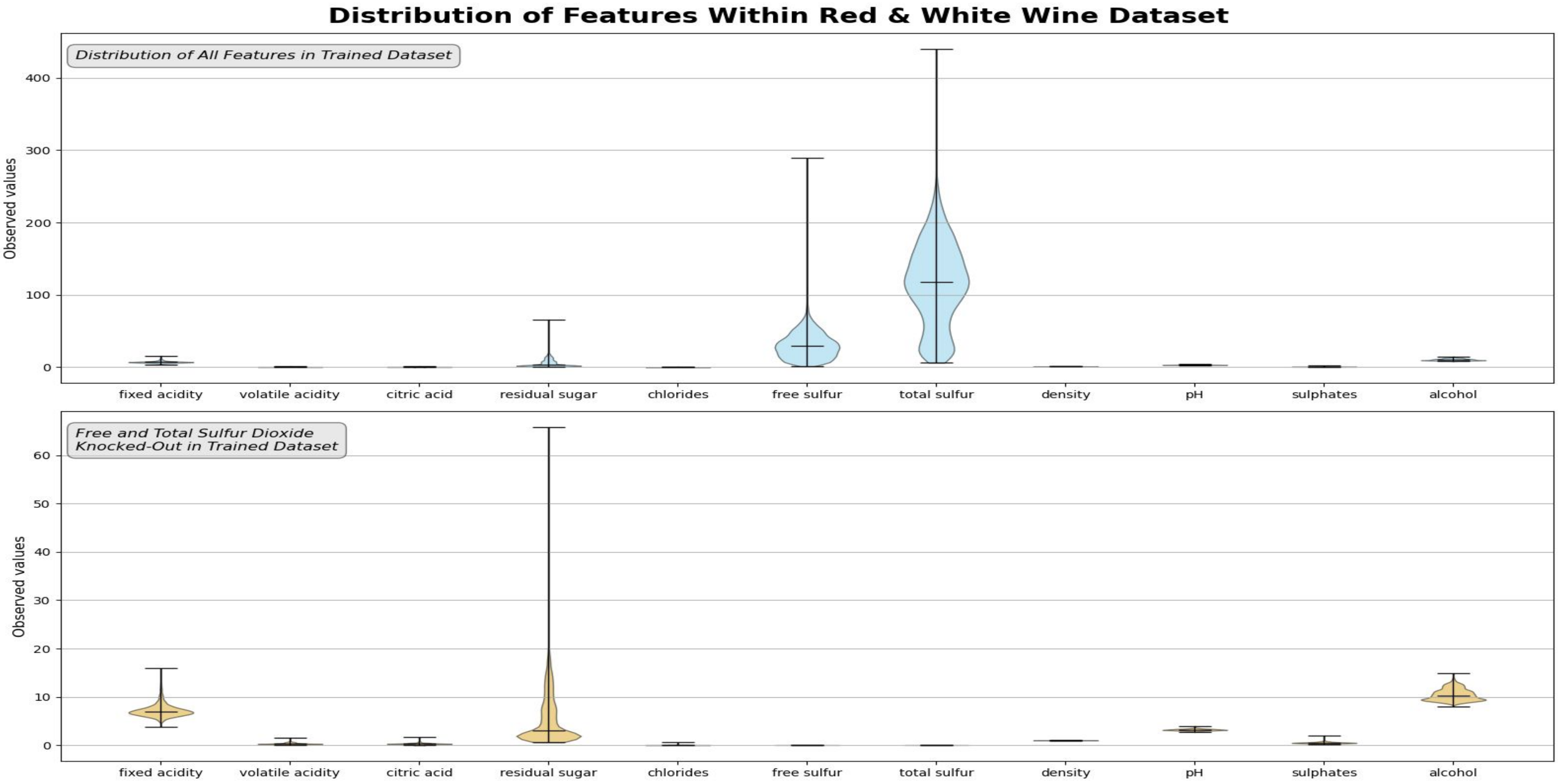
End of presentation

Bonus slides follow this slide

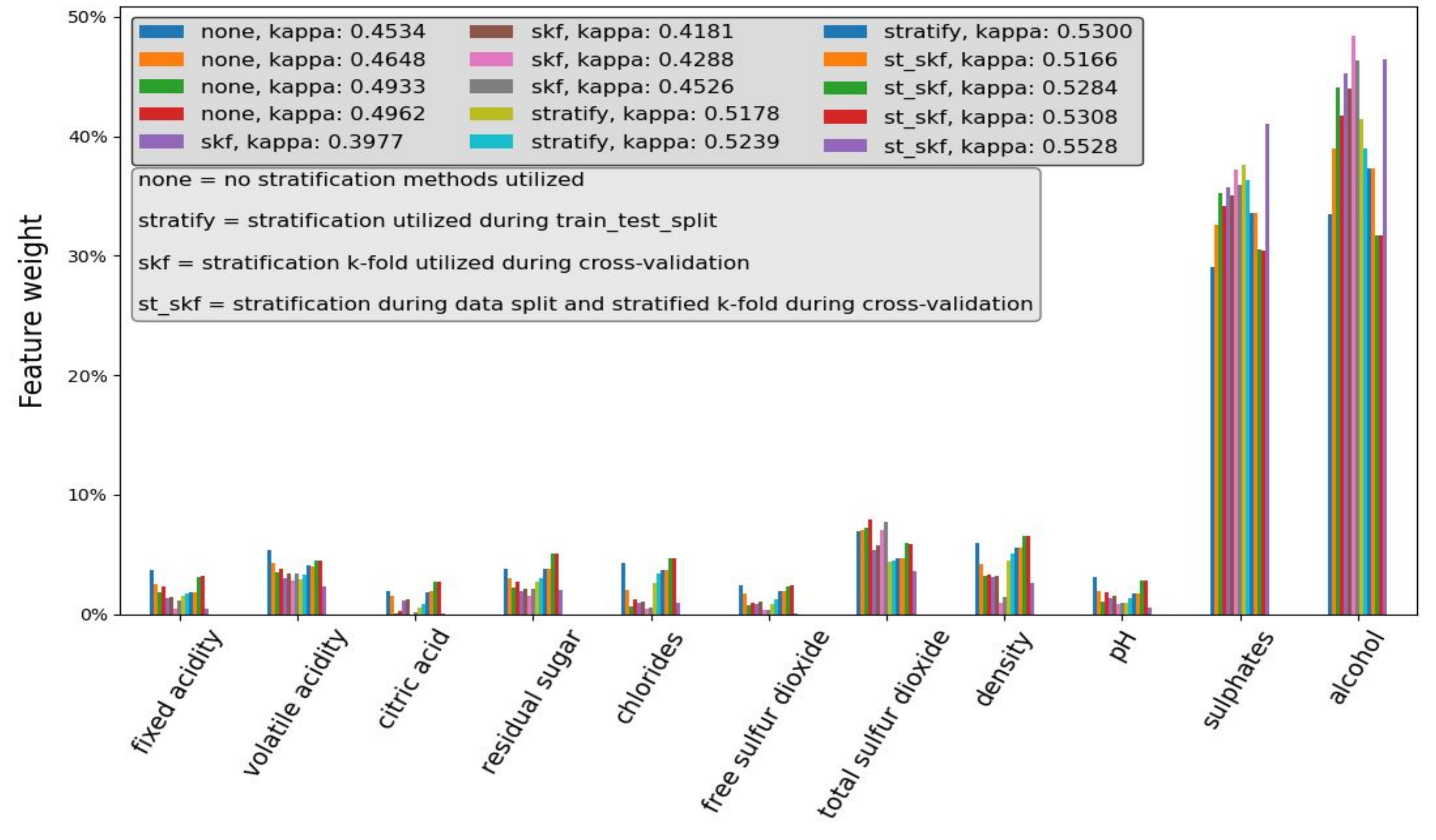
Distribution of Feature Values



Distribution of Feature Values in ‘Real’ Wine



Feature weights of Kaggle Data



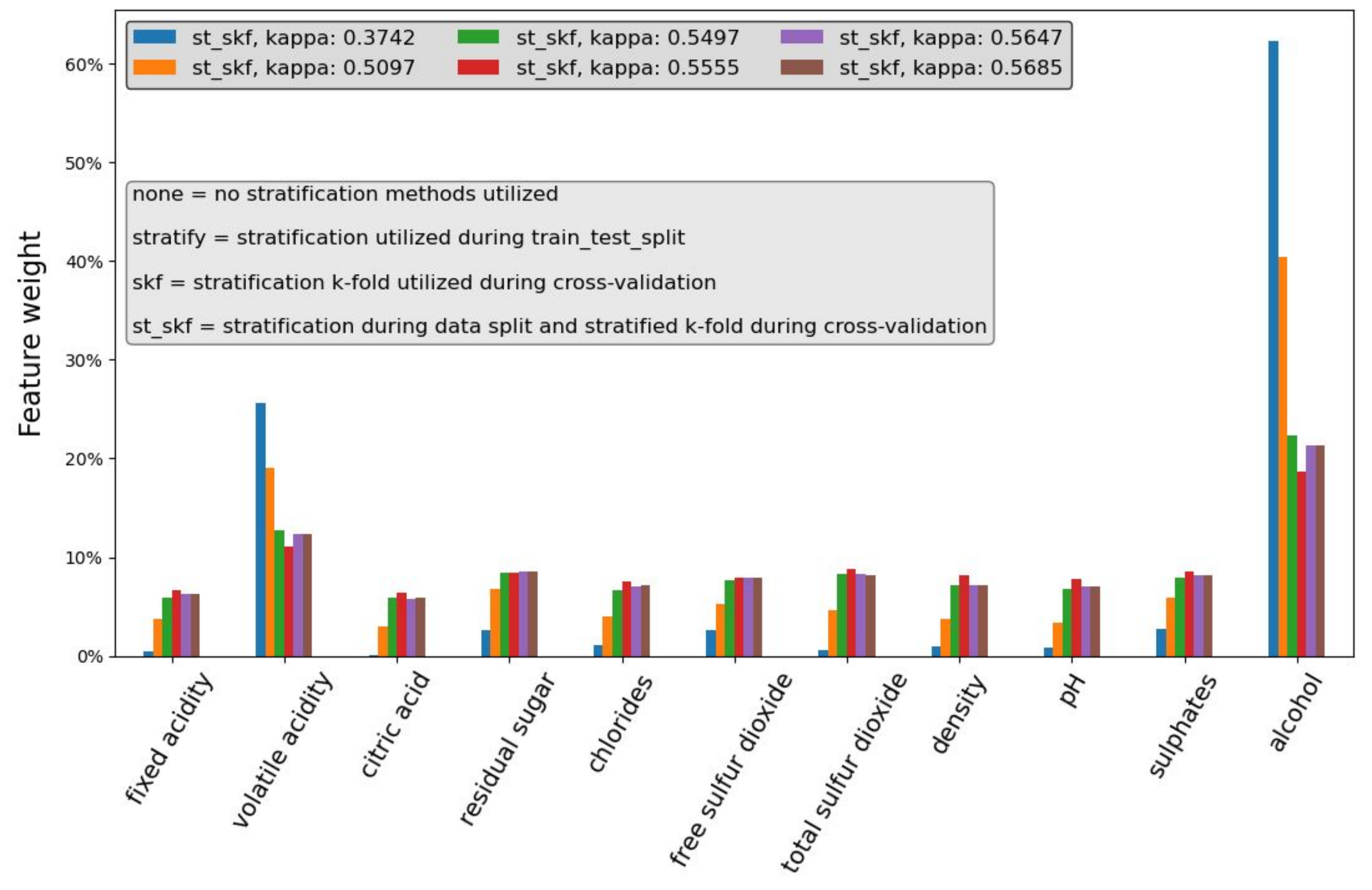
none = no stratification methods utilized

stratify = stratification utilized during train_test_split

skf = stratification k-fold utilized during cross-validation

st_skf = stratification during data split and stratified k-fold during cross-validation

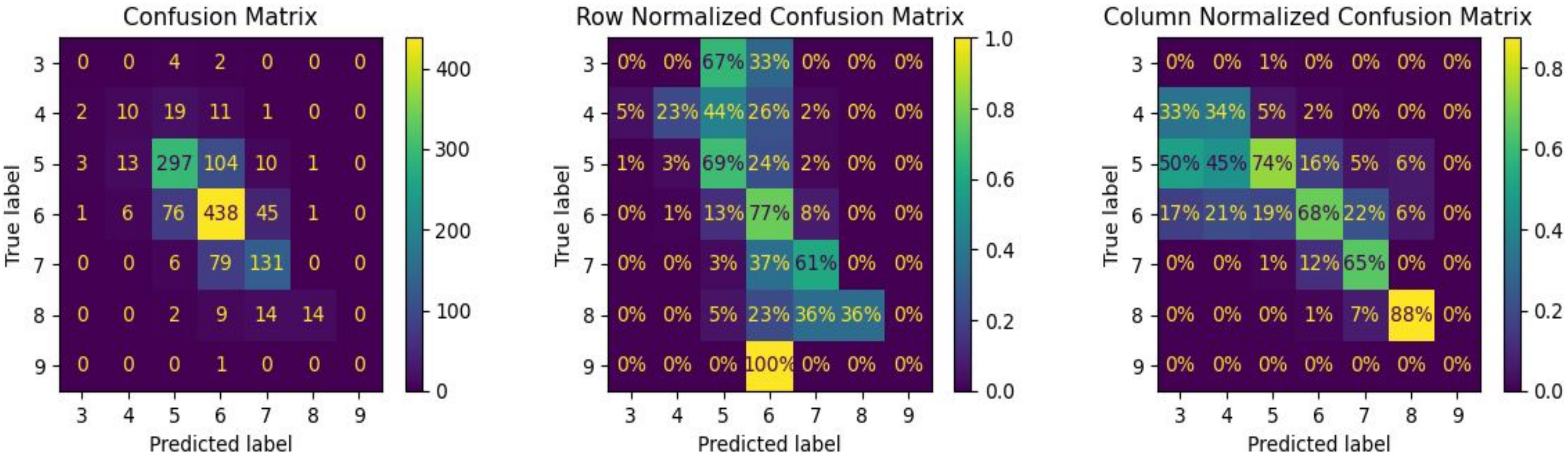
Feature weights of ‘Real’ Red & White Wine Data



Confusion Plot of ‘Real’ Red & White Wine Data

Learning Rate: 0.1, Tree Depth: 5, Number of Trees: 200

Quadratic Kappa Score: 0.6504, Model Compute Time: 23.92 sec



Confusion Matrix

	3	4	5	6	7	8	9
3	0	0	4	2	0	0	0
4	2	10	19	11	1	0	0
5	3	13	297	104	10	1	0
6	1	6	76	438	45	1	0
7	0	0	6	79	131	0	0
8	0	0	2	9	14	14	0
9	0	0	0	1	0	0	0

Row Normalized Confusion Matrix

	3	4	5	6	7	8	9
3	0%	0%	67%	33%	0%	0%	0%
4	5%	23%	44%	26%	2%	0%	0%
5	1%	3%	69%	24%	2%	0%	0%
6	0%	1%	13%	77%	8%	0%	0%
7	0%	0%	3%	37%	61%	0%	0%
8	0%	0%	5%	23%	36%	36%	0%
9	0%	0%	0%	100%	0%	0%	0%

Column Normalized Confusion Matrix

	3	4	5	6	7	8	9
3	0%	0%	1%	0%	0%	0%	0%
4	33%	34%	5%	2%	0%	0%	0%
5	50%	45%	74%	16%	5%	6%	0%
6	17%	21%	19%	68%	22%	6%	0%
7	0%	0%	1%	12%	65%	0%	0%
8	0%	0%	0%	1%	7%	88%	0%
9	0%	0%	0%	0%	0%	0%	0%

ROC-OvO plot