Predicting Style and Quality of Vinho Verde from its Physical Properties

Grigor Vardanyan MS student
Yerevan State University
Faculty Statistics: Applied Statistics and Data Science

Abstract: Now days there are lot of Sommelier (wine testing experts), who is manually test wines and tells us about wine quality. It is mean there is some functional dependency between properties of wines and quality. In this paper I will introduce some machine learning models for predicting wine quality based on its properties/features as we will try distinguishing between red and white wines

INTRODUCTION: According to the International Organization of Vine and Wine (OIV) during a year is produced millions of tons of wine and there is a need to understand quality of each separate wine. There are some researches to create sensors which based on chemical or physical properties try to understand quality of wine.

There is a dataset of wine, which contains wine characterization. Owners collated physical measurements from over 6000 examples of a local, young wine called vino Verde [3]. They had three experts to judge the quality of these wines. Quality of a wine was measured from 3 up to 9, and those are discreate variables.

In this project, I will show ML models which will classify wine style between red and white based on properties and other models which will try to predict quality of wine.

Related Work: There was a research for predicting quality with SVM models. They separated data between styles (red and white) and trained SVM for each type. Result was not so good. Accuracy for each model was 0.45 and 0.46.

Methods: For prediction I have used Logistic Regression [1] and XGBoost[2] models

Logistic regression[1] is a statistical model which uses a logistic function to a binary dependent variable model. Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1". For Coefficient estimation Logistic regression [1] use Maximum Likelihood Estimation (Eq 1).

$$\theta_{MLE} = arg_\theta \ max \log P\ (X|\theta)$$
$$= \ arg_\theta \ max \log \prod_i P(x_i|\theta)$$
$$= \ arg_\theta \ max \log \sum_i P(x_i\ |\theta)$$

Eq 1: Formula of maximum likelihood estimation

XGBoost[2] is new model by comparing to Logistic Regression[1]. It was developed based on Gradient boosting machine [4] algorithm and added more improvements. GBM [4] is additive model, were next weak learner tries to predict residuals of previous tress. GBM [4] and XGBoost[2] can be used as for regression as for classification tasks. Objective function for XGBoost (Eq. 2):

$$L^{(t)} = \sum_i^n l(\ y_i\ , y^{\wedge(t-1)} + \varphi_t(x_i)) + \ \Omega(\varphi_t)$$

Eq 2: Objective of XGBoost algorithm.

But in paper they approximated objective function with second order approximation.

$$L^{(t)} = \sum_{i}^{n} [\, l(\, y_i \,, y^{\wedge t-1}) + g_i \varphi_t(x_i)$$

$$+ \frac{1}{2} h_i \varphi_t^2(x_i)] + \;\; \Omega(\varphi_t)$$

$$g_i = \partial_{y^{\wedge t-1}} \, l \, (y_i, y^{\wedge (t-1)})$$

$$h_i = \partial^2_{y^{\wedge (t-1)}} \, l \, (y_i, y^{\wedge (t-1)})$$

Eq 3: Second order approximation of Eq 2.

XGBoost [2] has shown that it is very robust algorithm on many benchmarks, and now it is very popular algorithm.

Data set:

The dataset has 6497 rows. The dataset is unevenly split between two styles: 75% of examples are of white wines (4898) and 25% are of reds (1599) (Fig 1).

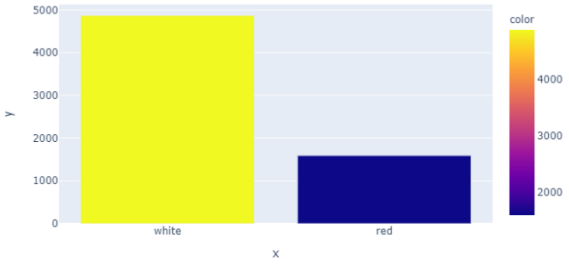


Fig. 1. Count of training data for each type of wine. Left white wine. Right red wine.

From point of quality we here also have unbalanced split of data between 7 qualities: 3 - 30, 4 - 214 ,5 - 2128, 6 - 2820, 7 - 1074, 8 - 192, 9 – 5 (Fig 2)

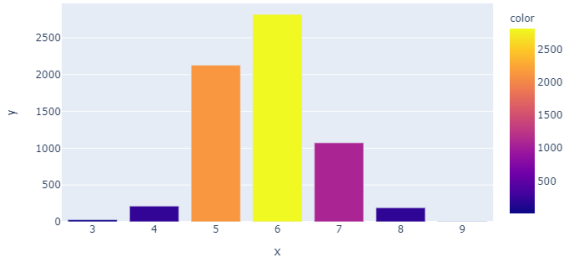
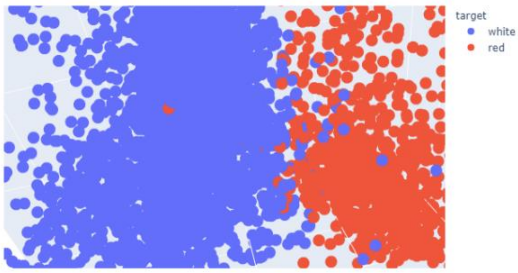


Fig. 2. Count of training data for each quality of wine.

Each wine or row is described by 11 physicochemical features, all continuous variables: fixed acidity, volatile acidity, citric acid concentration, residual sugar content, chlorides concentrations, free sulfur dioxide content, total sulfur dioxide content, density, pH, sulphates concentrations, and alcohol content. Besides 11 features we also have 2 target variables: style (Fig 1), quality (Fig 2).

Data exploration: From Fig-1 and Fig-2 we saw that we have unbalanced data. As mentioned in Dataset out all features are continues variables. First, I visualized data to understand do we have enough evidence based on visualization, to use liner classifiers. As our data is high dimensional, I reduced data with PCA to be able to visualize it (Fig 3).



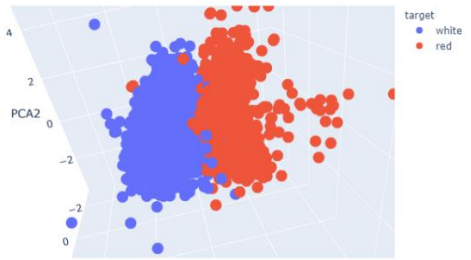


Fig. 3. Visualization for type of wine in 3D, where axes are first 3 components. First image is zoomed to see intersection of two classes.

In Fig 3 and 4 we can see that data is linearly well separable. We can assume that same thing will be in high dimensional case. We can prove our hypothesis based on Linear Regression [1] model.

Fig. 4. Visualization for type of wine in 2D, where axes are first 2 components.

Preprocessing: First I removed all missing values for dataset, then for both tasks: classifying type of wine (Fig 1) and quality (Fig 2), I have used StandardScalar from Scikit-Learn.

Training for wine type: Best parameters for each model was chosen based on Grid Search Cross Validation [5]. After hyperparameters search, best parameters were used for training and testing. Data was split between 80% and 20% for training and testing.

For type prediction was used Logistic Regression [1] and XGBoost[2].

Logistic Regression: Hyperparameter space for Logistic regression [1] we can see in Fig 5. Best result of Grid search CV on Fig 6. We can see from (Fig 7) test evaluation of Logistic Regression [1], based on best parameters (Fig 6).

```
'C': np.logspace(-4, 4, 20),
'solver': ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'],
'penalty': ['l1', 'l2'],
'class_weight': [None, 'balanced']
```

Fig. 5. Hyperparameter space of Logistic regression for Grid Search Cross Validation.

```
{'C': 1.623776739188721,
 'class_weight': None,
 'penalty': 'l1',
 'solver': 'liblinear'}
```

Fig. 6. Best parameters based on Grid search CV.

	precision	recall	f1-score	support
white	0.99	0.99	0.99	529
red	1.00	1.00	1.00	1604
accuracy			1.00	2133
macro avg	0.99	0.99	0.99	2133
weighted avg	1.00	1.00	1.00	2133

Fig. 7. Test evaluation of Logistic regression based on best parameters.

XGBoost: On Fig 8 we can see Hyperparameter space for XgBoost [2] model. Best result of Grid Search CV [5] which applied on Fig 8 parameters space we can see on Fig 9.

Test evaluation of XGBoost [2] model based on best hyper parameters are identical to best Logistic Regression [1] model. Result of test evaluation presented on Fig 10.

```
'gamma': [0.5, 1, 1.5, 2, 5],
'subsample': [0.6, 0.8, 1.0],
'colsample_bytree': [0.6, 0.8, 1.0],
'max_depth': [3, 4, 5, 6, 8],
'n_estimators':[100, 150, 200, 250, 300]
```

Fig. 8. Hyperparameter space of XGBoost for Grid Search CV.

```
{'colsample_bytree': 0.6,
 'gamma': 0.5,
 'max_depth': 3,
 'n_estimators': 300,
 'subsample': 1.0}
```

Fig. 9. Best parameters based on Grid search CV.

	precision	recall	f1-score	support
white	0.99	0.99	0.99	529
red	1.00	1.00	1.00	1604
accuracy			1.00	2133
macro avg	0.99	0.99	0.99	2133
weighted avg	1.00	1.00	1.00	2133

Fig. 10. Test evaluation of XGBoost based on best parameters.

Training for wine quality: From Fig 2 we see that our data is imbalanced. Especially quality 3, 4, 8 and 9 have few dataset (30, 214, 192, 5). Such cases we need to apply boosting models like XGBoost [2]. Idea of using XGBoost [2] is following as each new tree trying to predict misclassification or residuals of previous tree, less classes should be predicted by new trees.

I used same hyperparameter space (Fig 8) for quality prediction. Final hyperparameters and classification result based on test set presented on Fig 11 and Fig 12

```
{'colsample_bytree': 0.8,
 'gamma': 0.5,
 'max_depth': 8,
 'n_estimators': 300,
 'subsample': 0.8}
```

Fig. 11. Best parameters based on Grid search CV for quality prediction.

	precision	recall	f1-score	support
3	0.00	0.00	0.00	8
4	0.37	0.10	0.16	67
5	0.71	0.69	0.70	721
6	0.63	0.76	0.69	912
7	0.66	0.51	0.57	356
8	0.87	0.29	0.44	68
9	0.00	0.00	0.00	1
accuracy			0.66	2133
macro avg	0.46	0.34	0.37	2133
weighted avg	0.66	0.66	0.65	2133

Fig. 12. Test evaluation of XGBoost based on best parameters for quality prediction.

From Fig 12 we see that results for 3 and 9 is very bad as those has fewest data (40 and 5 among 6900). Next thing that we can try is group data between low quality middle quality and high quality. It will give us to have more data under one group. Under low we will have 3 and 4 qualities, under middle 5 and 6, and under high 7, 8 and 9.

I have trained XGBoost[2] model on above mentioned changes regarding to target variable. As an hyperparameter space I took same which was used for other XGBoost[2] models (Fig 8). Here we can see result (Fig 13).

	precision	recall	f1-score	support
middle	0.77	0.57	0.65	425
high	0.36	0.05	0.09	75
low	0.86	0.95	0.90	1633
accuracy			0.84	2133
macro avg	0.66	0.52	0.55	2133
weighted avg	0.82	0.84	0.83	2133

Fig. 13. Test evaluation of XGBoost based on best parameters.

Random Forest: In the end of all experiments I also applied Random Forest [6] to compare tree-based models together. As other models I used Grid Seach CV method to find best parameters.

	precision	recall	f1-score	support
middle	0.72	0.37	0.49	425
high	0.00	0.00	0.00	75
low	0.82	0.96	0.89	1633
accuracy			0.81	2133
macro avg	0.52	0.45	0.46	2133
weighted avg	0.77	0.81	0.78	2133

Fig. 14. Test evaluation of Random forest based on best parameters.

Balanced Data: As we saw that there are some classes which have higher frequency than others (e.g. 5, 6 and 7). During experiments I reduced amount of those classes until 214 (Fig 15). 214 is frequency of 4 class. As class 3 has 192 rows I and difference is not so high

(between 3 and 4) so I gave more rows to class 4 and reduced all data based on it. Reason not choosing same frequency as 3 or 9, is because they have 40 and 5 rows. So, it is very small for ML problem.

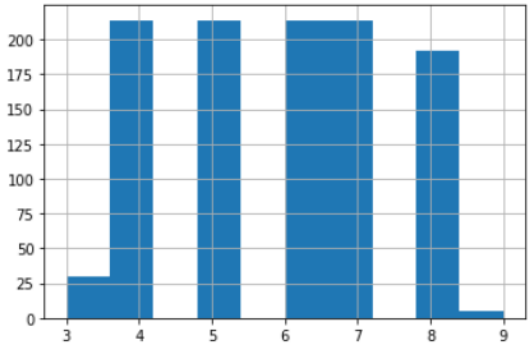


Fig. 15. Frequencies of balanced data.

	precision	recall	f1-score	support
3	0.00	0.00	0.00	11
4	0.55	0.48	0.51	79
5	0.49	0.52	0.51	77
6	0.34	0.31	0.33	64
7	0.31	0.38	0.34	61
8	0.46	0.53	0.49	64
9	0.00	0.00	0.00	2
accuracy			0.43	358
macro avg	0.31	0.32	0.31	358
weighted avg	0.42	0.43	0.43	358

Fig. 16. Prediction of balanced data.

Conclusion: We have seen that for both problems (classification of type and quality), we had unbalanced data (Fig 1 and 2).

First problem we were able to get very good result with simple model (Fig 7), even though new modern approache gave us same result (Fig 10). The explanation of it is, we have linear separable data. That is why we get such good accuracy on test result with Logistic Regression [1]. So, for type classification I chose Logistic Regression [1], as was mentioned it is simpler model, and testing results is same which was measured based on precision and recall.

For the second problem (quality prediction) we were not able to get good result, as unbalancedness is more revealed in this problem. I tried to reduce rows of those classes which have high frequency (Fig 15), but results were not so good as we saw (Fig 16). I was able to get more good result with transformation of target variables into 3 groups low, middle and high. For this experiment I used two model Random Forest [5] and XGBoost [2]. If we compare Fig 14 and 13 it is obvious that Boosting based model overcome ensemble-based model. So, if we will decide to use model which predicts between low, middle and high I will choose defiantly XGBoost [2] model based on Precision and Recall score.

Let's understand was the idea of transformation of target variables into low, middle and high good or no? Comparing Fig 12 and Fig 13 we can see that precision and recall of low quality (Fig 13) is higher than for 3 and 4 qualities(Fig 12), as well precision and recall for middle quality(Fig 13) is higher than 5, 6 and 7(Fig 12), but if we will look at quality 8 (Fig 12), we see that it has high precision but overall it decreased when it was under high quality(Fig 13).

Selection of model is more subjective for quality prediction. If for us is import low quality and middle quality intervals so yes, we will choose target variable transformation trick, but if we want to have model which is robust for high quality for 8 we will choose first approach, as precision and recall was very high for quality 8. Only class which we were not able predict well is 9 class which has 5 rows, which is very small data.

REFERENCES:

[1]. D. R. Cox, "The Regression Analysis of Binary Sequences," Journal of the Royal Statistical

Society. Series B (Methodological), vol. 20, no. 2, pp. 215–242, 1958

[2]. Tianqi Chen, Carlos Guestrin “XGBoost: A Scalable Tree Boosting System”

[3]. S. Tempere, S. Per´ es, A. F. Espinoza, P. Darriet, E. Giraud-H ` eraud, ´ and A. Pons, “Consumer preferences for different red wine styles and repeated exposure effects,” Food Quality and Preference, vol. 73, pp. 110–116, apr 2019

[4] Jerome H. Friedman – “Greedy function approximation: A Gradient Boosting Machine”.

[5] Petro Liashchynskyi, Pavlo Liashchynskyi - Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS

[6]. Leo Breiman - RANDOM FORESTS