Which wine is better ?

Chang Bo Cian
Yen-hsuan (Kuan-wei) Tseng

June 14, 2024

Abstract

- Grape wine is one of the most complex beverages, with its taste influenced by numerous factors. Previously, Piyush Bhardwaj et al. (2016) predicted wine quality using machine learning based on physical and chemical properties, identifying distinct patterns in high-quality wines. However, this method is not easily accessible to the general public. Therefore, this report aims to identify market-available features that can help recognize widely acclaimed wines.

Data Understanding

- Xwines is an open dataset collected from the web in 2022 and pre-processed for broader free use.
- It includes ratings on a scale from 1 to 5 for 228,000 different types of wines produced in 62 countries.
- There are over 20 million observations, covering the following features:

WineID	Vintage	Rating	Date	WineName	Type	Elaborate	Grapes	Harmonize	ABV	Body	Acidity	Country	RegionName	WineryName
136103	1950	4.0	2019-10-14 11:20:52	Lambrusco Emilia	Red	Varietal/100%	['Lambrusco']	['Beef', 'Pasta', 'Lamb', 'Game Meat']	7.5	Full-bodied	High	Italy	Emilia	Riunite

- Source: `https://github.com/rogerioxavier/X-Wines`

Motivation

- Imagine you are a consumer, unable to decide which wine to choose despite recommendations from store staff. (You might need to choose one out of five)
- Imagine you own a wine cellar and need to select which wines to import from the existing 228,000 options.
- Or, as the owner of a winery, you struggle to understand the preferences of the general public, leading to consistently poor sales for certain wines.

Approach

- Let's start from the perspective of the wine importer and consumer.
- Initially, we considered an OLS model like this:

$$\begin{aligned}\text{Rating} = \beta_0 &+ \beta_1(\text{Type}) + \beta_2(\text{Acidity}) \\ &+ \beta_3(\text{Grapes}) + \beta_4(\text{Body}) \\ &+ \beta_5(\text{Country}) + \beta_6(\text{Vintage}) + \dots\end{aligned}$$

Challenges

- However, there are more than 10,000 unique values in total, leading to the curse of dimensionality.
- One might consider using step-wise selection or LASSO, but they are invalid.
- What about group LASSO?

$$\underset{\beta}{\text{minimize}} \; \|y - X\beta\|_2^2 + \lambda \sum_{g=1}^{G} \|\beta_{lg}\|_2$$

Sommelier's Knowledge

- In the wine industry, each winery usually has its own master whose techniques can significantly affect the taste.
- Climate is closely related to the quality of grapes, with different vintages experiencing different climatic conditions.
- What if we only consider winery$\times$year? It might be a good proxy for most of the variables, such as Body, Acidity, ABV, etc.

Updated Model

- Now, the question becomes: In which winery$\times$year were the best wines produced?

$$\text{Rating} = \sum_i \sum_j \beta_{ij}(\text{winery}_i \times \text{vintage}_j)$$

Data Processing & Cleansing

- For users who rated the same wine multiple times, take the most recent rating.
- Consider red wines first.
- Filter out winery$\times$vintage combinations with fewer than 500 ratings.
- For the same wine with ratings from different users, take the average.

Data Exhibition

```
columns_to_group = ['WineID', 'Label']
df = df.groupby(columns_to_group)['Rating'].mean().reset_index()
```

WineID	Label	Rating
168335	Alfredo Roca 2014	3.336207
181668	Wente Vineyards 2014	4.045455
168194	Trapiche 2015	3.579365
174913	Penfolds 2018	3.500000
156421	Matarromera 2016	3.833333

Data Transformation

```
X = sm.add_constant(pd.get_dummies(Label, sparse=True).astype(int))
y = df['Rating']
```

d'Arenberg 2011	d'Arenberg 2012	d'Arenberg 2013	d'Arenberg 2014	d'Arenberg 2015	d'Arenberg 2016
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
...	...	...	...	...	...

- The data size becomes 64,189 $\times$ 5,150.

Concerns

- There may be some missing controls, such as:
  - Storage conditions of the wine
  - Drinking methods
  - Environment
- Moreover, the system of $X^T\beta = y$ is inconsistent; the solution can only minimize the norm $||X\beta - y||$.
  - $\hat{B} = X^{\dagger}y$, where $X^{\dagger}$ is the pseudo-inverse of X.

Large Scale Hypothesis Testing

- Among 5,150 coefficients, more than 1,000 terms are below the significance level of $p = 0.05$.
- The Family-Wise Error Rate (FWER) is $1 - (1 - 0.05)^{1000} = 1$.
  - As a wine cellar owner, you'll have a 100% chance of importing the wrong merchandise.
- We then apply the idea of Bonferroni correction. The concept is to extract a list of coefficients from the OLS results that are high and significant, ensuring the probability of a Type I error for that list is below 5%.

Large Scale Hypothesis Testing, Cont'd

- Pick n, the number of wines you want on your recommendation list.
- We have $(1 - \alpha/n)^n > (1 - \alpha)$, where α is the original significance level.
- For example, with $n = 50$ and $\alpha = 0.005$, the probability of encountering a misleading recommendation in this "Top 50 Wines" list is only 0.5

Results

Wine	Coefficient	t-value	P-value
Vega Sicilia 2015	0.931377	3.977395	0.00006976**
Vega Sicilia 2005	0.927762	3.961956	0.00007443**
Silver Oak 2014	0.916108	5.050453	0.00000044****
Vega Sicilia 2000	0.909533	3.884110	0.00010282**
Vega Sicilia 2013	0.909367	3.883401	0.00010312**
Vega Sicilia 2012	0.909832	3.885386	0.00010228**
Vega Sicilia 2006	0.899816	3.842614	0.00012186**
Vega Sicilia 2004	0.895414	3.823813	0.00013154**
Vega Sicilia 2010	0.891497	3.807088	0.00014075**
Vega Sicilia 2007	0.862506	3.683281	0.00023046**
Vega Sicilia 2009	0.870633	3.717990	0.00020100**
Château Margaux 2004	0.872212	3.724734	0.00019570**
Vega Sicilia 2008	0.897957	3.834676	0.00012586**
Vega Sicilia 2011	0.850571	3.632315	0.00028113**
Vega Sicilia 2003	0.854087	3.647331	0.00026521**
Nosotros 2014	0.840403	3.588895	0.00033235**
Pago de Carraovejas 2018	0.828289	4.084297	0.00004427**
Silver Oak 2007	0.827969	4.082718	0.00004457**
Pago de Carraovejas 2009	0.819457	4.517625	0.00000627***
Caymus 2010	0.819157	4.515971	0.00000631***
Caymus 2012	0.808315	4.456195	0.00000836***
Vega Sicilia 2014	0.803720	3.432242	0.00059902*

Winery Aspect

- From the previous OLS results, we found that the vast majority of wineries cannot achieve stable ratings year after year, let alone those that have never made it to the rankings.
- Suppose you are the owner of one of these wineries, and you have no idea how to cater to consumer tastes.
  - Moreover, you cannot imitate previous successes because of the inherent differences caused by climate variations.

Bayes' Theorem

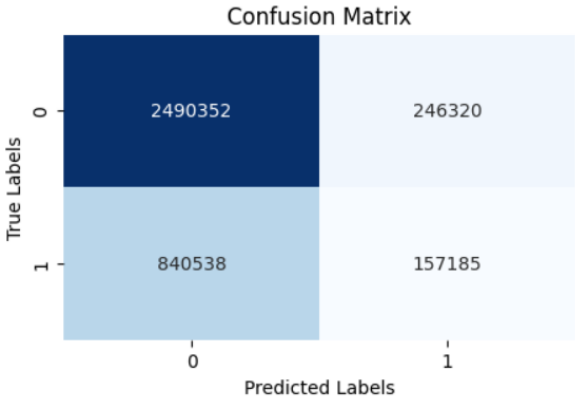
$$P(\text{outcome}|\text{data}) = \frac{P(\text{data}|\text{outcome}) \cdot P(\text{outcome})}{P(\text{data})}$$

- $P(\text{outcome}|\text{data})$: Posterior probability of the outcome given the data.
- $P(\text{data}|\text{outcome})$: Likelihood of the data given the outcome.
- $P(\text{outcome})$: Prior probability of the outcome.
- $P(\text{data})$: Prior probability of the predictor.

Naive Bayes Classifier

- $\hat{y} = \arg\max_{y_i \in Y} P(y_i | x_1, x_2, \ldots, x_n)$.
- $$\arg\max_{y_i \in Y} \frac{P(x_1, x_2, \ldots, x_n | y_i) \times P(y_i)}{P(x_1, x_2, \ldots, x_n)} = \arg\max_{y_i \in Y} \left(\prod_{k=1}^{n} P(x_k | y_i) \right) P(y_i).$$
- Features: 'Elaborate', 'Grapes', 'ABV', 'Body', 'Acidity'.
- Map ratings to 2 classes: 0 (Rating ≤ 1) and 1 (Rating ≥ 4).

Multinomial NB for Red Wine



- The results are not very satisfactory; we expected a higher trace value.

Classification Tree

- $\{(\mathbf{x}_i, y_i) : i = 1, \ldots, N\}, \quad \mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$.
- $n_L + n_R$ is the number of data in the previous node, a denotes one of the two classes of y.
- Seek the splitting variable j and split point s that minimize the sum of Gini impurity of the split leaves:

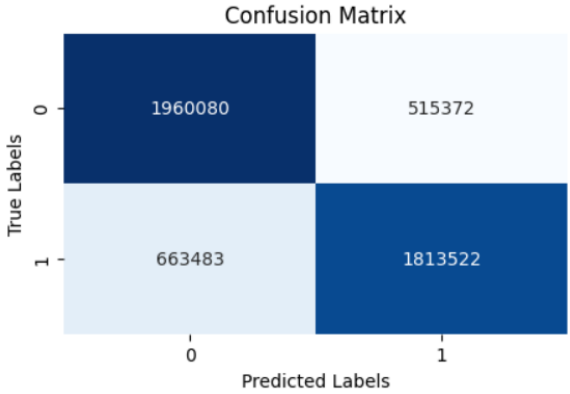
$$\min_{j,s} \left[1 - \left(\left(\frac{a_L}{n_L} \right)^2 + \left(\frac{n_L - a_L}{n_L} \right)^2 \right) + 1 - \left(\left(\frac{a_R}{n_R} \right)^2 + \left(\frac{n_R - a_R}{n_R} \right)^2 \right) \right]$$

RSCV to Find Hyperparameters

```
param_dist = {
    'criterion': ['gini', 'entropy'],
    'max_depth': randint(10, 30),
    'min_samples_split': randint(5, 20),
    'min_samples_leaf': randint(2, 10),
}

# Use precision as the criteria
random_search = RandomizedSearchCV(clf, param_distributions=param_dist,
n_iter=10, cv=5, scoring='precision')
random_search.fit(X, y)
```

Confusion Matrix of Tree



- Precision is 0.75 and 0.78.

Conclusion

- Code: `https://github.com/blossmuri/Grape-Wine-Rating-Inference`
- Readme: `https://muguet-de-mai.notion.site/X-wines-1b7a8220d5d547e3a315a12fa7454a85`
- Materials: Intel(R) Xeon(R) Gold 6226R$\times$2, NVIDIA RTX A6000$\times$2.