Text Mining Reviews for Wine Classification

Nicholas V. Nikolov

Department of Mathematics, Illinois State University, Normal, IL 61761

Ndnikol@ilstu.edu

**Abstract**

Text mining information from winemag.com was an experiment on how successfully a model can be built to classify eight different wines. This modeling project demonstrates the potential for a learning algorithm to understand the various qualities that distinguish different wines faster than a human. Text tokenization and ordering was used on wine reviews in order to develop the strongest predictive model possible. The outcome demonstrated certain limitations of random forest modeling as well as the strong potential of a binary adaptive boosting extension to multi-class prediction and the very popular deep learning artificial neural networks.

**Introduction**

The ability to distinguish unique flavors within various wines is a rare and often misunderstood talent outside of the world of professional wine production and reviewing. Many reviewers demonstrate a near unbelievable ability to discern various fruit flavors when they perform taste tests. If these reviews are, in fact, to be trusted then one should expect a statistical algorithm to be able to learn how to differentiate these various qualities and make useful predictions. An example, Cabernet Sauvignon acquires its characteristic dryness from the tannins that are common in this wine. Sommeliers will often taste these tannins and describe them as being “robust” or “strong.” When reading a review that describes strong tannins, a professional critic can immediately disregard sweet wines and may even be inclined to disregard most white wines which tend to be less tannic than their red counterparts. The research goal was to determine if an algorithm could be utilized to train a computer in much the same way that a professional sommelier was taught.

The dataset, in its most primitive form, came with ten variables: Country of origin for the wine, a description (what is referenced as review), a designation of where the grapes were grown, a points score (the data did not have values less than 80), the price of the bottle, the province from where it came (states in the United States), region describing more specifically where the wine was from, the variety (originally over 100 different classes), and the winery from where the wine had come.

The data was collected from Winemag.com and presented on Kaggle.com for further research. As such, every piece of information presented in this study exists online, and the dataset can have new values added to it as additional wines are bottled and reviewed.

**Results**

Initial Pre-processing

Initially, the dataset was narrowed. The naming criteria for European wines tends to include information about where the wine came from. This elevated the number of wine varieties into the hundreds and made the process of classification severely more difficult. As such, it was decided that the focus would be on US wines only. Further cleaning still had to be done as some special characters did not read in correctly. For example, some text that appeared was BergstrÃ¶m and RosÃ©. These were detected and filtered out to simplify the classification. Red blends were also omitted due to the difficulty for discerning what type of wine was used in the blend.

|  |  |
| --- | --- |
| **Wine Variety** | **Observations** |
| Cabernet Sauvignon | 7067 |
| Chardonnay | 6377 |
| Malbec | 503 |
| Merlot | 2190 |
| Pinot Gris | 787 |
| Pinot Noir | 9008 |
| Riesling | 1598 |
| Zinfandel | 2554 |

From the original dataset; designation, region, and winery were all omitted. This decision stems from the dimensionality caused by including these variables. Including these as predictors would have led to close to several hundred columns in the design matrix. As will be seen, even without this information, the model can be trained to predict exceptionally well.

The classification was ultimately constricted to an eight-factor response made up of Cabernet Sauvignon, Chardonnay, Malbec, Merlot, Pinot Gris, Pinot Noir, Riesling, and Zinfandel. The classes were imbalanced as seen to the right. This would inherently threaten the class sensitivity of wines like the Malbec and the Pinot Gris. In fact, in later experimentation, this problem became so severe that the models would begin to omit Malbecs altogether. This caused problems in the model evaluation process where confusion tables could not be made using simple R commands. These types of class imbalance issues are quite common and are often difficult to address but, as we shall see, properly selected and tuned models can overcome these hurdles.

Before beginning the research, it was necessary to extract the relevant information from the description variable which reviewed wines for their various qualities. For example, reviewers often discussed the tannic quality of the wine which affects how dry a wine is. Another piece of information that was important was to extract the various fruits that appeared in the wines. For example, Malbecs are frequently described as having a plum flavor while Pinot Noirs and Cabernet Sauvignons frequently boast flavors of cherry.

 The initial design was to tokenize individual words in the reviews (a process known as unigram tokenization). This essentially wrote out every review such that each word held a single position in a vector. The next step was to eliminate irrelevant words known as stop-words such as; if, and, or, but, etc. from this vector. This vector could then be searched for occurrence frequency and sorted in order of descending occurrence. This way one could study which words were used to describe the wines the most. This had to be further studied to extract only the words that could be used to describe a specific wine. For example, “bodied” could not describe whether a wine was full, medium, or light-bodied. Further, wine did not bring additional information to what and individual already knows to be a wine. On the other hand, dry and red could be used to narrow the studied wine. So only the relevant words were selected for further study.

Once relevant words were selected, a search was conducted within all wines to encounter instances of that word appearing. Simultaneously, a dummy vector was created to record instances of such words appearing. In any case where a word was detected in a review for a particular wine, the dummy vector for that wine would have a 1 while reviews that did not have the particular word would demonstrate a value of 0. In other words, every utilized word was converted into a dummy variable and these dummy variables were concatenated into an updated dataset. Thus, the updated dataset was variety, price, score, province, and the full set of dummy variables.

Initial Random Forest

The initial goal was to achieve 67% accuracy for overall prediction with no initial consideration for class errors. This random forest only utilized the unigram tokenization described above and ultimately contained 70 words for prediction. It was found found that continuously adding additional tokens led to consistent but diminishing improvement to the model accuracy, meaning models were tested with fewer words. By 70 words, an amply trained random forest was predicting with roughly 68% accuracy.

This is where the class imbalance issues truly resonated. No random forest model had an issue with completely omitting a class, but every tuned variation had instances of very high class error. Malbec and Pinot Gris, for example, had close to 90% misclassification. Pinot Noir and Cabernet Sauvignon tended to perform fairly well, frequently classified with less than 10% misclassification.

Further Tokenization

It waas determined that the diminishing returns of adding further unigram tokens was mainly due to the fact that additional words weren’t adding the necessary information to the model. Thus, 2-gram and 3-gram tokens were studied. These variables offered further information on the characteristics of the wine. For example, we could not distinguish between red and black cherries as well as the various bodies of wine. This list was also used to search for descriptions of tannin. E.g., reviewers described a wine as having, “strong tannins” or “robust tannins.” As discussed earlier, description of tannins is very important. In the dataset, almost 6,000 descriptions were made on the tannic quality of the wine. In the unigram model, only “tannic” could be included and so this strong descriptor had to be ommited. Adding more and more dummy variables led to a model with 126 predictors. This is where some serious issues began to arise. No matter how the random forest model was tuned, a strong prediction outcome could not be derived. Typically random forests are expected to be fairly strong out-of-the-box predictors but this consistent adding of tokens had led to a worsening of the prediction accuracy (now around 50%).

Multigram Random Forest

The overall prediction accuracy for the random forest model was now around 50%. The class error for Malbec skyrocketed to almost 100% (now in many instances a Malbec could be completely omitted from the predictions). The best class error was for the biggest class, Pinot Noir, at around 30% misclassification. Cutoff values were adjusted to attempt to classify more wines into the smaller classes but it was soon found that the improvement in class sensitivity generally came at the cost of the class specificity. In other words, improvements were marginal and came with a cost.

Multigram Bagging Trees

|  |  |
| --- | --- |
| **Wine Variety** | **Class Error** |
| Cabernet Sauvignon | 53.17% |
| Chardonnay | 47.83% |
| Malbec | 93.01% |
| Merlot | 86.93% |
| Pinot Gris | 79.61% |
| Pinot Noir | 39.30% |
| Riesling | 56.00% |
| Zinfandel | 81.88% |

An alternative method to consider was bagging in order to implement every variable for consideration in each tree. This actually worsened overall performance from 50% prediction accuracy to approximately 45% prediction accuracy. As see to the right, some of the peak class errors were slightly improved but at the cost of the predictive power in classes like Pinot Noir. It is also disappointing how high the class error was for Pinot Noir. Considering the prevalence of this wine in the dataset, it would have been expected to perform with far greater accuracy. What was worse is that the model classified many Chardonnays as Pinot Noir which is not a good sign of an adequate model. Ultimately these were not acceptable outcomes for a strong predictive model.

Interjection: Binary Consideration Using Adaptive Boosting

This section is titled *interjection* because this method wasn’t studied in detail due to the superiority of the artificial neural networks considered later in this paper (but earlier in the chronological process of developing a predictive model). The reason this method had been dismissed many times was exactly due to the severe class imbalance that had caused issues before. The concern was that something like Malbec would be completely overwhelmed (classifying everything as not Malbec) while the Pinot Noir would be overpredicted. However, the results were actually fairly impressive. The Malbec demonstrated 32.9% class misclassification error which is a considerable improvement over simple random forest and bagging techniques. Additionally, the class misclassification error for Pinot Noir was 24.82%. Overall sensitivity improved as well. If one considered all seven classes that were not classified as Malbecs as false negatives then the comparison would be a sensitivity of .2206 for the random forest and .3239 for the binary adaptive boosting method.

Both binary classification models were developed through adaptive boosting using 3000 trees and an interaction depth of 2. Additionally, a shrinkage parameter of 0.01 was introduced.

Artificial Neural Network

Artificial neural networks were finally considered using the h2o framework. All models utilized cross entropy as a loss function and several activation functions were considered within the layers. The main debate, so to speak, was between utilization of the rectifier function and the maxout function. Both activation functions performed admirably and the final decision was to use the maxout function shown below:

This activation function is very powerful because it can approximate any function. It is a very popular activation function in the field of speech recognition and it was hoped, by extension, that it would give good results in the text-mining domain. As it turns out, this function tended to outperform the rectifier function in most comparisons.

The method of determining the best model stemmed from cross-validation on a separate validation (something of an explicitly selected 2-fold cross validation). This had the effect of greatly speeding up the process of searching for and evaluating various combinations of layers for the deep learning architecture. All layer combinations, between 10 and 40 layers in a 2-layer network (the results tended to exceed that of a single-layer neural network), were evaluated. Each model architecture was evaluated using 20 epochs to avoid overfitting. What was found

|  |  |
| --- | --- |
| **Wine Variety** | **Class Error** |
| Cabernet Sauvignon | 30.18% (-22.99%) |
| Chardonnay | 13.75% (-34.08%) |
| Malbec | 17.02% (-75.99%) |
| Merlot | 33.74% (-53.19%) |
| Pinot Gris | 27.5% (-52.11%) |
| Pinot Noir | 19.30% (-20%) |
| Riesling | 13.50% (-42.5%) |
| Zinfandel | 32.65% (-49.23%) |

was that larger networks did not perform better than the smaller networks.

In fact, the strongest predictor turned out to be a neural network with two hidden layers of 12 and 10 neurons respectively. The improvement can be seen in the class misclassification rates to the right. The cost of the severe class imbalance that hindered model adequacy previously was significantly improved. Of additional importance is the fact that there is diminished variability between class errors. Even the first unigram random forest had some extremely high class errors. This variability has been significantly reduced allowing for better comprehension on how the model is actually performing. Additionally, a data analyst would not feel discomfort in presenting this model as truly being capable of predicting with 78% accuracy. Included below are sample outcomes for two randomly selected observations in the dataset. This shows the probabilities associated with the model prediction.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| True Wine | Cabernet Sauvignon | Chardonnay | Malbec | Merlot | Pinot Gris | Pinot Noir | Riesling | Zinfandel |
| Chardonnay | 0.0053 | 0.9630 | 0.0003 | 0.0013 | 0.0160 | 0.0010 | 0.0111 | 0.0018 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| True Wine | Cabernet Sauvignon | Chardonnay | Malbec | Merlot | Pinot Gris | Pinot Noir | Riesling | Zinfandel |
| Cabernet Sauvignon | 0.7763 | 0.0000 | 0.0058 | 0.2178 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

Conclusion

The process of multiclass classification is certainly challenging. As it turns out, eight class classification is now quite simple. Modern convolutional networks have demonstrated an ability to identify hundreds of items appearing on a live feed.

This project served as a proof of concept for the power of text mining. Many methods were explored, and it was found that the artificial neural network was the most sophisticated at dealing with multiclass classification and the attached class imbalance of the data used. The final model demonstrated was the feed-forward neural network which learned using stochastic gradient descent with backpropagation. This model had two hidden layers of 12 and 10 neurons respectively, both of which utilized the maxout activation function. The loss function utilized was cross entropy, most prevalent in the field of classification, and the model was trained over 20 epochs.

One of the big interests for further research would be to develop a formal method for conducting the binary adaptive boosting technique within the multiclass problem we are faced with. It would be ideal to develop this classifier in such a way that it can compare the outputted response probabilities and determine which classifier contains the highest value. For example, if the boosted trees classifier for Pinot Noir outputs a value of 0.88 for observation 1 and the classifiers for the other classes output values less than 0.6, then we could classify that observation as a Pinot Noir. This more manual technique may demonstrate significant predictive accuracy improvements over even the best artificial neural network that was observed. Additionally, comparing use of different activation functions within the same model would be of value (E.g., Keras with the TensorFlow backend allows one to construct each layer individually). This would allow a combination such as a layer activated using the rectifier function and a second layer utilizing the maxout function. It is suspected that this might give superior outputs to what was demonstrated by h2o.

Additional improvements to the model may be derived from incorporation of higher tokens such as 4-gram and even 5-gram tokens. This would warrant exploration though it is important to note that there were not a lot of repeating instances of 4-gram observations. However, it would still be worth exploring. Further, the neural network demonstrated a relative resiliency to the higher dimensionality of the design matrix which, at the conclusion of the text-mining portion of the model building, had 127 columns. If there are additional qualities of the wine that could be used to improve prediction, then it would be worth extracting those words and implementing them into the model as well.

Ultimately this project is seen as having been a significant success in demonstrating the power of various statistical algorithms as well as the power, prevalence, and importance of proper text-mining experience.