Classification of Wine Varieties in a High Dimensional Setting

Joseph Doan
Computer Science and Engineering
University of California, San Diego
jtd020@ucsd.edu

Rebecca Kreitinger
Computer Science and Engineering
University of California, San Diego
rkreitin@eng.ucsd.edu

William Hogan
Computer Science and Engineering
University of California, San Diego
whogan@eng.ucsd.edu

Christopher Liu
Computer Science and Engineering
University of California, San Diego
cmliu@ucsd.edu

***Abstract*—In this work, we train a transformer to predict wine varieties based on a combination of textual features. We experiment with various combination of features and three different loss functions: negative log-likelihood, KL-divergence, and a custom LASSO loss function. We attain best performance by training our model using a negative log likelihood loss function on input features tokenized and concatenated together.**

***Index Terms*—deep-learning, classification, cross-entropy loss, Lasso loss function, KL-Divergence loss**

I. Introduction

Wine tasting is a skill to evaluate and identity a wine based on sensory information alone. A sommelier is a wine expert who is able to taste the nuances between different wines. It is known to be a difficult task that requires years of dedication to be able to accurately identity and categorize wine. Our goal is to create a learning algorithm that performs classification on the variety of wines based on the written description given by a sommelier. Producing this classifier will assist sommeliers in determining information about specific wines. Additionally, this could be a tool used in the wine industry for wine tastings and amateur sommeliers who want to improve their abilities. As part of our project, we wish to provide insight on two questions:

1) **Which features are ideal for training a text-based multi-classifier?** The four feature groupings we will be testing are tasting descriptions only, all features (region, winery, tasting descriptions, etc.), all features with stop-words removed, and all features except tasting descriptions. By testing different feature groups, we can determine the best way to utilize the data for our classifier.
2) **Which loss function performs the best for this multi-classifier problem?** We experimented with three different loss functions: negative log-likelihood, KL-divergence, and a custom LASSO loss function. Our goal is to optimize our performance by finding which loss function is ideal for our classifier.

Section II highlights similar work pertaining to classification of wine as well as work within exploring loss functions. Next, **Section III** gives an overview of the dataset used from the online database community Kaggle[1] and how we prepared the data for our model. **Section IV** describes in detail how our model is constructed as well as the loss functions implemented for our experimentation. For each loss function the primal and dual are provided. Next, **Section V** highlights our experiments and methodology to generate our results for comparing feature decisions and loss functions. We conclude in **Section VI** with our results and findings along with our analysis.

II. Related Work

There are many other projects that have tried to classify wine on different datasets. A dataset we heavily considered using rather than our current dataset was the UCI Machine Learning Repository Wine Data Set which provides information on three classes of wine based on the wine's chemical attributes [1]. The difference between the UCI dataset and the one used in this project is that all 13 features of the UCI dataset are numerical whereas the dataset we use for our problem, as discussed in Section III, is primarily textual. Additionally, evaluations on loss functions for text categorization and measuring performance for various loss functions have been explored.

A. Physiochemical Wine Classification

Though the UCI dataset is primarily numerical, there are a number of published papers that have attempted to use the data set with other machine learning approaches and feature selection techniques in order to classify wine. However, more specifically, the UCI wine dataset provides physiochemical features, and is typically classified by a "quality score" as the dataset only contains red and white variants of a single variety of wine ("Vinho Verde"). A number of the literature we looked through used very straightforward ML approaches, such as Support Vector Machines, k-Nearest Neighbors, and Random Forests. Most of the works showed that Random

[1] https://www.kaggle.com/

Forests seemed to have the best performance, though some had different accuracy evaluations compared to each other, primarily due to having different buckets for classes in terms of wine quality [2][3][4]. There was another paper relating to graph neural networks, which seemed to have a high accuracy, upwards of 98%, but the focus of that paper was related to testing graph neural networks in general rather than the wine dataset [5].

B. Other Works Using Same Dataset

Nevertheless, there are also other papers that use the same dataset as we chose [6]. One such example paper uses Long Short-Term Memory models for multiple categories, comparing it with a base model of Naive Bayes [7]. The objective of their report contains not only classification, but also generation. However, in terms of classification, the report used only the description, training their models in order to predict 5 of the 11 specific categories. Their models had a resultant accuracy that was around 75% for some categories, with 95% for the country category, but an average of 38% for the points category; average accuracy across all categories being around 54.3%. The generative model, on the other hand, does the reverse functionality of our report's, by generating a wine's description based on the other features of the entries. Their LSTM model results in an average test accuracy of 33.5%, with a test description perplexity of 51.851, though with incoherent output. Thus, through our report, we may discover what features of the dataset best contribute to classifying wine variety, and also see the weights of the features in terms of their correlation to the wine variety feature.

C. Text Analysis/Loss Functions

In terms of methodology of text analysis and which loss functions to attempt, we used a few papers to give insight. For text categorization, we used a paper that does an analysis on loss functions for classification using text categorization [8]. Some of the classifiers analyzed were SVM, linear regression, logistic regression, neural networks, Rocchio-style, Prototypes, kNN, and Naive Bayes. The results show that linear regression, logistic regression, and neural networks performed the best, which we will later take advantage of in our models. In terms of loss functions specifically, we referenced a paper that analyzes many loss functions in order to probabilistically determine whether or not they have different convergence rates [9]. Their report concluded that hinge and logistic loss performed significantly better than classic square loss. Thus, we will later use these loss functions in our report.

III. Dataset

For our dataset, we used the publicly available "Wine Reviews" dataset from Kaggle [10]. It features 130k wine reviews scraped from WineEnthusiast, a website containing the largest periodical for wine and spirits [11]. The wine reviews each contain 11 features: country, description, designation, points, price, province, region 1, region 2, taster name, taster Twitter handle, title, and winery. Table I represents an example of the major features. The wine description is the primary feature our model relies on. Wine descriptions have been written by professional sommeliers and have a mean length of 40.38 words with a standard deviation of 11.11. Each wine review is labeled with a wine variety which is the variable we train our model to predict. There are 708 unique wine varieties in the dataset.

country	price	province	variety	winery
Portugal	15	Douro	Portuguese Red	Quinta dos Avidagos

description
This is ripe and fruity, a wine that is smooth while still structured. Firm tannins are filled out with juicy red berry fruits and freshened with acidity. It's already drinkable, although it will certainly be better from 2016.

TABLE I
Example entry from dataset

We randomly assign 80% of the 130k wine reviews to create a training set. The remaining data is split to construct a development set (10%) and testing set (10%).

IV. Model

Given that our task is to classify data that primarily contains textual features, we chose to leverage a transformer for our deep-learning model. Transformers are very effective in many natural language processing (NLP) classification applications [12]. It solely relies on attention mechanisms to relate different positions of a text sequence in order to compute a representation of the sequence. Moreover, pre-trained transformers such as BERT [13] have attained state-of-the-art performance on numerous different natural language processing tasks. For our model, we fine-tuned a BERT model to classify wine varieties. We used the BERT base model which features 12 stacked layers of encoders and decoders, 12 attention heads, and a hidden unit dimension of 768 resulting in 110M learnable parameters. For a complete list of our hyperparameter settings, please see the project repository at https://github.com/wphogan/wine_classification.

In order to experiment with different types of models, we decided to focus on changing the loss function. This allowed us to adjust our model so that we could achieve a higher accuracy score on our test data.

The first loss function model that we used was based off the *cross entropy* function. This is the loss function that was used with the base BERT model. Here, we used the maximum entropy problem as this would give us the minimum value for cross entropy [14].

The primal of this problem was formulated as:

$$\begin{aligned} \min \quad & f_0(x) = \sum_{i=1}^{n} \quad x_i log(x_i), \quad x \in R^n_{++} \\ \text{s.t.} \quad & Ax \preceq b \\ & 1^T x = 1 \end{aligned} \tag{1}$$

Since we know the conjugate of $f_0(x)$ is $f_0^*(y) = \sum_{i=1}^n e^{y_i - 1}, \quad y \in R$, we can formulate the dual function as such:

$$g(\lambda, v) = -b^T\lambda - v - \sum_{i=1}^n e^{-a_i^T \lambda - v - 1}$$

$$= -b^T\lambda - v - e^{-v-1} \sum_{i=1}^n e^{-a_i^T}$$

where a_i is the ith column of A. From here, we set our KTT conditions to be:

$$\begin{aligned} Ax &\preceq b \\ 1^T x &= 1 \\ -b^T \lambda &= 0 \\ v &= log(\sum_{i=1}^n e^{-a_i^T \lambda} - 1) \end{aligned} \tag{2}$$

To maximize $g(\lambda, v)$, we set $v = log(\sum_{i=1}^n e^{-a_i^T \lambda} - 1)$. Therefore the dual problem is:

$$\begin{aligned} \max \quad & -b^T\lambda - log(\sum_{i=1}^n e^{-a_i^T \lambda}) \\ \text{s.t.} \quad & \lambda \succeq 0 \end{aligned} \tag{3}$$

The second type of loss function that we experimented with was *KL divergence*. Similar to cross entropy, the goal was to try and minimize the KL divergence formula, which is given by:

$$KL(P, Q) = \sum_{i=1}^n \quad p_i log(p_i / q_i) \tag{4}$$

Since the goals of cross entropy and KL divergence is to try and find the value Q such that Q = P, this is the same as the maximum entropy problem that we described earlier [14]. As such, we used the same primal/dual problem to solve this.

The final loss function model that we decided to test out was to add a regularization parameter to our cross entropy function. Similar to the *LASSO* problem [15], the goal of this regularization parameter was to add a penalty that would penalize more dense features from our input data. This way, our solution, while potentially performing worse on our training data, would have a less likely chance of overfitting. We formulated this function by taking the norm-1 of our weights/targets and multiplying a regularization parameter λ. This was then added to the cross entropy function.

The primal of this problem was formulated as:

$$\begin{aligned} \min \quad & L_E + \lambda ||x||_1 \\ \text{s.t.} \quad & L_E = \sum_{i=1}^n \quad x_i log(x_i), \quad x \in R_{++}^n \\ & Ax \preceq b \\ & 1^T x = 1 \\ & \lambda \succeq 0 \end{aligned} \tag{5}$$

Following our previous steps for cross entropy, we reformulated this to as the following dual problem:

$$\begin{aligned} \max \quad & -b^T v - log(\sum_{i=1}^n e^{-a_i^T v - \lambda}) \\ \text{s.t.} \quad & v \succeq 0 \\ & ||A^T x||_\infty \leq \lambda \end{aligned} \tag{6}$$

where the condition for $||A^T x||_\infty$ was derived from the dual norm of $||x||_1$ [16].

The goal of this dual problem was to choose what would be the best regularization parameter λ to use. In order to save time and computational power, we solved this by testing out various values of λ from 0 to 1 and then choosing the range that performed the best.

V. Experiments

A. Feature Experiments

We conducted four experiments with different features in order to discover the combination of features that resulted in the best model accuracy. For all the experiments, we used a negative log-likelihood loss with a softmax classification layer.

Experiment 1—Wine descriptions only: In the first experiment, we only use the wine description to predict the wine variety. We tokenized each description using a pre-trained BERT sub-word tokenizer. We consider this our baseline experiment and compare all our other experiments to it. This experiment will reveal if wine descriptions alone contain enough unique and information-rich data for the model to accurately predict the corresponding wine variety.

Experiment 2—All features: For the second experiment, we took the tokenized wine description and concatenated all other features available in the dataset to it. The additional features were tokenized and pre-pended to the wine description. This experiment will shed light on the importance of the additional features in the dataset (e.g. wine region, winery, etc.).

Experiment 3—All features, stopwords removed: The third experiment was similar to the second experiment however English stopwords were removed from the wine descriptions using the NLTK English library [17]. For this experiment, we seek to determine if model performance is improved by removing words with low information density.

Experiment 4—Without wine descriptions: The fourth experiment consisted of all the features except for the wine descriptions. We consider this an ablation experiment that will help us determine how important the wine description is to the model's predictive performance.

B. Loss Function Experiments

We also conducted experiments with different loss functions, namely negative log-likelihood, KL divergence, and a custom implementation of a LASSO loss function. As detailed in Equation 5, the LASSO loss function depends on a pre-set value for lambda. We empirically determine the value of lambda that results in the best model performance by conducting a series of experiments with a range of lambda values. We evaluate the following values for lambda: 0.1, 0.2, 0.3, 0.4, and 0.5. We present the results of these experiments in Section VI.

VI. Results and Discussion

The results of our feature experiments are found in Table II. Here, we see that our fine-tuned BERT model only reaches 63.3% accuracy when trained and tested exclusively on wine descriptions. This tells us that wine descriptions from professional sommeliers are not sufficiently unique and information-rich for the BERT model to learn to differentiate between the 708 wine varieties. The accuracy of the model increases greatly when we train the model on additional features, jumping from 63.3% to 96.6% accuracy. This highlights the importance of the wine's country, region, winery, and the other features in determining the wine variety. Interestingly, the removal of stopwords from the wine descriptions slightly hinders performance. The no-stopword experiment achieved an accuracy of 96.0%, whereas the "All features" experiment which featured unaltered wine descriptions, achieved 96.6% accuracy. We suspect this may be due to the fact that BERT is pre-trained on a large body of text that contains stopwords and, therefore, the model performs best when fine-tuned on text that also contain stopwords. Lastly, we notice that removing the wine descriptions from the training decreases the model's performance. This illustrates the importance of wine descriptions to informing the model's predictions.

TABLE II
Results from feature experiments using BERT model and a negative log likelihood loss.

Input Features	Accuracy
Wine descriptions only	0.633
All features	**0.966**
All features, stop words removed	0.960
Without wine descriptions	0.840

In Figure 1, we compare the model's accuracy using various values of lambda for the LASSO loss function. We observe that $\lambda = 0.2$ results in a model accuracy of 96.2% which is the best model performance of the models trained using the LASSO loss function.

Table III contains the results from our experiments with various loss functions. In each experiment, we concatenated and tokenized all of the textual features to train and test each model. Of the loss functions, the negative log likelihood resulted in the best performance, followed by our custom implementation of the LASSO loss function with $\lambda = 0.2$. Finally, the KL divergence loss attains the lowest performance of the three loss functions. With this dataset, we conclude that cross entropy and LASSO loss functions are better suited for training a model to predict the large number of wine varieties.

Fig. 1. Graph comparing model accuracy using a LASSO loss function with various values of lambda.

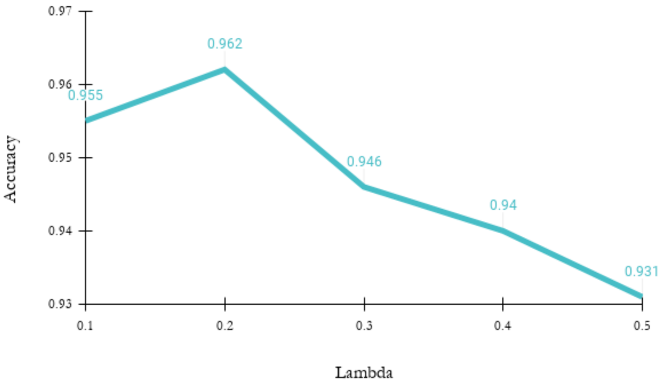

TABLE III
Results from loss experiments using BERT model.

Loss Function	Accuracy
Neg. Log Likelihood	**0.966**
KL Divergence	0.929
Lasso (λ = 0.2)	0.962

VII. Conclusion

In this work, we looked at what features related to wine reviews would help us best determine variety of a certain wine, as well as which loss functions are most optimal in optimizing the performance of such a text-based multi-classifier.

From the Feature Experiments, we concluded that a model trained on unaltered wine descriptions and concatenated extra features most accurately classified an entry into the correct wine variety. We also did analysis of wine descriptions only, and without the wine descriptions, discovering that the other features, rather than the wine descriptions were most effective in determining wine variety. For other wine classification problems, it is likely that the empirical values related to the wine can be expected to be heavily correlated when correctly classifying wine variety. Though it should be noted that the reviewer's descriptions still significantly affected accuracy, reaching 96.6% with versus 84.0% without.

From the Loss Function Experiments, we concluded that negative log likelihood and LASSO loss functions were most effective in our model. We were unable on which parts of our experiments contributed most to the differences in accuracy. However, we can assume that another text-based multi-classifier like this one is most effective using a negative log-likelihood loss function, as ours was.

A. Future Work

Though wine reviews seems to be a very niche topic and a very specific use case, we could extend our same suppositions about our model to another dataset with similar features and possible objective. Another dataset with many different varieties and some text based description could be used to train our model, and we would suspect similar results. One example of such a dataset could be product reviews, and classifying categories of items.

Acknowledgments

Thank you to Professor Cheng and teaching assistants Chester Holtz, Po-Ya Hsu, James Liu for all their guidance that made this project possible.

References

[1] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.

[2] Yeşim Er and Ayten Atasoy. “The Classification of White Wine and Red Wine According to Their Physicochemical Qualities”. In: *International Journal of Intelligent Systems and Applications in Engineering* 4 (Dec. 2016), pp. 23–23. DOI: 10.18201/ijisae.265954.

[3] Anurag Sinha and Atul kumar. “Wine Quality and Taste Classification Using Machine Learning Model”. In: *International Journal of Innovative Research in Applied Sciences and Engineering* 4 (Oct. 2020), pp. 715–721. DOI: 10.29027/IJIRASE.v4.i4.2020.715-721.

[4] Y. Pristyanto, N. A. Setiawan, and I. Ardiyanto. “Hybrid resampling to handle imbalanced class on classification of student performance in classroom”. In: *2017 1st International Conference on Informatics and Computational Sciences (ICICoS)*. 2017, pp. 207–212. DOI: 10.1109/ICICOS.2017.8276363.

[5] Xinhan Di et al. “Neighborhood Enlargement in Graph Neural Networks”. In: *CoRR* abs/1905.08509 (2019). arXiv: 1905.08509. URL: http://arxiv.org/abs/1905.08509.

[6] António Carloto. *WINE QUALITY RATINGS VERSUS PRICE IN THE WINE ENTHUSIAST MAGZINE*. Oct. 2017. DOI: 10.13140/RG.2.2.26160.05123.

[7] Frederick Robson and Loren Amdahl-Culleton. “Classy Classification : Classifying and Generating Expert Wine Review”. In: 2018.

[8] Fan Li and Yiming Yang. *A Loss Function Analysis for Classification Methods in Text Categorization*. 2003.

[9] Lorenzo Rosasco et al. “Are loss functions all the same?” In: *Neural computation* 16.5 (2004), pp. 1063–1076.

[10] Zack Thoutt. *Wine Reviews Dataset*. 2017. URL: https://www.kaggle.com/zynicide/wine-reviews.

[11] *Wine Enthusiast Magazine: Wine Ratings, Wine News, Recipe Pairings*. Mar. 2021. URL: https://www.winemag.com/.

[12] Ashish Vaswani et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 5998–6008. URL: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

[13] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://www.aclweb.org/anthology/N19-1423.

[14] Wikipedia. *Kullback–Leibler Divergence: Principle of minimum discrimination information*. URL: https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence#Principle_of_minimum_discrimination_information.

[15] Wikipedia. *LASSO*. URL: https://en.wikipedia.org/wiki/Lasso_(statistics).

[16] Wikipedia. *Dual Norm*. URL: https://en.wikipedia.org/wiki/Dual_norm.

[17] Steven Bird. “Nltk: The natural language toolkit”. In: *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*. 2002.