How Good is Your Computer's Nose? Using A.I./M.L. Techniques with Expert Wine Reviews to Predict a Wine's Characteristics*

Jeff T. Sheng[1], Andy Tsao,[2] and Vincent Sheu[3]

Abstract— **In this paper, we examine the the performance of several machine learning models in the prediction of wine characteristics, based on sommelier-produced wine reviews. We find that simple models that take sentiment score or word frequencies as features performs well when predicting wine rating. However, these baseline methods deteriorate when we move to predicting characteristics with a more complicated relationship such as grape varietal or price. In this setting, we find that recurrent neural networks such as GRU or LSTM networks are much better at capturing the nonlinearities within the data.**

I. INTRODUCTION & TASK DEFINITION

How well can artificial intelligence and machine learning techniques predict characteristics of what a human has described, given only a short description written by an expert? In particular, for prediction tasks where there is less of a linear relationship in the data, how much better are machine learning techniques using non-linearities such as RNNs compared to more traditional linear predictors? To test this, we use short wine reviews written by sommeliers to examine how well a computer can accurately predict a wine's rating, price, varietal (grape), and country of origin, using different AI and ML techniques such as Naive Bayes, TFIDF word counts and word frequencies, sentiment analysis, and variations of a recurrent neural nets (including GRUs and LSTMs).

A. Scope and Evaluation

We chose wine reviews because they are highly descriptive, yet also short and often limited by a set word length. In a few sentences, a sommelier must try to convey enough information about the wine they are describing that both a connoisseur of wines and a non-expert can find the wine review useful. Moreover, wine consumers often want to know more than just how the wine tastes; for example, they may care about what kind of grape the wine comes from, and perhaps even its country of origin. Other important pieces of information such as price and rating can also be incorporated as part of the formulation of a wine's description, since the sommelier often has access to these data points when crafting their review.

Because of this density of textual information, wine reviews are apt for general testing of different linear and nonlinear text classifiers across a variety of different outcomes. Which models might be better at what prediction tasks? Our evaluation metrics are two-fold: one is accuracy for each individual prediction task, and one is general accuracy across the various characteristics we seek to predict: rating, price, country of origin and grape varietal.

To get a better understanding of the scope and task, we can look at some sample reviews to formulate a general hypothesis of what models might be more accurate than others:

1) Example Reviews:

- **Positive:** "Lush, plush, and absolutely delicious, this shows a range of softly approachable flavors including blackberry, cherry, currant, anise, cocoa, mineral, buttery, oak, and spice. The tannins are thick but finely ground. It's firmly set in the modern cult style and may not be an ager but it sure impresses now." Review for Constant 2009 Cabernet Sauvignon, Points: 94, Price: $130, Napa Valley, California, United States.
- **Negative:** "A rough, pinchy, nose with leather and funk is no way to begin. The palate stays hard and spiky with burnt, rustic, murky flavors. Roasted and mossy on the finish and not very good overall." Review for Cruz Alta 2007 Grand Reserve Malbec Mendoza, Points: 80, Price: $20, Mendoza, Mendoza Province, Argentina.

What we can see here is that there are key differences in these reviews in terms of sentiment, and possibly on price, if it turns out that price is strongly correlated with sentiment. We also see that there are unique words such as 'cocoa,' 'rustic,' and 'burnt.' How strongly correlated are these words with a wine's origin and grape variety? While accuracy per prediction task is important, we are also interested in predictions across all tasks. Sentiment analysis, for example, might be highly accurate in predicting rating, but very poor for country of origin.

II. INFRASTRUCTURE & LITERATURE REVIEW

A. Data

We derived our data from a publicly-available dataset courtesy of Zach Thoutt, posted on Kaggle.com. The data is a scraping of the Wine Enthusiast website on November 22, 2017. Each wine entry has the following fields:

*This work was not supported by any organization

[1] J. Sheng is with the Department of Sociology, Building 120, Stanford University, California, USA, and the Department of Computer Science, Gates Hall, Stanford University, California, USA. `jtsheng at stanford.edu`

[2] A. Tsao is with the Department of Statistics, Sequoia Hall, Stanford University, California, USA. `andytsao at stanford.edu`

[3] V. Sheu is with the Computer Science Department, Gates Hall, Stanford University, California, USA, and with the School of Law, Crown Quadrangle, Stanford University, California, USA. `vsheu at stanford.edu`

- **Points:** the number of points Wine Enthusiast rated the wine on a scale of 1-100. The dataset only contains wines rated 80 and above.
- **Title:** Wine review title (typically includes vintage)
- **Variety:** Type of grapes used to make the wine (i.e. Pinot Noir)
- **Description:** A few sentences from a sommelier describing the wine's taste, smell, look, feel, etc.
- **Country:** Country that the wine is from
- **Province:** Province or state that the wine is from
- **Region 1:** Wine growing area in a province or state (i.e. Napa)
- **Region 2:** Subset of Region 1 (i.e. Rutherford inside the Napa Valley) (sometimes blank)
- **Winery:** Winery that made the wine
- **Designation:** Vineyard within the winery where the grapes that made the wine are from
- **Price:** Cost per bottle
- **Taster Name:** Name of the sommelier who tasted and reviewed the wine
- **Taster Twitter Handle:** Twitter handle of the taster

From the original dataset, we cleaned by removing wines (with varietals) that only had a small number of data points (the 10% rarest wines), as the small number of reviews for rare wines would likely produce models with low training error and high test error, from predictors being overfitted.

In addition, we also manually reviewed and sanitized the datasete.g. date ranges with were replaced with a hyphen (-) (i.e. 20172019 was replaced with 2017-2019). Finally, we also removed wines with missing fields; e.g. any wines without price data were removed from the dataset. This produced a final dataset size of approximately 100,343 entries.

III. Approach: Models, Methods & Literature Review

Here, we describe the various models and methods that we use in our approach to the problem; specifically: Naive Bayes, TFIDF word counts and word frequencies, sentiment analysis, and three different variations of a recurrent neural net, which include a bidirectional RNN, GRU and LSTM.

A. Oracle: Labeled Data

Our oracle for all parts is the information contained in each review. Within each review, each wines score, description, and price are all human-generated (the first two by a sommelier; the latter by the winery that produced the wine). The other fields in each review represent the correct answere.g. the grape varietal used to produce the wine, the winery/designation, and location the wine was made at are all objective truths. With each model, we strive to meet these oracles (e.g. predicting a score for a wine and comparing it to the actual score).

B. Naive Bayes: A Simple Baseline

Under the strong assumption that word frequencies in a document are independent, we can model each document in a conditional probability framework. For a given class C of documents (for instance, all wine reviews of Pinot Noir) and an unlabeled document d, we compute the conditional probability $\mathbb{P}(C \mid d)$ using Bayes' rule:

$$\mathbb{P}(C \mid d) \propto \mathbb{P}(d \mid C)\pi(C).$$

Given a training set of labeled wine reviews, the prior probability $\pi(C)$ is simply the fraction of data points with the label C. To compute the conditional probability $\mathbb{P}(d \mid C)$, we gather the corpus S_C of all reviews with label C and construct an empirical distribution of words. Each word w that appears in S_C has associated to it a probability $p_w^{(C)}$, so that $\sum_{w \in S_C} p_w^{(C)} = 1$.

Under the assumption of word independence, we get that

$$\mathbb{P}(C \mid d) \propto \prod_{w \in d} \left(p_w^{(C)}\right)^{n_w} \pi(C),$$

where n_w is the number of occurrences of w in the document d.

Finally, we classify each document by the group that gives the highest conditional probability:

$$\hat{C}(w) = \arg\max_C \mathbb{P}(C \mid d).$$

The Naive Bayes approach tends to work well when there are specific words that strongly correlate to certain classes. Since many wines tend to be described using the same terms (e.g. acidic, fruity, tannic), we don't expect Naive Bayes to perform well when predicting grape varietal. When predicting rating, we expect Naive Bayes to be a worse version of sentiment analysis, since it can possibly pick out the extremely positive or negative words. However, unlike most sentiment analysis algorithms, Naive Bayes fails to capture the relationship between words since it assumes word independence in a document.

C. Word Counts and Word Frequencies

A *term frequency-inverse document frequency* (TFIDF) classifier is a modification to the standard bag-of-words approach to modeling word frequencies in documents. Words are represented as vectors in a high-dimensional space, where each dimension represents a word in the corpus of training documents.

The values of the vector $\vec{d} = (d^{(1)}, \ldots, d^{(|F|)})$ are calculated as a combination of *term frequency* $TF(w, d)$, which is given by the number of occurrences of w in document d, and *document frequency* $DF(w)$, the number of documents that contain the word w. More precisely, we have

$$d^{(i)} = TF(w_i, d) \cdot IDF(w_i) = TF(w_i, d) \cdot \log\left(\frac{|D|}{DF(w)}\right).$$

Intuitively, weighting by inverse document frequency downweights words that appear in many documents, such as common stop words or, in our case, words that pertain to many different categories of wine.

TFIDF-weighted vectors provide a baseline from which we can run several different types of models. In [1], Joachims describes a clustering algorithm where each new document the distance metric between two documents d_1 and d_2 is given by the dot product $\langle d_1, d_2 \rangle$. In our analysis, we chose to run a logistic regression with the TFIDF-weighted vectors as features. However, each document only contains a small number of words relative to the entire corpus, which means that our feature matrix is sparse. We therefore chose do reduce the dimensionality of our problem via an SVD. The results we present in Section V. use logistic regression with 500-dimensional features.

D. Sentiment Analysis and Predictors

1) Literature Review: In the course, we implemented unigram- and bigram-based sentiment analysis techniques, trained on limited, genre-specific labeled data. Our initial literature review explored additional possibilities to implement. Given that sentiment analysis is a growing field, with a multitude of papers published each year, we searched for surveys published recently (2017). Our search, in [2] and [3], turned up the following approaches:

1) Important Notions: Common to all methods.
   - Subjectivity/Objectivity: Dividing text into subjective (which contains sentiment) and objective (which contains facts) subsets.
   - Polarity: Classifying text as positive, negative, or neutral in sentiment.
   - Sentiment Level: Analyzing at the word, phrase, sentence, or document level.
2) Approaches: Two main categories.
   - Subjective lexicon: Collections of words labeled with sentiment scores; aggregation of scores produces a phrase, sentence, or document-level score. Examples include dictionary-based and corpus-based approaches, the latter of which can be further broken down into statistical (words that appear in positive/negative texts are more likely to be positive/negative) and semantic (words similar to other positive/negative scores are more likely to be positive/negative) approaches.
   - Machine learning: Supervised, including probabilistic classifiers, linear classifiers, decision tree classifiers, and rules-based classifiers.

While domain-specific dictionary-based training (to account for terms specific to the wine industry) could outperform a machine-learning based approach, such a model would not be easily generalizable. Thus, given our labeled dataset, we assumed a supervised machine-learning approach would be the best choice. Instead of implementing techniques separately, we used Google Cloud's platform, under the informed assumption ([4]) that a state-of-the-art sentiment analysis engine would perform at least as well as both the unigram/bigram models from previous work, and most combinations of techniques we could implement. In addition, this gave us scalability for larger data sets.

2) Model: For this model, we experimented with sentiment and quantitative metrics as predictors. When wine shopping, customers often use price as a proxy for quality (here, represented by score). We therefore started with price as a linear predictor for a wine's quality (represented by a wines score, or points).

Next, we attempted to predict wine score characteristics from wine reviews via sentiment analysis. Each wine review contains a description of the wine, written by a master sommelier reviewing the wine. Using Googles Cloud platform, we built infrastructure to conduct sentiment analysis on wine reviews, generating a sentiment score for each review (ranging from [-1, 1] (inclusive)), to be used as a predictor. After plotting various combinations of factors as predictors, we settled on sentiment, price, and sentiment + price as predictors of score; and sentiment as predictors of price, country, and grape varietal.

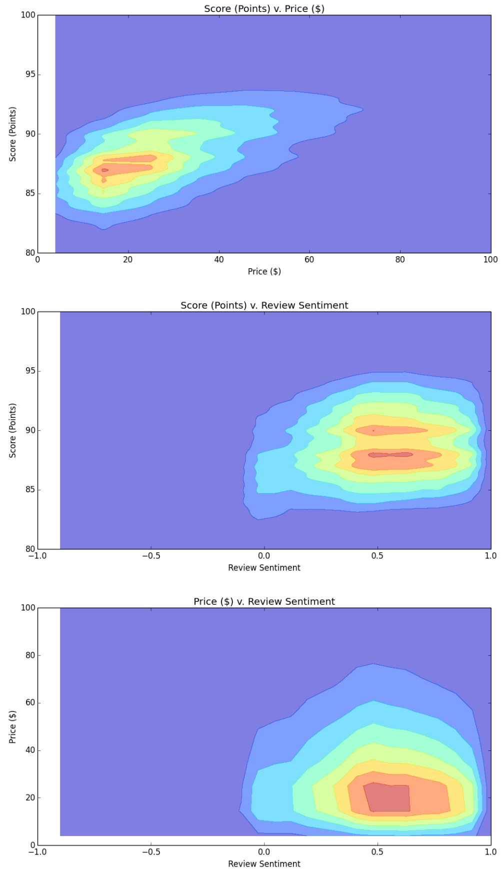


Fig. 1: Wine Score (Points) v. Price ($), Score v. Sentiment, and Price v. Sentiment. n = 100,377. Colours closer to red on the spectrum indicate more data points. Scores range from [80, 100], inclusive; review description sentiment ranges from [-1, 1], inclusive. We omit displaying results for wine priced outside of [0,100] (inclusive), as outside of a few outliers (e.g. one wine priced at $3,300), the significant part of this heatmap is displayed.

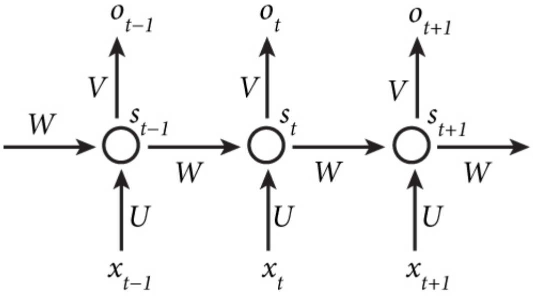
Finally, we trained both linear and multinomial logistic models on a subset of the dataset, tested on the remainder of the dataset, and compared models via their mean squared test errors.

Intuitively, we expect that sentiment would be a reasonable method for predicting wine score and price. A wine's points/score is meant to be a direct indicator of its quality, and quality is a direct causative factor of how a wine made the reviewer feel. Of all of the data we have available, that feeling is likely to be conveyed through the sentiment of the review. Economics also suggests that people are willing to pay more for better wine; thus, we also expect a strong correlation (and predictive relationship) between wine price and score/points.

However, we also expect that sentiment would be a poor predictor of wine country of origin and grape varietal. The opposite result would suggest, roughly, that a particular country's wines are mostly going to be reviewed positively/negatively (not wholly unreasonable given that certain regions are "known" as wine-producing regions, but still implausible), or that most wines of a certain varietal (produced from a certain grape) are of higher/lower quality (and thus more likely to produce positive/lower sentiment reviews).

E. Deep Learning with RNNs, GRUs, and LSTMs

We chose to use neural networks to model nonlinearities in our data. Traditionally, a neural network assumes that all inputs and outputs are independent of each other. This is certainly not the case for language-based prediction tasks, as stated in our overview of the Naive Bayes algorithm.



We ran various experiments using variations of a one-layer recurrent neural network, as shown in the above figure. We first represent each word using 200-dimensional GloVe word vectors [5] and feed these vectors into the neural network one at a time. The RNN has three trainable matrices U, V, and W, input nodes $x_1, \ldots, x_T$, hidden nodes $s_1, \ldots, s_T$, and output nodes $o_1, \ldots, o_T$, where T is the number of words in the document. Given x_t and s_{t-1}, o_t is computed as

$$o_t = V \tanh(Ux_t + Ws_{t-1}).$$

The final output node o_T is then normalized as a probability vector over the different categories, via the softmax function. The training parameters are then updated by backpropagation using the gradient of the cross-entropy loss. An Adam Optimizer was used, and ten epochs of training were run – each going through all entries in the training set, in a random order, in batches of size 32. The intuition for using a recurrent network is that o_T contains some information about $x_1, \ldots, x_{T-1}$, whereas our simpler bag-of-words models assumed independence between words.

While RNNs capture relationships between words in a document, it does so in a very rigid way. That is, the final output vector o_T is most influenced by the word x_T and least influenced by x_1. This can cause problems in lengthier documents that have important words near the beginning. We remedy this by using gated recurrent unit (GRU) [7] and long short-term memory (LSTM) networks [6]. Moreover, our final models all used a bidirectional implementation to better allowing us to incorporate context before and after a given word when making our predictions. These models have additional trainable parameters and were developed to be less sensitive to gap length between important terms.

IV. APPROACH: RESULTS AND ANALYSIS

We first present our evaluation criteria and overall results, comparing the accuracies of each of our models and then analyze them with more detail to understand why certain models performed better than others for certain tasks.

A. Predicting Rating

TABLE I: Predicting Rating from Sommelier Reviews

Model	Mean Squared Error
Naive Bayes	13.88
TFIDF	3.55
Price	7.87
Sentiment	8.95
Sentiment + Price	7.34
RNN	4.70
GRU	4.28
LSTM	3.49

Wines are evaluated on a 100-point scale. Ratings correspond to rough recommendations; WineSpectator, for example, uses these score band breakdowns:

- 95-100 Classic: a great wine
- 90-94 Outstanding: a wine of superior character and style
- 85-89 Very good: a wine with special qualities
- 80-84 Good: a solid, well-made wine
- 75-79 Mediocre: a drinkable wine that may have minor flaws
- 50-74 Not recommended

As expected, the Naive Bayes model baseline is outperformed by every other model. More surprising is the performance of price alone as a predictor of score; while in general, we expect people to be more willing to pay more for better wine, price alone was typically off by 2.81 points– meaning that it not only outperformed Naive Bayes, but had a reasonable likelihood of placing a wine in the correct score band.

Sentiment alone also did fairly well, though not as well as price alone. Given that a wine's score is meant to convey the quality and enjoyability of a wine, this matches our expectation that a wine review description's sentiment would correlate well with the wine's score. In addition, sentiment and price, together, outperformed either individual predictor, being off by 2.71 points on average.

TFIDF outperformed all previously-discussed models, with an average deviation of 1.88 points. More likely than not, TFIDF would place the wine in the correct score band. Because TFIDF allows the model to weight words important to each individual review more heavily, the model is possibly better-able to account for words that indicate positive or negative traits in specific types of wine (e.g. words like ”sweet” may be more appropriate for California cabernet sauvignons, as these wines are known for their sweeter flavours; however, such a taste may be more inappropriate for an Old World wine).

The performance of the TFIDF predictor lead us to expect the rating to have a fairly linear relationship with word features. We verified that using an RNN does not vastly improve the MSE. While RNNs are built for analyzing data with nonlinear relationships, we note that it is robust in the sense that it performs well even on linear data.

One of the striking things was the success of our LSTM model compared to our other neural network models. We believe this is due to the long-term ‘memory’ that LSTMs have [8] which allows the recurrant neural network to ‘remember’ more of the information as it unrolls. It may be that in predicting rating, the entirety of the description is more important for accuracy.

B. Predicting Price

TABLE II: Predicting Price from Sommelier Reviews

Model	Mean Squared Error	Accuracy of Price Category
Naive Bayes	NA	0.30
TFIDF	1553.65	0.38
Sentiment	1885.82	0.26
RNN	905.38	0.37
GRU	757.40	0.41
LSTM	760.63	0.39

Like score bands, wines have price categories as well. WineFolly uses the following categories:

- Value: between $4 and $10
- Popular Premium: between $10 and $15
- Premium: between $15 and $20
- Super Premium: between $20 and $30
- Ultra Premium: between $30 and $50
- Luxury: between $50 and $100
- Super Luxury: between $100 and $200
- Icon: above $200

Based on these breakdowns, we consider each model’s test mean squared error in two ways: (1) how many points the expected error is, and (2) whether each model correctly predicted wine price categories.

As expected, the Naive Bayes model baseline is not very accurate (30% accurate at predicting price categories). More surprising is the poor predictions of every other model as well. While our dataset contains a large range of prices (with the highest-priced wine priced at $3,300 and 15 wines priced $1,000 or above), over 99% of wines come in at $200 or below; thus, error of over $27 represents a significant deviation.

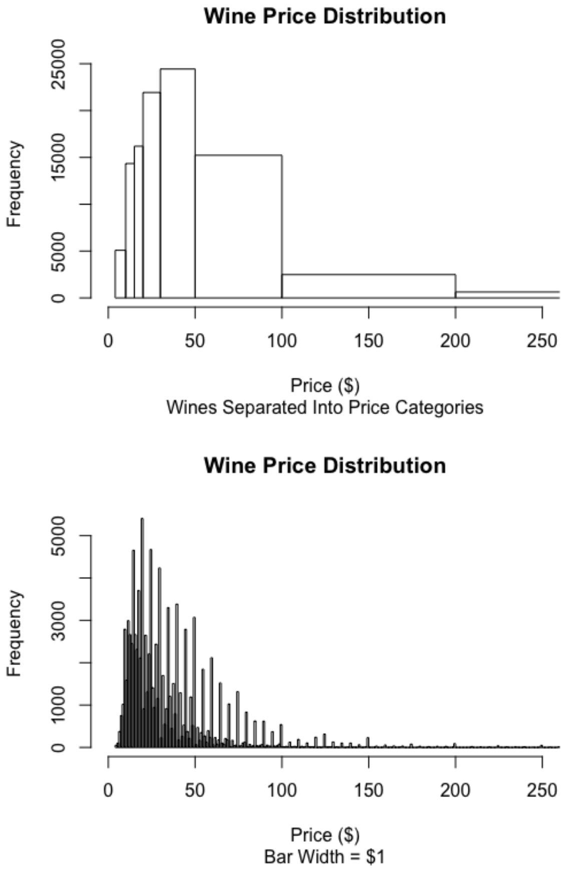


Fig. 2: Wine Price Distribution. We leave off wine prices over $250, as most wines (over 99%) are clustered below $200. Of note: wines tend to spike at prices such that (wine price) mod 5 = 0, suggesting that wine pricing is not purely coupled to objective measures like score/points.

We conjecture some possible explanations for this behaviour.

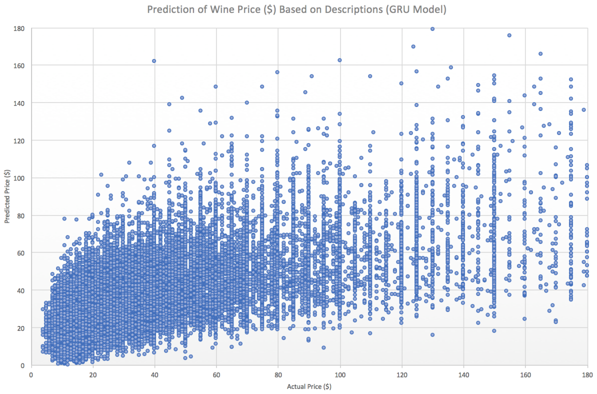


Fig. 3: Price predicted by GRU vs. true price

1) Our linear models are all uniformly poor at price-prediction, as they fail to take into account some piece of data not supplied in the fields of data we are studying. This theory is plausible, as there are many

factors that go into pricing a wine. Some confounding ones could include:

- Wines that are ”in-style” are likely to be priced higher, despite no change in review sentiment.
- Wines from a particular region could be priced higher or lower depending on whether they are or are not favoured; e.g. a New World-style wine from an Old World-winery could need a lower price to sell to an Old World-based customer base.

2) Our models are fairly good predictors of what price should be; it is the wineries that are mispricing their own wines. There is evidence to suggest this theory as plausible as well:

- Wine is a candidate to be subject to premium pricing strategies; like jewelry, wine is a product that people may associate higher quality with for no reason other than its price; thus, some wineries may be engaging in pricing schemes decoupled from wine quality.
- Wine is a field steeped in tradition. Thus, some wineries may be pricing their wines too low or too high, for reasons other than those that might be reflected in a review, or a score (because ”that's how they've always done it”). However, in some views, tradition is a proxy for lack of efficiency; if this theory holds, using a price prediction model may enable wineries to better price their wines (and, e.g., gain more profit from previously-underpriced wines).

3) The most expensive wines are typically from a highly-respected winery or have been aged for many years. These are properties can be read on the bottle and therefore would not show up on the wine review.

Possibly due to a combination of the preceding factors, it is reasonable to expect that there is a significant nonlinear relationship between the price of wine and the words in the wine review. Our RNNs were able to capture part of this nonlinearity and reduced the MSE by more than half. Despite this accuracy reduction of half, our best neural network, a GRU (closely followed by the LSTM model) was still approximately $27 off, which in some cases were wrong by 2-3 categories. This also would explain the difference we found between the accuracy improvements based on price and price category: while the neural nets gave us a lower mean squared error, they did not necessarily improve much on price as a category prediction.

C. Categorical Response Prediction

Tables III and IV summarize of our results for predicting country of origin and grape varietal, respectively. As expected, sentiment is not a very good predictor. These wine reviews contain wines across a range of varietals, countries, and qualities, and it is unreasonable to assume that a certain kind of wine consistently earns poor reviews.

It is interesting to note the discrepancy between Naive Bayes and TFIDF. Both of these models are based on word frequencies, and neither model takes into account interaction effects between different words. We suspect that the reason behind this phenomenon is that the conditional probabilities for each of the categories is very small, while the variance among different documents is large. It then follows that picking the category with the maximum conditional probability is likely to be error-prone. On the other hand, logistic regression tends to be a more robust way to classify categorical data.

TABLE III: Predicting Country of Origin from Sommelier Reviews

Model	Accuracy
Naive Bayes	0.70
TFIDF	0.83
Sentiment	0.49
RNN	0.81
GRU	0.87
LSTM	0.87

TABLE IV: Predicting Grape Varietal from Sommelier Reviews

Model	Accuracy
Naive Bayes	0.38
TFIDF	0.63
Sentiment	0.13
RNN	0.50
GRU	0.68
LSTM	0.66

Our neural networks were able to outperform all of our baseline models. For the problem of predicting grape varietal, we believe that it is important to look at interaction effects between various words in the review. For instance, Chardonnay wines are typically described as being very alcoholic and oaky. A recurrent neural network does a better job at capturing this interaction than a simple word frequency analysis.

Finally, each of our models appears to perform better when predicting country of origin than predicting grape varietal. This is simply due to the fact that there are only 8 distinct countries in our dataset, and wines from the United States make up approximately half of the data. Thus, the trivial predictor that always predicts the United States will achieve a success rate of around 0.50.

V. ERROR ANALYSIS

In addition to our experiments detailed above, we also modeled on segmented price data. Pricing ranged from $4 to $3,300, with 8 distinct price categories. While segmenting by score bands was also a possibility, our dataset contained wines rated 80 and above, and a segmentation by score bands (only 4 in our data) would not be as helpful.

The mean square error of score prediction doubles between the lowest price category ($4-$10) and the highest price category ($200+). However, this is intuitively reasonable. As demonstrated previously, price and sentiment are each

TABLE V: MSE for Points v. Sentiment & Price, segmented by price category

Price Min, Max ($)	n	MSE (points v. sentiment & price)
4, 10	2295	3.228146
10, 15	12482	3.572711
15, 20	15444	4.583651
20, 30	23099	5.541829
30, 50	25598	6.077117
50, 100	17758	5.987407
100, 200	2943	5.828646
200, 5000	724	6.441793

relatively accurate predictors of wine score, and the lower price categories are smaller (and thus less likely to have as much variation).

To eliminate variation caused by the difference in price category sizes, we also calculated MSE for price bands of width $5, as shown in Figure 4.

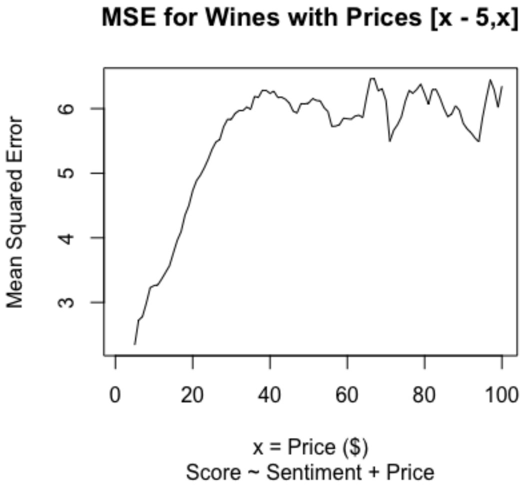


Fig. 4: MSE (Score Sentiment + Price), for wines priced between $(x - 5) and $x

This result is more illuminating, and confirms that our model is more accurate for cheaper wines than more-expensive ones. We hypothesize that the lower-end wine market is more efficient and standardized (e.g. it is more straightforward to ask about the difference between a $5 and $20 bottle of wine than it is to ask about the difference between a $500 and $2,000 bottle of wine). Furthermore, some of the factors that confound wine pricing (outside of information available in the description) are more likely to be prevalent at higher price categories (e.g. luxury item pricing) than lower price categories.

VI. CONCLUSIONS

In this project, we analyzed the predictive power of several models on text-based features. Our simplest model, a sentiment analysis, collapsed our review to a single number and performed adequately when predicting wine ratings. However, it lacked the ability to predict any of our categorical responses.

We fit more complicated linear models using TFIDF-weighted word frequency vectors, which did a much better job predicting country of origin and wine varietal. However, that model still had trouble predicting wine price, which seems to have a strong non-linear dependency on our features.

This led us to use recurrent neural networks, which were specifically designed to deal with such nonlinearities. Our best neural network models were able to greatly reduce the MSE when predicting price, while remaining at least as accurate when predicting the categorical responses. Due to their ability to handle both linear dependencies as well as nonlinear ones, we find that neural networks generalize best across all of our prediction tasks.

A. *Future Directions*

In the future, we would like to continue improving the predictive power of our models by increasing the complexity of our neural networks. Currently, our RNNs have only one nonlinear layer, and believe that adding additional layers may improve performance.

ACKNOWLEDGMENT

We'd like to thank Percy Liang, Stefano Ermon, Steve Mussman, and the CS 221 TA team for their instruction, guidance, and inspiration. We'd also like to thank the colleagues and friends who attended the CS 221 poster session and/or offered valuable feedback for our project.

REFERENCES

[1] T. Joachims, A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization, Research Report, Dept. Computer Science, Carnegie Mellon Univ., Pittsburgh, PA, 1996.

[2] H. Kaur, V. Mangat and Nidhi, "A survey of sentiment analysis techniques," 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, 2017, pp. 921-925.

[3] S. Rajalakshmi, S. Asha and N. Pazhaniraja, "A comprehensive survey on sentiment analysis," 2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN), Chennai, 2017, pp. 1-5.

[4] Natural Language Processing - Research at Google, Google. [Online]. Available: https://research.google.com/pubs/NaturalLanguageProcessing.html. [Accessed: 15-Dec-2017].

[5] J. Pennington, R. Socher, and C. D. Manning. 2014. GloVe: Global Vectors for Word Representation.

[6] S. Hochreiter and J. Schmidhuber. 1997. Long Short-Term Memory. Neural Comput. 9, 8 (November 1997), 1735-1780. DOI=http://dx.doi.org/10.1162/neco.1997.9.8.1735

[7] K. Cho, B. Merrie, C. Gulcehre F. Bougares, H. Schwenk, D. Bahdanau, and Y. Bengio, "Learning Phrase Representations using RNN EncoderDecoder for Statistical Machine Translation," 2014 Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), October 25-29, 2014, Doha, Qatar pages 1724-1734.

[8] Y. Bengio, P. Simard, and P. Fransconi, "Learning Long-Term Dependencies with Gradient Descent is Difficult," 1994 IEEE Transactions on Neural Networks, Vol 5, No 2.