Predicting the Region of Wine Based on Descriptions

Team: Sandeep Alankar, James Springer, Kevin Yang **Project Mentor TA:** Brian Chen

1) Abstract

In this project, we investigated the problem of determining where a wine originated from based on several wine attributes including variety and price as well as user reviews on the taste of the wine. We chose this problem because we were motivated by our collective interest in wine tasting as well as wanting to study if wine descriptions and origination can be correlated. We have two main contributions in this project, one data and one analysis. For the data contribution, we preprocess and encode the numerical data in a way that the natural language processing models can use and contextualize them with the text reviews. For the analysis contribution, we run several different machine learning models to handle this classification task to investigate the performance of various models as well as the sensitivity to a variety of hyperparameters.

Through this project, we have collected data, cleaned and processed it to serve as input to various machine learning models, and tuned the models to determine the optimal strategy to solve our problem. Based on our results, the BERT model resulted in the highest train, validation and test accuracies while the AdaBoost with decision tree model performed the best on the training set.

2) Introduction

We plan to predict the region that a bottle of wine originally comes from as a function of several wine attributes including variety and price as well as user reviews on the taste of the wine. More directly, the input of our system will be a combination of numerical attributes such as the price of the wine as well as the tokenized versions of the wine review, and the output of our system will be the region/province/country that the wine originates from.

We will begin by using a wine reviews dataset found on Kaggle to serve as our initial dataset. We can tokenize the wine descriptions to identify common words and taste adjectives that can help our predictions task. We have identified other wine datasets that go more in depth on the quality of wine that could be joined with the reviews dataset. However, when joining with the dataset, it significantly reduces the number of samples so we have decided to omit it. After collecting the data, we can then split the data into 70/20/10 training/validation/testing sets and begin modeling on them.

Since this is a classification task, the primary measurement of performance will be the accuracy of our model at predicting the region the wine comes from. Within this task, we can set granularity levels such as predicting the country that the wine comes from vs the exact region that the wine comes from. Afterwards, we will compare the results from various, tuned machine learning models to determine what would work the best for our problem.

As additional motivation, we have noticed that some vineyards are guilty of using industrial biocides to enhance the flavors of grapes grown in that soil and although doing so may produce a better tasting wine, such methods harm the environment and reduce the soils' ability to release greenhouse gasses. By investigating common tasting notes and descriptions of wines and linking them to the region that the wines were produced, we may be able to correlate areas where wines are produced and if they might be using industrial biocides based on the descriptions. This could help identify wine from areas to avoid, being more environmentally conscious going forward.

3) Background

The wine reviews dataset posted to Kaggle contains numeric and textual information on over 100,000 different wines. This dataset has spurred numerous applications of machine learning in the domain of wine, including estimating a wine's quality and predicting the variety of a wine. One such area that has not been studied, however, is the prediction of a wine's geographic origin. We created models to predict three different granularities of geographic origin; these are, in increasing order of granularity: region, province, and country. Classification tasks in general have been applied to this dataset, and the methods used in these influence our application. We addressed a shortcoming of these approaches, as well, by encoding numeric and categorical features into the text instead of using them individually or separately. Two such influential Kaggle submissions are described below.

1. Kaggle Submission. Koji 2017
   https://www.kaggle.com/code/kitakoj18/exploring-wine-descriptions-with-nlp-and-kmeans
   The following Kaggle submission uses natural language processing and k-means to perform an exploratory analysis on the dataset. It does not attempt to perform any classification or prediction but performs important preprocessing on the text data. We used similar methods in our natural language processing pipeline of the description features.
2. Kaggle Submission. Zsolt Diveki 2018
   https://www.kaggle.com/code/diveki/classification-with-nlp-xgboost-and-pipelines
   The following Kaggle submission uses gradient-boosted decision trees to predict a wine's variety and performs a thorough analysis on the dataset. This is a somewhat similar classification problem to ours and showcases some of the challenges posed with this dataset and methods to overcome them. For example, the sample sizes for each region in this dataset are highly varied, ranging from less than ten instances to thousands for a particular region. We used similar methods to overcome this and used ensembles of decision trees for our classification problem.

4) Summary of Our Contributions

Our group created a novel data pre-processing step by encoding numeric and categorical features into the description of a wine. We additionally performed a thorough analysis on the effectiveness of different models for this application and analyzed their sensitivity to hyperparameters. At the start of the project, we planned to contribute a novel algorithm that combined the numeric, categorical, and text features. After experimenting with the data, however, we found a data pre-processing step produced better results and have thus shifted to the following contributions.

1. Contribution in Application/Data: We encoded numeric and categorical features as text data by prepending it to the description for each wine in a data preprocessing step. This allowed the natural language models to use these features contextually when parsing the rest of the text. Additionally, this type of classification had not been done previously on this dataset.
2. Contribution in Algorithm: N/A
3. Contribution in Analysis: We analyzed the effectiveness of different models and their respective hyperparameters on predicting a wine's geographic origin. Additionally, the dataset contains different granularities of geographic location including country, province,

and region. We trained a model for each of these respective outputs and performed an analysis on each model's accuracy in predicting the relative location.

5) Detailed Description of Contributions

5.1.1 Data Preprocessing

This dataset contains numeric, categorical, and text features. We preprocessed the numeric and categorical features of this dataset into text to reduce it to a single feature type. The process of this is as follows:

1. For categorical features, encode each category as a single word and prepend it to the wine's description. Encoding the information this way allows the natural language processing models to contextualize these features with the rest of the text.
2. Encode numeric features as categorical data and repeat step (1). For example, the *price* feature can be encoded as "expensive" for quartile 1, "standard" for quartiles 2 and 3, and "affordable" for quartile 4.

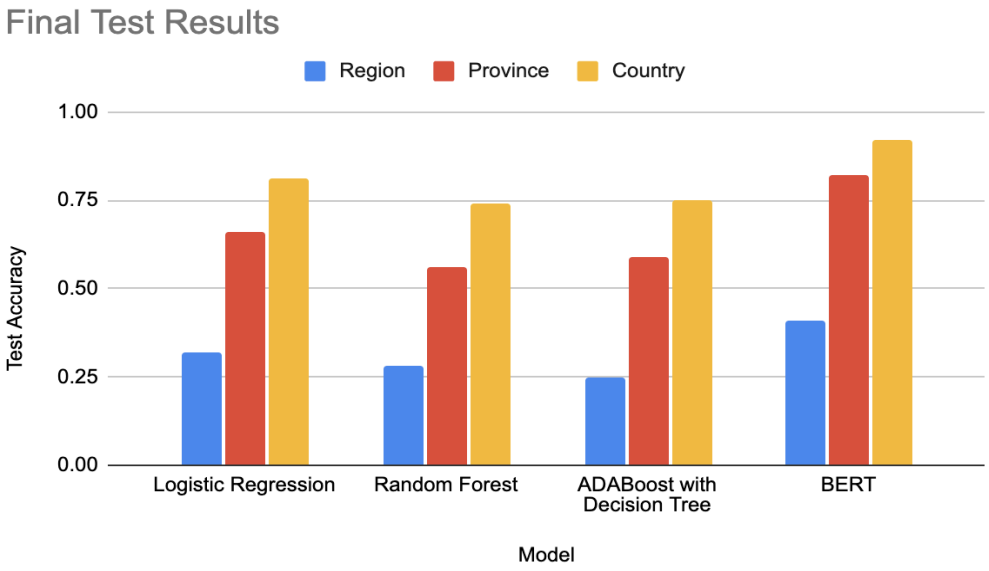
After this initial preprocessing step, varying word vectorization techniques were used depending on the model. For non-neural network based models, standard bag of words and tf-idf vectorization were used with word stemming to represent the description [Diveki, 2018]. Neural networks can use the text as is and will benefit from the other features' context as described above. Finally, the output space for this classifier is huge. The output group *region* has 1230 unique values with many only having a single instance of training data. It was unfeasible to predict classes with such little data; therefore we had to prune the output classes with less than 100 instances. The resulting dataset still contains over 90,000 instances and the following number of output classes.

Desired Classification	Output Classes
Region	215
Province	71
Country	14

5.1.2 Analysis Contribution

Our primary analysis will involve comparing the accuracy of different models at this classification task and gauging their sensitivity to hyperparameter changes. As noted, the main models we will use are Logistic Regression, Adaboost (with decision tree as the base model), Random Forest, and a Neural Network. We will tune the hyperparameters by comparing the accuracy on a held out validation set of the data and using methods such as cross fold validation. With the results from the various models, we will be able to compare and contrast what works best for our wine origination problem.

5.2 Experiments and Results



5.2.1 Logistic Regression

Output Class	Train Accuracy	Validation Accuracy	Test Accuracy	Solver	Penalty
Region	42.80%	31.55%	31.57%	saga	l2
Province	70.7%	67.6%	65.97%	saga	l2
Country	84.10%	80.79%	80.75%	lbfgs	l2

For logistic regression, we experimented with the following hyperparameters: solver(lbfgs, sag, saga) and penalty(l2, none). Since there were 6 combinations of solver and penalty, we tried all of them to find our results. There was one more solver that could handle multiclass classification, newton-cg, but it was not converging so we chose to omit that one. For the classification of region and province, the saga solver with the l2 penalty performed the best. For the classification of the country, the lbfgs solver with the l2 penalty performed the best, showing that the lbfgs solver might work better on a smaller set of classes while saga works better for larger one.

5.2.2 Random Forest

Output Class	Train Accuracy	Validation Accuracy	Test Accuracy	number of estimators	ccp alpha	max depth	min samples leaf	min samples split
Region	47.00%	28.60%	28.00%	500	0	30	2	5
Province	71.20%	57.50%	56.70%	50	0	40	1	10
Country	81.40%	74.50%	74.10%	50	0	40	1	10

For random forests, we experimented with the following hyperparameters: number of estimators, cost complexity alpha, max depth, min samples leaf, and min samples split. Due to

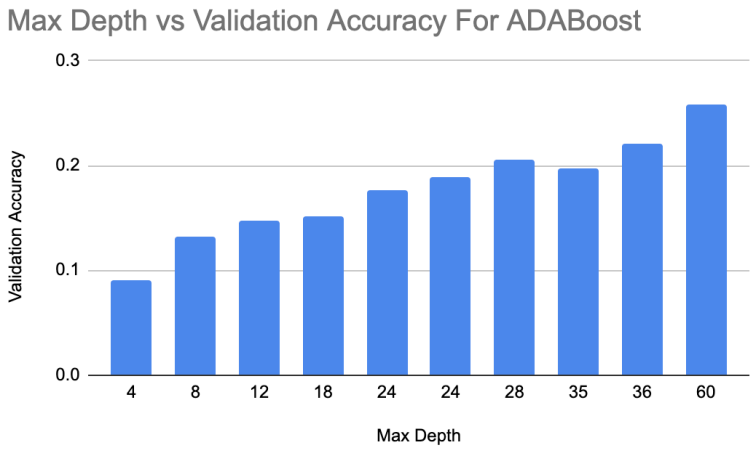
the high number of hyperparameters, a grid search was not feasible and a randomized search was used in its place. For classification of country and province, we found that a lower number of estimators and medium complexity trees produced the best results. For classification tasks with a relatively lower number of output classes, the model was particularly sensitive to the number of estimators and would quickly underfit if more than 75 trees were used. This is in stark contrast to the prediction of the wine's region, which had over 200 output classes. For this classification problem, we found that a high number of estimators was the most crucial parameter with medium to low complexity trees performing the best. We achieved the best results using 500 estimators and were limited by the compute capabilities of Google Colab; better performance on classifying regions could have been achieved with more RAM.

5.2.3 AdaBoost with Decision Tree

Output Class	Train Accuracy	Validation Accuracy	Test Accuracy	Number of Estimators	Max Depth
Region	99.70%	25.76%	25.29%	100	60
Province	99.98%	59.63%	58.55%	62	65
Country	99.99%	75.45%	74.62%	71	48

For AdaBoost, we first wanted to use both logistic regression and decision trees as the base model. However, with the amount of time logistic regression took to converge on our dataset (around 30 minutes), it was not feasible to do so with our given resources so we decided to focus on decision trees as the base model instead. For the number of estimators and max depth hyperparameters, we employed a random search in the range of 1-100 for the number of estimators and 1-75 for the max depth. After training 10 different models with the random hyperparameters for each of the output classes, we soon realized that the accuracies were very sensitive to the max depth of the tree while the number of estimators did not affect the results as much. The higher max depth would lead to better accuracies across training, validation and testing; however, it also did tend to overfit the model as we can see by the high training accuracies as compared to the validation and test accuracies. For this reason, this model is not a great option for our classification task.

An example of the max depth hyperparameter's effect on the validation accuracy:



5.2.4 Neural Network

Output Class	Train Accuracy	Validation Accuracy	Test Accuracy
Region	42.8%	41%	41.3%
Province	82.69%	81.4%	81.6%
Country	93.3%	91.3%	91.6%

For all our BERT model testing, we trained over 3 epochs with a learning rate of 1e-5. We used a pre-trained BERT model for our testing trials, specifically the BERT base model, which consists of 12 layers of transformer encoders, 12 attention heads, a hidden layer of size 768, and 100 million parameters. Because of the massive size of this base model, we could only train over 3 epochs and this process still took about 35 minutes per epoch using a GPU runtime. In our original testing phase, we tried a higher learning rate to speed up the training process but found that a learning rate of 1e-5 gave us higher resultant accuracies while still not drastically increasing the train time. Also, due to compute constraints, we trained our BERT model on only the first 15,000 rows of our dataset because training it with the full 91,612 rows would take about 130 hours to train per epoch. Regardless of this smaller dataset, we were still able to achieve higher train, validation and test accuracies compared to our other models and we believe that, given enough compute resources and time, training our BERT model on the entire wine dataset would result in even higher accuracies across the train, validation, and test sets.

6) Compute/Other Resources Used

We trained our logistic regression, random forest, and AdaBoost with decision tree models using a CPU runtime but when training our BERT model, we had to switch to a GPU runtime due to the large size of the base BERT model. We were able to successfully train our models using the standard Colab compute resources but often ran into GPU limits while training our BERT models, so we had each group member train BERT with different hyperparameters on their own Google account to avoid GPU usage limits stopping us from training the model.

7) Conclusions

Through working on this project, we were able to successfully classify the region a wine originated from various attributes like its price, variety, and user reviews. We tested our predictions on various model architectures and through extensive testing, we found that the BERT model resulted in the highest train, validation and testing accuracies. This project allowed us to explore various topics in machine learning that we were previously interested in, such as data preparation, neural networks, and natural language processing. As seen in our report, we were able to produce meaningful results and offer multiple models that predicted a wine's region of origination with varying accuracies depending on hyperparameter values and the model architecture.

Back when we initially came up with the idea for this project, we were not aiming to use a neural network to make predictions on our dataset but as we learned more about natural language processing both in and out of class, we discovered the BERT architecture, which was difficult to implement but ended up making the best predictions on the train, validation and test sets compared to our other models. In the future, this project could be improved by further fine-tuning the hyperparameters of each model and having access to more compute resources.

(Exempted from page limit) **Broader Dissemination Information:**

Your report title and the list of team members will be published on the class website. Would you also like your pdf report to be published?
NO

(Exempted from page limit) Attach your midway report here, as a series of screenshots from Gradescope, starting with a screenshot of your main evaluation tab, and then screenshots of each page, including pdf comments. This is similar to how you were required to attach screenshots of the proposal in your midway report.

(Exempted from page limit) Supplementary Materials if any (but not guaranteed to be considered during evaluation):

Predicting Region of Wine Based on Description
Team: Sandeep Alankar (CIS 5190), James Springer (CIS 5190), Kevin Yang (CIS 5190)

1) Introduction

We plan to predict the region that a bottle of wine originally comes from as a function of several wine attributes including variety and price as well as user reviews on the taste of the wine. More directly, the input of our system will be a combination of numerical attributes such as the price of the wine as well as the tokenized versions of the wine review, and the output of our system will be the region that the wine originates from.

We will begin by using a wine reviews dataset found on Kaggle to serve as our initial dataset. We can tokenize the wine descriptions to identify common words and taste adjectives that can help our predictions task. We have identified other wine datasets that go more in depth on the quality of wine that could be joined with the reviews dataset. However, when joining with the dataset, it significantly reduces the number of samples so we have decided to omit it. After collecting the data, we can then split the data into 70/20/10 training/validation/testing sets and begin modeling on them.

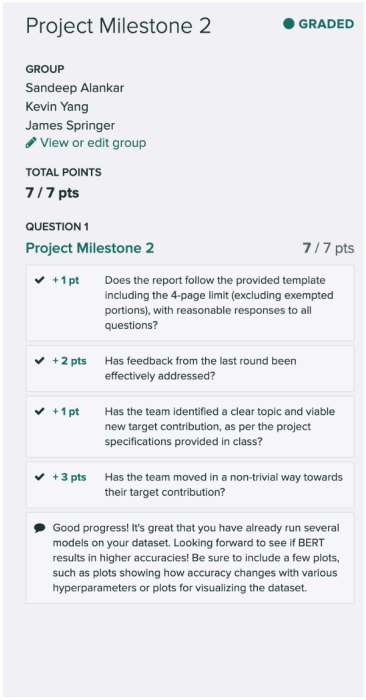
Since this is a classification task, the primary measurement of performance will be the accuracy of our model at predicting the region the wine comes from. Within this task, we can set granularity levels such as predicting the country that the wine comes from vs the exact region that the wine comes from. Additionally, we can use metrics such as precision/recall and AUC/ROC to measure our performance.

Motivation:

Some vineyards are guilty of using industrial biocides to enhance the flavors of grapes grown in that soil and although doing so may produce a better tasting wine, such methods harm the environment and reduce the soils' ability to release greenhouse gasses. By investigating common tasting notes and descriptions of wines and linking them to the region that the wines were produced, we may be able to correlate areas where wines are produced and if they might be using industrial biocides based on the descriptions. This could help identify wine from areas to avoid, being more environmentally conscious going forward.

2) How We Have Addressed Feedback From the Proposal Evaluations

The key feedback that we received from the TA on our proposal was in regards to how we will approach the processing of our text descriptions. The TA asked if we will take the average of all word embeddings or focus on the presence of certain words. Our initial approach will be to identify the N most common words in all of the descriptions and then build a bag of words and tf-idf vectors to be able to use the frequency and weights of the words as inputs to the models that we create. The other feedback that we received was about how we can mix the text features with the numerical ones. By using the approach of bag of words and tf-idf, we can simply join the vector of word frequencies with the rest of the data. Additionally, we are exploring other methods for converting all feature types to text described in 5.1.1 that will help with our neural network based models.



3) Prior Work We are Closely Building From

1. Kaggle Submission. Koji 2017
   https://www.kaggle.com/code/kitakoj18/exploring-wine-descriptions-with-nlp-and-kmeans
   The following Kaggle submission uses natural language processing and k-means to perform an exploratory analysis on the dataset. It does not attempt to perform any classification or prediction but performs important preprocessing on the text data. We will be using similar methods in our natural language processing pipeline of the description features.
2. Kaggle Submission. Zsolt Diveki 2018
   https://www.kaggle.com/code/diveki/classification-with-nlp-xgboost-and-pipelines
   The following Kaggle submission uses gradient-boosted decision trees to predict a wine's variety and performs a thorough analysis on the dataset. This is a somewhat similar classification problem to ours and showcases some of the challenges posed with this dataset and methods to overcome them. For example, the sample sizes for each region in this dataset are highly varied, ranging from less than ten instances to thousands for a particular region. We will be using similar methods to overcome this and will also be attempting to use ensembles of decision trees for our classification problem.

4) What We are Contributing

Our group is attempting to predict a wine's region from both numeric and text features. We were originally planning to contribute a novel algorithm that combined both these types of features. After experimenting with the data, however, we found the numeric features were either given near-zero weights or too much weight with respect to the text features. We are now shifting to a data and analysis contribution described below to better integrate these two types of features.

1. Contribution in Application/Data: We will be encoding the numeric and categorical features as text data and prepending it to the description for each wine in a data preprocessing step. This will allow the natural language models to use these features contextually when parsing the rest of the text. Additionally, this type of classification has not been done previously on this dataset.
2. Contribution in Algorithm: N/A
3. Contribution in Analysis: We will be analyzing the effectiveness of different models and their respective hyperparameters on predicting a wine's region. Additionally, the dataset contains different granularities of geographic location including country, province, and region. We will train a model for each of these respective outputs and perform an analysis on each model's accuracy in predicting the relative location and the features important in determining this.

5) Detailed Description of Each Proposed Contribution, Progress Towards It, and Any Difficulties Encountered So Far

5.1.1 Data Preprocessing

This dataset contains numeric, categorical, and text features. We will be preprocessing the numeric and categorical features of this dataset into text to reduce it to a single feature type. The process of this is as follows:

1. For categorical features, encode each category as a single word and prepend it to the wine's description. Encoding the information this way allows the natural language processing models to contextualize these features with the rest of the text.
2. Encode numeric features as categorical data and repeat step (1). For example, the *price* feature can be encoded as "expensive" for quartile 1, "standard" for quartiles 2 and 3, and "affordable" for quartile 4.

After this initial preprocessing step, varying word vectorization techniques will be used depending on the model. For non-neural network based models, standard bag of words and tf-idf vectorization can be used with word stemming to represent the description [Diveki, 2018]. Neural networks can use the text as is and will benefit from the other features' context as described above. Finally, the output space for this classifier is huge. The output group *region* has 1230 unique values with many only having a single instance of training data. It is unfeasible to predict classes with such little data; therefore we will be pruning the output classes with less than 100 instances. The resulting dataset still contains over 90,000 instances and over 200 unique output classes.

5.1.2 Analysis Contribution

Our primary analysis will involve comparing the accuracy of different models at this classification task and gauging their sensitivity to hyperparameter changes. As noted, the main models we will use are Logistic Regression, Adaboost (with decision tree and logistic regression as the base model), Random Forest, and a Neural Network. We will tune the hyperparameters by comparing the accuracy on a held out validation set of the data. Some examples of the hyperparameters are the penalty and solver for Logistic Regression and the depth and number of estimators for Random Forest. Additionally, we have some interesting parameters on our data set. One example is that we are limiting our bag of words and tf-idf vectors to dictionaries of a limited size. We can treat these as hyperparameters and tune/analyze their sensitivity as well.

5.1.3 Methods

In order to properly predict the region of origination for a bottle of wine given several of its attributes, we aim to test out multiple approaches and compare their performances to see which approach results in the highest prediction accuracy over a large dataset. To preprocess the text of the wine reviews, we referenced the Kaggle submission in which the author implemented natural language processing on the wine reviews dataset [Koji, 2017]. We adopted similar techniques in our project as described in 5.1.1. An important part of preparing our data for these algorithms was to select the right features, remove any duplicates or nan values, and perform target feature processing [Diveki, 2018]. Once our data was properly preprocessed, we tested different prediction and classification algorithms on our dataset and measured the resultant performance.

Going forward, we would like to implement Bidirectional Encoder Representations from Transformers (BERT) and compare its performance to that of the base/boosted prediction and

Project Milestone 2 — GRADED

GROUP
Sandeep Alankar
Kevin Yang
James Springer
View or edit group

TOTAL POINTS
7 / 7 pts

QUESTION 1
Project Milestone 2 — 7 / 7 pts

✔ + 1 pt	Does the report follow the provided template including the 4-page limit (excluding exempted portions), with reasonable responses to all questions?
✔ + 2 pts	Has feedback from the last round been effectively addressed?
✔ + 1 pt	Has the team identified a clear topic and viable new target contribution, as per the project specifications provided in class?
✔ + 3 pts	Has the team moved in a non-trivial way towards their target contribution?

Good progress! It's great that you have already run several models on your dataset. Looking forward to see if BERT results in higher accuracies! Be sure to include a few plots, such as plots showing how accuracy changes with various hyperparameters or plots for visualizing the dataset.

Project Milestone 2 — GRADED

GROUP
Sandeep Alankar
Kevin Yang
James Springer
View or edit group

TOTAL POINTS
7 / 7 pts

QUESTION 1
Project Milestone 2 — 7 / 7 pts

✔ + 1 pt	Does the report follow the provided template including the 4-page limit (excluding exempted portions), with reasonable responses to all questions?
✔ + 2 pts	Has feedback from the last round been effectively addressed?
✔ + 1 pt	Has the team identified a clear topic and viable new target contribution, as per the project specifications provided in class?
✔ + 3 pts	Has the team moved in a non-trivial way towards their target contribution?

Good progress! It's great that you have already run several models on your dataset. Looking forward to see if BERT results in higher accuracies! Be sure to include a few plots, such as plots showing how accuracy changes with various hyperparameters or plots for visualizing the dataset.

classification algorithms we have implemented thus far. We are choosing to explore this model over other neural network architectures useful for natural language processing such as recurrent neural networks (RNNs) or long short term memory network (LSTM) because BERT is based on a transformer architecture which solves the problem of vanishing and exploding gradients that RNNs have and the issue of slow speed and limited long term memory that LSTMs have. BERT will be useful for us to process the thousands of wine reviews in the dataset we are using because as a contextual model, it "generates a representation of each word that is based on the other words in the sentence" [Mohan, 2020]. We plan to utilize BERT to extract high-quality features from our wine review dataset and fine-tune the model on a classification task to improve the quality of our predictions.

5.2 Experiments and Results

As stated above, we aim to implement multiple classification algorithms and models and compare the performance of each to determine the best model to use for an effective and accurate wine region prediction system. So far, we have run a base model for logistic regression on our preprocessed dataset with an accuracy of 0.563, a random forest of depths 20, 30, 50, and 100 to find the highest accuracy being the one with the highest depth which resulted in an accuracy of 0.578. We also tested ADABoost on a base logistic regression model to get an accuracy of 0.500, a based decision tree stump to obtain an accuracy of 0.397, and decision trees of depth 20, 50, and 100 which confirmed to us that the decision tree with the highest depth would indeed have the highest resulting prediction accuracy when compared to other iterations of the same algorithm with a lower depth. We also performed a kmeans analysis on our bag of words vectorization of description for each variety and found rough clusters in the data. We visualized these clusters to gain a better understanding of the structure of our data. The above accuracies are a result of our numerical data being represented as solely numeric data; however, when we prepended our numeric features as categorical data, our accuracies decreased to 0.328 for logistic regression, 0.278 for a random forest with depth 100, and 0.256 for boosted logistic regression. In the future, implementing BERT should result in higher accuracies using our numeric and categorical data as text and we hope to document this increase in accuracy in our future testing and analysis.

6) Risk Mitigation Plan

In order to build a minimum viable project given that we run into obstacles and implementation issues, our risk mitigation plan includes several elements. Our original idea for this project was to make an algorithmic contribution, but we realized that since we were dealing with both numerical and text-based features, our numerical features did not correlate with NLP when applied at the end of our testing pipeline. Therefore, we pivoted and decided to make a data contribution instead; we prepended numeric features to the wine's description as categorical data. If our project becomes too computationally intensive, we can use Amazon SageMaker Lab's GPU along with performing dimensionality reduction on our dataset. Finally, if we run into implementation issues with BERT, we can test out other neural network architectures that are useful for NLP applications such as RNNs and LSTMs.

Project Milestone 2 ● GRADED

GROUP
Sandeep Alankar
Kevin Yang
James Springer
View or edit group

TOTAL POINTS
7 / 7 pts

QUESTION 1

Project Milestone 2 **7** / 7 pts

✔ + 1 pt	Does the report follow the provided template including the 4-page limit (excluding exempted portions), with reasonable responses to all questions?
✔ + 2 pts	Has feedback from the last round been effectively addressed?
✔ + 1 pt	Has the team identified a clear topic and viable new target contribution, as per the project specifications provided in class?
✔ + 3 pts	Has the team moved in a non-trivial way towards their target contribution?

Good progress! It's great that you have already run several models on your dataset. Looking forward to see if BERT results in higher accuracies! Be sure to include a few plots, such as plots showing how accuracy changes with various hyperparameters or plots for visualizing the dataset.

No next page