Predicting the origin of wine with a random forest classifier

Anthony Vetturini, Anuhya Edupuganti, Maxwell Little, Srividya Gandikota
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213
avetturi@andrew.cmu.edu aedupuga@andrew.cmu.edu
mnlittle@andrew.cmu.edu sgandiko@andrew.cmu.edu

Abstract

Wine fraud is a multibillion-dollar industry, with a single bottle of fake wine costing up to millions of dollars. We aim to use a dataset containing properties of legitimate wine, which can be used to predict the region of origin of the wine, and thereby test the authenticity of wine by the region it claims to originate from. To select the best classification model, we compared the accuracy score of 4 classifiers: Random Forest, Logistic regression, K nearest neighbors, and Support Vector machines. With feature importance obtained from random forest, we showed that just 4 features of the 13 presented in the dataset - Proline, Color Intensity, Flavonoid and Alcohol content of the wine - were sufficient to train the classifiers to obtain the accuracy score of 1.0. While the classification is performed on a small dataset with wine originating from 3 regions, the success of the Random Forest model presents an exciting opportunity to expand the wine dataset using just the main 4 features to define wine properties, which can save cost and time compared to extracting 13 features. Furthermore, in testing for authenticity of wine, just a small sample of the expensive wine needs to be extracted to test for the 4 features.

1. Introduction

The wine industry is riddled with fake wines sourced from various regions of the world throughout history, with some of these faked bottles of wine reaching over $1 million dollars [1]. Maureen Downey, a world-renowned expert in wine counterfeiting, estimates that wine fraud is a multi-billion-dollar industry [2]. A very common method of counterfeiting wines (and other spirits) is to refill legitimate bottles with fake wine (and other ingredients), and then reseal and sell them [2].

Our objective is to use a dataset that contains various properties of legitimate wines, such as magnesium content, total phenols, and proline, to determine region of origin for Italian wines [3]. If proven successful, this model could be used in wine auctions where a coravin could be used to remove small samples of wine from an up-for-sale-bottle to test authenticity of the Italian region of origin. The use of a coravin does not destroy or alter the bottle in any way, and simply extracts a small portion of wine that can then be chemically tested to validate the wine being sold at auction and prevent the fraud market from further growing.

In this study, various classification-based machine learning (ML) algorithms were used including Multilinear Logistic Regression, Random Forest, K Nearest Neighbor, and SVM. Overall, we found that the Random Forest model produced the most reliably accurate results when used for testing. In each model, we tested various hyperparameters that could be used in the scikit-learn API [4]. We also analyzed the effects of feature reduction on each model's accuracy as the data set was relatively limited compared to the number of features (13 features for 178 data points).

2. Related Work

There have been many studies using the wine data set provided by the University of California - Irvine [3]. An interesting study was performed that used the data set (amongst other data sets) to study the effects of weight and learning rate initialization in a multilayer perceptron [5, 6]. There have also been many studies using the data set to analyze clustering. In one study, the authors found that the wine data set was too sparse to amplify the block structure of the affinity matrix, which led to erroneous results for the wine dataset (although other benchmark tests proved the effectiveness of their designed methodology) [7].

A commonality these previous studies is that the wine dataset was used as a test set amongst other data sets to prove a concept [5 - 7]. However, no shown literature uses the wine dataset as a focus point or seeks to address comparisons between ML models.

Our work deviates from previous efforts as we aim to identify the simplest and most accurate classifier for the data and reduce the number of features towards the practical goal of reducing time and cost of testing wines for authenticity.

Feature	Explained Variance	Total Explained Variance
Alcohol	0.36	0.36
Malic acid	0.19	0.55
Ash	0.11	0.67
Alcalinity of ash	0.07	0.74
Magnesium	0.07	0.80
Total phenols	0.05	0.85
Flavanoids	0.04	0.89
Nonflavanoid phenols	0.03	0.92
Proanthocyanins	0.02	0.94
Color intensity	0.02	0.96
Hue	0.02	0.98
OD280/OD315 of diluted wines	0.01	0.99
Proline	0.01	1.00

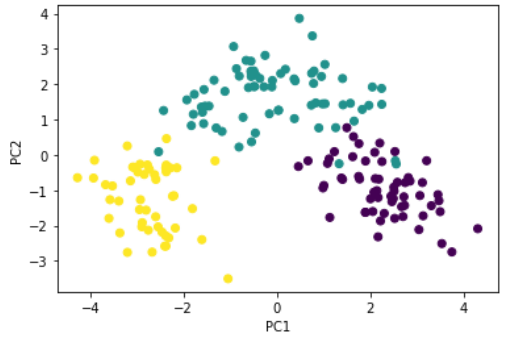

Figure 1. Results of PCA analysis. Top) Explained variance of each feature with total explained variances as a sum of explained variance of the corresponding features. Bottom) Wine dataset plotted with two principal components

3. Data

The data set we are using was found on the UCI Machine Learning Repository [1]. The data set contains 13 features including: alcohol content, malic acid, ash, alkalinity of ash, magnesium content, total phenols, flavonoids, nonflavonoid phenols, proanthocyanins, color intensity, hue, OD280 content, and proline. These data points were found via a chemical analysis of wines grown in the same region of Italy but from three different cultivars. These will be referred to as 'Region 1','Region 2' and 'Region 3' throughout the paper.

The dataset contains 13 features to describe the wine with only 178 data points. When analyzed with Principal Component analysis, it was found that most variance was explained by 'Alcohol' and 'Malic Acid' features, which account for 55% of total variance. The explained features are listed in the top image of Figure 1. Applying PCA on the entire dataset, the first two principal components are visualized in the bottom image in Figure 1. The graph captures the balanced nature of the dataset, with about 60 data points for each region. Most importantly, it reflects on the predictive power of the dataset, showing nearly distinct clusters. Thus, before even applying classification models, we can expect high classification accuracy from our present dataset, and confirm that we do not have large outliers.

The data was normalized via the Z-score method, which means that the features were normalized to have a mean of 0 and standard deviation of 1. The Z-score method tends to do well with a small number of outliers by describing them in relation to feature distribution and brings all features to comparable scale [8].

4. Methods

Random Forest, K Nearest Neighbor, Multiclass Logistic Regression, and SVM Classification models were each individually developed to analyze the data set. Below is a discussion on how each model was setup for our analysis. The following "Experiments" section will discuss individual (and cumulative) experiments run with each models.

4.1. Random Forest

Random forest for classification is an ensemble learning method that reports the mode of the outcome of multiple decision trees performing Classification, where a decision tree for classification is a supervised learning algorithm that repeatedly splits data based on discrete values of different features [9].

A key feature of random forest is "bagging" or "bootstrap aggregating", which refers to the trees of the random forest sampling random subsets of the dataset with replacement. Furthermore, random forest trees also train with a random subset of features. Together, these reduce correlation among trees [9]

4.2. K Nearest Neighbor

The K-nearest neighbors' (KNN) algorithm is a supervised learning classifier, typically non-parametric, that focuses on using neighboring nodes or proximity to make a prediction in regards to an individual data point. One key feature of this method is that the algorithm relies on distance between points, such that if a feature is widely distributed or large scales, normalizing the data would improve the accuracy of the system. Functionally, weighting the points in such a manner that closer points impact more is a useful technique for classifying points appropriately. These central features allow for the correct classification, which is heavily optimized for the region of wine prediction [12].

4.3. Multiclass Logistic Regression

Multiclass Logistic Regression is a supervised learning classifier which relies upon softmax activation to learn the weights of features from a training dataset to make predictions. There are many different solver types that can be used, but a common one is cross-entropy which uses the maximum likelihood estimation (MLE) as its derivation. Here we also note that the softmax function is both nonlinear and non-uniform which can lead to issues depending on how the weights are learned during training [10].

Another issue with logistic regression is that since it is a statistical analysis model, it may be over-fit on datasets with a large amount of features and lower amount of training points. Since our dataset contains many features for a

relatively low amount of datapoints, we will focus on feature reduction and data normalization to combat overfitting in our experiments section.

4.4. SVM Classification

Support Vector Machines (SVMs) are supervised learning models that are useful for classifying data in a variety of formats. We selected them as one of our potential models because they are capable of training without large datasets and can generalize their predictions well without overfitting [11].

5. Experiments

Below is a discussion of our experiments with each individual model. We discuss how the models were setup, tuned, and analyzed. Below in the "Results" section is a further discussion comparing all models.

5.1. Random Forest

The random forest classifier from sklearn ensemble models used in this project is defined as *RandomForestClassifier(criterion='gini', n_estimators=65)* [13]. 'Criterion' refers to measuring the quality of the decision split to obtain the most information gain in a decision tree and we use Gini Impurity.

Gini Impurity is given by Gini = $1 - \sum(i = 1, n)(1 - (p_i)^2)$, where p_i is the probability of sample belonging to i class, total n classes. The decision tree algorithm considers the feature resulting in lowest Gini Impurity with each split.

Decision trees by default split until they reach a pure node, which indicates that all the samples in the node belong to the same class and can no longer be split. Other ways to split are 'logloss' and 'entropy', which are more computationally expensive than Gini and don't offer additional benefits in this specific project.

N_estimators refers to the number of decision trees used to build the random forest. The value of 65 was the lowest number of trees that could give accurate predictions for this dataset through a Design of Experiments.

Other features of the random forest were set to default as defined in sklearn's Random Forest Classifier.

To measure the accuracy of the random forest classifier, the accuracy score function was used, which computes the fraction of predicted labels that match the given labels exactly in the test set [14]. The accuracy score of the random forest classifier was 1.00 (or 100%), meaning that all the predicted labels matched the original labels exactly. This accuracy score can be visualized by the confusion matrix, which shows that none of the test samples were mislabeled as seen in Figure 2.

Since random forest operates on random sampling of features, it also provides information on most important features that define the data set. This is visualized using the

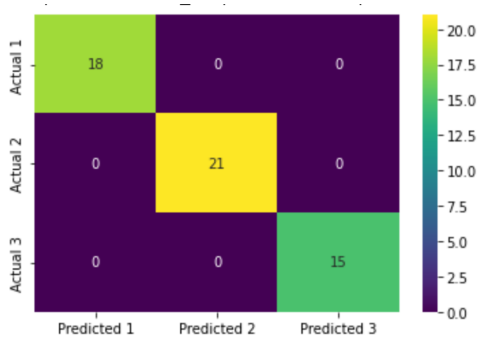

Figure 2. Covariance matrix comparing test dataset labels to predicted labels.

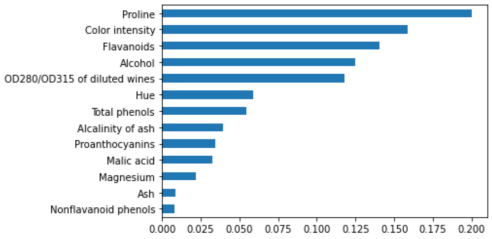

Figure 3. Feature importance from Random Forest

feature importance attribute of the random forest classifier [13]. This attribute reflects on the decrease in the impurity of the node (using Gini index) when using a specific feature to split, as well as the probability of reaching that node, averaged over all trees of the random forest [13].

The feature importance for our data set is shown in Figure 3. This plot shows the most important feature to be 'Proline', and the least to be 'Nonflavonoid phenols'. The importances reported are normalized, so that the sum of importances add up to 1 and the relative importance of each feature can be interpreted. Based on this graph in Figure 3, it can be seen that it is possible to reduce the number of features to describe our wine in order to accurately predict its region. This is beneficial in saving time by necessitating the collection of only the necessary features to predict the region that the wine is from.

5.2. K Nearest Neighbor

Similar to the Random Forest Model, the accuracy of this classifier was determined by the sklearn.metrics function to identify how many of the KNN classified points matched the actual region. The confusion matrix test determined that typically, the wine from region 1 was classified more incorrectly than the other two regions which may be attested to the fact that it has less than 50% support comparatively to the other two.

Since KNN classification depends on a hyperparameter input of the number of neighbors per point, it also

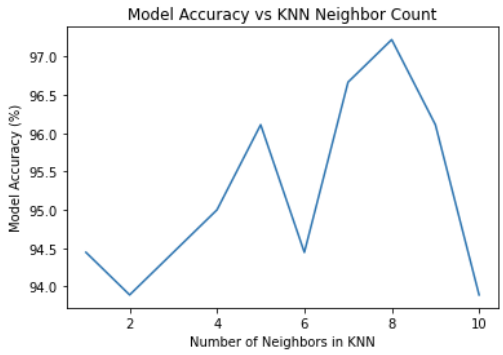

Figure 4. KNN Accuracy vs Number of Neighbors

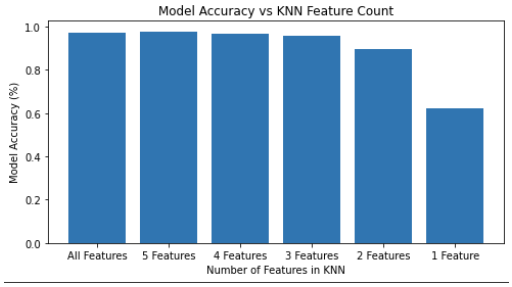

Figure 5. KNN Accuracy of KNN Relative to Feature Count

meant that finding out the importance of neighbor count was highly impactful to the accuracy. For this reason, an analysis was conducted such that multiple trials were run in order to find the ideal count for the highest accuracy. As seen in Figure 4, the highest accuracy typically ranged between 7 and 8 neighbors, which led to the executive decision to compare this model to the others at 7 neighbors specifically.

The next analysis depicts the impact of the feature reduction on the accuracy of KNN classification prediction. When analyzing the difference between the main five features compared to all 13 features, there is practically no difference between the two, sometimes even being more accurate at 5 features relative to all features combined. This can be seen in Figure 4. This was beneficial because it allows us to reduce out necessary features required for predicting the region a wine is from and also identified that the cheaper features to test were the most important, saving time and money

5.3. Multiclass Logistic Regression

While the data set is relatively small, there are still many hyperparameters that can be analyzed within scikit-learn for the logistic regression module [4]. It should be noted that further analysis is provided in Appendix: Logistic Regression that discusses further hyperparameter tuning and analysis.

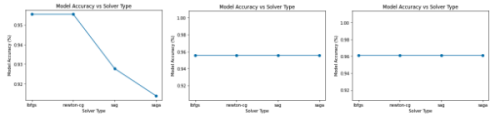

Figure 6. Effects of solver type on the model accuracy. On the left is model training with no feature reduction, in the middle is model training with no feature reduction but normalized data, and on the right is with feature reduction and normalized data.

Due to the small dataset, the effects of solver type and regularization effects were analyzed. These two parameters were selected because they seemed to be the most influential when it came to returning model accuracy values. If data is normalized in a plot, the data is normalized using the Z-score method. Also, any feature-reduced plots go from 13 features to 4. The four features used in this analysis are: Proline, Color Intensity, Flavonoids, and Alcohol content.

The first parameter analyzed was the model solver. Sklearn provides lgfgs (cross-entropy), newton, sag, and saga solvers. To test the solvers the training / test split was held at 80% train 20% test. Also, L2 regularization was used across all solvers and a maximum of 10,000 iterations with convergence of 1E-4 was used. Ten training cycles were run and the accuracies were averaged to form the plot below in Figure 6. On the left is a plot with no feature reduction or data normalization, in the middle is the plot with no feature reduction and data normalization, and on the right is with feature reduction and data normalization.

We can see the effects of the solver type on accuracy is greatly influenced by the normalization of data, with the cross-entropy solver holding the strongest between the simulations. During some simulation runs the newton-cg solver would drop in model accuracy with unnormalized data, whereas the cross-entropy solver held very constant. This study is also reproducible, where the general trend of normalized data not being solver-dependent (whereas unnormalized data was solver dependent) remains similar although the accuracy values may change slightly.

From analysis shown in Appendix: Logistic Regression, it was determined that L2 regularization provides the best model accuracy results across all solvers. Therefore, the next step in analyzing this regularization bias was to look at the effects of the inverse weight term, c, in scikit-learn. To test this value, the training / test split was held constant at 80 / 20 and L2 regularization was used. The cross-entropy solver was also used and a maximum of 10,00 iterations and convergence criterion of 1E-4 was used in the analysis.

From the above Figure 7, we can see that the feature reduction doesn't affect the effects of lowering the value of the inverse weight term. However, when the data is normalized, the model is able to hold higher accuracies for larger values of c. However, when c is decreased past the value

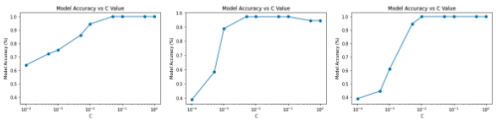

Figure 7. Effect of inverse weight on L2 regularization. On the left is model training with no feature reduction, in the middle is model training with no feature reduction but normalized data, and on the right is with feature reduction and normalized data.

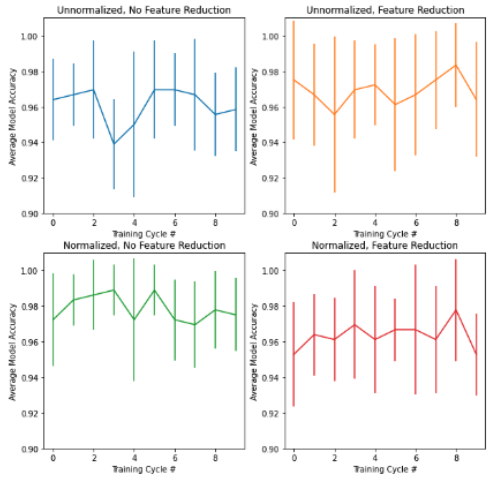

Figure 8. Plots of average model accuracy versus the training cycle number with error bars correlating to standard deviation. Table 1 above denotes the correlation between the x tick labels and the solver type. Top left: Unnormalized features with no feature reduction. Top Right: Unnormalized features with feature reduction. Bottom Left: Normalized data with no feature reduction. Bottom Right: Normalized Data with feature reduction.

of 10E-2, the model accuracy drops off and is much less accurate as compared to the un-normalized data on the plot on the left in Figure 2 for lower values of c. Overall, this was a reproducible result across many simulation runs.

However, as discussed in the Methods section, a prominent shortcoming of multiclass logistic regression is that cross-entropy is a non-linear and non-uniform solver. To test this, we analyzed the standard deviation of the accuracies across training cycles. This is important as it can shot how consistent logistic regression is at predicting the small dataset.

In the above Figure 8 is the average model accuracy versus the overall training cycle. The x axis contains the training cycle number. Each training cycle performed included 10 model fits with a training set correlating to the title of the graph. The error bars represent the standard deviation taken across the 10 model fits in each training cycle. Therefore, overall, 100 model fits and data points were included in this plot.

What we see here is that no matter the data set, the standard deviation can vary greatly from training cycle to training cycle. It seems from Figure 8 that the normalized data set with no feature reduction was the "most consistent" (i.e. had the lowest error bars). However, when analyzing wine that is being sold for millions of dollars, consistency is a very important statistic to measure. Even in the normalized and no feature reduction plot (lower left corner of Figure 8), we see that some training cycles have very high standard deviations (such as training cycle number 4). This serves as further proof that the logistic regression model for this dataset is not good due to the inconsistencies found.

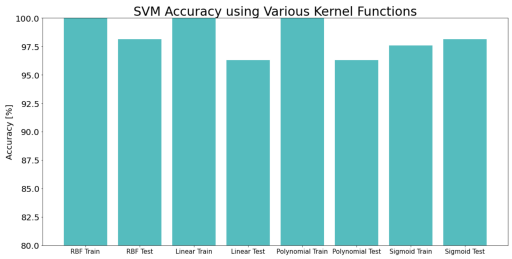

Figure 9. SVM base performance with various kernel functions.

Further analysis with multilinear logistic regression was completed (and is shown in Appendix Logistic Regression). However, from these results it was found that the multilinear logistic regression was not very consistent for the various hyperparameters that scikit-learn allows you to tune. Even with feature reduction and data normalization, logistic regression remained inconsistent at best, and is found to be unreliable for the wine dataset. The model accuracies were fairly random across further simulations as compared to other model types discussed in this paper.

5.4. SVM Classification

In order to create an SVM classifier with optimal performance, we performed a number of trials using various kernel functions and hyperparameters. The results without hyperparameter tuning are shown in Figure 9. In terms of accuracy, we found that the classifier using the RBF kernel had the best performance on the test dataset with an overall accuracy of 98%.

Without hyperparameter tuning, the RBF kernel had the best performance. We believe that this is because the RBF kernel is one of the most "flexible" kernels, able to encapsulate a variety of patterns of data. Therefore, after the train-test split it was still able to generate an excellent fit for any pattern.

To continue improving these models, we performed hyperparameter tuning on a few of them. The one that showed the most improvement was the polynomial kernel. Surprisingly, when the degree of the polynomial kernel function was set to one, creating a linear SVM, the test accuracy became 100% Figure 10. The difference between this and the

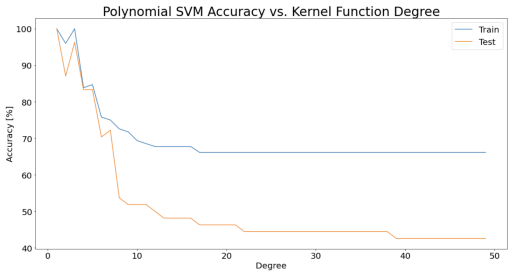

Figure 10. Polynomial SVM accuracy vs. kernel function degree.

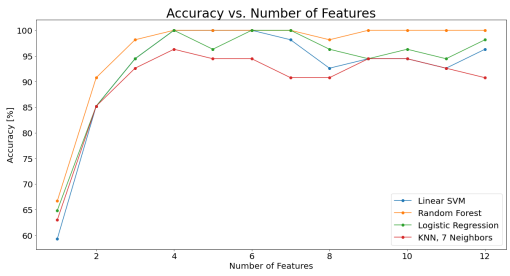

Figure 11. Accuracy vs number of features used to train each classification model.

original linear SVM turns out to be due to a slight difference in the Scipy implementation. For higher degree polynomials, the results rapidly became worse, suggesting that as a general statement a polynomial fit is not suited for the data.

After discovering that this linear SVM implementation performed best, we began experimenting with feature reduction and testing its consistency. Although the performance is quite good, it has some problems with consistency, dropping below 100% accuracy in 36% of runs (randomizing the train-test split each time). This model was eventually chosen to compare against the others. The results of feature reduction and this comparison appear in Figure 11 in Results; although the performance is good, it is again not as consistent as our result using the random forest classifier.

6. Results

The comparison for accuracy of the 4 classification models with respect to the number of features used for training is depicted in Figure 11. The figure shows random forest to be the best classifier with an accuracy score of 1.0. For the random forest model, the accuracy score almost plateaus at 4 features, indicating that just the information on 'Proline', 'Color Intensity', 'Flavonoids' and 'Alcohol features' are sufficient to train an accurate random forest classifier.

These four features are capable of being chemically tested to verify wine origin authenticity. Proline refers to specific amino acids found in wine grapes [15]. The color intensity is a measured wavelength of light through a wine (and is typically 520nm for red wines, and will decrease the longer the wine is aged) [16]. Flavanoids refer to polyphenols found in a wine which are chemical compounds [17]. Alcohol content is a standardized measure of how much alcohol is contained in a given beverage, which for wine is 5 fluid ounces per the NIH [18].

A benefit to these features is that there are readily available tools for purchase to chemically test wine. A coravin could be used to extract a small sample of wine from a bottle to chemically test. A coravin works by using a very thin needle that protrudes through the cork without displacing cork material (and without introducing oxygen to the wine inside the bottle) [19]. This sample of wine could then be tested against the wine cultivars' reported wine grape characteristics to verify the authenticity of the wine. Specifically, since older wines tend to fetch higher prices, the color intensity test will be important as the wavelength of light through the wine can only change with age of wine, which can be very important for analyzing wine [16].

Having fewer features used to predict regions of origin for the wine means that lower cost and time goes into sampling the wine, which would make this model useful for wine auctioneers who want to be vigilant in verifying their sold wine. Shown in Figure 11 we see that the Random Forest model leads to highly accurate analysis with the least amount of features. We determined that four features is enough to verify the wine authenticity based on the input dataset.

7. Conclusions

Overall, this paper analyzed many different supervised machine learning models to fit a small dataset to classify region of wine. Many models, such as KNN, SVM, and Logistic Regression saw some success, but each had their own internal issues. Random Forest predictions is 1) accurate at predicting region of wine, 2) is able to do so with only 4 features, and 3) was reproducible without large variations in the accuracy.

We have learned that individually, these models have varying benefits and drawbacks. We have also developed an understanding (and produced visualizations) on the various hyper-parameters that scikit learn offers, and what their relative affects are on a small dataset [4].

If further developed, this model could be used by wine auctioneers to boast their guarantee of wine authenticity, as wine fraud is a large issue plaguing the industry [1]. In terms of model development, we would be interested to see an expanded dataset, and to see if Random Forest still holds up to be the most optimal. Also, the dataset used in this article was missing some features, and so it might be interesting to see if there are other reported features from cultivars that could "plug in" to these models and develop reproducible

results [3].

Experimentally, we think it would also be interesting to have wine auctioneers test wines with coravins and readily available testing to verify if wines actually match their reported cultivar values in the training dataset.

8. Author Contributions

A.V.: Developed multiclass linear logistic regression model, writing paper, visualizations, data curation

A.E.: Developed random forest model, writing paper, visualizations, data curation

M.L.: Developed random forest model, writing paper, visualizations, data curation

S.G.: Developed K nearest neighbor model, writing paper, visualizations, data curation

9. Supplemental Information

Supplemental appendices are attached to this report in a separate file. The appendices contain further information as to the design and hyper-parameter tuning of each model. Also attached is source code for each of the models.

References

[1] International Journal of Wine Research Vol. 2, pp. 105-113, Holmberg, "Wine Fraud."

[2] Micallef, "What's In Your Cellar?"

[3] "UCI Machine Learning Repository: Wine Data Set

[4] API design for machine learning software: experiences from the scikit-learn project, Buitinck et al., 2013.

[5] G. Thimm and E. Fiesler, "High-order and multilayer perceptron initialization," in IEEE Transactions on Neural Networks, vol. 8, no. 2, pp. 349-359, March 1997, doi: 10.1109/72.557673.

[6] Thimm G, Fiesler E. Optimal setting of weights, learning rate, and gain. IDIAP; 1997.

[7] Fischer, Igor, and Jan Poland. "Amplifying the block matrix structure for spectral clustering." Proceedings of the 14th annual machine learning conference of Belgium and the Netherlands. Citeseer, 2005.

[8] https://developers.google.com/machine-learning/data-prep/transform/normalization

[9] Lecture 9,Amir Barati Farimani, Chaaran Arunachalam, Nalini Jain

[10] Lecture 6,Amir Barati Farimani, Chaaran Arunachalam, Nalini Jain

[11] Lecture 17,Amir Barati Farimani, Chaaran Arunachalam, Nalini Jain

[12] Lecture 14,Amir Barati Farimani, Chaaran Arunachalam, Nalini Jain

[13] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.Ran

[14] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accura

[15] Long D, Wilkinson KL, Poole K, Taylor DK, Warren T, Astorga AM, Jiranek V. Rapid method for proline determination in grape juice and wine. J Agric Food Chem. 2012 May 2;60(17):4259-64. doi: 10.1021/jf300403b. Epub 2012 Apr 20. PMID: 22480274.

[16] Pérez-Caballero, V., F. Ayala, J.R. Echávarri, and A.I. Negueruela. 2003. Proposal for a new standard OIV method for determination of chromatic characteristics of wine. Am. J. Enol. Vitic. 54:59-62.

[17] Fernandes I, Pérez-Gregorio R, Soares S, Mateus N, de Freitas V. Wine Flavonoids in Health and Disease Prevention. Molecules. 2017 Feb 14;22(2):292. doi: 10.3390/molecules22020292. PMID: 28216567; PMCID: PMC6155685.

[18] "What Is A Standard Drink? — National Institute on Alcohol Abuse and Alcoholism (NIAAA)."

[19] "Coravin Wine System — Coravin Wine Preserver — How to Use Coravin."