Identification of Red/White Wine Types Using Machine Learning Algorithms

Garson Chow and Mei Shin Lee
CS-UY 4563 Intro to Machine Learning
Final Project

https://github.com/garsonbyte/Machine-Learning-Project

1 Introduction

Roughly 60% of Americans claim to enjoy an occasional glass of wine, whether it's at a special dinner or at home. Two of the most popular wines are red wine and white wine, which more than 60% of wine consumers drink.

Both red and white wines, while strikingly different in appearance, have similar distributions of pH, sulfuric oxides, chlorides, and other elements. The goal of our project is to apply machine learning algorithms to correctly classify a red or white wine sample and to predict wine quality. A dataset provided by UCI was used for our statistical analyses. Preliminary steps included preprocessing the data into the desired form, and then applying, PCA (Principal Component Analysis), an unsupervised learning algorithm to find trends in our unstructured data. Afterwards, we implemented three supervised learning algorithms (Polynomial Regression, Logistic Regression, and Support Vector Machines) to test the accuracy of the predicted results.

Another goal of our project was to illustrate the effect of regularization and feature transformation on our data. Lasso (L1) and Ridge (L2) regularization were applied to the linear and logistic regression models. Three different kernels were used and compared within our SVM model.

2 Overview of Data

The UCI dataset contained 6497 records. There were 1599 red wine samples and 4898 white wine samples. The unequal distribution of wine types will be taken care of when accessing the efficiency of each learning algorithm described in the sections below. Since our data is unevenly distributed, confusion matrix results for both logistic regression and SVM are displayed in its original and normalized form for comparison.

Wine Type	**Number of Samples**
Red Wine	1599
White Wine	4898
Total	6497

Number of Red/White wine samples

The dataset contained eleven identifying features and two target variables (wine quality and wine type). An outline of feature and target variables is shown below.

# 0-12	**Column**	**Non-Null Count**	**Data Type**
0	fixed acidity	6497 non-null	float
1	volatile acidity	6497 non-null	float
2	citric acid	6497 non-null	float
3	residual sugar	6497 non-null	float
4	chlorides	6497 non-null	float
5	free sulfur dioxide	6497 non-null	float
6	total sulfur dioxide	6497 non-null	float
7	density	6497 non-null	float
8	pH	6497 non-null	float
9	sulphates	6497 non-null	float
10	alcohol	6497 non-null	float
11	quality	6497 non-null	int
12	Red(0)/White(1)	6497 non-null	int

Dataset Features and expected values

3 Linear Regression

A linear regression model was created to predict the quality of white wine samples. Red wine samples were not used to train or test the linear model due to slight differences in wine properties. Such properties might skew the estimated wine quality prediction for white wine.

A total of 4898 samples was used to test and train the model. The wine samples were re-distributed with a train to test ratio of 0.8 to 0.2. The distribution of samples is shown in the table below.

	Number of Samples
Train Ratio (80%)	3918
Test Ratio (20%)	980

Test/Train split for the Linear Regression model

The linear regression model was built from Python's sklearn library. It yielded the following results:

RSS	**TSS**	$\mathbf{R^2}$
558.38351	757.31734	0.26268

Results from Linear Regression Model

From the results, as shown above, the model had a high RSS value, which meant that the model was not able to find a clear linear trend between predictive features and wine quality rating. The R^2 value of the model is 0.26268, which means that only 26.268% of the variance in wine quality is explained by the features used in the model.

In hopes of better results, a modified version of the linear regression model was created. We can note that wine quality ratings are integers on a scale of 1-10. Each record's prediction was rounded to the nearest whole number. For instance, a predicted rating of 4.38 will be rounded down to a 4, while a predicted rating of 4.78 will be rounded up to a 5. The accuracy was used as a measure of efficiency. The results of the new algorithm are shown below.

Predicted Correctly	Predicted Incorrectly	Accuracy
520	460	0.53061

Results of modified Linear Regression algorithm

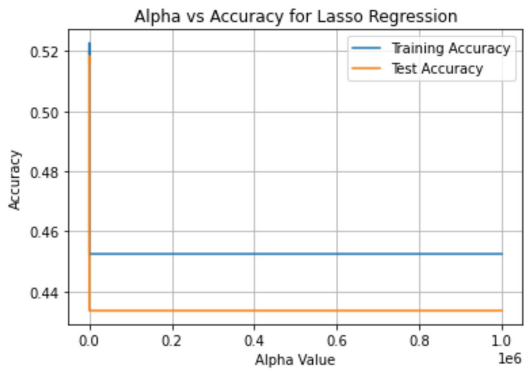
A probable cause of poor results may have been due to overfitting. Two types of regularization, Lasso and Ridge, were further added to the linear regression model to prevent overfitting. Different alpha values (penalty terms) were tested on the modified linear regression model. The accuracy rates are noted in the table below.

Alpha	Accuracy
0	0.52244
0.001	0.52244
0.01	0.52142
0.1	0.46224
10	0.43673
100	0.43673
10000	0.43673
1000000	0.43673

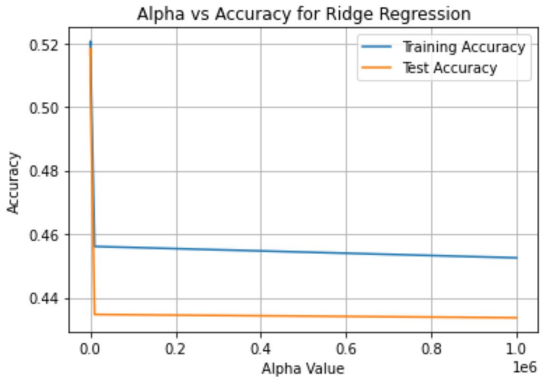
Lasso regularization with modified linear regression model

Alpha	Accuracy
0	0.52244
0.001	0.52244
0.01	0.52346
0.1	0.52346
10	0.52142
100	0.51836
10000	0.44081
1000000	0.43673

Ridge regression with modified Linear Regression Model



Lasso Regularization with Linear Regression Model.



Ridge Regularization with Linear Regression Model.

The relationship between alpha values and accuracy were plotted. As we increased the cost of the penalty (alpha) term, the accuracy decreased. This is an explainable

trend, since increasing the penalty term will decrease the variance of the model. Decreasing variance helps make the model more generalizable, so that it doesn't model the noise in our data, but the overall trends itself.

A polynomial transform of degree 5 was applied to our data in an attempt to improve our model's performance. The goodness of fit value was 0.280, which means that 28% of the variance was explained by the features. Next, we rounded the predicted target values to the nearest integer and found the accuracy to be 51.4%. This proves that polynomial transform is not suitable for this dataset. A probable cause of such results could have been due to an overly complex model, which leads to overfitting.

A Bayesian linear regression with ridge regression model was used as a feature transformation to predict wine quality. Unlike traditional linear regression, this model uses the concept of Bayesian inference and normally distributed error terms to determine the weights of the coefficients. Our implementation was run once to find the RSS, TSS, and R^2 values. Then, using our modified algorithm with rounding for our final predictions, the accuracy of the predictions was calculated.

RSS	**TSS**	**R^2**	**Accuracy**
2247.14110	3067.92547	0.26753	0.52244

Results of Bayesian Linear Regression on White Wine

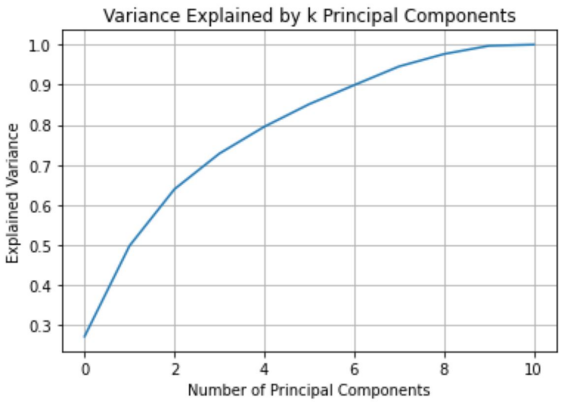
Overall, the modified linear regression algorithm gave us the best accuracy rates. But prior to modified a traditional linear regression for better results, it can be noted that the R^2 values from both linear and Bayesian regression were roughly the same, around 0.26. When both algorithms were modified to round up or down to the nearest integer value for wine quality, the accuracy for the modified traditional linear regression algorithm was slightly better than the one for Bayesian linear regression. Linear regression, in general, did not give us very ideal results, which implies that wine quality can not be expressed in terms of its eleven features. In many cases, wine quality is subjective, meaning that different people have different tastes, and in turn, prefer a specific type or brand over another. Therefore, wine quality is hard to predict when a person's mood, taste, and perceptions vary greatly.

4 PCA, Principal Component Analysis

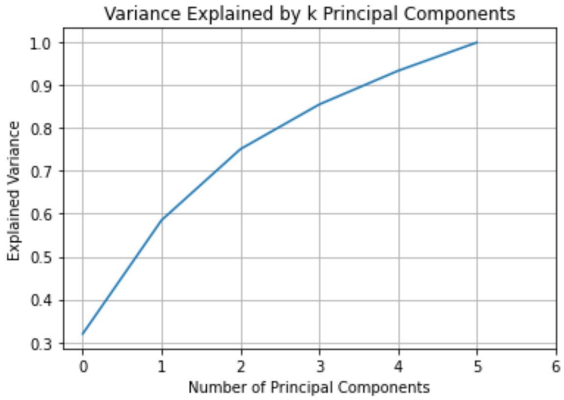
After processing and eliminating the wine quality column from the feature space, there were 11 features remaining to train our models. To reduce the high dimensionality of our dataset and to avoid overfitting, we applied the Principal Component Analysis (PCA) technique to our feature vector. As we reduce the dimensionality of the feature space, we gain the ability to better generalize the model.

We performed PCA to determine the optimal number of principal components to contain 85% of the explained variance. In the example below, we transformed the feature vector from an 11-dimensional plane to a 6-dimensional plane whilst keeping 85% of the variance. The smaller feature space will reduce future modeling computations and allows us to understand the data in a more simplified form. In effect, we are losing some information about the dataset.

100% of the data's variance was explained using 10 principal components without transformation. For testing purposes, we kept 85% of the explained variance. After transforming with PCA, 100% variance was explained in about 5 components.



Explained variance as a function of the number of principal components before transforming



Explained variance as a function of the number of principal components after transforming

There wasn't an explicit change in the structure of the data after performing PCA. Regardless, we applied the transformed feature space to Logistic Regression and SVM modeling.

With applying logistic regression on the transformed feature vector, the accuracy score, precision score, recall score, and f-score were recorded in the table below for different c-values (inverse of regularization).

c-value	**Accuracy**	**Precision**	**Recall**	**Fscore**
0.0001	0.73230	0.73230	1.0	0.84547
0.001	0.53769	0.66762	0.73424	0.69934
0.01	0.45692	0.63057	0.62394	0.62724
0.1	0.44538	0.62459	0.60819	0.61628
1	0.44307	0.62337	0.60504	0.61407
10	0.44307	0.62337	0.60504	0.61407
100	0.44307	0.62337	0.60504	0.61407

Logistic Regression with PCA

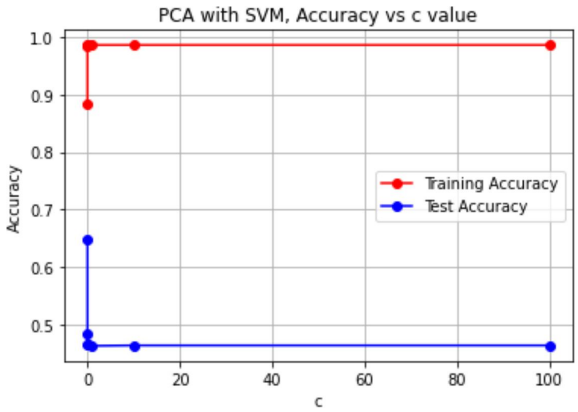
The best accuracy after using the logistic regression algorithm came from adding more regularization (smaller c-value). Regularization decreases variation, and the model should generally classify better for testing sets. Moreover, the F-score was significantly higher with the smallest c-value. This is due to the fact that both the precision and recall were higher. Meaning, more positive results were identified correctly. Similarly, after applying the SVM algorithm with a PCA transformed feature space, the most accurate predictions came from an iteration of the smallest c-value. The results are shown in the table below as well.

c-value	**Accuracy**	**Precision**	**Recall**	**Fscore**
0.0001	0.71	0.72619	0.96953	0.83040
0.001	0.46076	0.63252	0.62920	0.63085
0.01	0.44076	0.62214	0.60189	0.61185
0.1	0.43923	0.62132	0.59978	0.61036
1	0.44230	0.62296	0.60399	0.61333
10	0.43923	0.62132	0.59978	0.61036
100	0.43923	0.62132	0.59978	0.61037

SVM with PCA (Linear Kernal)

To visualize how regularization affects the accuracy for both modeling techniques, the accuracies as a function of the c-values were plotted for both. The graphs are shown below.

Graph of the accuracy as a function of c-values (PCA with Logistic Regression)



Graph of the accuracy as a function of c-values (PCA with SVM)

As we increased the c-values (reducing regularization), the training accuracy increased whereas the testing accuracy decreased. This pattern applies to both Logistic Regression and SVM algorithms. Thereby, we found better results with regularization as it decreases variation in the validation set.

5 Logistic Regression

In an attempt to perform classification on the wine dataset, the logistic regression algorithm was used to train our model. With two separate datasets, one being for red wine and the other for white wine, we combined both sets into a single CSV file whereby an additional column was created for our class variable: Red(0)/White(1).

We found that there were some attributes that had more or less the same values regardless of the type of wine. In effect, these attributes could not be included as factors to determine whether a sample was of type red or white. As an example, the density of red and white wine are roughly the same. The density values range from $0.994 - 1.001$. The differences in the densities between red and white wine are too minuscule to provide any deterministic values. Moreover, the alcohol contents

are similar as well. All drinks categorized under wine should contain $11 - 13\%$ alcohol by volume; thereby alcohol content should not be included in the feature vector.

The dataset for classification contained eight identifying features. The features used to perform logistic regression are shown below.

# 0-7	**Column**
0	fixed acidity
1	volatile acidity
2	citric acid
3	residual sugar
4	chlorides
5	free sulfur dioxide
6	total sulfur dioxide
7	sulphates

Dataset Features for Logistic Regression

A total of 6497 samples was used to train and test the model. For one round of cross-validation, the wine samples were shuffled with a train ratio of 0.8 and test ratio of 0.2. The distribution of samples is shown in the table below.

	Number of Samples
Train Ratio (80%)	5197
Test Ratio (20%)	1300

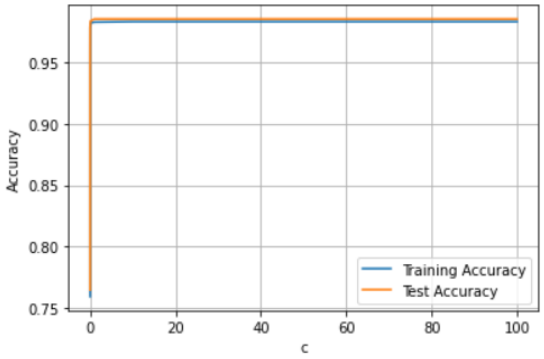
Distribution of samples for Logistic Regression for one round cross validation

At first, the logistic classifier was fit under a linear model without regularization ($c = 10^8$) and the model was trained using the modified training set. In all cases, *sklearn* applies regularization to all logistic regression models where the c-value is set to 1. To simulate a logistic regression classifier without regularization, the c parameter must be set to a large value. In our case, the c value was set to 10^8. Using that model to predict the classes for the testing set, we ended up with an accuracy of 98.5% with an F-score of 98.9%. The underlying results were more than satisfactory. For possible improvements to reduce overfitting and variation, regularization was

added to the cost function. Both Ridge and Lasso regularization were applied to the linear models.

5.1 Ridge Regularization

For logistic regression with ridge regularization (L2), different c-values were tested to determine how varying regularization levels affect the model's overall accuracy. After fitting the logistic regression model with the training set and allowing varying c-values for the "L2" penalty, the following accuracy scores, precision scores, recall scores, and F-scores were recorded below.



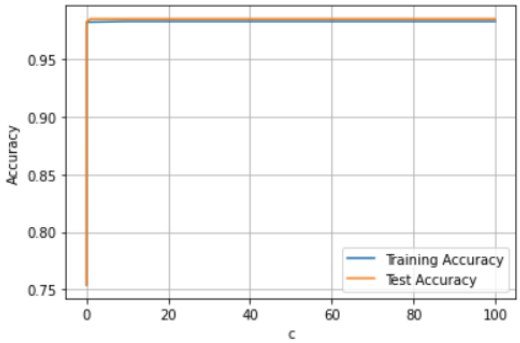
c-value	Accuracy	Precision	Recall	Fscore
0.0001	0.76465	0.76205	1.0	0.86496
0.001	0.96230	0.95682	0.99489	0.97548
0.01	0.97923	0.98375	0.98877	0.98625
0.1	0.98384	0.99179	0.98673	0.98925
1	0.98538	0.99383	0.98673	0.99027
10	0.98538	0.99383	0.98673	0.99027
100	0.98538	0.99383	0.98673	0.99027

Plot of Accuracy vs C value and table of results for a Logistic Regression with Ridge

As the regularization added to the cost function decreases (larger c-value), the more accurate predictions the model outputs. The best resulting scores came when $c > 1$.

5.2 Lasso Regularization

Similarly, the logistic regression model with lasso regularization was tested with varying levels of regularization (c-values). The penalty was set to "L2" and the following accuracy scores, precision scores, recall scores, and F-scores were recorded in the table below.



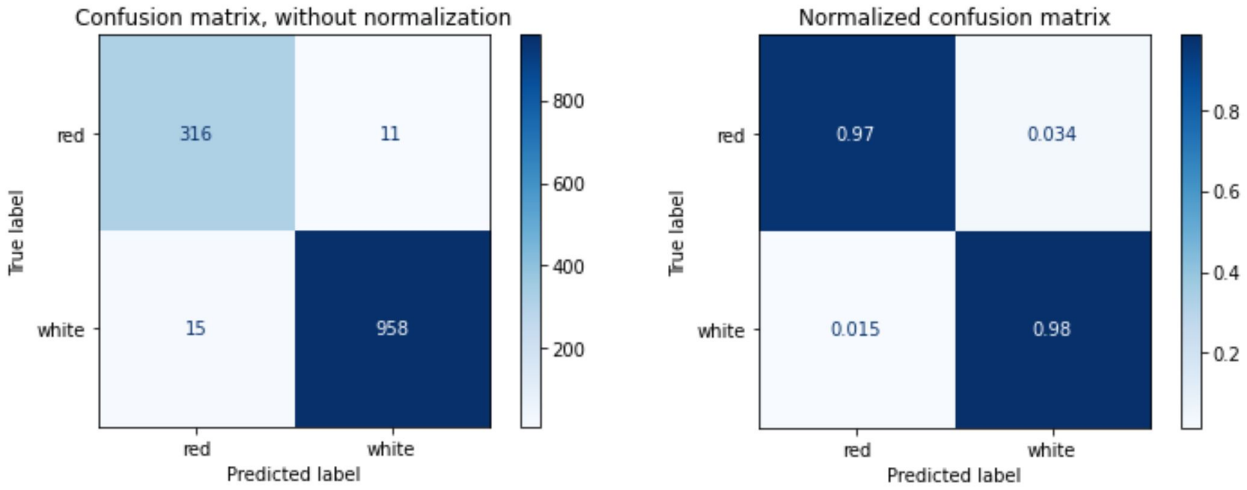
c-value	Accuracy	Precision	Recall	Fscore
0.001	0.80691	0.79658	0.99897	0.88637
0.01	0.97461	0.98071	0.98571	0.98320
0.1	0.98307	0.99077	0.98673	0.98875
1	0.98538	0.99383	0.98673	0.99027
10	0.98538	0.99383	0.98673	0.99027
100	0.98538	0.99383	0.98673	0.99027

Plot of Accuracy vs C value and table of results for a Logistic Regression with Lasso Again, as the level of regularization on the model decreased (larger c-value), the more accurate predictions the model outputted. The best resulting scores came when $c > 1$.

5.3 Overall Results for Logistic Regression

The data presented within our dataset was skewed heavily towards white wine samples. To address the skewness, the results were normalized prior to calculating final percentages for the confusion matrix. In this case, the confusion matrix was created from the logistic regression model without regularization.

The left matrix displayed the raw values of the error metrics with classification: true positives, false positives, true negatives, and false negatives. From the given values, there were only 11 out of 327 red wines that were misclassified and 15 out of 973 white wines that were misclassified. However, there are almost three times more samples of white wine than those of red. Accordingly, a normalized confusion matrix was created for the imbalance of classes to keep a more visual interpretation of the particular class that is being misclassified. From the normalized matrix, 97% of the red wines were classified correctly and 98% of the whites were also classified correctly. Since we had more white wine examples to train the logistic regression classifier, it makes sense that it will produce better results. As the number of data points increases, the lower the out-of-sample error will be.



Confusion matrices for Logistic Regression model, normalized and unnormalized results

The best accuracy scores and F-scores came from the iterations where $1 < c < 100$. For $c < 1$ the accuracy score was measurably lower using the "L1" or "L2" penalty than without regularization. This meant that adding some regularization where $\lambda = \frac{1}{c}$ would indeed lower variation, however, when adding too much regularization ($c < 1$), there would be a trade-off between variation and bias. If you add higher levels of regularization, you would decrease variation significantly, but lose some of the general trends of the data. It is probable that the accuracy decreased due to that reasoning. When comparing both ridge and lasso regularization, the latter technique had better results when $c < 1$. Otherwise, both regularization metrics more or less had the same accuracy scores and F-scores. Overall, as the strength of the regularization term increased, the lower the accuracy was.

6 SVM

SVM, or the support vector machine supervised learning method, was used to classify the wine type of a specific sample. Similar to our procedure in logistic regression, the data was split into train and test groups, with an 80:20 percent train to test ratio was. The new distribution of samples for generating the SVM model is shown below.

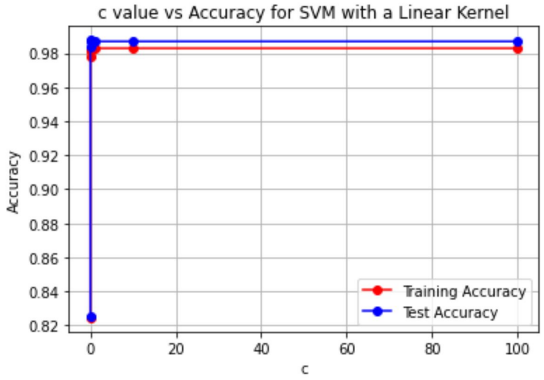
	Number of Samples
Train Ratio (80%)	5197
Test Ratio (20%)	1300

Distribution of samples after a test/train split

Three different kernels were used with the SVM method. Models were created with linear, RBF, and polynomial kernels. Additionally, regularization parameters were varied throughout the model. A larger c-value results in lower regularization and a higher c-value results in higher regularization. The tables below list the accuracy, precision, recall, and f-score rates associated with each c value. The f-score value is used to determine the efficiency of the SVM algorithm since the dataset is skewed towards having more white wine samples.

6.1 Linear Kernel

The c-value that produced the best results was 1. A plot depicting the relationship between c-value and accuracy was also created. As the c-value increased (lesser regularization), the accuracy of the model increased.

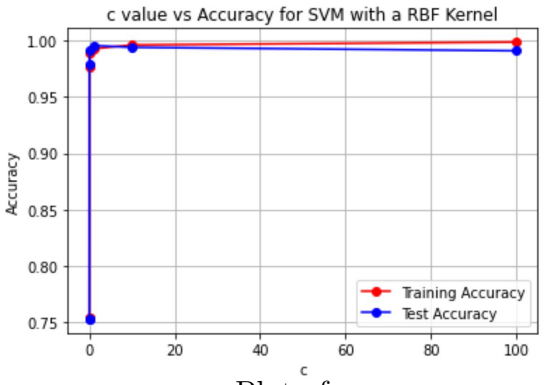


c-value	Accuracy	Precision	Recall	Fscore
0.0001	0.82846	0.81490	0.99794	0.89718
0.001	0.97384	0.97477	0.99076	0.98270
0.01	0.97923	0.98565	0.98666	0.98616
0.1	0.98230	0.98870	0.98769	0.98819
1	0.98461	0.99176	0.98769	0.98972
10	0.98384	0.99175	0.98666	0.98920
100	0.98384	0.99175	0.98666	0.98920

Plot of accuracy vs c value and table of results for a linear kernel

6.2 RBF Kernel

The c-value that produced the best results was 10. A plot depicting the relationship between c-value and accuracy was also created. As the c-value increased (lesser regularization), the accuracy of the model increased.

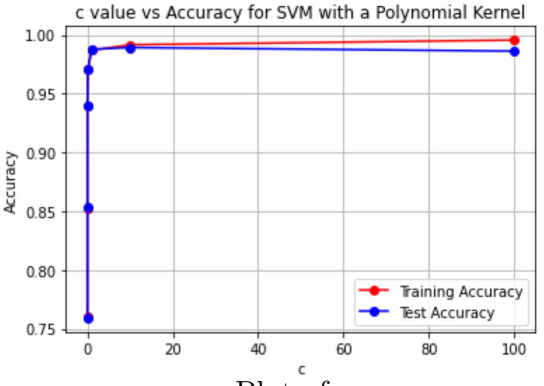


c-value	Accuracy	Precision	Recall	Fscore
0.0001	0.76692	0.76692	1.0	0.86808
0.001	0.76692	0.76692	1.0	0.86808
0.01	0.97538	0.97536	0.99297	0.98409
0.1	0.98538	0.988023	0.99297	0.99049
1	0.98769	0.99099	0.99297	0.99198
10	0.99	0.992	0.99498	0.99349
100	0.98923	0.99199	0.99398	0.99298

Plot of accuracy vs c value and table of results for a RBF kernel

6.3 Polynomial Kernel

The c-value that produced the best results was 1. A plot depicting the relationship between c-value and accuracy was also created. As the c-value increased (lesser regularization), the accuracy of the model increased.



c-value	Accuracy	Precision	Recall	Fscore
0.0001	0.77230	0.77107	1.0	0.87074
0.001	0.85230	0.83851	1.0	0.91216
0.01	0.93692	0.92479	0.99899	0.96046
0.1	0.96615	0.96038	0.99699	0.97834
1	0.98461	0.98510	0.99498	0.99001
10	0.98462	0.98607	0.99398	0.99000
100	0.98385	0.98703	0.99198	0.98949

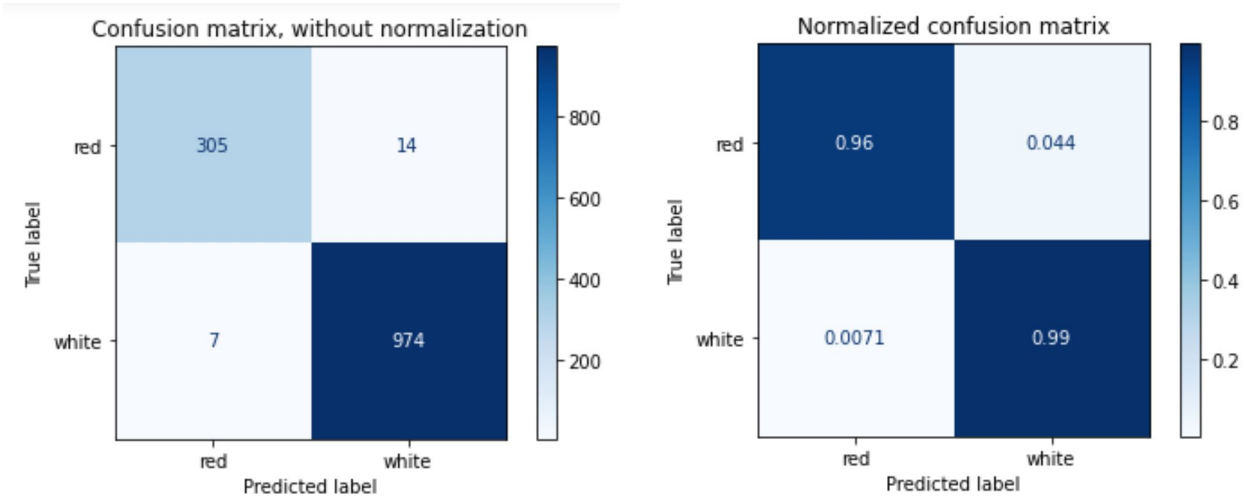
Plot of accuracy vs c value and table of results for a RBF kernel

6.4 Best Overall Results for SVM

An overall trend that was demonstrated throughout all three kernel methods is as the c-value increased, the f-score values increased. $c = 10$ generally provided the best f-score values. As the c-value increased, the variability within the model increased since regularization was decreased. The three kernels presented us with equally good

results.

Two confusion matrices were created for the results generated from a linear kernel. The first confusion matrix displays the raw numbers that fell into each category, which includes, true positive, false positive, false negative, and true negative. From the outline of the results, one can deduce that raw values will not be a good measure of the accuracy of the algorithm. Therefore, normalization was performed on top of the generated values. The normalized confusion matrix displays the percentage of each type of wine that was predicted correctly and incorrectly.



Confusion matrices for linear SVM kernel - normalized and unnormalized results

From the normalized confusion matrix, 99% of all white wine samples in the dataset was predicted correctly, while only 96% of all red wine samples in the dataset was predicted correctly. This discrepancy stems from the fact that we have more white wine samples in both our train and test sets than red wine samples. This demonstrates the observation that as we increase the number of training samples within our dataset, the better our predictions will be.

7 Conclusion

7.1 Principal Component Analysis

We had a total of 11 features with the goal of reducing the number of dimensions needed for computations. The expected explained variance was set to 85%, which resulted in five principal components. From our transformed data points, there was no explainable structure. A logistic and SVM model was created using our transformed features, but the results were not ideal without regularization. (Refer to PCA section).

7.2 Linear Regression

Our dataset was not well suited for a linear regression model. From the results generated from the traditional linear regression model and the Bayesian linear regression model, both had very poor goodness of fit measures. This implies that there is no clear linear trend between the features and the overall result.

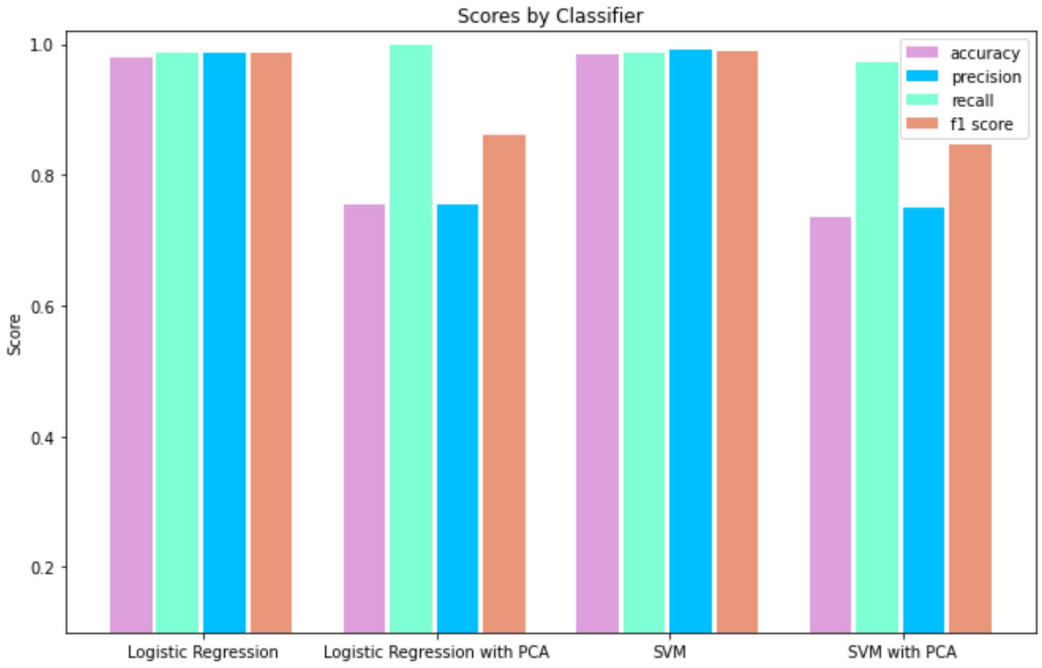
We hypothesize that the features independently did not contribute much to the overall wine quality. Since only about 26% of the variance was explained by the features, the remaining 74% of variance must be caused by external factors. We suspect that some individuals may be biased towards certain brands, different wine types and grape varieties. It is said that white wine has a sweeter taste compared to red wine, so if the rater prefers a sweeter taste, they might gravitate towards giving better ratings to white wine.

Additionally, our dataset was heavily populated with wine ratings that ranged from $5 - 8$, which made up 4710 out of 4898 samples (96%). There weren't any high or low rated wine samples, it was harder for the model to predict any low or high rating values. Since our data were mostly clustered around this range, this resulted in poorer predictions, which in turn led to an overall lower accuracy rate.

7.3 Logistic Regression and SVM

Unlike linear regression which operates within a continuous range, logistic regression and SVM work to classify samples into discrete values. The logistic regression and SVM models presented a relatively high accuracy score. When regularization was added to the models, the accuracy decreased. We noticed that accuracy is not a

good measure for our dataset since our classes were imbalanced. F-score was calculated from precision and recall rates, which was a more reliable estimate. When we added regularization to our model it did not improve our results (refer to Logistic Regression and SVM section).



Comparison between PCA applied data vs non-PCA applied data

To illustrate the difference between PCA applied data versus non-PCA applied data, we calculated the scores by classifiers and compared the results using a bar graph. As shown, applying PCA to reduce data dimensionality decreased the accuracy and precision rates significantly. Since the explained variance were lowered by the PCA transformation, the accuracy and precision was also lowered.