ENGR-UH 4560

Selected Topics in Information and Computational Systems: Machine Learning

Name: Nishant Aswani

Net ID: nsa325

Assignment Title: Mini-Project 2

Wine Classification and Support Vector Machine (SVM)

Nishant Aswani, nsa325@nyu.edu

Selected Topics in Information and Computational Systems: Machine Learning(ENGR-UH 4560), Professor Hwasoo Yeo

1 Introduction

Multiclass classification is a traditional problem, involving labeling a given data point into discrete categories.

Support Vector Machines (SVM), a common model of classification, operates on the concept of finding a hyperplane that best separates training examples while maintaining a large margin [1]. Ideally, the classification model uses a soft-margin, allowing for a slack, for which the model is penalized.

2 Methodology

2.1 Understanding the Data

The data is a chemical analysis of three types of wines with 13 features. The histograms of each feature among the three types were plotted (Appendix A) to gain an empirical understanding of the important features. Looking at the figures in Appendix A, it is clear that certain features distinguish between the three types better than others, namely 'alcohol', 'flavanoids', and 'total-phenols'. On the other hand, there are other features which may be helpful in binary classification. For example, the 'proline' feature may help determine if a datapoint is Type 0 or not.

Because the SVM model uses a "One vs. Rest" (OVR) model, it was decided not drop any features. In fact, keeping the features positively contributes to the model because certain features (such as 'proline') can be used to solidify classification of at least 1 type.

2.2 Data Preprocessing

It was quickly verified that the dataset was not missing any values, as all 178 values were available for the 13 features. The dataset was then saved into a dataframe and was split into 70% training and 30% testing data.

Data scaling plays an important role in training SVM models: SVM assumes that attributes with a larger range play a more dominant role [2]. This potential bias can be visualized with Figure 1. On the left, without scaling, 'alcohol' would have dominated the remaining features simply because it exists within a larger domain. Min-Max scaling, however, scales all the features to be within the same domain, while maintaining the natural skew. As a result, the model remains largely unbiased and still has access to the natural density of each feature (see right graph in Figure 1).

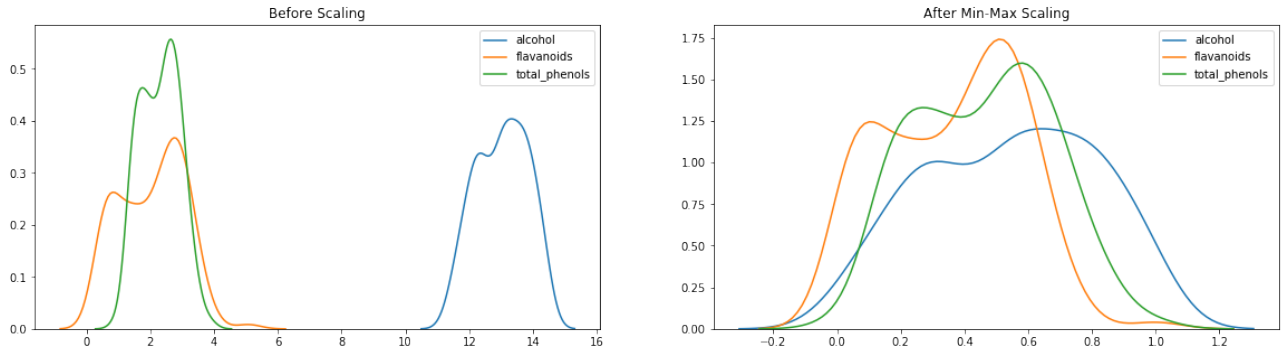


Figure 1: Before (a) and after standardization (b) of the features. Only a select few features are shown.

2.3 Setting a Baseline

Due to the comparative nature, it was decided to set a baseline model to which the modified SVM would be compared against:

```
clf = SVC(kernel='rbf', gamma='auto')
```

Often used as a "reasonable first choice" [2], the model was given an 'rbf' kernel. Here, C defaults to 1, while γ defaults to $\frac{1}{numFeatures} = \frac{1}{13}$. Since the baseline was meant to be a barebones classifier, there was no data scaling involved.

To discuss the hyperparameters, γ controls the maximization of the margin in a SVM [1]. However, as SVM ideally allows for some leeway, C is introduced as a parameter to control how harsh the margin is, essentially controlling overfitting vs. underfitting [1]. Here the model is penalized everytime a datapoint is given some slack.

2.4 Modifying the SVM

As described in Section 2.2, a MinMax scaler was used to remove the feature bias, which was predicted to improve the accuracy. The SVM also used a linear kernel to test the model for how it would perform. Once again, C was set to 3, while γ to $\frac{1}{13}$.

It was also realized that sklearn has another library specifically for the case of linear support vector classification. This arises because sklearn developed two separate interfaces for two popular optimization libraries. The SVC model minimizes the regular hinge loss function, while the LinearSVC model minimizes the squared hinge loss function, among other minor differences in implementation.

For curiosity, the LinearSVC was also trained with C=3. For a fair comparison, the SVC with kernel=linear was changed to use the "one vs. all" strategy.

```
linear_svm_clf = Pipeline([
                ("scaler", preprocessing.MinMaxScaler()),
                ("svm_clf", SVC(kernel='linear', gamma='auto',
                probability=True, C=3, decision_function_shape='ovr'))
                ])

built_linear_svm_clf = Pipeline([
                ("scaler", preprocessing.MinMaxScaler()),
                ("svm_clf", LinearSVC(C=3))
                ])
```

2.5 Using GridSearch to Confirm Parameters

The GridSearch library conducts an exhaustive search of all the provided parameters and was used as a method to confirm whether the handpicked parameters were the best ones or not. The following parameters were provided to GridSearch with k=5 cross validation.

```
Cs = [0.001, 0.01, 0.1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
gammas = [0, 0.0001, 0.0002, 0.0005, 0.0007, 0.001, 0.01, 0.1, 1]
shapes = ['ovr', 'ovo']
kernels = ['linear', 'rbf']
```

2.6 Performance Measurement

Several performance tools were used to compare the championing SVM with the baseline SVM: confusion matrix, classification report, precision recall (PR) curve, and the receiver operating curve (ROC) curve.

The confusion matrix simply shows the number of samples predicted in each bin compared to their actual classification. The classification report extracts information such as precision, accuracy, and recall from the confusion matrix.

The ROC curve plots the true positive rate (TPR) against the (FPR); the former refers to how good the model is at predicting the correct classification, while the latter refers to the tendency the model to incorrectly classify a sample [3]. The PR curve is more useful in cases of imbalanced data, where it looks at how well the model is at classifying points in the minority class [3]. Both these curves were obtained with k=5 cross-validation, which splits the data into 5 groups, using all combinations of test and train to obtain performance metrics.

3 Results

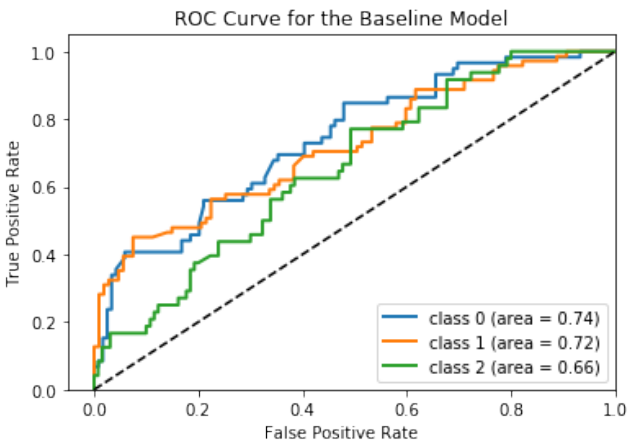
3.1 Baseline SVM

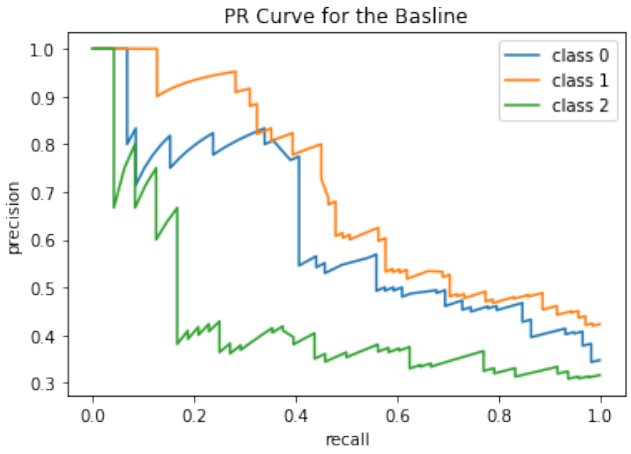
The table below summarizes the confusion matrix for the baseline SVM. We see that the baseline model incorrectly favored classification for Type 1.

	0	**1**	**2**
0	3	14	0
1	0	23	0
2	0	13	1

Table 1: Confusion matrix for the baseline model

The graphs below shows the ROC and PR curves for the baseline model





After 500 runs, each with a different train/test split, we see the following accuracy score for the baseline model:

```
k=5 Cross Validation Mean: 0.4277
k=5 Cross Validation Var: 0.0009
```

3.2 Manually Modified SVM

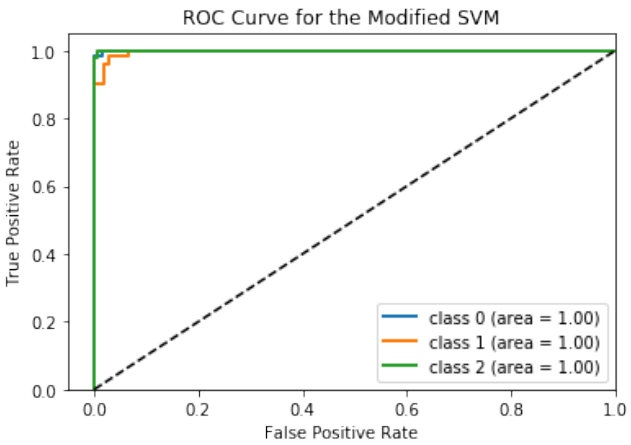
The SVC library with kernel=linear was used for the modified svm, as the LinearSVC model did not allow for probability values.

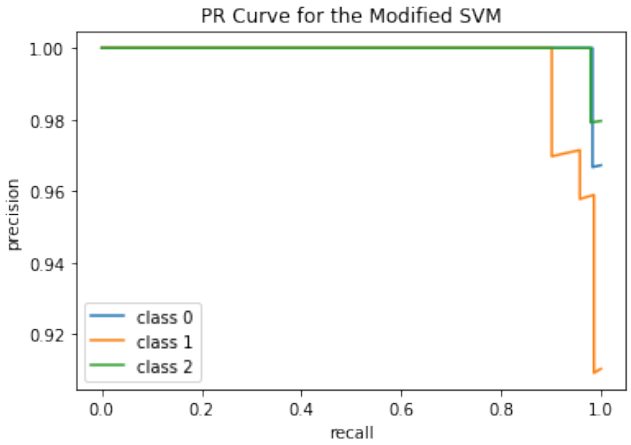
The table below summarizes the confusion matrix for the modified model; we see that the confusion matrix results in a perfect classification for this specific split.

	0	**1**	**2**
0	17	0	0
1	0	23	0
2	0	0	14

Table 2: Confusion matrix for the baseline model

The graphs below shows the ROC and PR curves for the modified SVC model.





Once again after 500 runs, each with a different train/test split, we see the following accuracy score for the baseline model:

```
k=5 Cross Validation Mean: 0.9663
k=5 Cross Validation Var: 0.0005
```

3.3 GridSearchCV Determined SVM

Unsurprisingly, the GridSearch method was able to find the ideal parameters for the SVC library.

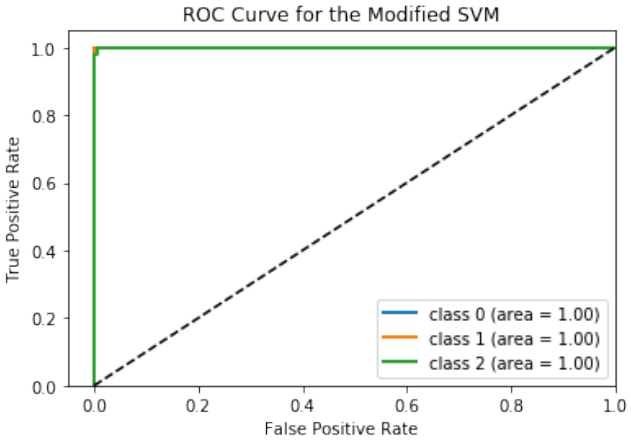
```
{'clf__C': 3, 'clf__decision_function_shape': 'ovr'
             ,'clf__gamma': 1, 'clf__kernel': 'rbf'}
```

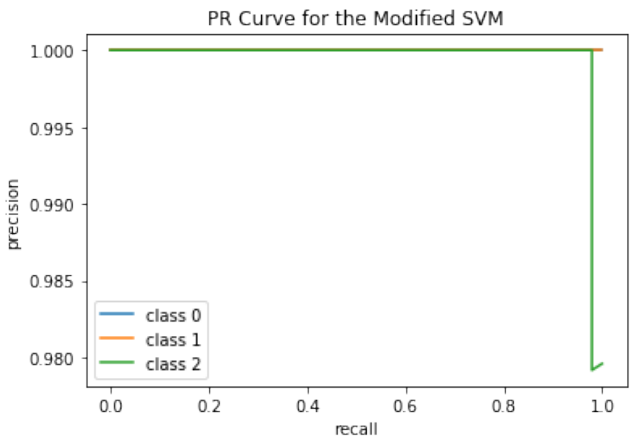
The table below summarizes the confusion matrix for GridSearchCV's resulting model.

	0	**1**	**2**
0	20	0	0
1	0	23	0
2	0	1	10

Table 3: Confusion matrix for the baseline model

The graphs below shows the ROC and PR curves for the GridSearchCV resulting SVC model.

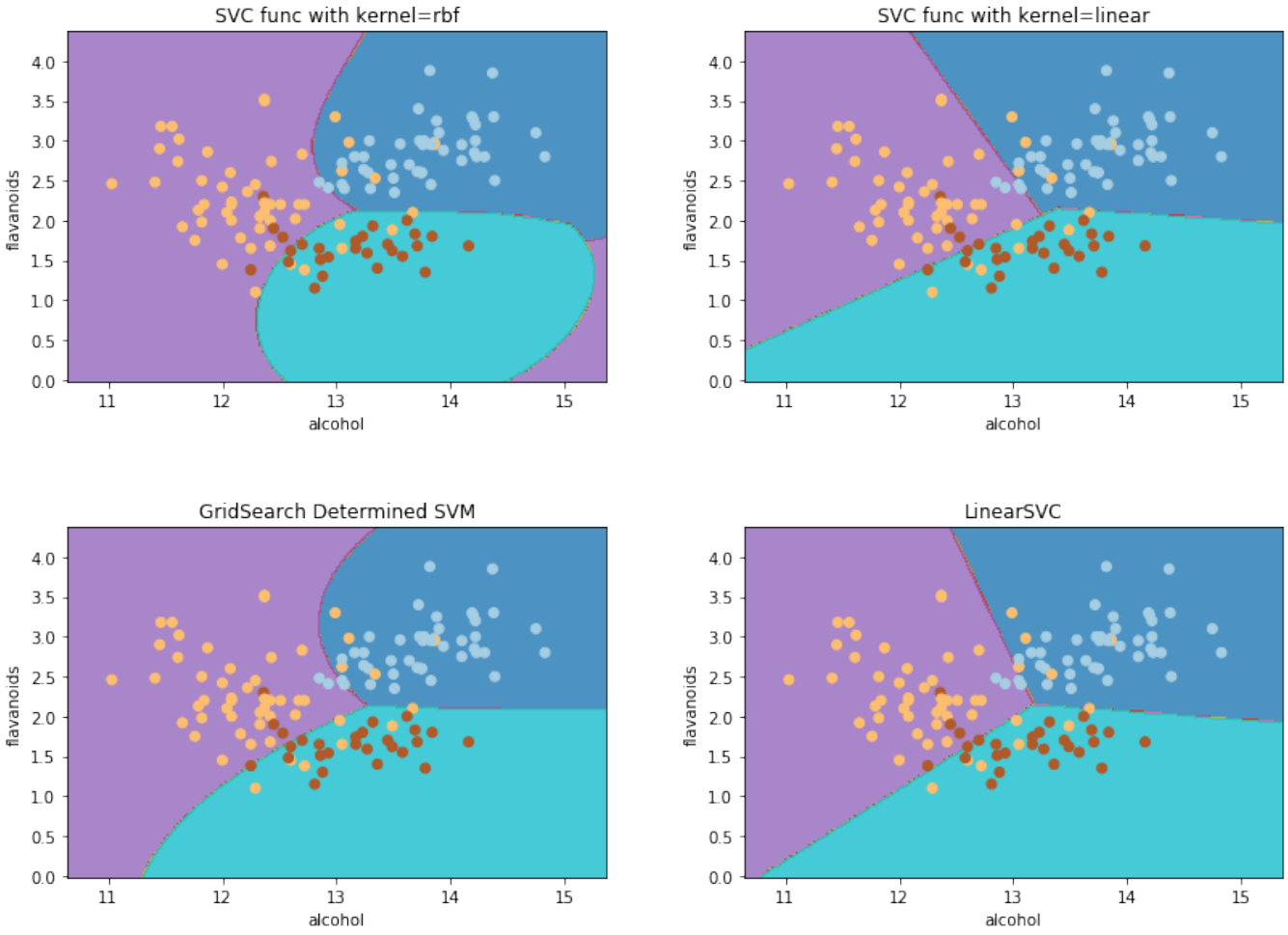




After 500 splits, the following was the accuracy score:

```
k=5 Cross Validation Mean: 0.9944
k=5 Cross Validation Var: 0.0001
```

3.4 Visualizing the SVM Contours



The figures above show the SVM contours had the models been trained for only two features: 'alcohol' and 'flavanoids'. The plots are shown mostly for visualization purposes, as such a plot would not be possible for all 13 features.

4 Discussions and Conclusion

As shown by the results, the GridSearch resulting SVM model has better performance metrics when compared to the baseline SVM model, but not much better than when manually tuned. The confusion matrix shows that the rbf kernel showed a bias for Type 1, often incorrectly classifying samples into this type.

The ROC curves paint a similar picture for performance. Theoretically, the best curve steeply goes through the top left of the graph, maximizing the area under the curve (AUC). We see that this is the case for both the modified SVMs. Although, in the manually modified SVM, Type 1 tends to have a slightly higher false positive rate. This implies that the bias for Type 1 seen in the confusion matrix of the baseline model is actually still visible in the linear kernel.

Looking at the PR curve for the manually modified SVM, we once again see that the model performs fantastically for Type 0 and Type 2. In the case of Type 1 classification, recall is slightly lower, implying that the false negatives are a larger value than the false positives. The PR of the baseline model shows the worst performance for classifying data as Type 2, continuing the trend from the ROC curve. Although glancing at the histogram in Appendix A does not provide an obvious explanation, it may be that the model finds Type 2 to be the most overlapped feature, making it difficult, and thus less likely, to classify a sample as that type.

Finally, we look at the SVM contours to visualize how the kernels impact the boundaries. We see that the baseline model actually favors the Type 1 class, as it has the largest area, explaining why there is a higher false

positive rate. Simply, the boundaries allow for more lenient classification for Type 1.

Overall, the GridSearch resulting model had the best metrics; although, the manually modified model was not far off. Interestingly, the LinearSVC library had a better accuracy score than the GridSearch resulting model.

5 Appendix A

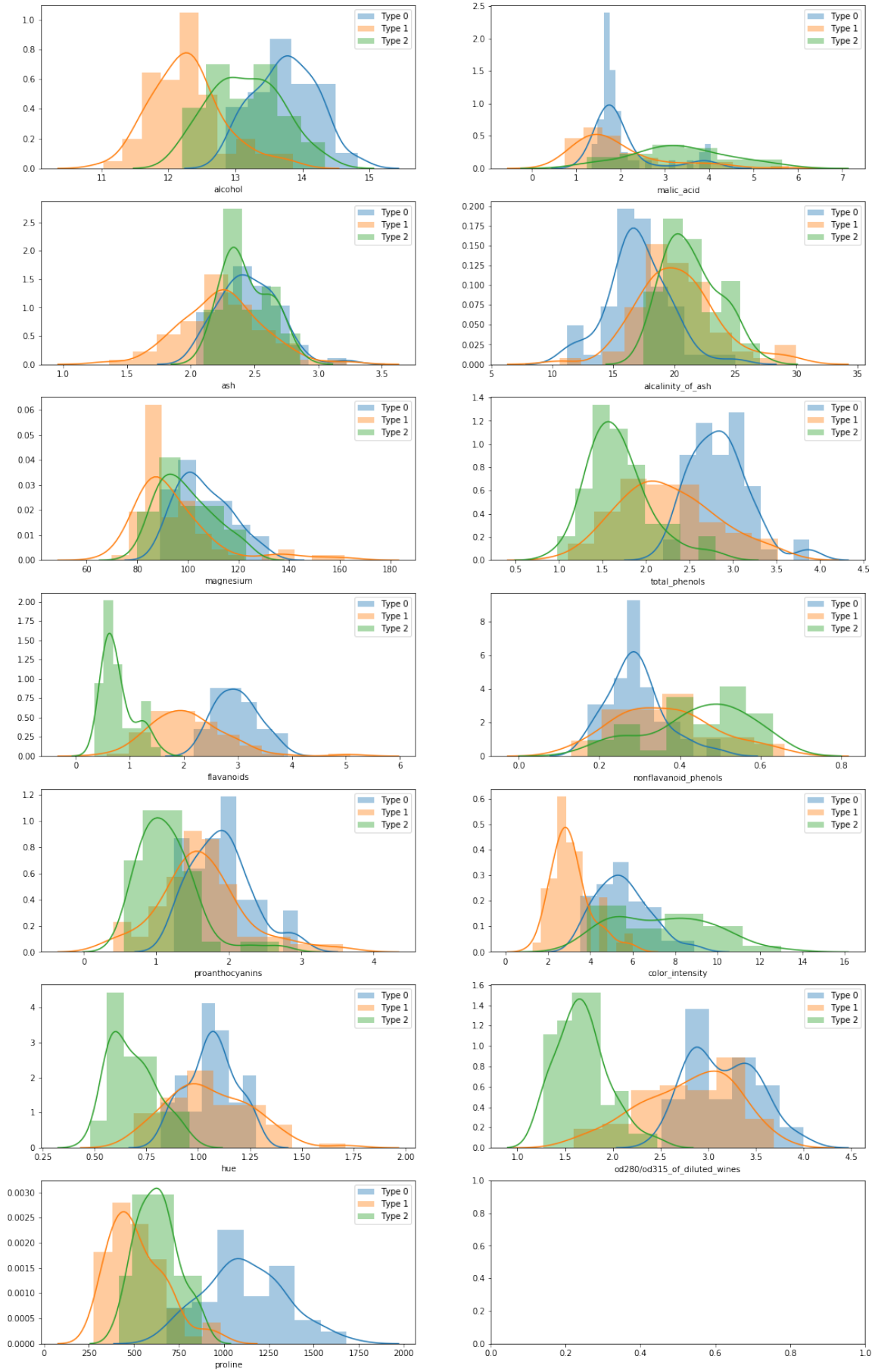


Figure 2: Histogram of wine features between wine types

References

[1] Hal Daumé III. "A course in machine learning". In: *Publisher, ciml. info* 5 (2012), p. 69.

[2] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. "A practical guide to support vector classification". In: (2003).

[3] *How to Use ROC Curves and Precision-Recall Curves for Classification in Python*. https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/.