Haixuan Guo and Colby Wight

Project 7 - Logistic Regression and Support Vector Machines

Introduction

For half of this study we will see how well we can determine whether or not a mushroom is poisonous by the way it looks and smells. This can be important to anyone who enjoys the outdoors or mushrooms. In order to determine this we will employ the use of some simple visualizations as well as two machine learning algorithms that can do the job for us. We learn that we can tell a lot about if a mushroom is poisonous or not. In the rest of the report, we will discuss how to predict the quality of wines. Our analysis indicates that 'alcohol','density','volatile acidity', 'chlorides' are most predictable for wine quality. RBF SVM(Radial Basis Function Support Vector Machine) has excellent performance on predicting new data, while Logistic Regression is more efficient compared with Support Vector Machines. Slides

MUSHROOM CLASSIFICATION

Dataset

This dataset contains examples of different mushrooms and whether or not they are edible. For each mushroom we have different traits of the mushroom such as: the color and shape of the mushroom cap, the surface of the mushroom cap, odor, bruises, how the gill is attached, the size of the gill, the color of the gill, the stalk shape, the stalk root, stalk color above and below the ring, the habitat it was found in, and the mushroom population (e.g. abundant, clustered, numerous, isolated). This dataset came in a nice clean format that required little cleaning. All of the predictor variables were categorical so it required creating dummy variables before doing machine learning. The Mushroom dataset can be found at link

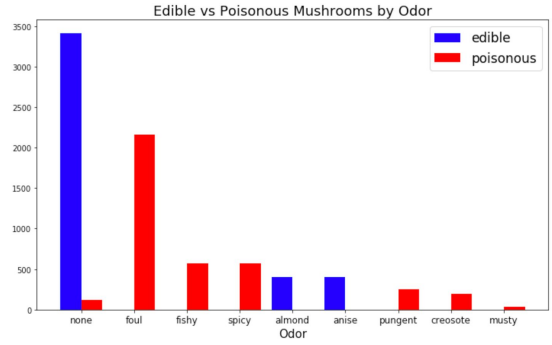
Analysis Technique

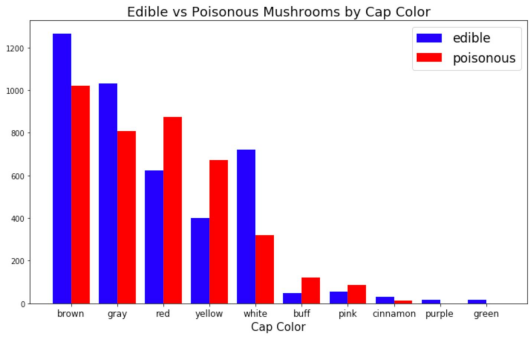
Before performing machine learning on the dataset we will start by doing some simple visualization. Some simple grouped bar graphs will be used to help see the difference in the amount of edible and poisonous mushrooms depending on different attributes. We will then choose to use a subset of the attributes, cap color and odor, that we believe are easy for non experts to use in determining whether or not they should risk eating a mushroom. We will build logistic regression and support vector machine models to see how well this can be done for the two attributes. We choose logistic regression because this may give us some insight into what features are most important in determining edibility and we choose svm's in case there are more complex relationships that may not be caught by a linear approach. We will use three different kernels for the svm's and see which one performs the best. Finally we will use all of the attributes to create models to see if we can obtain better accuracy than using only the two simple attributes.

Results

We saw from the two group bar plots that by only using these two variables ('Cap color' and 'Odor') we may actually have a better shot than we thought originally. By far, the odor of the mushrooms looks to be more helpful. We see that all the mushrooms that smell either 'foul', 'fishy', 'spicy', 'pungent', 'creosote', or 'musty' are all poisonous. So without doing any machine learning we can create a fairly strong rule that if a mushroom smells bad do not eat it. On the other hand, mushrooms with a 'almond' or 'anise' scent were all edible in this dataset so we can say if it smells good it is probably safe to eat. The problem is with mushrooms that don't have a

scent. We see that there are a lot of these and that while most of them are edible, some could kill you still. Looking at color the seperating is a little bit less clear. We see that white mushrooms are likely but by no means assured to be edible and we see that yellow and red mushrooms are more likely to be poisonous. The only mushrooms from this sample that are safe to eat are purple and green ones, but there are not that many of those. We see that by color alone we do not want to make this potentially life or death decision. See the graphs below:





Looking at the results above we may have a good chance to classify mushrooms well with a simple machine learning algorithm using just these two variables. With our logistic regression model we got a test set f-score of 0.981 with a precision of 0.963. Not bad odds if you are in need of food, but not necessarily something you want to bet your life on. With svm's we only do a fraction of a percent better. Using a linear kernel we get a test set f-score of 0.983 and a precision of 0.967. This was the best result obtained. Other kernels were not as good again indicating the idea of low variance in the data that do not require as much complexity of the polynomial or rbf kernels. We want to add more variables to see if we can get better odds.

Finally for the model with all the variables we were able to classify all of the test data correctly using logistic regression and svm with a linear kernel while the svm with polynomial and rbf kernel fared slightly worse. We learned if you want to be certain before eating a mushroom you need to look at more than just its color and odor.

WINE CLASSIFICATION

Dataset

The dataset is from kaggle competition wine quality classification. Based on physicochemical tests, we get 11 continuous input variables which are 'fixed acidity', 'volatile acidity', 'citric acid','residual sugar', 'chlorides', 'free sulfur dioxide','total sulfur dioxide', 'density', 'pH', 'sulphates', 'alcohol'. Based on sensory data, we get a categorical output variable - quality ranked from 3 - 9. The sample size is 6487. However, the dataset is imbalanced since there are a few excellent and poor wines. The total number of wines labeled in Quality 5,6,7 accounts for over 90% of all wines. Therefore, the wine quality is reassigned into three quality levels - low(Quality 3,4,5), middle(Quality 6), high(Quality7,8,9). Their proportions are around 2:2:1.

Analysis Technique

Because our input variables are quantitative and our output variable is qualitative, we visualized our data by drawing violin plots which display the distribution density of all features in different quality levels. Additionally, we paired all the features in a pairplot in order to find the most predictable features. Before modeling, all explanable features were standardized with mean = 0,

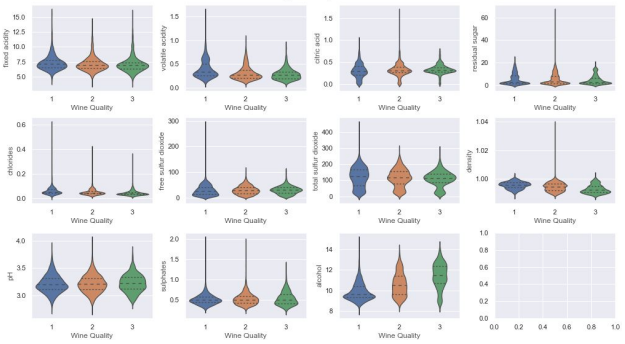
std =1. LogisticRegression and Support Vector Machines with linear kernel, polynomial kernel and RBF kernel respectively were employed to find the best model in predicting the wine quality after giving the result of physicochemical tests. What's more, since our output is multiple classes, we applied OneVsOneClassifier and OneVsRestClassifier on all models separately, and then compared the performance of the two classifiers on each model.

Results

It's noticeable from graph 2.1 that the distribution of alcohol is distinctive in three quality levels. The more alcohol, the better quality the wine has. In terms of pairwise attributes (see appendix A), chlorides matched with other attributes have the best performance in classifying wine quality.

The heatmap on the right describes the correlations of all variables. Features 'alcohol','density','volatile acidity', 'chlorides' are most highly correlated with wine quality. Based on those four attributes, the best model is RBF SVM, with accuracy = 0.5733.

Graph 2.1 Distribution of all attributes



Graph 2.2 Correlation of all attributes

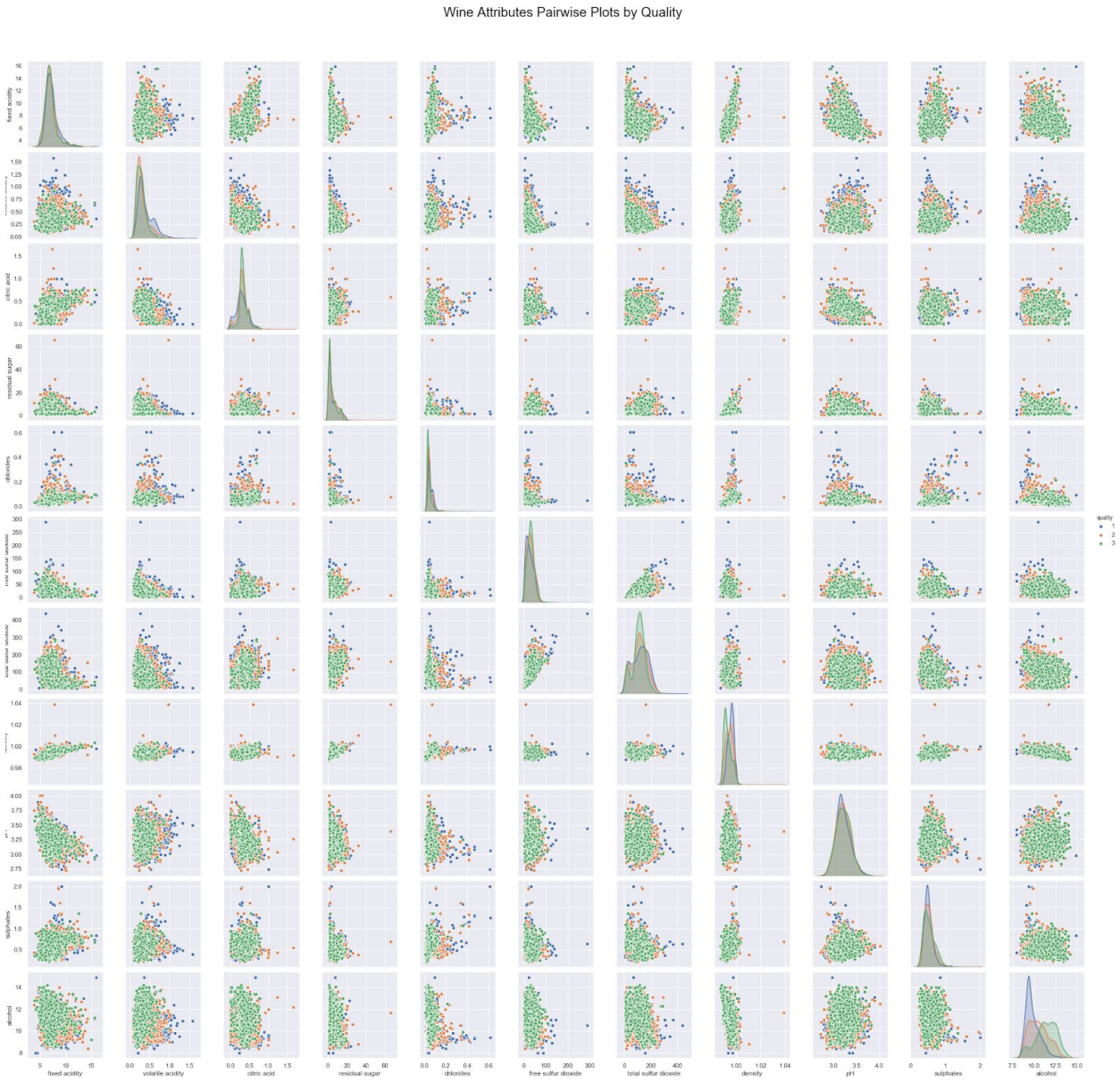
Although the performance of the four attributes above is impressive, there is still some information contained evenly in the rest of attributes. Therefore, we built models using all input attributes.

According to Appendix B, the best model is RBF SVM, with accuracy = 0. 6198.

The most efficient model is Logistic Regression , its running time = 0.0042, while the least efficient model is RBF SVM, with running time = 0.1731. Also note that Logistic Regression and linear SVM have better generalization ability as result of the accuracy score on training data is close to that on testing data, whereas the results of polySVM and RBF SVM showcase that the ability of both models to generalize to new data have decreased. Furthermore, we discuss the effects of two classification methods for multi class problems, one-vs-rest (OVR) and one-vs-one(OVO). It turns out that linearSVM and polySVM using OVO have better performance than those using OVR. Both methods have a slight influence on Logistic Regression and RBF SVM. Besides, it can be problematic that there is a smaller quantity of high quality wines compared to low quality and middle quality wines. By putting more emphasis on high quality wines, we balanced the class weight, and the overall accuracy scores drop slightly while the overall f-scores significantly increase.

From Appendix C, note that based on attributes "alcohol", "chlorides", the decision boundaries of the four models are linear. And it is hard to separate middle-quality wines from the rest wines.

Appendix A


Wine Attributes Pairwise Plots by Quality
fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol
quality
1
2
3

Appendix B:
model evaluation

	model	training data				testing data				running time
		accuracy	precision	recall	f1	accuracy	precision	recall	f1	
1	LR	0.5881	0.5931	0.5405	0.5523	0.5662	0.5715	0.5296	0.5407	0.004236
2	LR_ovo	0.5876	0.5918	0.5398	0.5511	0.5667	0.573	0.5302	0.5414	0.004716
3	LR_ovr	0.5865	0.5961	0.5323	0.5419	0.5672	0.5769	0.5194	0.5283	0.004002
4	linearSVM	0.5674	0.3903	0.4661	0.4212	0.5467	0.3796	0.4545	0.4087	0.083319
5	linearSVM_ovo	0.5674	0.3903	0.4661	0.4212	0.5467	0.3796	0.4545	0.4087	0.199082
6	linearSVM_ovr	0.5239	0.5414	0.4776	0.4529	0.5108	0.4754	0.456	0.4253	0.156633
7	linearSVM_balanced	0.5406	0.5371	0.5825	0.5327	0.5272	0.5198	0.5641	0.519	0.081158
8	polySVM	0.6176	0.6756	0.5529	0.5697	0.5677	0.614	0.5108	0.5216	0.09171
9	polySVM_ovo	0.6191	0.6783	0.5572	0.5754	0.5713	0.6155	0.5163	0.5282	0.245203
10	polySVM_ovr	0.6492	0.6626	0.613	0.6268	0.5908	0.5942	0.5558	0.5665	0.184949
11	polySVM_balanced	0.6336	0.6255	0.6331	0.629	0.5779	0.5715	0.5757	0.573	0.092145
12	RBFSVM	0.6651	0.6885	0.6179	0.6349	0.6108	0.6198	0.5662	0.5781	0.173058
13	RBFSVM_ovo	0.6695	0.6928	0.6255	0.6429	0.6092	0.6179	0.5667	0.5788	0.420209
14	RBFSVM_ovr	0.6659	0.6792	0.6352	0.6494	0.6046	0.6	0.5729	0.5814	0.334558
15	RBFSVM_balanced	0.6296	0.6239	0.6696	0.6281	0.5846	0.5773	0.6178	0.58	0.160097

Appendix C

Decision Boundary for four models

Red: low quality, green: middle quality, blue: high quality