Intermediate Report

Joe Despres

See the code in my github repo.

Problem Description

This project aims to discover the relationship between wine quality and chemical properties. The motivation for this project is that one purchasing wine seriously must consider different regions, grapes, growing climate, seasons, storage time, and substantial variations in quality and price. This is a vast and complex search space with the potential to find something of significant value. This is a problem well suited to Machine Learning. I will use Bayesian machine learning methods implemented with probabilistic programming package `pymc3` to model and predict the quality score a professional judge would rate the wine based on its chemical properties. The benefits of an accurate predictive wine model are that one could select excellent wine at a low price, lower the risk of purchasing low-quality wine, or even determine undervalued wine at auctions. I am not the first person to think of this, hence the published dataset.

Data

Here is one dataset on wine I will be using, I may add more. The University of California Irvine hosts a wine quality dataset where the outcome is the quality of a wine judge's score. This is an ordinal variable taking values between 1 to 10. However, the scores represented are between 4 and 8. This is an unbalanced assignment problem with a class distribution as follows.

4	5	6	7	8
20	735	947	392	78

These unbalanced class frequencies present some difficulties as the algorithms will have fewer examples of the 4's and 8's. Additionally, we do not have 9s or 10s. However, the nature of this problem makes it unlikely to have many highly rated wines. Exceptional wines are, by definition, rare. Therefore, algorithms capable of identifying highly-rated wines are valuable.

These are features based on physicochemical tests.

1 - fixed acidity
2 - volatile acidity
3 - citric acid
4 - residual sugar
5 - chlorides
6 - free sulfur dioxide
7 - total sulfur dioxide
8 - density
9 - pH
10 - sulphates
11 - alcohol

Progress

I resealed these with a min-max scaler because there is no way obvious way to force any of these features into a Gaussian distribution without damaging the underlying information. I engineered the following features type of wine and the number of judges agreeing on a score. Since these data come from two datasets of red

and white, it is obvious to make a new feature for **red or white**. After that, I found the **number of judges agreeing on that score** by collapsing duplicated columns into a single row. Then I recorded the number of times they were duplicated meaning multiple judges agreed on the wine. This saves computational resources and may be informative. Then I replaced the value of outliers defined as three standard deviations from the median with the value of three standard deviations from the median. This is a published curated dataset, therefore there is no reason to believe extreme values are outliers, but I don't want them skewing results. This concludes preliminary feature engineering. I will go back and make some minor changes after specifying my models. I have already fit some multinomial regression models primarily for benchmarking, they fit rather poorly with weakly informative priors. With ordinal multinomial regression using an 80% split the accuracy is roughly 58%. See the confusion matrix below.

	4	**5**	**6**	**7**	**8**
4	0	0	0	0	0
5	4	89	69	2	0
6	1	46	121	37	12
7	0	0	19	26	9
8	0	0	0	0	0

TODO

I want to experiment with an ensemble of Bayesian models and fit an ensemble of models. I want accuracy of 85%, using a 60% training set. After specifying a model, I will cross-validate my feature engineering. I will need to be mindful of coincidental associations. After that, I will experiment with some resampling and stratified sampling methods to even out the class assignments.