Project Midterm Report

ajd242, eer48, lfl42

1 The Problem

When valuing a bottle of wine we often consider the producer, its rating, the region which it is from and any reviews we have available to us. Unfortunately, we don't have a formula for determining if this wine is a good buy at its price point, but what if we could? Our goal is to determine if we can predict the selling price of a bottle of wine given its ratings, reviews, region, etc. This model can help producers understand the value of the wine they are making, and consumers to make smarter purchases. Ultimately, we hope our model for predicting the price of a bottle of wine will create a more efficient market for buying and selling wine.

2 Dataset

Issues Importing the Data

Our first approach was to import data using Python's pandas.read_csv. The first issue we ran into after importing occurred during inspection of our raw data in Excel. We noticed that many of the foreign characters such as 'é' were not being displayed properly in our imported dataset (see Figure 1.1). This improper formatting concerned us because the interpretability of the dataset was difficult for potential readers and us; however, after some research on UnicodeDecodeError's (such as the one given below), we found the problem to be that Excel expected characters to be encoded as UTF-8*. These incorrect outputs were a result of UTF-8 only including ASCII characters; however, our data had a significant amount of non-ASCII characters, resulting in the following error:

UnicodeDecodeError: 'utf8' codec can't decode byte 0xa5 in position 0: invalid start byte

We initially tried changing the file type and even started writing code to replace strange character sequences. This process was labor intensive and took way too long. Then we thought it would make sense to simply remove the data containing any foreign characters and in doing so, learned our dataset was biased towards non-foreign wines and reviews.

1. Tried changing the excel file's character encoding.
2. Tried changing the .csv file's character encoding.
3. Tried using "Apple Numbers" (an excel equivalent) to see if we could encode foreign characters.
4. Tried using Business Intelligence software (Microsoft Power BI) to identify and replace character patterns with an ASCII equivalent (i.e., changing 'Ã©' which was supposed to be an 'é' to 'e'). This was time intensive and was not a practical approach.
5. Tried reading the dataset into a .ipynb using Python's pandas.read_csv; However, one main issue we had was the presence of commas in reviews. This caused our data to transform the fields of those particular examples from the expected 11 fields to over 140. We recognize this transformation as an error in read_csv and our data which added onto our foreign language issues.

Our root problem was the character encoding of a .csv file in which our data was stored. We ended up using .json files instead of .csv to store our data. This type of file does not allow us to examine our raw data in Excel, but when importing the file into the .ipynb notebook we found that the method pandas.read_json read-in foreign languages efficiently and accurately without causing the fields to expand or transform. This was mainly a benefit of how .json files structure and store data in defined fields while also including the correct encodings for reading numbers as Float64 rather than strings.

Cleaning the Data

Trimming 'country' and 'variety' Features

We noticed that we had some "odd" countries that we didn't expect to see. Also they had a very limited amount of data points and thus would not be helpful in predicting prices for more "common" countries.
(150930 rows x 10 columns → 143622 rows x 10 columns) 129964 5946 last 13658
We also decided to remove grape varieties with less than 74 data points. Our reasoning for doing this is to capture the varietals that the average customer will come across when shopping for wine at the average to above average wine store. We capture 100 varietals from our data with some of our top ones displayed in Figure 4 below.
143622 rows x 10 columns → 129964 rows x 10 columns

Removing NaNs from our 'Points' and 'Price' Columns

Examining our data some more we realize we have some values in price (our output vector) which are NaN. See index 150922 for an example. We proceed to remove these values. We selected only the points and price columns as a means for identifying NaNs to remove. Our rationale for only using these columns is some wines have more detailed regions and sub-regions vs others, this is not a data entry error but simply due to industry structure and classification systems.

Our data trimming results in: (149568 rows x 10 columns → 135910 rows x 10 columns). Overall, we end up with about 90% of our original dataset which accounts for 90% of total wine production in the world.

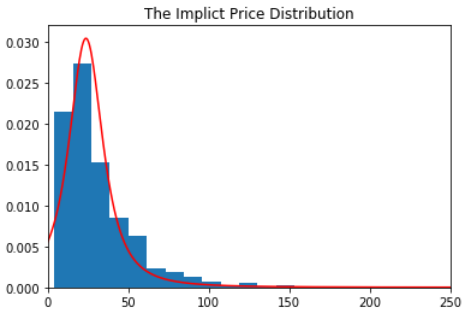
Finally, we transformed the 'country' feature via one-hot encodings for the purpose of using the country classification as a feature in our future regression analysis.

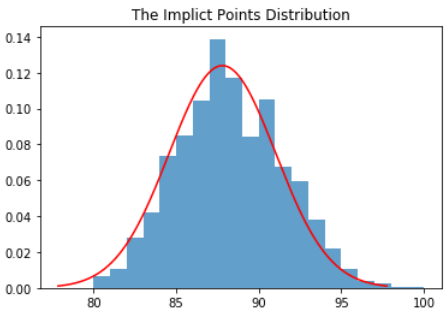
3 Preliminary Analysis

Exploring the Data

Implicit 'Price' and 'Points' Distributions

After visualizing the raw 'Price' data, we hypothesized both the t and exponential distributions would fit the data well. We proceeded to fit the distributions to the data and plot the results (will display as figures). From here we could dive deeper into goodness of fit through a chi-square or K-S goodness of fit test. Similarly, we fit both a t and normal distribution to the 'Points' data, plotted the results (will display as figures) and could conduct a GoF test.
The main purpose of creating and storing these estimations of the price and points distributions is to test the distributions of our predictions and see if they result in similar distributions. This gives us another metric to determine if our results are in line with the market's pricing and points distribution.



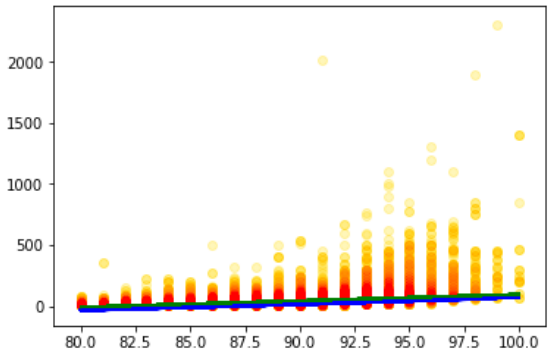


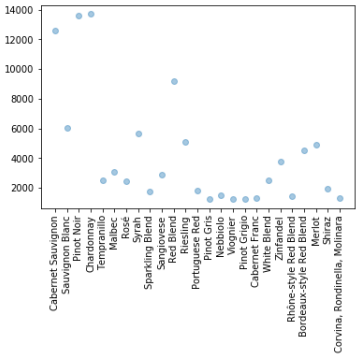
The rest of our data exploration so far consists of comparisons such as: 'Price' v. 'Points', Avg. 'Price' per 'Country', and Avg. 'Points' per 'Country' complete with figures to be posted when we have more space.

Preliminary Regression Analysis with Points and Countries

We ran a least squares regression using the points and country data to predict the price of the wine. Our first obstacle in this process was transforming the countries into 14 separate binary vectors using one-hot encoding. This

allowed the algorithm to learn the proper weight for each country (the blue line). Our results, as shown below, indicate that the algorithm implies that there is a significant difference in price simply due to what country it is from and also tends to shift the price lower than a least squares linear regression with training data being only points (the green line).





4 Testing

We plan to split our data into training, validation, and test sets. For this dataset, we decide against bootstrapping or cross validation, as it contains almost 150000 samples so we believe there to be enough data that these methods are not necessary. We noticed that the order of the data samples seemed to group wines with similar point rankings close together. For this reason we will randomly select samples to place into training, validation, and test sets (as opposed to simply splitting three ways by index).

The proportions of each set will be approximately 70%, 15%, 15%, respectively. We find it necessary to distinguish between validation and test sets so that we may use the former to fine-tune our model while using the latter in the final evaluation of the project. In order to examine and correct for the under- or over-fitting of our model at various stages, we plan to compare the training and validation set errors as we increase the input data amount; if the training error is much lower than the testing error but worsens as we increase training instances, we have identified over-fitting, while if both test and training error are high and do not improve as we increase training instances, we have identified under-fitting. To remedy these situations, we plan to apply the concepts of regularization, using our validation set to determine the weight of the regularizer.

5 Remaining Work

We plan to fit more complex linear models to the data as we transform more features into ones that can be analyzed using least squares linear regression. We hope to find that some features such as 'country', 'region', feature transformations (resulting from an NLP analysis of the 'description's), etc. have significant weight in predicting 'price'. From inspection, we also believe that the data may be non-linear and we will explore polynomial regressions as well. We also will explore using non-negative least squares regression to return only positive coefficients to get a better predictive information from the 'country' feature.

We will use decision trees to help predict 'price' by classifying our data. An example of our implicit reasoning for splitting our data into 'region's and 'varietal's is that the price of a Pinot Noir in California has little impact on the price of a Pinot Gris from Spain. We are thinking of using recursive binary splitting to fit the data, then use cost-complexity pruning to decide where to cut. We will be considering the bias-variance tradeoff as we perform this.

We will use natural language processing to examine the "description" column of each sample, which contains a couple of sentences from a sommelier describing the qualities of the wine. We plan to use sentiment analysis to transform the string to create a new feature for our original model. We also plan to examine the relationship between the "points" feature, a numerical ranking by a sommelier, with the amount of positive versus negative words in the description. Finally, we plan to use word embeddings to map each description to a high dimensional vector space. This will enable us to use a neural network to directly learn and predict the price of a wine given its description.