Predicting The Price of Wine

Domain Background

Anyone who has ever purchased wine from a wine shop will know the paralysis of standing in front of floor to ceiling shelves trying to decide which bottle to buy. Wine is hard to choose. The reason it's hard to choose is because there's so many different varieties offered by so many different brands in a huge range of prices. You can know that you want a Malbec from Argentina but that may still leave you with a plurality of options that range from $5-$100. And what makes one wine $5 and another $100?

The complexity of wine valuation is best illustrated by comparing it to its closest contemporary, beer. In the beer realm, price, type and brand are often the only criteria you have to make a choice. Further simplifying the matter, within these criteria, the price range is typically pretty narrow, each type (i.e. IPA or Lager) tastes very similar, and brand recognition is high. Therefore, as a consumer, when looking for a certain type of beer at a given price, I only need to exercise my opinion within a few narrow variables to be reasonably confident I'm getting a good value. Wine on the other hand has significantly more attributes. Not only are the varieties much more numerous and the brands far less ubiquitous, but a host of other factors such as vintage (year) and terroir (place) can have a big impact on flavor and quality. This means that when buying wine, rather than choosing from a succinct list, whittled down using a couple variables, buyers have to perform a multivariate optimization problem; the result of which, because prices vary so widely, doesn't guarantee receipt of a good value. Because of this wine valuation presents an interesting problem to try and solve using machine learning.

Problem Statement

For my capstone project I propose training a machine learning model that can use the attributes listed above as features to predict the price of a given bottle of wine.

Datasets & Inputs

For this project I will use the "Wine Reviews" data set from Kaggle [1]. It is made available under the CC BY-NC-SA 4.0 license which ensures it is free to use for non-commercial purposes. This data set is comprised of both CSV and JSON files of 130,000 wine reviews scraped from WineEnthusiast.com on November 22nd, 2017. Each file contains the following columns:

- **Points**: The number of points WineEnthusiast rated the wine on a scale of 1-100 (though they say they only post reviews for wines that score >=80)
- **Title**: The title of the wine review, which often contains the vintage if you're interested in extracting that feature
- **Description**: Tasting notes provided by the taster
- **Taster name**: WineEnthusiast taster that provided the descripion

- **Taster twitter handle**: Self explanitory
- **Price**: The cost for a bottle of the wine
- **Designation**: The vineyard within the winery where the grapes that made the wine are from
- **Variety**: The type of grapes used to make the wine (ie Pinot Noir)
- **Region_1**: The wine growing area in a province or state (i.e. Napa)
- **Region_2**: Sometimes there are more specific regions specified within a wine growing area (i.e. Rutherford inside the Napa Valley), but this value can sometimes be blank
- **Province**: The province or state that the wine is from
- **Country**: The country that the wine is from
- **Winery**: Wine-maker that produced the wine

The columns within this dataset (except for price), after cleaning and preprocessing, will serve as the input features for the model. Price, since this will ultimately be my prediction, will constitute the label used to train the model.

Solution Statement

In order to predict wine prices I propose building a regression model that will use the various attributes of a wine to predict its price.

Benchmark Model

In the Medium article, "Vintage, AVA and Quality: A Study of Napa Valley Wines" the author tests several types of regression models - Linear, SVM, Decision Trees, Ensemble - to predict the price of wine, ultimately landing on a Random Forest Regressor with an R^2 value of 0.51 [2]. I'll use both this result and the author's method as a benchmark, fitting several types of regression models and comparing my R^2 values against her standard.

Evaluation Metrics

To evaluate my model I will use R^2, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). According to the Kaggle article, "Model Fit Metrics" [3], MAE and RMSE work well in combination and the addition of R^2 will allow me to compare my model with the benchmark described above.

RMSE is "the square root of mean squared error" and according to the article, it's "popular in the literature" but is "less resistant to outliers [than MAE], and thus reports a poorer-fitting model when outliers are not properly accounted for." Therefore, it "is considered … good … when doing certain things, like performing hyperparameter searches." MAE, as just mentioned, is "more resistant to outliers than MSE [and RMSE]" because it "computes the expected absolute error." Also, "MAE is a more useful reporting statistic because MAE is interpretable, while RMSE is not." By using these models in combination I will have MAE

available as an interpretable means of evaluate model performance and RMSE available when I need to tune my model during training.

Project Design

There are numerous models for performing regression analyses. In my research for this project I've found that most predictive regression studies deploy several models and compare the results of each to determine which is best for solving the problem at hand. For my capstone project I will employ the same approach.

I foresee the project having the following steps:

1. Data import: reading in and saving the data
2. Data analysis: cleaning, preprocessing and visualizing
3. Regression analyses: Training the following regression models
    a. Linear regression
    b. Polynomial regression
    c. Ridge regression
    d. Lasso regression
    e. Support Vector regression
    f. Decision tree regression
    g. Random forest regression
    h. XGBoost regression
4. Evaluation of performance: comparing error between the regression analyses
5. Model tuning: taking the regression model with the lowest error and tuning hyperparameters to try and further improve performance
6. Presentation of results

Sources

1. "Wine Reviews," Kaggle.com, Zach Thoutt, 2017: https://www.kaggle.com/zynicide/wine-reviews
2. "Vintage, AVA and Quality: A Study of Napa Valley Wines," Medium.com, Rachel Woods, 2019: https://medium.com/the-wine-nerd/vintage-ava-and-quality-a-study-of-napa-valley-wines-565e6f164c08
3. "Model Fit Metrics," Kaggle.com, Aleksey Bilogur, 2017: https://www.kaggle.com/residentmario/model-fit-metrics