Conclusion and Limitations

YZ Analytics

This project set out to make wine easier to understand for the average person. Through this project, we examined close to 130,000 wine reviews to create a complex web application via Shiny that allows users to explore wines and wineries through an interactive map and searchable catalog as well as to enter new wine observations to receive a prediction of how many point values their new wine would receive. Through the map and catalog explorer and prediction engine, anyone can learn more about wine and explore the different varieties around the world.

This project also involved text analysis beyond the scope of our Stat-495 class, researching into word ranking methods such as term frequency-inverse document frequency and learning how to incorporate document-term matrices into our prediction model. Our gradient boosting model showed that the most important features to predicting wine points are price, variety, and province. The most important word for prediction is "rich."

Limitations and Future Directions

There were several limitations to the work in this project. They will be discussed below, and various suggestions for future directions to address these limitations will be described.

Data Scraping

The Kaggle dataset we used originated from Wine Magazine's website and was scraped in 2017. However, in the most recent years, almost 100,000 more wines have been added to the website. A future direction this project could go into would be to use the python scraper provided by the Kaggle user who uploaded the original dataset (Thoutt, 2017) and scrape the newer wine reviews so there would be more up-to-date data.

Model Computation Run Time

A key limitation to the prediction aspect of this project was the sheer amount of computation involved. The model created in this project used only 200 of the words with the highest mean TF-IDF values, but models with many more words could have been created and would likely yield lower MSE values and better predictions. However, given how the gradient boosting algorithm is computationally intensive because it requires creating many small trees, we chose to limit the number of variables used in the dataset to run the model.

Future steps that could be made in order to speed up computation time and create a more accurate model could be to look into faster gradient boosting algorithm methods. In R, there are newer packages such as `XGBoost` and `LightGBM` that were developed specifically for decreasing run time of the computationally intensive gradient boosting algorithms, so the functions in these packages could be explored on the data in this project.