Capstone Project

Red Wine Rating System Final Report

1. **Define the Problem Statement:**

The goal of this project is to use machine learning to be able to accurately predict the quality rating a red wine would receive from a human judge based on its physical and chemical characteristics. Initially, I had also hoped to translate that into estimating how a wine would place in a wine competition, but quickly realized that was going to be too big of a problem to tackle for this assignment. So, for this project I will just focus on building a model that will accurately predict the quality rating, which will serve as a guide to a wine drinker to know if the wine will be good to drink. It would also be useful to the winemaker, who is monitoring these key characteristics throughout the winemaking process.

In other words, based on physical/chemical characteristics, I want my model to answer the questions: How would a human judge rate the quality of this wine? Based on that rating, will this wine taste good?

2. **Model Outcomes or Predictions:**

As stated in my capstone EDA report, the problem could be approached two ways from a machine learning perspective. One could take a Linear Regression approach and train models to produce a numerical score and map that score to the closest human judge rating (0-10). Alternatively, another approach could be to view the human rating (0-10) as each being separate class and train classification models to produce the predicted rating.

After discussion with my advisor (Savio), it was clear that the classification model would be preferred as it is more definitive in terms of which rating class (0-10) the wine falls into. Also, the metric of accuracy in the classifier model is easier to interpret than MSE in the linear regression model.

3. **Data Acquisition:**

For this step I needed to find a dataset that I could work use within the scope of what I could accomplish based on the techniques learned at that point in the class. Many of the wine judging competitions do not publish a lot of detailed data on the wines selected and what is published is often based on vague descriptors such as taste profiles like “cherry” or “baking spice” or “spicy” or “structured”, and so that sound like fancy wine terms and are purely subjective.

To try and build a model that could vectorize all of those terms was clearly beyond the scope of this effort. (That will be something for future exploration.)

I found a dataset and discussed it with Savio, who agreed it was a good fit in the UCI machine learning repository (https://archive.ics.uci.edu/dataset/186/wine+quality)

The UCI dataset contains red Vinho Verde wines from Northern Portugal. Each of the wines is defined by 11 physical/chemical characteristics and then given a score between 0 and 10 by wine judges. 0 is terrible and 10 is outstanding wine. In the actual dataset all of the wines judged fall in the range of 3 to 8.

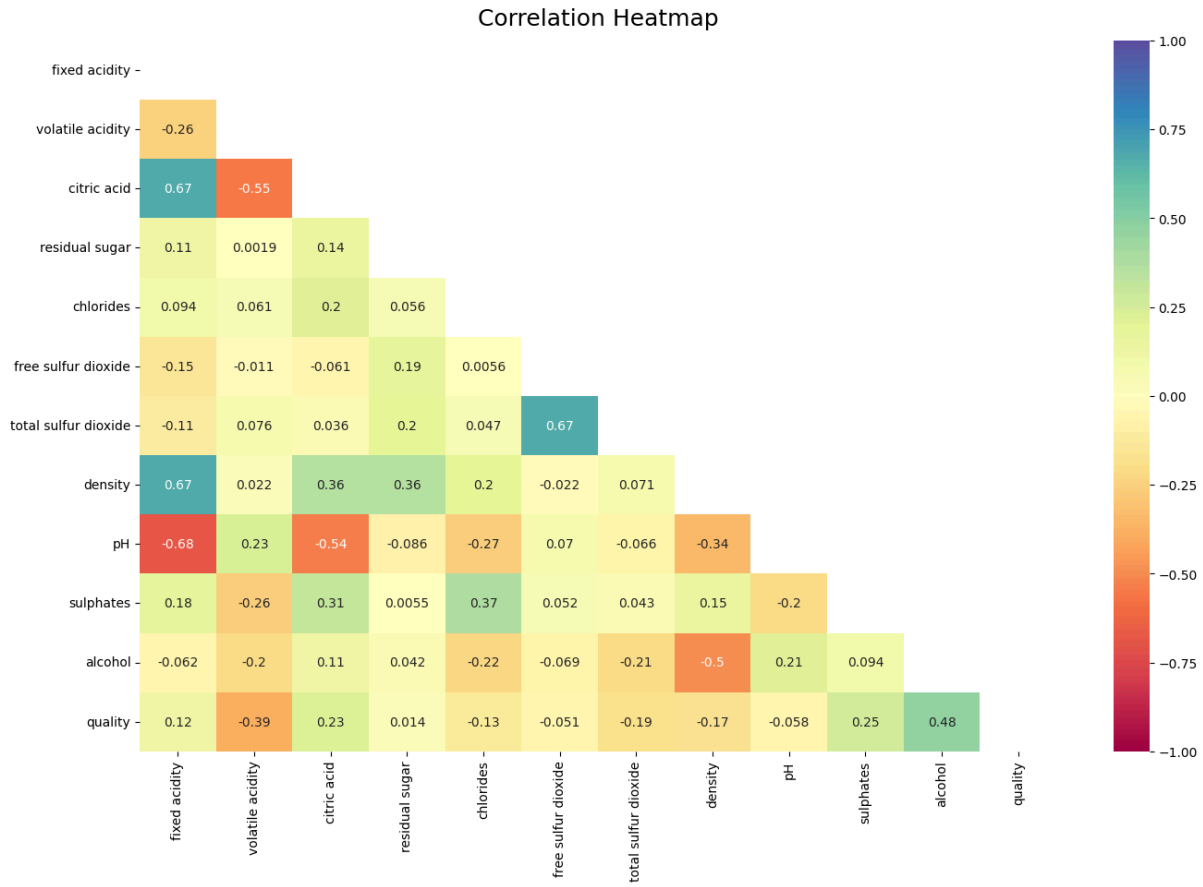
See excerpt below:

```
red_vinho.head()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

In my research, there were very few datasets that could even come close to this in terms of the number of quantitative features of the wine and the resulting target value of a human rating (score). The majority of datasets only quantify residual sugars and alcohol content. Those simple features, while very influential, do not provide enough information to separate high quality from low quality wines.

As you can see from the correlation heat map there are a number of features that have a strong impact (positive and negative correlation) on the resulting quality rating.



The heatmap shows strong positive correlation to alcohol and residual sugars, and also other factors such as sulphates, citric acid, and fixed acidity. Note, also there is noteworthy negative correlation to volatile acidity, total sulfur dioxide, density, and chlorides. So, it is pretty clear a more extensive set of variables (features) would produce a better result.

4. **Data Preprocessing/Preparation:**

Preparing the data was more straightforward than some of the datasets I have worked with throughout the course. As noted in the accompanying Jupyter notebook, I ran through the steps to ensure there was no missing data or null values. I also checked for duplicate data and initially it appeared that there were some (240 samples of 1599), however after researching it and finding that each of those samples represents an independent observation (wine sample being judged) it would be best to keep them, otherwise information would be lost.

The data was all numerical so there was no need initially for techniques like "one-hot encoding" or "ordinal encoding". I also applied standard scaling where appropriate and did not with certain classifiers like decision trees, as noted in the Jupyter notebook. Additionally, thinking ahead I did want to

better understand the linearity of the relationship between key variables and the target ('quality'). I used a number of scatter plots as shown in the Jupyter notebook to conclude the relationship was pretty linear, and that polynomial transformation would not be beneficial to the model performance. (I had actually confirmed that during the EDA report where I played with polynomial degree 2 and did not get a better result.)

Later, after running initial models, it became clear that I would need to do some further feature engineering to further improve the results, as clearly documented in the accompanying Capstone Jupyter notebook. After some thought and exploring how some of the wine experts (such as those published in Wine Spectator Magazine) categorize wines, I decided to re-map the ratings in the dataset to something that would be useful, going back to the original goal, to a wine drinker of discriminating taste in selecting a good wine to drink. As documented in the Jupyter notebook, I mapped the ratings of 3 to 8 in the dataset as follows:

7-8: "very good"
5-6: "good"
3-4: "not good"

I explained in more detail why I chose those ratings and it was directly from a well respected source that many wine drinkers rely upon for selecting a good wine. This re-mapping of the data so that instead of there being 6 classes, there were now just three resulted in an improvement of the accuracy from mid-60% to mid-80% range— pretty substantial improvement.

5. **Modeling:**

First, as noted in the accompanying Jupyter notebook, I ran a number of classification models we learned in class before re-mapping the target variable ('quality'). In each case, I ran the base model and then used GridSearchCV to tune the main hyperparameters for the model.

Second, after doing feature engineering on the target variable ('quality') reducing it from 6 classes to 3 classes, I re-ran several of the most effective models from earlier to note the performance improvement.

6. **Model Evaluation:**

Results of the first round of modeling achieved the following results (best accuracy score for the type of model):

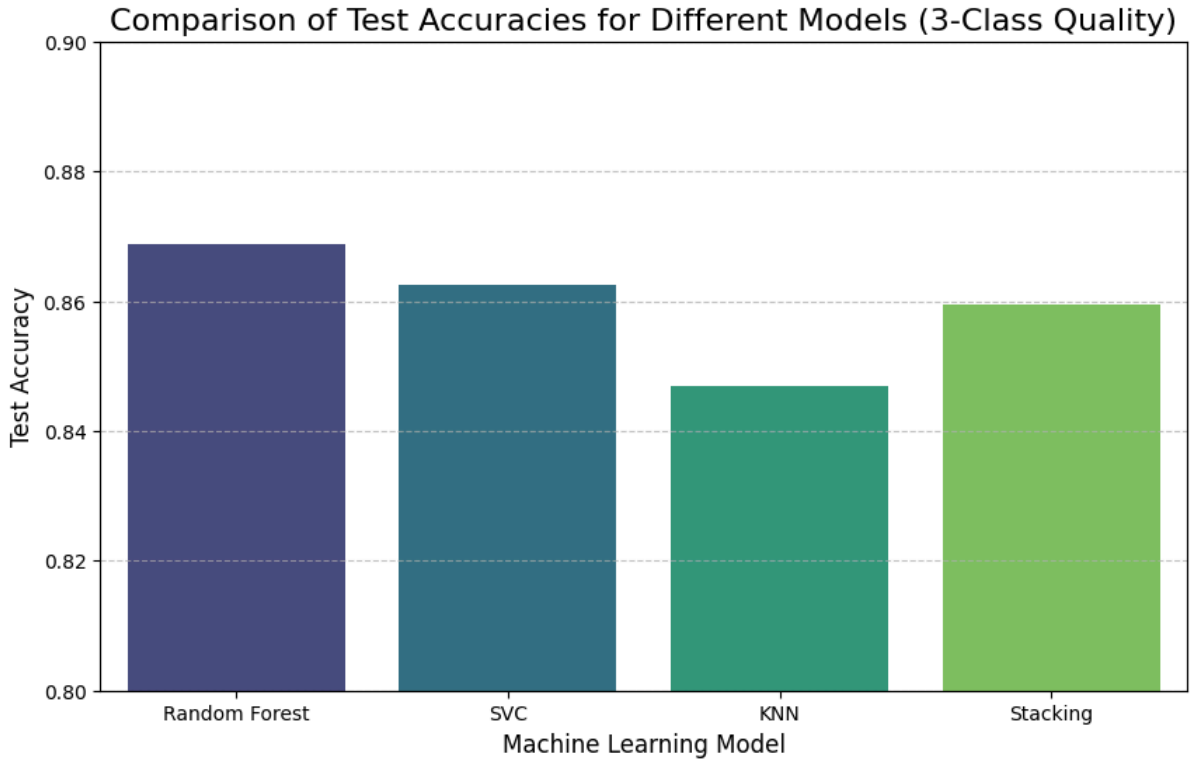
Round 1: Best score for each type of model (rounded to one decimal place)

Model	Cross-validation Score	Test Score
Logistic Regression	61.6%	56.9%
KNeighbors (KNN)	61.3%	62.2%
SVC	63.6%	60.9%
Decision Tree	59.7%	55.3%
Random Forest (ensemble)	69.2%	66.3%
Bagging on SVC (ensemble)	64.1%	60.3%
Stacking (ensemble, multiple classifiers)	66.7%	67.8%

Below are the results of round 2 of modeling, after reducing the number of 'quality' classes from 6 to 3.

Round 2: Best score for each type of model (rounded to one decimal place)

Model	Cross-validation Score	Test Score
Random Forest	86.6%	86.9%
SVC	85.1%	86.3%
KNeighbors	84.1%	84.7%
Stacking	85.9%	85.9%



One of the key advantages in choosing the classification modeling methods is the ease and clarity of interpretation of results. So, without too much effort it was easy to see that the Random Forest (#1) and SVC (#2) would be the models of choice for solving a problem like this. One of the other advantages of Random Forest (and Decision Trees) in general is being able to easily see what features are most important and following the path of how the dataset is classified. This aspect could be really helpful to the wine maker as they monitor the key properties of the wine throughout the winemaking process.

Conclusions:

There is a lot of work to be done to really make this machine learning project useful commercially, but it gave me a chance to use some of the methods learned throughout the program, and I do believe that with additional work and using even more powerful models such as neural networks this could be really useful. Not that long ago I had a chance to talk to a winemaker, Chris Barrett, at Pezzi King Winery in Healdsburg, California. He was really interested in this project and said he could see how they could take the data that comes from the wine testing labs and feed it into something like this and get some additional insight into how to adjust the profile of the wine to produce a higher selling product. I am excited to see how this will progress in coming years as I am sure we will see much more deployment of ML and AI in the agriculture industry in the near future.