Wine Quality Prediction Model
Architectural Decisions Document

Ronald Pereira

November 2020

1 Architectural Components Overview

1.1 Data Source

1.1.1 Technology Choice

Simple download from data source website.

1.1.2 Justification

Since we are using a public dataset available as a CSV (Comma Separated Values) file, we won't have to use any other methods of getting this data, so a simple download from the direct link itself (`https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009`) will suffice.

1.2 Data Repository

1.2.1 Technology Choice

A separated *data/* folder.

1.2.2 Justification

As the chosen data source is a simple CSV file, we can store it directly in our project as our unified train, validation and test datasets. For this purpose, a separated *data/* folder containing all data used in this project will be created for simplicity.

1.3 Discovery and Exploration

1.3.1 Technology Choice

Python 3.8, Jupyter notebook and some other data operation libraries (e.g. Pandas).

1.3.2 Justification

Python will be our chosen programming language chosen for this purpose, more specifically the newer 3.8 version release. This language has a lot of third-party libraries that supports our data science projects, going from ETL tasks to model building and deployment. Also, we will be using a interactive notebook to generate and share our source code and results with our internal team, as they can easily be integrated into the same file.

1.4 Applications / Data Products

1.4.1 Possible Applications / Data Products

We can use this project as a mean to assist some red wine vendors to decide which wines to buy or not, as well as giving a hint of their quality for price assignment.

We can also use this project as a mean to assist red wine producers as well, by explaining our data by model explanation strategies. In that way, we can use the model feature importance to guide the efforts of this producer, avoiding him to spend more money on less impactful wine characteristics. Uniting this with a SHAP[1] explanation, we can even tell the producer the relative impact of any characteristic on the quality output of the red wine.

2 Model Definition

As this task is for regression purposes because the prediction class in a continuous number from 0 to 10, with 10 being the highest red wine quality possible, we will use regression models to achieve better results.

2.1 Models

- Linear Regression
- Stochastic Gradient Descent Regression
- XGBoost Regressor
- Multi-layer Perceptron Regressor
- Deep Neural Network Regressor
- Ensemble of all past models by a Voting Regressor

[1] `https://shap.readthedocs.io/en/latest/`

2.1.1 Linear Regression

As the simplest regression model possible, this will be our baseline model, which we will aim to beat with each one of our other models regarding our regression metrics. For this purpose, we'll use Scikit-Learn Python API[2] to create, train and evaluate this model.

2.1.2 Stochastic Gradient Descent Regression

As our second chosen regression model, this is also a simple tree-based regression model. For this purpose, we'll also use Scikit-Learn Python API to create, train and evaluate this model.

2.1.3 XGBoost Regressor

As our more advanced tree-based regression model, we'll use a XGBoost[3] Regressor (eXtreme Gradient Boosting). For this purpose, we'll use the XGBoost Python API to create, train and evaluate this model.

2.1.4 Multi-layer Perceptron Regressor

As our first and most simple neural network model, we'll use a simple MLP Regressor via Scikit-learn Python API to create, train and evaluate this model.

2.1.5 Deep Neural Network Regressor

As our most complex neural network model, we'll use Pytorch[4] Deep Learning Python API to create, train and evaluate this model.

2.1.6 Voting Regressor

As we would also like to see, we'll build a Voting Regressor which will consider all our past models (except the Deep Learning model) to achieve better results at the cost of explainability. For this purpose, the Scikit-learn Python APi will also be used to create, train and evaluate this model ensemble.

2.2 Regression Metrics

We'll use various regression metrics to evaluate our models performance:

- R^2 score
- Explained Variance
- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)

[2] `https://scikit-learn.org/stable/`
[3] `https://xgboost.readthedocs.io/en/latest/index.html`
[4] `https://pytorch.org/`

3 Project Decisions

3.1 Model Selection

The metrics for each model evaluated is in Tables 3.1 and 3.1.

Model Name	R^2 Score	Explained Variance
Voting Regressor	0.484607	0.484758
XGBoost Regressor	0.472041	0.473053
Linear Regression	0.383096	0.383105
Stochastic Gradient Descent Regression	0.379352	0.379438
Multilayer Perceptron Regressor	0.207942	0.207972
Deep Neural Network Regression	0.139287	0.149065

Model Name	Mean Absolute Error	Mean Squared Error
Voting Regressor	0.448506	0.365285
XGBoost Regressor	0.397286	0.374191
Linear Regression	0.521572	0.437230
Stochastic Gradient Descent Regression	0.525082	0.439884
Multilayer Perceptron Regressor	0.522036	0.561371
Deep Neural Network Regression	0.639072	0.610030

In that way, we can choose the XGBoost Regressor model as our default model that is the second highest model performance and it can be easily explained by using it's features importances and SHAP Tree Explainer.

3.2 Command Line Interface

We are also supplying our clients with a CLI (Command Line Interface) program. All third-party Python 3.8 packages requirements are being sent with the project in the *requirements.txt* file.. An example of a sample execution is in Figure 1.

```
(.env) ronaldpereira@saurfang:~/dev/coursera/advanced_ds_capstone/ $ python3 red_wine_quality_prediction_cli.py
--- Wine Quality Prediction ---
Please enter the following characteristics:
Fixed Acidity: 5.6
Volatile Acidity: 0.85
Citric Acid: 0.05
Residual Sugar: 1.4
Chlorides: 0.045
Free Sulfur Dioxide: 12
Total Sulfur Dioxide: 88
Density: 0.9924
pH: 3.56
Sulphates 0.82
Alcohol: 12.9

Your wine quality is: 8.0

Do you want to enter another wine? (Y/n): n
```

Figure 1: Sample execution of a Red Wine quality by using our CLI.

3.3 Model Explainability

We are also including a model explainability to our clients within our presentation, as shown in Figures 2 and 3.

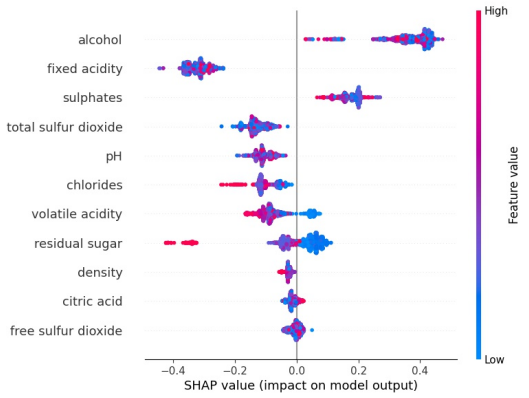

Figure 2: SHAP values for each feature in our XGBoost regressor model.

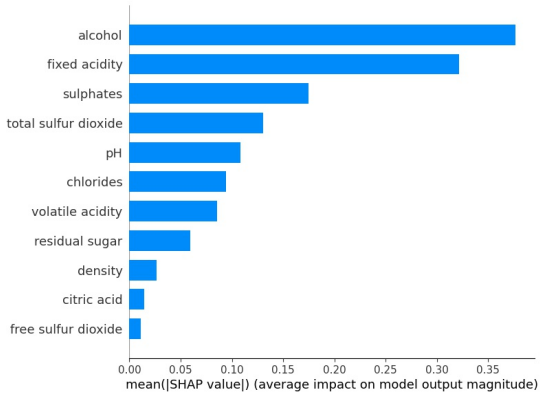

Figure 3: SHAP Feature Importances for each feature in our XGBoost regressor model.

As we can see in the images above, alcohol and fixed acidity are the most important features of our model, resulting in this being also an important feature to red wines, as our model uses real data in it. However, the separation for the fixed acidity is too difficult to see. So, we can see that low alcohol rates, low volatile acidity, and low residual sugar impacts more positively the quality of the product. In that way, they should be prioritized by red wine producers seeking to improve their product quality.