CIS 520 Fall 20200, Project Checkpoint

Wine Quality Prediction Using Machine Learning

Team Members:

- Spencer Webster-Bass; Email: websp@seas.upenn.edu
- Yunuo Chen; Email: yunuoch@seas.upenn.edu
- Jiecong Lu; Email: jiecong@seas.upenn.edu

---

1 Motivation

Wine has been an important alcoholic drink in many different cultures throughout history. Additionally, it has significant economic value. The global wine market had 378 billion dollars of total revenue in 2019 alone, and is expected to grow at an annual rate of 9.2% between 2020 and 2025. In order to succeed and profit in this highly competitive market, it is crucial for wine producers to have a detailed understanding of its product's quality. As a result, we aim to use machine learning algorithms to explore and identify the relationships between wine's quality and its various attributes.

2 Data Set

The data set can be downloaded from the UCI Machine Learning Repository[2], and is also available on Kaggle. The data set has 6497 instances (n = 6497) and 13 attributes (p = 13), which include wine type (white/red), fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, and quality.

Links To Data Set:
UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/wine+quality
Kaggle: https://www.kaggle.com/rajyellow46/wine-quality

3 Related Work

Cortez et al.[1] used support vector machine to perform regression tasks in their work "Modeling Wine Preferences by Data Mining from Physicochemical Properties.", and Keras tutorial[3] used neural nets to perform regression tasks on wine quality and also classification tasks on categorizing wine type (red/white).

4 Problem Formulation

Using the supervised machine learning algorithms listed in the Methods section, we will create a program that learns how 12 features (type, fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol) of any wine correlate with a wine's quality.

5 Methods

After inspection we noticed that there was missing information in the form of NaN values. To make our data as close to complete as possible, we applied regression based imputation. We used this imputed data for all

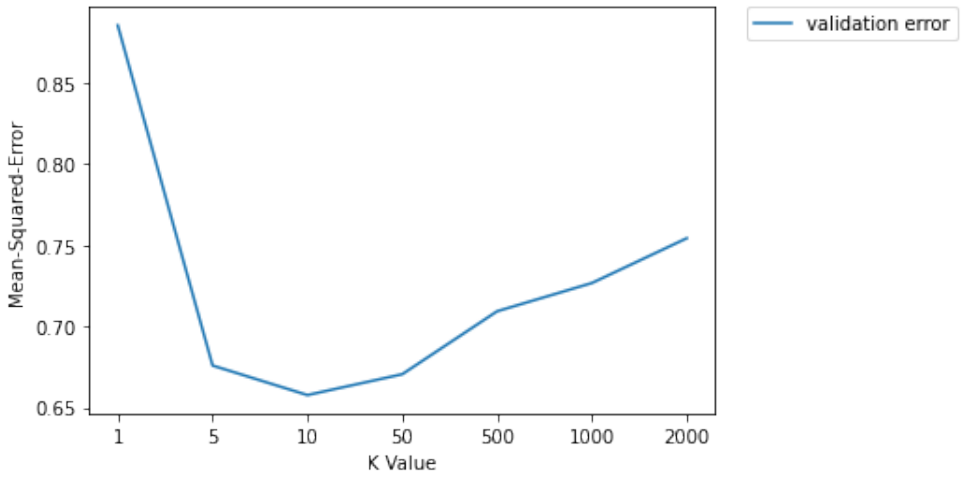
of our methods. Since we are using supervised machine learning methods, we apply 5-Fold Cross Validation to the imputed data to decrease overfitting when training models. To evaluate how well a model predicts compared to other models, we compare the mean of the squared errors (MSE) of the models. For selecting hyper parameters, we follow the same procedure, specifically we select the hyper paramters of the model to be the hyper parameters that minimize the MSE.

Our project is a comparison of multiple supervised algorithms. The supervised machine learning algorithms we plan to explore our processed data and solve our problem with include KNN, K-Means, Linear Regression, Ridge Regression, Lasso Regression, Elastic Net Regression, and Neural Nets. We plan to use Google Colab as our computing platform.

6 Experiments and Results

K-Nearest Neighbor Regression:

For k-NN we used the Scikit Learn KNeighborsRegressor module. All points in the data were uniformly weighted with the distance measure being the standard Euclidean metric. To tune the accuracy of our predictions, we tested multiple values of k specifically [1,5, 10, 50, 500, 1000, 2000]. The hyper parameter that was most accurate using the evaluation method in the Methods section was k = 10.



K	MSE
1	0.8851927610343452
5	0.6758844860310206
10	0.6577013019983317
50	0.6705293032072437
500	0.709265791397865
1000	0.7266520938374497
2000	0.7542083961309374

Ordinary Least Squares (OLS):

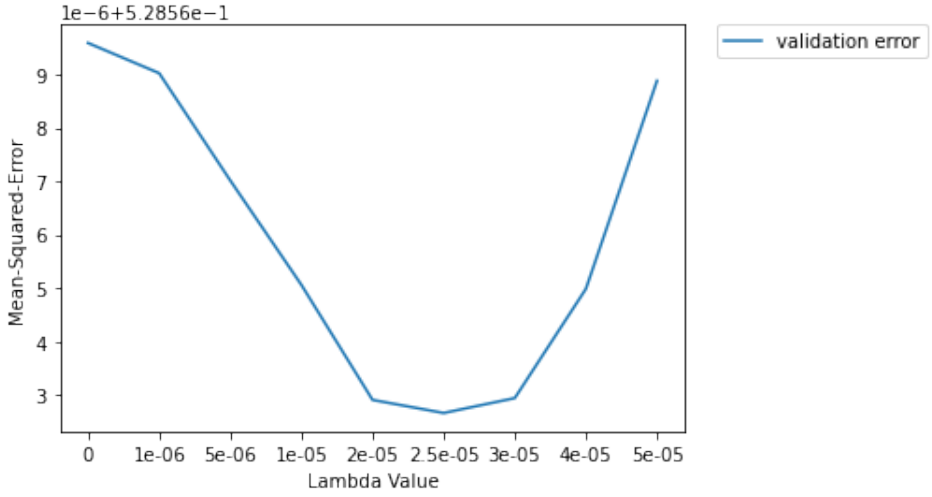
For OLS we used the Scikit Learn LinearRegression module. All points in the data were uniformly weighted with the distance measure being the standard Euclidean metric. Instead of using the y-intercept feature inside of scikit module we add a column of 1's to the data to account for this. For this model, the test MSE was 0.5645996519829907.

Ridge Regression:

For ridge regression we used the Scikit Learn linear_model.Ridge module. All points in the data were uniformly weighted. To tune the accuracy of our predictions, we tested multiple values for the regularization factor (lambda) specifically [0, 0.000001, 0.000005, 0.00001, 0.00002, 0.000025, 0.00003, 0.00004, 0.00005].

The hyper parameter that was most accurate using the evaluation method in the Methods section was lambda = 0.000025.

lambda	MSE
0	0.528569588910473
1e-06	0.528569023035523
5e-06	0.5285670136719369
1e-05	0.5285650580406622
2e-05	0.5285629049919726
2.5e-05	0.5285626599001644
3e-05	0.528562938969659
4e-05	0.5285649847718352
5e-05	0.5285688813710079



Logistic Regression:

For logistic regression we used the Scikit Learn linear_model.LogisticRegression module. All points in the data were uniformly weighted and penalized with an L2 penalty. For this model, the test MSE was 0.6635897435897435.

Neural Networks:

For the neural network we used the Scikit Learn MLPRegressor module. All points in the data were uniformly weighted. We used the lbfgs weight optimization solver which is a type of quasi-Newton methods. We also applied an L2 penalty with a regularization factor of $1 * 10^{-5}$. The neural network has 4 hidden layers each with 5 nodes. The model's test MSE was 0.5519886540076187.

7 Conclusion and Discussion

At this point we have trained and tested several algorithms to make prediction. Based on our observation, the simple model OLS can already provide certain level of accuracy (error 0.56 out of range [3,9]). Therefore, we choose OLS as our baseline model. All models accepted by this project must pass this threshold (i.e. lower error). Given this, K-Nearest Neighbor and Logistic Regression are not considered acceptable, whereas Ridge Regression and Neural Networks with appropriate hyperparameters are acceptable.

The rest part of this project will be conducted based on the above results. We further explore our models and algorithms to get better model. Possible ways for adjusting the models are: tuning hyperparameters, alternating structures, using ensemble methods etc.

References

[1] Paulo Cortez, Antonio Cerdeira, Fernando Almeida, Telmo Matos, and Jose Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009.

[2] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[3] Keras Tutorial. Deep learning in python. *DataCamp Community*.