Explorative Analysis and Predictive Modeling of Boston Housing and Wine Datasets

[Antony Zhao - Li Shichen - Yosr Takouti]

October 7, 2023

Abstract

In our first machine learning mini-project in the course of COMP551, two fundamental models; linear regression and logistic regression with Gradient Descent were implemented and analyzed using two different datasets. In our project's early phase, we worked on data acquisition, preparation, and analysis using dataframes to manipulate data effectively. To learn more about the datasets, basic statistics like dispersion and class distributions were produced seeking to improve our comprehension and spot any problems in the datasets. This project's results helped us to advance our understanding of fundamental machine learning models, statistical analysis, and data preprocessing. The best part of it was the hands-on understanding of model training, evaluation, and hyperparameter tuning. As we, Antony, Li, and Yosr, navigated through the diverse tasks and experiments on both datasets, we not only honed our technical skills but also gained a deeper appreciation for the iterative and exploratory nature of machine learning projects.

Introduction

Engaging with the Boston Housing and Wine datasets, our exploration utilized robust preprocessing and machine learning models a variant of linear Regression, to predict housing prices and categorize wines. This report unravels our analytical journey, findings, and insights, resonating with similar foundational works in these domains. Logistic regression iteratively determines the strongest linear combination of variables with the highest likelihood of identifying the observed outcome using elements of linear regression indicated in the logit scale. Selecting independent variables, making sure that pertinent assumptions are met, and selecting an acceptable model-building method are all crucial factors to take into account while performing logistic regression[1]. We performed extensive experiments throughout our analysis, including 80/20 train-test splits, 5-fold cross-validation, and studies of the effects of different training data sets and batch sizes on model performance. Insights on model behavior, convergence rates, and ideal hyperparameter configurations were gained from the results of these tests. As part of our study, we not only implemented these models from scratch but also thoroughly examined their performance indicators, such as accuracy, precision, recall, and F1-score for classification and regression, and mean squared error for regression. This report is divided into three parts starting from the part of the dataset that contains a description and visualization of data in both Boston Housing and Wine datasets then the results part that has all the outcomes of the experiments we made through this project and we concluded by the discussion and conclusion part where we discussed the most important aspects of the work.

Datasets

- **Boston Housing Dataset:** This dataset is widely used for regression problems, and it contains 12 features (with the removal of a feature with ethical problems) of 506 samples and 1 target value, MEDV, which represents the median value of owner-occupied homes in 1000 dollars. It was necessary first to look at important statistics like mean, median, and standard deviation for each feature in order to understand the scale and distribution. You can find some samples of data visualization in the appendix. https://www.kaggle.com/datasets/avish5787/boston-data-set

- **Wine Dataset:** This dataset is widely used for classification problems. The categorization of wines based on 13 chemical attributes of 178 samples of wine was analyzed. These attributes have various measurements related to their chemical composition such as alcohol content, phenol, and color intensity. The target variable here in our work is the class of wine which can be one of the three wine classes, then similar pre-processing steps were employed to ensure data accuracy and reliability. You can find some samples of data visualization in the appendix.
https://www.kaggle.com/datasets/dell4010/wine-dataset

Results

- **1. 80/20 train/test split**

We discovered some interesting results from the investigation of both datasets. In examining the Boston Housing dataset, the removal of features 'RAD' and 'TAX' appears to have had a profound impact on model performance. The MSE witnessed a remarkable reduction from 17.03 to 1.048 for the test set and 1.3134 for the training set. This substantial decrease in MSE implies a significant improvement in the model's accuracy in predicting housing prices. Figure 1 visualizes the real values versus predicted values for the Boston Housing dataset. The graph presumably demonstrates a closer alignment between the model's predictions and the actual values.
Turning attention to the Wine dataset, the classification metrics for both the test and training sets suggest a high level of model accuracy. With an accuracy score of 0.972, precision of 0.974, and recall and F1 score of 0.972 in the test set, the model exhibits robust classification performance. Impressively, the training set metrics all register perfect scores of 1 across the board, reflecting a model that excels not only in classifying instances correctly but also in minimizing false positives and false negatives. Figure 2 provides a visual representation of the model's performance on the Wine dataset. It is likely to showcase a clear demarcation between the predicted and actual classes, reaffirming the model's ability to discriminate between the different wine categories.

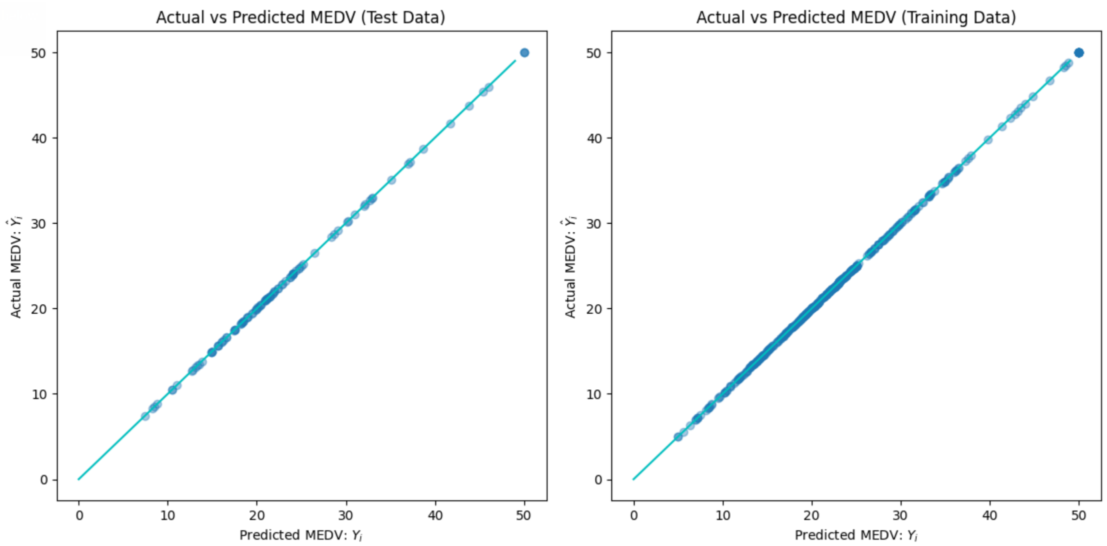

Figure 1: Real value VS Predicted Value for Boston Housing dataset for 3.1

- **2. 5-fold cross-validation technique**

These tendencies were further supported by the second experiment by cross-validation, which also showed consistent performance across folds and got a promising average of MSE being 0.0 ± 0.0 for both the training and the test of the Boston Housing dataset, which is too perfect to believe. Therefore, we put 'RAD' and 'TAX' back to the features, then we got 23.81 ± 1.52 for average test MSE and 22.28 ± 0.42 for average training MSE, which was way higher than our expectation. In order to make this statistics normal, we tried only one of the two features each time. However, dropping either one of them would make the average MSE for both training set and test set back to 0.0 ± 0.0 again, as similar as the result

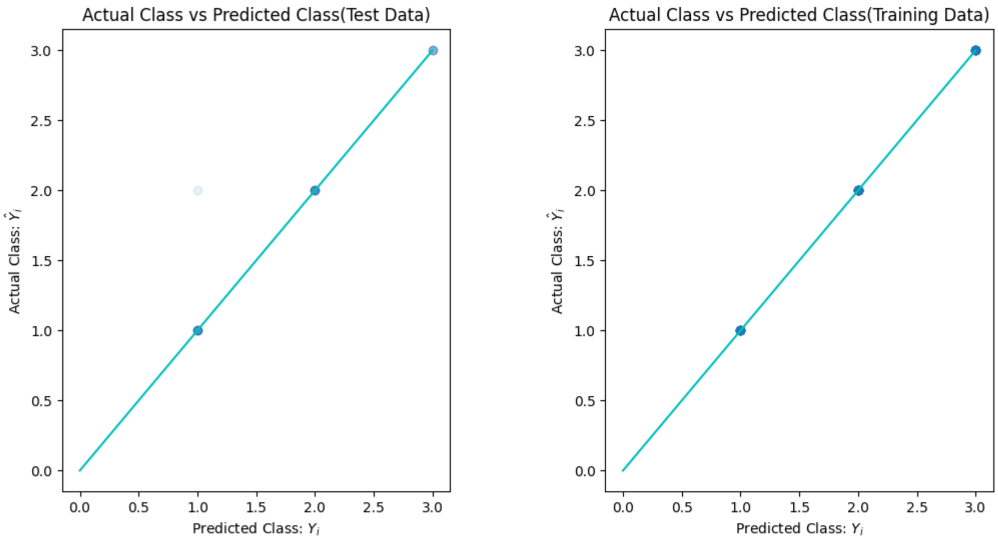

Figure 2: Real value VS Predicted Value for Wine dataset for 3.1

we got from dropping both of them.
Concerning the Wine dataset, after the 5-fold cross-validations, we got 1.0 for five times for all of Accuracy, Precision, Recall and F1 score for Training data. For test data, the we got around 0.97 to 1.0 for Accuracy, Precision, Recall and F1 score in five tests.

- **3. Sample growing subsets of the training data (20%,30%,...80%)**

As we follow up on the third experiment on one hand with the Boston Housing dataset, we noticed that when we increase the subset of the training data from 0.2 to 0.8, we got an increasing MSE from 0.0542 to 0.0643 for the train set and a decreasing MSE from 0.1645 to 0.0441 for the test set. On the other hand with the Wine dataset, we noticed that for the train set accuracy dropped over time from 1.0 to 0.992 but for the test set, it went from 0.944 to 1.0. For this experiment, you can check the graphs of MSE and accuracy in the function of training size of both datasets for respectively linear and logistic regression in Figure 3 of the appendix section.

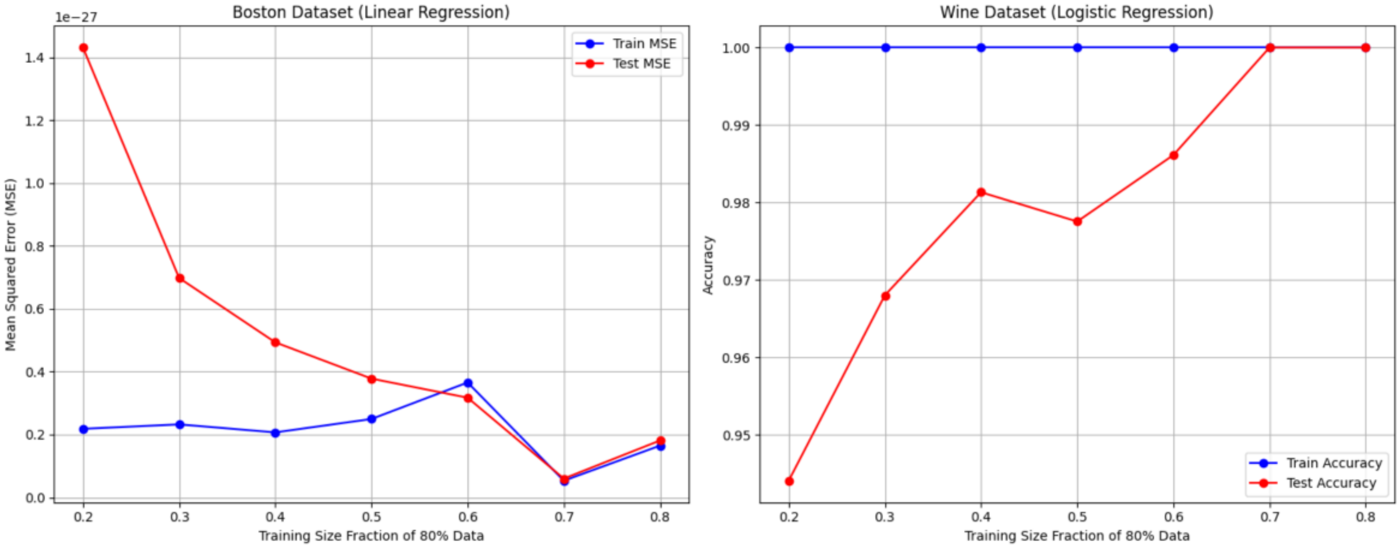

Figure 3: MSE and Accuracy flowing map according to changes of Learning Rates for 3.3

- **4. Growing minibatch sizes, e.g., 8, 16, 32, 64, and 128**

For our fourth experiment, when we increased the minibatch sizes for the SGD, we got the MSE of

9.813, 7.021, 0.0038, 0.0899, and 0.5483 for the Boston dataset using Mini Batch Linear Regression. Interestingly, with the changing batch sizes from 8 to 128, we got a fixed accuracy of 0.9444 for the Wine dataset using Mini Batch Multiclass Logistic Regression.

- **5. Five different learning rates**

  For the fifth experiment, we proceeded to compare the performance of both linear and logistic regression with five learning rates of 0.001, 0.01, 0.1, 0.5, and 1.
  For the first dataset, we got an oscillating MSE for respectively 366.18, 12.92, 0.906, 1.750, and 1.717 according to the chosen learning rates; given that the linear model would work better with a learning rate of 0.1.
  For the second dataset, the accuracy was fixed a 0.972 for the 0.001, 00.1, and 0.1 learning rates and achieved its best of 1.0 for the 0.5 and 1 learning rates. So the the logistic model would work better with a learning rate of 0.5 and 1.

- **6. variety of parameter configurations**

  As we proceeded to our sixth experiment, we chose to test the best configuration for both of the dataset by changing the value of learning rates and max iterations list. Our learning rates list contains five different learning rates: 0.001, 0.005, 0.01, 0.05, 0.1. Our max iterations list contains 100, 500, 1000, 1500, 2000.
  After trying multiple times, we got the best configuration for the Boston dataset is to train with the model with 0.1 for the learning rate and 500 for max iterations, and the lowest MSE we get is 0.89995. The best configuration for the Wine dataset is to train with the model with 0.001 for learning rate and 100 for max iterations, and the best accuracy we get is 0.97222.

- **7. Gaussian Basis Functions**

  Coming to the most interesting experiment of our series, we have implemented Gaussian basis functions to enrich the feature set for the Boston dataset. We got 1.331 for MSE in the train set and 1.230 for the test set which are lower than the model without them then the additional features have effectively contributed to the model's performance.

- **8. Comparison of analytical linear regression solution with mini-batch stochastic gradient descent based linear regression solution**

  Last but not least, as you can see in Figure 4 for the results of experiment 8 respectively for the analytical linear regression and the mini-batch SGD, we got an MSE of 1.0484 for linear regression, and 0.003891 for the the mini-batch SGD for linear regression. Detailed metrics and visualizations are presented in the Figure 4.

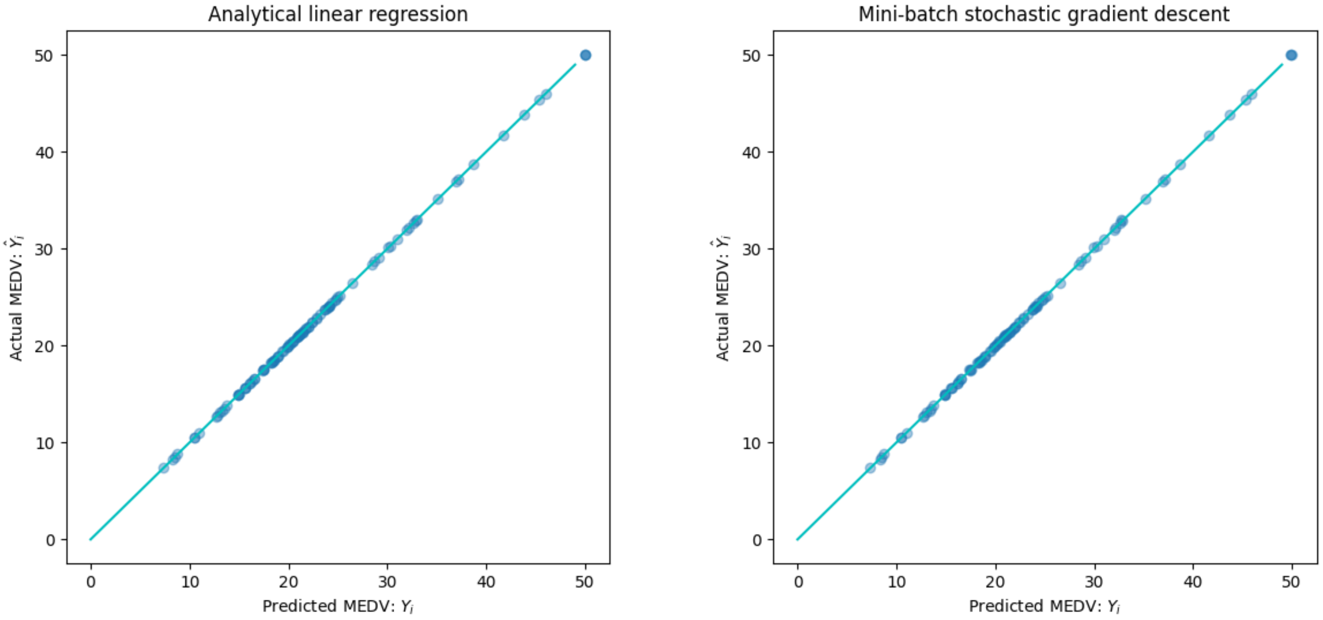

Figure 4: Analytical linear regression VS Mini-batch SGD linear regression for 3.8

Discussion and Conclusion

This project came to be a great opportunity for us to go through linear, logistic regression, and mini-batch SGD optimizer through a comprehensive exploration using two benchmark datasets. We went from preprocessing and analyzing data to implementing models and doing several experiments on them to compare the outcomes and get the best performance optimization.

In fact, starting from feature preprocessing, we saw that using normalization and handling missing data techniques could significantly increase model performance over time. Also by adding L2 regularization to the loss function, we noticed that it helped from overfitting and improved generalization. Additionally, our feature analysis indicated that certain features might be tangential to the target, complicating the model's fitting and prediction tasks. Omitting such features can lead to significant improvements in model accuracy.

The feature elimination strategy in the Boston Housing dataset led to a substantial improvement in predictive accuracy, as evidenced by the significant reduction in MSE. The Wine dataset, on the other hand, showcases a model with impressive classification metrics, particularly in the training set where all metrics reach perfection. These findings underscore the importance of feature selection in enhancing model performance in both regression and classification tasks.

Another key observation was the balance required in mini-batch SGD between convergence speed and accuracy. While mini-batch SGD demonstrated potential for faster convergence, tuning the hyperparameters requires a lot of effort, particularly for the learning rate and batch size. The choice of convergence criteria, especially when considering the noisy nature of the gradient in SGD, emerged as a crucial aspect of the optimization process.

In the future, in order to improve the performance of the models, investigating more into optimization methods such as RMSprop[2], momentum[3], and Adagrad[4] can be helpful. At the same time, when dealing with data collected, it is a good practice to perform data analysis and preprocessing to make sure only ”good” data is being fed into the model.

Statement of Contributions

Each team member significantly contributed across all phases of the project, ensuring a holistic involvement in preprocessing, modeling, and documentation.

Appendix

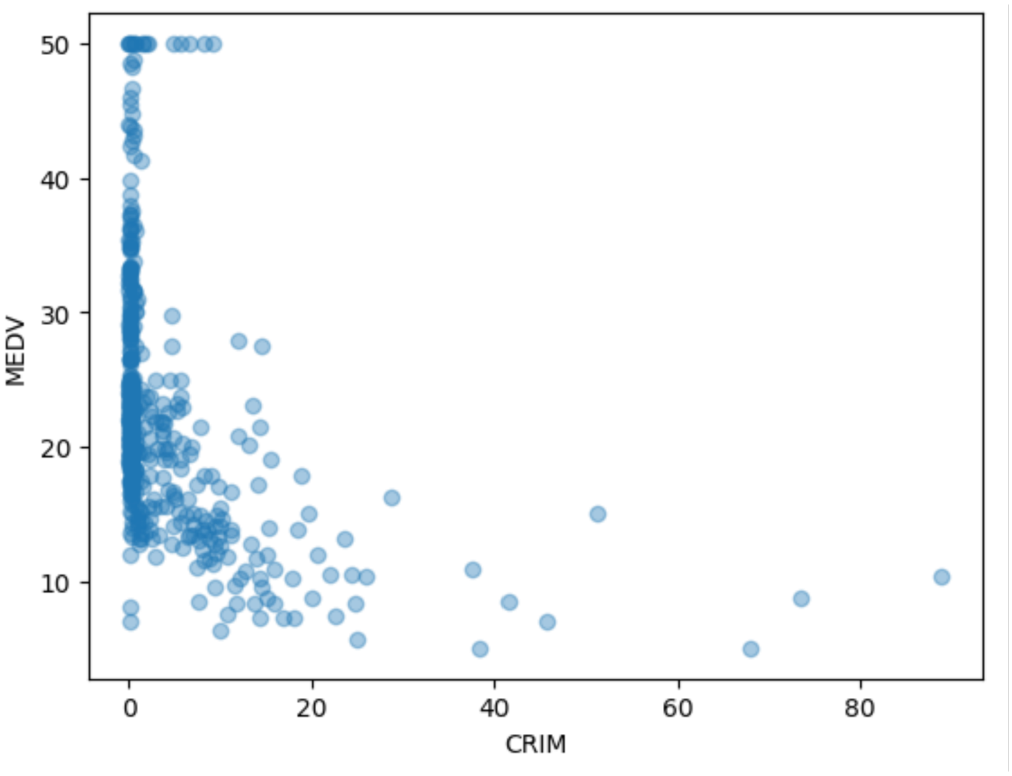

Figure 5: Data Visualization CRIM for Boston Housing dataset

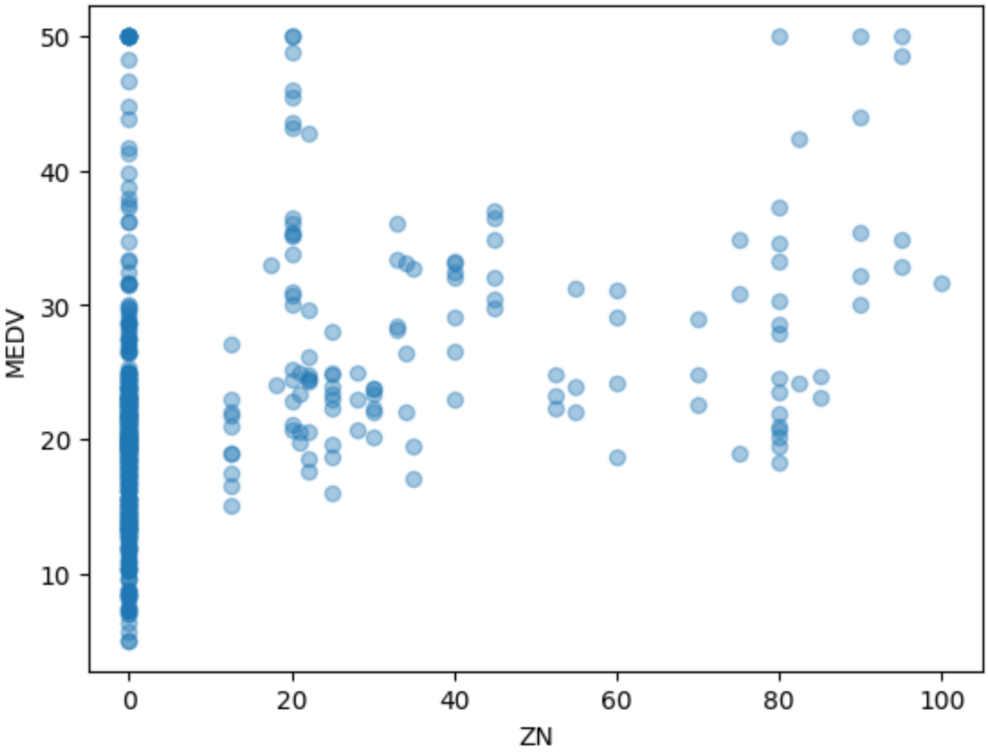

Figure 6: Data Visualization ZN for Boston Housing dataset

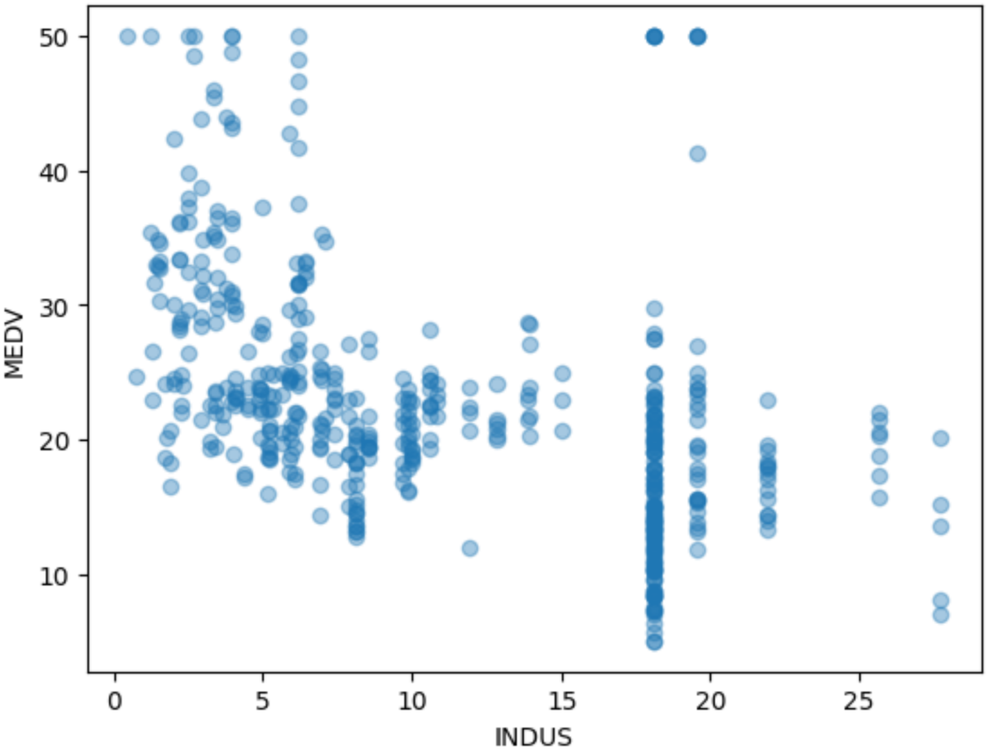

Figure 7: Data Visualization INDUS for Boston Housing dataset

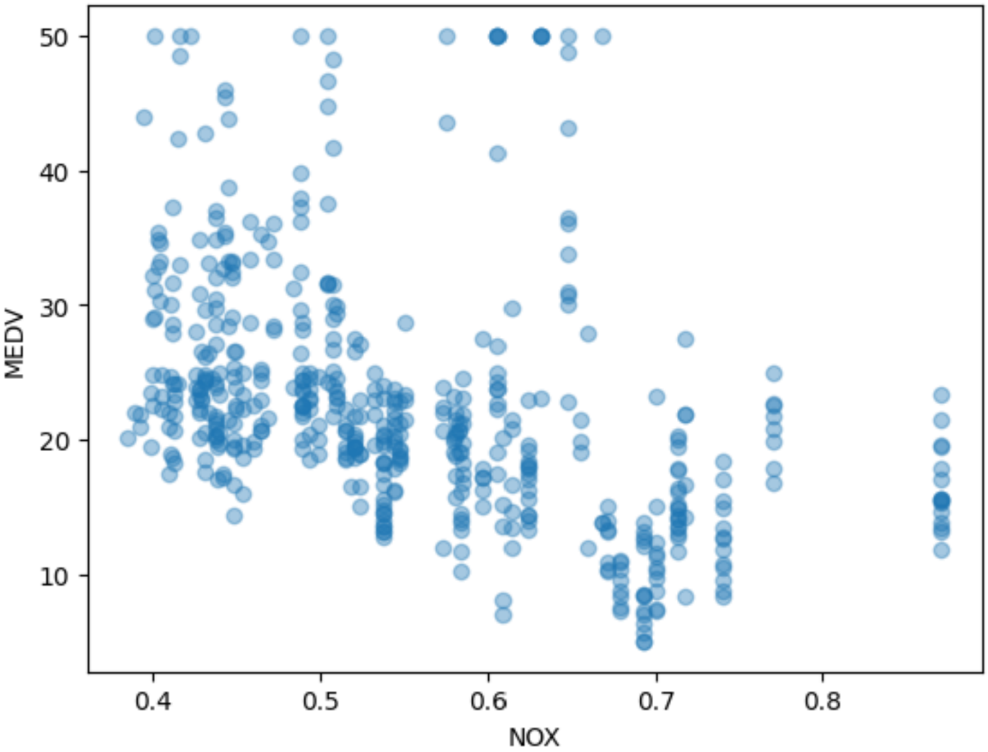

Figure 8: Data Visualization NOX for Boston Housing dataset

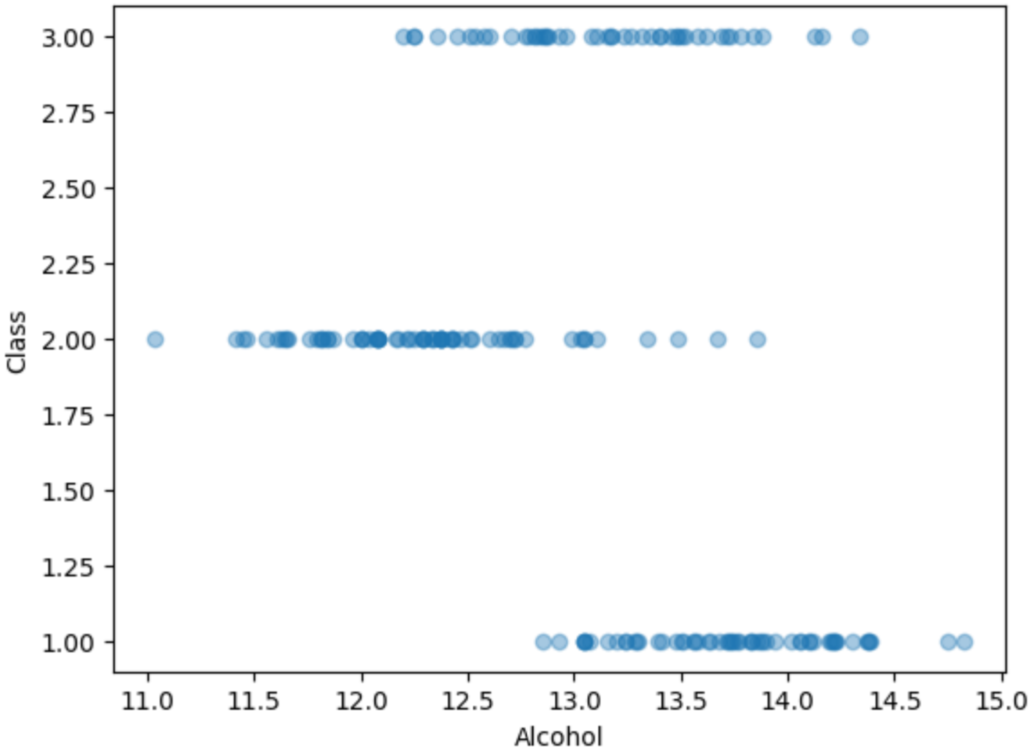

Figure 9: Data Visualization Alcohol for Wine dataset

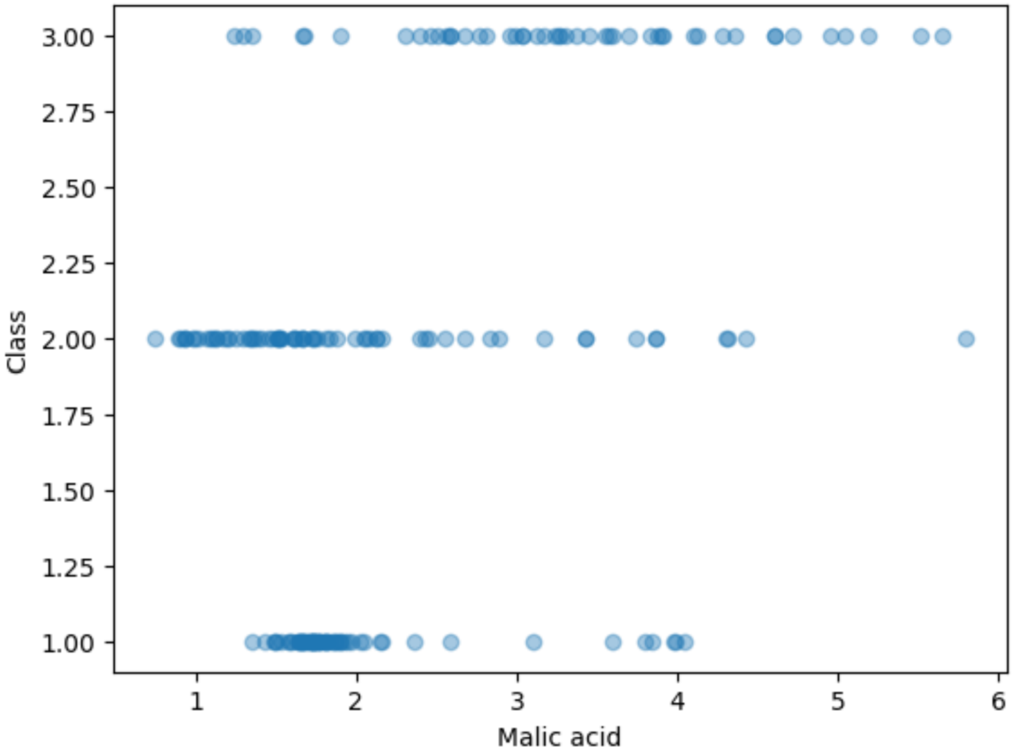

Figure 10: Data Visualization Malic acid for Wine dataset

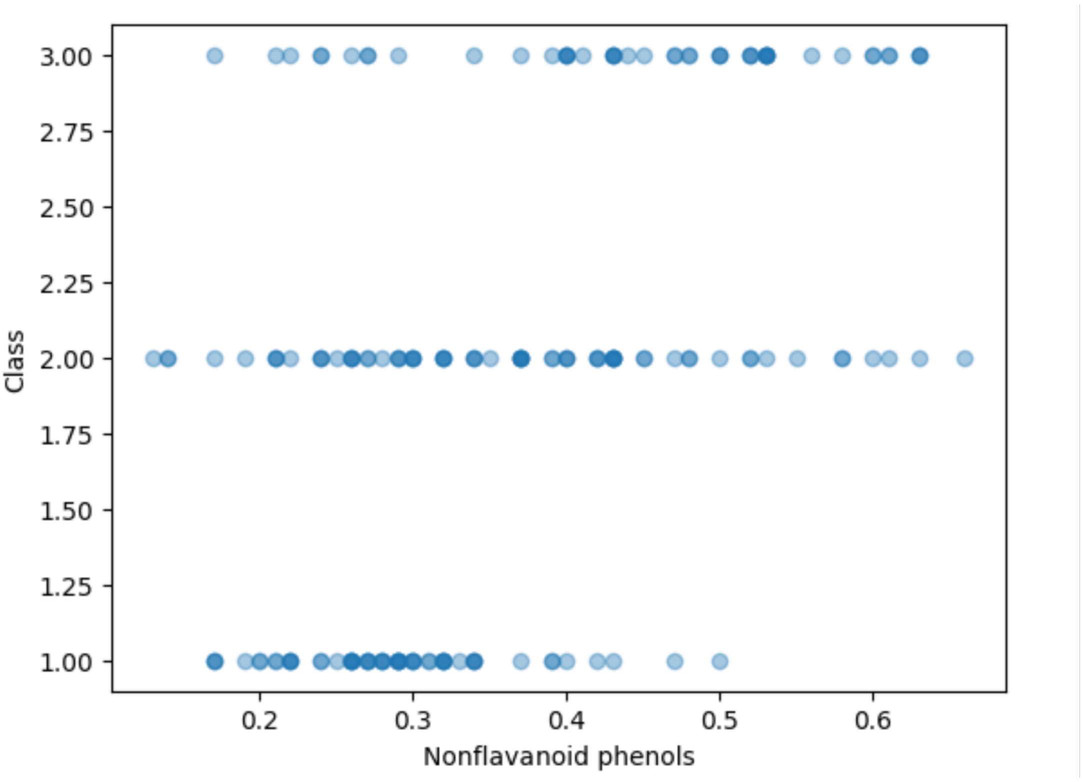

Figure 11: Data Visualization Nonflavanoic phenols for Wine dataset

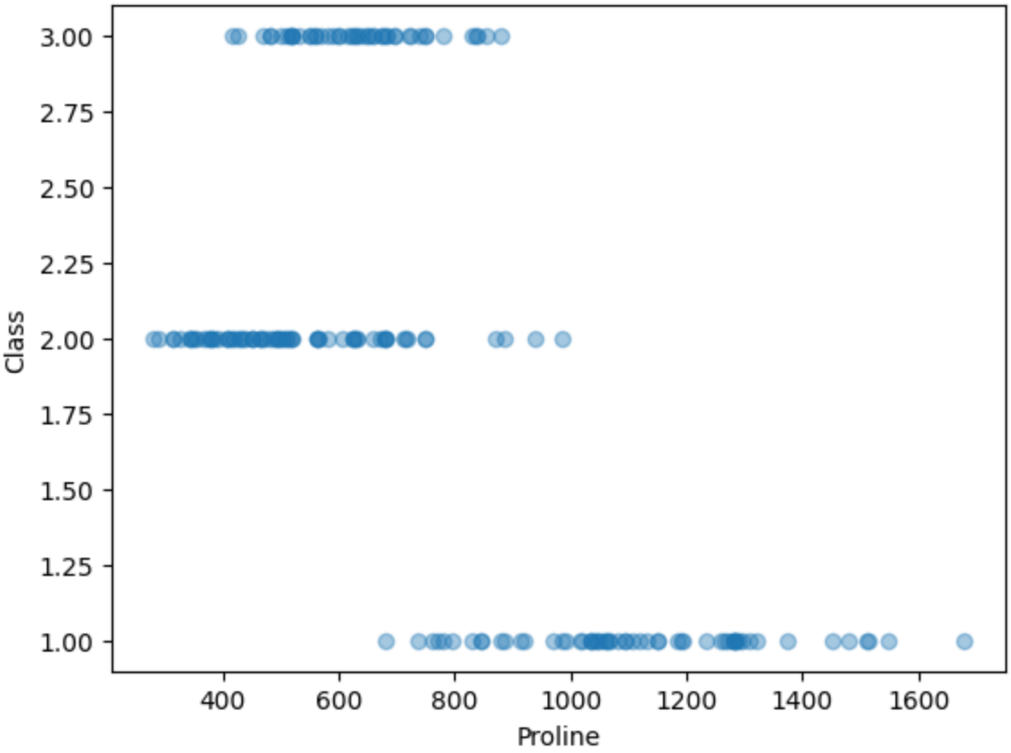

Figure 12: Data Visualization Proline for Wine dataset

References

1. [Stoltzfus, J. C. (2011). Logistic regression: a brief primer. Academic emergency medicine, 18(10), 1099-1104.]
2. [Hinton, G. (2012). Neural Networks for Machine Learning. Coursera video lectures.]

3. [Gorbunov, E. A., Bibi, A., Sener, O., Bergou, E., Richtárik, P. (2019). A Stochastic Derivative Free Optimization Method with Momentum. arXiv preprint arXiv:1905.13278.]
4. [Duchi, J., Hazan, E., Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Journal of Machine Learning Research, 12, 2121-2159.]

Acronyms

MSE Mean Squared Error.

RMSE Root Mean Squared Error.

SGD Stochastic Gradient Descent.