COMP551-A1

Daniel Chernis 260707258
Aidan Sullivan 260733921
Mashood Ahmed 260562403

September 28 2019

Abstract

In this project the performance of linear classification models (Logistic Regression model and Linear Discriminant Analysis model (LDA)) on two benchmark datasets (red wine and breast cancer) was investigated and analyzed. We found that the logistic regression approach has achieved worse accuracy than LDA and was significantly slower to train. For LDA The accuracy found was 74% in prediction was found for red wine whereas 96% accuracy was found for breast cancer data. For Logistic Regression the accuracy is 91% for cancer data and 74% for wine data. Additionally, after testing an increase in learning rates, Logistic regression accuracy peeks with a learning rate of 2, 400 iterations, and approaches LDA accuracy.

1 Introduction

The task of the project was to examine 2 different machine learning models. Those are Linear Discriminant Analysis (LDA) and Logistic Regression. These 2 models were implemented based on their specification, and then extensively tested, mostly comparing their running times, their accuracy on the provided datasets, and the effects of the hyperparameters for the functions. Two datasets were used to test the implemented models. Both were provided from UCI, one of them was the Wine Quality dataset, specifically the red wine set. The other dataset was the Wisconsin Breast Cancer dataset.

These datasets were automatically filtered when they're loaded in by the program. In order to test the accuracy of each model, a k-fold prediction algorithm was implemented. It optionally shuffles the dataset before sending it through the k-fold cross examinator.

2 Datasets

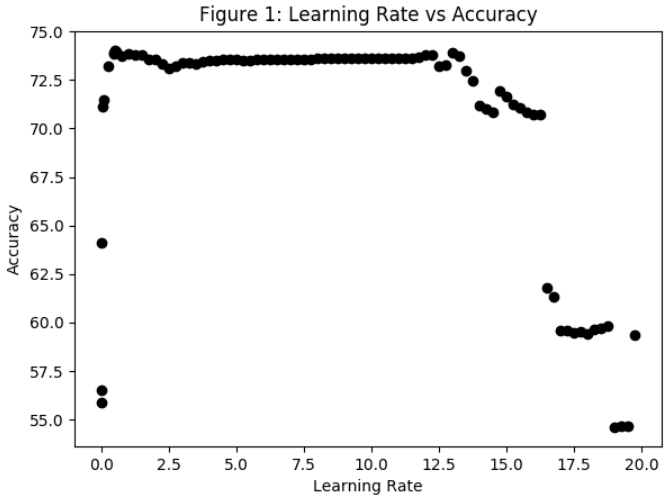
There are 2 datasets that were used. These are the UCI datasets wine quality (red) and Wisconsin breast cancer. To use the data presented here, the data was

passed through a simple filter. The filter would first remove all datapoints that were invalid and didn't follow the correct format as the other points. Secondly, the target field to be used as an output was converted to a binary value (1,0) based on where it lied on a given threshold specific to each dataset.

Finally the dataset was normalized, which is to say all the data went through a function that squeezed the data between 0 and 1. While this step is not strictly necessary, it is popular for many ML applications as it makes the data more uniform and easier to evaluate.

3 Results

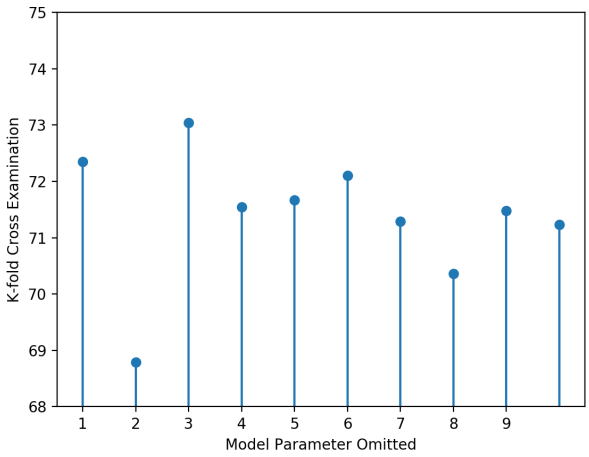
The first experiment explored the effect of varying the learning rate on the logistic regression model. This test was completed using the red wine dataset. The figure below illustrates the results. Around the highest point at approximately 1, more finer points were also taken to explore the most effective point. The graphs seem to fall off before 1, plateau until 12.5 and then fall off again.



In experiment 2, the raw effectiveness was examined between the different models on the datasets. The initial results show that LDA and Logistic Regression results were similar with slight variations. The logistic regression model used its default parameters of 0.5 for a learning rate and 1000 iterations.

To improve accuracy, it was hypothesized that there may be a feature that is introducing noise into the system. As such, tests were made that removed one feature from the feature set each time and ran the logistic regression. In addition several different model parameters we're substituted with different values to converge on the best result. After experimenting with several parameters to remove, and changing the input parameters to the model, it was determined that The 9th column from the wine dataset could be removed without affecting

results, which was the pH levels column. The other minor tweaks were to make the learning rate 2 and having 400 iterations. These changes reduced the running time and marginally increased accuracy from 74.04% to 74.23%.



In other testing, when the number of iterations was varied, the running time scaled with it, but accuracy was improved. For example, when the number of iterations was increased from 1000 to 10000, the model further improves accuracy by 1%. That being said, this also increased the running time by 10 times, which is quite significant.

4 Discussion and conclusion

From experiment 1, it is clear that logistic regression improves in accuracy as the number of iterations and the learning rate increases. However there is a ceiling on this. At some point the learning rate gets so high (around 12.5) that it oscillates around the ideal point and eventually ends its iterations on a non-ideal point. This is why the accuracy for a non-ideal learning rate provides a response that's far off from ideal. When the learning rate is too low, there is the risk that it doesn't converge to a value in time. These can all be directly observed in figure 1.

Experiment 2 demonstrated that each of the models are similarly effective. Logistic regression takes longer to fit a model, however they are still effective since their parameters can be more easily tweaked. This can provide slight accuracy gains despite being slower on average.

Logistic Regression may appear less accurate at the beginning, but with a powerful enough processor and enough time, the number of iterations may be increased and thus make that model more accurate. With a good learning rate, then the value with find a local minima as by the time it completes all of its

iterations. This implementation gave it 1000 iterations, which is likely enough to get it to converge to a local minima. In experiment 3 it was possible to lower the number of iterations safely as long as the learning rate was an appropriate size that it would still converge. This experiment managed to lower it down to 400 iterations without reducing accuracy by increasing learning rate to 2. The results from this experiment were fairly marginal, which suggest the initial arbitrary values that were chosen as temporary learning rates and iteration counts were not too far off from ideal. It was observed that modifying the starting loss value didn't have a huge impact on final result.

LDA is a deterministic approach. It is able to perform a dimensionality reduction technique that reduce number of dimensions (i.e. variables) in a dataset while retaining as much information as possible. LDA is not based on any distribution assumptions. We only implicitly require equality of the population covariance matrices. Under the additional assumptions of normality, equal prior probabilities and miss-classification costs, the LDA is optimal in the sense that it minimizes the miss-classification probability. The most important drawback is loss of information. This is less of a problem when the data are linearly separable, but if they are not the loss of information might be substantial and the classifier will perform poorly. There might also be cases where the equality of covariance matrices might not be a tenable assumption. If it is found that the populations are normal with unequal covariance matrices a quadratic classification (QDA) might be used instead.

5 Statement of Contributions

Aidan wrote the code to organize the models. This is the model.py, helpers.py and the main.py files, including the K-Fold function. He tested the data to improve accuracy for the given models. Aidan contributed to the logistic regression model. Aidan completed experiments 1-3. Aidan helped write the report.

Daniel wrote the code for the LDA model and ran it for both data sets. This implies writing a fit(), predict(), and accuracy function for LDA. Daniel also wrote the data pre-processing and cleaning used to extract the valid/usable for modelling. Daniel helped write the report.

Mashood wrote the code for Logistic Regression with gradient descent.