Machine Learning 101: Implementing Logistic Regression and LDA from scratch

Hair A. Parra B.[1], Gengyi Sun[2] and Hao Shu[3]

Abstract— **In this project, we explored the implementation "from scratch" of two classic supervised classification algorithms on a binary classification task: Logistic Regression and LDA, linear discriminant analysis. We evaluated them on two benchmark data sets, which we first preprocessed using standard data cleaning techniques such as normalization. We also explored some basic feature engineering as well as the implementation of validation techniques such as cross-validation and grid parameter search. We found out that although both algorithms yielded similar performance, given that the datasets are small, the running time was slower in Logistic Regression due to the richer complexity and hyper-parameters.**

I. INTRODUCTION

Although contemporary state-of-art classification methods we count with are increasingly popular and efficient, it is often a good idea to first try classical machine learning algorithms, since they can, often be more quickly implemented and more easily explained. In this project, we explored from-scratch implementations of two popular machine learning algorithms : **Logistic Regression** and **Linear Discriminant Analysis** , examples of discriminative and generative learning algorithms, respectively. Excellent resources describing the mathematical formulations can be found at Standford CS229 notes by Andrew Ng (see section[1] and [2]). The data sets we employed are the **redwine dataset**, which consists of red wine samples from the north of Portugal [3], as well as the **Breast Cancer Wisconsin**, which consists of Dr. Wolberg reports in his clinical cases[4]. In addition to the basic algorithm implementations, we also applied standard data cleaning techniques, basic feature engineering (to be described later in this paper) and also implemented common procedures such as data-shuffling, cross-validation, grid search and accuracy metrics.

II. DATASETS

A. Red wine quality dataset

The **red wine quality dataset** contains information on 0. fixed acidity , 1. volatile acidity ,2. citric acid , 3. residual sugar, 4. chlorides, 5. free sulfur dioxide, 6. total sulfur dioxide, 7. density, 8. pH, 9. sulphates, 10. alcohol, and quality level (ranging from 1-10), for 1599 distinct data points. Since we are working on a binary classification task, we assigned the label "1" = "good quality" to all data points with quality ≥ 6 and label "0"="bad quality" otherwise. For these labels, we obtained the following distribution showed in Figure 1.

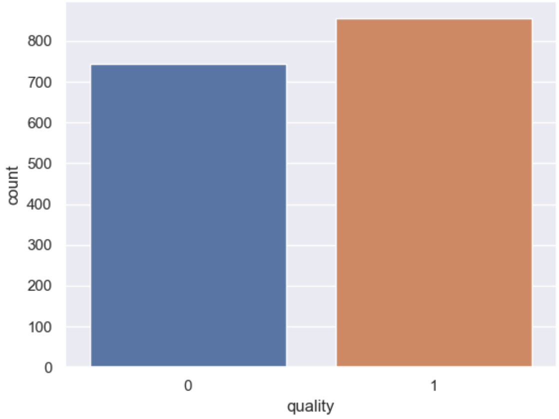

Fig. 1: The redwine dataset counts with 855 'good quality' targets and 744 'bad quality' target points.

Additionally, since the original features have significantly different scales, we normalized the original data. Finally, we observe that the data is somehow balanced.

B. Breast Cancer dataset

The **Breast Cancer dataset** contains information on attributes "Clump Thickness", "Uniformity of Cell Size", "Uniformity of Cell Shape", "Marginal Adhesion", "Single Epithelial Cell Size", "Bare Nuclei", "Bland Chromatin", "Normal Nucleoli" and "Mitoses", in order to predict whether the cancer is Benign ('Class'=4) or Malign ('Class'=2). We recorded these as "0" and "1", respectively. The distribution is showed in Figure 2.

For this data set, we also removed all rows with missing entries, and normalized each feature column as well. We do observe that there is a bigger imbalance in the distribution, which might affect actual performance.

C. Creating new features for the red wine data set

In order to improve the accuracy on the red wine data set, we decided to explore two basic streams, which are described below. 1). attempting to add higher-order features (logarithmic, interaction terms, quadratic terms), as well as

[1] **Hair Parra** is a 4th year, B.A. student at McGill University, major in Computer Science, Statistics and Linguistics with focus on NLP and Machine learning. Email: jair.parra@outlook.com Site: https://blog.jairparraml.com/

[2] **Gengyi Sun** is a third year software engineering student at McGill University.

[2] **Hao Shu** is a third year software engineering student at McGill University

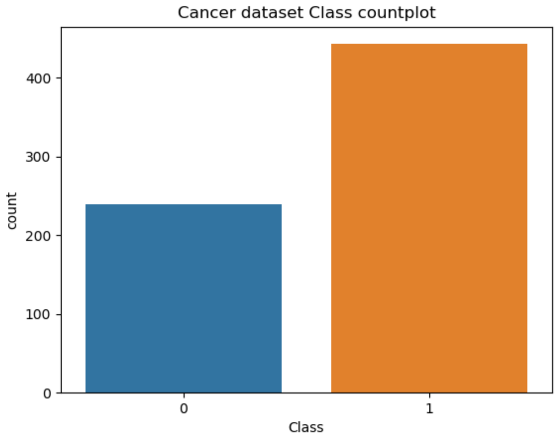

Fig. 2: The breast cancer data set consist of 443 ’1=Benign’ counts, and 239 ’0=Malignant’ counts.

2). finding the best subset of the current features. In the first case, we also made sure to add only features that had a Pearson correlation coefficient of at least $p \geq 0.6$, then, we concatenated these to the original features, and performed 5-fold cross-validation search with different alpha rates and maximum number of epochs parameter (since we are using gradient descent-based training). For the second case, we tried all possible subsets of features (except for single features and the empty subset), and obtained their respective accuracies using 5-fold cross-validation with fixed alpha rate = 0.002 and maximum number of 150 epochs. By this procedure, we found that the best features were obtained using columns [1,2,5,6,7,9,10]. The parameter configurations of the best accuracies obtained by these methods are reported in the *Results* section.

D. *Ethical Issues*

These data sets are relatively ”easy” data sets, quite well organized and rather clean. In addition, their size is also relatively small, and in the case of the cancer data set, we can see that not only it is unbalanced, but also it is quite outdated as well (1992). It is not ridiculous to think that contemporary data in both cases might be quite different, and therefore models trained on these might actually not be useful in the real world. If such models were to be used in the real world, it is possible that they might actually be very wrong. In particular, consider the cancer data set: since it is unbalanced, it makes it easier for a model to produce biased predictions and get high accuracy. This would mean that we would be having a lot of false positive predictions in patients, diagnosing them as having a ”benign” type of cancer when in reality it could be malignant, which could put the patient’s life in greater risk. In brief, although these data sets are useful for learning and research purposes, bigger, better documented and more balanced data sets should be employed in practice.

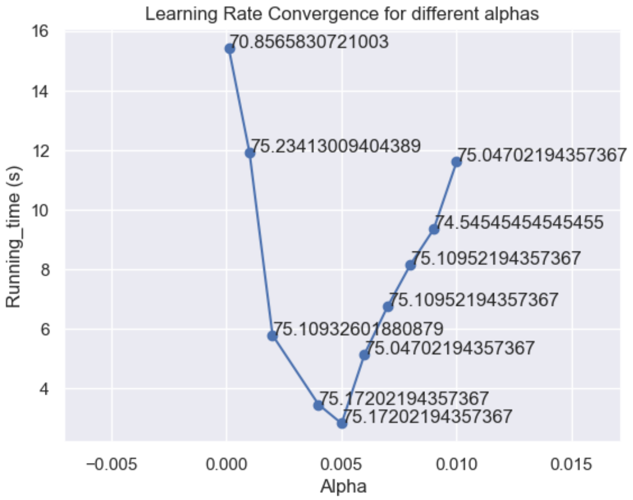

Fig. 3: Running Time vs Learning Rate for small alphas

III. Results

A. *Logistic Regression and Learning Rates*

As previously described, in order to study the impact in both performance and accuracy of the model, we decided to try the following learning rates `alphas = [0.001, 0.01, 0.1, 0.5, 1.0, 1.5,2.0]`. For fixed hyper-parameters `threshold=0.1, epochs=100,`, the results can be observed in figure 4. We observe that for a smaller learning rate (0.001) , not only we obtain better performance (74.48% acc) , but also we obtain faster convergence, while for bigger learning rates overall we get lower accuracy and convergence performance (with a lot more variance).

We then decided to explore with smaller alpha rates, in particular, `alphas = [0.0001, 0.001, 0.002, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01]`. Referring to figure 3, we observe that from 0.001 to 0.005, we indeed obtain a faster convergence rate, however, our best accuracy is obtained when `alpha = 0.001` .

B. *Comparison of between LDA and logistic regression*

1) Runtime: We tested the two algorithms with the wine dataset, and found that the logistic regression had a run time of $4.96s \pm 42.31\mu s$ whereas LDA had a run time of $0.02s \pm 56.45\mu s$, indicating that LDA is 199.43% faster. Although this is relative to the machine in which the tests are performed, it can be seen that the running time is significantly faster for the LDA. We stipulate that this is the case since Logistic Regression is an iterative algorithm, and the data set sizes are rather small. However, with bigger data sets, the matrix multiplication, transpose and in particular inversion operations might significantly affect the running time.

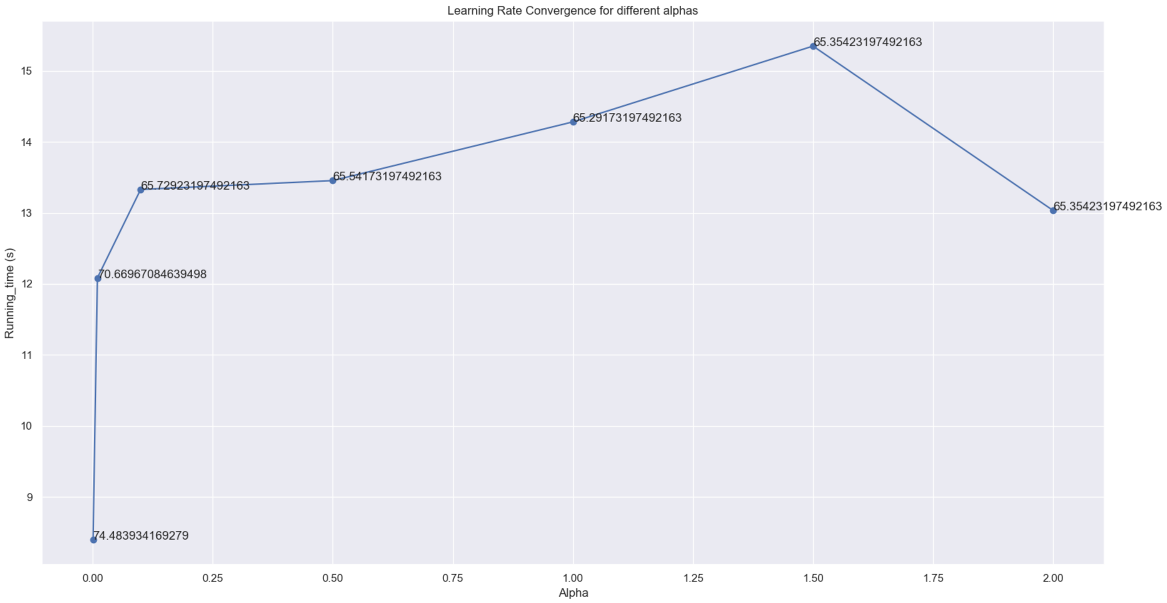

Fig. 4: Running Time vs Learning Rate

2) Accuracy: Without feature engineering, on the original red wine data set and using the best parameters `alpha=0.001` and `epochs=100` found with grid search cross-validation, we were able to obtain 74.48% cross-validation accuracy with Logistic Regression. On the other hand, LDA yielded almost the same cross-validation accuracy, 74.42%. Similarly, for the breast cancer dataset, Logistic Regression yielded 97.22%, while LDA yielded 95.9%.

C. Further Exploration to Improve Performance

1) Exponential, Quadratic and Interaction Terms: We discovered that additional feature sets composed of exponential, quadratic and/or interaction terms did not help improving accuracy significantly. In fact, using only interactions yielded a cross-validation accuracy only almost as good as the one we obtained without them. We also tried using logarithmic features, but these produced numerical instability and precision underflow, so we do not report them here. The results can be observed in the following table:

	Learning Rate	Epochs	Accuracy
Exponential	1.0	150	66.04%
Quadratic	0.001	100	74.17%
Interaction	0.001	50	74.23%
Quadratic + Interaction	0.001	150	73.67%
Original	0.001	100	74.42%

2) Trying all possible subsets: By exhausting all possible feature susbset combinations on the red wine data set, we were able to find a subset that slightly increased our model's. The resulting best feature subset contains the features fixed acidity, volatile acidity, chlorides, free sulfur dioxide, total sulfur dioxide, pH value and sulphates. By making such feature selection, the model achieved an accuracy at 75.11%.

IV. Discussion and Conclusion

Sometimes, the best way to understand a concept is doing it yourself. In this project, we had the opportunity to implement from scratch Logistic Regression and Linear Discriminant Analysis, two classic examples of discriminative and generative machine learning algorithms, thus getting a deeper insight on these. We were able to study important standard practices such as data cleaning, cross-validation for model selection, basic feature engineering and the importance of the impact of hyper parameter tuning on performance.

Further exploration can account for improvements in the algorithms implementations, further feature engineering amongst the best subset of features we found, obtaining bigger sets as well as further dividing the original data set in traintest sets and performing cross-validation only on the train set. We would also like to explore application of Principal Component Analysis (PCA) along with LDA, which can potentially further improve accuracy (see [5]). Finally, we would like to apply different algorithms such as Naive Bayes and SVM, and compare performance.

V. Statement of Contributions

All members have made significant contributions towards this project. The amount of work for each member is described as follows:

Hair Parra : Github Repository setup, data cleaning, Logistic Regression implementation, data shuffling, accuracy, cross-validation and grid search CV functions, feature engineering, visualizations, report write-up contribution.

Gengyi Sun : LDA algorithm implementation, source research, code testing, alpha learning rate exploration, report write-up contribution.

Hao Shu : LDA algorithm implementation, source research, code testing, data analysis, report write-up contribution.

VI. Source Code

The code used for this project can be found publicly at https://github.com/JairParra/Logreg_LDA_from_scratch .

References

[1] A. Ng, "5. logistic regression, cs229 lecture notes," *online:* *http://cs229.stanford.edu/notes2019fall/cs229-notes1.pdf*, 2019.

[2] A. Ng, "5. logistic regression, cs229 lecture notes," *online:* *http://cs229.stanford.edu/summer2019/cs229-notes2.pdf*, 2019.

[3] P. Cortez, "Wine quality dataset," *online:* *https://archive.ics.uci.edu/ml/datasets/wine+quality*, 2014.

[4] D. W. H. Wolberg, "Breast cancer wisconsin (original) data set," *online:* *https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original)*, 1992.

[5] Q. L. Jieping Ye, Ravi Janardan, "Two-dimensional linear discriminant analysis," *online:* *http://papers.nips.cc/paper/2547-two-dimensional-linear-discriminant-analysis.pdf?fbclid=IwAR16lwZohntu0sqh3nzYyfx74fkvKQENvG1EWMvKUaDczR1LVmWkh_QbmQg*.