Analysis of Logistic Regression and Naive Bayes for four datasets

Edwin Pan, Eric Shen and Suzy Liu

***Abstract*— In this project, we investigated the performance of two classification models: Logistic Regression and Naive Bayes Classifier. We applied these two models to four datasets in order to compare their accuracy and have a better understanding of generative and discriminative models. We found that Logistic Regression yielded slightly higher accuraciy overall at the cost of lower performance (slower) when compared to the Naïve Bayes model. Also of note is that we applied normalization to our datasets in order to improve the classification accuracy of two models.**

I. Introduction

Preliminaries:

Linear Regression and Naive Bayes are two popular models used in machine learning and there are ongoing comparisons between discriminative classifiers and generative classifiers. In this experiment, we implemented these two linear classification models in order to investigate their performance and accuracy on four datasets. The four datasets we used are: the Ionosphere Dataset, the Adults Dataset, the White Wine Quality Dataset (Wine dataset in short), and the Breast Cancer Wisconsin Datasets (diagnostic) (Cancer dataset in short). Unlike the other three datasets which have purely numeric features, the adults dataset has textual features. So, we applied one-hot encoding to the adult dataset in order to able to apply Logistic Regression (LR) on it. We did, however, try to create a model that would naturally tolerance of these textual features. In particular, we created what we called a Hybrid Naive Bayes model which follows the principle of Naive Bayes but adapts itself to binary, categorical, and Gaussian features.

Project goal:

In this project, we planned to find out the difference between discriminative classifiers[1] and generative classifiers[2]. We first pre-processed all the datasets by analyzing and cleaning the data entries obtained from them accordingly. We implemented and ran both models on four datasets and then predicted the outcomes for each dataset: whether the radar return from ionosphere is good or bad, for the ionosphere dataset; whether an individual's income is over 50K, for the adults dataset; whether wines are considered good or bad, for the wines dataset; and whether tumors are benign, for the cancer dataset. We also explored and compared the performances of our classifiers as we evaluated the classification accuracy of the classifiers on each datasets. We also tested out different learning rates on our logistic regression ionosphere model so to analyze the influence of learning rate on this classifier.
In particular, for datasets of n features, we have those features represented by an input instance vector as

$$\{X_1, X_2, \ldots, X_n\}$$

and we have a target Y. For Logistic Regression model, it directly models the posterior probability[3] of

$$P(y \mid x)$$

but for Naive Bayes, it models the joint distribution of the feature X and target Y, and then predicts the posterior probability given as

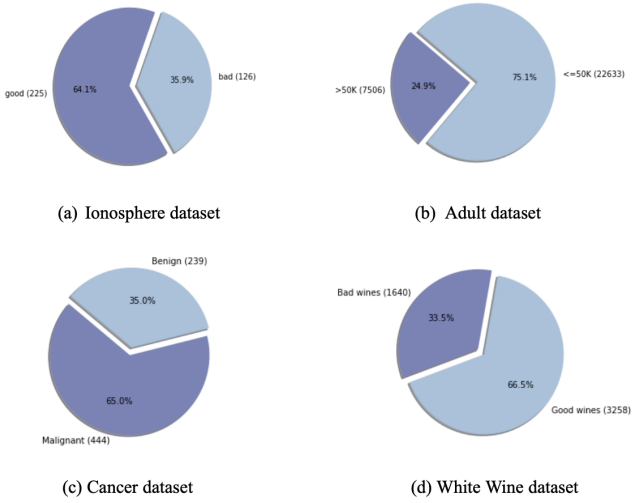
$$P(y \mid x)$$

Findings:

Discriminative models and generative models are two categories of models that are commonly used in machine learning. Logistic regression as a discriminative model, studies the relationship between one categorical dependent variable and a set of independent variables, it estimates and

classifies data entries based on probability[4]. In contrary, naïve bayes uses Baye's Theorem in order to do the classification task, and it is a probabilistic machine learning model[5]. In general, under the basic definition of these two models, naïve bayes converges more quickly than logistic regression. In this project, Logistic Regression and Naive Bayes perform with similar accuracy on four datasets. However, we found that logistic regression has a better accuracy than naive bayes. For example, in the ionosphere dataset, we get around 91% accuracy for LR but slightly lower for Gaussian Naive Bayes.

II. Datasets



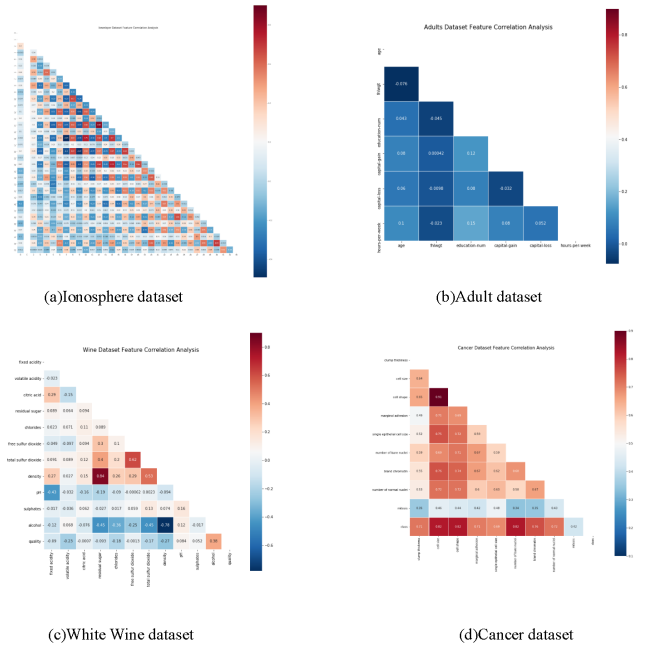
(a) Ionosphere dataset (b) Adult dataset

(c) Cancer dataset (d) White Wine dataset

Fig. 1: Proportion of classes in four datasets

For all dataset, we briefly cleaned and removed any missing values in them, and we also removed duplicates values in Adult dataset in order to have a better result after applying two models on them[6]. For all other features, which look malformed, for example, the second feature in the ionosphere dataset. We ran experiments on both datasets with and without outliers, however whether to keep the outliers or not only has a slightly small influence to our final result, and removing the outliers does not help with improving the accuracy. Additionally, removing any malformed data would exclude quite a number of datas in our dataset, and may lead to inaccurate results. Therefore, we just keep all datas as we respect these datas are also meaningful and keep the datasets inclusive. In order to improve the performance of our models, we applied minmax normalization to our datasets, and it returned with more numerically stable data based on computations with standardization. This step helped us avoid the risk of getting error and losing data.[7]

In order to have a better visualization of out datasets, we introduced pie chart to visualize the distribution of positive and negative classes in four datasets, which can be seen in fig. 1. Additionally, we used correlation analysis[8] and generated correlation graphs for all four datasets, which illustrates how features are distributed and correlate with targeted value, which is shown in fig. 2.



(a)Ionosphere dataset (b)Adult dataset

(c)White Wine dataset (d)Cancer dataset

Fig. 2: Correlation graph four datasets

Ionosphere: This dataset contains 351 usable examples with 100 percent clean data. There are 34 features and 1 class label for each. No additional transformations on the input vectors were used either to take advantage of clumps of positive-class values noticed in some pairwise feature graphs.

Adult: This dataset contains 32561 instances, and 2399 missing value. Thus, it has 30162 usable examples, which also contain 23 duplicates. So we'll use the remaining 30139 examples for our experiments.

Cancer: This dataset contains 699 instances with

Accuracy for four datasets

	Logistic Regression	Gaussian Naive Bayes	Hybrid Naive Bayes
Ionosphere	0.91667	0.83333	0.77778
Adult	0.82609	0.78960	0.22089
Wine	0.71633	0.69436	0.69436
Cancer	0.985507	0.96193	0.96047

TABLE I: For all four datasets, we used 10% of entire dataset as test set to get the Accuracy. We also apply 5-fold cross validation on training set, which is the rest 90% of the entire dataset.

Train-Test Ratio Table for Ionosphere Dataset

	Logistic Regression	Gaussian Naive Bayes	Hybrid Naive Bayes
9:1	0.91428	0.88571	0.82857
8:2	0.91428	0.87142	0.8
7:3	0.847619	0.82857	0.79047
6:4	0.85714	0.85	0.77857
5:5	0.83428	0.80571	0.76
4:6	0.79523	0.81904	0.78571
3:7	0.73469	0.81632	0.76326
2:8	0.67142	0.84642	0.825
1:9	0.65079	0.77777	0.77777

TABLE II: This is the train-test ratio test based on Ionosphere dataset in order for us to compare the accuracy of two models on the dataset with diffrent size of data (by controlling the trainin size)

100 percent clean data, and there are 10 features with 1 class label for each.

White wine: This dataset contains 4898 valid instances in total, each with 11 features and 1 class label for each. We analyzed the distribution of wines according to the quality score. We split the quality score into two classes, which are score above 5 and score below and include 5 respectively. This makes the classification into binary.

III. Results

We used the Ionosphere dataset to test the optimal learning rate and number of gradient iteration. After multiple trials, we figured out 0.001 learning rate can give us a really accurate result, and any learning rate lower than that (e.g. 0.0001) will take us a lot of time to converge. At the same time, any learning rate that is larger than 0.001 (e.g. 10) will give us a really inaccurate result and may cause forever convergence. Therefore, we chose 0.001 with 1000 iterations, for all the following experiments.

First, we compared the accuracy of Logistic Regression, Gaussian Naive Bayes, and Hybrid Naive Bayes model on all four datasets. According to TABLE I, the accuracy of LR on Ionosphere is 91.7%. Then we got 83.3% for Gaussian and 77.8% for Hybrid Naive Bayes. For adult dataset, since it has categorical features, we used one-hot-encoding to let Logistic Regression and Gaussian Naive Bayes able to learn the dataset. We also invented an Hybrid Naive Bayes, which learned adult dataset without using one-hot-encoding. Thus, we got 82.6%, 79.0% and 22.1%, respectively for Logistic Regression, Gaussian Naive Bayes, Hybrid Naive Bayes. Overall, we can see LR perform a slightly better than Naive Bayes on all four datasets.

Second, we compared the performance of logistic regression on different learning rates on the Ionosphere dataset. Logistic regression model's ability to approach ideal weights and thereby higher accuracy is dependent on its learning rate and having enough gradient descent iterations. We found that all models had a certain threshold where learning rates higher than this threshold

led to unstable accuracy over iteration functions. However, we also found that very low learning rates took more time and yielded diminishing returns in quality for this time.

fig. 4 illustrates an older Ionosphere Logistic Regression Model's accuracy over gradient iterations and its associated weights over gradient iterations when a learning rate of 0.001 is applied for 1000 gradient descent iterations. Notice the instability caused by the learning rate manifested in both the accuracy over iterations and weights over iterations.

fig. 5 illustrates the same model's graphs when the learning rate is decreased from 0.001 to 0.0004. Notice how the oscillations are significantly smaller and may give an observer more confidence in concluding that they have reached peak accuracy.

Lower is not always better as demonstrated by fig. 6 and fig. 7: Lower learning rates cause the logistic regression model to take more time to reach the optimal weights - in other words, they progress slower - and do not always guarantee better results. We tested our ionosphere logistic regression model with learning rate at 0.0004 at 250 iterations as well as learning rate at 0.0001 at 1000 iterations and found that the functions of accuracy over time were identical if we took every 4 iterations for learning rate 0.0001 to be the equivalent to 1 iteration for learning rate 0.0004. To conclude, lower learning rate worsen the speed performance of our logistic regression model. However, high learning rates can cause unstable accuracy over gradient iterations trends. It is likely best practice to use learning rates that are sufficiently low to produce good accuracy trends over time, but not so low that they take an unnecessary amount of time to process.

Finally, we compared the accuracy of the two models as a function of the size of the dataset (by controlling the training size), we did it based on Ionosphere datasets. We divided our dataset into training dataset and test dataset with training-test ratio as 90%, 80%, 70%, 60%, 50%, 40%, 30%, 20% and 10% , the result is shown in fig. 3 and TABLE II. We used 0.001 as the learning rate with 1000 iterations. In the graph, it is shown that for LR, it boosts the accuracy from 65% to

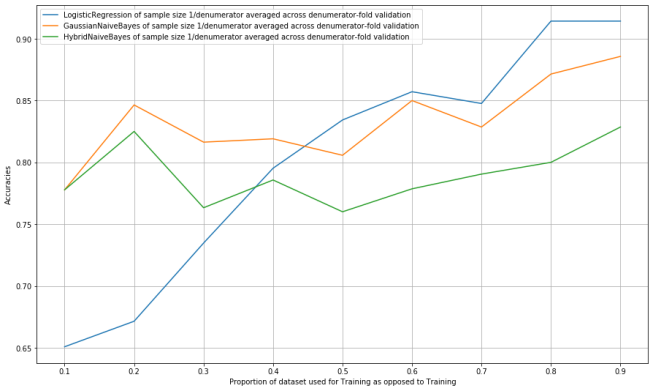


Fig. 3: Train-Test Ratio Graph

91.4% for training-test ratio 1:9 and 9:1 (train:test) respectively. For Gaussian Naive Bayes, it has quite stable accuracy along the whole experiment. Same as the Hybrid Naive Bayes, which is slightly lower than Gaussian Naive Bayes.

IV. Discussion and Conclusion

In this project, we are asked to implemented two models to solve binary classification problems. However, in real-word scenarios, we may inevitably encounter situations where more than 2 classes are present, and we might need to implement other type of models in order to deal with multi class classifications.

One challenge the adults dataset posed was the presence of textual features. Gaussian Naive Bayes classifiers and Logistic Regression classifiers are designed to take in numeric inputs rather than categorical inputs. They are therefore incompatible with many of the adults dataset's features. While we mentioned earlier that we resolved this issue with one-hot encoding, we had actually also attempted to write a Hybrid Naive Bayes classifier that would accept features of all types - whether they be binary, Gaussian, or even string-based categoricals. The implementation of a so-called Hybrid Naive Bayes classifier is possible as the conditional probabilities for each feature are mutually exclusive of other features. Interestingly, our Hybrid Naive Bayes model never actually performed better than our Gaussian Naive Bayes model on the numeric datasets and the one-hot encoded version of the adult dataset. It did, however, give an above-random accuracy of the adult dataset with textual features.

Another challenge in the project was the presence of numeric features that were not of similar scale. Interestingly, logistic regression does not appear to cope well with having features on a different numeric scale. In particular, our cancers dataset's first feature had values hovering around a million while the other features remaining within single and double digits. This threw off the logistic regression's gradient descent as it would adjust the weights of the small-valued features like they were large-valued features. The resulting graphs of accuracy and cost over gradient descent iterations were extremely erratic and did not appear to converge. The solution was, of course, the normalization of our input features such that they all worked on a standard of 0 to 1; and our model's cost now converged downwards. Seeing the success normalization provided us with the cancer dataset, we applied normalization to all other numeric features in other datasets afterwards, which improved the logistic regression's accuracy.

In conclusion, we implemented two classification models, Logistic Regression model and Naive Bayes model, then applied both of them on the four real-world datasets to compare the performance and classification accuracy. We also used pie charts to visualize the distribution of class in each dataset and used the correlation graph to help us to have a better visualization on how features are distributed and correlate with targeted value. We closely studied how the performance of these two models and their classification accuracy are related to normalization, learning rates and gradient descent iterations.

V. Statement of Contributions

Each group member took one dataset and did all the three tasks and all group member contributed to adult dataset. To finalize the result, Suzy combined and made a new verison of preprocessing, Edwin and Eric finalized two models and ran the experiments.

•Edwin: Ionosphere dataset, Adult dataset, writeup
•Eric: Cancer dataset, Adult dataset, writeup
•Suzy: White wine dataset, Adult dataset, writeup

References

[1] K. P.Murphy, "Discriminative training," in *Machine Learning A Probabilistic Perspective*, p. 620, 2012.

[2] T. Jebara, "Generative versus discriminative learning," in *Machine Learning: Discriminative and Generative*, pp. 17–60, 2004.

[3] C. M. Bishop in *Pattern Recognition and Machine Learning*, pp. 21–24, 2006.

[4] S. Walker and D. Duncan, "Estimation of the probability of an event as a function of several independent variables," in *Biometrika*, pp. 167–178, 1967.

[5] M. E. Maron, "Automatic indexing: an experimental inquiry," *Journal of the ACM (JACM)*, vol. 8, no. 3, pp. 404–417, 1961.

[6] H. H. E. Rahm, "Data cleaning: Problems and current approaches," 2000.

[7] J. Grus in *Data Science from Scratch. Sebastopol, CA: O'Reilly*, pp. 99,100, 2015.

[8] F. J. Anscombe, "Graphs in statistical analysis," in *The American Statistician*, pp. 17–21, 1973.

VI. Appendix

Ionosphere Logistic Regression: Learning Rate 0.001 over 1000 gradient descent iterations:

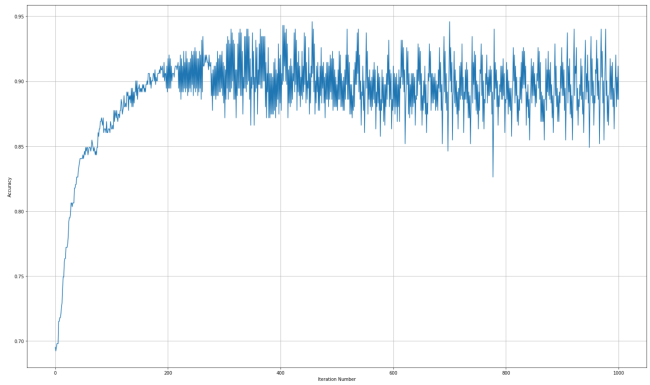


Fig. 4: Graph1

Ionosphere Logistic Regression: Learning Rate 0.0004 over 1000 gradient descent iterations:

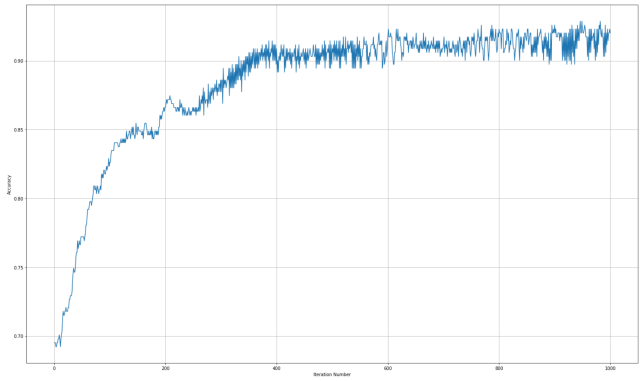


Fig. 5: Graph2

Ionosphere Logistic Regression: Learning Rate 0.0001 over 1000 gradient descent iterations:

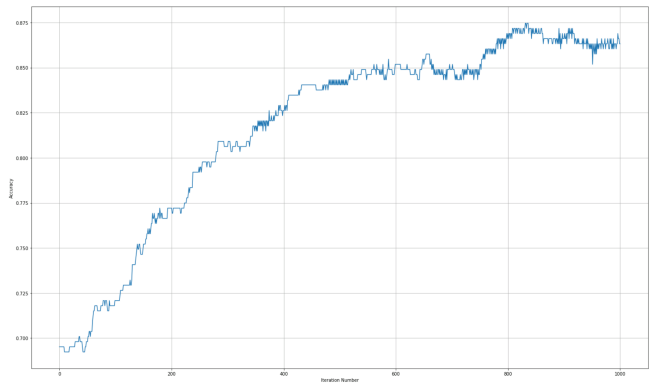


Fig. 6: Graph3

Ionosphere Logistic Regression: Learning Rate 0.0004 over 250 gradient descent iterations, properties used for the 5-fold cross validation:

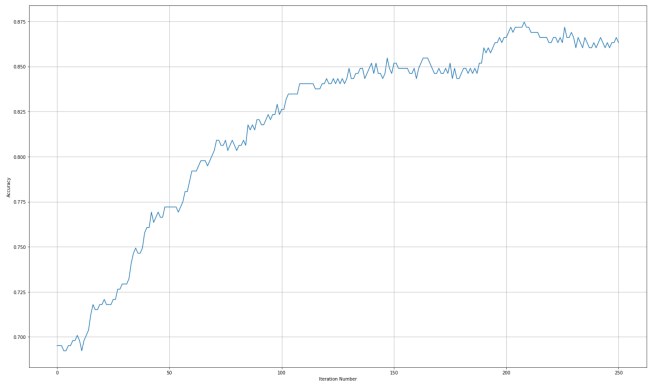


Fig. 7: Graph4