COMP 551 - Project 1 Report

Omar IBRAHIM 260726420
Justin SUN 260707684
Mohamed BOUAOUINA 260765511

***Abstract*—In this project, we investigated the performance of Naive Bayes and Logistic Regression on four benchmark datasets. We used hyperparameter tuning and full-batch gradient descent with 5-fold cross validation to accurately train our models. The highest reported accuracy for LR is 86% with the ionosphere dataset while the lowest is 62% with the adult dataset. For NB, the highest accuracy we achieved was 79% and the lowest is 29% which is similar to the reviewed literature. In terms of training time, we found that Naive Bayes is much faster for all small datasets and performed significantly worse (34% more) for the much larger dataset.**

***Index Terms*—machine learning, classification logistic regression, naive bayes, cross validation, hyperparameter tuning, full batch gradient descent**

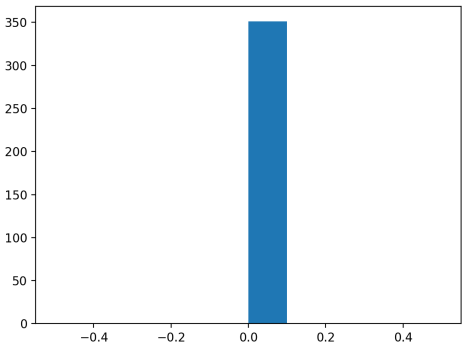


Fig. 1: Feature 1 distribution

I. Introduction

In this project, we investigated the performance of Naive Bayes and Logistic Regression on four benchmark datasets. The datasets vary in size of instances and features which helps us compare both NB and LR for speed, accuracy, and scalability. According to the literature, the standard Naive Bayes does not scale well and requires some changes like implementing the NBTree [1]. The highest reported accuracy for LR is 86% with the ionosphere dataset while the lowest is 62% with the adult dataset. For NB, the highest accuracy we achieved was 79% and the lowest is 29% which is similar to the literature [1]. In terms of training time, NB is much faster for all small datasets and performed significantly worse (34s more) for the much larger dataset. We confirm that LR should be used for larger datasets.

II. Datasets

A. Ionosphere

The Ionosphere data-set has 34 continuous features. We removed rows with missing features. After plotting the frequency distribution for each feature, we found that removing features 0, 1, 13 is best due to their distribution being similar to the figure below.

B. Adult

The adult data-set has 32,561 instances and 14 features. We removed all instances with missing data since the data-set is already large enough, and turned the *sex* column into a binary feature. Next, we normalized continuous features like *fnlwgt* and *hours-per-week* to get rid of overflow errors. The education-num and education features are mapped 1:1 so we removed the education-num features. By the native-country feature plot below, we found that we need to remove countries except US, Germany, Philippines, Mexico.

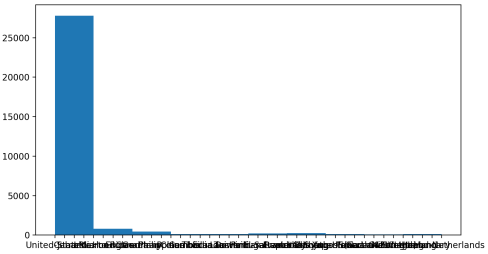


Fig. 2: native-country frequency distribution for 41 countries

We also made scatter plots for pair-wise features similar to the figure below which helped us identify features that can be removed like *capital-gain* and *capital-loss*.

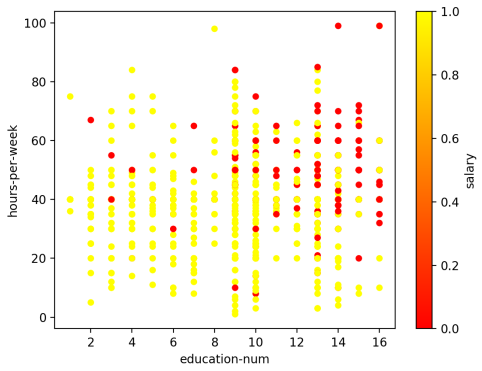


Fig. 3: Education x num-of-hours scatter distribution

C. Mammogram Mass

The mammogram mass data-set has 4 categorical features (BI-RADS, shape, margin, density) and 1 continuous feature (age). By testing both, it seems like label encoding gave us a better accuracy than one-hot encoding. We also implemented a function *fillMissing* which fills NA values with the average of the column.

D. Tic-Tac-Toe

The Tic-tac-toe data-set has 958 instances, each with 9 features, one feature for each cell in a 3x3 tic-tac-toe game board. Each cell can either be empty (b), x (x), or o (o). Testing one hot encoding gave really bad prediction accuracy because of the linear dependency between features. Instead, we did label encoding using

$$f(z) = \begin{cases} 0 & \text{if a = o} \\ 1 & \text{if a = b} \\ 2 & \text{if a = x} \end{cases}$$

III. Results

A. Accuracy

Before assessing the final accuracy of the Logistic Regression model on the data sets, we performed some hyper parameter tuning to determine which combination of learning rate, gradient threshold, and maximum number of GD iterations to use on each data set to get the optimal training and performance. Please refer to our HPTuning.py script in the submitted code folder for our procedure.

dataset	Logistic Regression	Naive Bayes
ionosphere	0.86	0.79
adult	0.62	0.28
mam	0.75	0.79
tictactoe	0.69	0.68

As instructed in task 3, we ran both LR and NB on all 4 datasets using 5-fold cross-validation and took the average of the accuracies obtained on the folds. The results can be seen in the table below which we generated using matplotlib.pyplot's functions from a Pandas DataFrame. Both LR and NB perform similarly for all datasets to the exception of the adult dataset where NB performed worse by 34%. As expected, this is because the other datasets are of similar size (1000 instances) and NB performs well for small datasets. These results were obtained using the CompareAccuracy.py script in the submitted code folder.

B. Learning rate of gradient descent

After having examined the overall accuracy of each model on the data sets, we explored the accuracy of Logistic Regression versus the number of gradient descent iterations it is allowed to take using the GDPlots.py script. Figures 4, 5, and 6 plot the evolution of accuracy on the ionosphere dataset as the GD iterations are increased for learning rates of 0.5, 0.1, and 1 respectively. We notice that, as the learning rate increases, the accuracy converges to its final value faster. This is particularly the case for the learning rate of 1 which converges almost immediately to its final accuracy of approximately 83%. It is interesting to note that 1 is the optimal learning rate returned by the hyper parameter tuning for the ionosphere dataset.

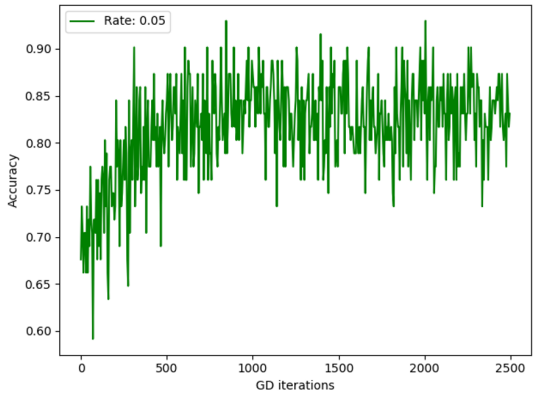


Fig. 4: Plot of accuracy versus gradient descent iterations with learning rate of 0.5

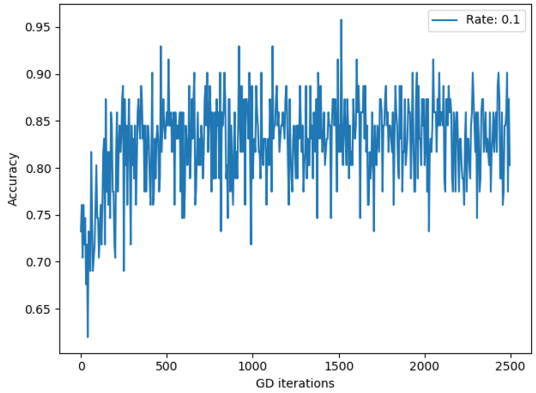


Fig. 5: Plot of accuracy versus gradient descent iterations with learning rate of 0.1

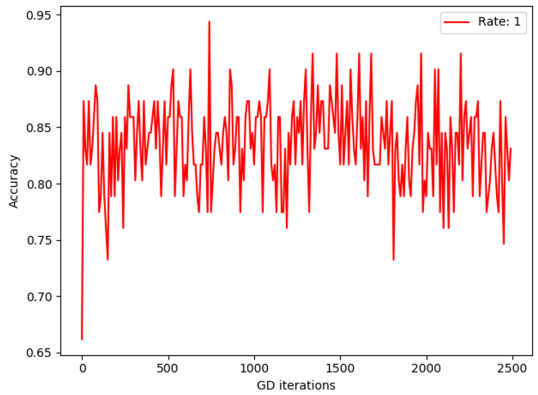


Fig. 6: Plot of accuracy versus gradient descent iterations with learning rate of 1

C. Accuracy as function of training size

Shown below are the plots of accuracy versus size of training data used as input from each dataset. Each of the figures show the accuracies of both the Naive Bayes and Logistic Regression models plotted against the size of training data used as input for the models. Accuracy of naive bayes stays very low across all sizes of training data and is greatly outperformed by logistic regression, as discussed in III-A. In Figures 8 and 9, we how naive bayes can perform better on smaller subsets of the data, as expected.

Fig. 7: Plot of accuracy versus size of training data used as input from the adult dataset. Initial size of training data is 5000 rows with a step size of 5000 rows.

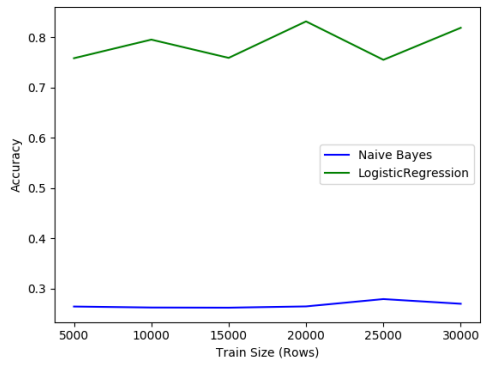


Fig. 8: Plot of accuracy versus size of training data used as input from the ionosphere dataset. Initial size of training data is 20 rows with a step size of 50 rows.

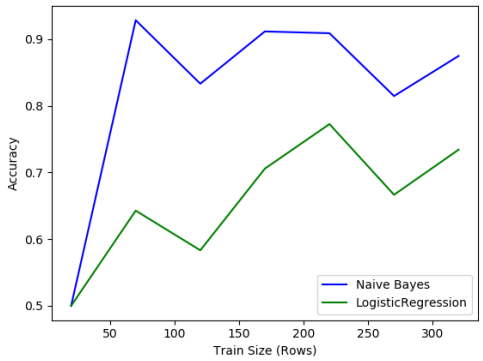


Fig. 9: Plot of accuracy versus size of training data used as input from the mammography dataset. Initial size of training data is 50 rows with a step size of 50 rows.

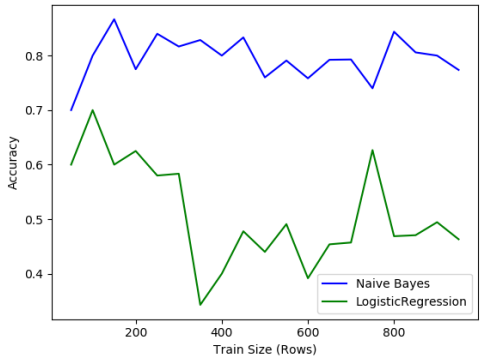
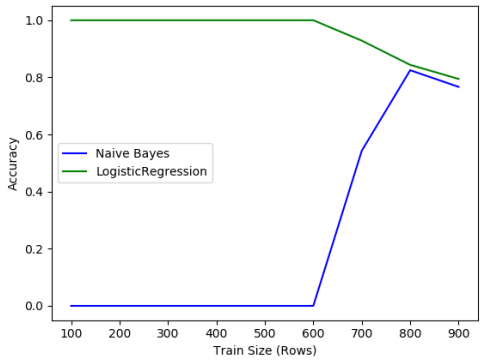


Fig. 10: Plot of accuracy versus size of training data used as input from the tic-tac-toe dataset. Initial size of training data is 100 rows with a step size of 100 rows.



D. Cleanup

We cleaned up all 4 datasets to help us get better accuracy and some of the cleanup steps are described below. We

experimented with *removeMissing* and *fillMissing*. Fill missing looks at each instance with missing data, and for the feature it replaces it with the average value of the feature. It seemed to perform better for mammogram-mass dataset where the accuracy increased **from 45% to 79%** in NB as reported above. This is probably because the feature distribution becomes more Gaussian which is how we setup our NB.

Further, we tested both one hot encoding and label encoding for the tic-tac-toe dataset. Label encoding performed better by increasing the accuracy **from 34% to 68%**. We utilized the fact that label encoding for categorical features gives more importance to some categories more others by assigning 0 to features that are filled by the opponent, 1 to empty cells and 2 to features that have 'x'.

E. Speed

Naive Bayes outperforms Logistic Regression in all three small datasets as seen in figure 11. On average in the 3 smaller size datasets, NB is better by 0.06s in the training phase. For a much larger dataset like Adult, NB does worse by 34.3s which is expected due to the large size.

Fig. 11: Training time for NB and LR in seconds

Dataset	Execution Time (Naive Bayes)	Execution Time (Logistic Regression)
Adult	38.220901188	3.7115035200000013
Ionosphere	0.0025988839999990885	0.028426675999999134
Mammograph	0.005990171999992526	0.17731913099999475
Tic-Tac-Toe	0.006720506000001478	0.0023262830000021495

IV. Conclusion

Our key takeaways include how naive bayes can outperform logistic regression on smaller datasets but fails when datasets get large such as in the adult dataset. In addition, we realized the importance of hyperparameter tuning for Logistic Regression as accuracy and convergence speed both improved when the right hyperparameters (learning rate and stopping conditions) were chosen. Furthermore, we experimented with data pre-processing and found that filling data using the feature average helps with prediction accuracy. We also experimented with plotting scatter plots for each pair-wise feature to analyze relationships between important features which helped us reduce the unnecessary features and therefore get better predictions by avoiding overflow errors. Data normalization was also necessary, specifically for NB because it helped keep the features Gaussian which helped NB approach. Finally, we would like to further investigate adding L1 or L2 regularization to the logistic regression model and determine how the regularization affects the performance of our logistic regression model.

V. Statement of Contributions

Omar:

- Cleaning of data (Clean.py)
- Preprocessing of data (Processor.py)
- Project report

Justin:

- Naive Bayes model implementation (NaiveBayes.py)
- Timing experiment (Timing.py)
- Accuracy vs test size experiment (TestSizePlots.py)
- Project report

Mohamed:

- Logistic Regression implementation (LogisticRegression.py)
- Cross validation and evaluate_acc implementations (CrossValidation.py)
- Hyperparameter tuning implementation (HPTuning.py)
- Accuracy vs gradient descent with different learning rates experiment (GDPlots.py)
- Comparison of accuracies between models experiment (CompareAccuracy.py)
- Project report

References

[1] Kohavi, R. (1996). Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid. [online] Available at: http://aaai.org/Papers/KDD/1996/KDD96-033.pdf [Accessed 11 Feb. 2020].