writeup

Ningping Wang, Yanying Zhang, Cathrine Ao

February 2023

1 Abstract

Machine learning has become crucial in several industries, including energy efficiency and qualitative bankruptcy analysis. In this project, basic analysis was performed, and two machine learning models, linear regression and logistic regression were implemented and evaluated for their performance in understanding and fitting the Energy Efficiency and Qualitative Bankruptcy datasets. In addition, the impact of various training data sizes, mini-batch sizes, and learning rates was analyzed for their effects on models' performance using the cost function, and convergence speed was analyzed using the number of iterations. Overall, it shows that the effect remained close to what is expected for theory on model performance and convergence speed, and each will be further discussed in the report separately. In addition, by incorporating techniques such as normalization, L1 regularization for feature selection on uncorrelated variables, and momentum, the models' accuracy was improved, resulting in accuracy scores above 0.88 returned by different evaluation metrics for both models on both training and test datasets.

2 Introduction

This project aims to implement two machine learning models to analyze two given datasets and train the data with these tools. Techniques such as normalization, to ensure that the regularisation term λ regularises/affects the variable involved in a (somewhat) similar manner; L1 regularization, to forces weak features in both datasets to have zero as coefficients which lead to producing sparse solutions, inherently performing feature selection, and to helps to prevent overfitting using the introduced hyperparameter λ to shift away from the very w and to reduce the complexity of the model and forcing it to generalize better to unseen data.
Momentum was implemented to accelerate the optimization process, e.g. decrease the number of function evaluations required to reach the optima or improve the optimization algorithm's capability, resulting in a better final result in our experiment.

On the other hand, since we balance the minibatch of the model and try to predict the effect, we have the assumption that it could have a severe influence on the training result. Still, it is shown by our test result that the minibatch has the least effect on the model training. That is under the assumption that a small minibatch can cause more noise on the weight. And due to we get a relevantly moderate minibatch, it is shown by our data that minibatch has less effect on the performance of weight.

Finally, the linear regression model implemented for dataset 1 only achieved moderate success with 88% accuracy for Y1 and 86% accuracy for Y2. At the same time, the logistic regression for dataset 2 classification was found to be implemented with great success with an accuracy of 0.96 to 1.0 calculated using different evaluation matrices such as R2 score and confusion matrix. Based on the values in this matrix, we can say that the model is doing a good job of correctly classifying samples, with a high number of true positive and true negative classifications.

3 Datasets

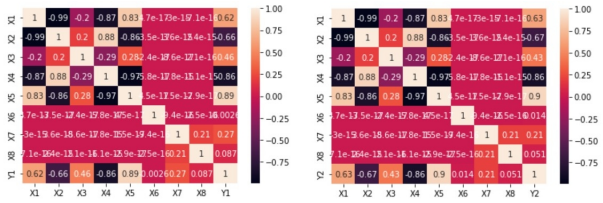
In this project, the two datasets consisted of numerical and categorical data separately. Where for the categorical dataset 2, we have performed a label transformation for its value to facilitate analysis. First, we clean up the data set by removing all the missing values through the command in Pandas data frames. After the operation, the result turns out to be none of the entries exist, missing values or blank entries, indicating no missing values for both datasets.

3.1 Analysis on dataset 1

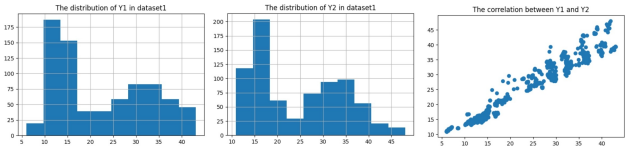
The first dataset denotes the energy efficiency regarding relative compactness, surface area, wall area, roof area, overall height, orientation, glazing area distribution, heating load, and cooling load. The predictor Xs are shown to correspondingly have a relative gamma, beta, burr and lognormal distribution. After applying data preprocessing and analysis of distribution fitter, we also came up with the descriptive statistics table below that stands for a quick understanding of the distribution of the data, checking for missing values, and identifying outliers of dataset 1 through the python command ".describe()." It shows that X2 has the highest standard deviation and mean, which can be concluded that it causes the most bias and variance in the model. While X7 has the least std, it contributes the least to the bias and variances of the model.

	X1	X2	X3	X4	X5	X6	X7	X8	Y1	Y2
count	768.000000	768.000000	768.000000	768.000000	768.00000	768.000000	768.000000	768.00000	768.000000	768.000000
mean	0.764167	671.708333	318.500000	176.604167	5.25000	3.500000	0.234375	2.81250	22.307195	24.587760
std	0.105777	88.086116	43.626481	45.165950	1.75114	1.118763	0.133221	1.55096	10.090204	9.513306
min	0.620000	514.500000	245.000000	110.250000	3.50000	2.000000	0.000000	0.00000	6.010000	10.900000
25%	0.682500	606.375000	294.000000	140.875000	3.50000	2.750000	0.100000	1.75000	12.992500	15.620000
50%	0.750000	673.750000	318.500000	183.750000	5.25000	3.500000	0.250000	3.00000	18.950000	22.080000
75%	0.830000	741.125000	343.000000	220.500000	7.00000	4.250000	0.400000	4.00000	31.667500	33.132500
max	0.980000	808.500000	416.500000	220.500000	7.00000	5.000000	0.400000	5.00000	43.100000	48.030000

Besides, the correlation plot below computes the pairwise correlation of columns X1-6 with Y1 and Y2, excluding NA/null values.

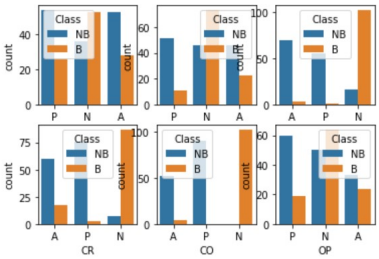


From the above result, we can conclude that the correlation coefficients between predictor X1-6 and target variables Y1 and Y2 share a similar pattern. X1 and X2 have a -0.99, and X4 and X5 share a -0.975 correlation coefficient value, indicating that the two pairs of variables are almost perfectly negatively correlated.
For the three plots we have below, the distribution of Y1 and the distribution of Y2 and the correlation between Y1 and Y2 show that Y1 and Y2 show a similar distribution pattern. Therefore, we could have to go with only analyzing one Y; however, to be safe, we would fit a linear regression model for Y1 and Y2.



3.2 Performance on dataset 2

Qualitative Bankruptcy Dataset
We count the uniqueness and frequency of the given data. To analyze which factor could cause a potential bankruptcy, we build diagrams to illustrate the distribution between industrial risk, management risk, financial flexibility, credibility, competitiveness, and operating risk towards the counts of Non-bankruptcy and bankruptcy. By interpreting the diagrams, we could conclude that all negative scores of risks and flexibility and competitiveness have a higher number/potential of bankruptcy.



Besides, the correlation plot below computes the pairwise correlation of the predictor variables, with the class variable being non-bankruptcy and bankruptcy, excluding NA/null values. We can see the correlation between different variables is between -0.8 to 0.8, and low correlation coefficient score variables will be dealt with later in the model fitting process with other methods; for now, we will take all the variables for developing the logistic model.

	0	1	2	3	4	5	6
0	1.000000	0.410763	0.070896	0.055691	0.197276	0.208684	-0.211682
1	0.410763	1.000000	0.192925	0.095471	0.156584	0.134761	-0.274010
2	0.070896	0.192925	1.000000	0.496241	0.657387	0.033582	-0.740179
3	0.055691	0.095471	0.496241	1.000000	0.518493	0.122844	-0.615203
4	0.197276	0.156584	0.657387	0.518493	1.000000	0.203098	-0.822841
5	0.208684	0.134761	0.033582	0.122844	0.203098	1.000000	-0.158676
6	-0.211682	-0.274010	-0.740179	-0.615203	-0.822841	-0.158676	1.000000

4 Part 3 Result Analysis

4.1 Extra methods' performance on linear and logistic model fitting

For the linear regression model using 80/20 train/test split, we have tried out four different methods w/o regularization and w/o momentum in our model training process. Here are the corresponding R2 Score performance results for the test set shown below:

Linear Regression		
	with momentum	without momentum
with L1 regularization	Y1: 0.90, Y2: 0.874	Y1: 0.90, Y2: 0.874
without L1 regularizatic	Y1: 0.906, Y2: 0.886	Y1: 0.906, Y2: 0.886

Logistic Regression		
	with momentum	without momentum
with L1 regularization	1	1
without L1 regularizatic	0.96	1

Based on the results above, we can conclude that the R2 score for no L1 regularization and with or without momentum is the two most successful predictive models on the test set of dataset 1 for both target variables Y1 and Y2. However, for logistic regression, we see that with momentum and regularization, without momentum and without regularization, and without momentum and with regularization, all give a better performance. The small size of the given data may cause this. Since we are given a simple and small data set with low variance and low bias, it is obvious to see that regularization and momentum result in less effect on the distribution.

4.2 linear regression and fully batched logistic regression on train/test set

Processing followed with momentum and regularization implementation performance analysis on the two models, we choose to build our models with momentum and no regularization for both models for the simplicity of the following analysis of the performance on training and test set:

For linear regression, for the test set, the performance on Y1 and Y2 are 0.906, 0.886 correspondingly; for the training set, the performance on Y1 and Y2 are 0.900, 0.871.

For logistic regression, the R2 score for the test set is 0.96; for the training set, the R2 score is 0.955.

In conclusion, we didn't see an overfitting/underfitting pattern appearing in our model performance which is ideal. Note: The performance analysis of regularization and momentum will be discussed further in part 3.

4.3 weights

1. Linear regression model weights:
Weights to Y1's linear regression model: [10.0381037,-1.94940664,-1.30382354,1.97482775, -3.9463163, 4.88263307, -0.06930763,2.72155888, 0.31874016] shows that w3,w5,w7,w8 are all positive weights, indicating that as the corresponding feature value x_i increases, the target variable $\hat{y}$ value increases; whereas, w1,w2,w4,w6 are all negative weights, indicating that as the corresponding feature value x_i increases, the target variable $\hat{y}$ value decreases. The intercept term, w0 = 10.0381037 is added to the linear regression as the predicted value of the target variable when all the features are equal to 0. In addition, with w5 being the highest magnitude in absolute term among the weights, it is interpreted as having the strongest association and impact between the feature X5 and the target variable $\hat{Y_2}$. The same analysis would be applied to the linear regression model fitted for Y2.

2. Logistic regression model weights:
Weights to the logistic regression model: [-1.4201842,-1.12614198,-0.92646682,-3.15611488,-2.31959987,-3.09993541,-0.46954195]; with all weights being negative, indicating that an increase in the features is associated with a

decrease in the probability of the positive class. In addition, the magnitude of variable 3 and 5 have the highest weight magnitude in absolute term, which provide information about having the strongest strength of the relationship between the feature and the target class; an increase in both features' value is associated with a decrease in the probability of the positive class. All negative weights somehow explain the result from the basic correlation analysis section; all features have negative correlation coefficient values with the Class variable. And having a higher risk and credibility score will result in a lower probability of being in bankruptcy.

4.4 growing training dataset

We mimic a growing training data subset through the given context to analyze the model's accuracy rate. It is shown on the graph for both linear models for Y1 and Y2 below and for logistic regression fitted for dataset 2, as the training set grows by 0.1 percent each time, the accuracy first increases. It is correct that in general, increasing the training data set size can lead to improved model performance on the test set. More training data provides the model with more information to learn from, allowing it to make more accurate predictions; this is exactly what we observed for our training set. However, test set accuracy reaches a peak and shrinks as the training set size grows larger, being below the performance of the training set, which is referred to as overfitting. It occurs when a model becomes too complex and starts fitting to the noise and fluctuations in the training data rather than the underlying patterns that are generalized on the test set. Which is, in theory, a correct pattern that we should observe as the training dataset grow.
.

4.5 minibatch size

Effect performance in linear regression and logistic regression: Given the results, we conclude that as the batch size increases, the R2 score increases in our case. This could be explained by an improvement in the generalization ability of the model and the capability of learning more robust and generalizable features. However, as the batch size increases larger and larger approaching batch size of 200 and 600 (fully batched: entire training dataset used in each iteration), it can also reduce the regularization effect and dampen the noise in the gradient updates, which lead to overfitting by increasing the complexity of the model and lower the accuracy on the test set as we observed in the plot with a decrease in performance analysis for our logistic regression. As a conclusion, batch size of 128 works the best for both regressions.

Effect on Convergence Speed: As observed in our linear and logistic models, an increasing batch size shows a faster convergence speed because it allows the model to make more updates per iteration and reduce stochasticity during the optimization process. However, it also increases the memory and computational requirements of the linear model. Therefore, as the batch size increases the larger and larger approaches batch size of 600 (fully batched: entire training dataset used in each iteration), the convergence speed decrease, as well as makes more updates per iteration, which leads to overfitting and higher variance in the gradients, which can slow down the convergence speed eventually.

4.6 various learning rate

Performance analysis: As observed for linear regression models, a small 0.001 learning rate results in the lowest performance, Y1 with 0.58 and Y2 with 0.3, categorized by underfitting, also due to the small max number of iterations that we set for the gradient update. The same applies to logistic regression; the lower the learning rate, the smaller the accuracy score. Therefore, it is not an ideal learning rate for our linear regression model. With a larger learning rate of 0.8 and 2, we did notice an outstanding improvement in the model's performance which is 0.87+ for Y1 and 0.77+ for Y2. And a separate analysis for a learning rate equal to 100, we get NaN as a result because a learning rate that is too high can cause the model to oscillate wildly and increase the values of the parameters to very large or even infinite.
We have also investigated the trend of the increase of learning rate effects on the regression model's loss:
As observed, a higher learning rate (0.8,2 green and orange lines in orange) effects in a dramatic decrease in the loss updates along gradient descent iterations updates, as the model are taking larger steps towards the minimum of the loss function and results, whereas a small learning rate of 0.001, line blue, results in a steady

decrease in the loss along iterations due to that the model's weights are updated more gradually.

Speed analysis:
As observed from the pattern plots on learning rate effects on a linear regression model and logistic regression, the higher the learning rate, the lower the number of iterations required for the model to converge due to that higher learning rate causes the model's weights to be updated more drastically with each iteration. However, if the learning rate is too high, the model can overshoot the optimal weights and oscillate or even diverge instead of converge.

Conclusion: We would have to use a learning rate that is small enough in to allow the model to converge to a good solution but not so small that convergence takes a long time in future modelling and is not too big, thus having some risk of overshooting the optimal solution and oscillating or diverging.

4.7 regularization strength

The regularization term is added to the loss function to penalize large coefficients and encourage small, simpler models; as it gets larger, it can lead to better generalization performance as the model will be less likely to overfit the training data. Therefore, we observed a comparably high accuracy with a small regularization weight; however, with the regularization weight gets too large, the model may underfoot the data, leading to a model that is too simple to fit the data effectively, therefore, resulting in a poor fit to both the training and test data. Therefore, we observed a decrease in the R2 score in our model as the weight increased. As the regularization strength increases, more and more features will be removed from the model. If important features are removed, the model's ability to make accurate predictions will decrease, resulting in a lower accuracy score

4.8 Decision boundaries

For the linear regression, the decision boundary is a straight line. We found that the outliers in the dataset can heavily influence its gradient, which results in overfitting. Therefore, we tried to implement L1 regularization to reduce the absolute values of the weights, thereby reducing the magnitude of the weights and making the model more interpretable. In addition, the decision boundary for the logistic regression is multivariate. It considers all five features to separate the two classes, each feature having a different weight and sign.
For the logistic regression, its decision boundary is a non-linear surface because its activation function (sigmoid) used for mapping the actual inputs to binary outputs is non-linear. In addition, the decision boundary for the logistic regression is multivariate. It considers all five features to separate the two classes, each feature having a different weight and sign.

5 Discussion and conclusion

As a conclusion, we could conclude that the more the training data was, the higher accuracy of the predicted testing results it turns out to be. For regularization and momentum, we could conclude that when the dataset was small, the model may not have enough information to overfit, the added penalty term may not have a significant impact on the model's performance, and the model has low variance and bias, the gradient information may not be robust enough to support the use of momentum. Thus, momentum and regularization have less to do with the result. As a future investigation, with the observed two and more identified predictor variables in dataset 1 to be highly correlated, we suggest implementing an L2 regularization to combat the multicollinearity by constricting the coefficient and by keeping all the variables and penalizing the insignificant predictors instead of implementing L1 regularization for feature selection modelling case that shrinks the coefficient of the less important variables to zero, thus removing some of these features altogether. Finally, we believe we can improve our assignment from two viewpoints in terms of feature enhancement. To prevent overfitting and improve the performance of the linear regression model, we may first choose only the most essential characteristics and eliminate the redundant features for linear regression. Then, we can attempt to map the distribution of the data and predictions and the decision boundaries to gain a deeper understanding of the algorithms.

6 Statement of Contribution

We split the task among three people, and each of us takes one of the parts correspondingly. Part1- Ningping, Part2 - Cathrine, Part3 analysis and report - Ningping, report - Yanying and Ningping. Yanying and Ningping also contributed to part 1 fitter distribution plot and debugging.

7 reference

Ando Saabas, (2014), Selecting good features – Part II: linear models and regularization,https://blog.datadive.net/selecting-good-features-part-ii-linear-models-and-regularization/: :text=L1%20regularization%20%2F%20Lassotext=Since%20each%2
Jason Brownlee,(2021),Gradient Descent With Momentum from Scratch,https://machinelearningmastery.com/gradient-descent-with-momentum-from-scratch/: :text=Momentum%20is%20an%20extension%20to,spots%20of%20the%20search%20
Deven Co,(2018), Ridge Regression and Multicollinearity: An In-Depth Review,/https://www.sas.com/content/dam/SAS/su
global-forum-proceedings/2018/2825-2018.pdf