Abstract

In this project, we aim to implement linear and logistic regression for 2 UCI datasets including the Energy Efficiency dataset and the Qualitative Bankruptcy dataset. The first data set aims to predict both heating and cooling load with linear regression and the second dataset focuses on predicting bankruptcy based on different categorical factors. For the first data set the data cleaning and feature selection process has been implemented and the linear regression with a closed-form solution was designed to predict two predictive variables with the R-squared score of 0.89 and 0.85 for Heating and Cooling load respectively. Moreover, the mini-batch stochastic gradient descent with different sizes of batch (8,16,32,64,128) with different learning rates (0.01,0.001,0.0001) has been investigated and the results showed that the minibatch of 32 with a learning rate of 0.01 had the closest weights to the closed-form solution and was the best in terms of speed of convergence and the MSE[1]. The second dataset was used to predict bankruptcy and the 6 explanatory variables were categorical. After preprocessing the data, the logistic regression with GD[2] and mini-batch stochastic GD has been implemented with different sizes of batches (8,16,32,64,128) with different learning rates (0.001,0.01,0.05,0.1). According to the confusion matrix, the accuracy of the GD logistic regression is 0.98 and the best result of the Stochastic Gradient Descent (SGD) based on the confusion matrix measurement is a learning rate of 0.05 with a mini-batch size of 16.

1. Introduction

This study investigates the implementation of linear and logistic regression on two data sets with two numerical and one binary variable, respectively. The **first dataset** entitled "Energy Efficiency" has 8 explanatory variables and two dependent variables to be predicted, cooling and heating load. In this project, the aim is to compare the linear regression's closed-form solution with the stochastic mini-batch gradient descent linear regression and compare the performance and weights to reach the accurate prediction of the response variable, regarding different batch sizes and learning rates. The data set has been introduced by Tsanas in 2012 [1], to compare a classical linear regression approach with a nonlinear non-parametric method, random forests, to estimate Heating and cooling load. In 2022, Senarathne [2] employed Bayesian Network to recognize the most important factors affecting HL (heating load) and CL (cooling load). They used the Bayesian network structure. They found that a decrease in building height results in high energy efficiency. Using the naïve Bayes classifier, Prasetiyo [3] found the top four features affecting heating and cooling load which are overall height, relative compactness, wall area, and glazing area. In this study correlation and threshold analysis with consideration of univariate k-best algorithms are implemented to find the best features; both a closed-form and mini-batch gradient descent linear regression are utilized and compared to predict HL and CL and find the optimized weights.

The **second dataset** entitled "Qualitative Bankruptcy" has 6 features with a binary target to predict if the company goes bankrupt or nonbankrupt based on those six risk factors [4]. We would like to use logistic regression with gradient descent in the different learning rates, Mini-batch stochastic logistic regression. Aruldoss 2014 demonstrated that in Bankruptcy the most proposed method is neural networks. On the same dataset, they illustrate the relationship between risk factors, and they evaluate the model by measuring the F-test [5]. On the same dataset, Uthayakumar 2020 proposed a Swarm intelligence that outperforms machine learning classifiers namely Logistic Regression, Multilayer Perceptron, and Random Forest in terms of various performance analysis factors [6].

2. Datasets and Preprocessing

2-1 First dataset: Energy Efficiency Dataset.

Energy efficiency analysis was conducted on 12 different building shapes. The buildings differ with respect to the overall height, the glazing area, among other things. This dataset houses a total of 768 observations

[1] Mean Squared Error
[2] Gradient Descent

and 8 features Relative Compactness (RC), Surface Area (SA), Wall Area (WA), Roof Area (RA), Overall Height (OH), Orientation (O), Glazing Area (GA), Glazing Area Distribution (GAD), with the goal of predicting two numerical response variables (Heating Load, Cooling Load).

2-1-1 Explanatory Data Analysis: We found that this dataset has only integer and float variables and does not hold any missing values. The pair plot and heatmap revealed a negative linear relationship between RC, and SA, OH has a high correlation with SA, RC, and RA, and a high correlation with target variables. O and GAD has the poorest linear correlations. We notice the highest negative correlation for target variables is with RA. There is also a high correlation (0.99) between target variables hence in further experiments, we decided in some cases to provide details for one target variable. Moreover, we found that the features have an overall uniform distribution with a combination of normal distribution for the target variables. We concluded that there are no outliers since there is no significant difference between the 75th quantile and the maximum value. We noticed also that the features are measured on different scales hence they do not contribute equally to the model fitting. To address this potential issue, feature-wise normalization, such as MinMax Scaling, is typically used prior to model fitting.

2-1-2 Feature Selection: We used the Select K-best function provided by Sklearn which chooses the best features that contribute most to the target variables according to a score. Obviously, it dropped O and GAD since they have the lowest correlation. Since there is a high correlation between different features, we opted to drop SA and RA since they have respectively high correlation with RC and OH.

2-2- Second dataset: Qualitative Bankruptcy.

This dataset contains 6 features with 250 observations. For predicting bankruptcy six categorized variables are presented including: Industry risk, Management risk, Financial flexibility, Credibility, Competitiveness, and Operating risk. This dataset has 3 attributes “N: Negative, P: Positive, A: Average” these belong to our features. The target class is binary which is “B: Bankrupt, NB: Nonbankrupt”. This dataset is categorical type the first we should tackle this issue, by converting these variables to categorical ones. In order to change these variables methods have been suggested 1) Lableencoder and 2) GetDummy, both methods follow a similar procedure by producing an extra column for each distinguished value.

3. Results

3-1- Report the performance of linear regression for an 80/20 train/test split.

Table 1 and Table 2 illustrate four metrics used to report the performance of the Test/Train set of the closed-form solution for linear regression. Certainly, for the training dataset, the error is lower than the test dataset, and the accuracy which is the R^2 score of the training set is higher for both targets than the test set.

Table 1: Closed-form solution Linear Regression Performance metrics for the test set.

Target Variables	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error	R2 score
Heating Load	2.11	10.06	3.17	0.894
Cooling Load	2.62	12.77	3.57	0.854

Table 2: Closed-form solution Linear Regression Performance metrics for the train set.

Target Variables	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error	R2 score
Heating Load	2.13	8.64	2.93	0.91
Cooling Load	2.20	9.9	3.14	0.89

3-2- Report the performance of logistic regression for an 80/20 train/test split.

In the first step, the logistic regression with a gradient descent was developed with various learning rates and the results were compared. The model was implemented with 4 learning rates (0.001, 0.01, 0.05, 0.1) to evaluate the performance of the model we would confusion matrix. After applying various learning rates

to our model and comparing results for lost function, F1 score, and accuracy we came to release that the best model is logistic regression with a learning rate of 0.05. The confusion matrix is provided in Table 3.

Table 3: Fully batch-sized logistic regression

	precision	Recall	F1 score	Support
Nonbankrupt "0"	0.97	1.00	0.98	31
Bankrupt "1"	1.00	0.95	0.97	19
Logistic regression with a learning rate of 0.05		**The accuracy** for the test set is 0.98 Log loss function 0.0173, AUC-ROC = 0.998		

3-3- Report of the weights of features and discussion on how each feature could affect the models.

3-3-1- Linear regression: As can be seen from the reported weights in Table 4 as the batch size increase from 8 the weights are getting closer to the based GD and the accuracy improves. The closest weights are seen in the batch size of 32. In terms of the effects of each variable on a predictive variable, all of the variables have positive coefficients meaning that by increasing the variables, our outputs would increase as well. Moreover, WA, and OH has the most impact on energy efficiency in terms of both HL and CL.

Table 4: Reported weight for the linear regression with the learning rate of 0.01.

Features	Weights closed-form regression	Weights of the Mini batch SGD				
		batch size 8	batch size 16	batch size 32	batch size 64	batch size 128
	Heating load					
Intercept	-6.20301964	-5.83246341	-5.73097944	-5.84767725	-5.4788360	-2.50679804e+88
RC	5.49256187	5.71357475	5.80704124	5.76358578	6.1095056	-2.36564199e+88
WA	20.28818004	19.89097703	19.82782611	19.86181009	19.4511123	-3.17612550e+88
OH	8.21551564	8.36288953	8.32690289	8.29268768	8.35537372	-3.69216516e+888
GA	7.56440445	7.42097639	7.36922067	7.3920628	7.14120547	-5.35158710e+88
	Cooling Load					
Intercept	-8.04492209	-7.5893332	-7.51482213	-7.58107817	-7.23296989	-2.75256431e+88
RC	2.80034256	3.1213951	3.20548614	3.16559115	3.52133359	-2.59756933e+88
WA	20.56271189	20.13912669	20.10033308	20.10918059	19.65346952	-3.48751258e+88
OH	6.10406754	6.24092876	6.220345	6.17788047	6.14525129	-4.05414472e+88
GA	12.81922164	12.63763605	12.60702526	12.62281179	12.45826479	-5.87625625e+88

3-3-2 Logistic regression: The weight for logistic regression with gradient descent is as follows: As we can see from Table 5 the highest weight belongs to financial flexibility and credibility. It is important to keep in mind Due to the fact that the outcome in logistic regression is a probability between 0 and 1, the weights in logistic regression are interpreted differently from the weights in linear regression. There is no longer a linear relationship between the weights and the probability.

Table 5: Weights for logistic regression.

	Average	**Negative**	**Positive**
Industry risk	-0.84170746	0.21051922	-0.58760995
Management risk	-1.32320374	0.90209938	-0.79769384
Financial flexibility	-2.90882551	4.08927091	-2.39924359
Credibility	-1.20799273	-3.33568332	-1.85412624
Competitiveness	5.53040944	-4.89508139	0.09243043
Operating risk	0.31509346	-1.62632208	-1.21879819

3-4- Effect of training data size (20% to 80%) with models' performance.

3-4-1 Logistic Regression: We apply a learning curve for the best learning rate "0.05" and afterward, we want to test what would be the result if we change the size of the training and test set. Thus, we utilized the learning curve for various dataset sizes. As from the learning curve can be inferred that the test set of 30% of the data size would have the lowest test set error. The results are depicted in Figure 1.

3-4-2- Linear Regression: The learning curve for linear regression shows that with a low training size, the mean squared error is high for training and testing sets which means with a low dataset size we could be in

an underfitting situation, it has even exceeded the test error for a training set size of 20%. This behavior could be explained by the poor generalization of the model since the model has few samples to learn from. With the increase in the training set size, the training error and test error return to normal behavior. For 60% training size, we observe a very low training error in comparison with a high test error. In this situation, the model is experiencing overfitting. From this learning curve, we conclude that the best training size is 80% as it reveals the best solution for the test error tradeoff.

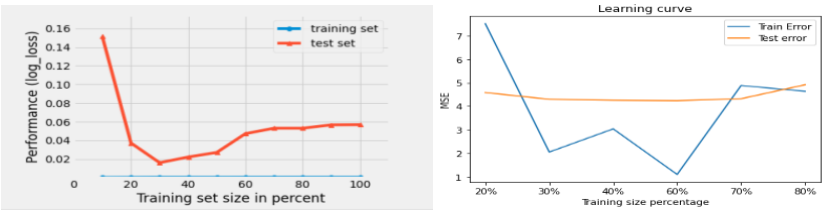

Figure 1: Learning curve for the test and the train in logistic regression (left side), and linear regression (right figure).

3-5- Convergence speed and performance of different batch sizes to the fully batched baseline.

3-5-1 Linear regression: The below plots highlight the learning curves for three different batch sizes with convenient learning rates. For batch sizes 8 and 16 with lower learning rates the cost function starts from

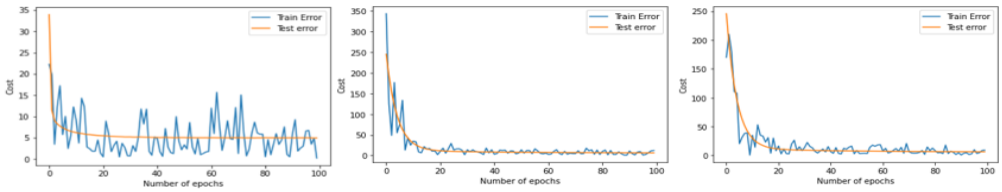

Figure 2: Learning curves for batch sizes 32, 16, and 8 with learning rates 0.01, 0.001, and 0.0001 respectively.

A high-point decreases significantly with the number of epochs until it reaches respectively 12.71 and 9.96, for mini-batch size 32 with learning rate 1e-2, it reaches the lowest result 3.02. Since batch sizes 16 and 8 have comparable and close performance, we opted to make a plot for the performance of both batch sizes 8 and 32 to compare the speed of convergence. In Figure 3, We observe that with batch size 8, the convergence is achieved after more than 20 epochs whereas for batch size 32 with a learning rate of 0.01, the convergence is achieved after 2 epochs. We conclude that batch size 32 with a learning rate of 0.01 has the best performance.

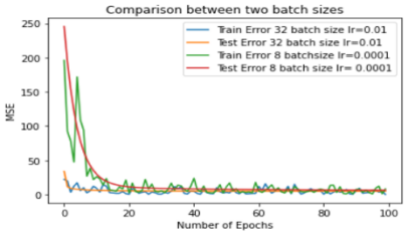

Figure 3: Learning Curves for batch sizes 32 and 8 and learning rate of 0.01 and 0.001.

3-5-2 Logistic Regression: As we increase the size of the batch Using it would degrade the quality of the model as measured by its generalization ability. We run the best logistic model (learning rate = 0.05) with different batch sizes as the number of batch sizes grows the Test Error will grow too. The model with low training error and low-test error would be 16 mini-batch size. It is demonstrated that too large of a batch size will lead to poor generalization. As can be inferred from Figure 4, if we decrease the number of batch sizes the training error and noise would increase. On the other hand, the test error would be decreased. The balance between the two extremes would be mini-batch 16 with a learning rate of 0.05.

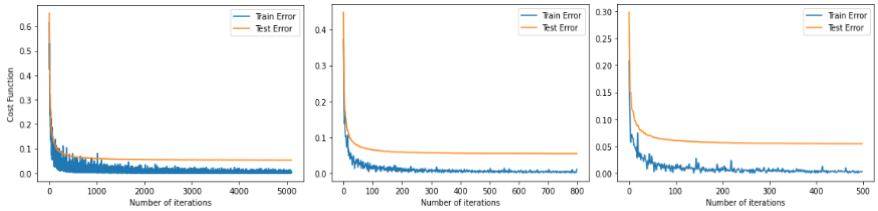

Figure 4: Mini-batch size of 4, 16, and 64 for stochastic logistic regression.

3-5-3 Linear and logistic regression with different learning rates.

For linear Regression, we used 4 different learning rates (0.1,0.01,0.001,0.0001). Since the two target variables are highly correlated, we reported the results for only one target which is HL. With a learning rate of 0.1, the training curve diverges. This divergence is explained by the high value of the learning rate since the weights update using a mini-batch SGD could not be conducted because, from the first step, the weights become far from the optimal value. For that, we decided to report the performance for the three other learning rates for batch sizes 16,32, and 64. The best results reported for the 3 batch sizes are with a learning rate of 0.01in terms of speed of convergence, whereas for a learning rate of 0.0001, the learning curve converges after around 200 iterations for batch size 64, 300 for batch size 32, and 500 for batch size 16. For a learning rate of 0.001, the performance is slightly different from the performance with a learning rate of 0.01. In Figure 5, we notice with the increase in the batch size, the noise in the training error decreases.

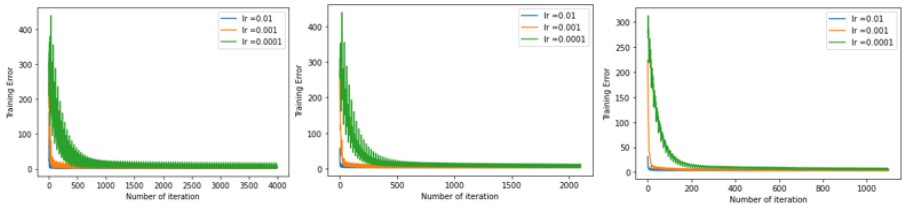

Figure 5: Training curves for batch sizes 16, 32, and 64 respectively with three learning rates.

3-6- Compare analytical linear regression solution with mini-batch SGD regression.

The weights for the two targets are not far from the closed-form solution weights for mini-batch sizes 8, 16, 32, and 64 for the same learning rate which is 0.01. The closest one is batch size 32. The table above in question 3.2 reporting the weights highlights how close the mini-batch gradient with learning 0.01 is to the analytical solution for linear regression. However, for batch size 128, the weights diverge and explode. We could say that it is because a high batch size would prevent the model from learning.

4. Discussion and Conclusion

By analyzing the **first dataset**, feature selection has played an important role in finding the best weights for the mini-batch gradient descent model. This strategy made our mini-batch gradient descent closer to the optimal solution. With the increase of the batch size, the noise in the training curve becomes lower because the batch size becomes closer to the full batch. We also noticed that the learning rate decreases with the batch size. We used different learning to evaluate the performance for Linear Regression with Mini batch gradient descent and we found that the learning 0.01 is the one that gave the closest optimal weight to the closed-form solution as well as revealed the fastest convergence and the best performance in terms of Mean Squared Error. Furthermore, we conclude that the balance between training and testing error is achieved for 80% of the training set size. In the **second dataset**, in logistic regression, we saw the gradient descent is slower compared to stochastic mini-batch-sized. SGD solves the main problem while in GD smaller size (batch) significantly increases the computation cost compared to GD on whole training data. Also, there is a balance between training size and test set which could reduce the test error in our case is 70/30. We conclude that the best model is gradient descent with a learning rate of 0.05 and an accuracy of 0.98.

5. Statement of Contributions

We divided the responsibilities equally in terms of coding and writing the report.
https://colab.research.google.com/drive/1UuNCEnE5dajj9U4AI76x_GYNphLVLul5?usp=sharing

6. Reference

1. Tsanas, A., & Xifara, A. (2012). Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and buildings*, *49*, 560-567.
2. Senarathne, L. R., Nanda, G., & Sundararajan, R. (2022). Influence of building parameters on energy efficiency levels: a Bayesian network study. *Advances in Building Energy Research*, *16*(6), 780-805.
3. Prasetiyo, B., & Muslim, M. A. (2019, October). Analysis of building energy efficiency dataset using naive bayes classification classifier. In *Journal of Physics: Conference Series* (Vol. 1321, No. 3, p. 032016). IOP Publishing.
4. Kim, Myoung-Jong and Ingoo Han. "The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms." *Expert Syst. Appl.* 25 (2003): 637-646.
5. Aruldoss, Martin & Lakshmi, T. & Venkatesan, V. (2014). An Analysis on Qualitative Bankruptcy Prediction Rules using Ant-Miner. International Journal of Intelligent Systems and Applications. 36-44. 10.5815/ijisa.2014.01.05.
6. J. Uthayakumar, T. Vengattaraman, P. Dhavachelvan, Swarm intelligence based classification rule induction (CRI) framework for qualitative and quantitative approach: An application of bankruptcy prediction and credit risk analysis,Journal of King Saud University - Computer and Information Sciences, Volume 32, Issue 6, 2020, Pages 647-657, https://doi.org/10.1016/j.jksuci.2017.10.007.