Mini Project 1: Getting Started with Machine Learning

Amani Jammoul 260381641 Annabelle Dion 260928845 Ethan Kreuzer 261050944

February 16, 2023

Abstract

The task for this project was to implement two machine learning models, Linear Regression and Logistic Regression, in order to gain a better understanding of their functionality, performance and limitations. We investigated their performance on two benchmarks datasets, using multiple different optimization techniques. We found that analytical linear regression performance varied in an unexpected way on test sets, smaller sized training sets performed surprisingly well, using ADAM with the gradient descent only improved the performance of linear regression slightly and the logistical regression performance exceeded expectations.

1 Introduction

For this project, we implemented both a Linear Regression and Logistic Regression model. We also implemented three optimizers that can be used with either one: Fully Batched Gradient Descent, Mini Batch Stochastic Gradient Descent, and Fully Batched Gradient Descent with ADAM. The performance of these models were evaluated with the different methodologies on two benchmark datasets. The first dataset [1] relates building parameters to two energy efficiency attributes and the second [2] processes categorical parameters to decide whether to classify a case under bankruptcy or non-bankruptcy . Our experiments revealed interesting findings. Analytical Linear Regression performed better on the test data over the training data even with a high split (80/20). Furthermore, we observed that the linear model was more effective at predicting 'Heating Load' than it was 'Cooling Load'. This was echoed in a paper on this dataset, which stated: "We demonstrated that we can accurately estimate HL with only 0.5 points deviation and CL with 1.5 points deviation from the ground truth (the simulated results)" [1]. There is a dramatic increase in performance when the training size is increased from 20 to 30%, but beyond 30% the performance fluctuates relatively little. Convergence speed with different batch sizes is not directly related to performance. Our implementation of non-linear bases did not improve our linear model. Lastly, the performance of the logistic regression model was remarkable, having essentially no error in its predictions no matter the scenario.

2 Datasets

2.1 Dataset 1

The first dataset contains 8 features which are used to predict 2 different outcomes, the heating load and cooling load. One important pre-processing step we did was normalizing the input values. As shown in Table 1, the mean of each feature was quite different, ranging from below 0.234 to 672. The standard deviations were also different. This caused our model to output abnormally large weights because it was trying to compensate for the range of very low to very high values. To fix this, we standardized the data so that, for all 8 features, the mean is 0 and standard deviation is 1. We did not change the outputs. Before performing our experiments, we also shuffled the dataframe to avoid any bias that may be occurring in the data.

2.2 Dataset 2

The data in dataset 2 are all categorical. The initial values are letters corresponding to certain categories. In order to work with this dataset, we converted the categories for each feature into binary one hot encodings by using the pandas "getdummies" function. This increased the number of columns in the dataframe as we needed more than one columns to encapsulate the encoding for the features with more than 2 categories.

The output is one of two classes: NB (Not Bankrupt) or B (Bankrupt). There are 107 occurance of the negative class (NB) and 142 occurances of the positive class (B). This means about 43% of the data is categorized in the negative class and 57% is in the positive class.

	Before Normalizing		After Normalizing	
	Mean	Standard Deviation	Mean	Standard Deviation
X1 (Compactness)	0.764	0.106	0.00	1.00
X2 (Surface Area)	672	88.1	0.00	1.00
X3 (Wall Area)	319	43.6	0.00	1.00
X4 (Roof Area)	177	45.2	0.00	1.00
X5 (Height)	5.25	1.75	0.00	1.00
X6 (Orientation)	3.50	1.12	0.00	1.00
X7 (Glazing Area)	0.234	0.133	0.00	1.00
X8 (Glazing Area Distance)	2.81	1.55	0.00	1.00
Y1 (Heating Load)	22.3	10.1	22.3	10.1
Y2 (Cooling Load)	24.6	9.51	24.6	9.51

Table 1: Feature and Output Statistics Before and After Normalization

3 Results

3.1 Linear Regression and Fully Batched Logistic Regression Performance

Table 2: MSE and Cross Entropy Errors

Test	Heating Load	Cooling Load	Bankrupcy
Unseen	3.58	4.57	9.99e-16
Train	4.55	5.31	9.99e-16

The data demonstrates that the linear model was slightly better at predicting the 'Heating Load' over the 'Cooling Load' feature. Furthermore, the linear model performed better on unseen data than the training data, which is unexpected. This does not necessary imply either under or over fitting. The logistic model outperformed the linear model, having essentially no error for both instances of testing.

3.2 Weights of Each Feature

Table 3: 'Heating Load' Weights

Bias	Compactness	Surface Area	Wall Area	Roof Area	Height	Orientation	Glazing Area	Glazing Area Dis
22.40	-4.21	12.46	-5.88	-18.06	7.48	-0.004096	2.65	0.308

Table 4: 'Cooling Load' Weights

Bias	Compactness	Surface Area	Wall Area	Roof Area	Height	Orientation	Glazing Area	Glazing Area Dis
24.68	-4.85	12.46	-6.65	-18.19	7.62	0.12	1.99	0.088

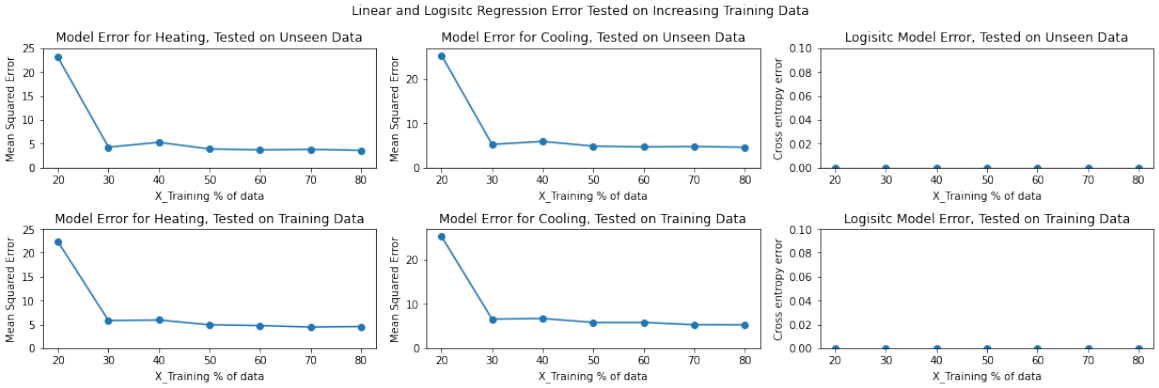
Table 5: "Bankruptcy" Weights

Bias	Industrial Risk$_N$	Industrial Risk$_P$	Management Risk$_N$	Management Risk$_P$	Financial Flexibility$_N$	Financial Flexibility$_P$
1.34	-0.35	0.39	0.044	0.62	-1.81	1.02
Credibility$_N$	Credibility$_P$	Competitiveness$_N$	Competitiveness$_P$	Operating Risk$_N$	Operating Risk$_P$	
-1.63	1.59	-2.91	2.19	0.19	0.88	

Comparing the weights of 'Heating Load' and 'Cooling Load' in Tables 3 and 4, we can see they are quite similar. In both models, 'Surface Area' and 'Roof Area' appear to have the biggest impact on the model predictions, with 'Surface Area' having a positive impact on the target and 'Roof Area' having a negative impact. On the opposite spectrum, 'Orientation' and 'Glazing Area Distribution' appear to have very little impact. Observing the weights for the logistic model, it appears that the 'Risk' features have little impact, whereas the 'Credibility' and 'Competitiveness' features have the largest impact on bankruptcy predictions.

3.3 Sample Growing Subsets of the Training Data

For the linear model, we can see for both 'Heating Load' and 'Cooling Load', there was always a drastic increase in performance when the training size was increased beyond 20 percent. However, the improvements in performance by increasing the sample size beyond 30 percent are relatively minor, but an interesting observation is that a 40 percent train size was worse than a 30 percent and all subsequent training sizes. Observations from 3.1 are repeated in this experiment. The linear model performed better on unseen data, was better at predicting 'Heating Load' than it was 'Cooling Load' and the logistic model had essentially no error, no matter the instance.



3.4 Comparing Batch Sizes with Mini-Batch Stochastic Gradient Descent

In order to compare different minibatch sizes, we split the data into an 80/20 train/test split while the learning rate (0.01) and number of epochs (64) remained consistent.

Figure 1 shows the convergence speed (cost vs epoch iteration) and final performance cost (size vs cost) for both models (using MSE for linear regression and CE loss for logistic regression). For linear regression, the fully batched technique had a baseline final cost of 3.47. The performance using mini-batch SGD resulted in final costs ranging from about 97-101. For logistic regression, the fully batched technique had a baseline cost of 0, while mini batch SGD had final costs ranging form 0.59-0.635.

The results show that, in general, the convergence speed becomes slower as the batch size increases. However, for linear regression, even though the smallest batch size had the quickest descent, the final performance was actually optimal with a batch size of 64. A batch size of 32 (with 64 epochs) would be the best configuration because it has the best trade off between convergence speed and final performance. For logistic regression, the (smallest) batch size of 8 is best because it is optimal in both convergence speed and final performance.

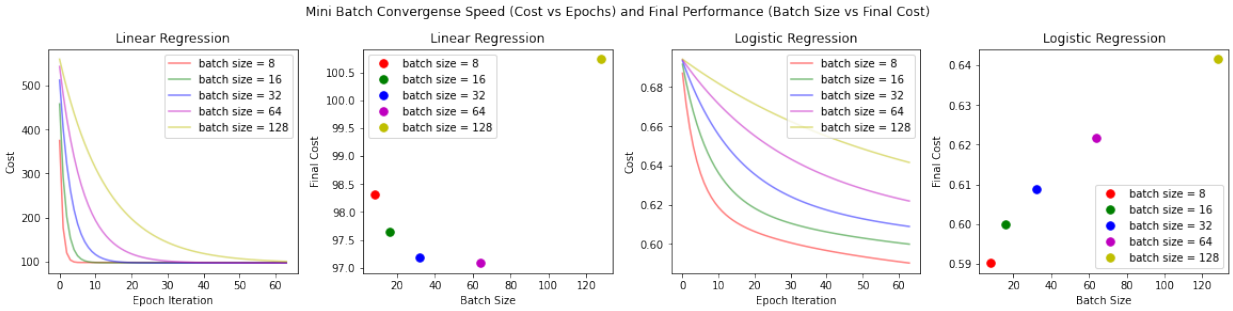


Figure 1: Mini Batch Convergence Speed and Performance with different batch sizes

3.5 Performance with Different Learning Rates

We compared the performance of the models with 6 different learning rates. Using the weight history, we computed the costs are each iteration and plotted them for each rate (Figure 2). In both cases, the performance increases as the learning rate increases.

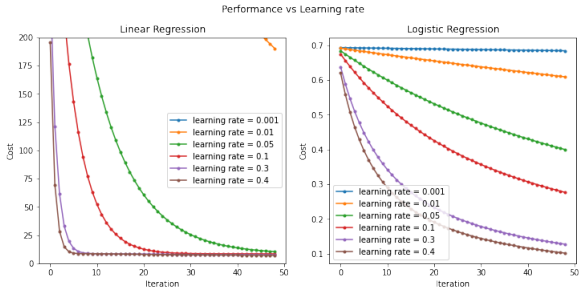


Figure 2: Performance vs Learning Rate, using Gradient Descent

3.6 Comparing Analytical vs Mini-Batch Stochastic Gradient Descent

A learning rate of 0.2 was used for the models in Figure 3. As is evident in the plots, the analytical solution performs a lot better since the result is a lot closer to a line with a slope of 1. With Mini-Batch SGD, the predicted values are a lot more clustered and are almost all within a certain range (between 15 and 30). Clustering can be due to the fact that the model is learning to fit patterns within the mini-batches instead of fitting to true patterns within the entire dataset.

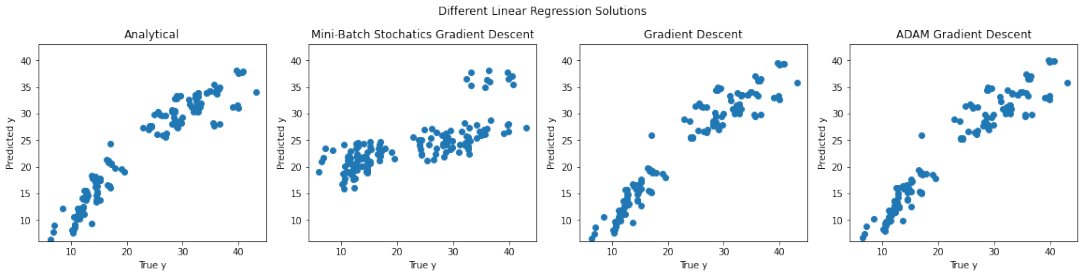


Figure 3: Scatter Plots of Analytical and Mini-Batch SGD

3.7 Evaluating ADAM Gradient Descent

In addition to the fully batched gradient descent and mini-batch SGD, we also implemented fully batched gradient descent with Adaptive Moment Estimation (ADAM). This involved keeping track of two exponential moving averages, one that uses moment to smooth out oscillations and another that adapts the learning rate to the parameter. As we can see in Figure 3, the scatter plots for both regular gradient descent and the one with ADAM are very similar and both perform well. However, the final performance for the one with ADAM performs the best. The final costs evaluated on the test set (20%) of dataset 1 for each technique can be found in Table 6.

	Analytical	Mini Batch Stochastic Gradient Descent	Full Gradient Descent	Full Gradient Descent with ADAM
Final Cost	4.26	27.72	3.47	3.39

Table 6: Final Costs for Different Techniques used with Linear Regression

3.8 Evaluating Non-Linear Bases

In this section we explored the use of non-linear bases with the intention of gaining insight on which features predictive ability could be improved by applying a non-linear function and which type of non-linear base is most appropriate with the ultimate goal of seeing if our analytical linear regression model could be improved in it's capacity to predict the 'Heating Load' feature with an 80/20 split. The performances were computed using MSE.

The approach was to test different models that have only a single feature. This feature would be one of the original features in the "ENB" data set with a non-linear function applied to it. This will allow us to determine which features performed the best as a non-linear base and are the most likely to improve our original analytical linear regression model. For computing the new feature, the standard deviation was kept to 1 and an array of 768 evenly spaced samples from 0 to 10 were used for the values of the parameter "mu". The idea was to ensure each sample in the feature had a different "mu" parameter in its computation. The following are the results when applying non-linear functions.

Table 7: Gaussian Feature Performance

Compactness	Surface Area	Wall Area	Roof Area	Height	Orientation	Glazing Area	Glazing Area Dis
30.08	29.35	37.90	16.67	15.18	49.24	48.92	48.67

Table 8: Sigmoid Feature Performance

Compactness	Surface Area	Wall Area	Roof Area	Height	Orientation	Glazing Area	Glazing Area Dis
67.63	49.08	57.86	51.53	73.10	49.66	55.10	48.62

The Gaussian Features seem to be much better predictors than the Sigmoid features, on average. Therefore, we will only use the "Roof Area" and "Height" Gaussian features in our new linear model, since we judge the error for the other non-linear features to be too high.

The following linear analytical model was trained with an 80/20 split with all 8 features, but with Gaussian "Roof Area" and "Height" features.

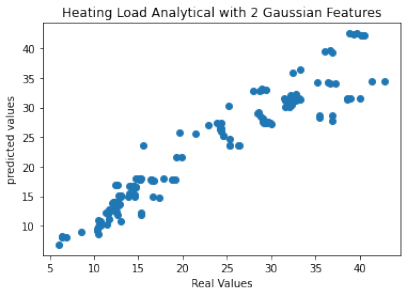


Figure 4: Linear Regression with 2 Non-Linear Gaussian Bases

The MSE for this model was 4.31, which is higher than the 3.58 error of the original model. We expected the use of nonlinear bases to improve the performance. Since it didn't, we suppect this is because the model became too expressive which lead to overfitting.

4 Conclusion

This project helped us gain a better understanding of two key machine learning models and a number of different optimization techniques that can be used alongside them. After running linear regression using 4 of these techniques, we found that fully batched gradient descent with Adaptive Movement Estimation (ADAM) was the best whereas Mini-Batch Stochastic Gradient Descent had the worst final performance. When using gradient descent, we found that a larger learning rate and a smaller batch size (with mini-batch SGD) always had a faster convergence rate at the start. However this quick convergence did not always reflect in a better final performance. In fact, for mini-batch SGD, the optimal batch size is effected by the number of epochs. Another interesting finding was that, when we implemented linear regression with non-linear bases, the final performance did not improve, and it actually worsened slightly. We believe this is due to our implementation of the non-linear features. It is likely that for the Gaussian features to be effective, the optimal values of the parameters "mu" and "s" would need to be discovered, instead of arbitrarily setting them as we did. Learning these parameters would be a worthwhile investigation for the future. Our logistic regression model was difficult to evaluate since the final costs where always so small and it always seemed to perform very well. Another, interesting investigation for the future could be to evaluate how the addition of regularization affects the models.

5 Statement of Contribution

All members worked on developing the models. Due to a prolonged period of illness and unrest, Annabelle was unable to contribute beyond this. Ethan ran experiments presented in subsections 1-3 and 8 under Section 3. Amani ran experiments presented in subsections 4-6.

References

[1] A. Tsanas and A. Xifara, "Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools," vol. 49, pp. 560–567, 2012.

[2] D. Dua and C. Graff, "UCI machine learning repository," 2019.