COMP551: MiniProject1 Report

Leila Wang, Raphael Wang, Oliver Wang

February 2023

Abstract

In this project, we investigated the performance of two machine learning models on two benchmark datasets. The linear regression model and the logistic regression with different optimizers (i.e., SGD, mini-batch SGD, Adams) were implemented to fit two datasets. Experiments showed that as the size of training sets increased, the performance of both models increased in testing data and decreased in training data. As the learning rate increased, The performance of both models converged to different values. For further investigation, Adams optimizer and L2 regularization are applied to the models.

1 Introduction

This project aims to implement linear and logistic regression on two distinct benchmark datasets and provide an analysis of these two models. Dataset 1 is the energy efficiency data set assessing the heating load and cooling load requirements of buildings. There are in total 8 features and two responses. We compared linear models with four types of optimizers on this dataset. Datasets2 makes predictions of bankruptcy based on expert-generated qualitative characteristics. Experts assess the qualitative risk factors and, based on their subjective expertise, assign appropriate levels to these characteristics, such as positive, average, and negative to classify bankruptcy from non-bankruptcy. [1] In this project, we will use logistic regression for this classification.

2 Datasets

2.1 Data 1: Distribution and Preprocessing

Data 1 includes 8 features and 2 dependent variables. There are no missing values or malfunctioning features. The scatter plots for each of the two output variables and each of the (normalized) input variables are shown in Figure 1. These scatter plots show that any functional relationship between the input variables and the output variables is not trivial.

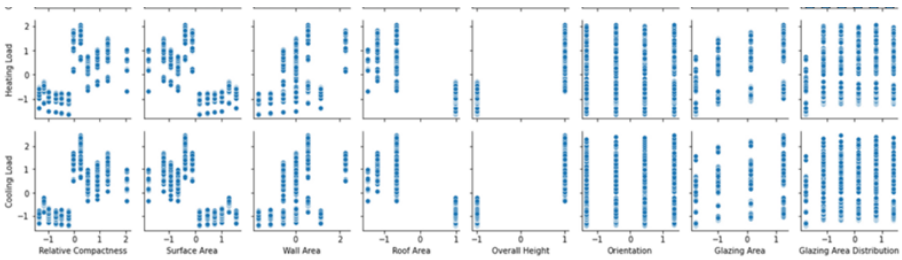

Figure 1: Scatter plot for each feature

Figure 2 illustrates the correlation heatmap. The roof area is found to be negatively connected (0.89) with the two response variables and overall height is found to be strongly positively correlated (0.9). The outcome demonstrates that the only insignificant factor is Direction.

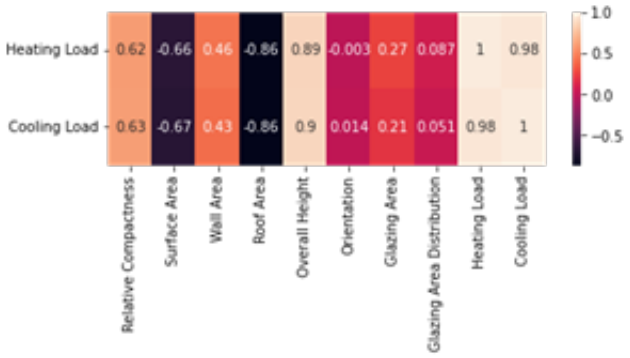

Figure 2: Matrix of Spearman rank correlation coefficient

2.2 Data 2: Distribution and Preprocessing

Data 2 contains 6 categorical predictors and 1 binary response. Therefore, we applied the one-hot encoding to convert each categorical variable into 3 new categorical columns (to make those columns into binary values of 1 or 0). Data 2 does not have missing values, thus no data cleanness needs to be done. Figure 3 illustrates the distribution of the 6 categorical predictors.

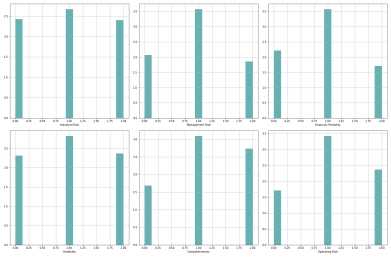

Figure 3: Data 2 Visualization: Histograms of all categorical predictors

3 Results

3.1 Data 1

3.1.1 The Performance of Linear Regression Model

According to Figure 4, testing data for both responses showed a higher cost than the training data. This occurred because testing data might contain data that was uncommon and was more error-prone when making predictions whereas the training dataset was assessed on the same data and the cost function was minimized. The same idea was also reflected in training data having a slightly higher R-squared value than test data which indicates a better fitting in training data.

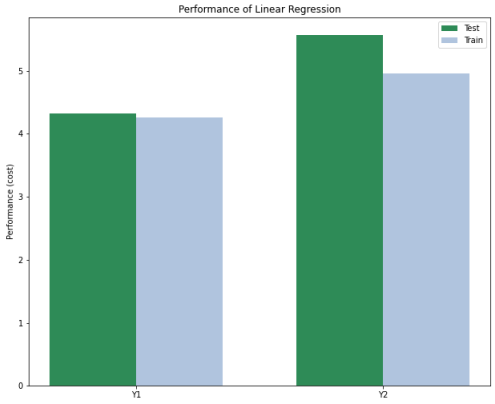

Figure 4: Performance of Linear Model for Data 1

3.1.2 The Weights of the Trained Model

According to the weights (Figure 5) for the linear regression model, we observed that height, surface area and relative compactness(RC) are the most significant features which agrees with the work of Tsanas and Xifara. [2]

index	heat_weights_df	cool_weights_df
Relative Compactness	-7.375310	-8.019792
Surface Area	-9.175629	-9.798465
Wall Area	3.128260	2.686447
Roof Area	1.205493	1.764034
Overall Height	7.508334	7.757159
Orientation	-0.043767	0.116185
Glazing Area	2.709908	2.005080
Glazing Area Distribution	0.370599	0.096161
bias	22.295060	24.610579

Figure 5: Weights of Linear Model for Data 1

3.1.3 Performance Under Growing Subsets of the Training Data (20%,30%,...80%)

For the linear model, the change in training size led to different trends of performance in the fitting of testing and training data. Showed in Figure 6, the MSE of all linear models decreased as the training size increased which means that there is a positive correlation between testing performance and training size. It means that with larger training data, predictions of the linear models on unseen data are more likely to be true. On the contrary, the performance of linear models in training data became worse as the training size increased. With a higher volume of training data, it was more difficult for the model to fit all of them which increased the cost and thus led to lower performance. For instance, a small dataset may follow a linear trend but as more data are included, the model may need a polynomial base function to fit them.

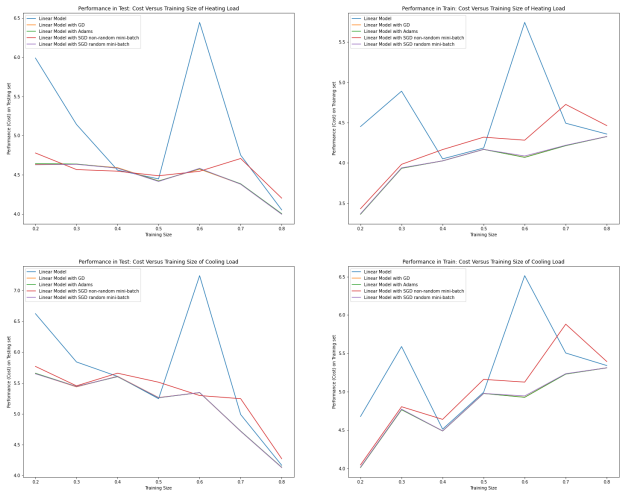

Figure 6: Relation of performance and training size of linear models

3.1.4 Performance Under Growing Mini-Batch Sizes

Five different batch sizes were employed for linear regression (Figure 7). As batch size grows, the cost convergence speed decreases. This shows that the estimation of the gradient will be less accurate the smaller the batch. We can observe that the batch with size 8 converges the slowest and batches with 64 and 128 converge at the highest speed. Although the batch with full size converge relatively slowly, it was still declining when other batch sizes approached the asymptotes. This might be accounted for by the various movements that tiny batch sizes produce. Compared to larger batch sizes, a gradient with a small batch size oscillates much more. Although this oscillation can be seen as noise, it aids in escaping the local minima in non-convex loss landscapes, so it converges to minima faster while a full batch might get trapped in a saddle point.

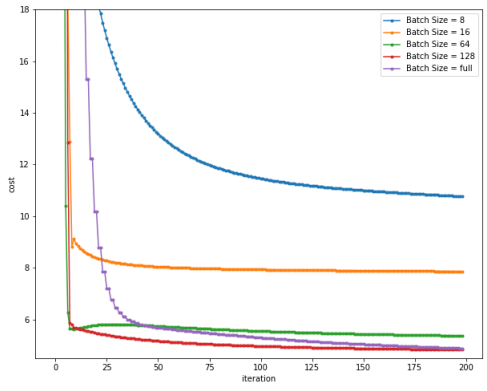

Figure 7: Covergence speed of SGD models with different batch sizes

3.1.5 Performance Under Different Learning Rates

Since there was no improvement in each step when the learning rate was zero, the performance of the linear model is the worst, indicated by the highest cost and lowest R^2. When the learning rate is extremely low (e.g. 0.001), the convergence rate is low accordingly. If the cost function is convex, it will eventually attain a global minimum. The cost of the linear model for HL converged at 5.65 and at 4.64 for CL (Figure 8).

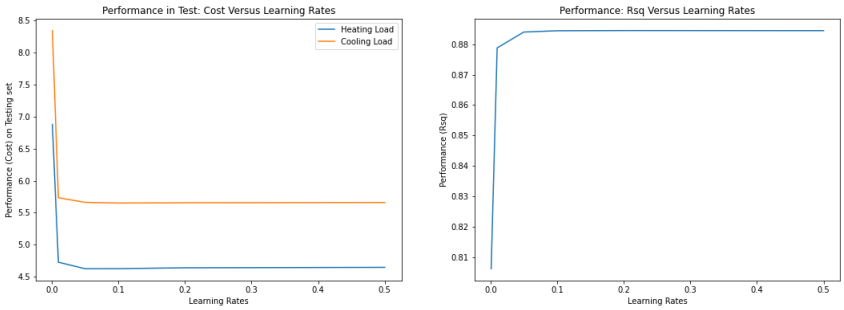

Figure 8: Performance and Learning rate

3.1.6 Performance Comparison: Analytical Linear Regression Model and the Mini-Batch SGD Model

We also compared the analytical linear model with the mini-batch SGD linear model. Figure 9 shows that the cost of the mini-batch approach was substantially lower than the cost of the linear regression method for the most part and the fluctuation is smaller. We could be trapped at a saddle point too early and ended the training with parameters that were far from the required performance due to the non-convexity of the cost function. Using a mini-batch reduces this possibility because several batches will be taken into account during each iteration, ensuring a robust convergence. The plot shows a peak between 0.5 and 0.7 training size. This could be explained by the fact that a bad fit (perhaps caused by non-linearity) affects 50% to 70% of the selected trained data and results in a high MSE. Mini-batch optimization performed better at this x-axis location in comparison.

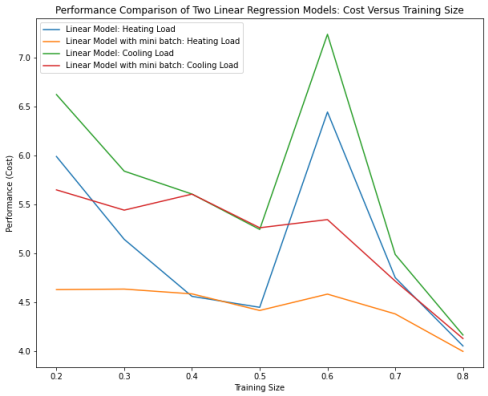

Figure 9: Performance of Linear Models with or without mini batch

3.1.7 Comparison of the performance: Multiple linear models

We compared the performance of various linear models on testing data with different sizes. The models include the basic linear model, the linear model with gradient descent, two SGD linear models, and Adams linear model. The difference between the two SGD models is whether the mini-batch (the subset) is random every time or not. The performance is represented by the cost and R^2 (Figure 10). According to the cost, the basic one had the highest fluctuation when encountering a biased training sample (60%). Nonrandom SGD also fluctuated when the proportion of the training sample is 70%. This is likely due to the nonrandomness which can become a biased batch. Adams, GD, and SGD models share similar patterns with smaller fluctuations. R^2 shows that GD has the best behavior. However, with more iterations, Adams is likely to surpass it according to the trend in Figure 10.

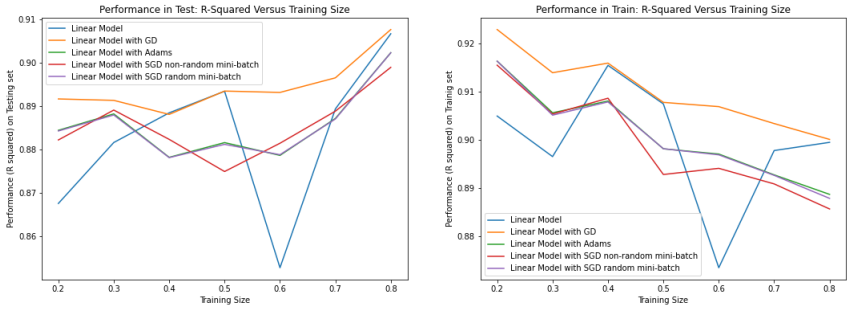

Figure 10: Performance of various Linear Models

3.2 Data 2

3.2.1 The Performance of Fully-Batched Logistic Regression Model

Figure 11 reports the Cost, Accuracy, and ROC AUC score of the fully batched logistic regression model in both training and testing data. It turns out that the accuracy values for both training and testing data are very close to 1 (0.995 and 1.0 respectively), which means the model did an almost perfect binary classification on data 2.

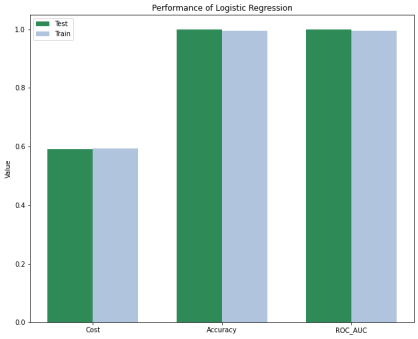

Figure 11: Task 3.1: Cost, Accuracy and ROC AUC score for training and testing data

3.2.2 The Weights of the Trained Model

Figure 12 reports the weights of our trained fully batch logistic regression model. We utilized an 80/20 train/test split for fully batched logistic regression for this section. We used one-hot encoding and obtained 19 weights parameters as each feature was divided into three-factor features plus the bias.

Figure 12: Task 3.2: Weights of the trained fully batch logistic regression model

	weights
Industrial Risk_A	0.030057
Industrial Risk_N	-0.018863
Industrial Risk_P	0.020754
Management Risk_A	0.019440
Management Risk_N	-0.023462
Management Risk_P	0.035969
Financial Flexibility_A	0.059656
Financial Flexibility_N	-0.078240
Financial Flexibility_P	0.050532
Credibility_A	0.043143
Credibility_N	-0.078076
Credibility_P	0.066880
Competitiveness_A	0.047111
Competitiveness_N	-0.097507
Competitiveness_P	0.082343
Operating Risk_A	0.012694
Operating Risk_N	-0.016461
Operating Risk_P	0.035714
bias	0.031948

3.2.3 Performance Under Growing Subsets of the Training Data (20%,30%,...80%)

We sampled growing subsets of the training data (20%, 30%, ...80%) to see its impact on the classification performance. Figure 13 illustrates the cost, accuracy, and ROC AUC score of growing training sample size. The logistic models show a similar trend as the linear regression models for data 1. As training data size increases, the training data cost increases while the testing data cost decreases; the testing accuracy increases while the training accuracy decreases. Since the model was trained with more data, it can generalize better to the unseen data. This is in accordance with what we learned from lectures.

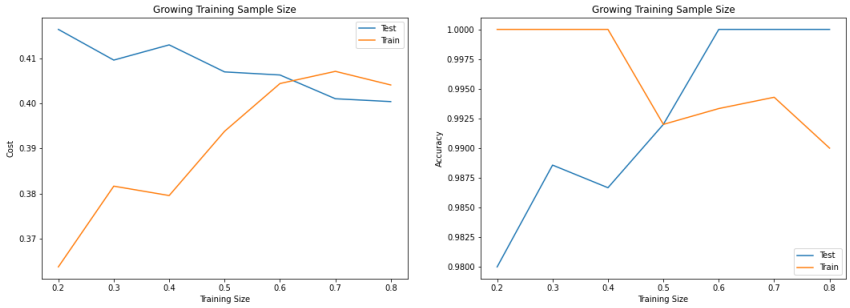

Figure 13: Task 3.3: Cost and Accuracy for training and testing data with respect to growing subsets of the training data

3.2.4 Performance Under Growing Mini-Batch Sizes

Five different mini-batch sizes were employed for the mini-batch logistic regression model. Batches of sizes 32, 64, and 128 yield similar convergence cost values of around 0.46. Batch size 8 converges to the highest cost. It is clear that the full batch size model converges at the slowest speed but to

the lowest cost for this model (it is still decreasing while other batch sizes plateaued). For the rest of the batch size plot, it is hard to rank their convergence speed since the difference between them is minute, but it seems to have a trend that smaller batch size converges at a slower speed than higher batch sizes.

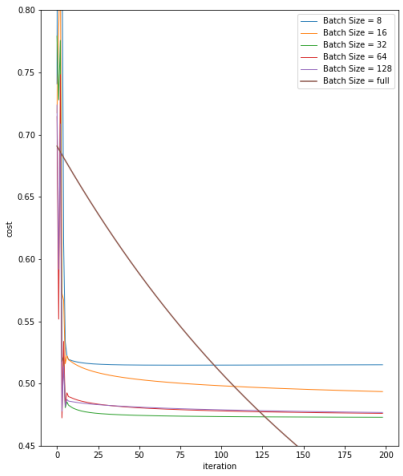

Figure 14: Task 3.4: Convergence speed of different mini-batch sizes

3.2.5 Performance Under Different Learning Rates

We chose five different learning rates (0.001, 0.01, 0.05, 0.1, 0.2) to experiment with their impact on performance. Overall, the test data has a lower cost than the train data in every learning rate. The cost for both the test and train dataset decreased as learning rates increased. Note that the accuracy for both testing data and training data is very high, so it does not show a clear trend of how different learning rates can impact the model performance.

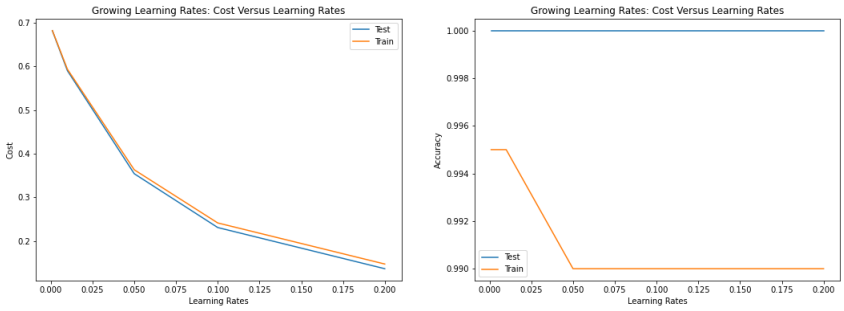

Figure 15: Task 3.5: Performance (cost and accuracy) under different learning rates

3.2.6 Adding Regularization to the Fully-Batched Logistic Regression Model

We also added L2 regularization to the fully-batched logistic regression model to observe how regularization techniques can impact network performance. We chose the λ to be equal to [0, 0.01,

0.1, 1, 2, 5, 10, 15] and observed the trend of cost and accuracy. Figure 16 shows that the cost of the logistic regression model increases when λ increases. This is obvious because L2 regularization adds a penalty term to the cost function which is proportional to λ. We also found that the accuracy of the classification decreases as λ increases. It is noteworthy that the accuracy is originally close to 1 when using small λ rates, but then gradually decreases and converges to 0.88 as λ increases. This might suggest that the original logistic regression model has overfitting issues since the accuracy is always found to be 1. After adding the L2 regularization to the model, the classification accuracy becomes stable at 0.88.

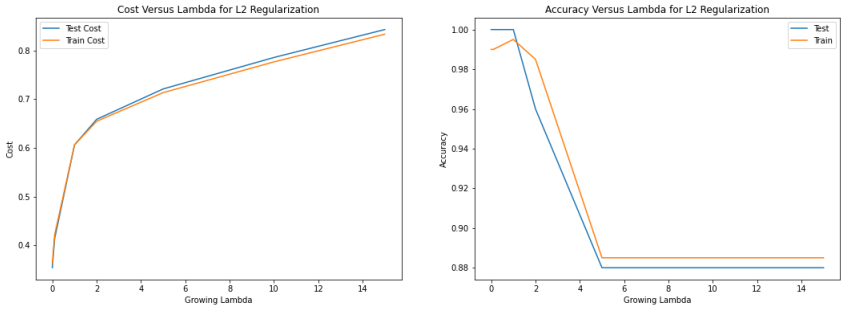

Figure 16: Cost and Accuracy when adding L2 Regularization to the fully-batched logistic regression model

4 Discussion and Conclusion

The approach taken in this project has some limitations. The first dataset's density plots and scatter plots provide convincing evidence that linear techniques are inappropriate for the data present in this application. In the future investigation, we can try various other more complicated machine learning techniques to get a more accurate mapping. These methods could be better suited to cases in which normality is absent.

Additionally, the plot below showed that our dataset has a problem with data-based multicollinearity. When two or more factors in a regression model are moderately or highly associated, multicollinearity occurs. Severe multicollinearity can raise the coefficient estimates' variance and make them more susceptible to even small model adjustments. The statistical power of the study is reduced by multicollinearity, which can also cause the coefficients to change sign and makes it more challenging to define the correct model.[5] To prevent this in the future, we can remove highly correlated variables and use Principal Component Analysis to extract new features (PCA).

There are several ethical concerns that may arise with machine learning programs. A key ethnic concern is sampling bias, which occurs when the data used to train a model is not representative of the correct population.[4] Certain ethnic groups, for example, may be overrepresented in real-world data, skewing the AI system's results. In the case of the second dataset, the sample size may influence the conclusion. For example, researchers gathering data in an affluent neighborhood would produce different accounts for each feature than data collected in a poor community. As a result, acquiring data from diverse regions may result in bias, and if the model is trained on biased data, the prediction may not reflect the true situation, resulting in ethnic ramifications.

Results of the work demonstrate that Nadam performed more excellently and steadily than other optimization strategies in terms of convergence rate, training speed, and performance. So we can also try this algorithm in the future. [3]

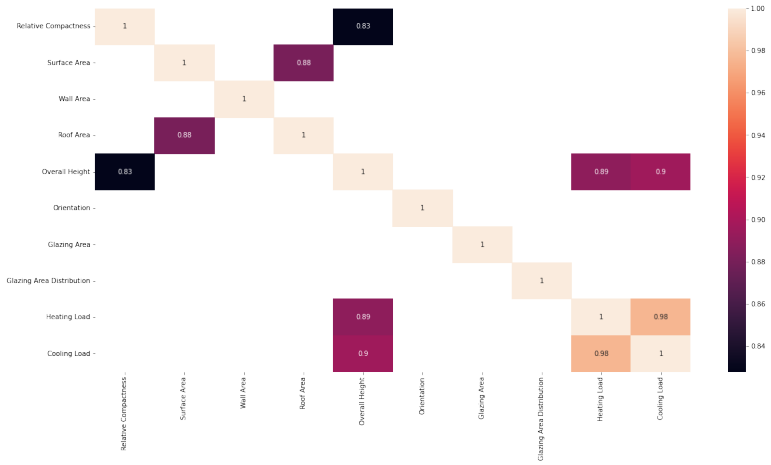

Figure 17: Multicollinearity for Data 1

5 Statement of Contributions

All three group members contributed to this project equally. The data preprocessing, model implementation and experiments for data 1 were mainly completed by Oliver Wang. The data preprocessing, model implementation and experiments for data 2 were mainly completed by Leila Wang. The finilizing steps and report written work were mainly contributed by Raphael Wang.

References

[1] Myoung-Jong Kim and Ingoo Han. "The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms". In: *Expert Systems with Applications* 25.4 (2003), pp. 637–646. ISSN: 0957-4174. DOI: https://doi.org/10.1016/S0957-4174(03)00102-7. URL: https://www.sciencedirect.com/science/article/pii/S0957417403001027.

[2] Athanasios Tsanas and Angeliki Xifara. "Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools". In: *Energy and Buildings* 49 (2012), pp. 560–567. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2012.03.003. URL: https://www.sciencedirect.com/science/article/pii/S037877881200151X.

[3] Ersan YAZAN and M. Fatih Talu. "Comparison of the stochastic gradient descent based optimization techniques". In: *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*. 2017, pp. 1–5. DOI: 10.1109/IDAP.2017.8090299.

[4] P. Bhandari. *Sampling bias and how to avoid it: Types examples, Scribbr*. 2022. URL: https://www.scribbr.com/research-bias/sampling-bias/.

[5] Jireh Yi-Le Chan et al. "Mitigating the Multicollinearity Problem and Its Machine Learning Approach: A Review". In: *Mathematics* 10.8 (2022). ISSN: 2227-7390. DOI: 10.3390/math10081283. URL: https://www.mdpi.com/2227-7390/10/8/1283.