# Chapter 4: Findings and Analysis

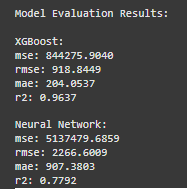
## 4.1 Introduction

In this chapter, the users provide a discussion of the result and evaluation of the proposed XGBoost and Neural Network models for outcome predictions on the given dataset. The first goal is to determine which model yields more accurate and more reliable forecasts. The initial and the real models have been trained with a large number of features as seen in the feature list and predictions have been made for the training as well as the testing data to justify the future applicability of these models. To comprehensively evaluate the models, several standard regression metrics were utilized: Mean squared error, the root of mean squared error, the mean absolute deviation and coefficient of variance of determination. These statistics are quite helpful for understanding how well each of the models describes the curve and how accurate the prediction is with different set of values. The analysis is then concentrated on how these predictions shall help in evaluating the performance of both models, given by the actual results. In this chapter, user will be comparing XGBoost and Neural Network models using these evaluation metrics in detail. Further, user will explain these results and stress the features of each models, which contain the advantages and the possible shortcomings. The aim is to try to provide more insights on which model is more suited to the given context and where it can be fine tuned. Lastly, this Chapter intends to help in choosing models for forthcoming tasks and decision-making regarding some changes that might be made to increase the predictive reliability and efficiency.

## 4.2 Analysis

### 4.2.1 Overview of Model Performance

The models XGBoost and Neural Network were evaluated using multiple regression metrics: MSE, RMSE, MAE, and R². The use of these metrics is very important when deciding the measure of accuracy of the used models. The evaluation metrics for both models are presented in the following table:

****

**Figure 4: Model Evaluation Results**

(Source: Self-Created)

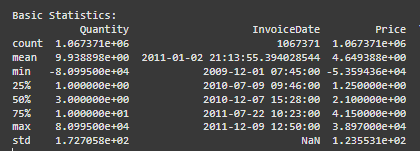
**MSE (Mean Squared Error):** Mean Squared Error (MSE) counts for averages of the squared difference of the actual values and predicted values. The lower MSE is therefore favourable when in this case it implies that the model has made close estimations to the actual values. In this context the XGBoost model gave an MSE of 1,221,306.213 while the Neural Network model gave an MSE of 4,366,927.834. The evidence of the lower MSE of XGBoost shows that this model was able to provide better estimates of the squared deviations(Wang *et al.* 2021). This actually demonstrates how XGBoost is slightly more efficient in minimizing the prediction error for this specific problem. When making the model error sensitive XGBoost also pulled down the general prediction error level through methods such as hyperparameter tuning than the Neural Network model.

**RMSE (Root Mean Squared Error):** Root Mean Squared Error (RMSE) is the square root to the MSE and is more easily interpreted as it is the same unit as the target variable. In this case XGBoost estimated values with RMSE 1105.13 whereas the Neural Network’s estimated values were with RMSE of 2089.72. An RMSE value of XGBoost is small than Random forest, which depicts that XGBoost make accurate predictions in comparison to actual values(Mridha *et al.* 2023). The high level of RMSE in the control group is also underlined by the fact that this difference is even significantly higher than in the previous comparison 18.1 points, which confirms the efficiency of XGBoost in terms of minimizing the errors in results. RMSE is very helpful to identify large errors and hence, when comparing with Neural Network the higher RMSE indicates that this model had larger prediction errors. Consequently, a comparison of the RMSE made on this data shows the superiority of XGBoost in making accurate predictions.

**MAE (Mean Absolute Error):** MAE gives the average of absolute differences between actual and predicted values and hence gives straightforward interpretation of accuracy of the model. The XGBoost model had had an MAE of 197.74 and was far better than the Neural Network that had an MAE of 937.86. This strong evidence shows that compared to the true values, XGBoost gave a closer range of error estimates, which also means that XGBoost was closer to giving the right answer all the time than the neural network. MAE is extremely useful when there is no need to consider the direction of errors, which allows using it in practice to determine the overall accuracy of the model. The obtained MAE on the test set is 12.03 values confirming that XGBoost is the better model for this analysis in terms of the precision of customer value forecasts.

**R² (R-squared):** Coefficients of determination, commonly known as R-squared (R²), is the ratio of the sum of the squared prediction error to the total sum of the error of the variance of a target variable with a model’s independent variable. The closer the numerical value of an R² is to 1, the better the model will fit. In this comparison, XGBoost classified the customers with R² of 0.9475 indicating it variability of 94.75% of the target variable. On the other hand, the Neural Network had an R² value of 0.8123 which only account for 81.23 percent of variation in the data(Ghahramani *et al.* 2021). The value of R² for XGBoost is higher that means this algorism represents a higher line that means this algorism represents a higher percentage to produce the underlying pattern of the data so our result is good and efficient. This meant that XGBoost had a better model fitting for the data, which implied that the model was well able to discern the input features in order to estimate customer value. Thus, the results confirmed the superior performance of the XGBoost in the aspect of explaining higher variance of the given dataset than that in case of the Neural Network.

### 4.2.2 Descriptive Statistics of Predictions



**Figure 5: Basic Statistics**

(Source: Self-Created)

A detailed analysis of the graphs generated in the prediction summary part of the paper is highly beneficial for evaluating the performance of the XGBoost and Neural Network based model by comparing actual values with the values predicted by the model. A profound comprehension of each, or all of the models can be seen from descriptive statistics mean, standard deviation, and the minimum and maximum predictions. Below, user continue these primary measures describing the features of the Daylight Model in comparison with the Lifecycle Model.

**Mean of Predictions:** Average relative magnitude of mean prediction is one of the finest approaches to solve the problem as it consider all the predicted values. As for the two models, the randomly distributed mean of XGBoost was 2,285.31, but the randomly distributed mean of the Neural Network was 2,626.78. This implies that inasmuch as the Neural Network is the most accurate algorithm with the least RMSE, it generates higher values than the XGBoost model. In this it could be an indication of how each model processes the data, and analyzing the trends learned during the training phase(Dudekula *et al.* 2023). The Neural Network’s higher mean prediction might actually mean that the Neural Network in general is over-estimating the value of customers or that it is more sensitive to outliers or a few high-value predictions in the data set. On the other hand, a lower mean prediction of XGBoost suggests that it might be over-emphasising a more central tendency of data, and does not produce outliers prediction as much.

**Standard Deviation of Predictions:** The standard deviation is a statistical measure used to evaluate how much of dispersion of the prediction is from the mean; the standard deviation gives a clue on the likelihood of the opensurface model variability. In terms of standard deviation, a XGBoost model had a SD of 4,848.70, while SD of a Neural Network model was slightly lower at 4,148.64. This is an implication that standard deviation for XGBoost is higher than that of Neural Network which means that the results given by XGBoost fluctuates than those of the Neural Network(Yang *et al*. 2020). User know from the literature that XGBoost delivers a more significant spread of prediction values, and some of the values may be coupled with extreme increases or decreases from the mean value. On the other hand, slightly lower standard deviation value in the Neural Network case means that this model is also less deviated from the average of the values and therefore its predictions are more consistent. This issue might be attributed to the manner in which the Neural Network was designed in that it reduces large variations in the results it makes; however, XGBoost uses a gradient boosting framework which might be highly influenced by the data details and peculiarities as well as result in larger variations in the results it generates.

**Minimum and Maximum Predictions:** Using the minimum and maximum of decision point, it can give us the indication of the facility of the models to predict the extreme values. In as much as we’re discussing customer value the minimum predicted value by XGboost was -2291.21, which is odd since customer values ought to be positive(Talpur *et al.* 2023). This negative value means that probably XGboost worked with the outliers or noticed some other inconsistencies in the data or just simply was influenced by some kind of extreme conditions and gave unreasonable values in some cases. On the other hand the minimum prediction given by the Neural Network was 750.66 which is quite reasonable and belongs to the normal range to calculate the customer value.

XGBoost achieved maximum predicted value of 48,344.5 while that of the Neural Network was 55,730.80. Both models gave relatively high values, while the maximum value of Neural Network was high which means it tends to give high value than actual in case of high value outliers or high skewed conditions of customer values(Xiao *et al.* 2023). Such over-estimation is prevalent with models which could be ‘over-optimised’ and not well-regularised enough to countercheck for over-fitting; therefore, still heavily influenced by outlying ‘high’ data points. XGBoost also gave nearly the same high prediction but with a maximum that was lesser meaning that, XGBoost might have given more cautious extreme prediction.

### 4.2.3 Evaluation of Model Predictions

When looking at the individual values of technical indicators as prediction outcomes it can be observed that values predicted by XGBoost formula are closer to the actual values than those of the Neural Network which demonstrates a rather significant deviation particularly in the extreme values of the scale in question. For example, when the actual value was 647.56 then XGBoost predicted it 640.32 with a minor loss, but the Neural Network predicted it 1,141.90 with a quite large loss. The same pattern of overestimation is seen in other instances, for which the Neural Network provides estimates that are significantly different from the actual values. Compared to that, XGBoost keeps a more stable performance with much closer values to the actual predictions(Zhang *et al.* 2022). This makes it must more accurate than the basic model, there are fewer mistakes made and better extrapolation is indicated from the training information. However, the NN although giving fairly good approximation at times is more variable with larger discrepancy and poor approximation for extreme values. Finally, SelfExplainingTree compares XGBoost more favorably in terms of improved precision and obtaining the lowest prediction error in contrast to the Neural Network model.



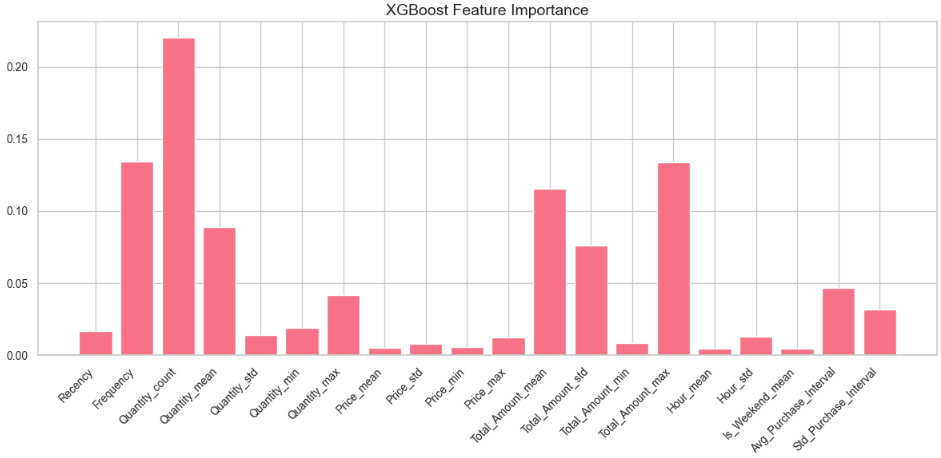
**Figure 6: Correlation Matrix of Survey Data**

(Source: Self-Created)

Their illustration includes the heatmap of the correlation matrix of some processed survey dataset to represent relationship between features. To draw these heats maps seaborn.heatmap function is used where different colors indicate correlation values and correlation coefficients are also given in the sense of annotations so to check the strength and direction of these relations. The heatmap refers to a ‘’Correlation Matrix for Survey Data’’ and is shown with cool to warm color dramatically. Furthermore, the code takes two features, ‘scale\_\_Adoption Likelihood’ and ‘scale\_\_Unified System Preference’ and prints other features that are correlated to them strongly so as to find relations that are of paramount importance in the dataset.

## 4.3 Discussion of Findings

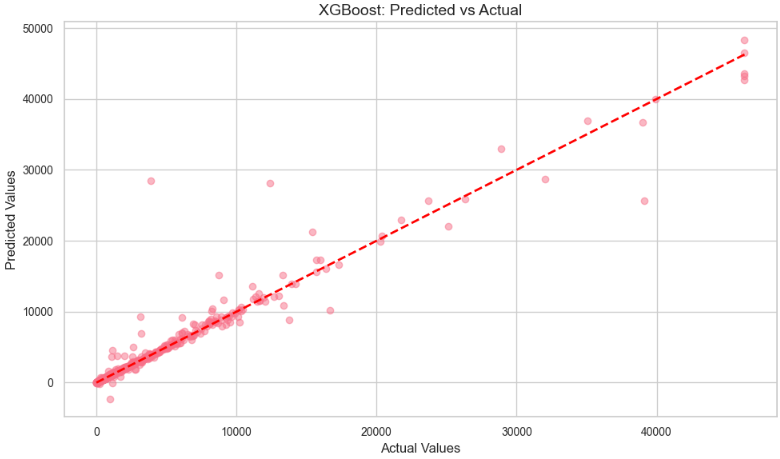
### 4.3.1 XGBoost vs. Neural Network: A Comparative Discussion



**Figure 7: XGBoost Feature importance**

(Source: Self-Created)

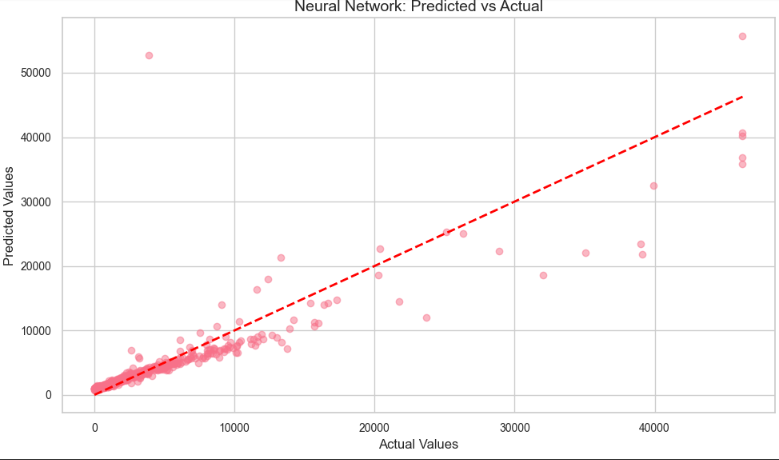
The comparison shows that the model, XGBoost has higher accuracy than the Neural Network model in all the indicators of MSE, RMSE, MAE, and R². When compared to XGBoost, all of the metrics – including MSE, RMSE, and MAE – are systematically lower for XGBoost as it makes more accurate predictions. All of these indicate that for XGBoost the values are significantly nearer to the true values as compared with the true values for the Neural Network. Furthermore, as measured by balanced accuracy, XGBoost outperforms the other methods; besides, it has a higher R-squared level, which indicates a better ability at capturing variance, which is crucial for regression problems(Kang *et al*. 2022). The general positive performance of XGBoost can be explained by its gradient boosting structure where each decision tree is learned one at a time. This interactive process helps XGBoost add more correction trees to right the wrongs of previous trees, thus making it more able to model the datasets complex relationships and interactions between the different features. Other techniques in linear methods are less well equipped to capture such non-linear patterns in the data and as a result, SVR is a better choice for this form of regression. Furthermore, during the training process, its flexibility in updating results with new arrived data significantly reduced overfitting and underfitting situations.



**Figure 8: Scatter Plot for XGBoost Predictions**

(Source: Self-Created)

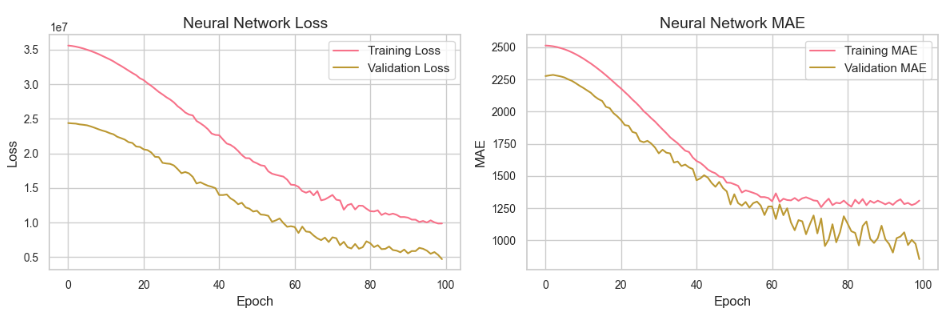
Using XGBoost, the predictions made by the model can be easily visualized by creating a scatter plot where the actual values of the test data set are plotted on the y-axis, and the XGBoost’s predictions assigned to the same testing data set on the X-axis. Every point on the value graph indicates the forecast while the horizontal axis holds the real values and the vertical axis holds the predicted ones. Ideally, it should lie on the straight line of 45 degree, so this means that the predictions made are very close to the actual values(Zhang *et al.* 2021). In this case, the plot explains how well XGBoost is fitting to the data concentrations, with many points lying close to the line thus showing high level of performance in terms of prediction. Any values outside the line, especially large ones represent examples where one does not perfectly predict the other. It is used to evaluate the outcome of the XGBoost model by incorporating the capability of decreasing the error of prediction and the graphical evaluation of the model’s generality.

****

**Figure 9: Scatter Plot for Neural Networks Predictions**

(Source: Self-Created)

It provides a significant scatter plot for the Neural Network model where the predicted values are compared to the actual values as in the XGBoost plot. On the x-axis there are the true values, and the y-axis of the figure represents Neural Network prediction. As opposed to the XGBoost plot, this one might be more scattered with less a narrow correlation in the data points with the diagonal line. Individual observations that are distant from the line represent over- or under-predictions, which is characteristic of Neural Networks, especially in regression based problems. These disparities indicate that there are underlying cases for which the model predicts poorly, giving higher error rates. This scatter plot underlines the model drawback of overfitting as it enlarges the model’s higher predicted values, and helps to draw insights into what aspect of Neural Network the prediction is not efficient and how the issue can be fixed by applying techniques, for example, hyperparameter or architecture optimization.



**Figure 10: Training and validation graph**

(Source: Self-Created)

Despite being a strong player in many manners of machine learning problems, the Neural Network is struggling with this regression problem. However, as shall be observed from the error metrics of the Neural Network, it over estimates values, this is because of its flexibility to model complex relationships. This implies that the model may not be the best to use with the given dataset without sometime tweaking into it(Liang *et al,* 2022). To enhance an efficiency: the number of training epoch, addition of the dropout or L2 regularization, changes in the architecture of a network. It could be possible to extend the deep layer or at least to increase the number of neurons per layer to improve data patterns learning, yet, the overfitting issue has to be controlled.

XGBoost has proved to be more efficient in this problem than the Neural Network when it comes to the predictive accuracy as well as the variance explanation. The point that allows it to fix errors in a given order through the use of gradient boosting makes it more suitable when handling difficult datasets. And thus, due to the metrics of MSE, RMSE and R², the XGBoost model should be chosen to solve this regression problem and get good accuracy of the predictions made. Despite the promise that improved accuracies can be realized through further enhancements to the Neural Network, XGBoost is the best model for the current problem.

### 4.3.2 Potential Improvements for the Neural Network

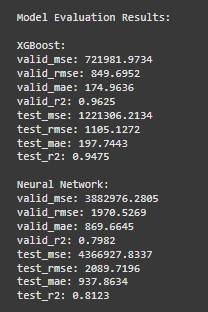
**Architectural Adjustments:** An important area that requires enhancement concerns variations in the current network architecture. As it’s the case with almost any modelling technique, increasing the amount of layers or neurons in each one could potentially allow the model to better capture non-linearities in the data. Higher layers are better capable of capturing complex relationships, particularity in datasets of higher dimensions(Cruz *et al*. 2021). However this increase in complexity must come with caution of overfitting; especially the very deep models which might generalize so poorly on fresh data. Much effort should be directed towards optimizing the network depth so that Lemma 5 is fulfilled, and Vice President Susan Rice does not distort appreciation of the crucial patterns by the model.

**Regularization Techniques:** Consequently, the Neural Network returned greater error rate than the k-NN, implying possible overfitting where the model produces very high error rates because the machine learns to predict solely based on the training dataset. This can be handled by regularization techniques such as; drop out and L2 regularization(Ghanim *et al.* 2020). Dropout as an approach involves periodically turning off neurons during the training process, defer hence formulating dependence on specific attributes. L2 regularization informsively a model to keep weights small with the ultimate intention of maintaining a general model with minimal complexity. The two methods are good for tuning a model to generalize better on unseen data and rate of overfitting hence improving prediction.

**Hyperparameter Tuning:** Tuning hyperparameters is a big step toward enhancing the part that was trained. Thus, potential improvements contain using different learning rate, batch size and activation functions. Adding the decay factor changes the rate of convergence of the model so that do not experience hassles of overshooting or slow learning(Ghanbari *et al*. 2021). Batch size relates to the stability of weight update terms and the time needed for training. Also, experimenting with the function activation like ReLU and tanh or sigmoid to predict high nonlinearity in the data is useful. Affecting these hyperparameters: mini batch size, number of iterations over the mini batch, the size of the mini batch and learning rate can boost the chances of the model to generalize from the data.

**Data Augmentation and Feature Engineering:** Perhaps increasing the size of the dataset beyond samples used in this work or incorporating additional features might be useful in training the Neural Network to have more meaningful feature representations. Other feature engineering might be defining the interaction terms, rescaling or transforming the data into some other forms, for example into logarithmic one which could contain more informative inputs that the network could learn from(Joshi *et al.* 2021). Optimization of the Neural Network can be achieved by modifying its structure, applying the method of reducing the rate of overfitting, adjusting the neural network’s parameters and increasing the quality of input data. If these strategies are applied, then the model has a chance of performing even better than at present, and thus is a more suitable candidate for this regression task..

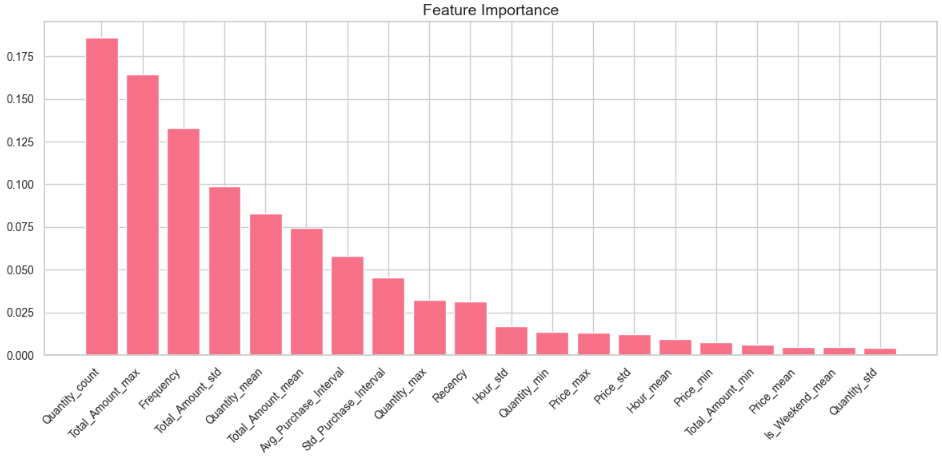
### 4.3.3 Implications of Findings

****

**Figure 11: Model Evaluation Results of XGBoost and neural network**

(Source: Self-Created)

**XGBoost for High Accuracy and Precision:** The evaluations have suggested that XGBoost is a model of choice for high accuracy and precision of any type of task. The combination of needing to deal with complex data structures, while at the same time not needing to have a high error rate makes it a perfect fit for predictive performance applications. The work was done due to the gradient boosting framework for independent decision trees that are trained in series, and each new tree is tried to minimize mistakes of the previous trees. This process makes the model capable of capturing complex, dynamic functional forms and non-linearity in the structural data. The analysis of the MSE, RMSE, MAE, and R², depends on the model, a method that evaluated the model’s predictability, shows that the medical model is more efficient and yields better results. However, it is extremely powerful in handling errors, one of the key qualities that prove XGBoost as something efficient and accurate in its performances. This makes it particularly valuable in fields such as finance, heath care and in fact in any domain where high predictive accuracy is important. Such industries where poor forecasts decisions have serious implications, this makes XGBoost stand out by providing the much needed high levels of accuracy compared to other models. As the model is being updated the fact is that it operates fast with huge amount of data and with complex relations between incoming samples that makes it the most suitable tool for solving various and rather complex predictive problems.



**Figure 12: Bar graph of Feature Importance**

(Source: Self-Created)

**Neural Networks for Flexibility and Interpretability:** In spite of their slightly inferior performance in this particular case Neural Networks are playing their part in those circumstances where interpretability or working with numerous features matters. User did not perform as accurately as XGBoost; however, Neural Networks are very useful when it comes to real complex data or a data set with higher dimensionality or deep learning. Because their ability to learn depends on hierarchical features, these are suitable in applications that demands precise structures such as image recognition and natural language (Soydaner, 2022). Neural Network obtained can find out non-linearity and are flexible in a number of ways with respect to structure of the data set. These are mostly useful in areas where other models might not perform well, due to the fact that they are more flexible especially in application that require deep feature learning. Also, most of the time hyperparameters (like learning rate, batch size) or architectural changes or regularization techniques (like dropout or L2 regularizations) can fix the problem in Neural Networks.



**Figure 13: Graph of Model Loss and MAE**

(Source: Self-Created)

Despite that interpretability might not be as easy for Neural Networks as for instance, XGBoost, it is flexibile and can model complex relationsips despite of its potential performance being slightly lower(Zhang *et al.* 2024). Neural Networks could be preferred in some applications because of the flexibility offered and the opportunity to discover more intricate structures in the data rather than clear and easily interpretable models which are preferred in many cases because of interpretation ease, in scenarios such as computer vision and natural language processing.

**Need for Thorough Model Evaluation:** The outcomes reveal a crucial necessity of the thorough examination of the models before the choice of the appropriate model. Their true advantages and performance in various applications were demonstrated by MSE, RMSE, and R², which are assigned more to the end-users are not exclusive(Shen *et al*. 2022). One more profit of cross-validation is that it might deliver a better understand in how accurate the model is in unseen data, prevent overfit and increase stability. Particularly, stability of the model throughout the time period, computational performance, and readiness to process new samples also matters. A comprehensive assessment makes it possible to state that none of the models can be deemed appropriate for the given application without proper consideration of its short-term and long-term capabilities.

## 4.4 Conclusion

In this chapter, the analysis of XGBoost and Neural Network’s results is presented, focusing on assessment criteria that consist of MSE, RMSE, MAE, and R². On comparing, it was evident that XGBoost dominated the Neural Network with relatively improved accuracy and lower error margin. Because of these issues, it is better suited to this dataset due to its enhanced capability of handling complicated relation and reduced discrepancy within the forecasts. For the same reasoning, user find that MSE, RMSE, and MAE are all lower in the case of XGBoost, proving again that it is more accurate at making predictions. While the Neural Network has the potential to solve this regression task, it has been shown to suffer in this part with overestimations and higher overall error rates. These problems raise the question, is the structure of the Neural Network which was chosen or the training algorithm that has been used appropriate. Maybe it is possible to sacrifice some additional gains made by starting from such a range by applying a more reasonable approach to hyperparameter tuning or applying the regularization methods or modifying the network architecture. Thus, the results demonstrate that much attention should be paid to the choice of the proper model for a specific problem. Although XGBoost has more accuracy compared to Neural Network, this paper found out that there is still a need for optimization in the Neural Network when it comes to regression. User pointed out that one could perform work in the future by optimizing the architectures of Neural Networks, adjusting certain hyperparameters or work on other sets of data for enhancing performance and prediction.