OPTIMIZING PARAMETERS FOR THE PREDICTION OF CREDIT RISK USING FEEDFORWARD NEURAL NETWORKS

INTRODUCTION

According to Investopedia, credit risk crystalizes in the event that a facility amount or obligation of a borrower is not met, causing the lender to lose. Credit risk is traditionally seen as the likelihood that a creditor does not receive the principal and interest from a loan extended to a borrower, culminating eventually in a cash flow challenges and increased operational cost due to loss recovery process.

Researchers have widely studied Credit risk in relation to bank lending decisions and profitability. One of the largest risk factors for banks and other financial institutions, difficult to offset, remains credit. Many improvements have been made regarding how credit is measured and dispersed across sectors to mitigate concentration, but the risk remains a big issue. As financial institutions increase in scale and complexity, and in pace of their transactions, there is a growing need for them to develop and adopt advanced techniques to manage risks and keep track of their exposures which are dynamic in nature. Fortunately, contemporary innovations in the field of information technology have made cheap the cost of data acquisition, management and analysis. These have made it possible to build financial systems that are robust and efficient at mitigating risks (Angelini, Roli and di Tollo, 2008).

For banks to be profitable, they need an accurate consumer loan default detection system. While we have witnessed tremendous improvement in the techniques of credit measurement, it is still no doubt a significant risk to financial institutions (Angelini, Roli and di Tollo, 2008).

Models that predict loan default have been developed, using analysis techniques that learn patterns about the current and historic data of the credit customer and using the information gained to predict about the credit customer ability to pay back on time (Tsai et al., 2009).

Optimizing the performance of feedforward neural network-training algorithms using a Kaggle public domain loan default dataset, to produce a loan default prediction model, and comparing the performance with those of other classifier models, including the Gaussian Naïve Bayes, Random Forest and Decision tree, is what this paper seeks to address. The original dataset has 148670 cases with each classed under 34 columns that form the attributes of the dataset. Customers will be eligible for a loan after an analysis of several parameters which include loan type, loan purpose, credit worthiness, income, credit score and others.

BACKGROUND

Attempts by several scientists to predict credit risk and maximize profits for banks and other financial institutions have proven to be successful. Several classification algorithms have been deployed and of those, neural network stands out as one of the best. A look at some of the experiments carried out by researchers and scientists in relation to credit risks, discussed in this section, produced results that are impressive, particularly for neural networks. This research will take a step towards systemically optimizing the performance of neural networks and other classifier algorithms and comparing their results.

Angelini, Roli and di Tollo, (2008) suggested that neural network is one of the best methods for the prediction of credit risk based on the result of their experiments which was effective with low error, and can be said to be state-of-the-art. The experiment performed on data of 76 small businesses from an Italian bank used a feed-forward neural network set-up in two neural architectures: a classical feedforward neural network which has the input layer, two layers within the hidden layer and an output layer of a single neuron, and an ad-hoc (special purpose) feedforward architecture of four layers feedforward network with the input neurons grouped by three where there exists a connection between each group and a neuron of the following layer.

The two networks were trained by supervised algorithm(backpropagation), which minimizes the error between desired and actual outputs by optimizing the network weight. The data used in their research was normalized and fed the network. The interval of the data is within the range of each input node by the logarithmic formula, while the choice of parameters for the network structure and algorithm was done tuning them empirically using a method analogous to the gradient ascent algorithm in the space of parameters.

I plan to deploy one architecture which is similar to the classical feedforward neural network, trained by supervised algorithm as used in the work of Angelini, Roli and di Tollo, (2008), but would differ in the normalization method and dataset and size of it. My hyperparameter optimization will be carried out by a systemic approach different from their work and would be performed in comparison with the performance of other classifiers.

OBJECTIVES

This research is aimed at predicting credit risk using a non-linear dataset from a financial institution and optimizing the parameters of different algorithms to compare their performance with that of a Feedforward Neural Network. To achieve this, the following shall be carried out.

- Models from different algorithms will be trained and tested on the dataset.
- Hyperparameter optimization using grid search will be performed on the different models and tested on the dataset to observe their performance.
- The performance of the optimized Neural Network shall be compared with those of other classifier algorithms.

METHODOLOGY

Artificial Neural Network(ANN)

Laitinen (1999) opined that artificial neural network is a nonlinear approach that has proven to be better and should be used instead of linear methods. They can be particularly useful in cases where the variables of the data being analysed are related in a complex manner (Pang, Wang and Bai, 2002). In an artificial neural network, the nonlinear relationship between inputs and outputs are modelled, and it is daunting a task to pull out information from internal system (Deepak and Shruti, 2018).

ANN (Artificial Neural networks), a machine learning system similar to biological neurons in its structure and function executes assignment by changing its parameters, in comparable manner neurons behave to execute cognitive task. ANN is a specifically structured inter-connected neurons, linked by associated weights responsible for behaviour of the ANN and determine information flow intensity (Deepak and Shruti, 2018).

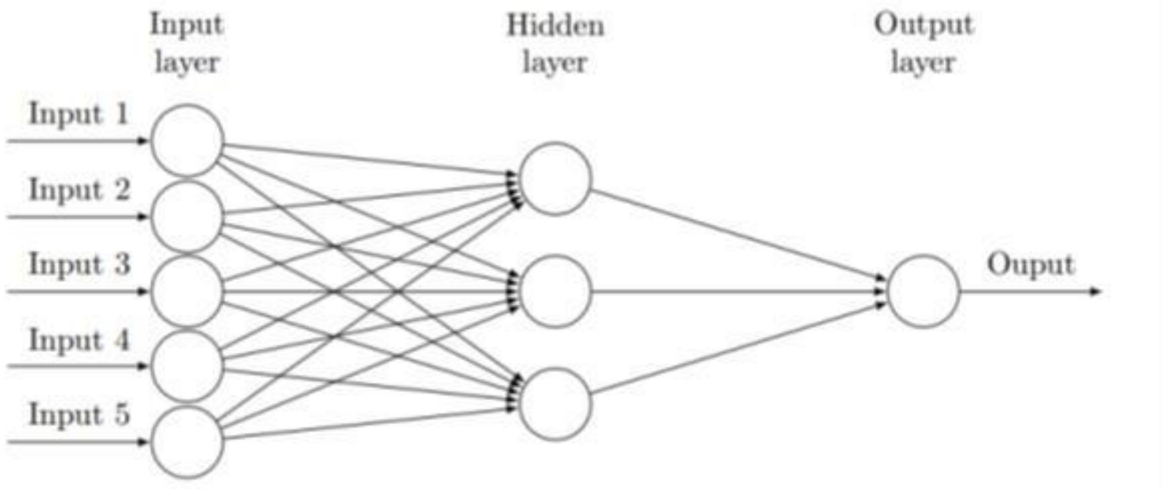

Fig.1. A Structure of Artificial Neural Network

The structure of ANN depicted in fig. 1 has three layers: input, hidden and output layers. The first layer(input) constitutes neurons responsible for stimulus reception which they pass to the second layer called the hidden layer. At the hidden layer, Information is firstly weighed and sent to the next level of layers based on the connections amongst neurons in the network. In accordance with the processing unit or a transfer function depicted in below figure (fig2), the information will then be sized.

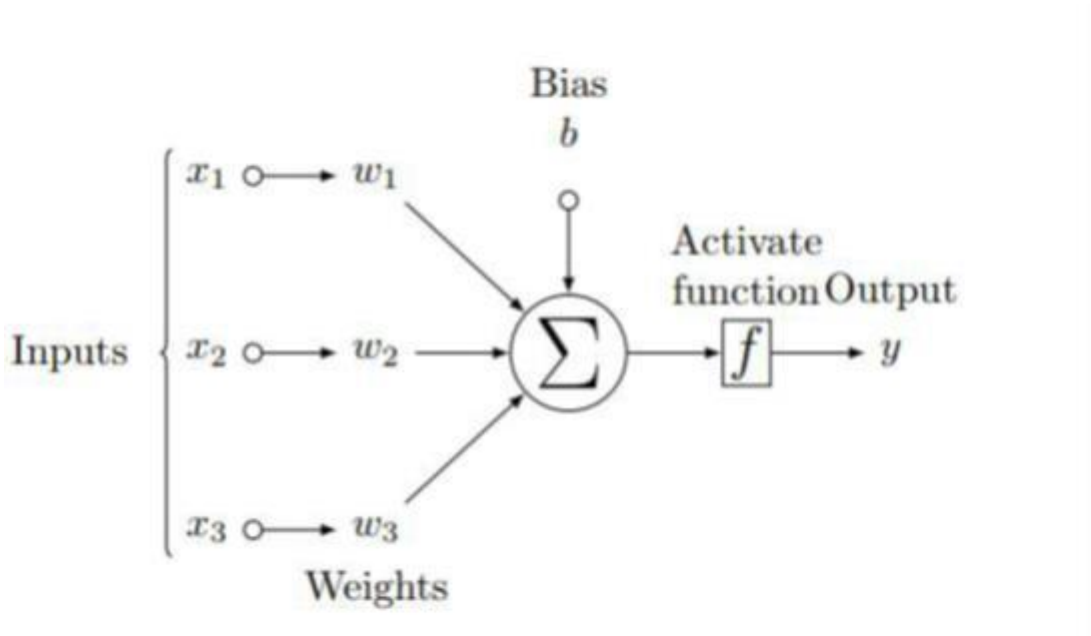

Fig.2. Mathematical Equation of Artificial Neural Network

There is a least value that activates a neuron called the threshold value and this performs the function of characterizing the neuron along with a transfer function. The summing of input neurons is performed by the hidden layer with several possible layers, which multiplies the weights with the summation to generate output neurons. There are two steps involved in producing the out. Firstly, The net multiplies input by weight on relevant connection, and then adds up the results. Secondly, the net activates the summation of the inputs by the activation function(Atiya, 2001).

$$Yi = \Sigma Wj,ij \in Iaj \text{ .... } (1)$$
$$ai = g(y) \text{.....................} (2)$$

A training collection of accurate examples is used to train the network model in supervised learning. The training set has in it pairs of many inputs and expected outputs. The neural network performs an action of tuning weights as errors are generated in it (Angelini, Roli and di Tollo, 2008). Classification is a common form of supervised learning, in which the network must learn to generalise relationships between corresponding input and output variables. The typical classification problem to predict credit default will be addressed in this study.

About Data Source

The data source is kaggle.com (loan default data) and this data has over 148 thousand records. 40 thousand records were randomly selected from the data and appropriate attributes were selected from 34 features for this experiment. The 32 selected attributes were subjected to dimensionality, reducing the dimension to 8. The following are the variables that make up the data set:

Dependent Variable**:** Status (0 and 1); This paper seeks to predict whether a borrower qualifies for a loan or not, in the near future. Here, a bad loan means that the borrower will default while a performing loan means that the borrower will not default but will be able to repay the full loan amount. The two classes predicted by the network are 0 which indicates borrower will default and 1 indicating that borrower will not default. Independent Variable: Following variables are some of the independent variables in the data: loan_amount, funded amnt; the requested amount by the borrower, funded_amnt_inv; amount approved and offered by the investors, term; the life cycle of the loan, rate_of_interest; interest rate on which loan has been given, instalment; periodic repayment amount, Credit_Score; Credit rating assigned to the loan application, emp_length; the borrower's duration of employment, income; the annual amount the borrower earns, issue_d; the date loan was granted, application_type; whether the application is individual or joint.

Model

The experiment in this study was designed to be carried out in two stages. In the first stage, default parameters were used to build models of three classifiers including Gaussian Naïve Bayes(GNB), Decision Tree(DT) and Random Forest(RF), and a Neural Network. I have used a classic feed-forward neural network in this work. The feedforward network consists of an input layer with data of 8 dimensions, 2 hidden layers and an output layer with one neuron that represents a classifier. Sigmoid function was used across the network to learn complex patterns, a dropout of 20, 20 and 10% were used. The input layer had 1000 units with 50 and 35 units for the first and second hidden layers respectively while a binary cross entropy loss function and an SGD optimizer with a learning rate of 0.01 and 15 epochs were specified. Supervised learning algorithm (back propagation algorithm) is used to train the network (Hornik, Stinchcombe, and White, 1989).

The algorithm optimizes the neuron weights by minimizing the error between actual and desired output. Error is $errori = Di - Ai$ for neuron i. Weights will be updated by formula $Wi,k = Wi,k + \phi Ikerrorig'(ini)$ where φ is the learning coefficient and Ik is the output from hidden layer. Algorithm will work until a stopping criterion is found (Pineda, 1987).

The second stage of the experiment involved hyperparameter optimization using grid search. Here, i specified a range of parameters for all the classifiers used in this experiment to compare their performance. The setup of the two stages of the experiment is represented in the tables below.

STAGE 1:

A, BASE MODEL(ANN)

INPUT LAYER	SHAPE: **8**
DENSE LAYER	NODES: **1000**, ACTIVATION: **SIGMOID**
DROPOUT LAYER	RATE: **0.2**
DENSE LAYER 2	NODES: **50 ,** ACTIVATION: **SIGMOID**
DROPOUT LAYER 2	RATE: **0.2**
DENSE LAYER 3	NODES: **35,** ACTIVATION: **SIGMOID**
DROPOUT LAYER 3	RATE: **0.1**
OUTPUT LAYER	NODE: **1**

Table 1: Baseline Model

B, OTHER CLASSIFIERS MODEL

GAUSSIAN NAIVE BAYES CLASSIFIER(GNB)	DEFAULT PARAMETERS
DECISION TREE CLASSIFIER	DEFAULT PARAMETERS
RANDOM FOREST CLASSIFIER	DEFAULT PARAMETERS

Table 2: Other Classifiers Model

STAGE 2:

A, MANUAL OPTIMIZATION OF ANN

Here, a manual approach of hyperparameter tuning was experimented for the Neural Networks to improve on the performance from the base model and this model was named 'model1'. The architecture of the model is tabulated in table 3 below.

INPUT LAYER	SHAPE: **8**
DENSE LAYER	NODES: **8**, ACTIVATION: **LEAKY RELU**
DENSE LAYER 2	NODES: **35 ,** ACTIVATION: **LEAKY RELU**
DENSE LAYER 2	NODES: **35 ,** ACTIVATION: **LEAKY RELU**
DENSE LAYER 3	NODES: **35,** ACTIVATION: **LEAKY RELU**
OUTPUT LAYER	NODES: **1,** ACTIVATION: **SIGMOID**
LOSS FUNCTION	BINARY CROSS ENTROPY
OPTIMIZER	ADAM
LEARNING RATE	DEFAULT: 0.001
EPOCHS	20

Table 3: Model 1: parameters arrived at by manually tweaking the hyperparameters.

B, SYSTEMATIC METHOD OF HYPERPARAMETER TUNING OF ANN AND THE OTHER TWO BEST PERFORMING CLASSIFIERS (DECISION TREE AND RANDOM FOREST).

The systematic approach employed in tuning hyperparameter in this research to optimize the performance of models is the grid search cv built and wrapped in KerasClassifier. The table 4 below shows a range of parameters and values specified for the experiment.

MODEL	OPTIMIZED PARAMETERS	
ANN	LEARNING RATE	0.001, 0.01
	ACTIVATION FUNCTION	'elu', 'sigmoid', 'leaky_relu'
	INITIALIZER	'uniform', 'normal', 'zero'
	LAYER1 NEURON	10, 20, 30, 40
	LAYER2 NEURON	10, 20, 30
	LAYER3 NEURON	10, 20
	LAYER4 NEURON	10
RF	N_ESTIMATORS	100, 300, 500
	MIN SAMPLE LEAF	1, 4, 12, 20, 40
DT	CRITERION	'gini', 'entropy'
	MAXIMUM DEPTH	np.arange(1, 100)
	MIN SAMPLE LEAF	1, 4, 12, 20, 40

Table 4: A range of parameters and values for grid search cv experiment

RESULT

BASELINE MODEL

The result of the baseline model is represented in table 5 below, and the chart below also shows the performance of the model. It can be seen that the model learned poorly and hence the reason for poor validation results.

MODEL	TRAINING ACCURACY	VALIDATION ACCURACY	TRAINING LOSS	VALIDATION LOSS
BASELINE ANN	0.51	0.50	0.69	0.69

Table 5: Results of Baseline Model

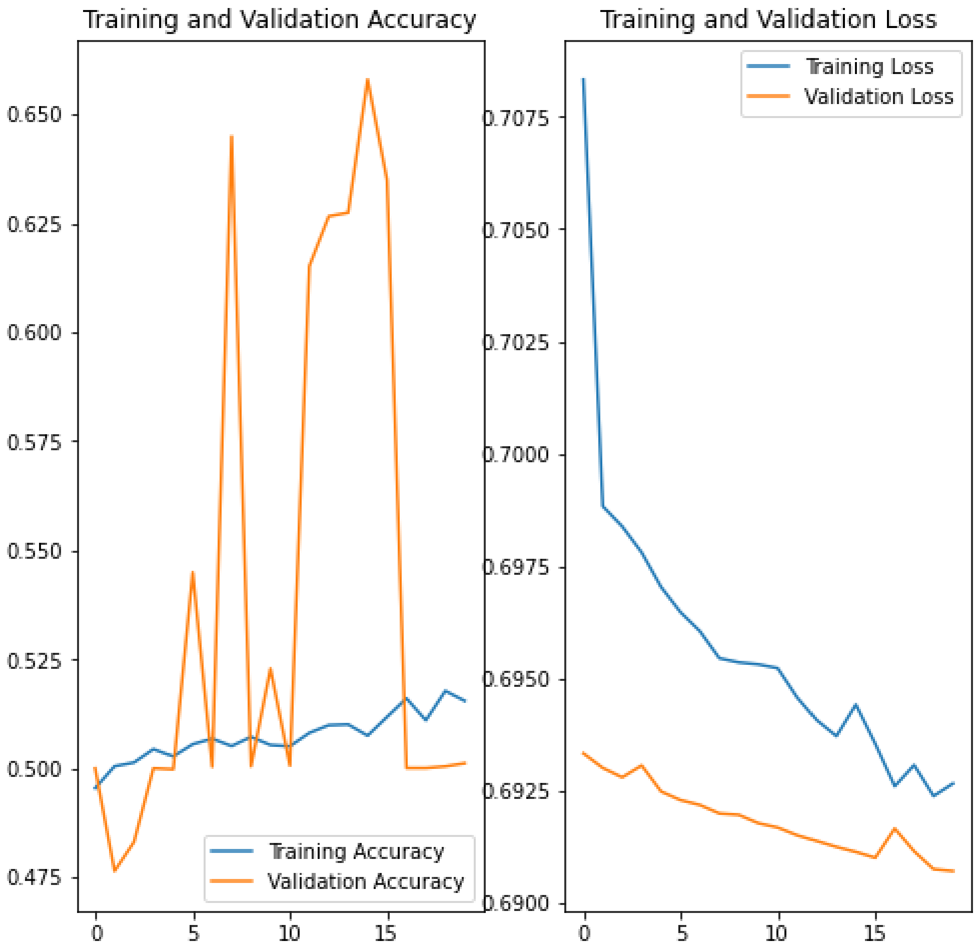

Fig 3: Training and validation accuracies of the base model.

OTHER CLASSIFIER MODELS WITH DEFAULT PARAMETERS

MODEL	TRAINING ACCURACY	TEST ACCURACY
GAUSSIAN NAIVE BAYES CLASSIFIER(GNB)	0.50	0.50
DECISION TREE CLASSIFIER	0.84	0.80
RANDOM FOREST CLASSIFIER	0.79	0.78

Table 6: Scores of the other classifier models using default parameters

The above result tabulated (Table 6) is obtained from training and testing the dataset on classifiers other than the neural networks. These experiments were performed using the default parameters of the algorithms. From the select classifiers tested, it can be seen that decision tree performed best with the dataset while Gaussian Naïve Bayes scored lowest. The poor performance of NB classifier could be for a number of factors including the high volume of data tested in the experiment.

MANUAL OPTIMIZATION OF NEURAL NETWORKS

MODEL	TRAINING ACCURACY	VALIDATION ACCURACY	TRAINING LOSS	VALIDATION LOSS	VALIDATION MSE
MODEL 1	0.87	0.87	0.28	0.29	0.08

Table 7: Results from manual tweaking of ANN

The result from model 1 tabulated above (Table 7) is obtained from an experiment with the neural network where hyperparameters were tweaked manually over time to obtain a good result. Below in Fig 4 is the plot of the training and validation results over the epochs of training and validation. This shows a gradual rise in both training and validation accuracies and gradual decline in both training and validation losses. A heat map of the confusion matrix is also presented below in Fig 5 which classifies the prediction of the model. The model has performed fairly good, predicting majority of the outcomes right as represented in the true positive and true negative regions of the plot.

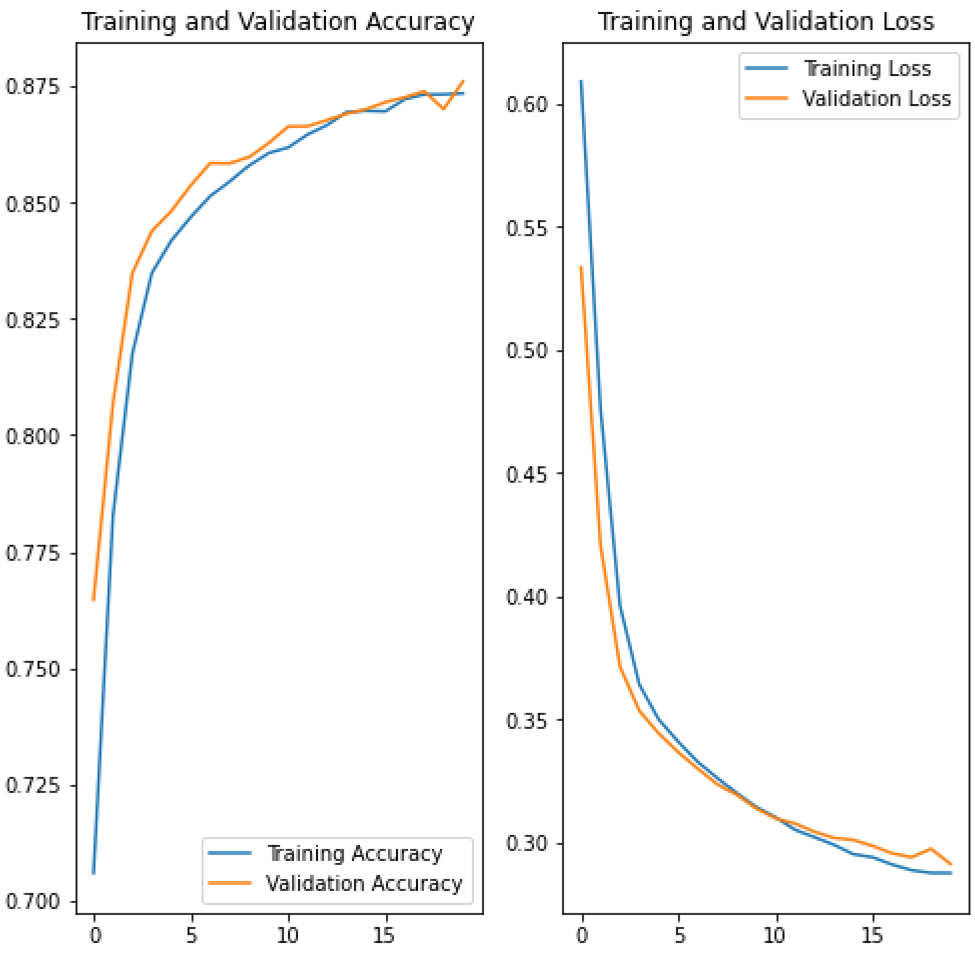

Fig 4: Training and validation accuracies of model 1

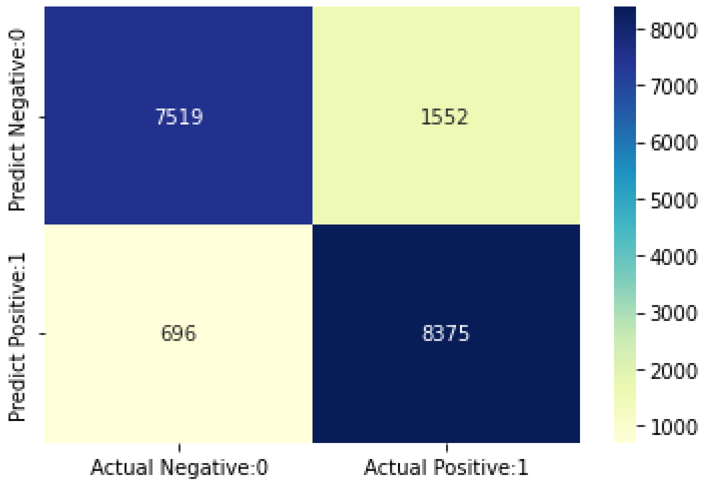

Fig 5: Confusion Matrix from the performance of model 1

MODEL	TESTED PARAMETERS		BEST PARAMETERS
ANN	LEARNING RATE	0.001, 0.01	0.01
	ACTIVATION FUNCTION	'elu', 'sigmoid', 'leaky_relu'	Leaky_relu
	INITIALIZER	'uniform', 'normal', 'zero'	normal
	LAYER1 NEURON	10, 20, 30, 40	40
	LAYER2 NEURON	10, 20, 30	20
	LAYER3 NEURON	10, 20	10
	LAYER4 NEURON	10	10
RF	N_ESTIMATORS	100, 300, 500	500
	MIN SAMPLE LEAF	1, 4, 12, 20, 40	1
DT	CRITERION	'gini', 'entropy'	gini
	MAXIMUM DEPTH	np.arange(1, 100)	18
	MIN SAMPLE LEAF	1, 4, 12, 20, 40	1

Table 8: Results from grid search hyperparameter tuning showing the best parameters

RESULT COMPARISM

OPTIMIZED MODEL	TRAINING ACCURACY	VALIDATION ACCURACY
FEED FORWARD NEURAL NETWORK (FFNN)	0.88	0.88
RANDOM FOREST	1.0	0.88
DECISION TREE	0.95	0.82

Table 9: Results from the optimized models

From the result of the test performed on the optimized models (Table 9), it can be seen that FFNN and Random forest performed best with 88% test accuracies while Decision Tree scored 82%.

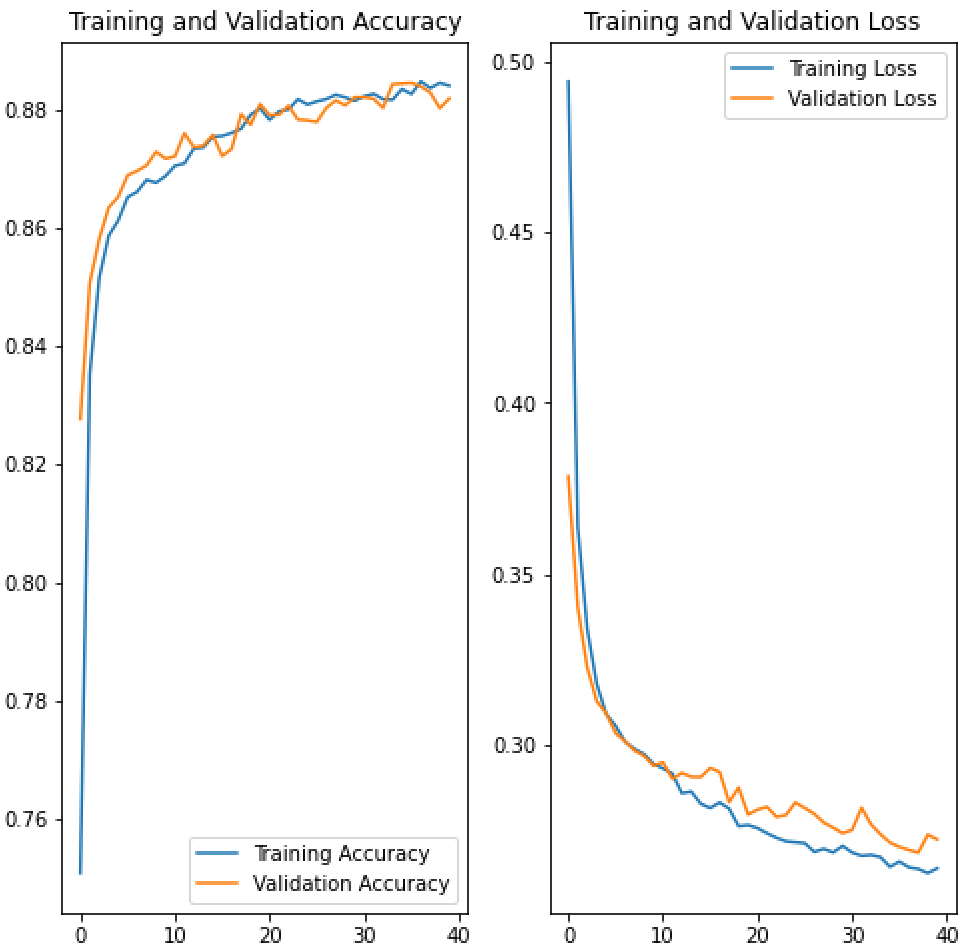

Fig 6: Optimized FFNN Model Plot

CONCLUSION

From the studies, it is evident that classification model performance can be optimized by both manual and systematic methods. The results from the experiments show that grid search cv, a systematic approach towards parameter tuning produced parameters that performed on a select classification algorithm better than the manual ways of tuning parameters. More also, the process is faster as it took a long time to arrive at the best combinations of parameters required using the manual method of parameter tweaking, this in the long run speaks to economics as time is very valuable.

Comparing the performance of the models, it can be observed that Neural Networks performed better than the other classification algorithms using the grid search cv. It was however observed that Decision Tree and Random Forest learned better but predicted poorer than the Neural Network.

Conclusively, the general performance of these models is impressive and has proven to be helpful in mitigating credit risk.

REFERENCES

1. Investopedia (2022) Credit Risk. Available at: https://www.investopedia.com/terms/c/creditrisk.asp (Accessed: 2 April 2022).
2. Angelini, E., di Tollo, G. and Roli, A. (2008) “A neural network approach for credit risk evaluation”, *The quarterly review of economics and finance*, 48(4), pp.733-755.
3. Tsai M., Lin S., Cheng C., and Lin Y. (2009) "The consumer loan default predicting model-An application of DEA-DA and neural network", *elsevier, Expert Systems With Applications*, vol.36, no.9, pp.11682-11690.
4. Laitinen, E.K. (1999) “Predicting a corporate credit analyst's risk estimate by logistic and linear models”, *International review of financial analysis*, 8(2), pp.97-121.
5. Pang, S.L., Wang, Y.M. and Bai, Y.H., (2002) Credit scoring model based on neural network. In Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on (Vol. 4, pp. 1742-1746). IEEE.
6. Deepak, K. G. and Shruti, G. (2018) "Credit Risk Prediction Using Artificial Neural Network Algorithm", *International Journal of Modern Education and Computer Science(IJMECS)*, Vol.10, No.5, pp. 9-16, 2018.DOI: 10.5815/ijmecs.2018.05.02.
7. Atiya, A.F. (2001) “Bankruptcy prediction for credit risk using neural networks: A survey and new results”, *IEEE Transactions on neural networks*, 12(4), pp.929-935.
8. Pineda, F.J. (1987) “Generalization of back-propagation to recurrent neural networks” *Physical review letters*, 59(19), p.2229.