Deep Learning: Churn Model of Customers

Yaozhuo Wang

251009638

ywan3223@uwo.ca

Abstract — Companies do not want to lose valuable customers. As a company scales to a large size, it will choose a more defensive strategy and focus on retaining existing customers. There will always be customers who are dissatisfied and decide not to buy the company's products. The company needs to calculate the dataset to obtain a churn model that will reduce the company's losses due to churn and provide the company with a profit margin. Therefore, the report will use Feedforward Neural Network (FFNN) as the network architecture to set up a churn model. The model will be tunned with the grid search. The result will be evaluated with cross validation.

Keywords — Churn model, predictive model, Forecasting, Feedforward Neural Network, FFNN, customers

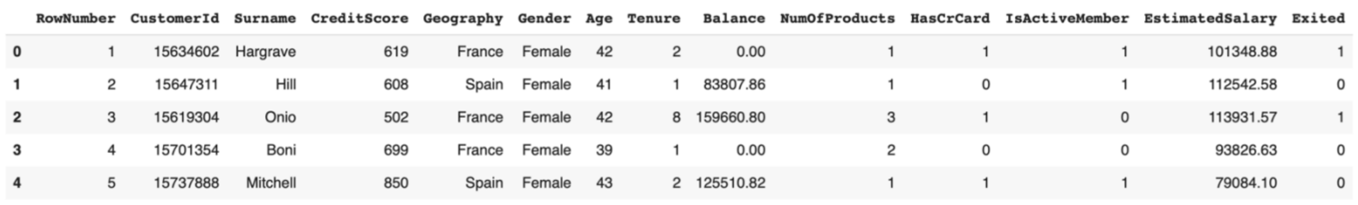
1. Problem Description

The churn model is a predictive model that estimates the propensity of a customer to churn at the individual customer level. It allows us to know how high the risk of customer churn is. Technically, a churn model is a binary classifier that divides customers into two groups. One group is the customer who have churned, and another group is those who have not. In addition, the model usually tells us the probability that a customer belongs to a certain group.

Customer retention strategy is a major concern in many industries, such as telecommunications, banking, insurance, retail, etc. To keep customers requires an in-depth and comprehensive analysis of issues related to customer churn. The bank has earlier proposed marketing management models such as customer relationship management and relationship marketing. at the same time, academia and business community has also promoted research related to customer churn behavior.[1] When the bank has an estimate of the risk of customers' churn through the churn model, they can remedy the high-risk group. Typically, the bank will reach out to these customers through a marketing event, reminding them that they haven't purchased from the company in a long time and offering them a discount. This strategy will boost the spending of regular customers to some extent.

2. Data Description

In this paper, the dataset is the details of customers in a bank from Kaggle website (Link: https://www.kaggle.com/datasets/shubh0799/churn-modelling).[2] The dataset includes 10000 rows and 14 columns with 140000 data. It displays the basic information of the bank customers, including their surname, country, credit score, gender, age, and estimated salary. The dataset also has data on the relationship between customers and the bank. It includes the time bond they have with the bank, the products they own, if they have a credit card, if they are still an active member of the bank, and the amount they left in the bank. First figure shows the first five rows of the dataset. As a new dataset used different from the assignment1, there is no new features added.



	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

Fig. 1. First five row of the dataset

A simple analysis of the data through histograms gives us the distribution of some of churned customers and active members. Most consumers are from France, but most of those who existed were from Germany, maybe due to a lack of resources and few clients. Additionally, a higher percentage of male client's turnover than female customers. Most clients have tenures of 1 to 9, and there is a high rate of churning within these ranges. Most customers only purchase 1 or 2 things, and most churned customers only purchased 1 product. Customers that use credit cards make up most churning customers. Unsurprisingly, the turnover rate for inactive members is higher, and the proportion of inactive members overall is relatively high.

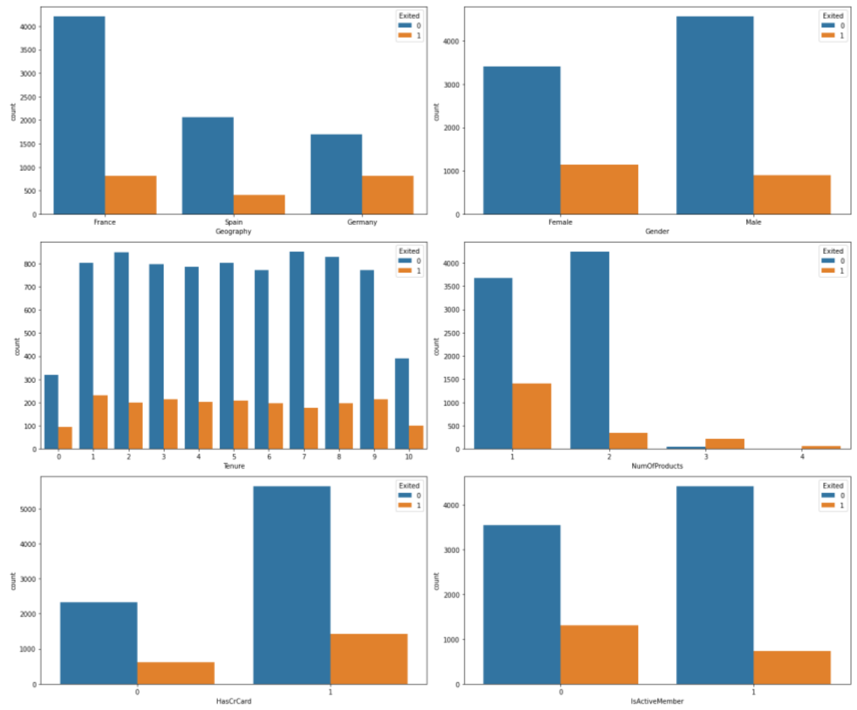


Fig. 2. Histogram of data seperated by features

3. Background - Feedforward Neural Network (FFNN)

3.1 Overview

Feedforward neural network is a type of artificial neural network in which there is no cycle in the connections between the nodes. Since input is only processed in one direction, the feed forward model is the simplest type of neural network. FFNN has two types of architecture with single layer and multiple layers. A single-layer neural network can compute a continuous output with logistic function instead of a step function.

$$f(x) = \frac{1}{1+e^{-x}}$$

The logistic function is one of the family of functions called sigmoid functions. With the continuous derivative, it is allowing to be used in backpropagation.

$$f'(x) = f(x)(1 - f(x))$$

Multi-layer perceptron network consists of multiple layers of computational units, usually interconnected in a feed-forward way. The output values are compared with the correct answer to compute the value of some predefined error-function. To slightly lower the value of the error function, the error is transmitted back across the network, and the algorithm modifies the weights of each connection. The network will typically converge to a state where the computation error is modest after repeating this method for enough training cycles.[3]

In this paper, we use the multi-layer Feedforward Neural Network to solve the churn modelling problem. As a multilayer feedforward network based on the combination of gradient descent and Newton's method. LM neural network has fewer iterations, faster convergence, and higher accuracy. The customer churn prediction model is a nonlinear model. When our data is not linearly separable linear model faces the problem of approximation, neural networks can easily make predictions to solve the problem. The hidden layer is used to add nonlinearity and change the representation of the data for better generalization of the function.[4] And there is no images in dataset with only numbers, FFNN should be the best choice.

3.2 Hyperparameters

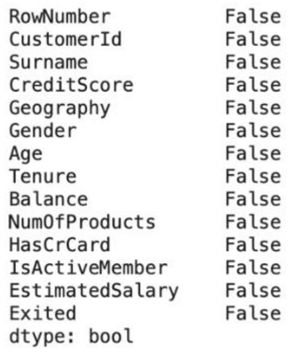
Feedforward Neural Network includes 11 hyperparameters which are learning rate, regularization parameter, number of layers, neuron of hidden layers, epoch, batch size, output neurons, cost function, weight initialization, activation, size of train data.[2] These can affect the learning speed and the final classification result of the neural network, where the learning speed of the neural network is mainly related to how fast the cost function decreases on the training set, and the final classification result is mainly related to the correct classification rate on the validation set. The regularization parameter mainly affects the classification rate of the neural network; Learning rate and cost function mostly affect the learning speed, which is reflected in the decreasing speed of the cost function curve of the training data; epoch and batch size impact the classification rate of the model and the training time.[5]

In this paper, the number of layers, neuron, batch size and optimizer are choosing to be optimized to affect the accuracy of classification. The actual gradient is more closely approximated by the bigger batch sizes. Therefore, large batch size is more likely to overfit the data or get stuck in the local optimum. Simultaneously, an optimizer is a function that modifies the attributes of the neural network and helps in reducing the overall loss and improve the accuracy. The influence of hyperparameters only indicates the primary influence on the neural network, and it does not mean that the hyperparameters only affect the learning speed or the accuracy.

4. Methodology

4.1 Data Pre-processing

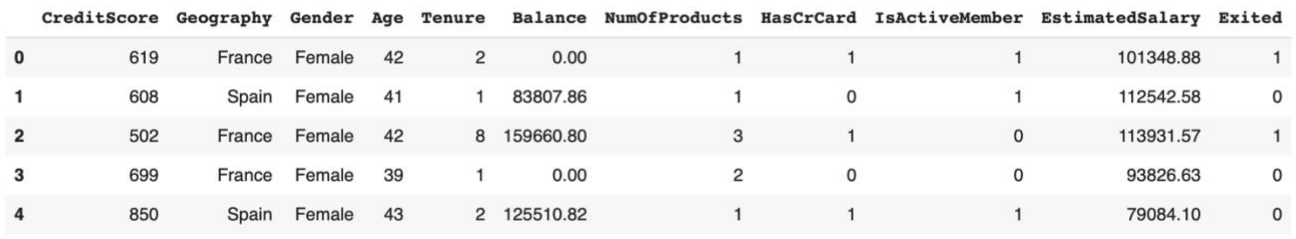
Checking Nan values of the dataset as the first step of pre-processing. And there are no Nan values in the dataset shows as the figure 2. Through a brief analysis of the questions and data shows in figure 3, some useless features need to be deleted which are Row Number, Customer id, and Surname. The total column



```
RowNumber          False
CustomerId         False
Surname            False
CreditScore        False
Geography          False
Gender             False
Age                False
Tenure             False
Balance            False
NumOfProducts      False
HasCrCard          False
IsActiveMember     False
EstimatedSalary    False
Exited             False
dtype: bool
```

Fig. 3. Nan values of the datase

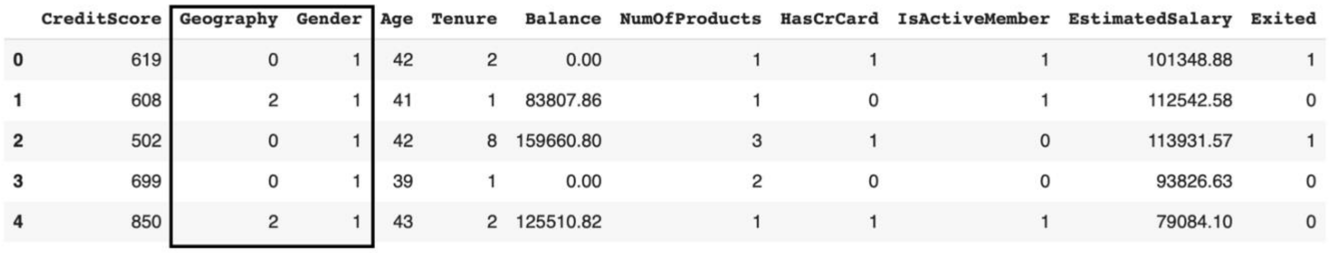
of dataset is 10 after dropping the useless data. As a prediction model that determines the probability whether a customer would discontinue doing business with the company, churn prediction model does not require the analysis of unique information data for everyone. The useless data will reduce the accuracy of model.



	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

Fig. 4. Data format of the dataset after encoding

As shown in the dataset, the features Country and gender represent observations in categorical format. Such observations should be transformed into numerical form to be utilized as appropriate input for the underlying machine learning algorithms. Assign a sequential number to each category in a simple conversion using the pandas library in python.



	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	619	0	1	42	2	0.00	1	1	1	101348.88	1
1	608	2	1	41	1	83807.86	1	0	1	112542.58	0
2	502	0	1	42	8	159660.80	3	1	0	113931.57	1
3	699	0	1	39	1	0.00	2	0	0	93826.63	0
4	850	2	1	43	2	125510.82	1	1	1	79084.10	0

Fig. 5. Data format of the dataset after encoding

As a quick utility that wraps input validation into a single call in a one-liner. A common need for machine learning estimators is the standardization of a dataset. The input layers of all neural networks must be standardized. It is necessary to normalize the dataset because the different size of the columns will affect the comparison.

4.2 Train and Test

After dataset has been preprocessed completely, the FFNN need to be built. The original FFNN model is combined with an input layer, a hidden layer, a dropout layer, a batch normalization layer and an output layer with sequential model. A sequential model is sufficient for a simple stack of layers where each layer has one input tensor and one output tensor.[6]

The input layer is a densely connected neural network layer with 10 units and 10 input shape. The number of neurons(unit) and input shape in the input layer is equal to the number of input variables in the data to be processed which is 10 features of dataset.

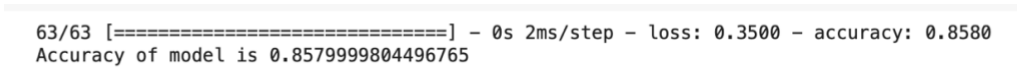
The first hidden layer is also a dense layer with unit, kernel, and activation function. Activation is the element-wise activation function supplies as the activation argument.[6] It plays a crucial role by introducing nonlinearity in neural networks. It allows neural networks to develop sophisticated representations and functions according to the inputs. The ReLU function used for the activation function is simple in composition and easy to implement. Compared with other activation functions, it can greatly improve the training speed and reduce the running time of the algorithm. Kernel is a weights matrix produced by the layer. Using normal initializing kernel weights in the model with appropriate values will avoid divergence or slowdown in neural network training.

Drop out layer is added after the hidden layer with a rate of 0.1. It introduced to implement the concept of inverted dropout which is a regularization technique to prevent high variance. Batch normalization layer is a layer that normalizes its inputs. The layer uses the mean and standard deviation of the most recent batch of inputs to normalize its output during training. [6]

The output layer is the dense layer with unit and activation parameter. The number of neurons in the output layer is equal to the number of outputs associated with each input which is 2. In the binary classification problem, sigmoid function is used as activation function in the last layer. Sigmoid has an output range of (0, 1) to use in binary classification tasks and is available to use in the last layer of the network to predict the probability of something existing.[6]

Configuring the model for training using compile method with the arguments of optimizer, loss, and metrics. RMSprop optimizer is used in the original model to keep the gradient divided by the root of the moving average of the square of gradients to ensure not update on variable slices and their accumulators.[6] Loss function is one of the two parameters necessary to compile a model. Computing the binary cross entropy loss between true values and predict values and evaluate the models with the metrics.

Keras uses NumPy arrays as the data type for input data and labels. The fit function is generally used to train the model. The training set is input, then batch size determines the size of training data, and epochs is the number of training sessions. validation spilt validates the dataset. And the accuracy of the original model is 85.8% as shown in the figure 5. To test data, use predict function to predict the value based on inputting x value and then compare with the test value of y to see the accuracy score.



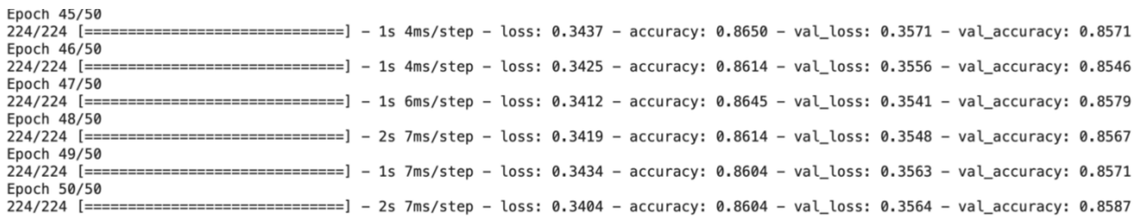
```
63/63 [==============================] - 0s 2ms/step - loss: 0.3500 - accuracy: 0.8580
Accuracy of model is 0.8579999804496765
```

Fig. 6. Accuracy of the original model

4.3 Tuning: Grid Search

To increase the accuracy of the feedforward neural network model, number of layers, number of neurons, optimizer and batch size are tuned in this paper.

To change the number of layers and neurons, four hidden layers added into the model which are same as the first hidden layer with 10, 8, 6, 4, 2 neurons. Setting gradually decreasing neurons in hidden layers can help improve the accuracy of the model. Increasing the number of hidden layers can reduce the network error and improve the accuracy. As the complexity of the network increases, it increases the training time and the tendency of overfitting.[7] To avoid overfitting happening in the model, a dropout layer and batch normalization layer are added after third hidden layer. Usually, it is sufficient to use the same number of neurons for all hidden layers with small size of train dataset. The picture below shows the last five lines of the tunned model processing.

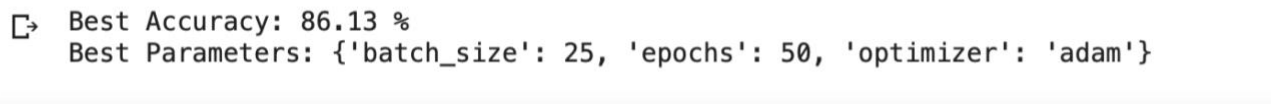


```
Epoch 45/50
224/224 [==============================] - 1s 4ms/step - loss: 0.3437 - accuracy: 0.8650 - val_loss: 0.3571 - val_accuracy: 0.8571
Epoch 46/50
224/224 [==============================] - 1s 4ms/step - loss: 0.3425 - accuracy: 0.8614 - val_loss: 0.3556 - val_accuracy: 0.8546
Epoch 47/50
224/224 [==============================] - 1s 6ms/step - loss: 0.3412 - accuracy: 0.8645 - val_loss: 0.3541 - val_accuracy: 0.8579
Epoch 48/50
224/224 [==============================] - 2s 7ms/step - loss: 0.3419 - accuracy: 0.8614 - val_loss: 0.3548 - val_accuracy: 0.8567
Epoch 49/50
224/224 [==============================] - 1s 7ms/step - loss: 0.3434 - accuracy: 0.8604 - val_loss: 0.3563 - val_accuracy: 0.8571
Epoch 50/50
224/224 [==============================] - 2s 7ms/step - loss: 0.3404 - accuracy: 0.8604 - val_loss: 0.3564 - val_accuracy: 0.8587
```

Fig. 7.   Tunned model

Grid Search is a tuning hyperparameter with Brute-force search. The computational complexity of the grid search grows exponentially as the number of hyperparameters grows. The principle is the same as finding the maximum value in an array.[8] The grid search is used in this report to list a smaller domain of hyperparameter values of batch size and optimizer. The batch size has two parameters of 25 and 32. The optimizer has two parameters of adam algorithm and rmsprop algorithm. Adam handles noisy samples and sparse gradients well compared with rmsprop. The grid search algorithm uses each set of hyperparameters to train the model and selects the combination of hyperparameters with the lowest validation set error. To get the more accurate results, the batch size should change from 32 to25, epochs should keep in 50, and optimizer should be change from rmsprop to adam.

Using the hyperparameter calculated by the grid search algorithm, compile the model with new hyperparameters. And after tunning, the accuracy has improved to 86.13% with the suitable parameters shows in the figure 8. The final step is to test data, use predict function to predict the value based on inputting x value and then compare with the test value of y to see the accuracy score.



```
Best Accuracy: 86.13 %
Best Parameters: {'batch_size': 25, 'epochs': 50, 'optimizer': 'adam'}
```

Fig. 8. Accuracy and hyperparameters after tunning

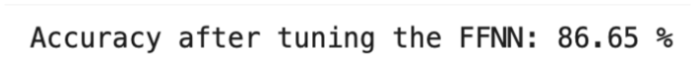
4.4 Evaluation: Cross validation and Accuracy

To evaluate the effect of models, Cross-validation is a statistical technique used to evaluate the ability of machine learning models. It is frequently used for a certain predictive modelling issue.[9] It produces skill estimates that often have a lower bias than other methods and is simple to comprehend and apply.

And there are few metrics can be used to evaluate the model. Accuracy is calculated by the accuracy_socre function. It compares the predicted result and the corresponding data in validation set to get the accuracy. The accuracy always improves with a higher score. The precision is the ratio $\frac{TP}{TP+FP}$ where tp is the number of true positives and fp the number of false positives. The precision means the classifier's accuracy is its capacity to avoid classifying a negative sample as positive. The higher value between 0 to 1, the precision is better. Recall is metric tries to determine what percentage of real positives was correctly identified with the equation of $recall = \frac{TP}{TP+FN}$. F1-score is used to measure a model's accuracy in a dataset. Additionally, it is used to assess binary classification schemes that label samples as positive side or negative side. All the accurate data of metrics will be displayed in the table. [10]

	Accuracy	Precision	recall	F1-score
Original model	0.83	0.86	0.96	0.92
Tunned model	0.86	0.88	0.97	0.92

After tunning the model, the most suitable parameter has been found. Update the new parameter into the model and create the new model as a function after changing the number of hidden layers according to the process of tuning. The model is evaluated using the KerasClassifier method with 5 hold validation is used to obtain more accurate results. As the figure9 shows, the accuracy of the model is 86.65% which has the higher accuracy than the original model.



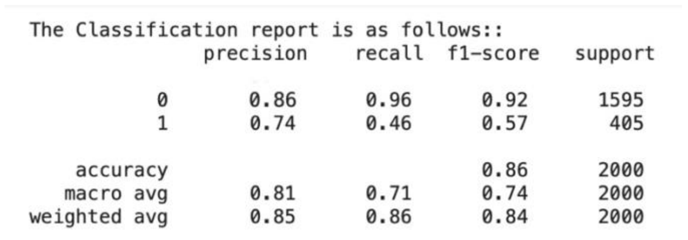
```
Accuracy after tuning the FFNN: 86.65 %
```

Fig. 9. accuracy with cross validation

5. Results

Four pictures below show the classification report with precision, recall, f1-score accuracy and confusion matrix of the original model and tunned model. The Confusion Matrix uses a table structure to display the results of classification judgments. The first row represents the result of Positive prediction, the second row represents the result of Negative prediction; the first column represents the result of True and second column represents the instance of False. Thus, it shows four results of TP, FP, TN, FN. [11]

From the confusion matrix, there is a small increase in the number of actual leaving customers and actual active customers being identified. This indicates that the accuracy of the data has improved to some extent. This indicates that the accuracy of the model has improved to some extent after tunning. However, there is no significant change in the recall and f1-score data, which indicates that there is still some room for improvement in the model.



```
The Classification report is as follows::
              precision    recall  f1-score   support

           0       0.86      0.96      0.92      1595
           1       0.74      0.46      0.57       405

    accuracy                           0.86      2000
   macro avg       0.81      0.71      0.74      2000
weighted avg       0.85      0.86      0.84      2000
```

Fig. 10. Classification of the original model

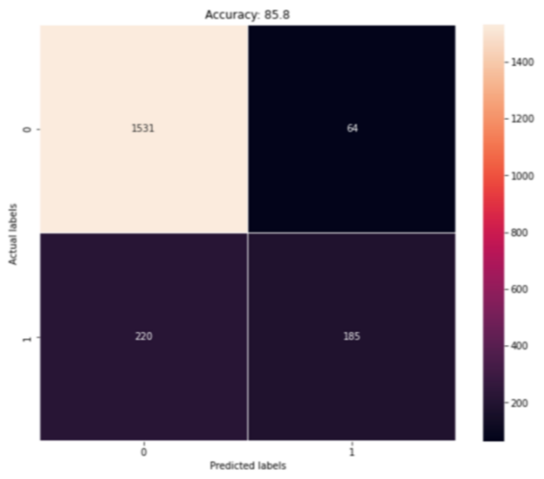
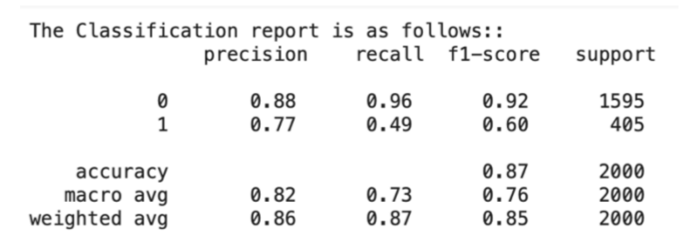


Fig. 11. Confusion Matix of original model



```
The Classification report is as follows::
              precision    recall  f1-score   support

           0       0.88      0.96      0.92      1595
           1       0.77      0.49      0.60       405

    accuracy                           0.87      2000
   macro avg       0.82      0.73      0.76      2000
weighted avg       0.86      0.87      0.85      2000
```

Fig. 11. Classification report of the tunned model

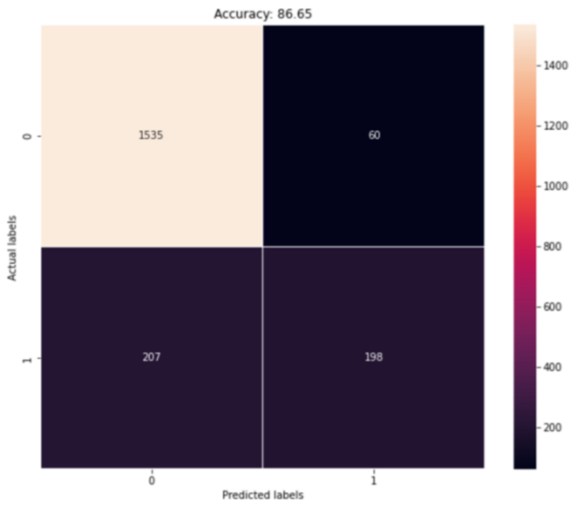
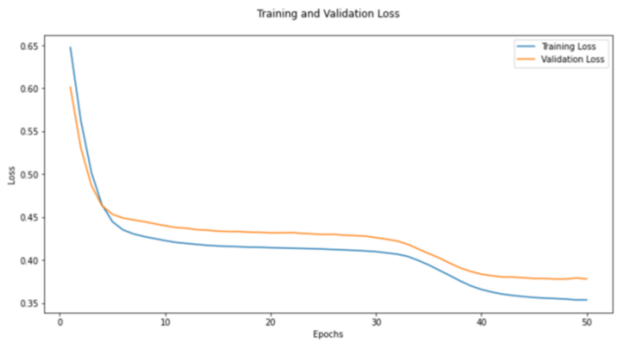
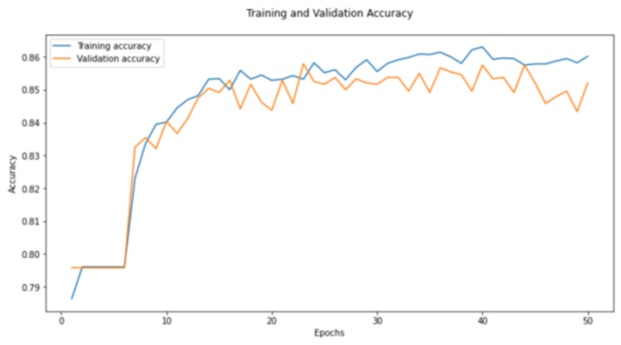


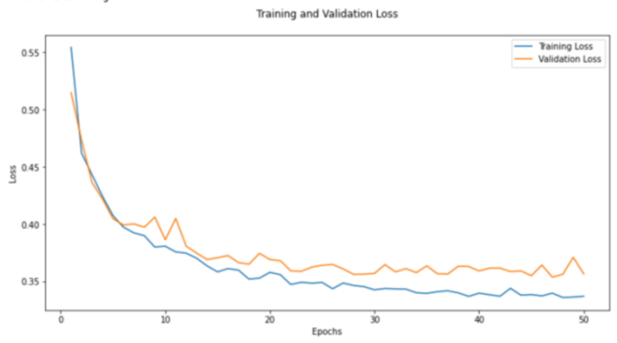
Fig. 12. Confusion Matix of tunned model

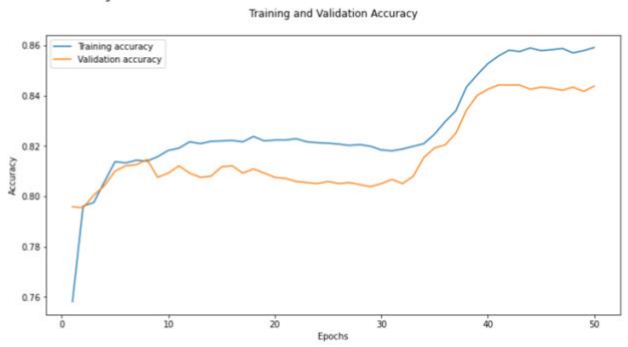
The first two pictures below are validation loss and validation accuracy of original model. The second two picture shows the validation loss and validation accuracy of the tunned model. Combining the accuracy and line graphs on the previous page to analyze, pictures shows that validation loss always higher than training loss. Validation accuracy of original model is lower than training accuracy , but validation accuracy of tunned model is higher than training accuracy. This indicates that the accuracy obtained by the tunned model is higher than that of the original model. However, due to the increase in layers, the model is starting to fit on noise and is beginning overfit.

From the overall performance, the tunned model exhibits better and more powerful prediction ability than the original model, providing more accurate prediction for the chrun modelling. Although the changes in recall and f1score are small, the tunned model still has a great improvement in accuracy. The tunned model is successful in the whole process of deep learning through train, test, and tunning.









Reference

[1] B. Huang, M. T. Kechadi, and B. Buckley, “Customer churn prediction in Telecommunications,” *Customer churn prediction in telecommunications*, 12-Aug-2011. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0957417411011353. [Accessed: 16-Nov-2022].

[2] S. Kumar, “Churn modelling,” *Kaggle*, 20-Jun-2020. [Online]. Available: https://www.kaggle.com/datasets/shubh0799/churn-modelling. [Accessed: 16-Nov-2022].

[3] M. Gori, “Feedforward Neural Network,” *Feedforward Neural Network - an overview | ScienceDirect Topics*, 2018. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/feedforward-neural-network. [Accessed: 16-Nov-2022].

[4] D. Svozil, V. Kvasnicka, and J. Pospichal, “Feedforward Neural Network,” *Feedforward Neural Network - an overview | ScienceDirect Topics*, 08-Apr-1998. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/feedforward-neural-network. [Accessed: 16-Nov-2022].

[5] F. Hutter, L. Kotthoff, and J. Vanschoren, “Automated machine learning: Methods, systems, challenges,” *OAPEN Home*, 01-Apr-2020. [Online]. Available: https://library.oapen.org/handle/20.500.12657/23012. [Accessed: 16-Nov-2022].

[6] K. Team, *Keras*. [Online]. Available: https://keras.io/. [Accessed: 16-Nov-2022].

[7] J. Zhang and A. J. Morris, “A sequential learning approach for Single Hidden Layer Neural Networks,” *Neural Networks*, 11-Feb-1998. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0893608097001111. [Accessed: 16-Nov-2022].

[8] B. H. Shekar and G. Dagnew, “Grid search-based hyperparameter tuning and classification of Microarray Cancer Data,” *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, 2019.

[9] M. Stone, “Cross-validation:a review,” *Series Statistics*, vol. 9, no. 1, pp. 127–139, Jun. 2007.

[10] D. M. W. Powers, “Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and Correlation,” *arXiv.org*, 11-Oct-2020. [Online]. Available: https://arxiv.org/abs/2010.16061. [Accessed: 16-Nov-2022].

[11] K. C. McGwire and P. Fisher, “Spatially variable thematic accuracy: Beyond the confusion matrix,” *Spatial Uncertainty in Ecology*, pp. 308–329, 2001.