Machine Learning Report: Churn Prediction

Neural Network Classification Model

HUANG SHAN
0430-37-9985
Department of Economist
Kyoto University

February 1, 2026

Contents

Abstract

This report presents a deep neural network model for predicting customer churn using the Iranian Churn dataset provided by prof.Akita(a cleaned part originaly from the UCI Machine Learning Repository). The dataset contains 990 samples with 14 numerical features, balanced for churn and non-churn cases. Several neural network architectures were tested, and the final model (model4) consists of two hidden layers with 80 and 30 neurons respectively, each followed by a dropout of 0.3 to prevent overfitting. The model was trained using the Adam optimizer with binary crossentropy loss and early stopping based on validation loss. The final evaluation on the test set achieved a high accuracy of 92.93% and a ROC-AUC of 0.9616, indicating strong predictive performance. Random seeds were set to improve reproducibility. Slight variations in results may still occur when running the code on different machines.

1 Introduction

1.1 Background

Customer churn refers to the number of existing customers who discontinue a service over a given period of time. According to IBM, customer churn reflects customer satisfaction and loyalty, and it is closely related to a company's revenue and long-term profitability. A high churn rate often indicates potential problems in a company's products, pricing strategies, or service quality.

In the telecommunications industry, customer churn is a very important measurement. Due to intense market competition, and the customers' highly require of internet speed, signal, one would easily change service providers. Therefore, understanding and predicting customer churn is necessary for telecommunications companies to gain customers and keep a competitive advantage in the market.

1.2 Research Aim

The aim of this study is to build and apply a effective Deep Neural Network (DNN) model for this specific cleaned, small scaled homework data to predict whether a randomly selected customer will churn or not (binary outcome: 1 for churn, 0 for non-churn).

2 Data Description

- **Dataset source**: Telecommunications customer churn dataset given directly by Prof. Akita, it's a part of the data derived from UCI ML Repository, "Iranian Churn" https://archive.ics.uci.edu/dataset/563/iranian+churn+dataset
- **Number of samples**: $N = 990$
- **Features**: 13 customer characteristics including usage behavior, billing information, and demographic variables

- **Target variable**: *Churn*, a binary indicator equal to 1 if the customer churned and 0 otherwise

2.1 Variable Description

According to the original dataset website from UCI ML Repository, "Iranian Churn", the data contains:
Anonymous Customer ID
Call Failures: number of call failures
Complains: binary (0: No complaint, 1: complaint)
Subscription Length: total months of subscription
Charge Amount: Ordinal attribute (0: lowest amount, 9: highest amount)
Seconds of Use: total seconds of calls
Frequency of use: total number of calls
Frequency of SMS: total number of text messages
Distinct Called Numbers: total number of distinct phone calls
Age Group: ordinal attribute (1: younger age, 5: older age)
Tariff Plan: binary (1: Pay as you go, 2: contractual)
Status: binary (1: active, 2: non-active)
Churn: binary (1: churn, 0: non-churn) - Class label
Customer Value: The calculated value of customer

2.2 Descriptive Statistics

Table 1 reports the descriptive statistics of the variables used in this study for the full sample and separately for churn and non-churn customers.

Table 1: Descriptive Statistics of Customer Characteristics (Mean (SD))

Variable	**Overall**	**Churn = 0**	**Churn = 1**
Call Failure	7.54 (7.55)	7.60 (7.27)	7.48 (7.83)
Complains	0.21 (0.41)	0.01 (0.11)	0.40 (0.49)
Subscription Length	32.20 (8.79)	32.50 (8.06)	31.89 (9.47)
Charge Amount	0.62 (1.28)	1.02 (1.61)	0.23 (0.62)
Seconds of Use	3288.58 (3708.95)	5010.53 (4385.11)	1566.63 (1539.20)
Frequency of Use	53.28 (52.10)	77.42 (59.77)	29.13 (26.32)
Frequency of SMS	45.11 (84.86)	74.42 (110.19)	15.80 (23.52)
Distinct Called Numbers	18.84 (15.67)	25.29 (17.04)	12.39 (10.87)
Age Group	2.78 (0.82)	2.76 (0.92)	2.80 (0.71)
Tariff Plan	1.05 (0.23)	1.10 (0.30)	1.01 (0.11)
Status	1.46 (0.50)	1.17 (0.38)	1.75 (0.43)
Age	30.46 (7.99)	30.29 (8.97)	30.64 (6.89)
Customer Value	313.81 (417.27)	502.82 (510.13)	124.81 (129.43)

2.3 Data Description

The dataset consists of 990 observations with a very balanced distribution between churned and non-churned customers(very close to 1:1). Table 1 summarizes the descriptive statistics of the variables used in this study.

Clear differences can be observed between churn and non-churn customers. Customers who churn show substantially lower service usage, including shorter call duration, lower frequency of use, fewer SMS messages, and fewer distinct called numbers. In addition, churned customers tend to have a significantly lower customer value and a higher incidence of complaints. In contrast, demographic variables such as age and subscription length show relatively small differences between the two groups.

3 Data Preprocessing

3.1 Feature Engineering

Before training the neural network, the dataset underwent several preprocessing steps to ensure that the input data were suitable for deep learning.

Firstly, the index column was removed. Then the target variable, *Churn*, was separated from the feature variables. Additionally, continuous variables, including *Call Failure, Complains, Subscription Length, Charge Amount, Seconds of Use, Frequency of Use, Frequency of SMS, Distinct Called Numbers, Age, Customer Value*, were standardized using z-score normalization.

Besides, categorical variables like *Age Group, Tariff Plan, Status*, were transformed using one-hot encoding to allow the neural network to learn patterns from categorical features. After these transformations, the feature matrix consisted of 19 columns.

3.2 Train/Test Split

Finally, the data were split into training and test sets. 90% of the data (891 observations) were used for training, while 10% (99 observations) were reserved for testing. The split was stratified by the *Churn* variable to maintain the balance of positive and negative classes in both sets.

4 Methodology

4.1 Model Choice

A deep neural network was chosen as the predictive model for this study. The selection is motivated by two factors. First, the assignment specifically requires the use of a neural network(ha ha ha). Second, DNNs are widely used in modern machine learning due to their ability to capture complex nonlinear relationships between features and target variables, as above it's suitable for customer churn prediction.

4.2 Neural Network Architecture

Several neural network architectures were experimented with during the model selection stage. After comparing their predictive performance and generalization ability, the fourth model (model4) was selected as the final specification.

The neural network was implemented using TensorFlow and Keras. The architecture consists of three main components:

- **Input Layer:** 19 neurons corresponding to the 19 input features after preprocessing.
- **Hidden Layers:** Two fully connected hidden layers with 80 and 30 neurons respectively, both using the ReLU activation function. Dropout layers with a dropout rate of 0.3 were added after each hidden layer to reduce overfitting by randomly deactivating neurons during training. The ReLU activation function was chosen because it is widely used in modern industry applications and is known for its effectiveness in mitigating the vanishing gradient problem.
- **Output Layer:** A single neuron with a sigmoid activation function is used since a binary outcome (churn or non-churn) is expected.

The model was compiled with the *binary cross-entropy* loss function and the *Adam* optimizer. The network was trained for up to 200 epochs with a batch size of 32. An early stopping mechanism monitored the validation loss with a patience of 10 epochs, restoring the best-performing weights to prevent overfitting. To ensure reproducibility, random seeds were fixed for NumPy, Python, and TensorFlow. Minor numerical differences may still occur due to hardware and backend implementations, which is common in deep learning models.

5 Model Training

4 models(*model, model2, model3 and model4*) were trained and compared in this assignment. The 4 models are generally the same and the difference is just the value of dropout rate, layers' dense etc. The selected neural network (model4) was trained using supervised learning on the preprocessed training dataset.

5.1 Training Procedure

The model was trained by minimizing the binary cross-entropy loss function using the Adam optimizer. To improve generalization performance and prevent overfitting, dropout regularization and an early stopping strategy were applied during training. The dataset was split into training and validation sets, with 10% of the training data used for validation.

5.2 Hyperparameter Settings for *model4*

- **Optimizer:** Adam optimizer, which adaptively adjusts the learning rate for each parameter and is well-suited for training deep neural networks.
- **Learning Rate:** The default learning rate of the Adam optimizer was used.

- **Loss Function:** Binary cross-entropy.
- **Epochs:** The model was trained for a maximum of 200 epochs. Early stopping was employed with a patience of 10 epochs.
- **Batch Size:** 32 observations per mini-batch.
- **Regularization:** Dropout layers with a dropout rate of 0.3 were applied after each hidden layer to reduce overfitting.

After training, the final model performance was evaluated on a held-out test set that was not used during training or validation.

6 Results & Evaluation

6.1 Evaluation Metrics

To evaluate the performance of the neural network models, both *test accuracy* and *ROC-AUC* were used. Additionally, learning curves analysis is also considered. Since the dataset was preprocessed and balanced by the instructor, with an approximately 1:1 ratio between churn and non-churn observations, test accuracy is the major measurement. In addition, ROC-AUC was reported to further compare models with similar accuracy values, as it reflects the model's ability to discriminate between the two classes across different classification thresholds.

6.2 Model Performance Comparison

Table 2 summarizes the performance of all four neural network models evaluated on the test dataset. While *model2* and *model4* achieved the same highest test accuracy, *model4* obtained the highest ROC-AUC among all models, indicating stronger overall discriminative power.

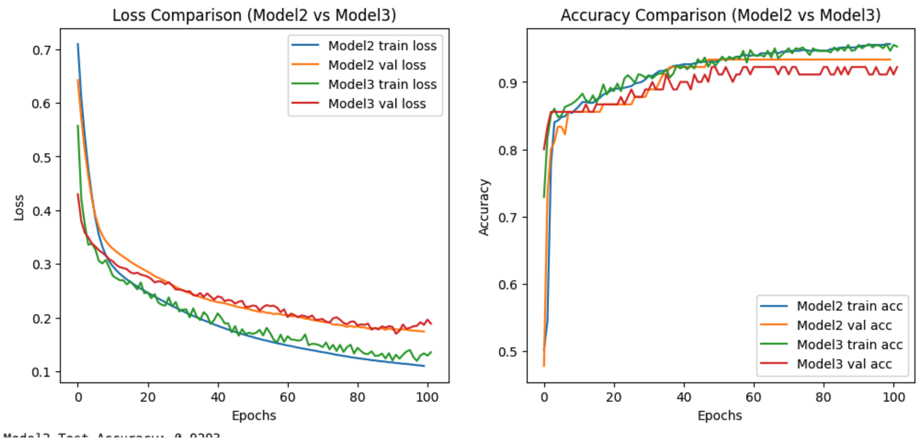
Table 2: Performance Comparison of Neural Network Models

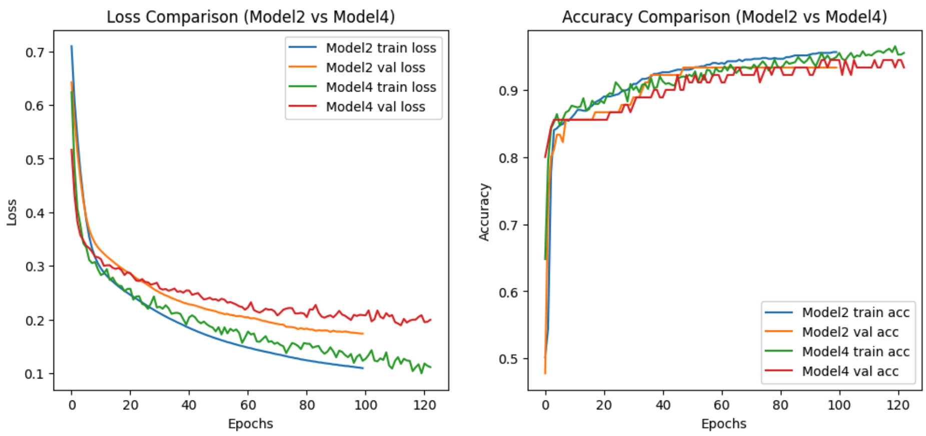
Model	Hidden Layers	Dropout	Test Accuracy	TEST ROC-AUC
model	42, 24	0.2, 0.2	0.9091	0.9592
model2	32, 16	None	0.9293	0.9596
model3	100, 30	0.3, 0.3	0.9192	0.9600
model4	**80, 30**	**0.3, 0.3**	**0.9293**	**0.9616**

6.3 Learning Curves Analysis

The learning curves of the models were analyzed by comparing training and validation loss and accuracy across epochs. model showed relatively close training and validation loss curves but achieved lower overall performance. *model2* exhibited higher training and validation accuracy, but signs of mild overfitting were observed due to the gap between training and validation curves.

model2, model3 and model4 incorporated dropout layers, which helped reduce overfitting and improved generalization. In particular, *model4* demonstrated stable convergence behavior, with smooth validation loss curves and reduced variance across epochs. Although its validation loss was slightly higher than that of *model2* in some epochs, the overall training process was more stable and less prone to overfitting.





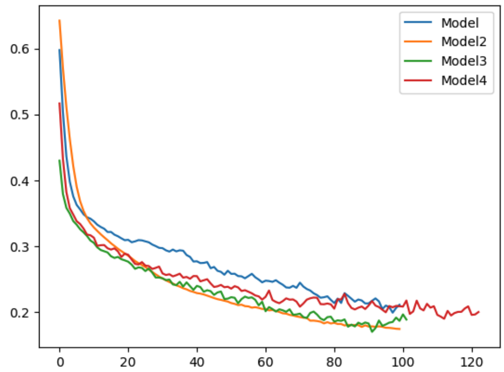


Figure 1: Loss Comparsions of 4 models

6.4 Final Model Selection

Considering both quantitative metrics and learning curve behavior, *model4* was selected as the final model. Although *model2* achieved the same test accuracy, *model4* outperformed all other models in terms of ROC-AUC and showed better generalization characteristics due to the use of dropout regularization. Therefore, *model4* provides the best balance between predictive performance and was chosen as the final neural network architecture for this task.

7 Discussion / Analysis

Overall, all four neural network models achieved relatively high predictive performance, indicating that the preprocessed dataset contains informative features for predicting customer churn. The balanced class distribution further contributed to stable training behavior and reliable evaluation using test accuracy.

Among the evaluated models, *model4* demonstrated the best overall performance. Although *model2* and model4 achieved the same test accuracy, *model4* obtained the highest ROC-AUC. This indicates that *model4* is more robust when the decision threshold is varied.

From the learning curve analysis, *model* showed limited model capacity, resulting in lower accuracy despite relatively stable training and validation loss curves. *model2*, which increased the number of hidden units without regularization, achieved higher training accuracy but exhibited mild overfitting, as evidenced by a larger gap between training and validation curves.

model3 and *model4* incorporated dropout layers, which effectively mitigated overfitting by randomly deactivating neurons during training. In particular, *model4* benefited from a deeper architecture combined with dropout regularization, achieving a better balance between model complexity and generalization. Although *model4*'s validation loss was not always the lowest at every epoch, its overall training process was more stable and resulted in superior test-set performance.

Despite these strengths, several limitations should be noted. First, the neural network architecture and hyperparameters were selected through manual experimentation rather than systematic tuning methods such as grid search or cross-validation. Second, while neural networks provide strong predictive performance, they lack interpretability compared to traditional statistical models, making it difficult to directly assess the contribution of individual features to churn prediction. Third, though Adam was adapted in 3 models, he learning rate is set automatically. Furthermore, since the out come of testing accuracy is very close, due to the shortage of reproductability, for another only one time comparison the other models may behave superior to *model4*, average testing accuracy is encouraged in the future as a better model comparison method.

Future work could explore more advanced regularization techniques, alternative network architectures, or model explainability methods such as SHAP or permutation feature importance. Additionally, applying cross-validation or testing the model on a different dataset would further strengthen the robustness of the results.

In summary, the analysis demonstrates that incorporating regularization techniques such as dropout is crucial for improving generalization in neural networks. *model4* achieves the best trade-off between accuracy, robustness, and stability, making it a suitable final model for customer churn prediction in this study.

8 Conclusion

This study applied deep neural networks to a customer churn prediction problem using a balanced and fully preprocessed dataset. After exploring multiple network architectures, a neural network with two hidden layers(80, 30) and dropout regularization(both layers 0.3) was selected as the final model.

The empirical results show that neural networks can effectively capture nonlinear relationships in the data and achieve high predictive performance. Among the evaluated models, the selected architecture achieved the best balance between test accuracy and ROC-AUC, indicating strong generalization ability and robust classification performance.

Overall, this assignment demonstrates the practical effectiveness of deep learning methods for binary classification tasks such as churn prediction. Future work may focus on systematic hyperparameter tuning and improving model interpretability to further enhance the applicability of neural networks in real-world decision-making.

References

[1] Kyoto University. *Machine Learning Course Materials (Customer Churn Dataset)*. Available at: `https://panda.ecs.kyoto-u.ac.jp/portal/site/2025-204-A613-000/tool/3ced5b5d-0eb8-4636-8416-f65512bca2ad`

[2] IBM. *What is Customer Churn?* Available at: `https://www.ibm.com/think/topics/customer-churn`

[3] UCI Machine Learning Repository, *Iranian Churn Dataset*, `https://archive.ics.uci.edu/dataset/563/iranian+churn+dataset`