University of Missouri–St. Louis

Department of Computer Science

A Project Report on

Customer Churn Prediction

Submitted by:
Sachina Koirala
Student ID: 18281333

Submitted to:
Dr. Badri Adhikari

November 21, 2024

1 Introduction

“Churn” is the term used to describe when a customer stops using a certain organisation’s services. Customer churn, specially in the banking industry, is a critical issue as it directly affects customer retention and revenue. Predicting churn involves analyzing multiple factors, such as customer demographics, account activities, product engagement, and financial behavior.

With an interest in applying Artificial Intelligence to real-world problems, I aim to explore how neural networks can be utilized to predict customer churn in banks. This project focuses on building a neural network model to identify customers at risk of leaving.The insights gained will be beneficial in designing personalized customer retention strategies.For example: Offer a gift voucher or any promotional pricing and lock them in for an additional year or two to extend their lifetime value to the company.

2 Dataset

The dataset was obtained from Kaggle Data Science website called the ”Binary Classification with Bank Churn Dataset”. The dataset consists of 175,028 rows (customers) and 9 columns (features), including both input features and the target column (Exited), which indicates whether the customer churned.

2.1 Dataset Description

Following are the columns in this dataset:

- CreditScore: Customer’s credit score, indicating financial reliability.
- Age: Customer’s age in years.
- Tenure: Number of years the customer has been with the bank.
- Balance: Customer’s account balance in monetary terms.
- NumOfProducts: Number of bank products used by the customer.
- HasCrCard: Whether the customer owns a credit card (1 = Yes, 0 = No).
- IsActiveMember: Whether the customer is an active bank member (1 = Yes, 0 = No).
- EstimatedSalary: Estimated annual salary of the customer.
- Exited: Target variable indicating if the customer churned (1 = Yes, 0 = No).

This dataset also includes both numerical and categorical features:

- Numerical Features: ’CreditScore’, ’Age’, ’Tenure’, ’Balance’, ’EstimatedSalary’
- Categorical Features: ’NumOfProducts’, ’HasCrCard’, ’IsActiveMember’

2.2 Input Data Visualization

The histogram plot of every input features showing their maximum and minimum value as well as how they are distributed can be seen in the figures below.

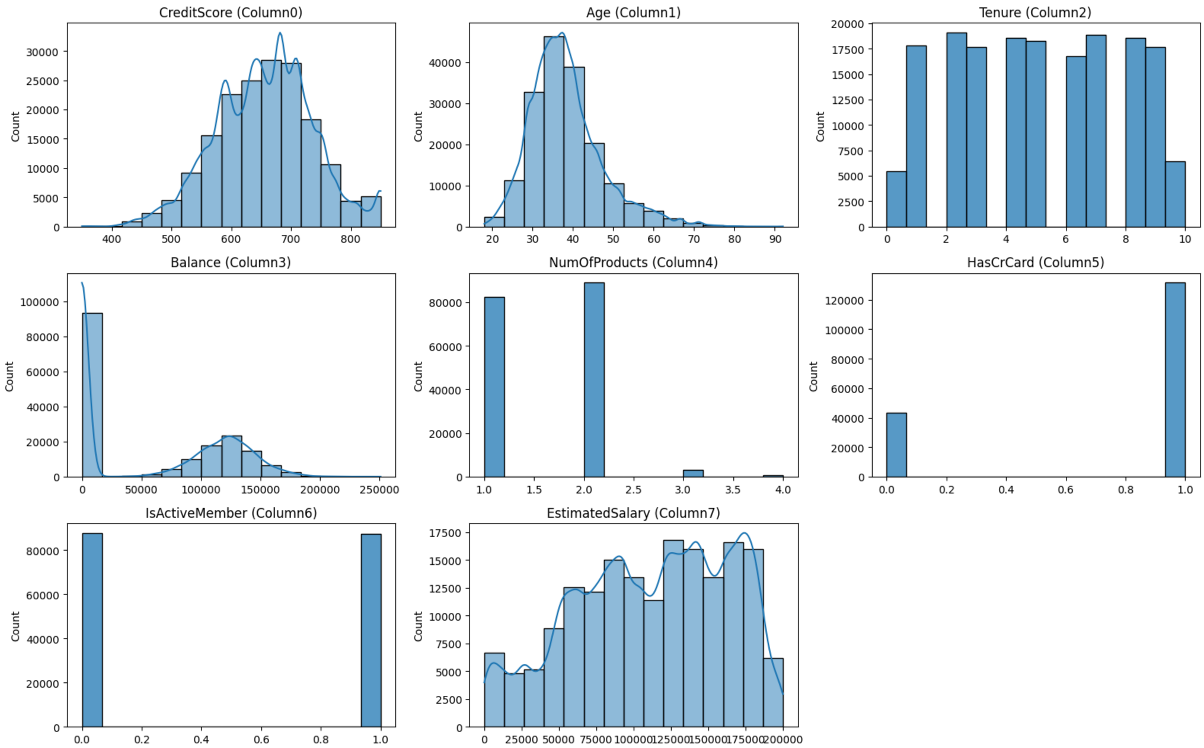


Figure 1: Input Data Distributions

Table 1: Statistics of Input Features

Feature	**Min**	**Max**	**Mean**	**Std**
CreditScore	350.00	850.00	656.11	81.15
Age	18.00	92.00	38.17	8.97
Tenure	0.00	10.00	5.02	2.81
Balance	0.00	250898.09	56676.77	62982.24
NumOfProducts	1.00	4.00	1.55	0.55
HasCrCard	0.00	1.00	0.75	0.43
IsActiveMember	0.00	1.00	0.50	0.50
EstimatedSalary	11.58	199992.48	111863.30	50814.97

2.3 Output Data Visualization

Following is the distribution of output labels:

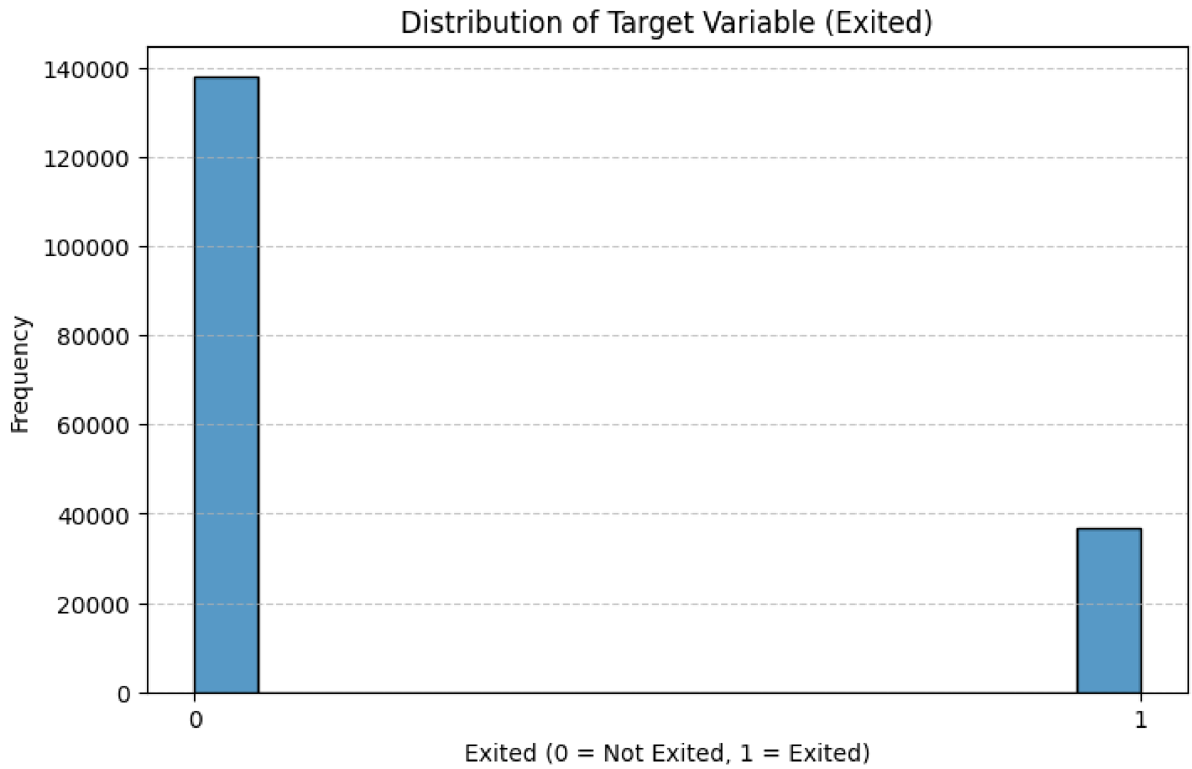


Figure 2: Output Data Distributions

3 Data Processing

3.1 Data Splitting

First of all, the data was randomly shuffled and then the dataset was split into training,validation and test.

3.2 Data Normalization

Normalization is done in this project to ensure that all input features have a similar scale. This helps the model to converge faster and prevents features with larger magnitudes from dominating the learning process.

In this project, the numerical features were normalized using the Z-score normalization technique. This transformation ensures that the numerical columns have a mean of 0 and a standard deviation of 1. The formula applied is:

$$X_{\text{normalized}} = \frac{X - \text{mean}(X)}{\text{std}(X)}$$

This method centers the data around zero with a standard deviation of one. The following features were normalized:

- CreditScore

- Age
- Tenure
- Balance
- EstimatedSalary

The training set was normalized using its own mean and standard deviation, while the validation set was normalized using the same statistics from the training set.

4 Modeling

Artificial Neural Network (ANN) was used to create the model.

5 Varying Neural Network Architecture

A neural network model based on layers, known as ANN (Artificial Neural Network) is build. It looks a bit like the following diagram.

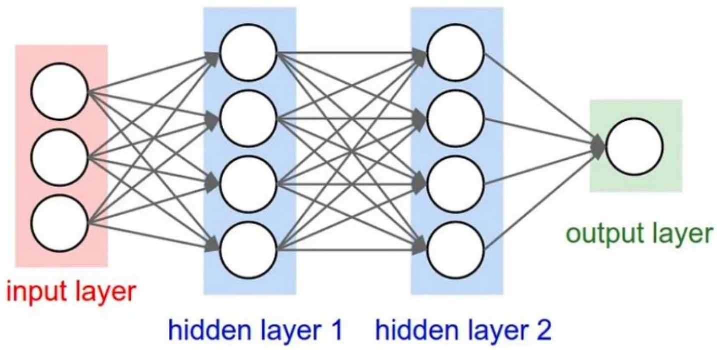


Figure 3: Neural Network architecture

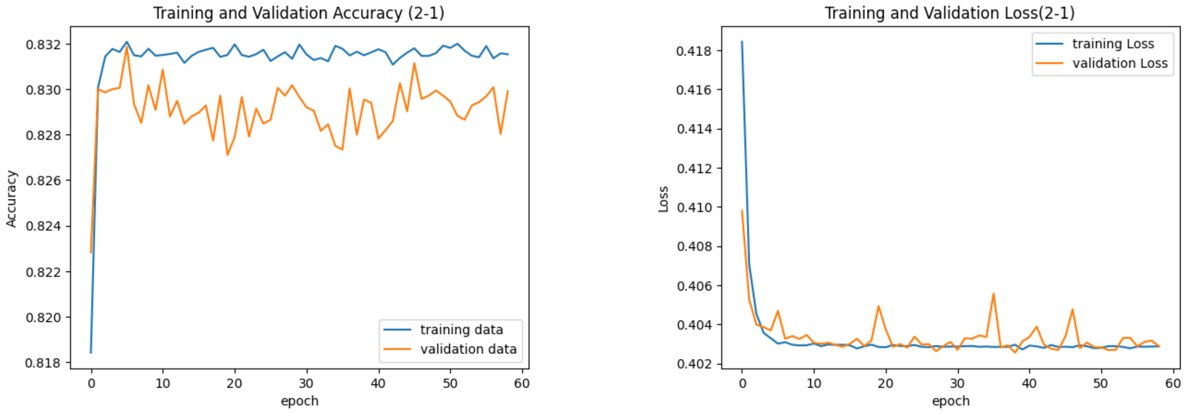
5.1 Model 1

This model consisted of two layers: one hidden layer with 2 neurons and an output layer with 1 neuron. The hidden layer used the ReLU activation function to introduce non-linearity, while the output layer used a sigmoid activation for binary classification. The performance measure is shown as:

Table 2: Performance of Model 1 (2-1)

Training Accuracy	Validation Accuracy	Validation Loss
83.23%	82.82%	0.4037

5.1.1 Learning Curve for Model 1



(a) Training and validation accuracy. (b) Training and validation loss.

Figure 4: Comparison of training and validation metrics for Model 1.

5.2 Model 2

This model takes input features and passes them through two hidden layers, the first with 16 neurons and the second with 8 neurons, both using activation functions to introduce non-linearity. Finally, the output layer with 1 neuron produces the prediction. The performance model is as shown as:

Table 3: Performance of Model 2 (16-8-1)

Training Accuracy	**Validation Accuracy**	**Validation Loss**
85.59%	85.69%	0.3405

5.2.1 Learning Curve for Model 2

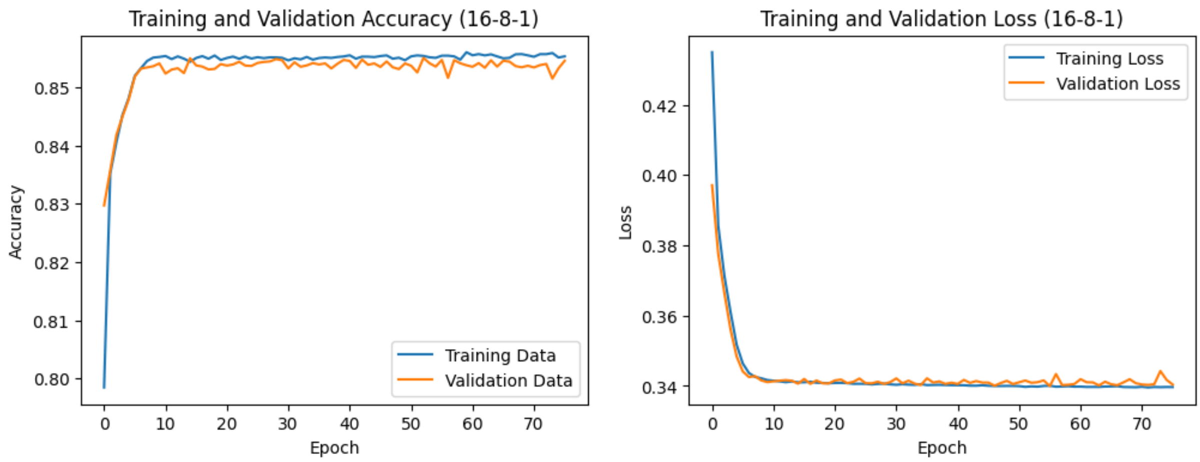


Figure 5: Comparison of training and validation metrices for Model 2

5.3 Model 3

This model consisted of two layers: one hidden layer with 4 neurons and an output layer with 1 neuron. The performance measure is as shown as:

Table 4: Performance of Model 3(4-1)

Training Accuracy	Validation Accuracy	Validation Loss
84.12%	84.20%	0.3648

5.3.1 Learning Curve for Model 3

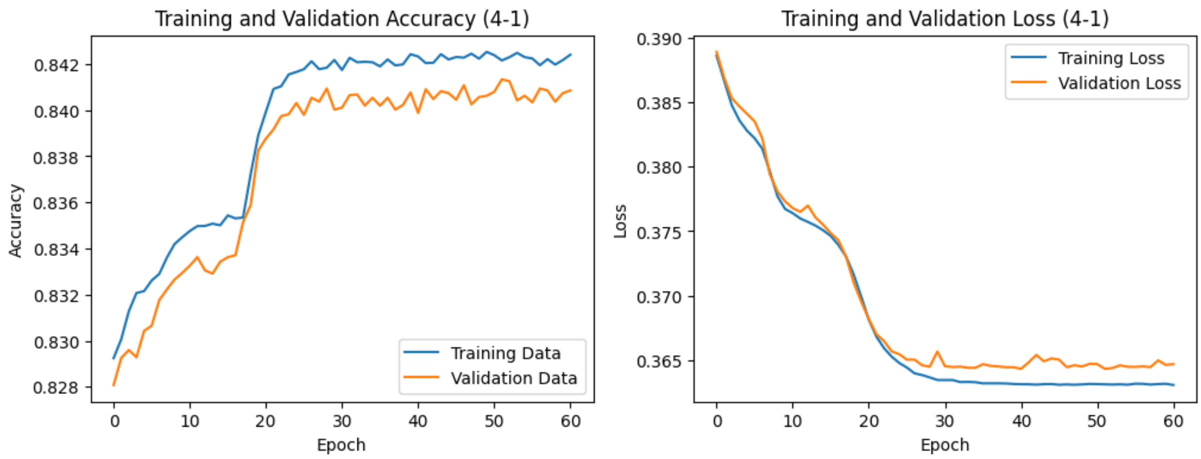


Figure 6: Comparison of training and validation metrices for Model 3

5.4 Model 4

This model consisted of two layers: one hidden layer with 4 neurons and an output layer with 1 neuron. The performance measure is shown as:

Table 5: Performance of Model 4 (8-1)

Training Accuracy	Validation Accuracy	Validation Loss
85.17%	85.16%	0.3514

5.4.1 Learning Curve for Model 4

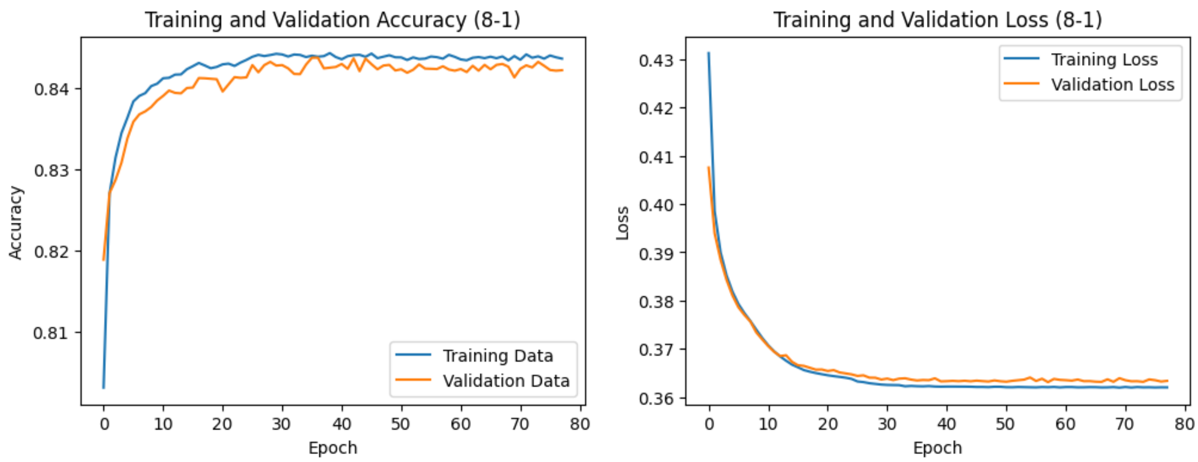


Figure 7: Comparison of training and validation metrices for Model 3

5.5 Model 5

In this model, the first layer has 64 neurons followed by layers with 32, 16, and 8 neurons, all using the ReLU activation function.The final layer has 1 neuron with a sigmoid activation function, making it suitable for binary classification (where an output is a probability between 0 and 1). The performance measure is shown as:

Table 6: Performance of Model 5 (64-32-16-8-1)

Training Accuracy	Validation Accuracy	Validation Loss
85.17%	85.16%	0.3514

5.5.1 Learning Curve for Model 5

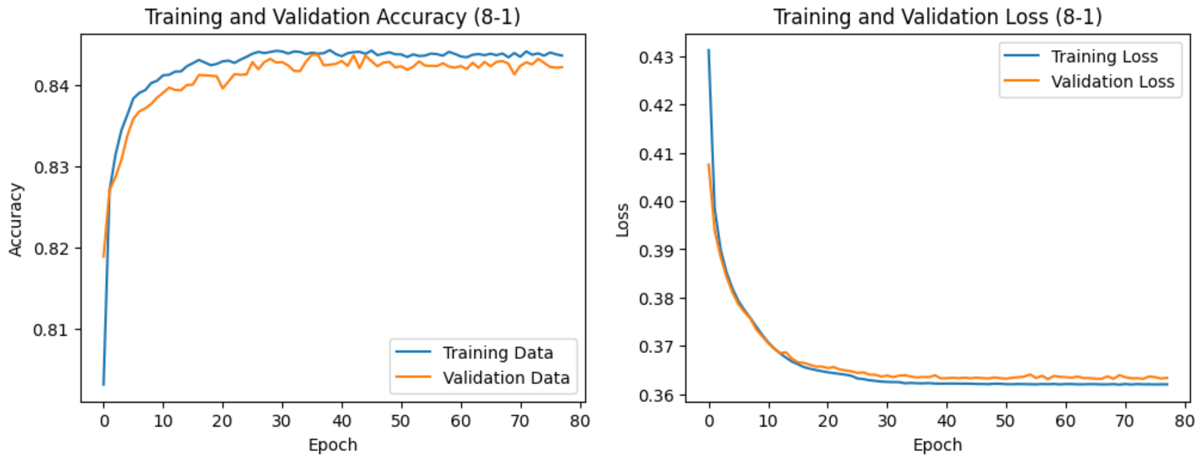


Figure 8: Comparison of training and validation metrices for Model 5

5.6 Logistic regression Model

I also build a logistic regression model for predicting customer churn. The model was created using Keras's `Sequential` API, consisting of a single dense layer with one neuron and a sigmoid activation function, which is appropriate for binary classification. The training process stopped after 21 epochs as the validation loss did not improve further.

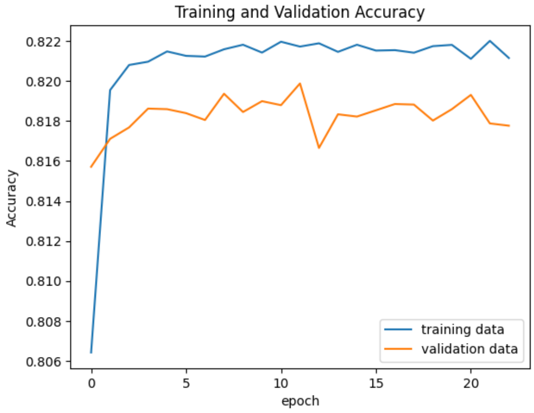
The table given below shows the performance of the logistic regression model. The given

Table 7: Performance of Logistic Regression Model

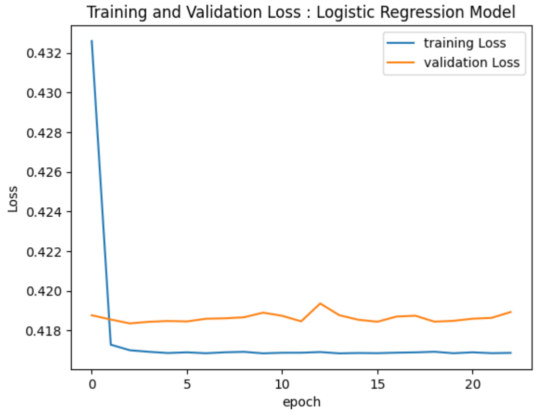
Training Accuracy	Validation Accuracy	Validation Loss
85.85 %	85.50%	0.3411

graph below shows the learning curve for the logistic regression model.

5.6.1 Learning Curve for Model 1



(a) Training and validation accuracy.



(b) Training and validation loss.

Figure 9: Comparison of training and validation metrics for Model 1.

6 Feature Reduction

To be continued...