

Deep Learning Applications in Management Analytics

Krishan Gupta | 261151116

Predicting Customer Churn using Feed-Forward Neural Networks

Introduction

Customer churn is a critical issue faced by businesses across various industries. It refers to the loss of clients or customers who cease to interact with a company or business within a specific period. A high churn rate can have significant negative impacts on a company's success, both tangible and intangible. Therefore, businesses strive to retain as many customers as possible to maintain a stable revenue stream and foster long-term growth.

In this report, we explore the application of feed-forward neural networks to predict customer churn. By leveraging machine learning techniques, we aim to develop a predictive model that can identify customers who are at a high risk of churning. This proactive approach enables businesses to take targeted actions to retain these customers and minimize the impact of churn on their bottom line.

Dataset and Preprocessing

To build our churn prediction model, we obtained a customer churn dataset from Kaggle, specifically the "Telco Customer Churn" dataset provided by IBM. This dataset contains information about customers of a telecommunications company, including their demographic details, services subscribed, and whether they churned or not.

Data source: <https://www.kaggle.com/yeanzc/telco-customer-churn-ibm-dataset>

Shape of Dataset: (7033,33)

The first step in our analysis was to pre-process the dataset. We performed the following pre-processing tasks:

- **Dropped unnecessary columns such as**

- 'Churn Label', 'Churn Score',
- 'CLTV', and 'Churn Reason'

as they were not required for model building.

- **Removed univariate columns like**

- 'CustomerID',
- 'Count',

- 'Country',
- 'State', and
- 'Lat Long'

as they did not contribute to the predictive power of the model.

- Handled missing values in the 'Total_Charges' column by replacing them with 0.
- Converted the 'Total_Charges' column to numeric data type.
- Performed one-hot encoding on categorical features.
- **Converted all the predictors to Floats to feed to Neural Network.**

After preprocessing, we split the data into training and testing sets using the `train_test_split` function from scikit-learn. This allows us to evaluate the performance of our model on unseen data and assess its generalization capability.

Model Architecture and Training

Feed-forward neural network was implemented using **PyTorch** to predict customer churn. The architecture of our baseline model consists of an **input layer**, **two hidden layers**, and an output layer. The input layer size corresponds to the number of features in the pre-processed dataset, while the hidden layers have 100 neurons each. The output layer produces a single value representing the probability of churn.

We used the **ReLU activation function** in the hidden layers and the sigmoid activation function in the output layer¹. The binary cross-entropy loss function was employed to measure the difference between the predicted probabilities and the actual churn labels. The **Adam optimizer** was used to update the model's weights during training¹.

To train the model, we utilized the **PyTorch Lightning framework**, which simplifies the training process and provides useful utilities for monitoring and logging. We trained the model for 10 epochs and evaluated its performance on the test set.

Model Evaluation

After training the **baseline model**, we evaluated its performance using several metrics. The accuracy of the model on the test set was 80.01%, indicating that it correctly predicted the churn status for 80.01% of the customers. However, accuracy alone does not provide a complete picture of the model's performance, especially in **imbalanced datasets (73% Non-churned and 27% Churned)** like ours, where the number of churned customers is relatively small compared to non-churned customers.

To gain a better understanding of the model's performance, we computed the confusion matrix and classification report¹. The confusion matrix revealed that the model had a recall of 44%, meaning it correctly identified 44% of the actual churned

customers¹. The precision was also relatively low, indicating a high number of false positives.

We prioritized recall as our evaluation metric because the cost of losing a customer is higher than the cost of retaining a customer. By focusing on recall, we aim to minimize the number of customers that the model predicts will not churn but actually do churn.

Experimenting with Different Architectures and Hyperparameters

To improve the performance of our churn prediction model, we experimented with different network architectures, activation functions, and learning rates. We defined three different architectures additional to initial base model:

Architecture	Hidden Layer Sizes	Activation Function	Learning rate
Base	100,100	ReLU	0.01
Arch_1	100,100	ReLU	0.011
Arch_2	200,100	Tanh	0.0011
Arch_3	100,100,5	LeakyReLU	0.0051

Hyperparameters of Different Architectures

Other Parameters which we set same for all the architectures:

Batch Size: 64

Optimizer: Adam

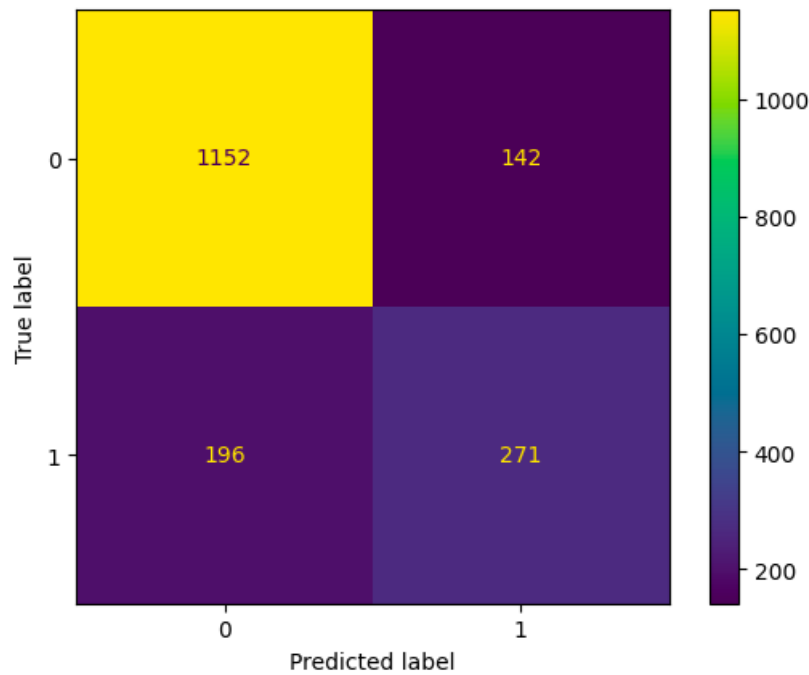
Loss Function: Binary Cross Entropy

Epochs: 20

We trained and evaluated each architecture using the same training and testing data. The evaluation results showed that Architecture 3 achieved the highest recall of 58%, while Architecture 1 and Architecture 2 had recall values of 41% and 42%, respectively.

Architecture	Recall	Accuracy
Base	44%	80.01%
Arch_1	41%	79.55%
Arch_2	42%	78.53%
Arch_3	58%	80.80%

All Iterations Results



Arch_3 Confusion matrix

Conclusion

In this report, we explored the application of feed-forward neural networks for predicting customer churn. By leveraging a real-world dataset and preprocessing techniques, we developed a baseline model that achieved an accuracy of 80% and a recall of 44%.

Through experimentation with different architectures and hyperparameters, we identified an improved model (Architecture 3) that achieved a recall of 58%. This indicates that the model is capable of identifying a higher percentage of customers who are likely to churn, enabling businesses to take proactive measures to retain them.

However, it is important to note that churn prediction is an ongoing process, and the model should be continuously monitored and updated as new data becomes available. Businesses should also consider incorporating additional features and data sources to enhance the predictive power of the model.

Furthermore, churn prediction is just one aspect of customer retention strategies. Businesses should also focus on improving customer experience, addressing customer feedback, and implementing targeted retention campaigns to minimize churn and foster long-term customer loyalty.

In conclusion, feed-forward neural networks provide a powerful tool for predicting customer churn. By leveraging machine learning techniques and experimenting with different architectures, businesses can develop accurate and reliable churn prediction models. These models enable proactive customer retention efforts, ultimately leading to improved customer satisfaction and business success.

References

- Customer churn prediction using machine learning- A comprehensive overview: <https://www.leewayhertz.com/ai-and-ml-in-customer-churn-prediction/>
- How and Why to Calculate and Predict Customer Churn: <https://www.pecan.ai/blog/how-why-churn-analysis-prediction/>
- Predicting Customer Churn: <https://www.avaus.com/blog/predicting-customer-churn/>
- Amplitude: <https://amplitude.com/blog/churn-prediction>