**MGSC 673 – Assignment 2**

Deep Learning Applications in Management Analytics

Konstantin Volodin – 261083570

Table of Contents

[1. Introduction 2](#_Toc135605782)

[2. Data Review 2](#_Toc135605783)

[2.1. Data Cleaning 2](#_Toc135605784)

[2.2. Data Preprocessing 3](#_Toc135605785)

[2.3. Data Splitting 3](#_Toc135605786)

[3. Network Architecture 3](#_Toc135605787)

[3.1. Network Architecture Experimentation 3](#_Toc135605788)

[3.2. Experimentation Results 4](#_Toc135605789)

[4. Final Model Results 4](#_Toc135605790)

[Appendix 5](#_Toc135605791)

# Introduction

Customer churn, also known as customer attrition or customer turnover, refers to the phenomenon where customers discontinue their relationship with a business or service provider. It is a crucial concern for companies across various industries. Understanding and predicting customer churn is of paramount importance as it directly impacts a company's profitability and market competitiveness. By accurately identifying potential churners in advance, businesses can proactively take measures to retain valuable customers, optimize their marketing strategies, and allocate resources effectively.

In this report, we will delve into the application of feed-forward neural networks to predict customer churn in the telecom industry. Specifically, we will utilize the PyTorch Lightning framework to develop and train our neural network models. We will explore different network structures, including variations in the number of hidden layers, neurons per layer, and activation functions, to identify the optimal architecture for predicting customer churn. Finally, we will select the most promising model based on performance metrics and thoroughly evaluate its predictive capabilities, thereby providing valuable insights for businesses aiming to mitigate customer churn and enhance customer retention strategies.

# Data Review

In this report, we will utilize a customer-level dataset obtained from Kaggle. The dataset consists of a substantial collection of 100,000 records, encompassing 98 relevant variables. These variables can be categorized into two main sections: (1) usage statistics, which comprise 67 features such as average monthly revenue and minutes used by each customer, and (2) customer demographics, which encompass 31 columns including attributes like location, presence of children, ethnicity, and more. The target variable, *churn* indicates whether a customer has discontinued product usage within a 31-60 day timeframe after the observation period. Our goal is to leverage this dataset to construct a robust predictive model that can estimate the probability of customer churn within the subsequent two-month period.

## Data Cleaning

During the initial analysis of the dataset, it was observed that only 26% of the dataset contains all the relevant features, indicating the presence of missing values that require appropriate processing. Notably, the usage statistics data exhibits a high level of cleanliness, with less than 1% of the data missing. In contrast, the customer demographics data appears to be less clean, with approximately 10% of the data being absent.

Specifically, three features stand out as requiring attention, as each of them has roughly 40% of the data missing. These features are *numbcars*, *HHstatin* and *dwllsize* with a missing rate of about 50%, 40%, and 40% respectively. To address these missing values, we will employ the median imputation method for numerical data and the most frequent imputation technique for categorical data. These approaches will help ensure the integrity and completeness of the dataset, enabling us to proceed with subsequent stages of analysis and modeling effectively.

## Data Preprocessing

The data preprocessing stage involves distinct steps for handling categorical and numerical variables. Numerical variables undergo scaling to achieve unit variance and a mean of 0, ensuring comparability and interpretability. Categorical variables with fewer than 10 features are one-hot encoded, expanding them into binary columns. However, categorical variables with more than 10 features are currently dropped, with future consideration for entity embedding to incorporate them into the model. After preprocessing, the transformed dataset comprises 108 features, consisting of the original variables and newly created one-hot encoded columns.

## Data Splitting

The dataset is divided into three subsets, with 40% allocated for training, 10% for testing, and 50% for validation. The training is used for calculating the model's weights through iterative training epochs. The test set, serves as an independent evaluation set, allowing us to assess the model's performance after each training epoch. This evaluation helps us monitor the model's generalization and identify any potential overfitting or underfitting issues. The validation set is only used at the very end of the training process to evaluate the final model's performance, providing an unbiased assessment of its predictive accuracy. By withholding this set until the end, we can ensure that the model has not been influenced or optimized based on the validation data, enabling us to gauge its true performance on unseen instances.

To expedite the experimentation process, we only 40% of the available data for the training dataset. This decision allows for faster iterations and initial insights into model performance. Furthermore, a minibatch size of 500 is utilized during training.

# Network Architecture

For the purposes of this project, we employed a dense neural network for modeling. Dense neural networks, also known as feed-forward neural networks, are a popular modeling approach in the field of machine learning and artificial intelligence. These networks consist of multiple layers of interconnected nodes, where each node in a layer is connected to every node in the subsequent layer. This dense connectivity allows for the propagation of information and the extraction of complex patterns and relationships from the input data.

For our baseline architecture, we will establish a preliminary structure that serves as a starting point for our modeling efforts. This baseline architecture will provide a foundation for comparison and serve as a benchmark for evaluating the performance of alternative network configurations. Additionally, we will explore the impact of varying layer sizes, the number of layers, and other architectural parameters. This experimentation enables us to assess the influence of these design choices on the model's predictive accuracy and identify optimal network configurations that yield the best results for predicting customer churn in the telecom industry.

## Network Architecture Experimentation

To evaluate different combinations of models we start with a baseline model and change parameters one by one. The baseline model serves as our initial reference point, featuring a single hidden layer comprising 256 neurons, a dropout rate of 25%, RELU activation functions, and utilizing the Adam optimizer.

To systematically investigate the effects of various architectural parameters, each parameter is modified independently, without considering parameter interactions. To ensure thorough exploration, we train each parameter combination for 50 epochs. This duration is typically sufficient, as it extends beyond the point at which the model begins to overfit, providing us with meaningful insights into the model's convergence and generalization capabilities.

In our experiments, the following will be done:

* Vary the number of neurons per layer: configurations of 32, 64, 128, 256, and 512 neurons.
* Explore the effects of altering the number of hidden layers: from 1 to 3 hidden layers.
* Investigate different dropout rates: ranging from 0.1 to 0.6.
* Explore the effects of using different activation functions: ReLU, LeakyReLU, ELU, and Tanh.
* Experiment with different optimizers, each with a learning rate of 0.001: Adam, SGD, and LBFGS.

## Experimentation Results

To select the best network architecture, we analyze the test error over time for each parameter configuration throughout the 50 epochs. By examining the performance of different parameter sets, we can identify the combinations that demonstrate the lowest test error across the entire training process. Based on our experimentations (see Figures 1-5), the following are the results:

* The layer size of 64 tends to generalize the best.
* Network architectures with 2 hidden layers can achieve the best performance**.**
* A dropout rate of 0.3 shows the best regularization effects for preventing overfitting.
* The LeakyReLU activation function outperforms other activation functions in capturing nonlinear relationships within the data.
* The Adam optimizer exhibits the fastest convergence among the different optimizers evaluated.

# Final Model Results

Based on the experimentation results, we have selected the final model that demonstrates the highest predictive accuracy for customer churn prediction in the telecom industry. The chosen model incorporates a network architecture with a layer size of 64 and utilizes 2 hidden layers and a dropout rate of 0.3 is chosen.

The selected configuration was trained for 100 epochs using the same train/test/validation split as described earlier. To determine the final model, we identified the epoch that resulted in the lowest test error. This epoch represents the point at which the model achieved the best trade-off between bias and variance, indicating its optimal performance.

The final model was carefully selected based on the evaluation of various metrics. Epoch 41, with a loss value of 0.2301, emerged as the optimal point for model performance. It achieved an accuracy score of 57.08%, a recall score of 42.5%, a precision score of 59.3%, and an F1 score of 49.5%. While the final model demonstrates improvement over random guessing, as the churn dataset is evenly split at 50/50, further enhancements are required to achieve better performance. The obtained accuracy of 57.08% indicates room for improvement, prompting future exploration of advanced techniques and alternative model architectures to enhance the predictive capabilities for customer churn in the telecom industry.

# Appendix

Figure 1: Best Hidden Layer Size

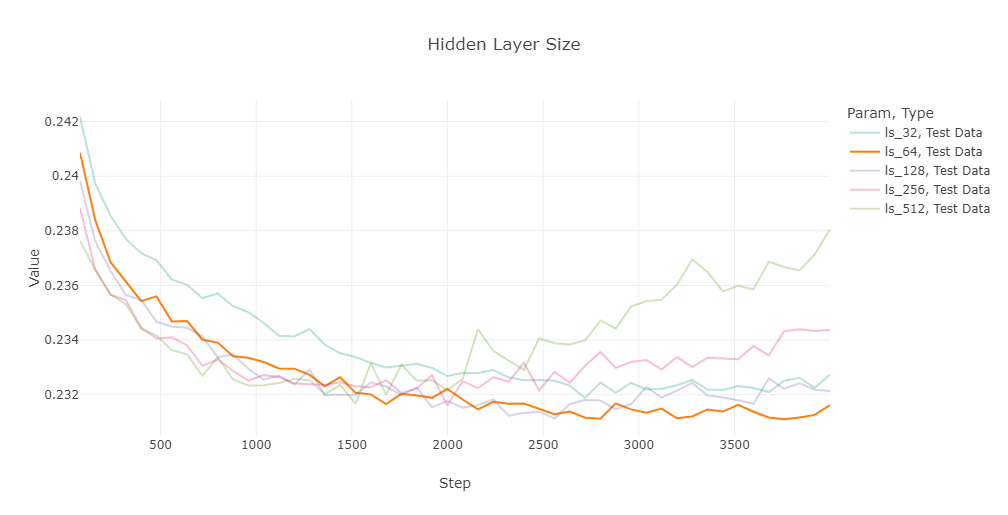


Figure 2: Best Number of Hidden Layers

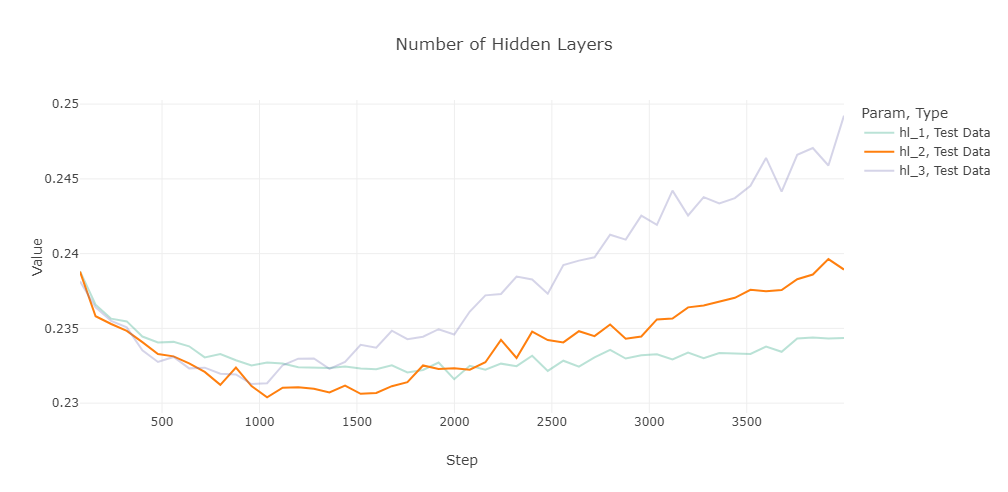


Figure 3: Best Dropout RateA picture containing text, diagram, plot, line

Description automatically generated

Figure 4: Best Activation Function

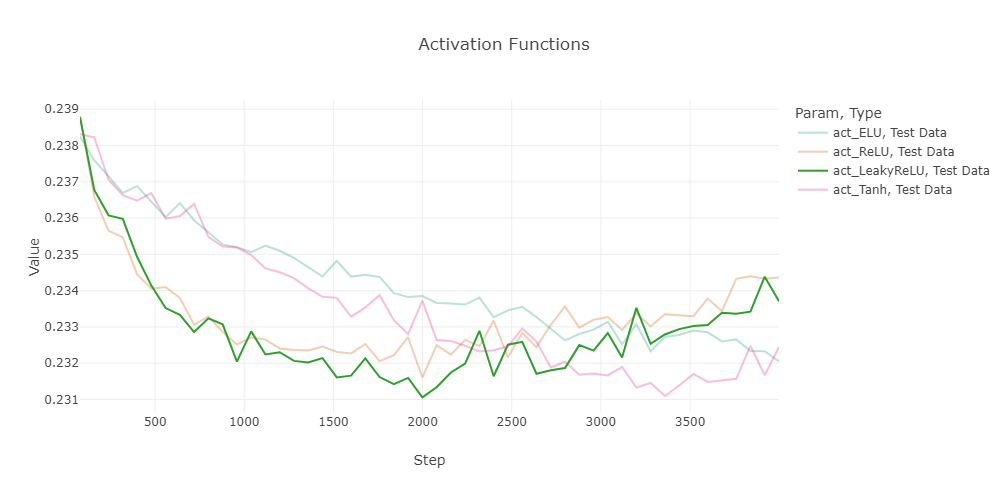


Figure 5: Best Optimizer

