**Predicting Customer Churn in Cellular Network Services Using Neural Networks: A Comprehensive Analysis**

Dorin Buzilov

sba20274

# Table of Contents

[**Table of Contents 2**](#_yryrm36z2q3p)

[**ABSTRACT 3**](#_qqs0fl9v6pw9)

[**Introduction 3**](#_e7tgeqtwcr5n)

[**Literature Review 4**](#_z02o573y5xqr)

[Conclusion of Literature Review 8](#_4hf051frpmrf)

[**Research Problem and Statement of Objectives 10**](#_cc7wiqwfazmz)

[Problem Definition 10](#_k7fcggu0zxbu)

[Research Objectives 10](#_plmqohq9gjku)

[**Data 11**](#_lemh8mz33en4)

[**Data Dictionary 11**](#_305gxlm92ud5)

[**Big Data 11**](#_8fpltw3ykw0r)

[**RDMS 12**](#_774muhe0h1i7)

[**HDFS 12**](#_v1bb87bb6q07)

[**Comparison between RDBMS and HADOOP 13**](#_mlwh23by0dft)

[**HDFS Processing Solutions 14**](#_sxa1cjs2mx95)

[Hadoop - MapReduce 14](#_1n3dwgdk9jmt)

[Apache Spark 14](#_g3ie6t56e4qm)

[MapReduce vs Spark: Performance 15](#_c5v125i03wkz)

[MapReduce vs Spark: Ease of Use 16](#_z7morage395g)

[Framework Selection 16](#_eicz0jiiybnw)

[Ecosystem Overview: 16](#_h50csdv5h28r)

[**Data Cleaning 17**](#_q822z5fd8g3)

[**Exploratory Data Analysis 17**](#_5fse5aii4y30)

[**Evaluation of Neural Networks 19**](#_ujqwtd7jd596)

[Model Selection 20](#_82359g8vkneb)

[**Model Performance 21**](#_6w60pnuopiuf)

[Upsampling 21](#_67vf2owi8t6d)

[**Evaluation 22**](#_88kjmk8k7a9f)

[**Conclusion 24**](#_2pmpcooy29rx)

[**References 25**](#_bob7xo9dhetf)

# ABSTRACT

This study investigates the phenomenon of customer churn in the context of cellular network services and proposes a novel approach utilizing neural networks for predictive analysis. With the rapid evolution of telecommunication technologies, retaining customers has become paramount for service providers to maintain competitiveness and sustainability. Leveraging a dataset encompassing diverse customer attributes and behavioral patterns, a multi-layered neural network model is developed to predict churn propensity accurately. Through extensive experimentation and comparative analysis, the efficacy of the neural network approach is evaluated against conventional methods, showcasing superior predictive performance and robustness. Furthermore, insights derived from feature importance analysis provide valuable understanding into the underlying factors driving customer churn. The findings of this research contribute to advancing the understanding of customer behavior in telecommunications and offer actionable insights for proactive churn management strategies, thereby empowering service providers to mitigate churn effectively and enhance customer retention efforts.

# Introduction

In today's highly competitive telecommunications industry, customer churn has emerged as a critical challenge for service providers. With the continuous evolution of telecommunication technologies and the increasing options available to consumers, retaining customers has become essential for maintaining competitiveness and ensuring long-term sustainability. Customer churn, referring to the phenomenon of customers switching from one service provider to another or discontinuing services altogether, not only leads to revenue loss but also undermines brand reputation and market position.

Given the significance of customer retention in the telecommunications sector, there is a growing interest in predictive analytics techniques to forecast churn propensity accurately. Traditional methods often rely on statistical models and heuristic approaches, which may have limitations in capturing the complex and dynamic nature of customer behavior. As a response, there is a need for more sophisticated and data-driven approaches that can leverage the wealth of customer data available to service providers.

In this context, this study seeks to investigate the phenomenon of customer churn in the cellular network services domain and propose a novel approach utilizing neural networks for predictive analysis. Neural networks, a class of machine learning algorithms inspired by the biological structure of the human brain, have demonstrated remarkable capabilities in handling complex patterns and relationships in data. By leveraging a dataset comprising diverse customer attributes and behavioral patterns, this research aims to develop a multi-layered neural network model that can accurately predict churn propensity.

The primary objective of this study is to evaluate the efficacy of the neural network approach in comparison to conventional methods for churn prediction. Through extensive experimentation and comparative analysis, the predictive performance and robustness of the neural network model will be assessed. Additionally, insights derived from feature importance analysis will provide valuable understanding into the underlying factors driving customer churn in the context of cellular network services.

# Literature Review

Building an effective customer churn prediction model using various techniques has become a decisive topic for business and academics in recent years. In order to understand how different studies have constructed their churn prediction models, this paper reviews some of the current studies as shown in Table 1.

**Table 2. Related Literature about Customer Churn**

| **Author(s)** | **Dataset** | **Dataset Size** | **Prediction Method** |
| --- | --- | --- | --- |
| Coussement and Van den Poel, 2008 | Newspaper subscriber data | 45,000 | Support Vector Machine,  Random Forest,  Logistic Regression |
| Hung, Yen and Wang, 2006 | Wireless telecommunication services | 160,000 | Classification (K-means clustering, Decision tree, Back Propagation Neural Network) |
| Sharma and Kumar Panigrahi, 2011 | Iranian telecom company customers | 2,427 | (ANN) Multi-Layer Perceptron |
| Qureshi et al., 2013 | Telecommunication traffic data | 106,000 | (ANN) Multi-Layer Perceptron  Logistic Regression,  Linear Regression,  K-means Clustering,  Decision Trees |
| Wu et al., 2021 | Dataset 1 from IBM customer status  Dataset 2 form Kaggle Telecom customer churn prediction competition 2020  Dataset 3 provided by Teradata center for customer relationship management at Duke University | Dataset 1 = 7,032  Dataset 2 = 4,031  Dataset 3 = 51,047 | Logistic Regression,  Decision Tree,  Random Forest,  Naive Bayes,  AdaBoost,  Multi-layer Perceptron,  K-means |
| Ahmad, Jafar and Aljoumaa, 2019 | SyriaTel Telecommunication company customer data | 70 Terabytes | Decision Tree,  Random Forest, Gradient Boost Machine Tree, XGBoost |
| Saleh and Saha, 2023 | Dataset 1 from IBM Telco,  Dataset 2 from Maven Telco,  Dataset 3 from Cell2Cell,  Datatset 3 collected from AAU students | Dataset 1 = 7,043  Dataset 2 = 4,601  Dataset 3 = 71,047  Dataset 4 = 288 | Random Forest,  AdaBoost,  Decision Tree,  Logistic Regression,  Extreme Gradient Boosting Classifier (XGBC) |

Coussement and Van den Poel (2008) conclude that SVMs are effective in predicting churn in subscription services, thanks to their ability to handle complex problems and noisy datasets by mapping non-linear inputs into high-dimensional spaces. Although SVMs outperform logistic regression with suitable parameter selection, they are surpassed by random forests. The study suggests using a grid search with 5-fold cross-validation for optimal parameter selection, enhancing SVM performance. While traditional methods like logistic regression remain viable, SVMs and random forests offer compelling alternatives when paired with proper parameter selection.

Hung,Yen and Wang (2006) study explores the use of neural networks, specifically back propagation networks (BPN), for predictive modeling. Drawing on previous research, the study suggests that a one-layer hidden layer and optimal network design may yield more accurate results. Using a 1-1-1 (input-hidden-output) training model with 43 inputs and one output, the study tests various combinations of key modeling parameters, such as the number of neurons in the hidden layer and learning rate. Results indicate that model (BPN) N21-R6 achieves the best performance based on R-square and MSE measurements. Neural network models outperform decision tree models in the initial 6 months, but a significant performance gap emerges between them afterward, raising concerns.

According to Sharma and Kumar Panigrahi, 2011 , their paper’s final model achieves over 92% overall accuracy in predicting customer churn. For future research, the paper proposes exploring dimensionality reduction and feature selection techniques during data pre-processing, incorporating other prediction techniques like support vector machines and genetic algorithms, and extending the churn prediction methodology to other sectors for comparison and accuracy testing.

In the paper by Qureshi et al., 2013 , they employed various machine learning algorithms, such as linear and logistic regression, Artificial Neural Networks, K-Means clustering, and decision trees (including CHAID, Exhaustive CHAID, CART, and QUEST), to categorize churners and active customers. They evaluated the results based on precision, recall, and F-measure. Among the algorithms tested, the best performance was achieved with the Exhaustive CHAID algorithm, a variant of the standard decision trees algorithm.

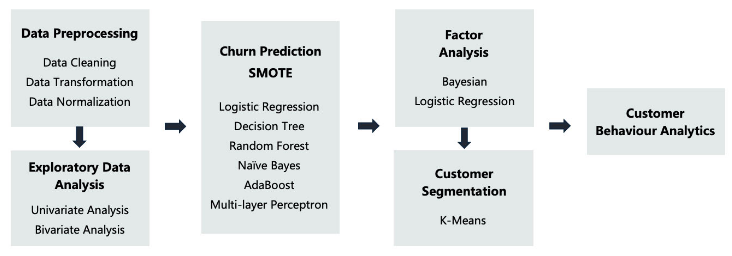
Decision Tree Variants:

1. **CHAID (Chi-squared Automatic Interaction Detection):** CHAID is a decision tree algorithm used for categorical target variables. It recursively splits the data based on categorical predictor variables by selecting the predictor that produces the most significant difference in the target variable, as determined by the chi-squared test.
2. **Exhaustive CHAID (Exhaustive Chi-squared Automatic Interaction Detection):** Exhaustive CHAID is an extension of CHAID that explores all possible splits at each node of the decision tree. This exhaustive search can lead to more accurate and optimal decision trees but may also be computationally intensive.
3. **CART (Classification and Regression Trees):** CART is a decision tree algorithm that can be used for both classification and regression tasks. Similar to CHAID, it recursively partitions the data based on predictor variables, but it selects the predictor and split point that minimize impurity (e.g., Gini impurity for classification or mean squared error for regression).
4. **QUEST (Quick, Unbiased, Efficient Statistical Tree):** QUEST is a decision tree algorithm designed for handling both categorical and continuous predictor variables. It constructs a decision tree by recursively partitioning the data based on statistical tests that maximize the homogeneity of the resulting subgroups.

Comparison:

* **CHAID vs. Exhaustive CHAID:** Both CHAID and Exhaustive CHAID are variants of the CHAID algorithm, but Exhaustive CHAID explores all possible splits at each node, potentially leading to more accurate trees at the cost of increased computational complexity.
* **CHAID/Exhaustive CHAID vs. CART vs. QUEST:** CHAID and CART are primarily used for classification tasks, while QUEST is versatile and can handle both classification and regression tasks. CART typically uses impurity measures like Gini impurity or entropy, while CHAID and QUEST use statistical tests like chi-squared or t-tests. Additionally, CHAID and QUEST can handle categorical predictor variables, while CART can handle both categorical and continuous predictors.

According to the paper published by Wu et al., 2021, churn management in telco operators is crucial but challenging due to customers' tendency to stay with their current operators. This leads to imbalanced datasets, impacting churn prediction accuracy. Advanced resampling methods like SMOTE and suitable evaluation metrics are explored in this research. Unlike previous studies focusing on either churn prediction or customer segmentation, this research integrates both. Churn prediction alone lacks understanding of underlying reasons, while segmentation alone neglects focus on churn customers. By combining both, an integrated telecom customer analytics framework for churn management is proposed, aiming to develop more effective retention programs and reduce management costs. The research achieves high F1-scores and AUC values for different datasets using various machine learning models such as AdaBoost, Random Forest, Multi-layer Perceptron, and Logistic Regression. Across the three datasets there was no conclusive advantage for one algorithm when compared to another. It is observed that when generalized among the three datasets Random Forest, Bayes AdaBoost and Multiple Layer Perceptron performed amongst the best. Conclusions drawn from the paper tell us that experimentation and hyperparameter tuning will be required to discover the optimal algorithm for our specific dataset. The use of SMOTE did not directly translate into higher accuracy or f-scores.



*Figure 2: Customer Churn & Segmentation Data Pipeline*

* **SMOTE** stands for "Synthetic Minority Over-sampling Technique." It is an algorithm used to address class imbalance in machine learning datasets, particularly in classification tasks where one class is significantly underrepresented compared to the other(s). SMOTE works by generating synthetic examples of the minority class by interpolating between existing minority class samples. It randomly selects a minority class sample and finds its k nearest neighbors within the feature space. Then, it generates new synthetic examples along the line segments connecting the selected sample to its neighbors. By creating synthetic examples, SMOTE helps balance the class distribution, which can improve the performance of machine learning models, particularly in scenarios where the minority class is of interest and its representation in the dataset is insufficient for effective learning.

The conclusion of Ahmad, Jafar and Aljoumaa, 2019 outlines the evaluation of different scenarios for churn prediction in a telecom company. It discusses the impact of varying sizes of training data and the use of statistical features versus Social Network Analysis (SNA) features.

* **SNA** stands for Social Network Analysis. It's a method used to study relationships and interactions between individuals, groups, or organizations. In the context of churn prediction in the telecom industry, SNA involves analyzing the social connections or relationships among customers to understand how these connections influence churn behavior. This analysis can provide insights into patterns of communication, influence, and social structure that may impact customer behavior and churn rates.

The analysis finds that increasing the volume of training data improves the performance of classification algorithms, with the addition of historical data providing marginal benefits. The optimal sliding window for extracting SNA features is identified as the last four months before the baseline. Integrating both statistical and SNA features significantly enhances the predictive model's performance, leading to a maximum AUC value of 93.3%. Additionally, the text highlights the importance of addressing the imbalance in the dataset and compares the performance of different classification algorithms. XGBOOST algorithm emerges as the top performer with an AUC value of 93.3%, making it the chosen classification algorithm for the predictive model. Overall, the integration of SNA features is shown to enhance churn prediction in the telecom industry.

## 

## 

## Conclusion of Literature Review

Results from the study conducted by Saleh and Saha, 2023 indicate that logistic regression (LR) and random forest (RF) consistently outperform other algorithms across various performance measures in different datasets. The study also examines the impact of data balancing techniques and identifies LR as particularly dependent on certain features like brand, which may not be recognized by other algorithms.

Of the surveyed literature, there is no consensus drawn on the ideal algorithm for churn prediction, however it is observed the five best-performing algorithms across multiple studies and scenarios for churn prediction are:

1. Random Forest (RF)
2. XGBOOST (Extreme Gradient Boosting)
3. AdaBoost (Adaptive Boosting)
4. Multi-layer Perceptron (MLP) Neural Network
5. Logistic Regression (LR)

These algorithms consistently demonstrated strong performance across various datasets and performance metrics, making them top choices for churn prediction tasks in the telecom industry. The purpose of this study is to demonstrate to which degree of accuracy can neural networks predict customer churn. The NN performance will be benchmarked against the four other algorithms to measure where hyperparameter tuning and resampling techniques increase or decrease the performance of the algorithm. It is also observed in the study conducted by Wu et al., 2021 that churn prediction and customer segmentation should be performed in tandem to effectively target customer churn. K-means clustering will be examined in its effectiveness for customer segmentation in this study.

Literature reports a few challenges faced by neural networks:

1. **Performance Gap**: Neural network models initially outperform decision tree models, but a significant performance gap emerges between them over time. This gap raises concerns about the long-term effectiveness and stability of neural network models for churn prediction tasks.
2. **Complexity and Design**: Achieving optimal performance with neural networks requires careful design and parameter tuning. The study suggests that a one-layer hidden layer and optimal network design may yield more accurate results. This implies that neural networks can be sensitive to architecture choices and parameter settings.
3. **Data Imbalance**: Neural networks may also be impacted by imbalanced datasets, which can affect their ability to effectively learn from the data. Imbalanced datasets can lead to biased predictions and reduced performance, highlighting the importance of addressing class imbalance issues in neural network training.

To address the challenges:

1. **Performance Gap**: I will perform continuous monitoring and evaluation of neural network models over time to help identify performance degradation and prompt adjustments or retraining. Experimentation with different architectures, activation functions, and regularization techniques may also help mitigate performance gaps.
2. **Complexity and Design**: I will conduct thorough experimentation and hyperparameter tuning to optimize the design of neural networks for specific churn prediction tasks. Techniques such as grid search, random search, or automated hyperparameter optimization algorithms can assist in finding the optimal network architecture and parameter settings.
3. **Data Imbalance**: I will employ strategies to handle imbalanced datasets, such as resampling techniques like SMOTE (Synthetic Minority Over-sampling Technique), can help alleviate the impact of class imbalance on neural network training. Additionally, using appropriate evaluation metrics that account for class imbalance, such as precision, recall, and F1-score, can provide a more comprehensive assessment of model performance. Regular monitoring of model performance and adjusting the training process as needed can also help address issues related to data imbalance.

# 

# 

# 

# 

# 

# 

# Research Problem and Statement of Objectives

## Problem Definition

The research problem proposed for this work can be clarified using the problem definition model as follows:

*Problem Identification:* The primary concern is predicting when customers will leave a cellular network service, termed as "churn." This is crucial for service providers to maintain customer satisfaction and revenue stability.

*Problem Clarification:* Comprehensively exploring the application of neural networks, a subset of machine learning algorithms, in predicting customer churn. Understanding the nuances of these models is essential for effective analysis.

*Problem Formulation:* The aim is to develop a robust framework that incorporates various factors influencing customer churn, such as usage patterns, customer demographics, and service quality metrics, into neural network models. The goal is to create a predictive tool that offers actionable insights for reducing churn rates.

## Research Objectives

To address the question the following questions are proposed:

*Objective 1:* Develop and optimize neural network models capable of accurately predicting customer churn based on historical data and relevant features.

*Objective 2:* Identify the most influential factors contributing to customer churn through comprehensive feature selection and analysis techniques.

*Objective 3:* Validate the developed models using real-world cellular network data and evaluate their performance in terms of prediction accuracy, reliability, and practical utility.

# 

# Data

This study uses the churn dataset from the UCI Repository of Machine Learning Databases at the University of California, Irvine. The churn dataset is randomly collected from an Iranian telecom company’s database over a 12 month period. It is a total of 3150 rows of data, each representing a customer with their information for 14 columns. The attributes that are in this dataset are call failures, frequency of SMS, number of complaints, number of distinct calls, subscription length, age group, the charge amount, type of service, seconds of use, status, frequency of use, and Customer Value. All of the attributes except for the attribute churn is aggregated data of the first 9 months. The churn labels are the generated state of the customers at the end of 12 months. The three months is a designated planning gap. The dataset is licensed under a Creative Commons Attribution 2.0 International (CC BY 4.0) license (Creative Commons, 2018). This allows for the sharing and adaptation of the datasets for any purpose, provided that the appropriate credit is given (UC Irvine, 2020).

# Data Dictionary

*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)

*Customer Value*: The calculated value of customer

*Age*: The age band of the customer corresponding to the age group ['30', '25', '15', '45', '55']

*Churn*: binary (1: churn, 0: non-churn) - Class label Customer

# Big Data

Most commonly, Big Data is a term that refers to data sets whose size (volume), complexity (variability), and rate of growth (velocity) makes them difficult to be captured, managed, processed or analyzed by conventional technologies or tools such as relational databases or desktop statistical tools or visualization packages. High volume, and high velocity and high variety of data make it compulsory for industry to adopt more flexible data management solutions. In order to understand this need we must appreciate the scale at which companies operate, examples include but not limited to; Meta which processes petabytes of data per day in many forms of messages & media, NASA preparing for space launches collecting sensor data, Rideshare apps like Uber collecting geo-spacial data as well as user data. The fact that data is always in motion presents an opportunity for extracting value (Khan, Fahim Uddin and Gupta, 2014) .

# RDMS

A relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model. Countless common databases presently in use are grounded on the relational database model. RDBMSs are a shared choice for storing information in new databases used for financial records, manufacturing and logistical information, personnel data, and other applications. RDBMS stock the data into assembly of tables, which might be linked by common fields (database table columns). RDBMS also offers relational operators to handle the data stored into the database tables. Most RDBMS use SQL as database query language. A significant feature of relational systems is that a particular database can be spread across several tables. This fluctuates from flat-file databases, in which all databases are self-reliant in a single table. Nearly all full-sized database systems are RDBMS. Small database systems, yet, use other designs that offer less flexibility in posing queries (Raphael and Kumar, 2016).

# HDFS

Hadoop includes a fault-tolerant storage system called Hadoop Distributed File System, or HDFS. HDFS is able to store huge amounts of information, scale up incrementally and survive the failure of significant parts of storage infrastructure without losing data. Hadoop creates clusters of machines and coordinates work among them. Clusters can be built with inexpensive computers. If one fails, Hadoop continues to operate the cluster without losing data or interrupting work, by shifting work to the remaining machines in the cluster. HDFS manages storage on the clusters by breaking incoming flies into pieces, called “blocks” and storing each of the blocks redundantly across the pool of servers. In the common case, HDFS stores three complete copies of each file by copying each piece to three different servers Bhosale and Gadekar (2014) .

# 

# 

# 

# Comparison between RDBMS and HADOOP

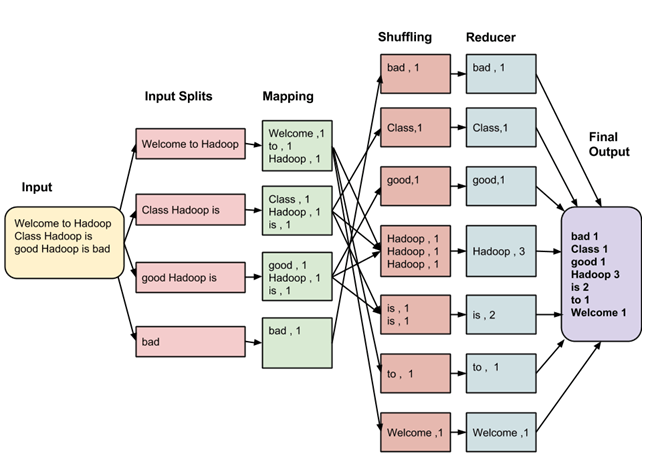
**Table 1. RDBMS and Hadoop Comparison**

|  | RDBMS | HADOOP |
| --- | --- | --- |
| Data Size | Gigabytes or Terabytes | Petabytes or Exabytes |
| Data Acceptance | Structured | Structured or Unstructured |
| Data Integrity | High (ACID properties) | Low |
| Data Schema | (Static) Required on Write | (Dynamic) Required on Read |
| Data Speed | Fast to Read | Fast to Write |
| Response Time | Near Immediate | Latency due to batch processing |
| Data Normalization | Required | Not-required |
| Scaling | Vertical | Horizontal |
| Scalability | RDBMS has scalability issues, because it scales by adding lots of RAM and CPU to a single or set of database servers. | Cheap to scale by adding low power machines which work in parallel to meet cluster demands |
| OLTP (OnLine Transactional Processing) | Supported | Not-supported |
| OLAP (OnLine Analytical Processing) | Supported | Supported |
| Best Fit | OLTP  Complex ACID transactions  Operational Data Storage | Data Discovery  Processing Unstructured Data  Massive Data Storage/Processing |

By the above comparative survey we have come to know that HADOOP is the best technique for handling Big Data compared to that of RDBMS. As the world moves on, the data used increases and therefore a better way of handling such a huge amount of data is becoming a tedious task. Analysis and storage of the so-called Big Data is handy only by the help of the new Hadoop eco-system than the traditional RDBMS being used till now. Hadoop is a large-scale, open source software framework dedicated to scalable, distributed, data-intensive computing. The framework breaks up large data into smaller parallelizable chunks and handles scheduling, maps each piece to an intermediate value, Fault tolerant, reliable, and supports thousands of nodes and petabytes of data, tried and tested in production, many implementation options.

# HDFS Processing Solutions

## Hadoop - MapReduce



*Figure 1: MapReduce Architecture*

The processing pillar in the Hadoop ecosystem is the MapReduce framework. The framework allows the specification of an operation to be applied to a huge dataset, divide the problem and data, and run it in parallel. From an analyst point of view this can occur on multiple dimensions. For example, a very large database can be divided into a smaller subset where analytics can be applied. In a traditional warehousing scenario this might entail applying an ETL operation on the data to produce something usable by the analyst. In Hadoop, these kinds of operations are written as MapReduce jobs in Java. There are a

number of higher level languages like Hive and Pig that make writing these programs easier. The outputs of these jobs can be written back to either HDFS or placed in a traditional data warehouse. There are two functions in MapReduce as follows:

***map*** – the function takes key/value pairs as input and generates an intermediate set of key/value pairs

***reduce*** – the function which merges all the intermediate values associated with the same intermediate key

## Apache Spark

Apache Spark is a unified analytics engine for large-scale data processing. Originally developed at UC Berkeley’s AMPLab, Spark was first released as an open source project in 2010. Spark uses the Hadoop MapReduce distributed computing framework as its foundation. Spark’s developers created it to improve upon several aspects of the MapReduce project, such as performance and ease of use, while preserving many of MapRerduce’s benefits.

The secret to Spark’s performance is that it runs in-memory on the cluster, and it isn’t bound to MapReduces’ two stage paradigm. That makes repeated access to the same data much faster. Spark can run as a standalone application or on top of Hadoop YARN, where it can read data directly from the HDFS. Spark includes a core processing engine, as well as libraries for SQL, machine learning, and stream processing. With API’s for Java, Scala, Python, and R, Spark enjoys a broad appeal among developers.

## MapReduce vs Spark: Performance

Apache Spark processes data in random access memory (RAM), while Hadoop MapReduce persists data back to the disk after a map or reduce action. Much like standard databases, Spark Loads a process into memory and keeps it there until further notice for the sake of caching. If you run Spark on Hadoop YARN with other resource-demanding services, or if the data is too large to fit entirely into memory, the Spark could suffer major performance degradation.

Map Reduce kills it’s processes as soon as the job is complete, so it can easily run alongside other services with minor performance differences.

Spark has the upper hand for iterative computations that pass over the same data many times. But when it comes to ETL-like jobs that’s when MapReduce excels.

According to the research in both (Shi et al., 2015) and (Ahmed et al., 2020) both studies concluded that overall Spark is at least 2x times faster than MapReduce for Word Count, k-means, and PageRank, re-

respectively. An exception to Spark’s performance advantage over MapReduce is the Sort workload, for which MapReduce is 2x faster than Spark. This is due to differences in task execution plans. MapReduce can over- lap the shuffle stage with the map stage, which effectively hides network overhead which is often a bottleneck for the reduce stage.

For typical machine learning (e.g., k-means and linear regression) and graph analytics (e.g., PageRank) algorithms, pars-ing text to objects is often the bottleneck for each iteration. Spark’s RDD caching addresses this issue effectively by reducing the CPU overhead for parsing.

Some notable observations from these studies include:

(1) Once tuned properly, the majority of workloads are CPU-bound for both MapReduce and Spark, and hence are scalable to the number of CPU cores.

(2) For MapReduce, the network overhead during a shuffle can be hidden by overlapping the map and reduce stages. For Spark, the intermediate data should always be compressed, because it's shuffle cannot be overlapped.

(3) For iterative algorithms in Spark, counter-intuitively, DISK ONLY configuration might be better than MEMORY ONLY. Because both of them address the object parsing bottleneck, but the former avoids GC and page swapping issues by eliminating memory consumption which makes it scalable to very large data sets

## MapReduce vs Spark: Ease of Use

Spark has pre-built APIs for Java, Scala, and Python, and also includes Spark SQL. It’s easy to write user-defined functions. Spark even includes an interactive mode for running commands with immediate feedback.

MapReduce is written in Java and is infamously very difficult to program. Apache Pig makes it easier, while Apache Hive adds SQL compatibility to the plate. MapReduce functions need to be written manually for any changes in data streaming, this can be time-consuming. In addition, MapReduce doesn’t have an interactive mode, although Hive includes a command-line interface.

## Framework Selection

In determining the best processing framework and data storage solution for predicting customer churn, there are three criteria which were analyzed and linked to the problem area.

**Data Volume and scalability:** The data volume at this level is small as we are working with a snapshot of the company’s data, however, any progress made at this level must be capable of being rolled-out to be used at a company wide level without any setbacks. With the needs of the company in mind, Apache Spark’s data streaming capabilities would be beneficial. It would allow for incremental processing of data and reacting to event’s in real-time, such as being able to offer customers thinking of leaving with deals tailored to their needs while minimizing cost as compared to their customer life-time value. Spark and MapReduce both offer horizontal scalability, however, Spark’s in-memory processing and optimized execution engine would offer better scalability.

**Latency Requirements:** Considering the latency of the churn prediction system. Apache Spark’s data streaming capabilities would be advantageous for low-latency and timely decision on the latest data.

**Complexity of Analysis:** As I will be applying machine learning models which can require iterative processing and optimization, then Spark’s support and its integration with machine learning libraries like MLib will be beneficial.

In conclusion Apache Spark’s data processing and data streaming capabilities, ease of use, and its ability to scale to meet the needs of real-time customer churn prediction, are the reasons for choosing it.

## Ecosystem Overview:

**Distributed Data Management:** Zookeeper

**Data Processing & Access:** Apache Spark

**Resource management:** YARN

**Data Storage:** HDFS

# Data Cleaning

The data was supplied by UCI Repository of Machine Learning Databases at the University of California, Irvine. The churn dataset is randomly collected from an Iranian telecom company’s database over a 12 month period. The dataset consists of 3150 rows and 14 columns. The dataset was inspected and required minimal cleaning to achieve the desired format required for model training. The feature labeled ‘Age’ was dropped due to duplicate information held within the ‘Age Group’ feature. The feature labeled Status was also dropped due to duplicate labeling of churn customers. All data entries were stored as string objects and required conversion to numerical format.

# Exploratory Data Analysis

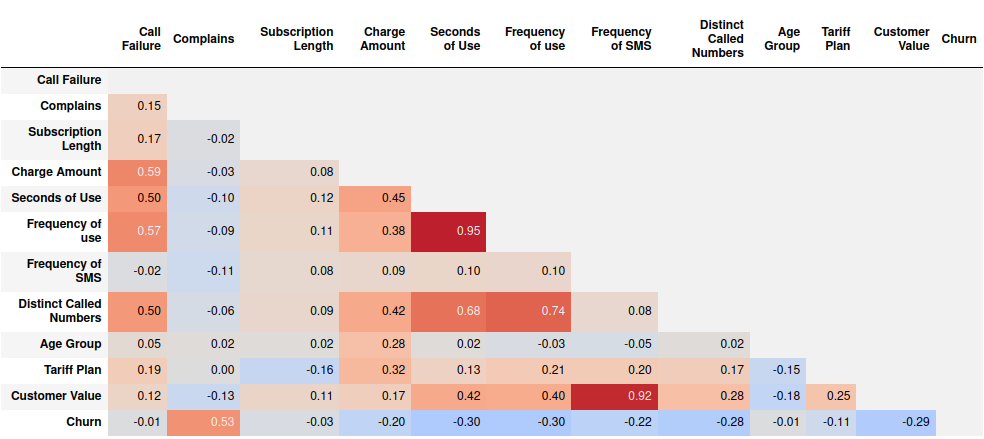
To identify driving features of churn in this dataset, correlation matrix and outlier analysis were performed to infer reasons for churn. The dataset is very small when compared to those of other papers, and unbalanced, only 495 churn customers compared to 2655 non-churn.



| Non-Churn (0) : | 2655 |
| --- | --- |
| Churn (1) : | 495 |

*Figure 2: Churn vs. Non-Churn Data*

The papers published by Wu et al., 2021 and Ahmad, Jafar and Aljoumaa, 2019 both concluded that there is a need to address imbalanced datasets either using SMOTE or other sampling methods. Upsampling was explored during the NN development phase and the outcomes will be addressed in the Model Performance section. The correlation matrix provides some insights into the major contributors of churn. Complaints suggest there may be strong indicators of churn prediction and customer value being highly correlated with the frequency of sms.



*Figure 3: Feature Correlation Corner Matrix*

The results of the analysis have produced insights into the factors surrounding churn.

* 79.5% of customers who logged a complaint have churned.
* 40.4% of all churn is linked to a customer complaint.
* Age Group: 25-45 years old (81.5% of complains come from 25-35 year old age category)
* Tariff Plan: 99% of complaints exist on Pay-as-you-go plan
* Subscription Length: On average 35 months.
* Customer Value: Customers who were associated with a complaint had an average Customer Value of 146 compared to the whole dataset average score of 470. This may seem like a low number, but when compared to the total churn data which has an average customer value score of 125, we can see that at the 50 percentile range the customer value score is 97 and jumps to 181 at the 75 percentile. In terms of capturing churned customers, those customers associated with a complaint actually represent the higher end of customer value.
* Frequency of SMS is the highest predictor of customer value, this is presuming that SMS carries the highest network charge and as a result drives up the company bottom line.

Addressing the causes of the complaints would retain these customers. As there is no narrative data on the complaint it is uncertain at this stage what are the causes. The scope of using a big data solution like apache spark will allow for the scaling of the analytics platform to accommodate narrative data in the future and be able to integrate NLP to address the reasons for the complaints.

# Evaluation of Neural Networks

There are several types of neural networks, each designed for specific tasks and data characteristics. Some of the most common types include:

1. **Feedforward Neural Networks (FNNs):** These are the simplest form of neural networks, where data flows in one direction, from input to output layer, without any cycles or loops.
2. **Convolutional Neural Networks (CNNs):** CNNs are primarily used for image recognition tasks. They consist of convolutional layers that extract features from input images and pooling layers that reduce dimensionality, followed by fully connected layers for classification.
3. **Recurrent Neural Networks (RNNs):** RNNs are designed to handle sequential data by maintaining internal memory states. They have connections that form directed cycles, allowing them to process sequences of inputs and capture temporal dependencies.
4. **Long Short-Term Memory (LSTM) Networks:** LSTM networks are a specialized type of RNNs that address the vanishing gradient problem. They have gates that control the flow of information, enabling them to capture long-term dependencies in sequential data.
5. **Gated Recurrent Unit (GRU) Networks:** GRUs are similar to LSTMs but have a simplified structure with fewer parameters. They are often used as alternatives to LSTMs for sequential data processing tasks.
6. **Autoencoder Neural Networks:** Autoencoders are unsupervised learning models that learn to encode input data into a lower-dimensional representation and then decode it back to the original data. They are commonly used for dimensionality reduction and feature learning.
7. **Generative Adversarial Networks (GANs):** GANs consist of two neural networks, a generator and a discriminator, that are trained simultaneously in a competitive manner. They are used for generating new data samples that are similar to the training data distribution.

When it comes to churn prediction, Recurrent Neural Networks (RNNs), particularly variants like Long Short-Term Memory (LSTM) networks, and Gated Recurrent Unit (GRU) networks, are often applicable. This is because churn prediction often involves analyzing sequential data, such as customer behavior over time, where capturing temporal dependencies is crucial. RNNs, with their ability to maintain memory states and process sequential data, are well-suited for such tasks. However, RNNs may suffer from vanishing or exploding gradients during training, especially on smaller datasets, FNNs typically do not face such issues.

Choosing a Feedforward Neural Network (FNN) for churn prediction or similar tasks can be advantageous for several reasons:

1. **Simplicity:** FNNs are relatively simple compared to other neural network architectures like recurrent or convolutional neural networks. They consist of layers of neurons arranged in a feedforward manner, making them easier to understand and implement.
2. **Universal Approximators:** FNNs have been proven to be universal function approximators, meaning they can approximate any function to arbitrary accuracy given a sufficiently large number of neurons in the hidden layers. This property makes them versatile for modeling complex relationships in data, including churn prediction.
3. **Efficient Training:** Training FNNs can be computationally efficient, especially for smaller datasets. Unlike recurrent neural networks, which may suffer from vanishing or exploding gradients during training, FNNs typically do not face such issues.
4. **Interpretability:** FNNs can offer some level of interpretability compared to more complex architectures like recurrent or convolutional neural networks. The sequential flow of data through layers makes it easier to understand how input features are transformed and combined to produce output predictions.
5. **Applicability to Tabular Data:** FNNs are well-suited for tabular data, which is common in churn prediction tasks. They can handle structured data with features represented as columns in a table format, making them a natural choice for such datasets.
6. **Availability of Tools and Libraries:** There are numerous tools and libraries available for building and training FNNs, such as TensorFlow, Keras, and PyTorch. These libraries provide pre-implemented neural network architectures and optimization algorithms, simplifying the development process.
7. **Flexibility:** FNNs offer flexibility in terms of architecture design. Researchers and practitioners can experiment with different numbers of layers, neurons, activation functions, and optimization techniques to tailor the network to the specific requirements of the churn prediction task.

From my literature review I found that Hung,Yen and Wang (2006) study explores the use of neural networks, specifically back propagation networks (BPN), for predictive modeling. Drawing on previous research, the study suggests that a one-layer hidden layer and optimal network design may yield more accurate results. Neural network models outperform decision tree models in the initial 6 months, but a significant performance gap emerges between them afterward, raising concerns.

## Model Selection

With all considerations for the advantages and disadvantages, this paper will explore the development of a Multi-layered perceptron neural network and utilize four other machine learning algorithms as comparison. Based on the literature review, the algorithms are:

1. Random Forest (RF)
2. XGBOOST (Extreme Gradient Boosting)
3. AdaBoost (Adaptive Boosting)
4. Logistic Regression (LR)

These algorithms consistently demonstrated strong performance across various datasets and performance metrics, making them top choices for churn prediction tasks in the telecom industry.

# Model Performance

A basic MLP model was developed with a 1-1-1 (input-hidden-output) layer setup. 12 inputs, 6 in the hidden layer and 1 output. This was then fine tuned using a hyperparameter testing pipeline to find the optimal setup.The parameters included the loss function, optimizers, accuracy parameters, number of hidden layers, how many nodes in each layer and how many epochs should the model run. The optimal model was determining to be:

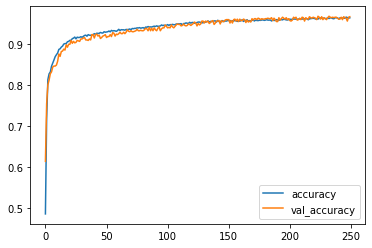
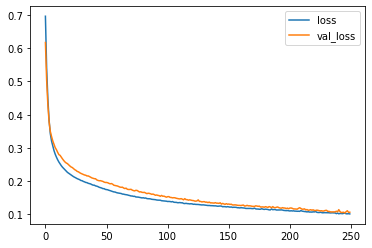
**Table 2. NN Optimized Hyperparameters**

| Optimizer | Nadam |
| --- | --- |
| Activation function Hidden Layer | Relu |
| Activation Output Layer | Sigmoid |
| Loss Function | binary\_crossentropy |
| Number of Hidden Layers | 1 |
| Nodes in Hidden Layer | 7 |
| Epoch | 250 |
| Training Set Accuracy | 95.4% |
| Testing Set Accuracy | 94.4% |

## 

## Upsampling

The dataset is suffering from class imbalance, where the number of churners is significantly smaller than non-churners. To address this issue, we will balance the target column by upsampling the minority class. This ensures that the model does not get biased towards the majority class during training. The same model performed better with the upsampled data scoring 95.67% on the testing data.

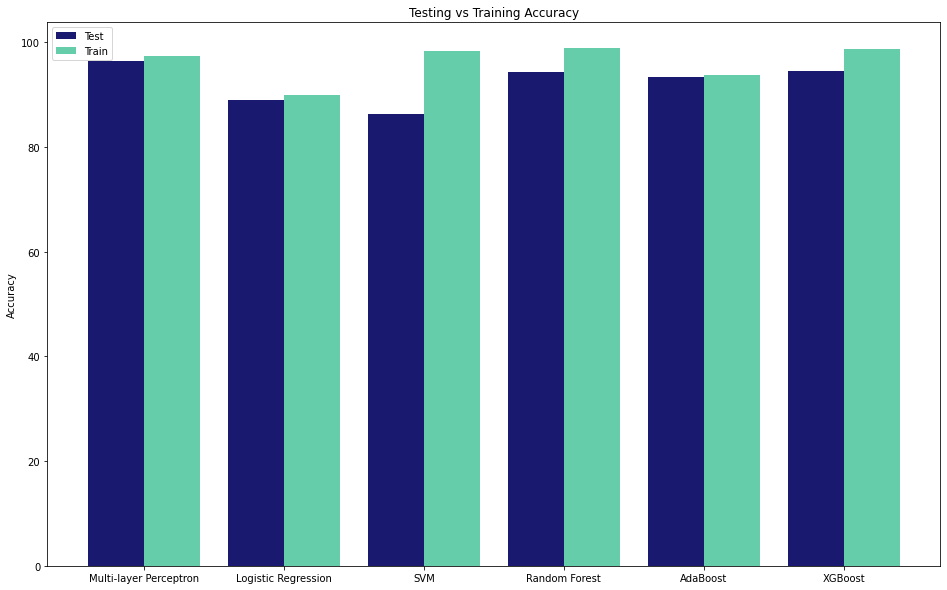


*Figure 4: Loss & Accuracy graphs*

**Table 3. Upsampling Comparison**

| Before | After |
| --- | --- |
| | Non-Churn (0) : | 2655 | | --- | --- | | Churn (1) : | 495 | | | Non-Churn (0) : | 2655 | | --- | --- | | Churn (1) : | 2655 | |

# Evaluation



*Figure 5: Model Results*

1. **Random Forest**, **AdaBoost**, **XGBoost** **and Multi-layer Perceptron** **(MLP)** achieved the highest accuracy scores, with values of 94.44%, 93.49%, 94.60% and 94.4%, respectively.
2. **Logistic Regression** also performed well, with accuracy scores of 89.05%.
3. **SVM** achieved a slightly lower accuracy score of 86.35%.

From these results, we can conclude the following:

1. **Multi-layer Perceptron** **(MLP)** performed well after upsampling unbalanced classes within the data. Most notably when compared to the ensemble methods, the MLP model maintained similar accuracy performance from training and test sets.
2. **Ensemble methods** like Random Forest, AdaBoost, and XGBoost tend to perform very well for customer churn prediction. These methods combine multiple weak learners to create a strong predictive model, which can capture complex relationships in the data effectively.
3. **Tree-based algorithms** such as Random Forest and XGBoost often excel in classification tasks like churn prediction due to their ability to handle non-linear relationships and interactions between features.
4. **Logistic Regression** also offers competitive performance, indicating that simpler models can still provide effective churn prediction when properly tuned and optimized.
5. **SVM** achieved the lowest accuracy score among the models tested, suggesting that it might not be the best choice for this particular churn prediction task given the dataset and problem complexity.

Overall, ensemble methods like Random Forest, AdaBoost, and XGBoost appear to be promising models for customer churn prediction based on their high accuracy scores. Multi-layer Perceptron (MLP) performed comparatively better than the models maintaining good performance throughout training and testing sets. As observed from research found during my literature review, there is great importance in hyperparameter tuning and using upsampling techniques to address class imbalances which will directly affect the performance of the model.

# Conclusion

The analysis of churn in this dataset reveals significant insights into the factors driving customer attrition. Despite the dataset's small size and class imbalance, valuable correlations between churn and various customer attributes have been identified. Notably, a high percentage of churned customers are associated with complaints, indicating a strong link between customer dissatisfaction and churn propensity. Addressing the root causes of customer complaints presents an opportunity for retention, although the lack of narrative data limits actionable insights at this stage. The potential integration of big data solutions like natural language processing (NLP) holds promise for scaling analytics platforms to accommodate narrative data, enabling deeper analysis of complaint reasons and more targeted retention strategies.

In evaluating neural network models for churn prediction, feedforward neural networks (FNNs) emerge as a promising choice due to their simplicity, universal approximation capabilities, and applicability to tabular data. While recurrent neural networks (RNNs) are commonly used for sequential data tasks like churn prediction, FNNs offer advantages in terms of interpretability, efficiency, and flexibility.

Comparative analysis of machine learning algorithms reveals that ensemble methods like Random Forest, AdaBoost, and XGBoost consistently achieve the highest accuracy scores for churn prediction. These models leverage the strengths of multiple weak learners to capture complex relationships in the data effectively. Additionally, simpler models like Multi-layer Perceptron (MLP) demonstrate competitive performance, highlighting the importance of proper tuning and optimization.

The observation that the dataset is relatively small compared to those used in the literature review raises concerns about the generalizability and reliability of the neural network's performance. Despite the dataset's size limitations, efforts to address class imbalances and normalize the data enhance the robustness of the neural network's performance. The normalization process ensures that data features are on a similar scale, preventing any individual feature from dominating the model's learning process. Furthermore, the establishment of a big data pipeline lays the foundation for expanding the neural network's capabilities to handle larger datasets effectively. By addressing these concerns and implementing strategies to enhance the dataset's quality and scalability, the research aims to maximize the neural network's predictive capabilities while ensuring its applicability to real-world scenarios.

Overall, ensemble methods such as Random Forest, AdaBoost, and XGBoost stand out as the most promising models for customer churn prediction, offering high accuracy and robust performance. Multi-layer Perceptron (MLP) performed considerably well after optimizing and upsampling, highlighting the importance of optimizing model hyperparameters. However, the selection of the final model should consider factors such as interpretability, computational complexity, and alignment with business requirements.

# References

Ahmad, A.K., Jafar, A. and Aljoumaa, K. (2019). Customer churn prediction in telecom using machine learning in big data platform. *Journal of Big Data*, [online] 6(1). doi:https://doi.org/10.1186/s40537-019-0191-6.

Ahmed, N., Barczak, A.L.C., Susnjak, T. and Rashid, M.A. (2020). A comprehensive performance analysis of Apache Hadoop and Apache Spark for large scale data sets using HiBench. *Journal of Big Data*, 7(1). doi:https://doi.org/10.1186/s40537-020-00388-5.

Bhosale, H. and Gadekar, D. (2014). A Review Paper on Big Data and Hadoop. *International Journal of Scientific and Research Publications*, [online] 4(10). Available at: https://www.ijsrp.org/research-paper-1014/ijsrp-p34125.pdf [Accessed 30 Jan. 2024].

Coussement, K. and Van den Poel, D. (2008). Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. *Expert Systems with Applications*, 34(1), pp.313–327. doi:https://doi.org/10.1016/j.eswa.2006.09.038.

Creative Commons (2018). *Creative Commons — Attribution 4.0 International — CC BY 4.0*. [online] Creativecommons.org. Available at: https://creativecommons.org/licenses/by/4.0/legalcode.

Hung, S.-Y., Yen, D.C. and Wang, H.-Y. (2006). Applying data mining to telecom churn management. *Expert Systems with Applications*, 31(3), pp.515–524. doi:https://doi.org/10.1016/j.eswa.2005.09.080.

Khan, M.A., Fahim Uddin, M. and Gupta, N. (2014). *Seven V’s of Big Data Understanding Big Data to Extract Value*. [online] sci-hub.se. Available at: https://sci-hub.se/https://ieeexplore.ieee.org/abstract/document/6820689 [Accessed 30 Jan. 2024].

Qureshi, S.A., Rehman, A.S., Qamar, A.M., Kamal, A. and Rehman, A. (2013). *Telecommunication subscribers’ churn prediction model using machine learning*. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICDIM.2013.6693977.

Raphael, R. and Kumar, R. (2016). *Big Data, RDBMS and HADOOP - A Comparative Study*. [online] *International Journal of Science and Research*. IJSR. Available at: https://www.ijsr.net/archive/v5i3/NOV162167.pdf.

Saleh, S. and Saha, S. (2023). Customer retention and churn prediction in the telecommunication industry: a case study on a Danish university. *SN Applied Sciences*, 5(7). doi:https://doi.org/10.1007/s42452-023-05389-6.

Sharma, A. and Kumar Panigrahi, P. (2011). A Neural Network based Approach for Predicting Customer Churn in Cellular Network Services. *International Journal of Computer Applications*, 27(11), pp.26–31. doi:https://doi.org/10.5120/3344-4605.

Shi, J., Qiu, Y., Farooq Minhas, U., Jiao, L., Wang, C. and Reinwald, B. (2015). *Clash of the Titans: MapReduce vs. Spark for Large Scale Data Analytics*. [online] Available at: https://www.vldb.org/pvldb/vol8/p2110-shi.pdf [Accessed 28 Mar. 2024].

Tobin, D. (2021). *Spark vs Hadoop MapReduce: 5 Key Differences*. [online] Integrate.io. Available at: https://www.integrate.io/blog/apache-spark-vs-hadoop-mapreduce/.

UC Irvine (2020). *UCI Machine Learning Repository*. [online] archive.ics.uci.edu. Available at: http://archive.ics.uci.edu/dataset/563/iranian+churn+dataset [Accessed 21 Mar. 2024].

Wu, S., Yau, W.-C., Ong, T.-S. and Chong, S.-C. (2021). Integrated Churn Prediction and Customer Segmentation Framework for Telco Business. *IEEE Access*, 9, pp.62118–62136. doi:https://doi.org/10.1109/access.2021.3073776.

I acknowledge the use of CHAT GPT for the purpose of assisting me in re-phrasing and restructuring sentences.