1. INTRODUCTION

1.1 Telecommunication Industry

Data from Telecommunication Industry is used in this research to predict the churn rate for a Telecom Company and identify features for explainability,

The global telecommunications market is continually transforming on account of the ongoing innovations and developments taking place consistently and at a fast pace. 5G trials and its deployment in the market will be a key focus in the years to come. Carriers are expected to strive for improving their network and offering expanded services to their customers by network densification and making use of small cells. Installing more fiber infrastructure and enhancing spectrum efficiency will also be the target for carriers.

There are various factors fuelling the growth of the global telecommunication market including, the advanced technology, intense market competition, and high investments in new telecommunication technologies such as wireless communication and satellite. Some of the other factors behind the growth of the market worldwide include: affordability of services, innovative services such as e-agriculture and e-education, and demand for high speed internet. On the other hand, the high cost of value added services may restrict the growth of the market. In addition to this, maintaining security will also pose a challenge.

Services provided by key players in the telecommunication market include providing storage area networks, storage products, storage networking services, entry level servers, enterprise networking services, 3G services, calling cards, broadband networks, and application networking services. The growing number of internet users worldwide has been increasing exponentially by the day and thus, within the telecommunications market, the market for internet-based services is currently thriving and is at its peak.

Various telecommunication products and services, increased popularity of the Internet of Things (IoT), and substantial technological advances are some of the major reasons for the growth of the global telecommunication market. Smart devices such as smartphones, tablets, smart watches, smart meters, sensors, smart buildings, and smart monitoring devices are being increasingly deployed across the globe. The smart cities projects worldwide is one instance where smart devices such as smart sensors and meters are used. The growing popularity of IoT that has enabled machine-to-machine communication, connecting innumerable smart devices such as air conditioners, refrigerators, music systems, food processors, and the like. All this has stoked the demand in the telecommunication market.

Factors such as cloud computing, increased coverage of satellites, and the development of 4G and 3G cellular networks are some other prominent growth drivers of the market for telecommunication products and services, worldwide.

On the other hand, stringent government policies can pose a roadblock to the market's growth. However, the increasing deployment of intelligent transportation systems (ITS) is likely to fuel the demand for these services, nullifying the negative effects of these restraints.

The telecommunication market can be broadly segmented into internet service providers (ISP), telephones, satellites, and cable communication. On the basis of usage, entertainment, point-to-point communication, infotainment, news, internetenabled services, and critical communication are the major categories

1.2 Customer Churn

The rate of customer retention measures the number of subscriptions cancelled by your customers within a certain period of time.For subscription companies, where all customers pay the same amount, customer and revenue will be equal, and the distinction between customer and income tax rate is moot. Note that while customer and revenue growth rates are very close to each other for each segment, the total rate of customer and revenue growth is very different.

In order for a company to expand its customer base, its growth rate (number of new customers) must exceed its rate (number of lost customers). Customer notice refers to the number of customers who have terminated their subscription within a certain period of time. Therefore, Revenue sharing (usually called MRR Churn) is more important than Customer or User sharing.

For example, it is very common to define churn rates at customer level (customer churn), subscription level (product churn) and level of recurring revenue (arm or ARR - weighted down). In addition, measuring churn can be complicated by low turnover, high turnover, high growth rates, different types of customers, the variety of renewal periods for subscriptions, the variety of contract MRR values and changes in the rate of churn itself over time. However, even if churn rate is large, we can improve the accuracy of churn rate calculation by choosing a measurement interval, where the total churn within the range is small. Too much aggregation over unjustified customer populations can distort the calculation of churn rate.

Similar to the subscriber's Churn, the income ratio determines the percentage of income lost in due to cancellations within a certain period of time. Also in this case, as in the case of customer allocation, we do not take into account income from new subscriptions during the calculation of the allocation.

Revenue growth is usually lower than that of customers when calculated at the same time. Just like gross income, company uses the revenues they lose from cancellations, but also takes into account the revenues they earns from customers during that period.

Customer discharge is a major problem for any business, but companies with business models based on subscriptions are particularly at risk for their harmful effects. The churn can have a paralysing effect on business innovation: 36 % of subscriptions believe that the threat of high turnover increases the search for more dynamic and disruptive service models.

The fight against postponement and increasing customer loyalty is essential to sustain market growth and ensure the long - term health of any subscription business. Net income also takes into account revenue growth as a result of higher - cost subscriptions and is a critical indicator of the health of operations. If the net income is close to zero, the growth of the upsells is eliminated, so all growth must come from new customers. When short - term turnover is exceptionally high, company check sales and marketing funnel to see if they are targeting the right segment of customers

In addition, the rate of churn connects directly to the lifetime value of the customer (LTV) for subscription. Therefore, reducing the churn is one of the most important steps company can take to improve the performance of subscription business.

Many companies focus on reducing voluntary deviations, e. g. on improving customer satisfaction. Banks, telecom companies, insurance companies, energy companies are among the many types of companies that often use customer analysis and retention rates as one of their most important business metrics. Customer retention is the percentage of contract customers or subscribers who leave a provider for a certain period of time. Companies usually charge monthly fees for customers, but it can also be done quarterly or annually. Companies use customer travel analysis to improve their ability to identify high - risk customers and thus reduce the number of customers.

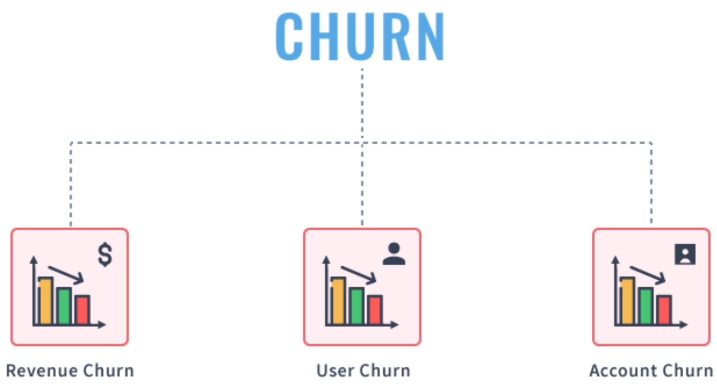


Figure 1 Types of Churn

By understanding and quantifying what is most important to customers, company can consistently deliver superior customer experience and measure their impact on churn customers and other important quantitative data, such as online sales, repeat purchase rates and more.

A very effective idea is to combine remuneration with important indicators of customer experience, such as the rate of resignation. Because many companies operate on a multi - level business model or offer freemium subscriptions, some customers are inherently more valuable than others.

companies are often most interested in seeing customer loyalty affects business revenues, it may be useful to calculate customer loyalty in terms of revenues. Companies use churn analysis to measure the rate at which customers leave their products, facilities or services.

There are many things that brands can do wrong, from complex onboarding when customers do not get easy - to - understand information about the use of the product and its ability to communicate poorly, such as lack of feedback or delayed responses to requests. The use of predictive modelling cases goes beyond proactive involvement with potential customers and the selection of effective retention actions.

Sales, customer success and marketing teams can also use data analysis expertise to align their actions. For example, if a customer shows signs of risk reduction, it is probably not a good time for the sale to provide information about additional services that the customer may be interested in

2. LITERATURE REVIEW

2.1 Machine Learning

Machine learning is a data analysis technique that teaches computers to do what is natural for humans and animals: learn from experience. Machine learning algorithms use computational methods to "learn" information directly from data without relying on a predetermined equation as a model. Machine learning algorithms find natural patterns in data that generate insight and helps in making better decisions and forecasts.

Just as there are almost unlimited machine learning applications, there is no lack of automatic learning algorithms. As automatic learning becomes increasingly important for business operations and AI is becoming more and more useful in business environments, the wars on the machine learning platform will only intensify. In today's world, every successful system has an automatic Learning algorithm at its core.

The processes involved in automatic learning are comparable to those of data mining and predictive modelling. Banks and other financial companies use machine learning technology for two key purposes: identifying key data information and preventing fraud. Machine learning is based on the ability to use computers to probe data for structures, although researchers do not have a theory of how the structure looks. Because machine learning often takes an iterative approach to data learning, learning can be easily automated.

In - depth learning combines the advances in computing power and special types of neural networks to learn complex patterns in large amounts of data. Data mining uses a variety of machine learning methods, but with a variety of purposes, machine learning also uses data mining methods as "unsupervised" or as a pre - processing step to improve the accuracy of the model. Context generalization is the ability of a machine to accurately execute new, invisible examples and tasks after experiencing a set of learning data.

There are different approaches to learning machines, from the use of base decision - making trees to the clustering, to the layers of artificial neural networks (the latter gave way to deep learning), depending on the task researchers are trying to perform and the type and amount of data they have at disposal.

2.2 Deep Learning

Deep Learning is a sub - area of Machine Learning that deals with algorithms inspired by the structure and function of the brain called artificial neural networks. Using the complementary processes, we create a fast and greedy algorithm capable of learning the deep, targeted networks of beliefs one layer at a time, provided that the two upper layers form an unprotected, associative memory. Recent advances in deep learning have improved to such an extent that deep learning outperforms people in certain tasks, such as classifying objects in images.

Deep learning models are trained using large data sets and neural network architectures, which learn functions directly from data without the need to extract the functions manually. Shallow learning refers to the methods of machine learning that multiply at a given level when more examples and training data are added to the network.

Deep learning is an artificial intelligence function that mimics the functioning of the human brain in data processing and creating patterns to be used in decision making. Deep learning, a subset of machine learning, uses a hierarchical level of artificial neural networks to complete the machine learning process.

While traditional programs construct linear analyses with data, the hierarchical function of deep learning systems enables machines to process data with non - linear approaches. Because deep learning algorithms require a ton of data to be learned, the increase in data creation is a reason for the increase in deep learning opportunities in recent years. In addition to increasing data creation, algorithms in - depth learning benefit from the increased computing power available today, as well as the spread of Artificial Intelligence (AI) as a Service.

The deep Learning toolkit allows to train CNN from scratch or use a pre - training model for transmission Learning. Advanced deep neural network models can be used to quickly apply deep learning to problems by learning to move or extract functions. Gpus love parallel processing and the modern approach in - depth Learning is very suitable for modern graphics cards.

High quality, robust, user - friendly, open and free libraries of codes to support in - depth Learning research such as Theano and Torch and some others have emerged, and it has led to rapid research into Deep Learning.

In particular, recent advances in deep neural networks, in which multiple layers of knots are used to gradually make more abstract representations of data, have allowed artificial neural networks to learn concepts such as categories of objects directly from raw sensory data.

Deep learning methods are multi - level representational methods, achieved by composing simple but non - linear modules, each of which turns the representation at a single level (starting with the raw input) into a representation at a higher, slightly more abstract level. Deep learning enables computational models consisting of multiple layers of processing to learn the representation of data with different levels of abstraction.

2.3 Neural Networks

Deep Learning is a category of machine Learning models algorithms using multi - layered neural networks. The tensorflow collects machine learning and deep learning models and algorithms (also known as neural networks) and makes them useful through a common metaphor. Systems specifically optimized for Deep Learning studios and support transparent, multi - GPU training (up to 4), Cluster analysis is a basic element of machine Learning and data science. 24

In - depth learning uses algorithms known as Neural Networks, inspired by the way in which biological nervous systems such as the brain are processed. The consumption of TensorFlow through Keras installs Deep Learning (also known as Deep structured Learning, hierarchical Learning or Deep Learning) is a class of algorithms for machine Learning, which use a cascade of many layers of non - linear processing for the extraction and transformation of functions. Deep Learning cluster monitoring allows monitor Deep Learning through the monitoring service Ganglia and TensorBoard, a visualization tool provided by TensorFlow.

Automatic learning algorithms usually look for optimal data rendering with a certain feedback signal (also known as an objective loss function). However, most algorithms for automatic learning only have the ability to use one or two layers of data transformation to learn the output display.

Like other algorithms for automatic learning, deep neural networks (DNN) perform learning by mapping the functions to targets through a process of simple data transformation and feedback signals, but DNN emphasizes learning the next layers of meaningful representation.

Neural networks were created in the field of computer science to answer questions that the normal statistical approach was not designed to answer. In the traditional machine learning approach, it is necessary to define the characteristics of data before modelling.

Deep learning is therefore a subfield of machine learning, which is a set of algorithms inspired by the structure and function of the brain, commonly known as Artificial Neural Networks (ANN).Deep learning is one of the latest trends in machine learning at this time, and there are many applications with which deep learning is manifested, such as robotics, image recognition and Artificial Intelligence (AI).

Convolutional neural networks (CNN) have extensive applications in image and video recognition, recurring neural networks (RNN) are used with voice recognition, and long - term memory networks (LTSM) are moving forward with automated robotics and machine translation.

Keras is an Open Source library of Neural networks written in Python that runs on top of the de or Tensorflow. Keras High - Level API manages the way we create models, define layers or configure multiple input output models. Keras does not handle Low - Level apis, such as computer graphics, tensors or other variables, because it has been handled by the rear - end engine. Backend is a term in Keras that performs all low - level calculations such as tensor products, convolutions and much more with the help of other libraries such as Tensorflow or Theano. Because Keras has an integrated support for parallel data so that it can process large amounts of data and speed up the time it takes to train them.

2.4 Artificial Neural Networks (ANN)

Artificial neuronal networks (ANNs) are statistical models directly inspired and partly based on biological neural networks In addition, there are intriguing benefits by combining artificial neural networks with other computational models (FDM, FEM, FVM), which can provide data for the development of the artificial neural network. In recent years, a combination of genetic algorithms and artificial neural networks has been demonstrated in connection with the development of hybrid methods.

Although neural models have been used for decades for tasks such as voice processing and image recognition, their widespread and intensive use in NLP is relatively new. Artificial Neural Networks are used for "unclear" solutions, where accuracy is not always necessary (or possible), such as weather forecasting.

Neural network functions have predictive power, which means that they can correctly predict future values in a time sequence, react and adapt to complex and unexpected stimuli, and perform classification tasks.

The training of artificial neural networks could be expressed as a problem of optimization of functions, which aims to identify the best network parameters (e. g. internal network weight and bias), minimizing errors in the initial data set. However, such resource savings can only be used if the network is implemented on hardware, while its software simulations are suffering from poor performance and lack of analysis and synthesis tools compared to the "traditional" artificial deterministic network.

In addition to statistical techniques, neural networks and in - depth learning, the concepts and techniques of signal processing, including non - linear processing and transformation. As already mentioned, the input data is converted by artificial neurons or processing units through the layers of the deep - learning neural network.

Artificial neural networks (ANNs) and more complex in - depth learning techniques are some of the most powerful tools for solving very complex problems and will continue to develop and use them in the future. ANNs are composed of multiple nodes, which imitate biological neurons of human brain. The neurons are connected by links and they interact with each other. The nodes can take input data and perform simple operations on the data. The result of these operations is passed to other neurons. The output at each node is called its activation or node value. Each link is associated with weight. ANNs are capable of learning, which takes place by altering weight values. The following illustration shows a simple ANN-

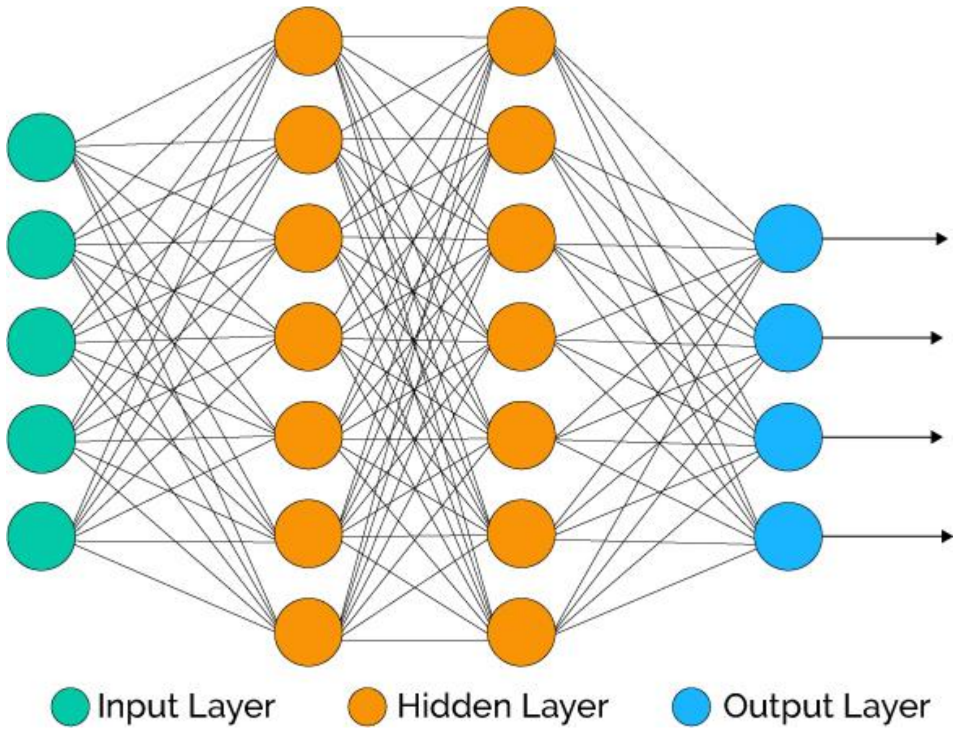


Figure 2 Layers of ANN

2.5 R Programming

R is a programming language and software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians and data miners for developing statistical software and data analysis. Polls, surveys of data miners, and studies of scholarly literature databases show that R's popularity has increased substantially in recent years. R is a GNU package. The source code for the R software environment is written primarily in C, FORTRAN and R. R is freely available under the GNU General Public License, and pre-compiled binary versions are provided for various operating systems. While R has a command line interface, there are several graphical front- ends available.

R and its libraries implement a wide variety of statistical and graphical techniques, including linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, and others. R is easily extensible through functions and extensions, and the R community is noted for its active contributions in terms of packages. Many of R's standard functions are written in R itself, which makes it easy for users to follow the algorithmic choices made. For computationally intensive tasks, C, C++, and FORTRAN code can be linked and called at run time.

Advanced users can write C, C++, Java, .NET or Python code to manipulate R objects directly. R is highly extensible through the use of user-submitted packages for specific functions or specific areas of study. Due to its S heritage, R has stronger object-oriented programming facilities than most statistical computing languages. Extending R is also eased by its lexical scoping rules. Another strength of R is static graphics, which can produce publication-quality graphs, including mathematical symbols. Dynamic and interactive graphics are available through additional packages.

R has Rd, its own LaTeX-like documentation format, which is used to supply comprehensive documentation, both on-line in a number of formats and in hard copy.

2.5.1 R Packages

R packages are a collection of R functions, complied code and sample data. They are stored under a directory called "library" in the R environment. By default, R installs a set of packages during installation. More packages are added later, when they are needed for some specific purpose. When we start the R console, only the default packages are available by default. Other packages which are already installed have to be loaded explicitly to be used by the R program that is going to use them. R Packages used in this project and their usage-

- **recipes**- Pre-processing data
- **rsample- S**ampling data

- **yardstick-** Package to estimate how well models are working using tidy data principals
- **keras -** Ports Keras from Python enabling deep learning in R
- **LIME-** Used to explain the predictions of black box classifiers. Deep Learning falls into this category

  The general approach lime takes to achieving goal is as follows:

  - For each prediction to explain, permute the observation n times.
  - Let the complex model predict the outcome of all permuted observations.
  - Calculate the distance from all permutations to the original observation.
  - Convert the distance to a similarity score.
  - Select m features best describing the complex model outcome from the permuted data.
  - Fit a simple model to the permuted data, explaining the complex model outcome with the m features from the permuted data weighted by its similarity to the original observation.
  - Extract the feature weights from the simple model and use these as explanations for the complex models local behaviour.
- **tidyquant-** Loads the tidyverse (dplyr, ggplot2, etc) and has visualization functions with theme_tq()
- **rsample-** Package for generating resamples.
- **corrr**: Tidy methods for correlation

3. RESEARCH METHODOLOGY

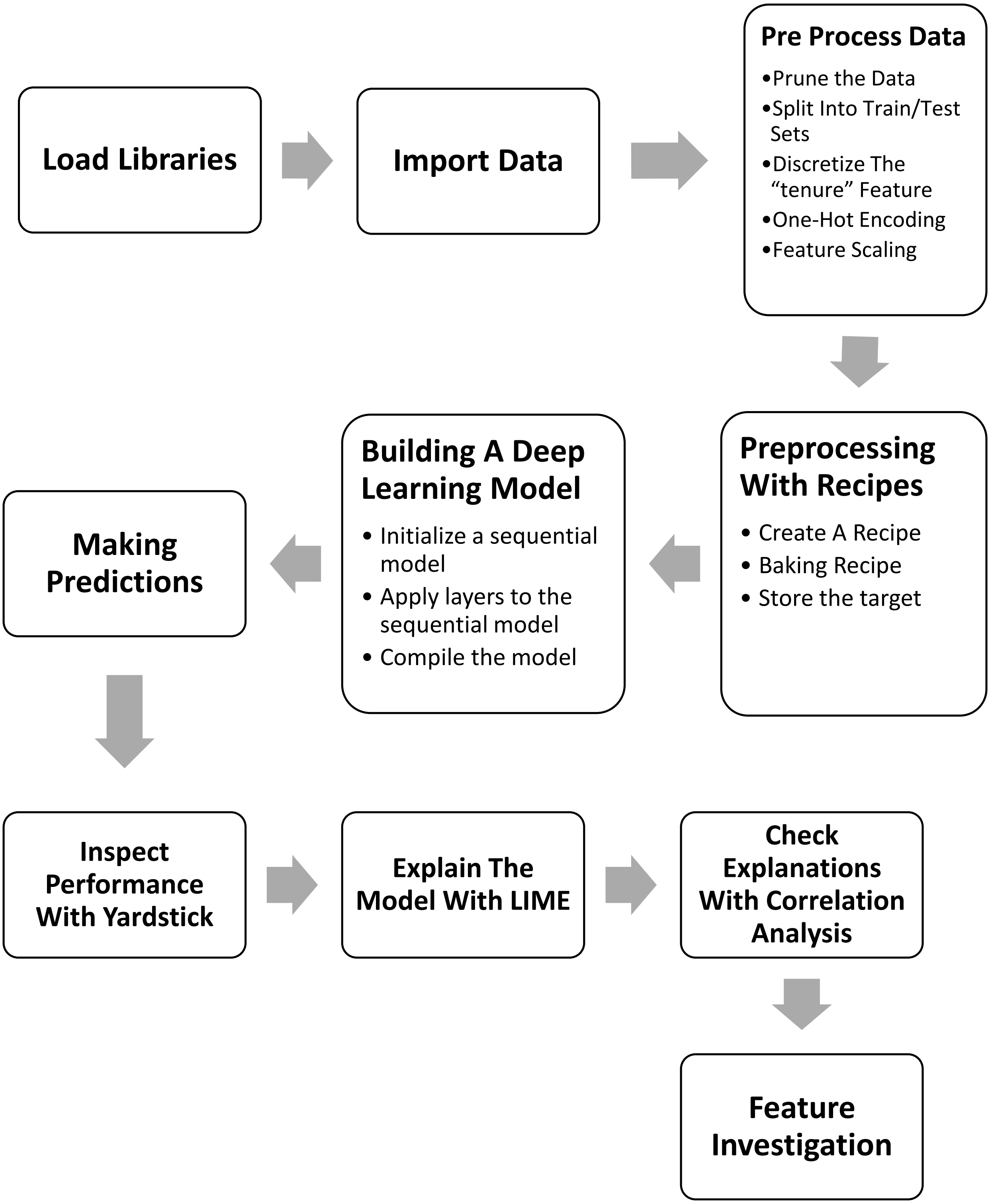


Figure 3 Steps of building Churn Model

4. CHURN MODELING WITH ARTIFICIAL NEURAL NETWORKS (KERAS)

4.1 Churn Modeling Overview

In this model keras is used to develop a sophisticated and highly accurate deep learning model in R. Preprocessing steps, formatting the data for Keras , inspection of various classification metrics is performed to create an ANN Model. ANN model is then explained with LIME. Then ANN LIME results is cross-checked with the a Correlation Analysis using the corrr package.

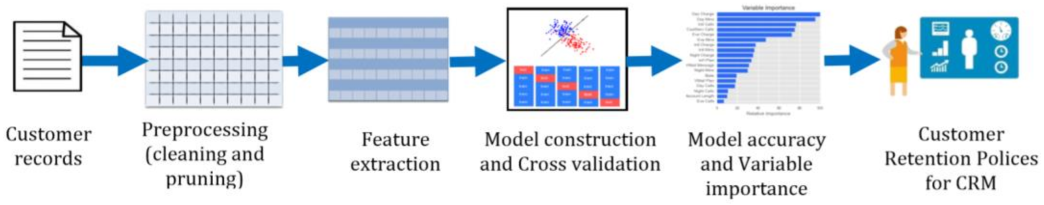


Figure 4 Churn Model Overview

4.2 Scope of the Study

The study is limited to dataset of Telecom Company that was obtained from IBM data source and is specific to that company only. Although the model can be replicated to develop ANN for another dataset.

4.3 Objective of the Study

The Objective of Churn Modeling is

- Building A Deep Learning Model
- Making Predictions about customers who are likely to terminate the subscription
- Inspect performance of Model
- Explain machine learning model classifiers.

4.4 Data Collection Source

Dataset is obtained from IBM Watson Telco Customer Data Set

The dataset includes information about:

Customers who left within the last month: The column is called Churn

Services that each customer has signed up for: phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies

Customer account information: how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges

Demographic info about customers: gender, age range, and if they have partners and dependents

4.5 Loading Libraries

```
# Load libraries
library(keras)
library(lime)
library(tidyquant)
library(rsample)
library(recipes)
library(yardstick)
library(corrr)
```

4.6 Importing Data

Functions Used

read_csv() - import the data into a tidy data frame

glimpse() - quickly inspect the data

```
# Import data
churn_data_raw <- read_csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")
```

```
glimpse(churn_data_raw)
## Observations: 7,043
## Variables: 21
## $ customerID       <chr> "7590-VHVEG", "5575-GNVDE", "3668-QPYBK"...
## $ gender           <chr> "Female", "Male", "Male", "Male", "Femal...
## $ SeniorCitizen    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ Partner          <chr> "Yes", "No", "No", "No", "No", "No", "No...
## $ Dependents       <chr> "No", "No", "No", "No", "No", "No", "Yes...
## $ tenure           <int> 1, 34, 2, 45, 2, 8, 22, 10, 28, 62, 13, ...
## $ PhoneService     <chr> "No", "Yes", "Yes", "No", "Yes", "Yes", ...
## $ MultipleLines    <chr> "No phone service", "No", "No", "No phon...
## $ InternetService  <chr> "DSL", "DSL", "DSL", "DSL", "Fiber optic...
## $ OnlineSecurity   <chr> "No", "Yes", "Yes", "Yes", "No", "No", "...
## $ OnlineBackup     <chr> "Yes", "No", "Yes", "No", "No", "No", "Y...
## $ DeviceProtection <chr> "No", "Yes", "No", "Yes", "No", "Yes", "...
## $ TechSupport      <chr> "No", "No", "No", "Yes", "No", "No", "No...
## $ StreamingTV      <chr> "No", "No", "No", "No", "No", "Yes", "Ye...
## $ StreamingMovies  <chr> "No", "No", "No", "No", "No", "Yes", "No...
## $ Contract         <chr> "Month-to-month", "One year", "Month-to-...
## $ PaperlessBilling <chr> "Yes", "No", "Yes", "No", "Yes", "Yes", ...
## $ PaymentMethod    <chr> "Electronic check", "Mailed check", "Mai...
## $ MonthlyCharges   <dbl> 29.85, 56.95, 53.85, 42.30, 70.70, 99.65...
## $ TotalCharges     <dbl> 29.85, 1889.50, 108.15, 1840.75, 151.65,...
## $ Churn            <chr> "No", "No", "Yes", "No", "Yes", "Yes", "...
```

4.7 Preprocessing Data

4.7.1 Pruning the Data

drop_na() –Dropping observations

```
# Remove unnecessary data
churn_data_tbl <- churn_data_raw %>%
    select(-customerID) %>%
    drop_na() %>%
    select(Churn, everything())
```

```
glimpse(churn_data_tbl)
## Observations: 7,032
## Variables: 20
## $ Churn            <chr> "No", "No", "Yes", "No", "Yes", "Yes", "...
## $ gender           <chr> "Female", "Male", "Male", "Male", "Femal...
## $ SeniorCitizen    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ Partner          <chr> "Yes", "No", "No", "No", "No", "No", "No...
## $ Dependents       <chr> "No", "No", "No", "No", "No", "No", "Yes...
## $ tenure           <int> 1, 34, 2, 45, 2, 8, 22, 10, 28, 62, 13, ...
## $ PhoneService     <chr> "No", "Yes", "Yes", "No", "Yes", "Yes", ...
## $ MultipleLines    <chr> "No phone service", "No", "No", "No phon...
## $ InternetService  <chr> "DSL", "DSL", "DSL", "DSL", "Fiber optic...
## $ OnlineSecurity   <chr> "No", "Yes", "Yes", "Yes", "No", "No", "...
## $ OnlineBackup     <chr> "Yes", "No", "Yes", "No", "No", "No", "Y...
## $ DeviceProtection <chr> "No", "Yes", "No", "Yes", "No", "Yes", "...
## $ TechSupport      <chr> "No", "No", "No", "Yes", "No", "No", "No...
## $ StreamingTV      <chr> "No", "No", "No", "No", "No", "Yes", "Ye...
## $ StreamingMovies  <chr> "No", "No", "No", "No", "No", "Yes", "No...
## $ Contract         <chr> "Month-to-month", "One year", "Month-to-...
## $ PaperlessBilling <chr> "Yes", "No", "Yes", "No", "Yes", "Yes", ...
## $ PaymentMethod    <chr> "Electronic check", "Mailed check", "Mai...
## $ MonthlyCharges   <dbl> 29.85, 56.95, 53.85, 42.30, 70.70, 99.65...
## $ TotalCharges     <dbl> 29.85, 1889.50, 108.15, 1840.75, 151.65,...
```

4.7.2 Splitting into Train/Test Sets

```
# Split test/training sets
set.seed(100)
train_test_split <- initial_split(churn_data_tbl, prop = 0.8)
train_test_split
## <5626/1406/7032>
```

Retrieving training and testing sets using training() and testing() functions.

```
# Retrieve train and test sets
```

```
train_tbl <- training(train_test_split)

test_tbl  <- testing(train_test_split)
```

4.8 Exploratory Data Analysis

```
Hmisc::describe(train_data)
```

OUTPUT

```
## train_data
##
##  20  Variables      5627  Observations
## ---------------------------------------------------------------------------
## Churn
##        n  missing distinct
##     5627        0        2
##
## Value         No   Yes
## Frequency   4131  1496
## Proportion 0.734 0.266
## ---------------------------------------------------------------------------
## gender
##        n  missing distinct
##     5627        0        2
##
## Value      Female   Male
## Frequency    2757   2870
## Proportion   0.49   0.51
## ---------------------------------------------------------------------------
## SeniorCitizen
##        n  missing distinct     Info      Sum     Mean      Gmd
##     5627        0        2    0.412      925   0.1644   0.2748
##
## ---------------------------------------------------------------------------
## Partner
##        n  missing distinct
##     5627        0        2
##
## Value         No   Yes
## Frequency   2907  2720
## Proportion 0.517 0.483
## ---------------------------------------------------------------------------
## Dependents
##        n  missing distinct
##     5627        0        2
##
## Value         No   Yes
## Frequency   3935  1692
## Proportion 0.699 0.301
## ---------------------------------------------------------------------------
## tenure
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     5627        0       72    0.999    32.39    28.08        1        2
##      .25      .50      .75      .90      .95
##        9       29       55       69       72
##
## lowest :  1  2  3  4  5, highest: 68 69 70 71 72
## ---------------------------------------------------------------------------
## PhoneService
##        n  missing distinct
##     5627        0        2
##
## Value         No   Yes
## Frequency    546  5081
## Proportion 0.097 0.903
## ---------------------------------------------------------------------------
```

```
## MultipleLines
##        n  missing distinct
##     5627        0        3
##
## Value                    No No phone service              Yes
## Frequency              2705              546             2376
## Proportion            0.481            0.097            0.422
## ---------------------------------------------------------------------------
## InternetService
##        n  missing distinct
##     5627        0        3
##
## Value              DSL Fiber optic          No
## Frequency         1936        2472        1219
## Proportion       0.344       0.439       0.217
## ---------------------------------------------------------------------------
## OnlineSecurity
##        n  missing distinct
##     5627        0        3
##
## Value                       No No internet service                 Yes
## Frequency                 2806                1219                1602
## Proportion               0.499               0.217               0.285
## ---------------------------------------------------------------------------
## OnlineBackup
##        n  missing distinct
##     5627        0        3
##
## Value                       No No internet service                 Yes
## Frequency                 2485                1219                1923
## Proportion               0.442               0.217               0.342
## ---------------------------------------------------------------------------
## DeviceProtection
##        n  missing distinct
##     5627        0        3
##
## Value                       No No internet service                 Yes
## Frequency                 2460                1219                1948
## Proportion               0.437               0.217               0.346
## ---------------------------------------------------------------------------
## TechSupport
##        n  missing distinct
##     5627        0        3
##
## Value                       No No internet service                 Yes
## Frequency                 2765                1219                1643
## Proportion               0.491               0.217               0.292
## ---------------------------------------------------------------------------
## StreamingTV
##        n  missing distinct
##     5627        0        3
##
## Value                       No No internet service                 Yes
## Frequency                 2259                1219                2149
## Proportion               0.401               0.217               0.382
## ---------------------------------------------------------------------------
## StreamingMovies
##        n  missing distinct
##     5627        0        3
##
## Value                       No No internet service                 Yes
## Frequency                 2211                1219                2197
## Proportion               0.393               0.217               0.390
## ---------------------------------------------------------------------------
## Contract
##        n  missing distinct
##     5627        0        3
##
## Value      Month-to-month       One year       Two year
## Frequency            3103           1167           1357
## Proportion          0.551          0.207          0.241
## ---------------------------------------------------------------------------
## PaperlessBilling
##        n  missing distinct
```

```
##     5627        0        2
##
## Value         No   Yes
## Frequency   2316  3311
## Proportion 0.412 0.588
## ---------------------------------------------------------------------------
## PaymentMethod
##        n  missing distinct
##     5627        0        4
##
## Value      Bank transfer (automatic)   Credit card (automatic)
## Frequency                       1242                      1214
## Proportion                     0.221                     0.216
##
## Value               Electronic check              Mailed check
## Frequency                       1871                      1300
## Proportion                     0.333                     0.231
## ---------------------------------------------------------------------------
## MonthlyCharges
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     5627        0     1503        1    64.75    34.41    19.65    20.05
##      .25      .50      .75      .90      .95
##    35.33    70.30    89.97   102.82   107.12
##
## lowest :  18.25  18.40  18.55  18.70  18.75, highest: 118.20 118.35 118.60 118.65 118.75
## ---------------------------------------------------------------------------
## TotalCharges
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     5627        0     5286        1     2277     2442    49.91    84.78
##      .25      .50      .75      .90      .95
##   396.20  1395.05  3783.20  5970.08  6908.14
##
## lowest :   18.80   18.85   18.90   19.00   19.05
## highest: 8547.15 8564.75 8594.40 8672.45 8684.80
## ---------------------------------------------------------------------------
```

4.9 Discretizing Data

```
# get the numerical features

num_features <- select_if(train_data, is.numeric) %>%  names()
library(ggthemes)
train_data %>%
  select(Churn, one_of(num_features)) %>%
  gather(key = key, value = value,-Churn) %>%
  ggplot(aes(x = value, fill = Churn))+
  geom_histogram(alpha = 0.7)+
  facet_wrap(~ key, scales = 'free')+
  theme_stata()+
  scale_fill_tableau()
```

OUTPUT:

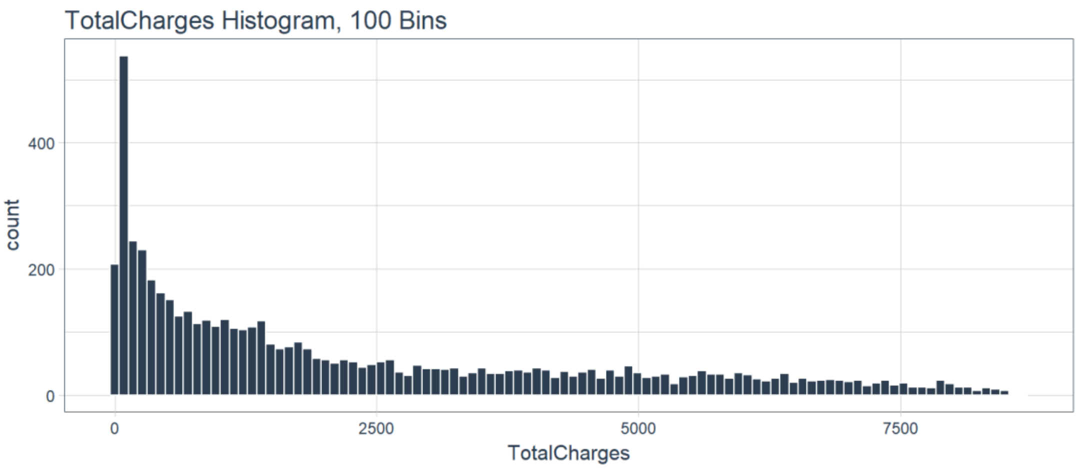
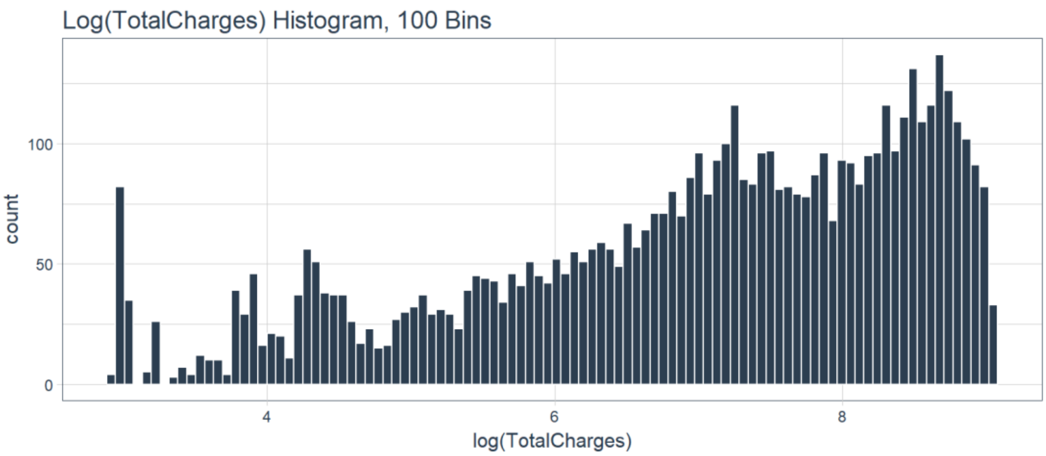
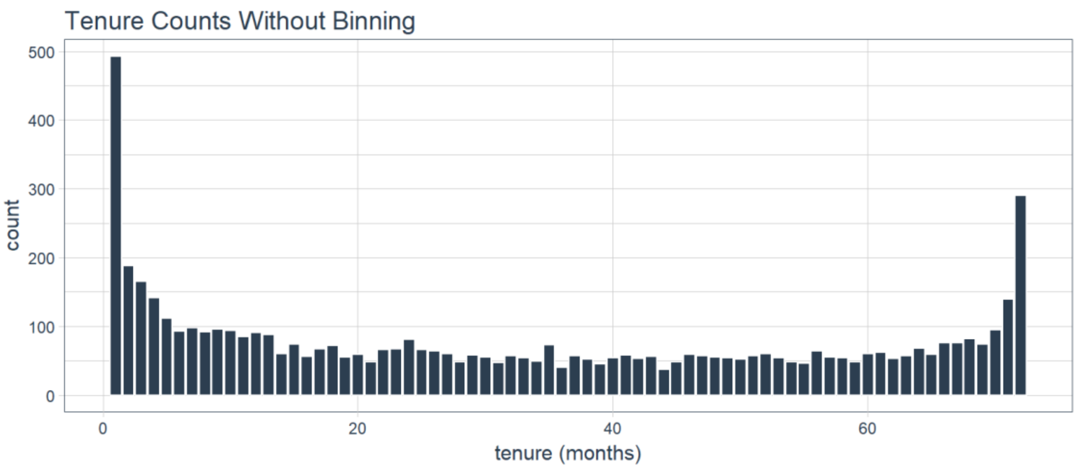
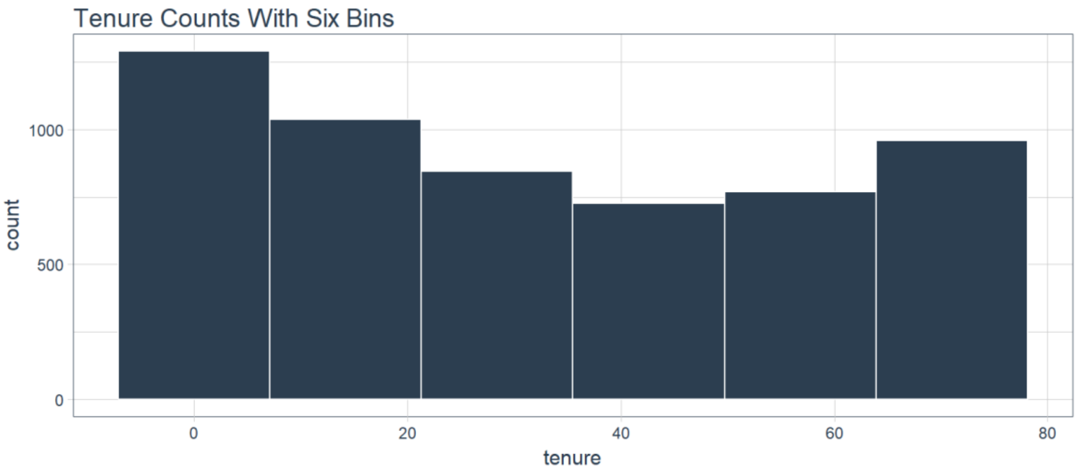


Figure 5 Total Charges Histogram







4.10 One Hot Encoding

One-hot encoding is the process of converting categorical data to sparse data, which has columns of only zeros and ones (this is also called creating “dummy variables” or a “design matrix”). All non-numeric data will need to be converted to dummy variables.

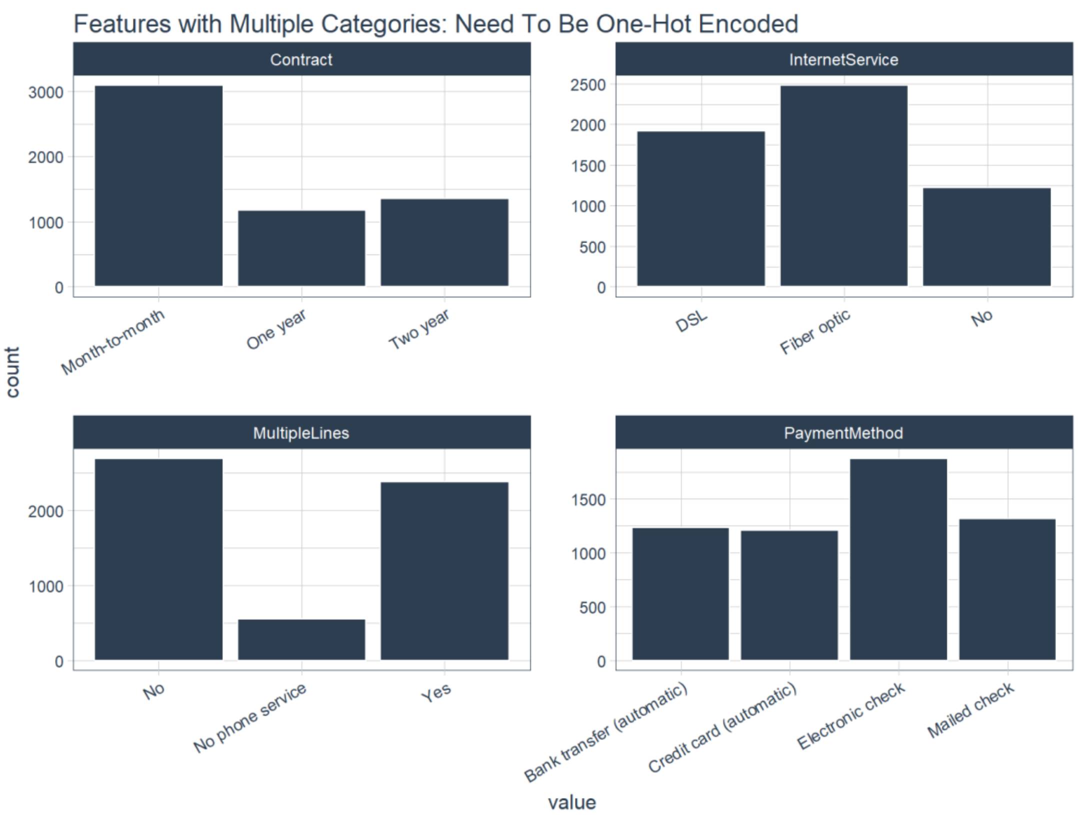


Figure 6 One Hot Encoding Chart

4.11 Transforming Data with recipe library

4.11.1 Creating the recipe

A “recipe” is a series of steps to perform on the training, testing and/or validation sets.

```
# Create recipe
rec_obj <- recipe(Churn ~ ., data = train_tbl) %>%
    step_discretize(tenure, options = list(cuts = 6)) %>%
    step_log(TotalCharges) %>%
    step_dummy(all_nominal(), -all_outcomes()) %>%
    step_center(all_predictors(), -all_outcomes()) %>%
    step_scale(all_predictors(), -all_outcomes()) %>%
    prep(data = train_tbl)
## step 1 discretize training
## step 2 log training
## step 3 dummy training
## step 4 center training
## step 5 scale training
```

OUTPUT:

```
## Data Recipe
## ## Inputs:
##
##       role #variables
##    outcome          1
##  predictor         19
##
## Training data contained 5626 data points and no missing data.
##
## Steps:
##
## Dummy variables from tenure [trained]
## Log transformation on TotalCharges [trained]
## Dummy variables from ~gender, ~Partner, ... [trained]
## Centering for SeniorCitizen, ... [trained]
```

4.11.2 Applying the recipe with bake() function

```
# Predictors
x_train_tbl <- bake(rec_obj, newdata = train_tbl)
x_test_tbl  <- bake(rec_obj, newdata = test_tbl)

glimpse(x_train_tbl)
```

OUTPUT:

```
## Observations: 5,626
## Variables: 35
## $ SeniorCitizen                       <dbl> -0.4351959, -0.4351...
## $ MonthlyCharges                      <dbl> -1.1575972, -0.2601...
## $ TotalCharges                        <dbl> -2.275819130, 0.389...
## $ gender_Male                         <dbl> -1.0016900, 0.99813...
## $ Partner_Yes                         <dbl> 1.0262054, -0.97429...
## $ Dependents_Yes                      <dbl> -0.6507747, -0.6507...
## $ tenure_bin1                         <dbl> 2.1677790, -0.46121...
## $ tenure_bin2                         <dbl> -0.4389453, -0.4389...
## $ tenure_bin3                         <dbl> -0.4481273, -0.4481...
## $ tenure_bin4                         <dbl> -0.4509837, 2.21698...
## $ tenure_bin5                         <dbl> -0.4498419, -0.4498...
## $ tenure_bin6                         <dbl> -0.4337508, -0.4337...
## $ PhoneService_Yes                    <dbl> -3.0407367, 0.32880...
## $ MultipleLines_No.phone.service      <dbl> 3.0407367, -0.32880...
## $ MultipleLines_Yes                   <dbl> -0.8571364, -0.8571...
## $ InternetService_Fiber.optic         <dbl> -0.8884255, -0.8884...
## $ InternetService_No                  <dbl> -0.5272627, -0.5272...
## $ OnlineSecurity_No.internet.service  <dbl> -0.5272627, -0.5272...
## $ OnlineSecurity_Yes                  <dbl> -0.6369654, 1.56966...
## $ OnlineBackup_No.internet.service    <dbl> -0.5272627, -0.5272...
## $ OnlineBackup_Yes                    <dbl> 1.3771987, -0.72598...
## $ DeviceProtection_No.internet.service <dbl> -0.5272627, -0.5272...
## $ DeviceProtection_Yes                <dbl> -0.7259826, 1.37719...
## $ TechSupport_No.internet.service     <dbl> -0.5272627, -0.5272...
## $ TechSupport_Yes                     <dbl> -0.6358628, -0.6358...
## $ StreamingTV_No.internet.service     <dbl> -0.5272627, -0.5272...
```

```
## $ StreamingTV_Yes                       <dbl> -0.7917326, -0.7917...
## $ StreamingMovies_No.internet.service   <dbl> -0.5272627, -0.5272...
## $ StreamingMovies_Yes                   <dbl> -0.797388, -0.79738...
## $ Contract_One.year                     <dbl> -0.5156834, 1.93882...
## $ Contract_Two.year                     <dbl> -0.5618358, -0.5618...
## $ PaperlessBilling_Yes                  <dbl> 0.8330334, -1.20021...
## $ PaymentMethod_Credit.card..automatic. <dbl> -0.5231315, -0.5231...
## $ PaymentMethod_Electronic.check        <dbl> 1.4154085, -0.70638...
## $ PaymentMethod_Mailed.check            <dbl> -0.5517013, 1.81225...
```

4.11.3 Storing the actual values as y_train_vec and y_test_vec

```
# Response variables for training and testing sets
y_train_vec <- ifelse(pull(train_tbl, Churn) == "Yes", 1, 0)
y_test_vec  <- ifelse(pull(test_tbl, Churn) == "Yes", 1, 0)
```

4.12 Deep Learning Model

```
# Building our Artificial Neural Network
model_keras <- keras_model_sequential()

model_keras %>%
    # First hidden layer
    layer_dense(
        units              = 16,
        kernel_initializer = "uniform",
        activation         = "relu",
        input_shape        = ncol(x_train_tbl)) %>%
    # Dropout to prevent overfitting
    layer_dropout(rate = 0.1) %>%
    # Second hidden layer
    layer_dense(
        units              = 16,
        kernel_initializer = "uniform",
        activation         = "relu") %>%
    # Dropout to prevent overfitting
    layer_dropout(rate = 0.1) %>%
```

```
    # Output layer
    layer_dense(
        units              = 1,
        kernel_initializer = "uniform",
        activation         = "sigmoid") %>%
    # Compile ANN
    compile(
        optimizer = 'adam',
        loss      = 'binary_crossentropy',
        metrics   = c('accuracy')
    )
model_keras
```

OUTPUT:

```
## Model
## ______________________________________________________________________
## Layer (type)                   Output Shape               Param #
## ======================================================================
## dense_1 (Dense)                (None, 16)                 576
## ______________________________________________________________________
## dropout_1 (Dropout)            (None, 16)                 0
## ______________________________________________________________________
## dense_2 (Dense)                (None, 16)                 272
## ______________________________________________________________________
## dropout_2 (Dropout)            (None, 16)                 0
## ______________________________________________________________________
## dense_3 (Dense)                (None, 1)                  17
## ======================================================================
## Total params: 865
## Trainable params: 865
## Non-trainable params: 0
## ______________________________________________________________________
```

Fitting the keras model to the training data

```
fit_keras <- fit(
    object           = model_keras,
    x                = as.matrix(x_train_tbl),
    y                = y_train_vec,
    batch_size       = 50,
    epochs           = 35,
    validation_split = 0.30
    )
# Print the final model
fit_keras
## Trained on 3,938 samples, validated on 1,688 samples (batch_size=50, ep
ochs=35)
## Final epoch (plot to see history):
## val_loss: 0.4215
##  val_acc: 0.8057
##     loss: 0.399
##      acc: 0.8101
```

4.13 Making Predictions

- predict_classes: Generates class values as a matrix of ones and zeros. Since we are dealing with binary classification, we'll convert the output to a vector.
- predict_proba: Generates the class probabilities as a numeric matrix indicating the probability of being a class. Again, we convert to a numeric vector because there is only one column output.

```
# Predicted Class
yhat_keras_class_vec <- predict_classes(object = model_keras, x = as.matrix(x_test
_tbl)) %>%
    as.vector()


# Predicted Class Probability
yhat_keras_prob_vec  <- predict_proba(object = model_keras, x = as.matrix(x_test_t
bl)) %>%
    as.vector()
```

4.14 Inspecting Performance With Yardstick

Using the fct_recode() function from the forcats package to assist with recoding as Yes/No values.

```
# Format test data and predictions for yardstick metrics
estimates_keras_tbl <- tibble(
    truth      = as.factor(y_test_vec) %>% fct_recode(yes = "1", no = "0"),
    estimate   = as.factor(yhat_keras_class_vec) %>% fct_recode(yes = "1", no = "0
"),
    class_prob = yhat_keras_prob_vec
)

estimates_keras_tbl
```

OUTPUT:

```
## # A tibble: 1,406 x 3
##     truth estimate  class_prob
##    <fctr>   <fctr>       <dbl>
##  1    yes       no 0.328355074
##  2    yes      yes 0.633630514
##  3     no       no 0.004589651
##  4     no       no 0.007402068
##  5     no       no 0.049968336
##  6     no       no 0.116824441
##  7     no      yes 0.775479317
##  8     no       no 0.492996633
##  9     no       no 0.011550998
## 10     no       no 0.004276015
## # ... with 1,396 more rows
```

4.15 Explaining Model with LIME Package

- model_type: Used to tell lime what type of model we are dealing with. It could be classification, regression, survival, etc.
- predict_model: Used to allow lime to perform predictions that its algorithm can interpret.

```
class(model_keras)
## [1] "keras.models.Sequential"
## [2] "keras.engine.training.Model"
## [3] "keras.engine.topology.Container"
## [4] "keras.engine.topology.Layer"
## [5] "python.builtin.object"
```

```
# Setup lime::model_type() function for keras
model_type.keras.models.Sequential <- function(x, ...) {
    return("classification")
}
```

```
# Setup lime::predict_model() function for keras
predict_model.keras.models.Sequential <- function(x, newdata, type, ...) {
    pred <- predict_proba(object = x, x = as.matrix(newdata))
    return(data.frame(Yes = pred, No = 1 - pred))
}
```

```
# Test our predict_model() function
predict_model(x = model_keras, newdata = x_test_tbl, type = 'raw') %>%
    tibble::as_tibble()
## # A tibble: 1,406 x 2
##            Yes        No
##          <dbl>     <dbl>
##  1 0.328355074 0.6716449
##  2 0.633630514 0.3663695
##  3 0.004589651 0.9954103
##  4 0.007402068 0.9925979
```

```
##   5 0.049968336 0.9500317
##   6 0.116824441 0.8831756
##   7 0.775479317 0.2245207
##   8 0.492996633 0.5070034
##   9 0.011550998 0.9884490
## 10 0.004276015 0.9957240
## # ... with 1,396 more rows
```

```
# Run lime() on training set
explainer <- lime::lime(
    x               = x_train_tbl,
    model           = model_keras,
    bin_continuous = FALSE)
```

```
# Run explain() on explainer
explanation <- lime::explain(
    x_test_tbl[1:10,],
    explainer     = explainer,
    n_labels      = 1,
    n_features    = 4,
    kernel_width = 0.5)
```

4.16 Checking Explanations with Correlation Analysis

```
# Feature correlations to Churn
corrr_analysis <- x_train_tbl %>%
    mutate(Churn = y_train_vec) %>%
    correlate() %>%
    focus(Churn) %>%
    rename(feature = rowname) %>%
    arrange(abs(Churn)) %>%
    mutate(feature = as_factor(feature))
corrr_analysis
```

```
## # A tibble: 35 x 2
##                          feature        Churn
##                           <fctr>        <dbl>
##  1                   gender_Male -0.006690899
##  2                   tenure_bin3 -0.009557165
##  3 MultipleLines_No.phone.service -0.016950072
##  4              PhoneService_Yes  0.016950072
##  5             MultipleLines_Yes  0.032103354
##  6               StreamingTV_Yes  0.066192594
##  7           StreamingMovies_Yes  0.067643871
##  8          DeviceProtection_Yes -0.073301197
##  9                   tenure_bin4 -0.073371838
## 10     PaymentMethod_Mailed.check -0.080451164
## # ... with 25 more rows
```

```
# Correlation visualization
corrr_analysis %>%
    ggplot(aes(x = Churn, y = fct_reorder(feature, desc(Churn)))) +
    geom_point() +
    # Positive Correlations - Contribute to churn
    geom_segment(aes(xend = 0, yend = feature),
                 color = palette_light()[[2]],
                 data = corrr_analysis %>% filter(Churn > 0)) +
    geom_point(color = palette_light()[[2]],
               data = corrr_analysis %>% filter(Churn > 0)) +
    # Negative Correlations - Prevent churn
    geom_segment(aes(xend = 0, yend = feature),
                 color = palette_light()[[1]],
                 data = corrr_analysis %>% filter(Churn < 0)) +
    geom_point(color = palette_light()[[1]],
               data = corrr_analysis %>% filter(Churn < 0)) +
    # Vertical lines
    geom_vline(xintercept = 0, color = palette_light()[[5]], size = 1, linetype =
2) +
    geom_vline(xintercept = -0.25, color = palette_light()[[5]], size = 1, linetyp
e = 2) +
```

```
    geom_vline(xintercept = 0.25, color = palette_light()[[5]], size = 1, linetype
= 2) +

    # Aesthetics

    theme_tq() +

    labs(title = "Churn Correlation Analysis",

         subtitle = "Positive Correlations (contribute to churn), Negative Correla
tions (prevent churn)",

         y = "Feature Importance")
```

4.17 Feature Investigation

Investigating features that are most frequent in the LIME feature importance visualization along with those that the correlation analysis shows an above normal magnitude.

```
# Tenure
churn_data_raw %>%
ggplot(aes(x = Churn, y = tenure)) +
geom_jitter(alpha = 0.25, color = palette_light()[[6]]) +
geom_violin(alpha = 0.6, fill = palette_light()[[1]]) +
theme_tq() +
labs(
title = "Tenure",
subtitle = "Customers with lower tenure are more likely to leave"
)

# Contract
churn_data_raw %>%
mutate(Churn = ifelse(Churn == "Yes", 1, 0)) %>%
ggplot(aes(x = as.factor(Contract), y = Churn)) +
geom_jitter(alpha = 0.25, color = palette_light()[[6]]) +
geom_violin(alpha = 0.6, fill = palette_light()[[1]]) +
theme_tq() +
labs(
title = "Contract Type",
subtitle = "Two and one year contracts much less likely to leave",
x = "Contract"
)

# Internet Service
churn_data_raw %>%
mutate(Churn = ifelse(Churn == "Yes", 1, 0)) %>%
ggplot(aes(x = as.factor(InternetService), y = Churn)) +
geom_jitter(alpha = 0.25, color = palette_light()[[6]]) +
geom_violin(alpha = 0.6, fill = palette_light()[[1]]) +
theme_tq() +
labs(
title = "Internet Service",
subtitle = "Fiber optic more likely to leave",
x = "Internet Service"
)
```

```
# Payment Method
churn_data_raw %>%
mutate(Churn = ifelse(Churn == "Yes", 1, 0)) %>%
ggplot(aes(x = as.factor(PaymentMethod), y = Churn)) +
geom_jitter(alpha = 0.25, color = palette_light()[[6]]) +
geom_violin(alpha = 0.6, fill = palette_light()[[1]]) +
theme_tq() +
labs(
title = "Payment Method",
subtitle = "Electronic check more likely to leave",
x = "Payment Method"
)

# Senior Citizen
churn_data_raw %>%
mutate(Churn = ifelse(Churn == "Yes", 1, 0)) %>%
ggplot(aes(x = as.factor(SeniorCitizen), y = Churn)) +
geom_jitter(alpha = 0.25, color = palette_light()[[6]]) +
geom_violin(alpha = 0.6, fill = palette_light()[[1]]) +
theme_tq() +
labs(
title = "Senior Citizen",
subtitle = "Non-senior citizens less likely to leave",
x = "Senior Citizen (Yes = 1)"
)

# Online Security
churn_data_raw %>%
mutate(Churn = ifelse(Churn == "Yes", 1, 0)) %>%
ggplot(aes(x = OnlineSecurity, y = Churn)) +
geom_jitter(alpha = 0.25, color = palette_light()[[6]]) +
geom_violin(alpha = 0.6, fill = palette_light()[[1]]) +
theme_tq() +
labs(
title = "Online Security",
subtitle = "Customers without online security are more likely to leave"
)
```

5. FINDINGS OF CHURN MODEL

5.1 Deep Learning Training Result

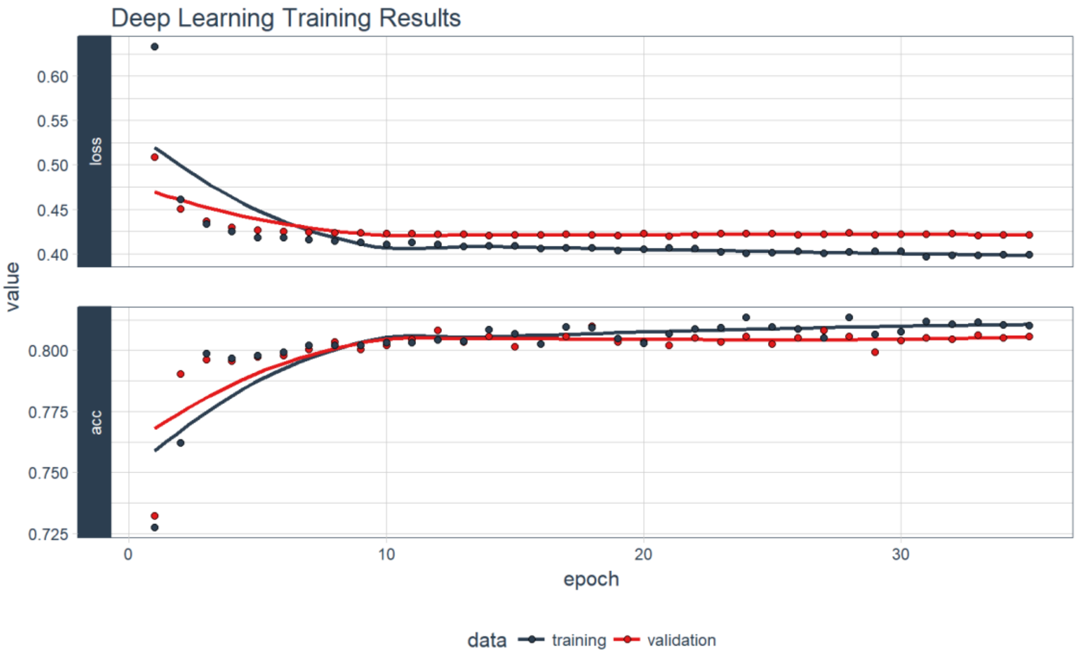


Figure 7 Deep Learning Training Result

The ANN Modelis able to achieve 82% accuracy on the telecom dataset

5.2 LIME Feature Importance Visualization

LIME is feature importance plot. This allows us to visualize each of the first ten cases (observations) from the test data. The top four features for each case are shown. The green bars mean that the feature supports the model conclusion, and the red bars contradict. A few important features based on frequency in first ten cases:

- Tenure (7 cases)
- Senior Citizen (5 cases)
- Online Security (4 cases

Figure 8 LIME Feature Visualization

5.3 LIME Feature Importance Visualization

Feature Importance Visualization demonstrates facetted heatmap of all case/label/feature combinations.

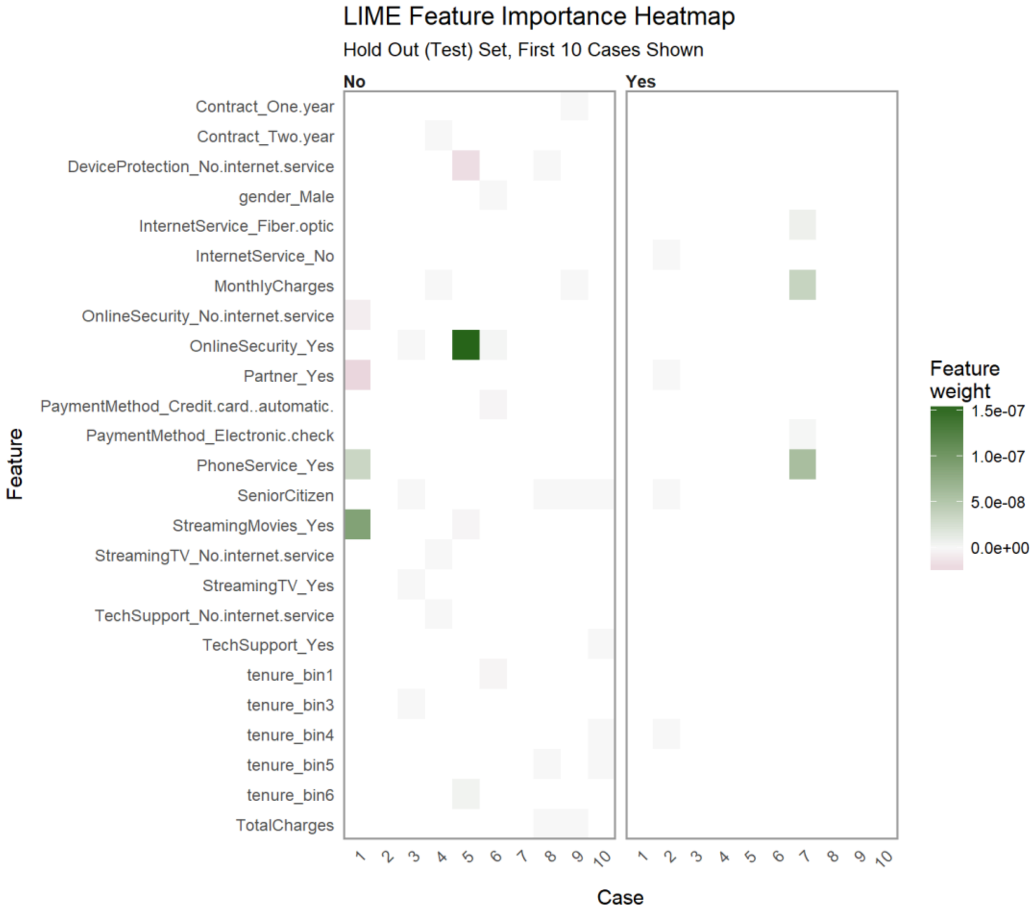


Figure 9 Feature Importance Visualization

5.4 Churn Correlation Analysis

The correlation analysis helps in quickly disseminating which features that the LIME analysis may be excluding. It can be concluded that the following features are highly correlated (magnitude > 0.25):'

Increases Likelihood of Churn (Red):

- Tenure = Bin 1 (<12 Months)
- Internet Service = "Fiber Optic"
- Payment Method = "Electronic Check"

Decreases Likelihood of Churn (Blue):

- Contract = "Two Year"
- Total Charges (Note that this may be a biproduct of additional services such as Online Security)

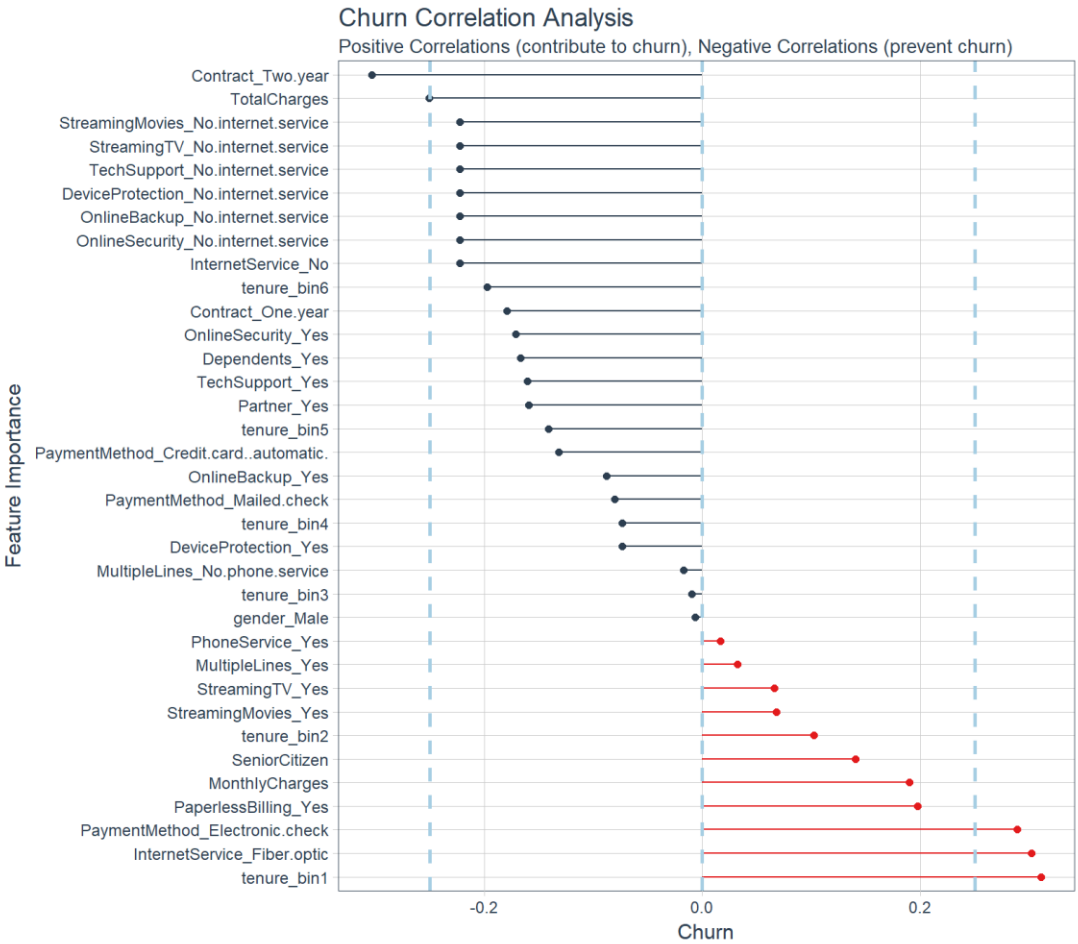


Figure 10 Churn Correlation Analysis

5.5 Feature Investigation Results

- Tenure (7/10 LIME Cases, Highly Correlated)
- Contract (Highly Correlated)
- Internet Service (Highly Correlated)
- Payment Method (Highly Correlated)
- Senior Citizen (5/10 LIME Cases)
- Online Security (4/10 LIME Cases)

Tenure (7/10 LIME Cases, Highly Correlated)

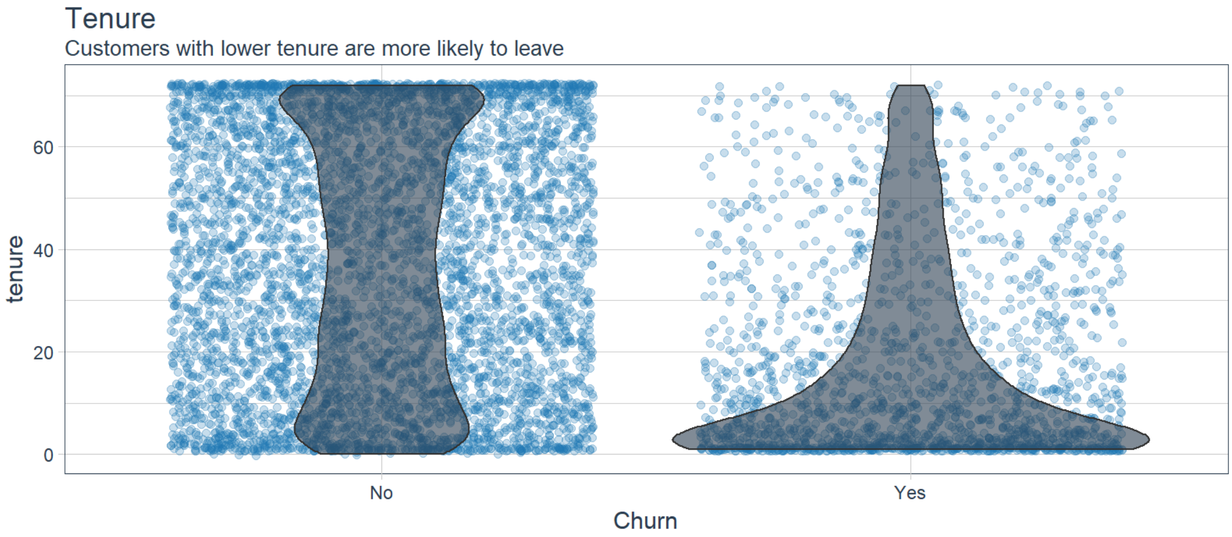
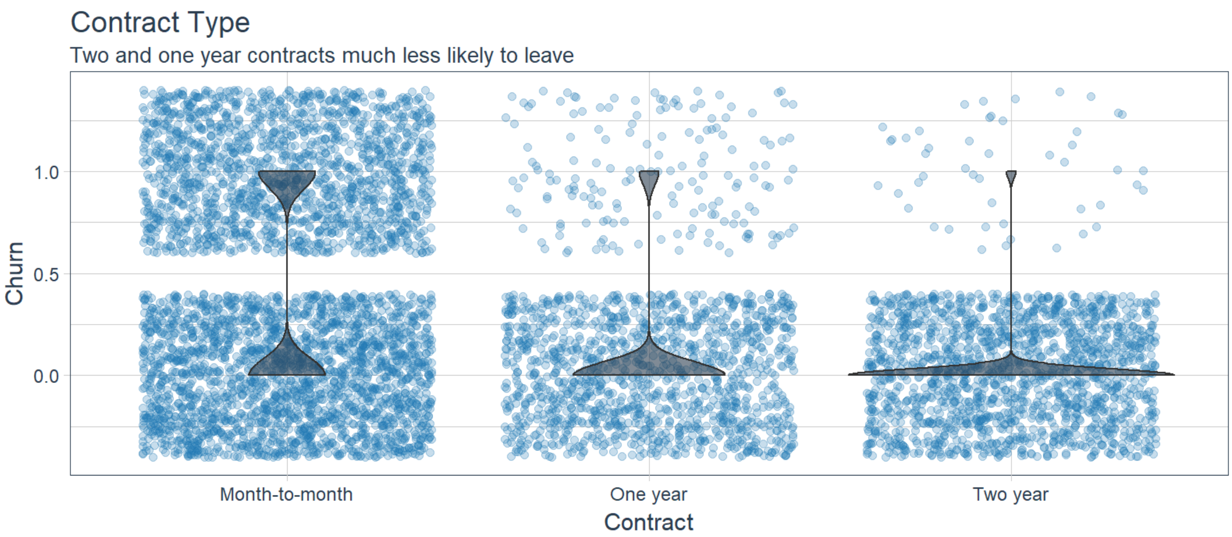
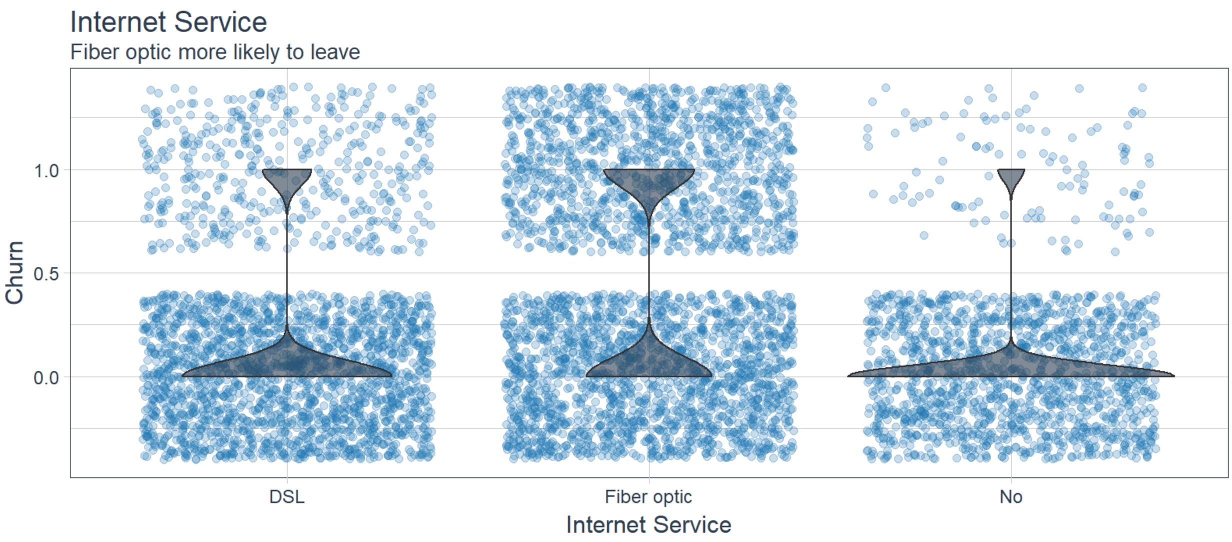


Figure 11 Feature Investigation Charts

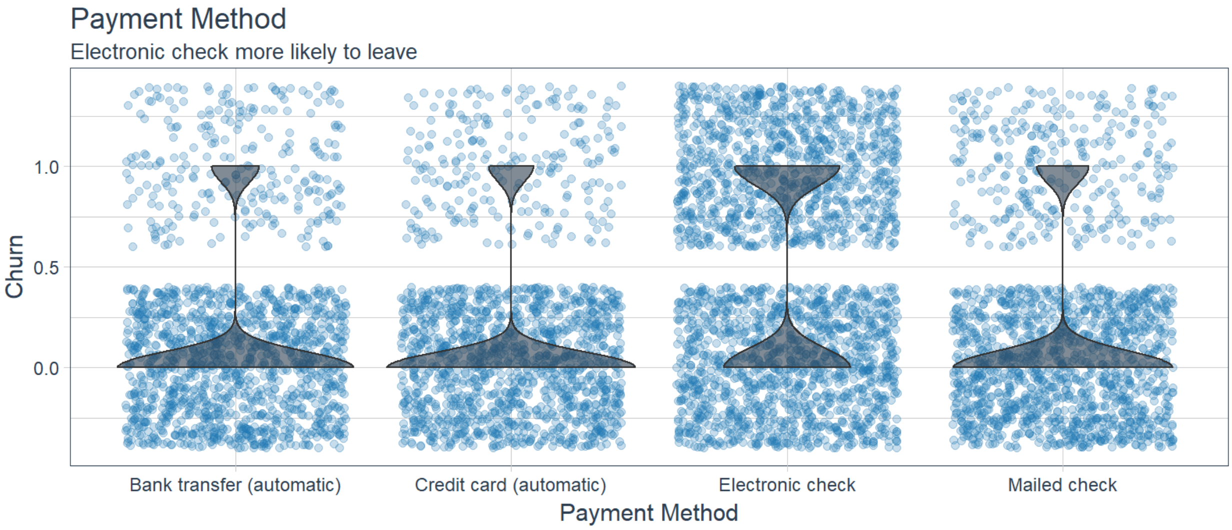
Contract (Highly Correlated)



Internet Service (Highly Correlated)

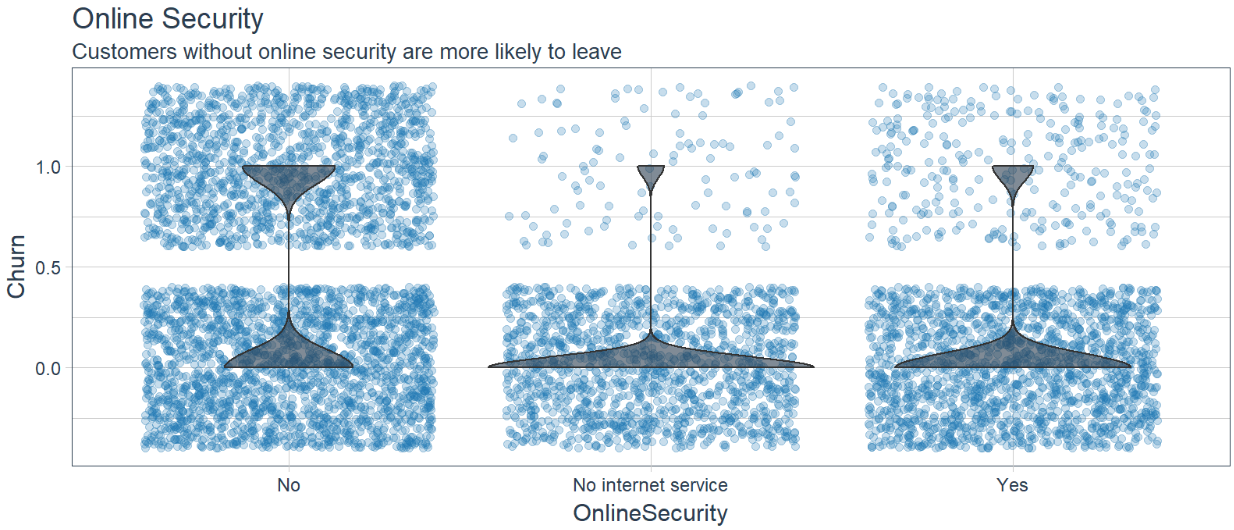


Payment Method (Highly Correlated)



Senior Citizen (5/10 LIME Cases)

Online Security (4/10 LIME Cases)



5.6 Performance of ANN Model

Confusion Table

```
##           Truth
## Prediction  no yes
##        no  950 161
##        yes  99 196
```

Accuracy

```
## # A tibble: 1 x 1
##    accuracy
##       <dbl>
## 1 0.8150782
```

Precision And Recall

```
## # A tibble: 1 x 2
##   precision    recall
##       <dbl>     <dbl>
## 1 0.6644068 0.5490196
```

F1 Score

```
## [1] 0.601227
```

REFERENCES

- https://www.pointillist.com/blog/reduce-churn-customer-journey-analytics/ 37 **(How to Reduce Churn Using Customer Journey Analytics**)
- http://chaotic-flow.com/saas-metrics-faqs-what-is-churn/ **(SaaS Metrics FAQs | What is Churn?)**
- https://www.zuora.com/guides/measure-subscriber-churn/ **(How to Measure Subscriber Churn – Zuora)**
- https://www.business2community.com/strategy/how-to-beat-the-average-churn-rate-for-subscription-service-enterprises-02158484 5 **(How to Beat the Average Churn Rate for Subscription Service Enterprises)**
- https://www.altexsoft.com/blog/business/customer-churn-prediction-for-subscription-businesses-using-machine-learning-main-approaches-and-models/ **(Customer Churn Prediction for Subscription Businesses using Machine Learning: Main Approaches and Models)**
- https://www.groundai.com/project/local-interpretable-model-agnostic-explanations-of-bayesian-predictive-models-via-kullback-leibler-projections/ **(Local Interpretable Model-agnostic Explanations of Bayesian Predictive Models via Kullback-Leibler Projections Local Interpretable Model-agnostic)**
- https://towardsdatascience.com/building-neural-network-from-scratch-9c88535bf8e9?gi=d5fdf80d37af **(Building Neural Network from scratch – Towards Data Science)**
- https://www.datacamp.com/community/tutorials/deep-learning-python 10 **(Keras Tutorial: Deep Learning in Python)**
- https://www.digitalocean.com/community/tutorials/how-to-build-a-deep-learning-model-to-predict-employee-retention-using-keras-and-tensorflow 15 **(How To Build a Deep Learning Model to Predict Employee Retention Using Keras and TensorFlow)**

ANNEXURE

List of Abbreviations

CNN- Convolutional Neural Network

ANN- Artificial Neural Network

RNN- Recurrent Neural Networks

Corr- Correlation

LIME- Local Interpretable Model-Agnostic Explanations

ML- Machine Learning

MLP - Multi-Layer Perceptron

NLP- Natural Language Processing