Customer Churn Prediction using "Telecom Churn Dataset"

by Michael Alabi

Abstract

Customer churn is a situation known as customer retention, customer turnover or customer defection i.e. the loss of clients or customer to other same competitors. Customer churn is major problem in large companies such as Telecommunication industry. Customer churn has direct effect on the revenues of the companies. As a result of its important concerns for Telecom company, they are seeking to develop ways to predict potential customer that could churn. The dataset for this project was provided by a Telecom company and it is available for download through www. kaggle.com. In this project, a data mining approach to predict customer churn which is a binary classification task which differentiates churners from non-churners. The aim of this project is to develop a predictive model to determine which customer have the potential to churn or stay with the company. This project utilizes both supervised learning (i.e. Logistic Regression, Decision Tree, Random Forest) and unsupervised learning algorithm (i.e. K-Means Clustering) to make churn prediction. In order to achieve the best model for churn, different variables and parameters were changed in order to produce new model, for instance, in logistic regression, three model were developed and compared/evaluate their performance. The models were evaluated with different model evaluation metric such as confusion matrix, precision, recall and area under curve receiver operating characteristics (AUC ROC) curve. Finally, the models were compared, and suitable model was chosen based on the metrics and customer retention plan was proposed to the Telecom company.

Keywords: Telecom Churn, Machine Learning, Supervised and Unsupervised, Model Evaluation.

1. Introduction

Customer churn is defined slightly differently by each organization or product. The customers that stop using a product or particular services for a given period of time are regarded as "churners". Due to this fact, customer churn is one of the most important elements in the Key Performance Indicators (KPI) of a products or services (Singh, 2019). In order for companies to create ways to retain their customer, a full customer lifecycle analysis would be necessary in most cases. The advent of machine learning techniques has provided an appropriate means of identifying customer churn based on the historical dataset and provides a retention plan for the company that we enable then to identify possible customer churn before it happens. Customer churn can be formulated as a binary classification problem and in this project both supervised machine learning (logistic regression, decision tree, random forest) and unsupervised machine leaning algorithms (K-means clustering) were considered in order to make prediction (Singh, 2019). The dataset is presented by a Telecom Company and it represents a collection of telecom company customers and whether or not they have left the service of the company or they are still with the company (Kumar and Chandrakala, 2016).

2. Background and Problem Statement

As earlier stated, the dataset for this project is provided by a Telecom company and the company is struggling with customer churn on a daily basis and in the past few months, the issues of customer churn became an important area of concern. As a result of this, the company stakeholders had a meeting in this regard and since most of the stakeholders believe in the power of Big Data Analytic to assist the company predict customer churn. The Telecom company decided to employ the service a Data Analytic company and this how our company came into the scene to assist the company makes some certain prediction based on the historical dataset provided and proposed relevant recommendations to reduce customer churn.

In order to effectively service our new client "Telecom Company" better, our team of data analytics expert have one-on-one discussion with the Telecom company stakeholder, to ask questions and to

understand their concerns and to position ourselves to deliver accurate service in predictive modeling. Therefore, the problem is that the Telecom company is having customer churn related issues and looking for ways to predict behaviour to retain customers for the products and services. This will be achieved by analyzing all the relevant customer data and developed focused customer retention program/plan.

3. Overview of the Data

Each row in the dataset represents a customer, each column contains customer's attributes as described in the column metadata data. The dataset contains 7043 rows (observations) and 21 columns (features):

- **Demographic:** Gender – Male/Female
  Age Range - In terms of Partner, Dependent and Senior Citizen.
- **Services:** Phone Services, Multiple lines, Internet Services, Online Security, Online Backup, Device Protection, Tech Support, Streaming TV and Streaming Movies.
- **Customer Account Information:** Tenure, Contract, Payment Method and Paperless Billing
- **Usages:** Monthly Charges and Total Charges
- **Target Variable:** Churn column which contains Yes/No i.e. whether the customer has left or not

4. Analytical Objectives/Solutions

This section contains the business objectives, analytical objective (i.e. predictive) and the hypothesis for this project as presented below:

4.1 Business Objective

The following are the stated business objective for this project

- ❑ To build a model which will predict whether a particular customer will churn or not.
- ❑ To Identify and Predict Possible Sources of Customer Churn
- ❑ To Provide Existing Customer Retention Plan Based on the Model Built

4.2 Analytical Objective: Predictive

In order to effectively provides suitable analytical solutions, the following questions have to be Considered:

- ❑ Can we predict why are the reasons for customer leaving (churn) the Telecom company Services?
  - High charges
  - Better offer from competitor
  - Poor customer services
  - Poor network coverage and etc.
- ❑ Can we predict if monthly charges are one of the reasons why customer churn?
- ❑ Can we predict the likelihood of existing and future customers being churn?

4.3 Stated Hypothesis

The Data Analyst Company will use Null and Alternative Hypothesis for the Telecom Churn Prediction.

Null Hypothesis:

- ❑ The exploratory data analysis and prediction from the model built ***will assist*** Telecom company to reduce churn and retain customers

Alternative Hypothesis:

- The exploratory data analysis and prediction from the model built ***will not assist*** Telecom company to reduce churn and retain customers.

5. Data Preparation

This section contains a comprehensive exploratory of the dataset, only single data was acquired, and it does not involve integration of any other dataset. In this section, data preprocessing of various variables and columns in the dataset will be presented to give the readers the fundamental understanding of the "telecom churn dataset" used for the predictive model.

5.1 Discover & Asses the Data

- The dataset contains only three numerical columns (Tenure, Total Charges and Monthly charges)
- Columns with two values such as Yes/No were converted into binary values (1's and 0's)
- There are no Outliers in the Dataset
- Customer ID column was dropped because it does not provide specific information about the customers
- Data with more than two categorical values were replaced with dummy values
- No duplicate in the Dataset
- The total charges columns contain 11 missing values and tenure of 0 months, it is assumed that these customers were brand new and had not yet bill as at the time of this data collection or may be the client have not stayed more than one month. These customers information was kept as part of the analysis and replaced with zeros.
- There are 18 Categorical Variables

5.2 Data Understanding

The CustomerID column contain no specific information about any customer, hence the column was dropped from the dataset. Based on the target variable "Churn" from the dataset, the proportion of the customer opting out of the "Telecom company" services were shows using simple pie chart that shows the distribution of the two classes in the dataset. It shows that 26.5% of the customer churn while 73.5% of the customer stay with the company as seen in Figure 1. The preprocessing of the 'Gender' column have categorical values and have to be converted to numerical value by using "get_dummies()" function. Based on the proportion of male and female churn, the variable is not informative as seen in Figure 2, and it is very difficult to decide if a customer is going to stop using the service or not and as a result of this, more preprocessing have to be carried out. Some variables required the use of "one hot encoding" because the columns contain more than two categorical values, and these columns are (Gender, PaymentMethod and Contract).

The columns that have been assigned dummies variables were dropped and such columns are (InternetService, Multiplelines, Contract, PaymentMethod, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies).

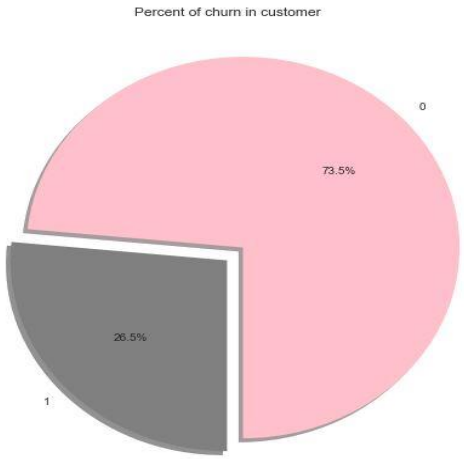

Figure 1. Simple pie chart that shows distribution of churn

Some of variables were also identified with two categorical values ‘Yes/No’ and the values were converted into 1’s and 0’s and the columns are (Dependents, PhoneService, Partner and PaperlessBilling). One hot encoding was not applied to these variables because they didn’t have more than two categorical values.

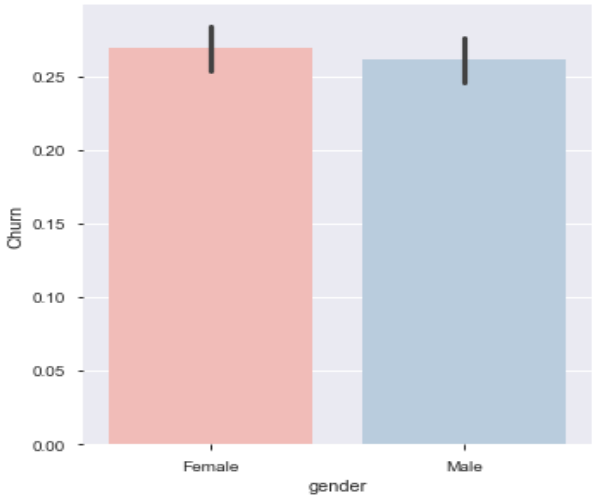

Figure 2. Simple bar chart that shows the distribution of gender

The dataset contains - 1 float, 2 integer and 18 string i.e. (float64(1), int64(2), object(18)). The missing values was also checked as shown below.

```
customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
```

```
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
dtype: int64
```

The two columns for MonthlyCharges and TotalCharges primarily have numerical values and no conversion were made in these values. However, the TotalCharges column has missing values/null values and the 11 missing value were identify from the original data and were replaced appropriately.

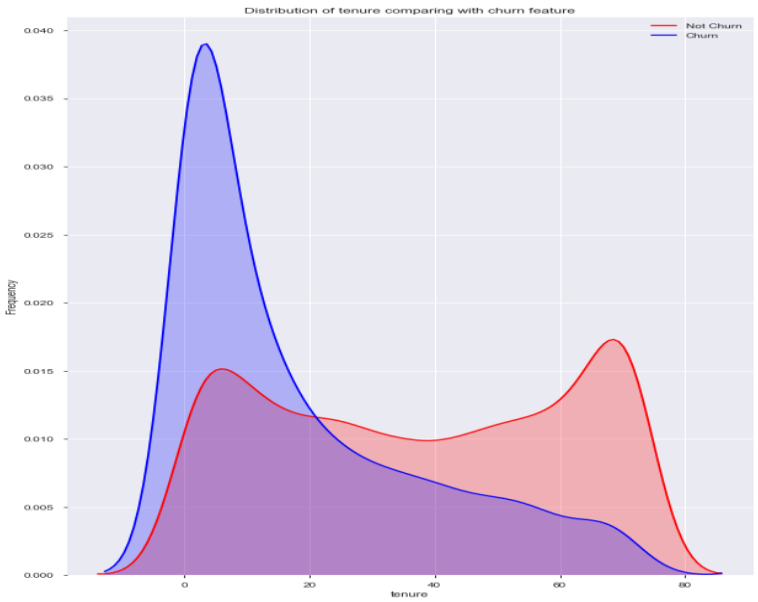

Figure 3. The Distribution of Tenure Vs Churn Features

In order to effectively analyze how long or number of months a customer has stayed with Telecom company. The “tenure” column was analyzed against “Churn” using distribution chart and seaborn library in Python was used for the plot. The data visualization in Figure 3 shows that in ‘Tenure’ column, most customer leave the Telecom company services before 20 months. This indicates that there is possibility that if customer stayed using Telecom company services for more than 20 months, it is highly unlikely that the customer will churn.

5.3 Information About the Dataset

The info() function in the python library was used to get information about the dataset as presented below:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 7043 entries, 0 to 7042 Data columns (total 23 columns):

#	Column	Non-Null Count	Dtype
0	SeniorCitizen	7043 non-null	int64
1	Partner	7043 non-null	int64
2	Dependents	7043 non-null	int64
3	tenure	7043 non-null	int64
4	PhoneService	7043 non-null	int64
5	MultipleLines	7043 non-null	int64
6	PaperlessBilling	7043 non-null	int64
7	MonthlyCharges	7043 non-null	float64
8	TotalCharges	7043 non-null	float64
9	Churn	7043 non-null	int64
10	gender_Female	7043 non-null	uint8
11	gender_Male	7043 non-null	uint8
12	Has_InternetService	7043 non-null	int64
13	Fiber_optic	7043 non-null	int64
14	DSL	7043 non-null	int64
15	PaymentMethod_Bank transfer (automatic)	7043 non-null	uint8
16	PaymentMethod_Credit card (automatic)	7043 non-null	uint8
17	PaymentMethod_Electronic check	7043 non-null	uint8
18	PaymentMethod_Mailed check	7043 non-null	uint8
19	Contract_Month-to-month	7043 non-null	uint8
20	Contract_One year	7043 non-null	uint8
21	Contract_Two year	7043 non-null	uint8
22	Cluster	7043 non-null	int32

dtypes: float64(2), int32(1), int64(11), uint8(9) memory usage: 804.8 KB

6. Data Preparation - Exploratory Data Analysis

To perform the data analysis, certain python libraries were used. The code below was used to load and initialize the various libraries, then the data was loaded.

```
import numpy as np                    # linear algebra
import pandas as pd                   # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt       # this is used for the plot the graph
import seaborn as sns                 # used for plot interactive graph.
import os
from pandas import Series, DataFrame
import scipy
from scipy.stats import spearmanr
import sklearn
from sklearn.preprocessing import scale             # importing scale function
from sklearn.model_selection import train_test_split  #  import split train and test
from sklearn import metrics                         # importing metrics to evaluate the model
from sklearn import preprocessing                   # importing preprocessing tools
from sklearn.preprocessing import StandardScaler    # Standard Scaling for fit_transform for large value
import warnings
warnings.filterwarnings("ignore")
```

As the beginning of the data preparation, one important python library was used to perform an exploratory data analysis of the entire dataset in one line of code called “panda_profiling.ProfileReport” as shown below:

pandas_profiling.ProfileReport(pd.read_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv'))

This present an overview data analysis to display each variable distribution, the correlation matrix, missing value in each column and sample as briefly display in Figure 4. The complete pandas-profiling of the dataset will be seen in the attached python code for the project.

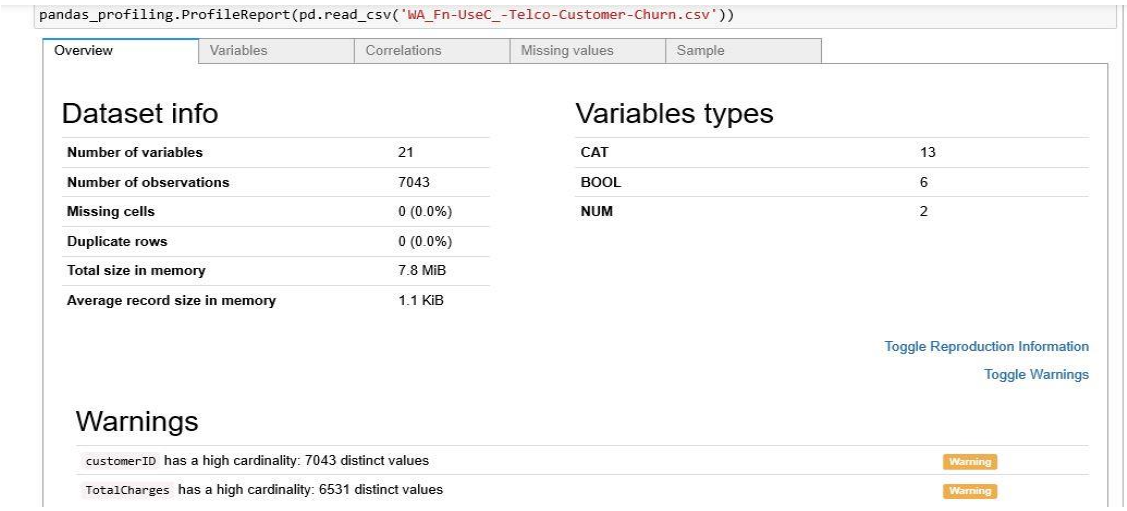

Figure 4. Overview of the dataset information and variable types

In order to understand the column with numerical values using “describe ()” function in python as shown in the Figure 5. This function provides the basic statistics about the column with numerical values. The author is able to see if there is any imbalance or variation in term of the magnitude of the dataset and the need to scale the numerical columns using “Standardscaler()” function.

```
In [6]: # Getting the basic statistics of the numerical columns
        data.describe()
```

Out[6]:

	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

Figure 5. Basic statistics of the numerical values column

6.1 Correlation Matrix Among the Variables/Features

The correlation matrix shows that the data follows a roughly straight-line trend. Therefore, the variables have an approximately linear relationship. There is no correlation between two variables (where the correlation is 0 or near 0). The darkest blue means there is a perfect positive correlation while light yellow

means there is a perfect negative correlation. From correlation matrix, features like Tenure, Monthly charges and Total charges are highly correlated with services like Multiple Phone Lines services and Internet services like Online Security, Online Backup, Device Protection, Tech Support, Streaming TV and Streaming Movies services.

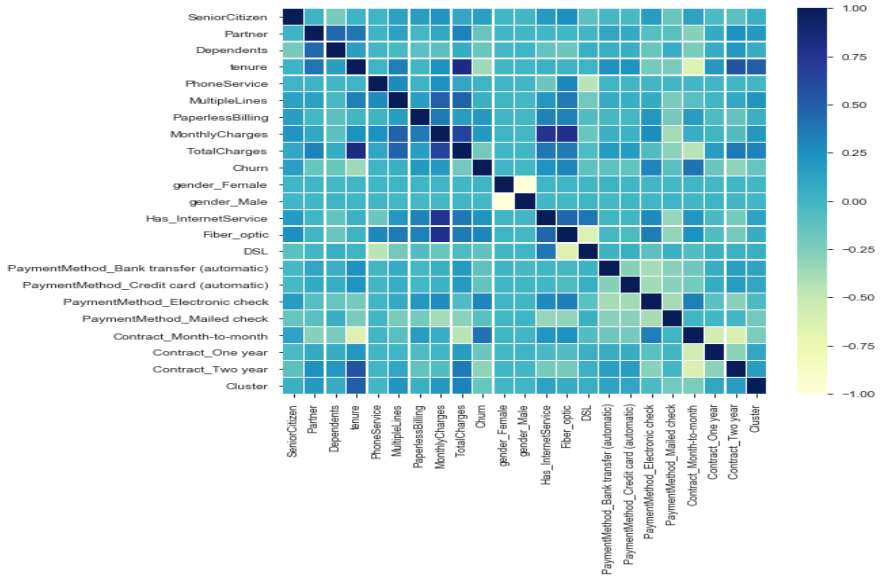

Figure 6. Correlation matrix among the variables in the dataset

6.2 Preprocessing of Each Column in the Dataset

In the section, preprocessing of each column in the dataset will be presented and this includes the process involved in transform and changing the data, assumption made and where the values where given dummy variables or conversion of Yes or No to 1’s and 0’s.

Target Variables – “Churn”: The churn column consists of “Yes or No” and since machine learning does not understand categorical variables. Therefore, the target class/variable was converted into 1’s and 0’s.

Gender: The 'Gender' column have categorical values and we have to convert the categorical value to numeric value using the pandas tools "get_dummies() function" and this allows one hot encoding to be done on the values present in the column.

Partner: The column has Yes or No Values, since we have only two categorical values in the column, the categorical variable Yes or No will be replaced by 1's and 0's. Since the categorical variables is not more than two, no need to use 'one hot encoding for the values.

Dependents, Phone Service and Paperless Billing: Since the 'Dependents', 'Phone Service' and 'Paperless Billing' columns contain only two categorical values. The categorical value can be converted to numeric values of 1's and 0's. The 'one hot encoding' was not applied since the categorical values does not exceed two.

Tenure: This allows the author to know how long or number of months a person has used with the Telecom company as previous shown in Figure 3. The column contains numerical values which implies that there is no conversion to be made in this situation.

Multiple Lines: This column consists of three categorical values and this was converted into numerical values and replaced with dummy variables using “one hot encoding” function in python.

Internet Service: This column consists of three categorical values and the categorical variables will be converted into dummy values using "one hot encoding" function in python.

OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovie: These columns contain two categorical values "Yes and No" and it was converted into 1's and 0's

Payment Method and Contract: These columns contain more than two categorical values and was converted to dummy values using get_dummies () function. This allows one hot encoding to be done on the values present in the column.

Monthly Charges and Total Charges: These columns contain numerical values in form of "float" i.e. the numerical values contain decimal number. There are 11 missing values in total charges columns which have been properly handled and replaced accordingly (see the attached python code for details). This section used box plot to analyse both the monthly and total charges to understand their effect on the churning customers. Based on the monthly and total charges, the box plot assist to identify between customers who churn the Telecom services and those that did not. More so, the box plot enables the author to recognize that there is an overlapping between the customers that churn and those that are not, i.e. it is not always that when monthly charges go beyond "80" that the customer churn. Despite the high monthly charges, some customers still stay with the Telecom company services and this is the same with the total charges in respect to churn as shown in Figure 7.

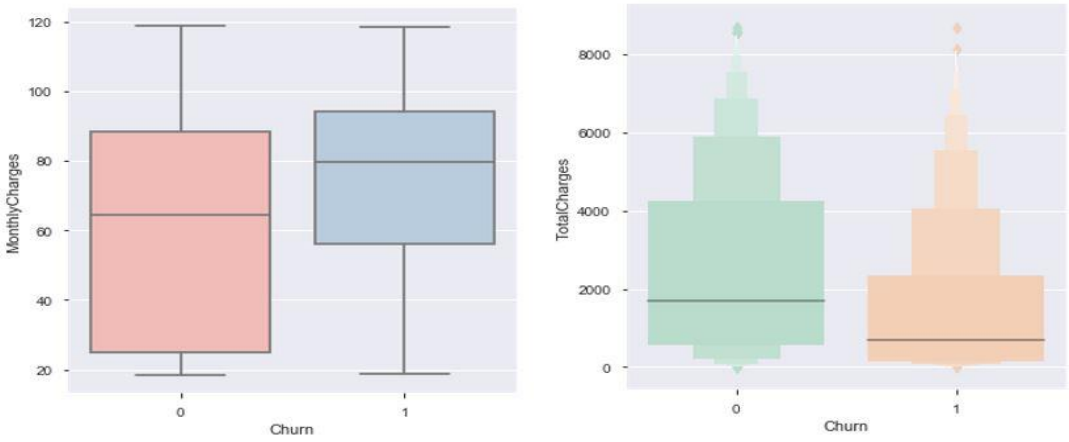

Figure 7. A box plot to show the relationship/effect of monthly and totally charges with respect to churn

After the preprocessing is done, and irrelevant columns have been dropped (see details in section 5.2), the rest of the columns for the model building were identified using the code below. The dataset presented below includes the data used for classification and the newly formed variables as a result of assigning dummies variables, one hot encoding and conversion of two variables "Yes and No" to 1's and 0's.

```
In [46]: data.columns

Out[46]: Index(['SeniorCitizen', 'Partner', 'Dependents', 'tenure', 'PhoneService',
                'MultipleLines', 'PaperlessBilling', 'MonthlyCharges', 'TotalCharges',
                'Churn', 'gender_Female', 'gender_Male', 'Has_InternetService',
                'Fiber_optic', 'DSL', 'PaymentMethod_Bank transfer (automatic)',
                'PaymentMethod_Credit card (automatic)',
                'PaymentMethod_Electronic check', 'PaymentMethod_Mailed check',
                'Contract_Month-to-month', 'Contract_One year', 'Contract_Two year'],
               dtype='object')
```

7. Model Development

This section introduces the splitting of the dataset into train and test and the model development, As earlier stated, three supervised learning techniques (Logistic Regression, Decision Tree and Random Forest0 and one unsupervised learning technique were used in the development of the model. Different parameters were used and changed to transform and form newly produced model. These models were compared and evaluated to see the suitable model with good performed that can be used to make accurate predictions using appropriate metrics such as AUC ROC, confusion matrix, precision, recall, precision-recall curve and accuracy.

7.1 Splitting the Telecom Churn Data into Train and Test Sets

The 'Telecom Churn' dataset has been split into train and test sets in the ratio of 75:25 (i.e. 75% training and 25% test data as shown in the python code below. The same ratio of data split is used across all the model (both supervised and unsupervised learning model) and the same data preparation applies to all the model development phase.

```
# Split dataset into training set and test set

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25, random_state=1) # 75% training and 25% test data
```

7.2 Model Evaluation Metrics

This section provides brief explanation of the stated evaluation metric to check the performance of the built models:

- Confusion Matric: A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made (Geeksforgeeks, 2020).
- Precision: Precision is about being precise, i.e., how precise your model is. In other words, you can say, when a model makes a prediction, how often it is correct. This implies that "what proportion of positive identifications was actually correct?". It describes how good a model is at predicting the positive class. Precision is referred to as the positive predictive value.
- Recall: This implies that "What proportion of actual positives was identified correctly?"
- AUC ROC Curve: Area Under ROC Curve (or ROC AUC for short) is a performance metric for binary classification problems. The AUC represents a model's ability to discriminate between positive and negative classes. An area of 1.0 represents a model that made all predictions perfectly. An area of 0.5 represents a model as good as random. A ROC Curve is a plot of the true positive rate and the false positive rate for a given set of probability predictions at different thresholds used to map the probabilities to class labels. The area under the curve is then the approximate integral under the ROC Curve (Brownlee, 2016).
- Accuracy: Classification accuracy is the number of correct predictions made as a ratio of all predictions made. This is the most common evaluation metric for classification problems, it is also the most misused. It is really only suitable when there are an equal number of observations in each class (which is rarely the case) and that all predictions and prediction errors are equally important, which is often not the case.
- Precision-Recall Curve: The precision and recall can be calculated for thresholds using the precision_recall_curve() function that takes the true output values and the probabilities for the

positive class as output and returns the precision, recall and threshold values.

7.3 Building the Machine Learning Model

This section presents the model development for the prediction of churn in the "Telecom Churn" dataset of our client. Both supervised and unsupervised models were developed as presented in the sub-sections below.

7.3.1 Supervised Machine Learning

This model relationships and dependencies between the target prediction output and the input features such as we can predict the output values for new data based on those relationships which it learned from the previous datasets (Fumo, 2017). The list of all the features to build the model is called "feature cols" which is recognized as X while the target variable is the churn column is labeled as Y and presents below.

```
In [47]:
feature_cols = ['SeniorCitizen', 'Partner', 'Dependents', 'tenure', 'PhoneService',
       'MultipleLines', 'PaperlessBilling', 'MonthlyCharges', 'TotalCharges', 'gender_Female', 'gender_Male',
        'Has_InternetService', 'Fiber_optic', 'DSL', 'PaymentMethod_Bank transfer (automatic)',
        'PaymentMethod_Credit card (automatic)','PaymentMethod_Electronic check', 'PaymentMethod_Mailed check',
        'Contract_Month-to-month', 'Contract_One year', 'Contract_Two year']

X = data[feature_cols]  # feature columns
Y = data.Churn
```

7.3.1.1 Logistic Regression Model

Logistic Regression is used when the dependent variable (target) contains categorical values. In this case, the dependent/target variable is "Churn" and contains categorical values. In this section, three new models were built with same data and same algorithm, but different parameters were used. The purpose of building three logistic regression models for this project is to modify how the model is built and to be able to choose the best with good accuracy and performance metrics. The logistic regression classifier i.e. [LogisticRegression()] from python was used to build the model as shown below.

```
from sklearn.linear_model import LogisticRegression

model_LogReg = LogisticRegression()
model_LogReg.fit(X_train, Y_train)

y_predict = model_LogReg.predict(X_test)
```

In order to build a perfect logistic regression model, the most important features toward predicting the target variables were identified using appropriate python code and the result outputs of important features are (Contract_Month-to-Month, Fiber_Optic, PaperlessBilling, Has_InternetService and SeniorCitizen) as shown in Figure 8. The first model was built using 75:25 ratio of training and test set with random state of 1 and max_iteration = 100 i.e. (X, Y, test_size=0.25, random_state=1). The second model was built using same test size and the random state was changed to 4, with aim to product new improve model. In the third model, the StandardScaler() classifier library in python was applied to fit and transform dataset both the train and test data (i.e. X_train and X_test). The details and processes for transforming the variables to produce new models will be found in the attached python code.

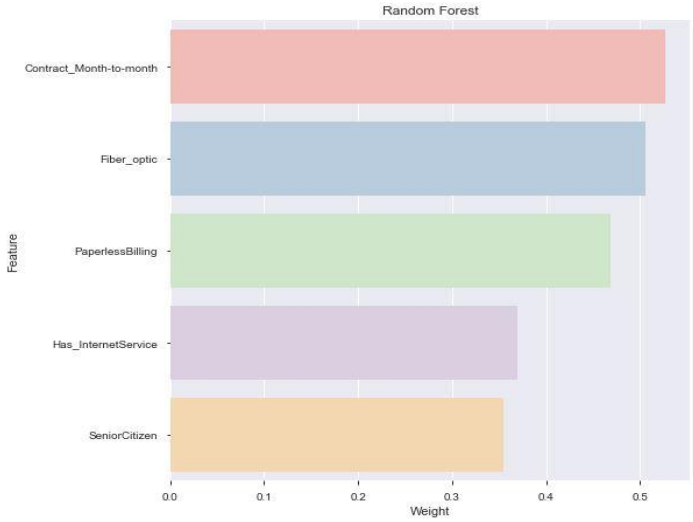

Figure 8. Important Features for Logistic Regression Model

Evaluating the Logistics Regression Models:

This section provides the evaluation of the models using some metrics such as the accuracy, precision, recall and the Area Under Curve Receiver Operating Curve (AUC ROC) as shown in the Figure 9. Firstly, the accuracy of the three models were considered. The first model has an Accuracy of 0.81, Precision is 0.75 and Recall is 0.73. The second model has an Accuracy of 0.81, Precision is 0.74 and Recall is 0.70. The third model has an Accuracy of 0.79, Precision is 0.73 and Recall is 0.70. Therefore, the overall accuracy of the three models is not the same and this implies that transforming the variables of the data and changing of parameter has impact on the model performance and there is a drastically difference in ranges of the inputs used.

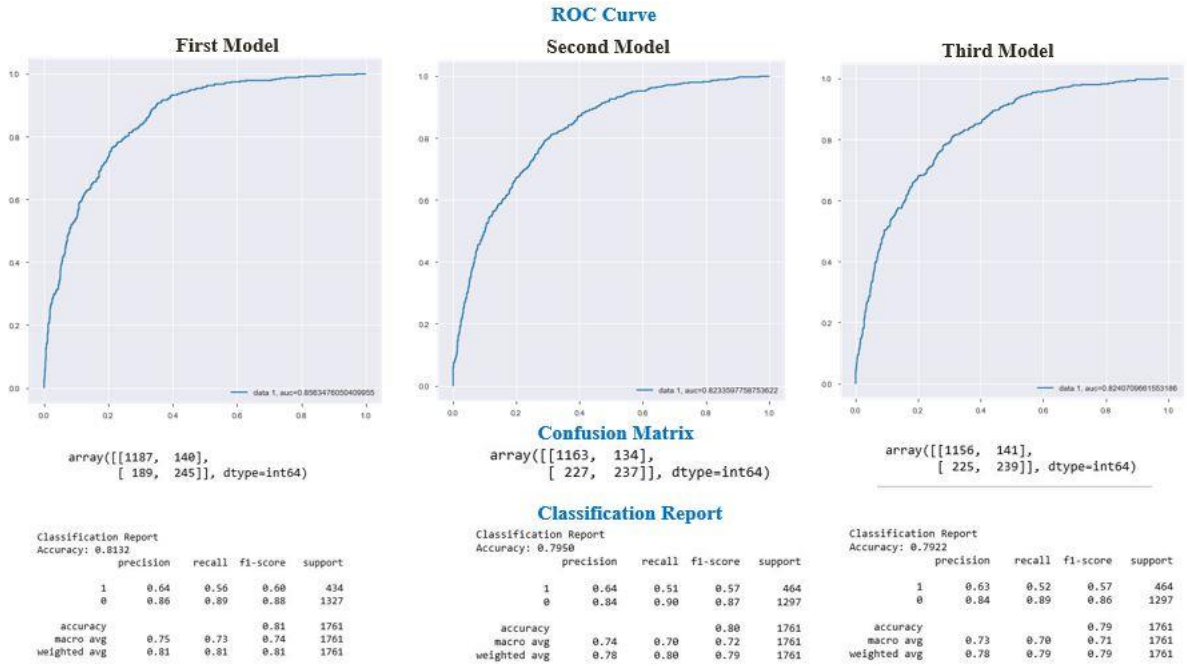

Figure 9. Evaluation Metrics for Logistic Regression Models

The confusion matrix is in the form of the array object. The dimension of this matrix is 2*2 because the models is binary classification and there are two classes 0 and 1. The diagonal values represent accurate predictions, while non-diagonal elements are inaccurate predictions. In the output, the first model has 1187 and 245 accurate prediction, and 189 and 140 inaccurate predictions. The second model has 1163 and 237 accurate prediction, and 227 and 134 inaccurate predictions. The third model has 1156 and 229 accurate prediction, and 225 and 141 inaccurate predictions. This shows that the first model has higher model performance than the second model because of its higher diagonal values which represent 1187 and 245 accurate predictions. The first, second and third models have AUC ROC curve of 0.8563, 0.8233 and 0.8241 respectively. The true positive rates were plotted against false positive rates. Since an excellent performance model is the one with AUC near to the 1. Comparing the three models AUC ROC curve, the first model has the highest value with 0.8563 which indicates that the model has good measure of separability and has 80% chance that it will distinguish between positive class and negative class.

In summary, based on the various evaluation metrics used to evaluate the Logistic Regression model, it is easy to deduce that the first model has higher accuracy, confusion matrix to predict accurately and higher AUC ROC curve. This is expected to make accurate telecom customer churn prediction.

7.3.1.2 Random Forest Model

A random forest classifier is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The random forest combines hundreds or thousands of decision trees, trains each one on a slightly different set of the observations, splitting nodes in each tree considering a limited number of the features. The final predictions of the random forest are made by averaging the predictions of each individual tree. For Random Forest Model, three new models were built with same data and same algorithm, but different parameters were used. The purpose of building three random forest models for this project is to modify how the model is built and to be able to choose the best with model with good accuracy and performance metrics. The Random Model classifier i.e. [RandomForestClassifier()] from python was used to build the model as shown below.

```
# importing Random Forest Classifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.datasets import make_classification

# Import RandomForest Classifier and indicates the number of tree

model_rf1 = RandomForestClassifier(n_estimators=200, random_state = 0)

y_train_array = np.ravel(Y_train)
```

The dataset was split into 75:25 ratio of training and test set. In order to have the appropriate model, the processes were repeated three by assignment different parameters (number of trees, n_estimators, random_state) to the output. Three different models were generated using same data and same algorithm. The first model used 200 as the number of trees with random_state = 0. The second model used 300 as the number of trees with random_state = 4. The third model used 500 as the number of trees and random_state = 6.

Based on the model evaluation used i.e. accuracy, precision, recall and the Area Under Curve Receiver Operating Curve (AUC ROC) as shown in the Figure 10. Firstly, the accuracy of the three models were considered in the evaluation. The first model has an Accuracy of 0.7808 Precision is 0.72 and Recall is 0.70. The second model has an Accuracy of 0.7797, Precision is 0.71 and Recall is 0.69. The third model has an Accuracy of 0.7853, Precision is 0.72 and Recall is 0.70. Therefore, the overall accuracy of the three models is not the same and this implies that transforming the variables of the data and changing of parameters has impact on the model performance and there is a drastically difference in ranges of the inputs used.

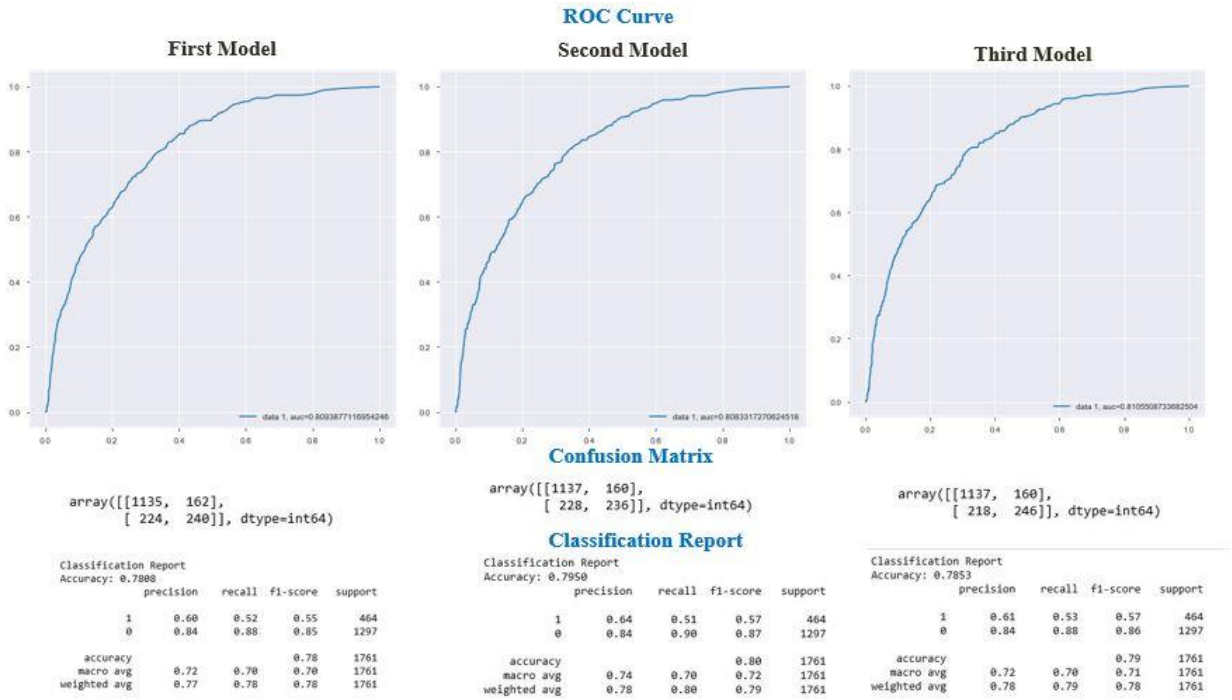

Figure 10. Evaluation Metrics for Random Forest Models

The dimension of the confusion matrix is 2*2 because the models is binary classification and in form of an array. The diagonal values represent accurate predictions, while non-diagonal elements are inaccurate predictions. In the output, the first model has 1135 and 240 accurate prediction, and 224 and 162 inaccurate predictions. The second model has 1137 and 236 accurate prediction, and 228 and 162 inaccurate predictions. The third model has 1137 and 246 accurate prediction, and 218 and 160 inaccurate predictions. This shows that the third model has higher model performance than the second model because of its higher diagonal values to accurately make predictions. The first, second and third models have AUC ROC curve of 0.8093, 0.8083 and 0.8105 respectively. The true positive rates were plotted against false positive rates. Since an excellent performance model is the one with AUC near to the 1. Comparing the three models AUC ROC curve, the three models has almost the same values which were approximately 0.81 and this shows that the models has good measure of separability and has 80% chance that they will be able to distinguish between positive class and negative class. Also, the same AUC ROC of the three models shows that transformation introduced to the variables of the dataset does not have significant impact on the model performance and no drastically difference of the inputs used.

In overall, based on the various evaluation metrics used to evaluate the Random Forest model, it is easy to deduce that the third model has higher accuracy, confusion matrix to predict accurately and good AUC ROC curve. This is expected to make accurate telecom customer churn prediction. The Random Forest

Model tree was printed and visualized in the python code but too large to be inserted in this project since it was designed to be visualized n_estimators by n_estimators.

7.3.1.3 Decision Tree Model

Decision Trees are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features (Scikit-learn, 2020). For the decision trees model, the dataset was split to 75% training and 25% test set. In this case, three models were built while experimenting with parameters. The decision classifier attribute used are "gini" and maximum tree depth were changed to produce new model and increase the number of trees depth i.e. (criterion="gini", max_depth=4). The code below was used to call the Decision Tree classifier as [DecisionTreeClassifier()], fit and predict the model.

```
# Import Decision Tree Classifier

from sklearn.tree import DecisionTreeClassifier

# Create Decision Tree classifer object

decision_tree = DecisionTreeClassifier(criterion="gini", max_depth=4)

# Train Decision Tree Classifer

decision_tree = decision_tree.fit(X_train,Y_train)

# Predict the response for test dataset

y_pred = decision_tree.predict(X_test)
```

In this case of decision tree, and to build a good decision tree model, the most important features toward predicting the target variables were identified using appropriate python code and the result outputs of important features are (Contract_Month-to-Month, Fiber_Optic, TotalCharges, Tenure and MonthlyCharges) as shown in Figure 11.

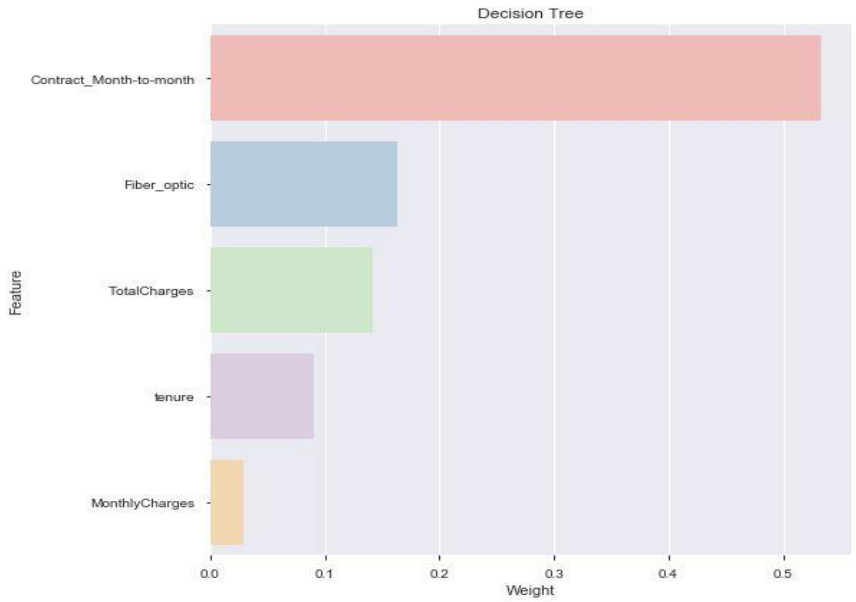

Figure 11. Important Features for Decision Trees Model

The maximum depth of trees for the first, second and third models are 4, 5 and 6 respectively. It was noticed that as the maximum depth of trees increases, the trees occupy more variables and provide more clearer picture of possible causes or areas where customer churn. Based on the model evaluation used i.e. accuracy, confusion matrix, Area Under Curve Receiver Operating Curve (AUC ROC) as shown in the Figure 12. Firstly, the accuracy of the three models were considered in the evaluation. The first model has an Accuracy of 0.7944. The second model has an Accuracy of 0.7950. The third model has an Accuracy of 0.7980. Therefore, the overall accuracy of the three models is not the same but very close, and this implies that transforming the variables and parameters of the depth of the trees have impact on the model performance and there is a drastically difference in ranges of the inputs used. The third model with maximum trees depth of 6 has the higher accuracy for the model.

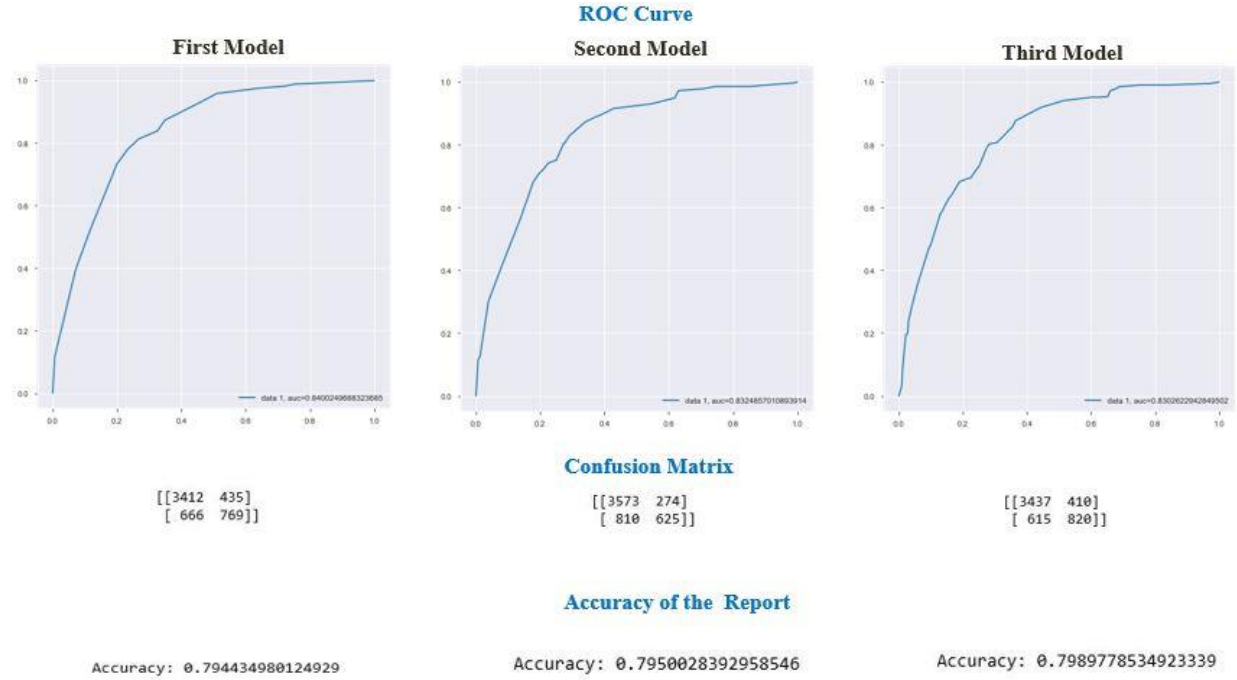

Figure 12. Evaluation Metrics for Decision Tree Models

The dimension of the confusion matrix is 2*2 because the models is binary classification and also in form of an array. In the output, the first model has 3412 and 769 accurate prediction, and 666 and 435 inaccurate predictions. The second model has 3573 and 625 accurate prediction, and 810 and 274 inaccurate predictions. The third model has 3437 and 820 accurate prediction, and 615 and 410 inaccurate predictions. This shows that the third model has higher model performance than the second model because of its higher diagonal values to accurately make predictions. The first, second and third models have AUC ROC curve of 0.8400, 08325 and 0.8303 respectively. The true positive rates were plotted against false positive rates. Since an excellent performance model is the one with AUC near to the 1. Comparing the three models AUC ROC curve, the second and third models have same AUC ROC of 0.83 while the first models 0.84. This shows that the models have good measure of separability and has 80% chance that they will be able to distinguish between positive class and negative class. Also, the same AUC ROC of the three models shows that transformation introduced to the variables of the dataset has little impact on the model performance and drastically difference in range of the inputs used.

In overall, based on the various evaluation metrics used to evaluate the Decision Trees model, it is easy to deduce that the third model has higher accuracy confusion matrix to predict accurately and good AUC ROC curve. This is expected to make accurate telecom customer churn prediction. For purpose of visualized, below is the decision tree with maximum depth of 4 as shown in Figure 13. The decision trees show that the possible customers churn based on three areas such as "Total Charges, Internet Service (specifically fiber optics), Payment Method and Monthly Charges". These are the possible area the Telecom company needs to look into and make adjustment where needed in order to retain their existing customers.

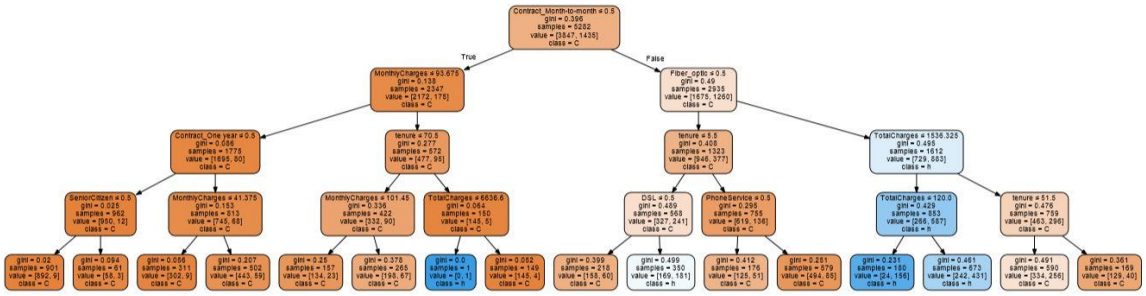

Figure 13. Decision Trees Visualization with Maximum Depth of Four

7.3.2 Unsupervised Machine Learning

There are no output categories or labels data based on which the algorithm can try to model relationships. These algorithms try to use techniques on the input data to mine for rules, detect patterns, and summarize and group the data points which help in deriving meaningful insights and describe the data better to the users (Fumo, 2017).

7.3.2.1 K-Means Clustering

A KMeans clustering model is an unsupervised learning algorithm that is refers to as a connection of data points aggregated together because of certain similarities. In this section, two important features will be considered for the KMeans clustering model and these two features are "Monthly Charges and Tenue". Based on the correlation matrix (see Figure 6), these two features have significant correlation. The simple box plot in Figure 7, shows the significance of these two variables/features to customer churn. In order to have an insight into the two features, a simple scatter plot was plotted to visualize the two significance of the variable to be used for the clustering of telecom customer churn as shown in Figure 14.

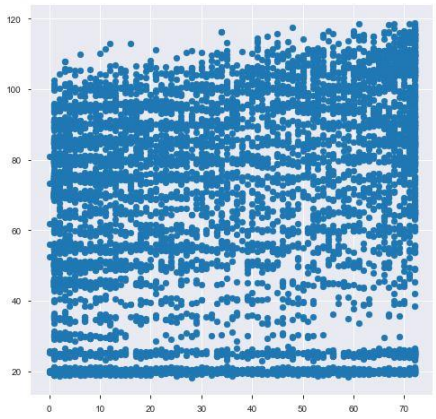

Figure 14. A Simple Scatter Plot to Visualize the Significance of the Two Features

Based on the scatter plot in Figure 14, the diagram does not show or reflect the effective grouping of the clusters for the Telecom customer churn prediction. Therefore, there is a need to apply KMeans clustering model to the dataset. The KMeans() clustering classifier was called as shown below:

```
# To Create a clustering scheme with 5 clusters.
model_kmeans = KMeans(n_clusters=5)        # use this for all inputs
model_kmeans_2vars = KMeans(n_clusters=5)   # use this for 2-input example below
model_kmeans
model_kmeans_2vars
model_kmeans_predict = model_kmeans.fit_predict(data)
model_kmeans_predict
```

Generally, K-Means clustering, or unsupervised learning problem has no target variable to predict. The relationship between the Tenure and Monthly Charges feature will be visualized using this code.

```
# Subset your dataset into 5 cluster subsets (dataframes)
fig = plt.figure(figsize=(8, 8))
C0 = data[data.Cluster == 0]
C1 = data[data.Cluster == 1]
C2 = data[data.Cluster == 2]
C3 = data[data.Cluster == 3]
C4 = data[data.Cluster == 4]
plt.scatter(C0.tenure, C0.MonthlyCharges, color='blue')
plt.scatter(C1.tenure, C1.MonthlyCharges, color='red')
plt.scatter(C2.tenure, C2.MonthlyCharges, color='green')
plt.scatter(C3.tenure, C3.MonthlyCharges, color='yellow')
plt.scatter(C4.tenure, C4.MonthlyCharges, color='black')
plt.xlabel('Tenure')
plt.ylabel('MonthlyCharges')
```

Figure 15 serves as the experimental phase because the 2 Dimensional plot does not show clear separation between the different clusters and this occur because the cluster were not built using only 2 dimensional variable (Tenure and Monthly Charges). However, the clusters were built using all the variables in the DataFrame of the dataset.

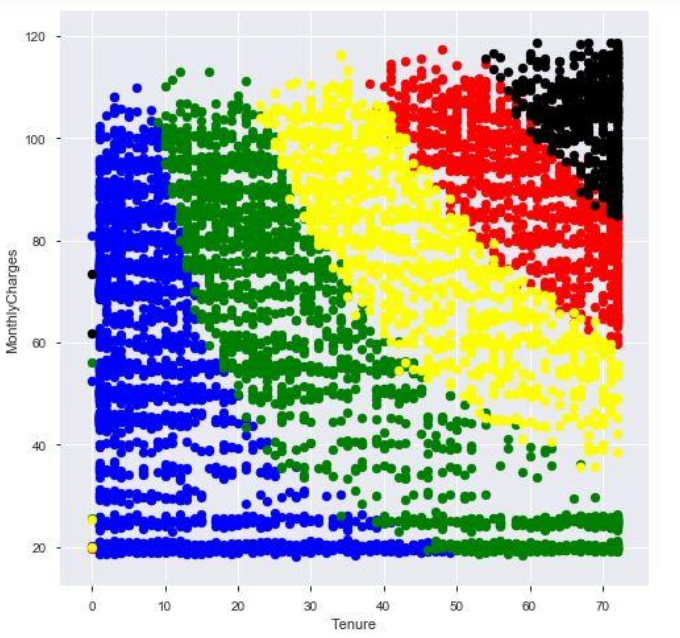

Figure 15. An Experimental KMean Clustering

The new clustering scheme needs to be created using the two input variables and the clustering column needed to be added using appropriate python code as shown below:

Out[290]:

	tenure	MonthlyCharges	Cluster
0	1	29.85	4
1	34	56.95	3
2	2	53.85	4
3	45	42.30	1
4	2	70.70	0
5	8	99.65	0
6	22	89.10	0

To evaluate how well the cluster formed, the inertia of the clusters was calculated which gives an inertia value of 1333107. The optimum number of clusters have to be determined using the "Elbow Curve Method"

```
# Let's evaluate how well the formed clusters are. To do that, we will calculate the inertia of the clusters:

# inertia on the fitted data
model_kmeans.inertia_
```

```
1331073215.721065
```

Elbow Method to Know the Best Number of Clusters different number of clusters were used in this case to experiment with the elbow method in order to figure out the best number of clusters to be used to

build and visualize the K-Means clustering model. The main goal is to minimize the within the cluster sum of square and maximize the distance between clusters using the python code below.

```
k_trials = range(1,15)

sum_squared_error = []

for k in k_trials:

    km_model = KMeans(n_clusters=k)

    km_model.fit(data)

    sum_squared_error.append(km_model.inertia_)

sum_squared_error
```

In order to have effective results from the elbow curve method, the number of clusters used are 5, 6, 7 and 8 respectively. However, the increase in the number of clusters does not reduce or have effect on the sum of square error.

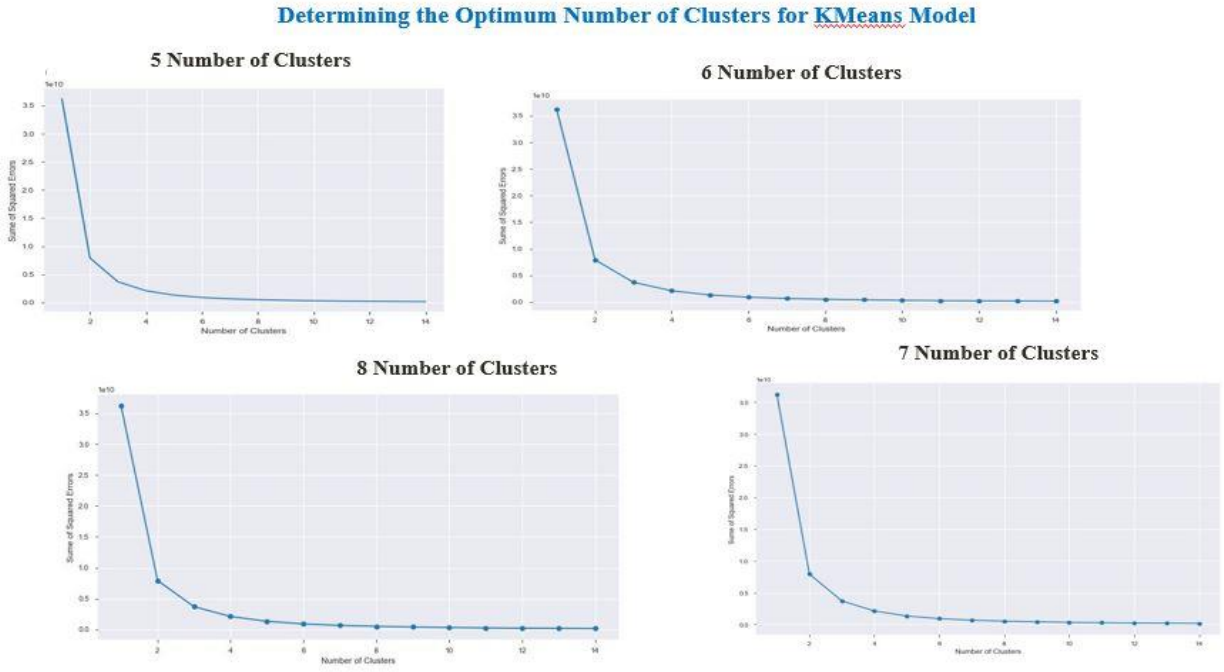

Figure 16. Determining the Optimum Number of Clusters for the KMean Model

Plotting the 5 Clusters using Only the Two Variables

From the various elbow curve shown above using number of clusters from (5 to 8). Any number of clusters between 3 to 5 can be chosen. Therefore, the optimal K for the dataset is 5 and the model is fit.

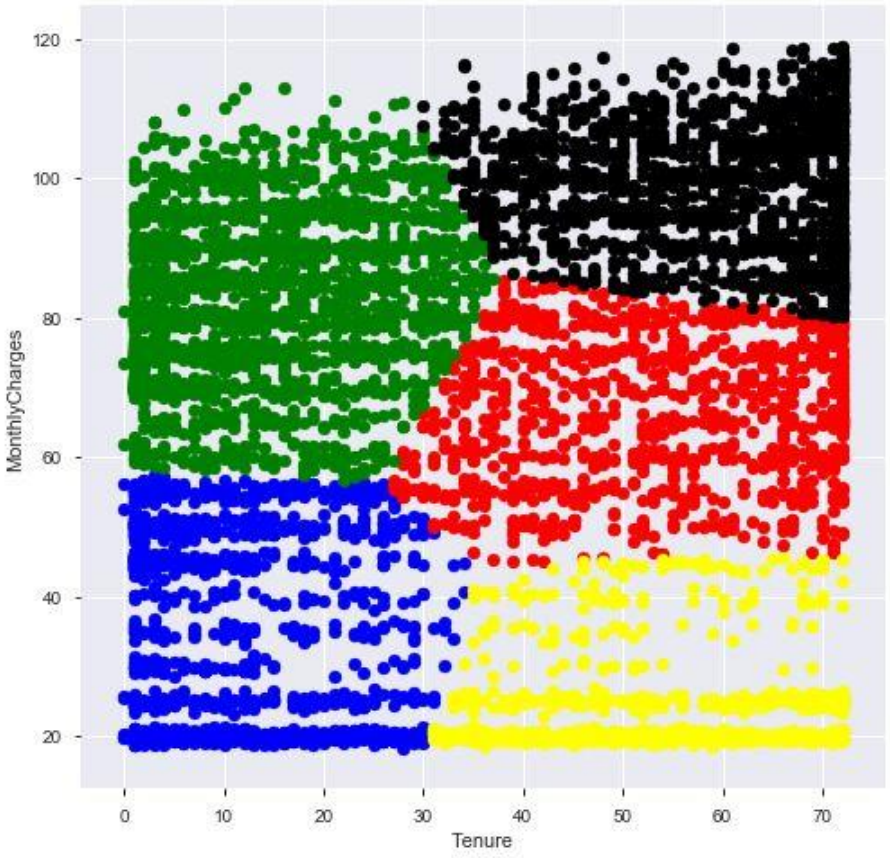

Figure 17. The 5 Number of Clusters Based on Elbow Curve Method

Based on Monthly Charges and Tenure, the following are deduced from the 5 type of clusters in Figure 17 as highlighted below:

- Low Tenure and Low Monthly Charges (Blue)
- Low Tenure and Medium/High Monthly Charges (Green)
- High Tenure and Low Monthly Charges (Yellow)
- High Tenure and Medium Monthly Charges (Red)
- High Tenure and High Monthly Charges (Black)

8. Recommendation and Conclusion

Based on the model evaluation, considering the accuracy of the supervised learning – Logistics, Random Forest and Decision Tree. Logistic Regression and Decision Tree performance are considered better on the dataset. The accuracy for Logistic Regression and Decision Tree is considered suitable for effective prediction. For the unsupervised learning model - K-Means clustering model is also performed better on the dataset. Based on the exploratory data analysis, the causes of churn can be related to [Total charges, Tenure, Monthly charges, Contract type, Payment method, Internet service type, Payment method, Paperless billing. As shown from K-Mean clustering, and looking at the clustering distribution, most customers churn occurs among the "Low Tenure and High Monthly Charges". From decision tree model, customer churn can also occur among customers using fiber optics. This model was generated based on

Churn and the existing telecom customers. Same model can be used on existing customers to find the probability of churn retention which can be put in place before customer leave their services. The K-Means clustering can assist to find churn probability distribution that can be used to identify customer with high or low risk to churn. The customer churn risk types can be measured using this range of scales as highlighted below:

- Very High >= 80%,
- High 60%<= or <= 80%
- Medium Risk 40%<= or <60%
- Low 20%<= or < 40%
- Very Low 0%< or < 20%

The model can be worked upon from time to time, as new data collected and analyzed. To make more accurate prediction, some other data sources can be included such as customer inquiries, seasonality in sales, more demographic information.

References

Ahmad, A.K., Jafar, A. & Aljoumaa, K. (2019). Customer churn prediction in telecom using machine learning in big data platform. *Journal of Big Data* 6,**(**28). https://doi.org/10.1186/s40537-019-0191-6.

Brownlee, J. (2016). Metrics to Evaluate Machine Learning Algorithms in Python. https://machinelearningmastery.com/metrics-evaluate-machine-learning-algorithms-python/. Date of access: 07 May 2020.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. and Wirth, R. (2000). CRISP-DM 1.0. Step-by-Step Data Mining Guide. https://www.the-modeling-agency.com/crisp-dm.pdf.

Fumo, D. (2017). Types of Machine Learning Algorithms You Should Know. https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861. Date of access: 08 May 2020.

Geeksforgeeks. (2020). Confusion Matrix in Machine Learning**.** https://www.geeksforgeeks.org/confusion-matrix-machine-learning/. Date of access: 07 May 2020.

Kumar, S. A and Chandrakala, D. (2016). A Survey on Customer Churn Prediction using Machine Learning Techniques. *International Journal of Computer Applications*, 154(10):13.

Scikit-learn. (2020). 1.10. Decision Trees. https://scikit-learn.org/stable/modules/tree.html. Date of access: 08 May 2020.

Singh, K. (2019). How to Train a Decision Tree Classifier for Churn Prediction. https://dimensionless.in/how-to-train-decision-tree-classifier-for-churn-prediction/. Date of access: 07 May 2020.

Authors

Michael Alabi
York University School of Continuing Studies
https://learn.continue.yorku.ca/user/view.php?id=32786