TELECOM CUSTOMER CHURN PREDICTION AND VISUALIZATION

Parvathy Vysakh
Artificial Intelligence & Machine Learning
Lambton College
Ontario, Canada
c0818539@mylambton.ca

Patricia Adolph
Artificial Intelligence & Machine Learning
Lambton College
Ontario, Canada
c0816792@mylambton.ca

Pooja Selby
Artificial Intelligence & Machine Learning
Lambton College
Ontario, Canada
c0821687@mylambton.ca

Suchithra Chandrasekharan
Artificial Intelligence & Machine Learning
Lambton College
Ontario, Canada
c0816811@mylambton.ca

***Abstract*— Customer churn is the percentage of customers that stopped using a company's product or service during a certain time frame. It is a major problem and one of the most important concerns for large companies. Due to the direct effect on the revenues of the companies, especially in the telecom field, companies are seeking to develop means to predict potential customer to churn. Therefore, finding factors that increase customer churn is important to take necessary actions to reduce this churn. The main contribution of our work is to develop as many data visualization as possible to get a better understanding of the data and make the process of exploratory data analysis streamline and easily analyze data using wonderful plots and charts. And this in turn helps in building a churn prediction model which assists telecom operators to predict customers who are most likely subject to churn. Predictive analytics use churn prediction models that predict customer churn by assessing their propensity of risk to churn. Since these models generate a small prioritized list of potential defectors, they are effective at focusing customer retention marketing programs on the subset of the customer base who are most vulnerable to churn.**

Keywords—NLP, Telephone service companies, predictive analytics, churn

I. INTRODUCTION

For Telecom companies it is key to attract new customers and at the same time avoid contract terminations (=churn) to grow their revenue generating base. Looking at churn, different reasons trigger customers to terminate their contracts, for example better price offers, more interesting packages, bad service experiences or change of customers' personal situations. In short, Customer churn is the act of a customer ending a subscription to a service provider and choosing the services of another company. Companies must reduce customer churn because it weakens the company[1].

Owing to fierce competition among telecom companies, customer churn is inevitable. Companies usually make a distinction between voluntary churn and involuntary churn. Voluntary churn occurs due to a decision by the customer to switch to another company or service provider, involuntary churn occurs due to circumstances such as a customer's relocation to a long-term care facility, death, or the relocation to a distant location. Churn analytics provides valuable capabilities to predict customer churn and also define the underlying reasons that drive it. The churn metric is mostly shown as the percentage of customers that cancel a product or service within a given period (mostly months).

Telecom companies apply machine learning models to predict churn on an individual customer basis and take counter measures such as discounts, special offers or other gratifications to keep their customers. A customer churn analysis is a typical classification problem within the domain of supervised learning.This project aims at predicting the Telecom customer churn and uses visualization as the key for helping the model building process [1].

Visual exploration of data is the first thing one tends to do when dealing with a new task. We do preliminary checks and analysis using graphics and tables to summarize the data and leave out the less important details. It is much more convenient for us, humans, to grasp the main points this way than by reading many lines of raw data. It is amazing how much insight can be gained from seemingly simple charts created with available visualization tools.

II. THE SUBJECTS OF THE STUDY

Data has been collected from Kaggle website. The customer churn dataset is an open-source dataset that contains 20 features and 3333 rows. The feature 'Churn' shows customer churn or non-churn based on existing conditions. Approximately 14.5% of the 'Churn' is 'T' label, and 84.5% of 'nonchurn' is 'F' label [2].Each row represents a customer; each column contains customer's attributes. The datasets have the following attributes or features:

State: string

Account length: integer

Area code: integer

International plan: string

Voice mail plan: string

Number vmail messages: integer

Total day minutes: double

Total day calls: integer

Total day charge: double

Total eve minutes: double

Total eve calls: integer

Total eve charge: double

Total night minutes: double

Total night calls: integer

Total night charge: double

Total intl minutes: double

Total intl calls: integer

Total intl charge: double

Customer service calls: integer

Churn: string

This data has is a large dataset with a lot of details about a particular telecom service providers attributes related to the state they provide service, the duration for which a customer remains a client, provision of voicemail or international call package, Total number, duration and charge for service and calls made locally and internationally etc. We will spent a considerable amount of time to understand it and to know its features and storing format. The information is a subset of telecom customer data from nine months of data sets contained about ten million customers and a total number of columns about ten thousand columns [2].

After importing both the training and testing data, we check information about the train set by calling the .info() method. The data does not contain any null values. There are two types of numerical data in the dataframe — int64 and float64. The categorical data is mentioned against object datatype

Fig. 1. Dataset description

The dataset has both quantitative and categorical data. Churn, is our target variable. It is binary: True indicates that that the company eventually lost this customer, and False indicates that the customer was retained. Later, we will build models that predict this feature based on the remaining features. This is why we call it a target[2].

III. MILESTONE

The pipeline used for this example consists of 6 steps:

- Step 1: Problem Definition
- Step 2: Data Collection
- Step 3: Exploratory Data Analysis (EDA)
- Step 4: Feature Engineering
- Step 5: Train/Test Split
- Step 6: Model Evaluation Metrics Definition

In the context of this project, this is a problem of supervised classification and Machine Learning algorithms will be used for the development of predictive models and evaluation of accuracy and performance. However, the main objective of the project is to analyze the dataset with as much as visualization and preprocessing as possible so that the dataset becomes good enough for implementing a powerful machine learning model. It seeks to find the most appropriate model for the business. We also evaluate the accuracy of the model predictions.

IV. REQUIREMENT

The libraries used are generic and are as follows:

- Pandas (Library for handling tables and help in Data manipulation and analysis)
- Matplotlib (Library for plotting and visualization purposes)
- Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python.

Visualization is the central part of Seaborn which helps in exploration and understanding of data. One has to be familiar with Numpy and Matplotlib and Pandas to learn about Seaborn

V. EXPLORATORY DATA ANALYSIS & VISUALIZATION

Exploratory data analysis (EDA) is a task of analyzing data using simple tools from statistics, simple plotting tools. Every machine learning problem solving starts with EDA. It is probably one of the most important part of a machine learning project. The main reasons why we use EDA are:

- Detection of mistakes
- checking of assumptions
- Preliminary selection of appropriate models
- determining relationships among the explanatory variables
- assessing the direction and rough size of relationships between explanatory and outcome variables.

The four types of EDA are univariate non-graphical, multivariate nongraphical, univariate graphical, and multivariate graphical. Data visualization will make the scientific findings accessible to anyone with minimal exposure in data science and helps one to communicate the information easily [6]. It is to be understood that the visualization technique one employs for a particular data set depends on the individual's taste and preference. Need for visualizing data are as follows:

- Understand the trends and patterns of data
- Analyze the frequency and other such characteristics of data
- Know the distribution of the variables in the data.
- Visualize the relationship that may exist between different variables

The number of variables of interest featured by the data classifies it as univariate, bivariate, or multivariate. For eg., If the data features only one variable of interest then it is a univariate data. Further, based on the characteristics of data, it can be classified as categorical/discrete and continuous data. Different types of analysis:

- Univariate (U) : In univariate analysis we use a single feature to analyze its properties.
- Bivariate (B): When we compare the data between exactly 2 features then its called bivariate analysis.
- Multivariate (M): Comparing more than 2 variables is called as Multivariate analysis.

Most common types of plots used in data visualization include Scatter plot (B),Pair plot (M),Box plot (U),Violin plot(U),Distribution plot (U),Joint plot (U) & (B),Bar chart (B) and Line plot (B). We need to import libraries for data visualization. First we need to import two important libraries for data visualization :Matplotlib is a python library used extensively for the visualization of data. While Seaborn is a python library based on matplotlib. Seaborn provides a high-level interface for drawing attractive and informative statistical graphics.

Univariate methods look at one variable (data column) at a time, while multivariate methods look at two or more variables at a time to explore relationships. Usually our multivariate EDA will be bivariate (looking at exactly two variables), but occasionally it will involve three or more variables. It is almost always a good idea to perform univariate EDA on each of the components of a multivariate EDA before performing the multivariate EDA.

With the growing market, the size of data is also growing. It becomes harder for companies to make decision without proper analyzing it. With the use of charts and certain graphs, one can make sense out of the data and check whether there is any relationship or not [6].

Data Visualization is the process of understanding the data in more detail using some plots and graphs. There are many libraries in Python that help us to do the same. One of the most famous libraries is matplotlib which can plot almost every type of plot that you can imagine. We will be using graphical plots from seaborn and matplotlib libraries for the accomplishment of our tasks. Seaborn is built on top of the matplotlib library. It has many built-in functions using which you can create beautiful plots with just simple lines of codes. It provides a variety of advanced visualization plots with simple syntax like box plots, violin plots, dist plots, Joint plots, pair plots, heatmap, and many more. Matplotlib is the most popular python plotting library. It is a low-level library with a Matlab like interface which offers lots of freedom at the cost of having to write more code.

Various plots are used to determine any conclusions. This helps the company to make a firm and profitable decisions. Once Exploratory Data Analysis is complete and insights are drawn, its feature can be used for supervised and unsupervised machine learning modelling.

In the Exploratory Data Analysis, we examine the customer churn in data, Churn, is our target variable. It is binary: True indicates that that the company eventually lost this customer, and False indicates that the customer was retained. Later, we will build models that predict this feature based on the remaining features. This is why we call it a target.

A. Customer Churn in data

With the aim of understanding the nature of customer churn in training data, we will plot a pie chart which shows us that just above 85% of the customers wish to stay back to receive services from the company while a minority of 14.6% choose to opt out. We're trying to predict users that left the company in the previous month. It's a binary classification problem with an unbalanced target.

Fig. 2. Pie Chart representation of target feature churn

B. Univariate Visualization

Univariate analysis looks at one feature at a time. When we analyse a feature independently, we are usually interested of its values and ignore other features in the dataset.

a. Quantitative features: Quantitative features take on ordered numerical values. These values can be discrete (i.e. integers), or continuous (i.e. real numbers) and usually express a count or measurement. There are several different approaches to visualizing a distribution, perhaps the most common approach to visualizing a distribution is the histogram. This is the default approach in displot(), which uses the same underlying code as histplot(). A histogram is a bar plot where the axis representing the data variable is divided into a set of discrete bins and the count of observations falling within each bin is shown using the height of the corresponding bar. For this analysis, we plot the histograms for the features total day minutes and total intl calls as histograms groups values into bins of equal value range. The shape of the histogram provides the idea about the underlying distribution type and we can also spot any skewness in the shape, knowing a distribution of the feature values is important when we use machine leaning methods that assume a particular type of it which is most often Gaussian [6].

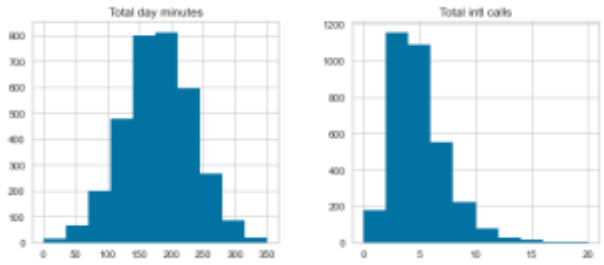

Fig. 3. Distribution of Total day minutes and Total intl calls using histograms

In the above plot, we see that the variable Total day minutes is normally distributed, while Total intl calls is prominently skewed right (its tail is longer on the right).

A density plot is like a smoother version of a histogram. Generally, the kernel density estimate is used in density plots to show the probability density function of the variable. A continuous curve, which is the kernel is drawn to generate a smooth density estimation for the whole data.

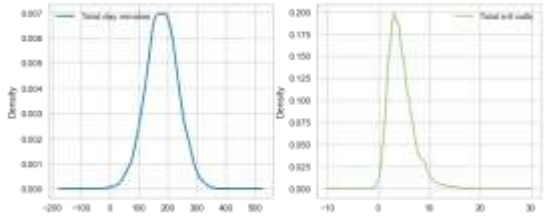

Fig. 4. Distribution of Total day minutes and Total intl calls using densityplot

A distplot graph function combines the matplotlib hist() function with the seaborn kdeplot() and rugplot() functions.

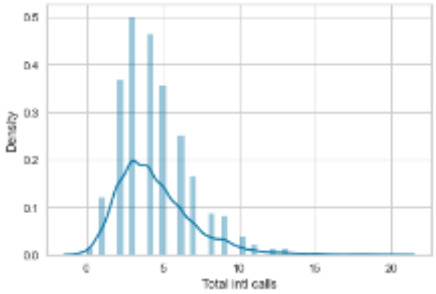

Fig. 5. Distribution of Total day minutes and Total intl calls using distplot

The above graph shows that the height of the histogram bars here is normed and shows the density rather than the number of examples in each bin.

A box-plot is a very useful and standardized way of displaying the distribution of data based on a five-number summary (minimum, first quartile, second quartile(median), third quartile, maximum). It helps in understanding these parameters of the distribution of data and is extremely helpful in detecting outliers. The box by itself illustrates the interquartile spread of the distribution; its length is determined by the 25th(Q1) and 75th(Q3) percentiles. The vertical line inside the box marks the median (50%) of the distribution. The whiskers are the lines extending from the box. They represent the entire scatter of data points, specifically the points that fall within the interval $(Q1-1.5\cdot IQR, Q3+1.5\cdot IQR)$, where $IQR=Q3-Q1$ is the interquartile range. Outliers that fall out of the range bounded by the whiskers are plotted individually as black points along the central axis.

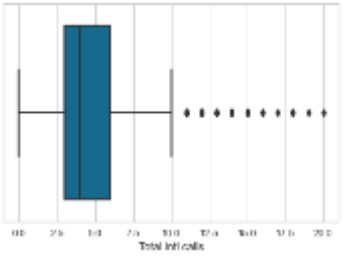

Fig. 6. Distribution of Total day minutes and Total intl calls using boxplot

We can see that a large number of international calls is quite rare in our data.

b. Categorical and binary features: Categorical features take on a fixed number of values. Each of these values assigns an observation to a corresponding group, known as a category, which reflects some qualitative property of this example. Binary variables are an important special case of categorical variables when the number of possible values is exactly 2. If the values of a categorical variable are ordered, it is called ordinal.

We check the distribution of our target variable (i.e. churn) using frequency table, which shows how frequent each value of the categorical variable is. By default, the entries in the output are sorted from the most to the least frequently-occurring values. In our case, the data is not balanced; that is, our two target classes, loyal and disloyal customers, are not represented equally in the dataset. Only a small part of the clients canceled their subscription to the telecom service.

The bar plot is a univariate data visualization plot on a two-dimensional axis. One axis is the category axis indicating the category, while the second axis is the value axis that shows the numeric value of that category, indicated by the length of the bar.The plot.bar() function plots a bar plot of a categorical

variable. The value_counts() returns a series containing the counts of unique values in the variable.

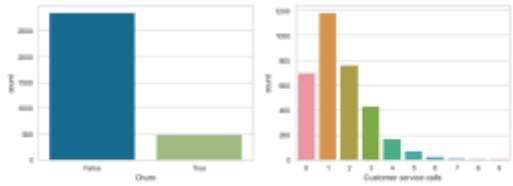

Fig. 7. Barplot for the distribution of categorical variables churn and customer service calls

The left chart above vividly illustrates the imbalance in our target variable. The bar plot for Customer service calls on the right gives a hint that the majority of customers resolve their problems in maximum 2-3 calls. But, as we want to be able to predict the minority class, we may be more interested in how the fewer dissatisfied customers behave. It may well be that the tail of that bar plot contains most of our churn. These are just hypotheses for now, so let's move on to some more interesting and powerful visual techniques.

C. Multivariate Visualization

Multivariate plots allow us to see relationships between two and more different variables, all in one figure. Just as in the case of univariate plots, the specific type of visualization will depend on the types of the variables being analyzed. When you have a bivariate data, you can easily visualize the relationship between the two variables by plotting a simple scatter plot.For a data set containing three continuous variables, you can create a 3d scatter plot. Heatmaps visualise data through variations in colouring. When applied to a tabular format, Heatmaps are useful for cross-examining multivariate data, through placing variables in the rows and columns and colouring the cells within the table. Heatmaps are good for showing variance across multiple variables, revealing any patterns, displaying whether any variables are similar to each other, and for detecting if any correlations exist in-between them.For a small data set with more than three variables, it's possible to visualize the relationship between each pairs of variables by creating a scatter plot matrix. You can also compute a correlation analysis between each pairs of variables.

a. Quantitative vs. Quantitative: As a first step towards analyzing the correlations among the numerical variables in our dataset this, we drop the non-numerical variables and use heatmap, scatterplot and scatterplot matrix for this analysis.

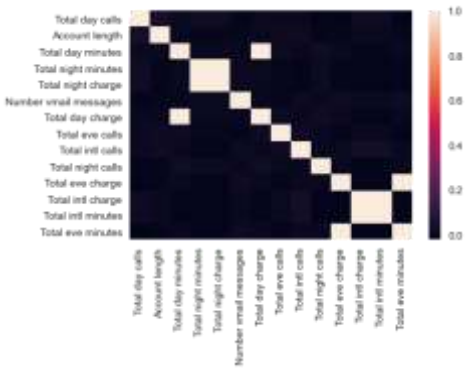

Fig. 8. Multivariate analysis using Correlation matrix

From the colored correlation matrix generated above, we can see that there are 4 variables such as Total day charge that have been calculated directly from the number of minutes spent on phone calls (Total day minutes). These are called dependent variables and can therefore be left out since they do not contribute any additional information, we get rid of them.

The scatter plot displays values of two numerical variables as Cartesian coordinates in 2D. Scatter plots in 3D are also possible.

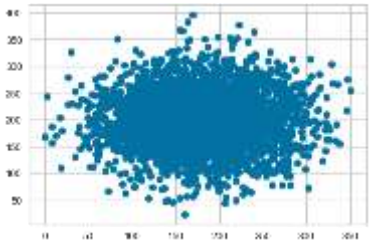

Fig. 9. Multivariate analysis using Scatterplot

We get an uninteresting picture of two normally distributed variables. Also, it seems that these features are uncorrelated because the ellipse-like shape is aligned with the axes.

A scatter plot matrix is a grid (or matrix) of scatter plots used to visualize bivariate relationships between combinations of variables. Each scatter plot in the matrix visualizes the relationship between a pair of variables, allowing many relationships to be explored in one chart.

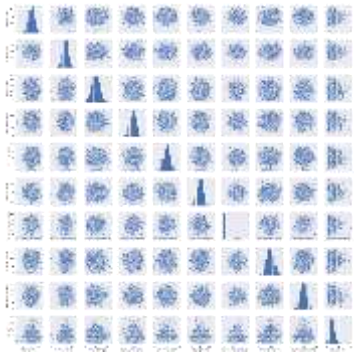

Fig. 10. Multivariate analysis using Scatterplot matrix

c. Quantitative vs. Categorical: In order to gain new insights for churn prediction from the interactions between the numerical and categorical features, we will try to interpret how the input variables are related to the target variable 'churn'. The points in scatterplot can be color or size coded so that the values of a third categorical variable are also presented in the same figure. Here we are going to use lmplot() and the parameter hue to indicate our categorical feature of interest.

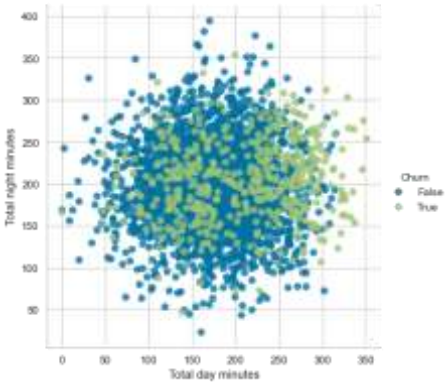

Fig. 11. Multivariate analysis using lmplot()

It seems that our small proportion of disloyal customers lean towards the top-right corner; that is, such customers tend to spend more time on the phone during both day and night. Box plots are used to visualize the distribution statistics of the numerical variables in two disjoint groups: the loyal customers (Churn=False) and those who left (Churn=True).

A possible solution which also uses matplotlib is to create the figure and subplots then pass the axes into df.boxplot() using the argument.

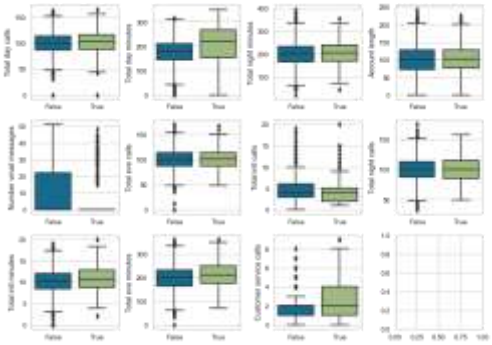

Fig. 12. Multivariate analysis using Scatterplot matrix

From the above charts, we can notice that there is a lot of discrepancy in distribution between the two groups is for three variables, 'total day minutes', 'cusotmer service calls' and 'number vmail messages'.

d. Categorical vs. Categorical: The variable Customer service calls has few unique values and, thus, can be considered either numerical or ordinal. The relationship between this ordinal feature and the target variable Churn will be shown using count plot. Let's look at the distribution of the number of calls to the customer service, again using a count plot. This time, let's also pass the parameter hue=Churn that adds a categorical dimension to the plot

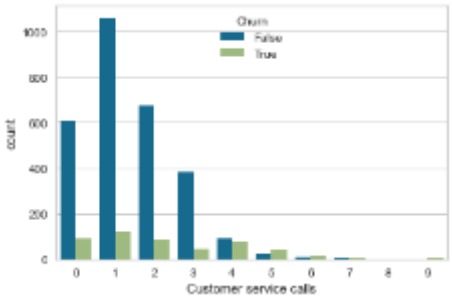

Fig. 13. Multivariate analysis customer service calls and churn using countplot

The graph above shows that the churn rate increases significantly after 4 or more calls to customer service.

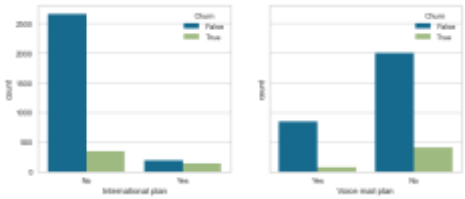

Fig. 14. Multivariate analysis of Churn and the binary features, International plan and Voice mail plan shows that when International Plan using countplot

The analysis of the relationship between Churn and the binary features, International plan and Voice mail plan shows that when International Plan is enabled, the churn rate is much higher; the usage of the international plan by the customer is a strong feature. We do not observe the same effect with Voice mail plan.

To see how Churn is related to the categorical variable State by creating a cross tabulation, we use the contingency table. It represents multivariate frequency distribution of categorical variables in tabular form. In particular, it allows us to see the distribution of one variable conditional on the other by looking along a column or row.

Fig. 15. Multivariate analysis of Churn using contingency table

In the case of State, the number of distinct values is rather high: 51. We see that there are only a few data points available for each individual state – only 3 to 17 customers in each state abandoned the operator. Let's ignore that for a second and calculate the churn rate for each state, sorting it from high to low.

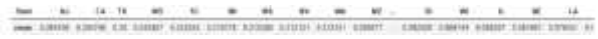

Fig. 16. Multivariate analysis of Churn using contingency table after sorting

At first glance, it seems that the churn rate in New Jersey and California are above 25% and less than 6% for Hawaii and Alaska.

e. Variable distribution: To plot multiple pairwise bivariate distributions in a dataset, you can use the pairplot() function. This shows the relationship for (n, 2) combination of variable in a DataFrame as a matrix of plots and the diagonal plots are the univariate plots. Several of the numerical data are correlated. (Total day minutes and Total day charge), (Total eve minutes and Total eve charge), (Total night minutes and Total night charge) and lastly (Total intl minutes and Total intl charge) are also correlated. We only have to select one of them.

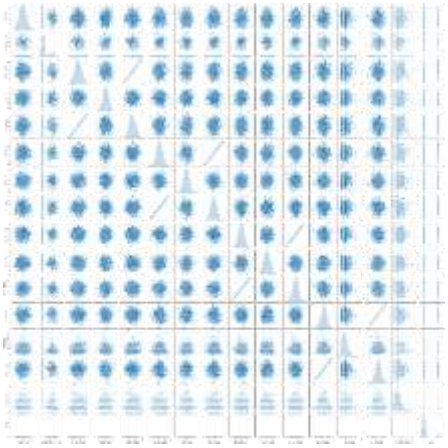

Fig. 17. Bivariate distributions using pairplot

VI. DATA PREPROCESSING

The data pre-processing involves checking out for lost values, seeking out for categorical values, part the dataset into preparing and test set and finally do a highlight scaling to constrain the range of factors. As a part of data preprocessing it is necessary to eliminate some of the correlated and unnecessary columns. Since the data set is a mixture of numerical and categorical columns, we also separate them. We use label encoding for the binary columns. In addition after the encoding and scaling, we drop original values merging scaled values for numerical columns. To have a better picture of the necessary variables, we prepare a summary report and correlation matrix as well [5].

Fig. 18. Training variable Summary

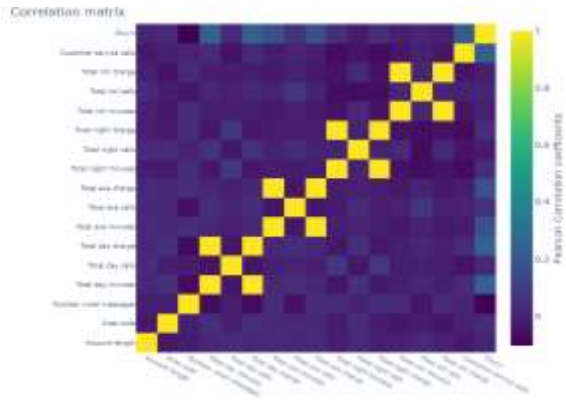

Fig. 19. Heat map

This is a correlation matrix plot to understand distribution of correlation between the numerical variables. As we can see from the plot, most of the variables are not too much related with each other, except for the variables that account for total minutes of day, evening, night and international calls with their charges, something which is to be expected of the nature of data we are dealing with.

Principal component analysis (PCA) is a multivariate data analysis approach that allows us to summarize and visualize the most important information contained in a multivariate data set. PCA reduces the data into few new dimensions (or axes), which are a linear combination of the original variables. You can visualize a multivariate data by drawing a scatter plot of the first two dimensions, which contain the most important information in the data [3].

Fig. 20. Visualizing data with PCA

A Radar Chart (also known as a spider plot or star plot) displays multivariate data in the form of a two-dimensional chart of quantitative variables represented on axes originating from the center. The relative position and angle of the axes is typically uninformative. It is equivalent to a parallel coordinates plot with the axes arranged radially. For a Radar Chart, use a polar chart with categorical angular variables, with px.line_polar, or with go.Scatterpolar. Here we plot the radar plot for churn and not churn customers[5].

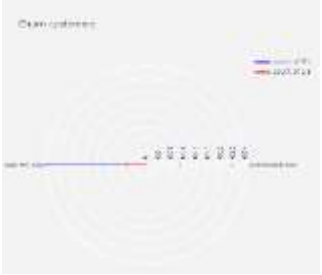

Fig. 21. Radar plot for churn customers

Fig. 22. Radar plot for not churn customers

VII. MODELING

As the first step towards model building we perform the train test split and based on the initial analysis logistic regression.

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The algorithm is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement [4].

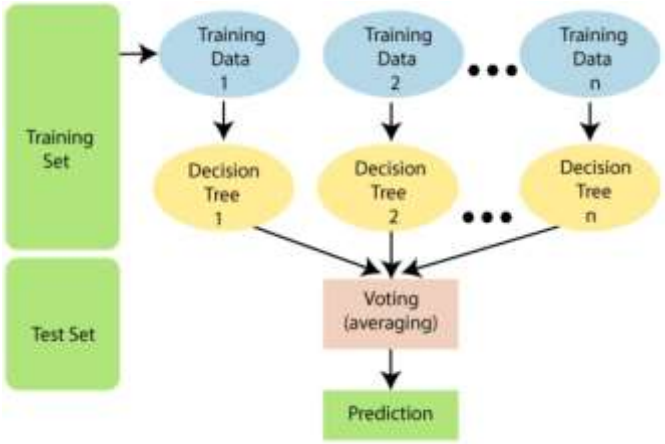

Fig. 23. Rnadom Forest Algorithm

A. *Below are two assumptions for a better Random Forest classifier:*

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.
- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

B. *Advantages of Random Forest*

- Random Forest is capable of performing both Classification and Regression tasks.

- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

C. Disadvantages of Random Forest

- Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.
- The Random Forest model is difficult to interpret.
- For large datasets with too many features, the random forest model consumes a lot of memory.
- The algorithm produces wild predictions for test observations never seen before by the training data. For example, for a training data containing two variables x and y with the range of x variable from 50 to 80: If the test data has x = 150, the algorithm would give an unreliable prediction.

D. Model Performance Metrics

While data preparation and training a machine learning model is a key step in the machine learning pipeline, it's equally important to measure the performance of this trained model. How well the model generalizes on the unseen data is what defines adaptive vs non-adaptive machine learning models.

By using different metrics for performance evaluation, we should be in a position to improve the overall predictive power of our model before we roll it out for production on unseen data.

Without doing a proper evaluation of the ML model using different metrics, and depending only on accuracy, it can lead to a problem when the respective model is deployed on unseen data and can result in poor predictions.

a. Confusion Matrix:

A confusion matrix is a matrix representation of the prediction results of any binary testing that is often used to describe the performance of the classification model (or "classifier") on a set of test data for which the true values are known. The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes. The AUC, we have got from our model is 0.81 which is high indicating the better performance of our model

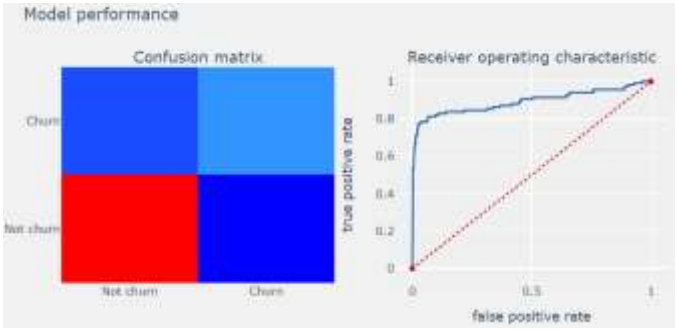

Fig. 24. Confusion Matrix and Area under Curve

The feature importance describes which features are relevant. It can help with better understanding of the solved problem and sometimes lead to model improvements by employing the feature selection. Visualizing the feature importance of our model, it is evident that the variable Total Day in Minutes is the most important features whereas Voice mail plan is the least important feature.

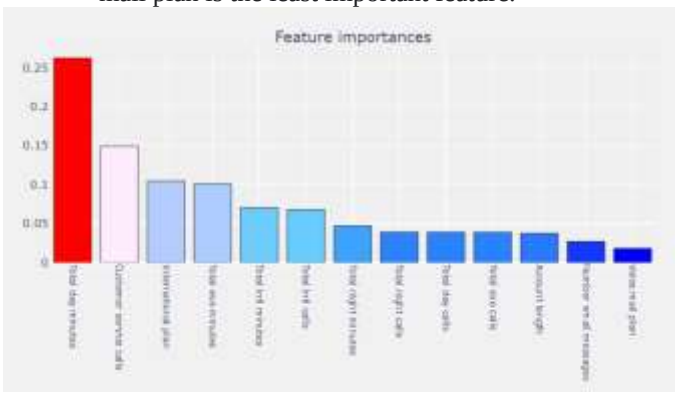

Fig. 25. Feature Importance

A set of different thresholds are used to interpret the true positive rate and the false positive rate of the predictions on the positive (minority) class, and the scores are plotted in a line of increasing thresholds to create a curve.

The score rate is plotted on the x-axis and the discrimination threshold is plotted on the y-axis. A diagonal line on the plot from the bottom-left to top-right indicates the "*curve*" for a no-skill classifier (predicts the majority class in all cases), and a point in the top left of the plot indicates a model with perfect skill [6].

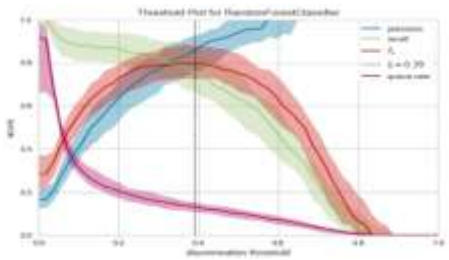

Fig. 26. Threshold plot for Random Forest Classifier

Classification report:

	precision	recall	f1-score	support
0	0.94	0.99	0.97	717
1	0.91	0.64	0.75	117
accuracy			0.94	834
macro avg	0.93	0.82	0.86	834
weighted avg	0.94	0.94	0.94	834

Accuracy Score: 0.9412470023980816
Area under curve: 0.8156313700246755

Fig. 27. Classification Report

Accuracy is a common evaluation metric for classification problems. It's the number of correct predictions made as a ratio of all predictions made. Accuracy of our model is: 94%

VIII. Conclusion

Proper churn management can save a huge amount of money for the company. Thus the economic value of customer retention can be summarized as :

- satisfied customers can bring new customers long-term customers are usually do not get influenced much by competitors
- long-term customers tend to buy more company can focus on satisfying existing customer's needs
- lost customers share negative experience and thus will have negative influence on the image of the company. Thus customer retention as a function of i.e. f{Price, service quality, customer satisfaction, brand image} may lead to better customer loyalty.

We will be analyzing and preprocessing the data through various univariate non-graphical, multivariate non- graphical, univariate graphical, and multivariate graphical methods and make the dataset ready for prediction of churn.

References

[1] García, D.L.; Nebot, À.; Vellido, A. Intelligent data analysis approaches to churn as a business problem: A survey. Knowl. Inf. Syst. 2017, 51, 719–774. [CrossRef].

[2] Kaggle.com. 2021. Customer churn prediction: Telecom Churn Dataset. [online] Available at: <https://www.kaggle.com/mnassrib/customer-churn-prediction-telecom-churn-dataset/data> [Accessed 18 November 2021].

[3] A. Tam, "Principal Component Analysis for Visualization," *Machine Learning Mastery*, Oct. 20, 2021. https://machinelearningmastery.com/principal-component-analysis-for-visualization/ (accessed Dec. 11, 2021).

[4] "Telecom Customer Churn Analytics | krtrimaIQ," krtrimaiq.ai. https://krtrimaiq.ai/telecom-churn-analytics.html (accessed Dec. 11, 2021).

[5] R. N. Sucky, "Understand the Data With Univariate And Multivariate Charts and Plots in Python," Medium, Jun. 12, 2020. https://towardsdatascience.com/understand-the-data-with-univariate-and-multivariate-charts-and-plots-in-python-3b9fcd68cd8.

[6] IBM Cloud Education, "What is Exploratory Data Analysis?," *www.ibm.com*, Aug. 25, 2020. https://www.ibm.com/cloud/learn/exploratory-data-analysis.