CUSTOMER CHURN PREDICTION USING MACHINE LEARNING

A Report work

In

BUSINESS RESEARCH METHODS

By

M. DATTA PRASAD	**160122672096**
S. KRUSHITHA	**160122672114**
N. VISHAL	**160122672125**

Under the guidance of

Dr. B. LAVANYA

Assistant Professor

Department of MBA

CBIT School of Management Studies

Chaitanya Bharathi Institute of Technology

(AUTONOMOUS)

ACKNOWLEDGEMENT

This project is an acknowledgement to the inspiration, drive and assistance contributed by many individuals. This project would have never seen light of this day without the help and guidance we have received. We would like to express our gratitude to all the people behind the screen who helped us.

It’s our privilege and pleasure to express our profound sense of gratitude to **Dr. B. Lavanya,** Department of MBA for her guidance throughout this dissertation work.

We would like to thank all the respondents who took time to fill the survey form .Lastly, we thank almighty, our parents, friends for their constant encouragement without which this assignment would not be possible.

By

M. DATTA PRASAD	**160122672096**
S. KRUSHITHA	**160122672114**
N. VISHAL	**160122672125**

TABLE OF CONTENTS

Title	Page No.
Abstract	1
Chapter 1: Introduction	2
Chapter 2: Review of Literature	4
Chapter 3: Research Methodology	4
3.1 Objectives of the Study	5
3.2 Research Methodology	5
3.2.1 Data Overview	7
3.2.2 Exploratory Data Analysis	7
3.2.3 Variable Distributions	7
3.3 Data Tools	8
Chapter 4: Data Analysis	8
4.1 Data Preprocessing	9
4.2 Model Building	10
4.2.1 Model Building Without PySpark	10
Model Evaluation	16
4.2.2 Model Building With PySpark	17
Model Evaluation	18
Chapter 5: Findings	21
Chapter 6: Limitations	21
Chapter 7: Suggestions	21
Chapter 8: Conclusion	21
Chapter 9: References	22

LIST OF TABLES

Table No.	Description
1	Definitions of Metrics
2	Performance Report of ML models over Training Dataset
3	Model Evaluation Results
4	Model Evaluation using Stratified data

LIST OF FIGURES

Figure No.	Description
1	Flow Chart of Proposed Model
2	Chart of Customer Churn Percentage in Training Data
3	Classification Report for Logistic Regression Model
4	Classification Report for Decision Tree Classifier Model
5	Classification Report for Random Forest Classifier Model
6	Classification Report for Gaussian NB Model
7	Classification Report for SVC Model
8	Classification Report for SVC-RBF Model

ABSTRACT

The Orange Telecom is a telecommunication service provider (TSP) which traditionally provides telephone, mobile phone networks, modern cloud service systems, mobile data transmission and similar services. Due to the high cost of acquiring new customers, customer Churn prediction has emerged as an indispensable part of telecom sectors' strategic decision making and planning process. It is important to forecast customer churn behavior in order to retain those customers that will churn or possible may churn.

We have collected Telecom Customer Churn Dataset which is a secondary data. We performed Exploratory Data Analysis in order to uncover the underlying structure. The prediction process is heavily data driven and often utilizes advanced machine learning techniques. In this project we understand the customer behaviors and identify the customers who will cancel their subscription whether in free or paid tier. We performing some preliminary analysis of the data, and generate churn prediction models - all with PySpark and its machine learning frameworks. We will finally encapsulate the differences between Apache Spark framework, Spark-MLlib and ML.

The objective of our project is to build a churn prediction model which can identify churn customers and non-churn customers and implementing this model by using Naïve Bayes algorithm, further we will implement the same model with other Machine Learning Algorithms, then compare the results of all the models & we will highlight which algorithm is best to build the churn prediction model.

1. INTRODUCTION

Customer churn is one of the pointing issues of today's rapidly developing and competitive Tele-communication industry. The focus of the telecom sector has shifted from acquiring new customer to retaining existing customers because of the associated high cost. The retention of existing customers also leads to improved sales and reduced marketing cost as compared to new customers. These facts have ultimately resulted in customer churn, prediction activity to be an indispensable part of telecom sector's strategic decision making and planning process. Customer retention is one of the main objectives of customer relationship management (CRM). Its importance has led to the development of various tools that support some important tasks in predictive modeling and classification.

Why Churn Is Important: Customer churn – shifting from one service provider to next competitor in the market. It is a key challenge in highly competitive markets, which is highly observed in telecommunication sector. Companies make revenue from their customers. This is the basics of every business. If one company is losing their customers whatever the reasons, revenues will be decreased. Churn reasons can be customer experience problems, high prices, and low quality and so on. It depends on the business area actually.

From the analysis of various surveys, customer Churn has been broadly classified into three kinds:

- Active churner (Volunteer): The customers who take the decision to quit the contract and choose the service or the product of the next provider.

- Passive churner (Non-Volunteer): When the company ceases the service to a customer due to some reasons like not paying the money for subscription or if they have identified that the customer is fraud.

- Rotational churner (Silent): The customers who discontinues the usage of service or the products without prior decision or prior knowledge to both company & customer. This generally happens when the customer involuntarily stop using the services or product of a company due to some of the reasons like his busy schedule or unavailability of particular product/service in their dwelling place.

The first two kinds of Churns can be predicted easily with the help of traditional approaches in term of the Boolean class value, but the third type of churn may exist which is difficult to predict because there may have such type of customers who may possibly churns in near future. It should be the goal of the decision maker and marketers to decrease the churn ratio.

Proposed Model

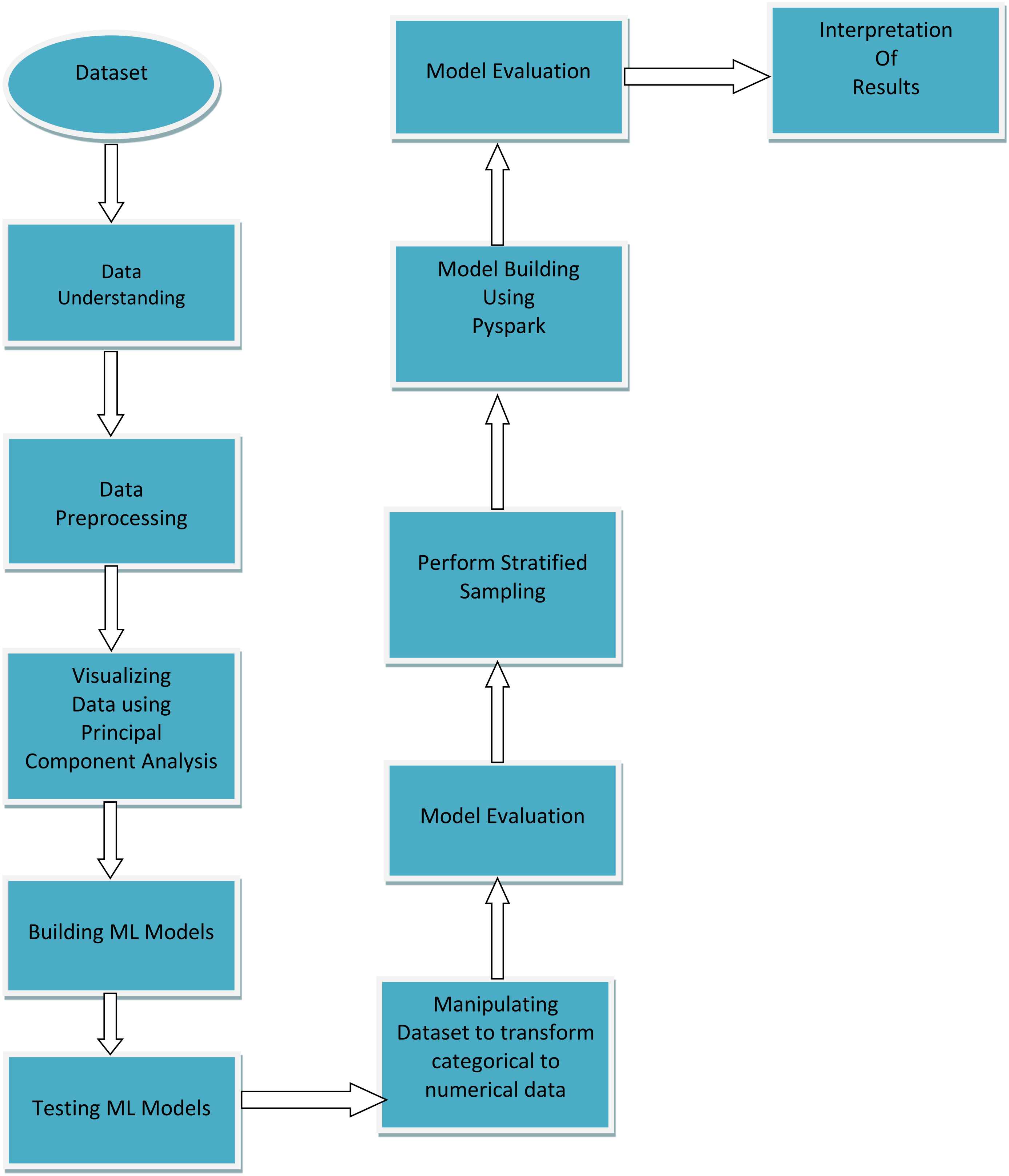

Dataset
Data Understanding
Data Preprocessing
Visualizing Data using Principal Component Analysis
Building ML Models
Testing ML Models
Manipulating Dataset to transform categorical to numerical data
Model Evaluation
Perform Stratified Sampling
Model Building Using Pyspark
Model Evaluation
Interpretation Of Results

Fig.no.1 Flow Chart of Proposed Model

2. REVIEW OF LITERATURE

A Amin, A Adnan, S Anwar (2023), “An adaptive learning approach for customer churn prediction in the telecommunication industry using evolutionary computation and Naïve Bayes”. This study proposed an adaptive learning approach for this perplexing problem of CCP using the Naïve Bayes classifier with a Genetic Algorithm (subclass of an Evolutionary Algorithm) based feature weighting approach.

X Xiahou, Y Harada (2022), “B2C E-commerce customer churn prediction based on K-means and SVM”. According to the characteristics of longitudinal timelines and multidimensional data variables of B2C e-commerce customers’ shopping behaviors, this paper proposes a loss prediction model based on the combination of k-means customer segmentation and support vector machine (SVM) prediction. The accuracy of the SVM prediction was higher than that of the logistic regression prediction.

T Zhang, S Moro, RF Ramos (2022), “A data-driven approach to improve customer churn prediction based on telecom customer segmentation”. This study aimed to develop a churn prediction model to predict telecom client churn through customer segmentation. It concluded that the telecom customer churn model constructed by regression analysis had higher prediction accuracy (93.94%) and better results.

D AL-Najjar, N Al-Rousan, H AL-Najjar (2022), “Machine learning to develop credit card customer churn prediction”. This paper aimed to develop credit card customer churn prediction by using a feature-selection method and five machine learning models. The analysis showed that all the machine learning models could predict the credit card customer churn model.

Nhi N.Y. Vo, Shaowu Liu, Xitong Li, Guandong Xu (2021), **“**Leveraging unstructured call log data for customer churn prediction”. In this research, they proposed a customer churn prediction model utilizing the unstructured data, which are the spoken contents in phone communication. The results showed that the model can accurately predict the client churn risks and generate meaningful insights using interpretable machine learning with personality traits and customer segments.

Jasroop Singh , ChadhaPraveen. L, Pratyush.S (2021), **“**Customer churn prediction system: a machine learning approach”. It was found that Adaboost and XGboost Classifier gives the highest accuracy of 81.71% and 80.8% respectively. The highest AUC score of 84%, is achieved by both Adaboost and XGBoost Classifiers which outperforms over others.

Manas Rahman, V. Kumar (2020), “Machine Learning Based Customer Churn Prediction in Banking”.

In this paper, a method predicts the customer churn in a Bank, using machine learning techniques, which is a branch of artificial intelligence, is proposed. The research promotes the exploration of the likelihood of churn by analyzing customer behavior. The KNN, SVM, Decision Tree, and Random Forest classifiers are used in this study.

Ahmad A.M., Jafar. A, Aljoumaa. K (2019), “Customer churn prediction in telecom using machine learning in big data platform”. The model was prepared and tested through Spark environment by working on a large dataset created by transforming big raw data provided by SyriaTel telecom company. the best results were obtained by applying XGBOOST algorithm. This algorithm was used for classification in this churn predictive model.

3. RESEARCH METHODOLOGY

3.1 Objectives of the Study

- To build a churn prediction model
- To identify churn customers and non-churn customers
- To implement this model by using Naïve Bayes Algorithm
- To implement the model with other Machine Learning Algorithms
- To compare the results of all the models
- To highlight the best algorithm to build a churn prediction model

3.2 Research Method

This study is based on Secondary data, which is Telecom Customer Churn Dataset. The data is understood clearly in order to decide the objectives & methodologies. The first step of data understanding includes identification of key variables or predicting variables in the dataset. Before working on any dataset, the dataset is understood completely because it is highly important to deal with missing values, remove outliers & know the categorical columns & numerical columns.

Telecom Customer Churn dataset consists of training set & testing set. The two sets are from the same batch, but have been split by an 80/20 ratio. As more data is often desirable for developing ML models, we will be using the larger set (that is, Churn-80) for training and cross-validation purposes, and the smaller set (i.e., Churn-20) for final testing and model performance evaluation.

3.2.1 Data Overview

```
Overview of the training dataset:

Rows: 2666
```

```
Number of features: 20

Features: ['State', 'Account length', 'Area code', 'International plan', 'Voice mail plan',
'Number vmail messages', 'Total day minutes', 'Total day calls', 'Total day charge', 'Total
eve minutes', 'Total eve calls', 'Total eve charge', 'Total night minutes', 'Total night
calls', 'Total night charge', 'Total intl minutes ', 'Total intl calls', 'Total intl charge',
'Customer service calls', 'Churn']

Missing values: 0

Unique values:

State 51
Account length 205
Area code 3
International plan 2
Voice mail plan 2
Number vmail messages 42
Total day minutes 1489
Total day calls 115
Total day charge 1489
Total eve minutes 1442
Total eve calls 120
Total eve charge 1301
Total night minutes 1444
Total night calls 118
Total night charge 885
Total intl minutes 158
Total intl calls 21
Total intl charge 158
Customer service calls 10
Churn 2
dtype: int64

Overview of the test dataset:

Rows: 667

Number of features: 20

Features: ['State', 'Account length', 'Area code', 'International plan', 'Voice mail plan',
'Number vmail messages', 'Total day minutes', 'Total day calls', 'Total day charge', 'Total
eve minutes', 'Total eve calls', 'Total eve charge', 'Total night minutes', 'Total night
calls', 'Total night charge', 'Total intl minutes ', 'Total intl calls', 'Total intl charge',
'Customer service calls', 'Churn']



Missing values: 0

Unique values:

State 51
Account length 179
Area code 3
International plan 2
Voice mail plan 2
Number vmail messages 37
Total day minutes 562
Total day calls 100
Total day charge 562
Total eve minutes 557
Total eve calls 94
Total eve charge 528
Total night minutes 568
Total night calls 96
Total night charge 453
Total intl minutes 132
Total intl calls 17
Total intl charge 132
Customer service calls 9
Churn 2
dtype: int64
```

3.2.2 Exploratory Data Analysis

Exploratory Data Analysis is performed in order to uncover the underlying structure. The structure of the various data sets determines the trends, patterns, and relationships among them.

For Customer Churn Analysis the foremost analysis that is to identify the percentage of churn customers & percentage of non-churn customers in a company. Based on this analysis, it is seen the distribution of Churn customers of the company in dataset which will help in building an efficient model. Donut chart is plotted for the dataset using go.pie. In go.Pie , data visualized by the sectors of the pie is set in values. The sector labels are set in labels. The sector colors are set in marker.

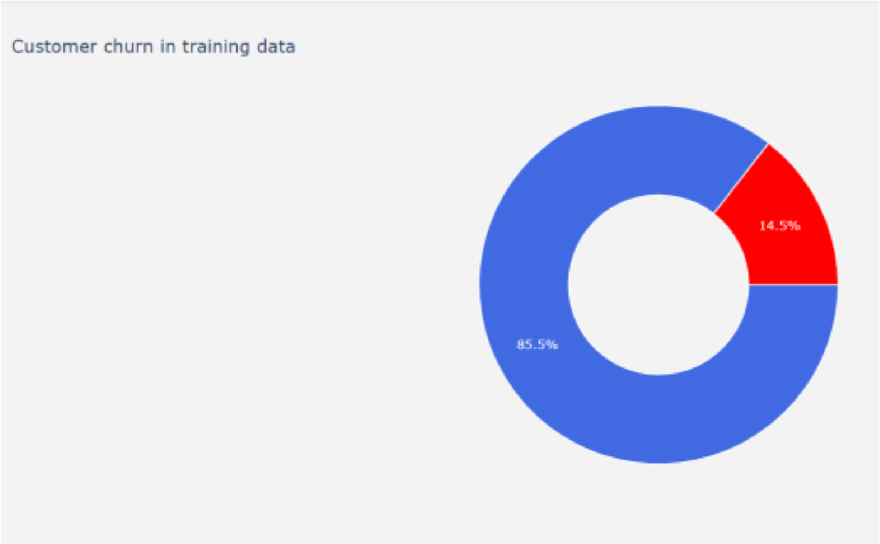


Fig.no.2 Donut Chart of Customer Churn Percentage in Training Data

It is seen that the training dataset of Customer Churn contains 85.5% of customer non-Churn rate & 14.5% of customer Churn rate.

3.2.3 Variable Distributions

To determine the most appropriate statistical analyses to use, it is important to understand the distribution of variables present in dataset. There are 307 samples in the Telecom Customer Churn dataset. Since the dataset is big, consider sample data & try to understand the distribution & relation between the variable through pair plot.

Understandings drawn from the result of pairplot: Several of the numerical data are much 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. The study need to select only one of them.

3.3 Data Tools

The study used JUPYTER NOTEBOOK for data preprocessing, data visualization, building models and evaluating models using Python and Machine Learning.

4. DATA ANALYSIS

System Design of Proposed Model:

❖ **Data preprocessing**
- Variable summary
- Correlation matrix
- Visualizing data with principal components
- Binary variable distributions in customer churn (Radar Chart)

❖ **Model Building**
- Logistic Regression
- Decision Tree Classifier
- Random Forest Classifier
- Gaussian Naive Bayes
- Support Vector Machine
- Support Vector Machine (linear)
- Support Vector Machine (RBF)

o **Model performances over the training dataset**
- Model performance metrics
- Compare model metrics

Work flow for building ML models using PySpark :

- Initializing a Spark session
- Fetching and Importing Churn Data
- Summary Statistics
- Correlations and Data Preparation

- **Using Spark MLlib Package**
  - Decision Tree Models
  - Model Training
  - Model Evaluation
  - Stratified Sampling

- **Using Spark ML Package**
  - Pipelining
  - Model Selection
  - K-fold cross validation
  - Model Evaluation

4.1 Data Preprocessing: It's obvious that there are several highly correlated fields. Such correlated data will not be very beneficial for the model training runs, so we are going to remove them. We will do so by dropping one column of each pair of correlated variables along with the State and Area code columns. In this Data preprocessing module first, we have removed highly correlated & the columns which doesn't have lot of impact on Churn Variable, and then we have assigned the target variable. Separate numerical & categorical Variables.

- **Statistical Summary:** After preprocessing the training Churn dataset, we have identified statistical summary & displayed the output in form of "Table". Summary statistics summarize and provide information about the data. It tells about the values in your data set. This includes where the mean lies and whether your data is skewed.

- **Correlation Matrix:** In order to measure the correlation coefficient between two set of variables we have plotted correlation matrix. The results depict that the variables "Voicemail plan" & "Number of voicemail messages" are highly correlated equal to 1.

- **Visualizing data with Principal Component Analysis:** Visualization is a crucial step to get insights from data. Customer Churn Dataset consists of many dimensions, depicting things in four or five dimensions is impossible because we live in a three-dimensional world and have no idea of how things in such a high dimension would look like. This is where a dimensionality reduction technique such as PCA comes into play.

  PCA in essence is to rearrange the features by their linear combinations. Hence it is called a feature extraction technique. One characteristic of PCA is that the first principal component holds the most information about the dataset. Because PCA is sensitive to the scale, we should normalize each feature by StandardScaler we can see a better result. Here the different classes are more distinctive. So, in Data Preprocessing step we have normalized each feature by StandardScaler.

- **Binary variable distributions in customer churn (Radar Chart):** Radar Charts are used to compare two or more items or groups on various features or characteristics. Hence, we have visualized Radar chart to compare the rate of Churn & non Churn customers with all the variable present in dataset. We get the result of binary variable distribution in customer churn.

4.2 Model Building: We are aware that as title of project "Customer Churn Analysis" suggests it is a kind of task that can be performed using Machine Learning Models, we have built machine learning models. But what if the size of the data is big. This is when the picture of PySpark comes into the picture. Every Telecom Company has massive data that gets generated daily about one particular customer and just imagine we are living in the era of digitalized world, every person living on the earth is a part of any one/two Telecom Companies customer. Thus, massive amounts of data will be generated. In order to find the Churn rate of customers & draw insights like what are the reasons for the customer to leave the services of one particular company to other & few other insights like how to be profitable to the company. Machine Learning plays a vital role in this process. We are going to dive deep to churn problems with scalable approach for machine learning on big data.

So, to build & evaluate ML models for Big Data, we can implement machine learning in Spark using its MLlib library.

- Initially considered sample data and build Machine Learning models & evaluated them.
- Next performed some preprocessing that fits the complete data to implement ML models in PySpark
- Then built ML models using PySpark & evaluate those models.

4.2.1 Model Building without PySpark

1. Logistic Regression
2. Decision tree classifier
3. Random Forest classifier
4. Gaussian Naïve Bayes
5. SVC
6. SVC using RBF Kernel

Initially, we defined a function which fits the algorithm to build the model & generate classification report, Accuracy Score & Model Performance plot which consists of Confusion matrix, ROC curve, Feature Importance plot & Threshold for each model build with a particular algorithm.

As the core problem of our project is classify whether the customer is churn customer or not, we have built various "Classification models" & generated classification report for each model. So let's understand the classification models which we have used & why we have used.

Machine Learning Models:

a. **Logistic Regression:** Logistic regression is a classification algorithm, used when the value of the target variable is categorical in nature. Logistic regression is most commonly used when the data in question has binary output, so when it belongs to one class or another, or is either a 0 or 1. So the classification of customer to one of the classes Churn customer (class=1) or non-Churn customer (class=0).

b. **Decision Tree Classifier:** Decision trees are a powerful and popular tool. They're commonly used by data analysts to carry out predictive analysis. They're also a popular tool for machine learning and artificial intelligence, where they're used as training algorithms for supervised learning i.e., categorizing data based on different tests, such as 'yes' or 'no' classifiers.

c. **Random Forest Classifier:** Random Forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems.

d. **Gaussian Naïve Bayes Classifier:** Naive Bayes can be extended to realvalued attributes, most commonly by assuming a Gaussian distribution. This extension of Naive Bayes is called Gaussian Naive Bayes.

e. **SVC- SVM - Linear Kernel:** SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

Linear SVM is used for linearly separable data, since our dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

f. **SVC – RBF (Radial Basis Function) Kernel:** RBF is the default kernel used within the SVM classifier. Since we're working on a Machine Learning algorithm like Support Vector Machines for non-linear dataset and we can't seem to figure out the right feature transform or the right kernel to use. In such cases Radial Basis Function (RBF) Kernel is our savior.
RBF Kernel is popular because of its similarity to K-Nearest Neighborhood Algorithm. It has the advantages of K-NN and overcomes the space complexity problem as RBF Kernel Support Vector Machines just needs to store the support vectors during training and not the entire dataset.

- **Classification Reports for the build Machine Learning Models:**

A classification report is a performance evaluation metric in machine learning. It is used to show the precision, recall, F1 Score, and support of your trained classification model.

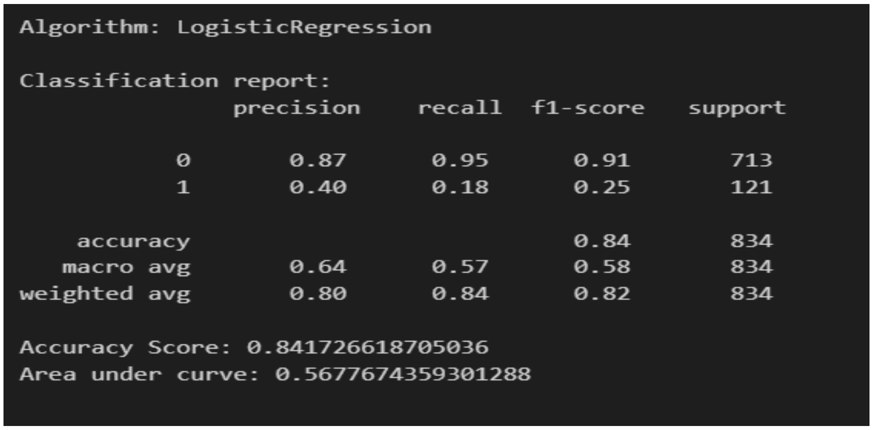
To understand the classification report of a machine learning model, you need to know all of the metrics displayed in the report.

Table No.1 Definitions of Metrics

Metrics	Definition
Precision	Precision is defined as the ratio of true positive to the sum of true & false positives
Recall	Recall is defined as the ration of true positives to the sum of true positives & false negatives
F1 Score	The F1 is weighted harmonic mean of precision & recall. The closer the value of F1 score is to 1.0, the better the expected performance of the model is.
Support	Support is the number of actual occurrences of the class in the dataset. It doesn't vary

Classification Reports for all the built machine learning models:

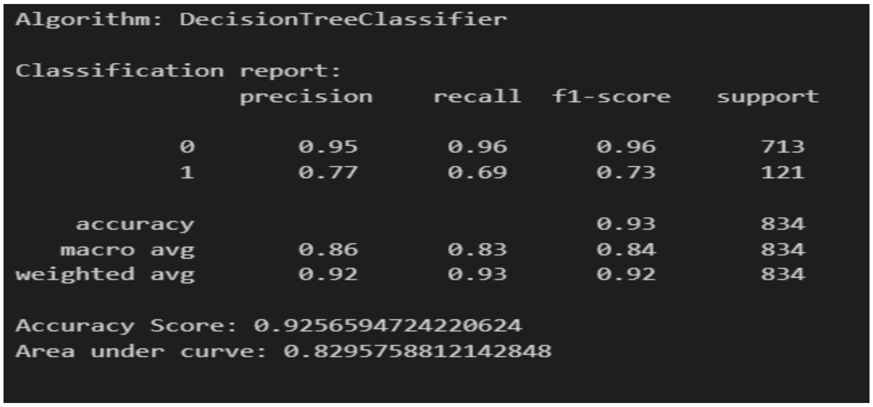
Algorithm: Logistic Regression



```
Algorithm: LogisticRegression

Classification report:
               precision    recall  f1-score   support

           0       0.87      0.95      0.91       713
           1       0.40      0.18      0.25       121

    accuracy                           0.84       834
   macro avg       0.64      0.57      0.58       834
weighted avg       0.80      0.84      0.82       834

Accuracy Score: 0.841726618705036
Area under curve: 0.5677674359301288
```

Fig.no.3 Classification Report for Logistic Regression Model

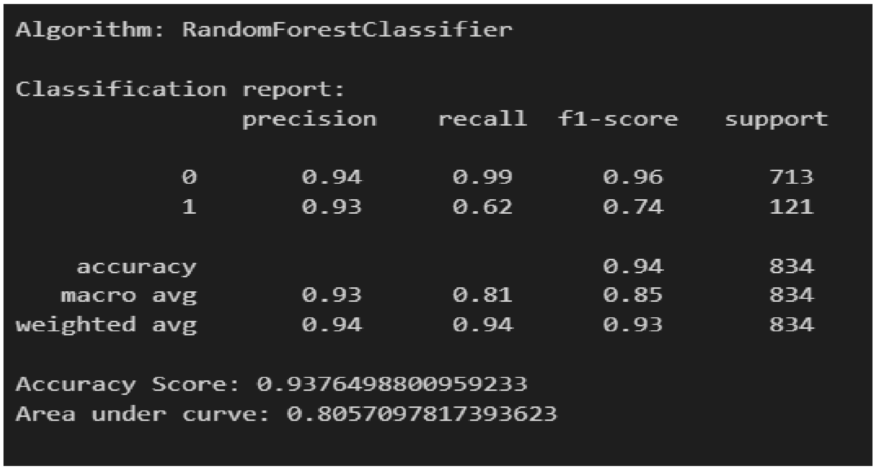
Algorithm: Decision Tree Classifier



```
Algorithm: DecisionTreeClassifier

Classification report:
               precision    recall  f1-score   support

           0       0.95      0.96      0.96       713
           1       0.77      0.69      0.73       121

    accuracy                           0.93       834
   macro avg       0.86      0.83      0.84       834
weighted avg       0.92      0.93      0.92       834

Accuracy Score: 0.9256594724220624
Area under curve: 0.8295758812142848
```

Fig.no.4 Classification Report for Decision Tree Classifier Model

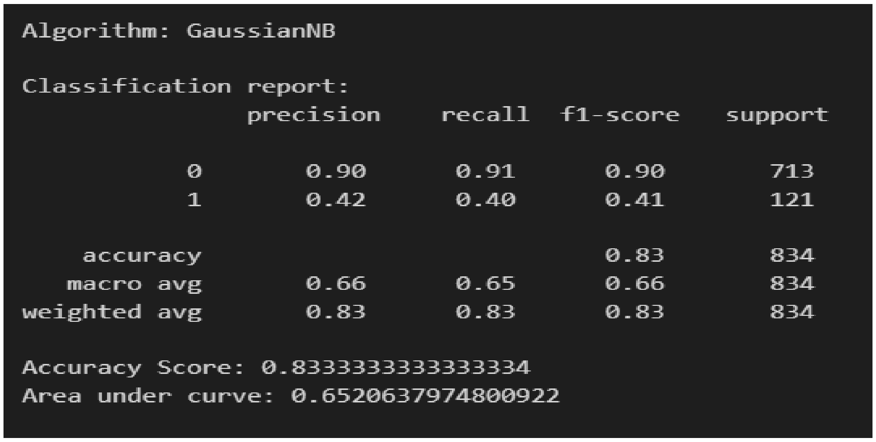
Algorithm: Random Forest Classifier



```
Algorithm: RandomForestClassifier

Classification report:
               precision    recall  f1-score   support

           0       0.94      0.99      0.96       713
           1       0.93      0.62      0.74       121

    accuracy                           0.94       834
   macro avg       0.93      0.81      0.85       834
weighted avg       0.94      0.94      0.93       834

Accuracy Score: 0.9376498800959233
Area under curve: 0.8057097817393623
```

Fig.no.5 Classification Report for Random Forest Classifier Model

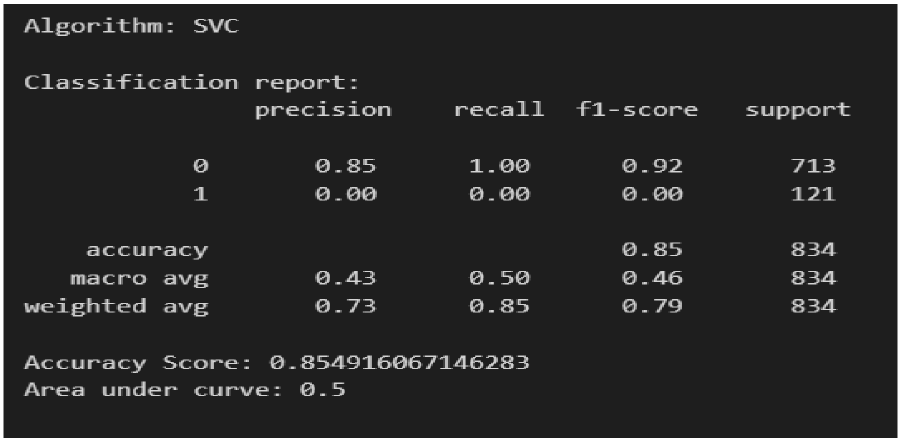
Algorithm: Gaussian NB



```
Algorithm: GaussianNB

Classification report:
               precision    recall  f1-score   support

           0       0.90      0.91      0.90       713
           1       0.42      0.40      0.41       121

    accuracy                           0.83       834
   macro avg       0.66      0.65      0.66       834
weighted avg       0.83      0.83      0.83       834

Accuracy Score: 0.8333333333333334
Area under curve: 0.6520637974800922
```

Fig.no.6 Classification Report for Gaussian NB Model

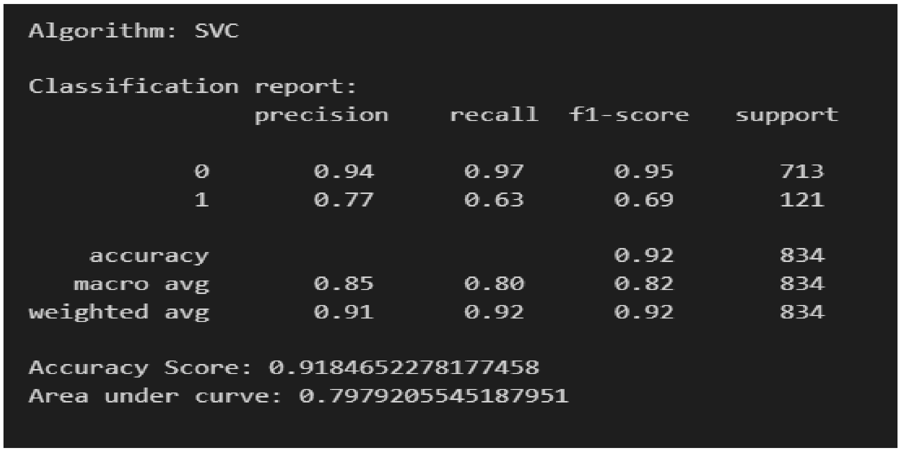
Algorithm: SVC



```
Algorithm: SVC

Classification report:
               precision    recall  f1-score   support

           0       0.85      1.00      0.92       713
           1       0.00      0.00      0.00       121

    accuracy                           0.85       834
   macro avg       0.43      0.50      0.46       834
weighted avg       0.73      0.85      0.79       834

Accuracy Score: 0.854916067146283
Area under curve: 0.5
```

Fig.no.7 Classification Report for SVC Model

Algorithm: SVC-RBF Kernel



```
Algorithm: SVC

Classification report:
               precision    recall  f1-score   support

           0       0.94      0.97      0.95       713
           1       0.77      0.63      0.69       121

    accuracy                           0.92       834
   macro avg       0.85      0.80      0.82       834
weighted avg       0.91      0.92      0.92       834

Accuracy Score: 0.9184652278177458
Area under curve: 0.7979205545187951
```

Fig.no.8 Classification Report for SVC-RBF Model

- **Model Performance Report over Training Dataset**

Table no.2 Performance Report of ML models over Training Dataset

Models	Accuracy	Recall	Precision	F1 - Score	ROC_AUC	Kappa metric
Logistic Regression	0.8417	0.1818	0.4	0.25	0.5678	0.1752
Decision Tree	0.9257	0.6942	0.7706	0.7304	0.8296	0.6875
Random Forest	0.9376	0.6198	0.9259	0.7426	0.8057	0.7087
Naïve Bayes	0.8333	0.3967	0.4211	0.4085	0.6521	0.3116
SVM (linear)	0.8449	0.0	0.0	0.0	0.5	0.0
SVM (RBF)	0.9185	0.6281	0.7677	0.6909	0.7979	0.6445

ROC_auc: "Area Under the Curve" (AUC) of "Receiver Characteristic Operator" (ROC). AUC-ROC curve helps us visualize how well our machine learning classifier is performing. Although it works for only binary classification problems, we can even extend it to evaluate multi-class classification problems too.

The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems. It is a probability curve that plots the TPR against FPR at various threshold values and essentially separates the 'signal' from the 'noise'. 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 positive and negative classes.

Kappa Metric: The Kappa Statistic or Cohen's* Kappa is a statistical measure of inter-rater reliability for categorical variables. "A Kappa Value of .70 Indicates good reliability". The kappa coefficient measures the agreement between classification and truth values. A kappa value of 1 represents perfect agreement, while a value of 0 represents no agreement.

- **Manipulating the Dataset:** While we are in the process of manipulating the datasets, let's transform the categorical data into numeric as required by the machine learning routines, using a simple user-defined function that maps Yes/True and No/False to 1 and 0, respectively. All these tasks will be done using the following get data function.

Using RDD (map ()) transformation PySpark map (map ())to apply the transformation function (lambda) on every element of RDD/DataFrame and returns a new RDD & then built Decision tree for data frame of 2 rows & 7n nodes.

Result of Decision Tree Classifier after transforming the data:

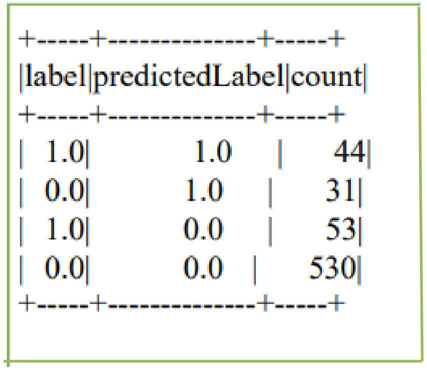
```
The two first rows of the training data RDD:
[LabeledPoint(0.0, [117.0,0.0,0.0,0.0,184.5,97.0,351.6,80.0,215.8,90.0,8.7,4.0,
1.0]), LabeledPoint(1.0, [161.0,0.0,0.0,0.0,332.9,67.0,317.8,97.0,160.6,128.0,5
.4,9.0,4.0])]
============================
DecisionTreeModel classifier of depth 2 with 7 nodes
  If (feature 4 <= 264.79999999999995)
   If (feature 12 <= 3.5)
    Predict: 0.0
   Else (feature 12 > 3.5)
    Predict: 1.0
  Else (feature 4 > 264.79999999999995)
   If (feature 3 <= 2.0)
    Predict: 1.0
   Else (feature 3 > 2.0)
    Predict: 0.0
```

- **<u>Model Evaluation</u>**

Result for Decision Tree Model Evaluation:

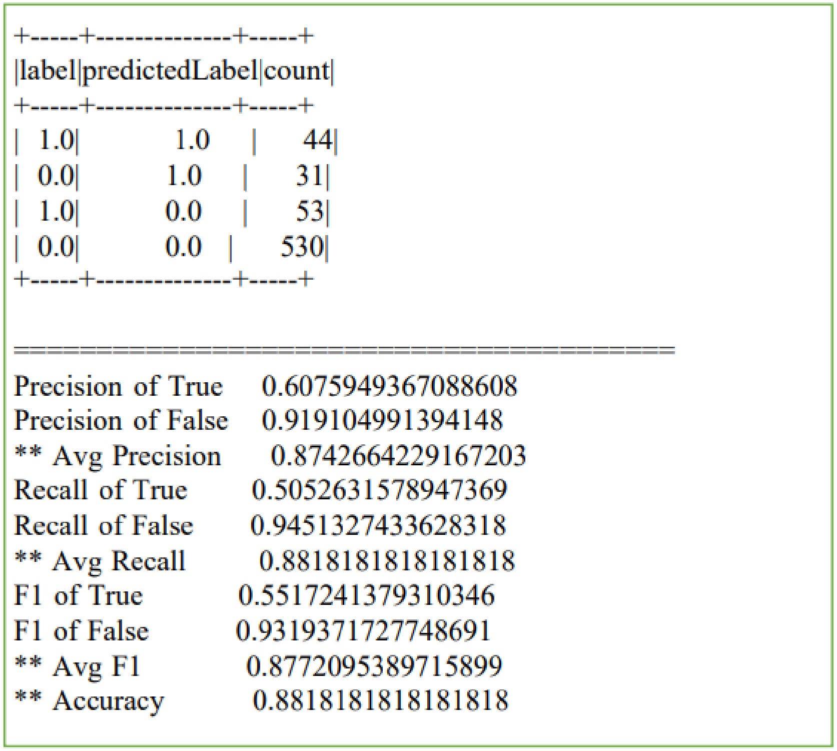
```
Confusion Matrix
[[530. 29.]
[ 55. 44.]]

Precision of True        0.6027397260273972
Precision of False       0.905982905982906
Weighted Precision       0.8603581722206031
Recall of True          0.4444444444444444
Recall of False         0.9481216457960644
Weighted Recall          0.8723404255319148
FMeasure of True         0.5116279069767442
FMeasure of False        0.9265734265734266
Weighted fMeasure        0.8641424137465702
Accuracy               0.8723404255319149
```

To check the above obtained results and comparing them with those that will be obtained using our new “printAllMetrics” function, we have displayed the confusion matrix that is used to compute all the variables of our new function:



```
+-----+--------------+-----+
|label|predictedLabel|count|
+-----+--------------+-----+
|  1.0|          1.0    |    44|
|  0.0|         1.0    |    31|
|  1.0|         0.0    |    53|
|  0.0|         0.0   |    530|
+-----+--------------+-----+
```

We have built a new model using the evenly distributed data set and see how it performs. The result displaying the performance of this model.



```
+-----+--------------+-----+
|label|predictedLabel|count|
+-----+--------------+-----+
|  1.0|          1.0    |    44|
|  0.0|         1.0    |    31|
|  1.0|         0.0    |    53|
|  0.0|         0.0   |    530|
+-----+--------------+-----+


========================================
Precision of True    0.6075949367088608
Precision of False   0.919104991394148
** Avg Precision     0.8742664229167203
Recall of True       0.5052631578947369
Recall of False      0.9451327433628318
** Avg Recall        0.8818181818181818
F1 of True           0.5517241379310346
F1 of False          0.9319371727748691
** Avg F1            0.8772095389715899
** Accuracy          0.8818181818181818
```

With these new recall values, we can see that the stratified data was helpful in building a less biased model, which will ultimately provide more generalized and robust predictions.

4.2.2 Model Building with PySpark

Let’s define a “get_dummy” function that transforms a given classical data frame to a new other one composed of dense vectors reliable to be running with Spark ML.

Once the required function is ready, let’s define the needed numericCols list by removing the Churn column and transforming the datasets:

We can see that the test dataset contains 667 samples.

➢ Model Evaluation Result for K-fold Cross Validation

Table no.3 Model Evaluation Results

Classifier name	UnderROC train	UnderROC test	Accuracy train	Accuracy test	F1 train	Wprecision train	Wprecision test	Wrecall train	Wrecall test
LR	0.594	0.579	0.863	0.858	0.863	0.835	0.824	0.863	0.885
NB	0.608	0.622	0.629	0.622	0.629	0.800	0.809	0.629	0.622
SVC	0.500	0.500	0.855	0.858	0.855	0.731	0.735	0.855	0.858
DT	0.858	0.881	0.950	0.961	0.950	0.949	0.961	0.950	0.961
RF	0.806	0.799	0.941	0.942	0.941	0.943	0.944	0.941	0.942

Metrics computed using stratified data (stratified CV data) for the training step:

Table.no.4 Model Evaluation using Stratified data

Classifier name	UnderROC train	UnderROC test	Accuracy train	Accuracy test	F1 train	Wprecision train	Wprecision test	Wrecall train	Wrecall test
LR	0.763	0.774	0.763	0.778	0.763	0.763	0.870	0.763	0.778
NB	0.602	0.600	0.602	0.585	0.602	0.602	0.801	0.602	0.585
SVC	0.741	0.759	0.741	0.760	0.741	0.741	0.864	0.741	0.760
DT	0.893	0.887	0.894	0.904	0.894	0.899	0.925	0.894	0.904
RF	0.885	0.874	0.885	0.867	0.885	0.886	0.913	0.885	0.867

➢ Model Evaluation Result for each Model built with PySpark

Naïve Bayes Classifier:

```
Weighted Precision 0.6024474355396268
Weighted Recall 0.6020304568527919
F1 0.6020378401496147
Accuracy 0.6020304568527919
============================
Precision of True 0.590818363273453
Precision of False 0.6136363636363636
** Avg Precision 0.6024474355396268
Recall of True 0.6128364389233955
Recall of False 0.5916334661354582
** Avg Recall 0.6020304568527919
F1 of True 0.6016260162601627
F1 of False 0.6024340770791075
** Avg F1 0.6020378401496148
** Accuracy 0.6020304568527919
```

Decision Tree Classifier:

```
Weighted Precision 0.8986643743022468
Weighted Recall 0.8944162436548224
F1 0.8940123670588299
Accuracy 0.8944162436548223
=========================
Precision of True 0.9396751740139211
Precision of False 0.8592057761732852
**Avg Precision 0.8986643743022469
Recall of True 0.8385093167701864
Recall of False 0.9482071713147411
**Avg Recall 0.8944162436548223
F1 ofTrue 0.886214442013129
F1 of False 0.9015151515151516
**Avg F1 0.8940123670588299
**Accuracy 0.8944162436548223
```

Logistic Regression Classifier:

```
Weighted Precision 0.7634563788643671
Weighted Recall 0.7634517766497462
F1 0.7633902874489755
Accuracy 0.7634517766497462
==========================
Precision of True 0.7637130801687764
Precision of False 0.7632093933463796
** Avg Precision 0.763456378864367
Recall of True 0.7494824016563147
Recall of False 0.7768924302788844
** Avg Recall 0.7634517766497462
F1 of True 0.7565308254963428
F1 of False 0.769990128331688
** Avg F1 0.7633902874489755
** Accuracy 0.7634517766497462
```

Linear SVC

```
Weighted Precision 0.7411923065395787
Weighted Recall 0.7411167512690355
F1 0.7411354330068094
Accuracy 0.7411167512690355
=============================
Precision of True 0.7336065573770492
Precision of False 0.7484909456740443
** Avg Precision 0.7411923065395787
Recall of True 0.7412008281573499
Recall of False 0.7410358565737052
** Avg Recall 0.7411167512690355
F1 of True 0.7373841400617919
F1 of False 0.7447447447447447
** Avg F1 0.7411354330068094
** Accuracy 0.7411167512690355
```

Random Forest Classifier

```
Weighted Precision 0.8857688739323354
Weighted Recall 0.8852791878172589
F1 0.8851936902420627
Accuracy 0.8852791878172589
==================================
Precision of True 0.8987068965517241
Precision of False 0.8733205374280231
** Avg Precision 0.8857688739323354
Recall of True 0.8633540372670807
Recall of False 0.9063745019920318
** Avg Recall 0.8852791878172589
F1 of True 0.8806758183738119
F1 of False 0.8895405669599217
** Avg F1 0.8851936902420628
** Accuracy 0.8852791878172589
```

The Result we have obtained for the best fit model is the count of Churn customers & count of non-Churn customers. Prediction Using Decision tree model for the testing dataset which we have given as the input.

```
+-----+-----+
|Churn|count|
+-----+-----+
|  0.0| 2850|
|  1.0|  483|
+-----+-----+
```

5. FINDINGS

We have found that among the entire machine learning predictive models built without PySpark, Decision Tree model and Random Forest model are highly accurate with accuracy score 0.925 and 0.937 respectively, with area under curve 0.829 and 0.805 respectively .

Stratified data has reduced the biasness in dataset. We have found that Decision Tree model is highly accurate among entire machine learning models built using PySpark.

The result obtained by the best fit prediction model i.e., Decision Tree model is that there are 2850 non churn customers and 483 churn customers.

6. LIMITATIONS

In this study we developed prediction models for the predefined dataset. There is no standard model which continuously learns the customer behavior and accurately predicts the customer churn.

7. SUGGESTIONS

In this study, we have predicted churn for Telecommunication Company. We can suggest that churn prediction model can be build for various sectors like banking, e-commerce and other businesses which are data driven.

Live prediction model can be build so that it would give accurate results, which will help in better customer retention. Retaining its customers is the greatest success for any business.

8. CONCLUSION

In this project, we have walked through a complete end-to-end machine learning project using the Telco customer Churn dataset. We started by cleaning the data and analyzing it with visualization. Then, to be able to build a machine learning model, we transformed the categorical data into numeric variables (feature engineering). After transforming the data, we tried 6 different machine learning algorithms using default parameters.

The Decision tree, Random Forest and Gradient-boosted tree are the more efficient classifiers. However, the best model created according to the cross validation process seems to be the Decision tree. Indeed, the different metrics obtained by using this classifier are higher as well as very close to each other comparing to those of the other classifiers.

Thus, we can conclude that the big data analytics with machine learning techniques have proven to be accurate and effective to predict customer churn in nearby future.

9. REFERENCES

A Amin, A Adnan, S Anwar (2023) An adaptive learning approach for customer churn prediction in the telecommunication industry using evolutionary computation and Naïve Bayes- Applied Soft Computing

Ahmad A.M., Jafar A, Aljoumaa K (2019). Customer churn prediction in telecom using machine learning in big data platform. Journal of Big Data, 6(28), pp. 1-24

D AL-Najjar, N Al-Rousan, H AL Najjar (2022). Machine learning to develop credit card customer churn prediction - Journal of Theoretical and Applied Electronic Commerce Research

M Rahman, V Kumar (2020). Machine learning based customer churn prediction in banking, 4th international conference, IEEE

NNY Vo, S Liu, X Li, G Xu (2021). Leveraging unstructured call log data for customer churn prediction- Knowledge-Based Systems

T Zhang, S Moro, RF Ramos (2022). A data-driven approach to improve customer churn prediction based on telecom customer segmentation- Future Internet

X Xiahou, Y Harada (2022). B2C E-commerce customer churn prediction based on K-means and SVM - Journal of Theoretical and Applied Electronic Commerce Research