TELECOM CHURN PREDICTION

A Project Report

Submitted by

BISHNU PRASAD JENA	**2101020310**
YADAVSRI CHINMAYA BEHERA	**2101020272**
PURNENDU MOHANTA	**2101020373**

In partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
C.V. RAMAN GLOBAL UNIVERSITY
BHUBNESWAR- ODISHA -752054

NOV 2024

C.V. RAMAN GLOBAL UNIVERSITY
BHUBANESWAR-ODISHA-752054

BONAFIDE CERTIFICATE

Certified that this project report "Telecom Churn Prediction" is the bonafide work submitted by **Bishnu Prasad Jena, 2101020310, Yadavsri Chinmaya Behera, 2101020272, and Purnendu Mohanta, 2101020373** from CGU-Odisha, Bhubaneswar, who carried out the project under my supervision.

Dr. Monalisa Mishra
HEAD OF THE DEPARTMENT
Department of Computer Science & Engineering

Dr. Santoshinee Mohapatra
SUPERVISOR
Assistant Professor, Department of Computer Science & Engineering

C.V. RAMAN GLOBAL UNIVERSITY
BHUBANESWAR-ODISHA-752054

CERTIFICATE OF APPROVAL

This is to certify that we have examined the project entitled **"Telecom Churn prediction"** is bonafide work submitted by **Bishnu Prasad Jena 7th Semester (Registration No. 2101020310), Yadavsri Chinmaya Behera 7th Semester (Registration No. 2101020272), and Purnendu Mohanta 7th Semester (Registration No. 2101020373)2024,** CGU-Odisha, Bhubaneswar.

We hereby accord our approval of it as a major project work carried out and presented in a manner required for its acceptance for the partial fulfillment for **Bachelor Degree of Computer Science and Engineering** for which it has been submitted. This approval does not necessarily endorse or accept every statement made,opinion expressed, or conclusions drawn as recorded in this major project, it only signifies the acceptance of the major project for the purpose it has been submitted.

Dr. Santoshinee Mohapatra
SUPERVISOR
Assistant Professor, Department of
Computer Science & Engineering

DECLARATION

We declare that this project report titled "**Telecom Churn Prediction**" submittedin partial fulfillment of the degree of **B. Tech in (Computer Science and Engineering)** is a record of original work carried out by us under the supervision of **Dr. Santoshinee Mohapatra**, and has not formed the basis for the award of anyother degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

Bishnu prasad Jena (2101020310)
Yadavsri Chinmaya Behera (2101020272)
Purnendu Mohanta (2101020373)

Bhubaneswar - 752054
22/11/2024

ACKNOWLEDGMENTS

We express our heartfelt gratitude to all who contributed to the success of our project on **Telecom Churn Prediction**.

To begin with, we would like to convey our heartfelt gratitude to Dr. Monalisa Mishra, Head of the Department, Computer Science & Engineering, for her directions, encouragement and throughout assistance during this project.

We wish to express our profound gratitude to Dr. Santoshinee Mohapatra, Assistant Professor, Department of Computer Science & Engineering for supporting this project with constructive comments and inputs at all levels of the project.

We appreciate the faculty members and the administration of CV Raman Global University for their support and for creating a conducive research atmosphere that helped us remain committed and focused all through. We equally recognize our colleagues and fellow students who provided suggestions, critiques and comments that enabled us to expand our horizons in addressing intricate intricacies of issues like medical image forgery detection.

Last but not least, we thank our relatives, parents and friends for the many words of encouragement and support they gave us which enabled us to go through trying times. Thank you.

Bishnu prasad Jena

Yadavsri Chinmaya Behera

Purnendu Mohanta

ABSTRACT

Customer churn, or customer attrition, is a significant concern for businesses, particularly in competitive industries. It represents necessitates high customer acquisition costs. Application of machine learning techniques to predict customer churn, enabling companies to proactively retain customers. The high cost of customer churn and the importance of customer retention. Leveraging machine learning models to analyze customer data and identify patterns that predict churn. Proactive customer retention strategies through early identification of at-risk customers.

These may include imbalanced datasets (where churned customers area smaller proportion), non-linear relationships between factors and churn, and the vast amount of data often generated in industries like telecommunications. By effectively predicting churn, companies can implement targeted interventions to satisfy at-risk customers and bolster customer loyalty.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Telecom Churn

Customer churn prediction is a business strategy that relies on data analysis to identify customers at high risk of leaving a service or cancelling a subscription. Companies usechurn prediction to proactively retain these customers, which can lead to significant cost savings and increased revenue [3]. It is generally more expensive to acquire new customers than to retain existing ones. By predicting churn, businesses can focus theirefforts on keeping valuable customers satisfied.

Churn prediction model analysis historical customer data to find patterns that indicatea likelihood of churn. This data can include demographics, purchase history, support interactions, and even how customers interact with the company's website or app. Bypinpointing at-risk customers, businesses can implement targeted interventions, such as offering discounts, providing additional support, or addressing specific customer issues.

1.2 Customer Churn

Churning of Customers, also known as shifting from one service to another service due to unsatisfactory reasons over a specific timeframe. It essentially means customerwho stop doing business with you. This can happen for various reasons, and understanding those reasons is crucial to reducing churn [1].

The customer churn rate is a key metric. It is calculated as the percentage of customerswho churn within a given period (usually a month or a year). High churn rates can significantly impact business growth. It costs more to acquire new customers than to retain existing ones. Additionally, loyal customers tend to spend more and often become brand advocates [6].

Active churn occurs when customers actively cancel or stop using your service, whichcan result from factors such as a poor product experience, lack of perceived value, or finding a better competitor. Passive churn happens due to circumstances beyond your control, such as expired credit cards or business closures.

Chapter 2

LITERTURE SURVEY

Telecom churn prediction explores various methodologies and advancements in predicting customer attrition, emphasizing its importance for improving customer retention and reducing acquisition costs. Early approaches relied on statistical models like logistic regression and decision trees, but the adoption of machine learning algorithms such as Random Forest, Gradient Boosting, and Support Vector Machines(SVM) has significantly improved prediction accuracy [2].

Deep learning techniques, including Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM), have further advanced churn prediction by effectively analyzing time-series and complex customer data. Key features influencing churn include demographics, usage patterns, billing history, and complaints, with feature selection and engineering enhancing model performance [5].

Evaluation metrics such as precision, recall, F1-score, and AUC-ROC address the imbalance between churn and non-churn cases, ensuring robust model evaluation. Additionally, customer retention strategies like loyalty rewards, personalized offers, and enhanced service quality are guided by predictive insights. However, challenges such as data scarcity, scalability, and privacy concerns persist, presenting opportunities for further research and innovation in the field

Chapter 3

DATA PREPARTION

Data preparation is a crucial step in churn prediction to ensure that the data isclean, relevant, and appropriately formatted for analysis. Figure 3.1 outlines the typical tasks involved in building a machine learning (ML)model.

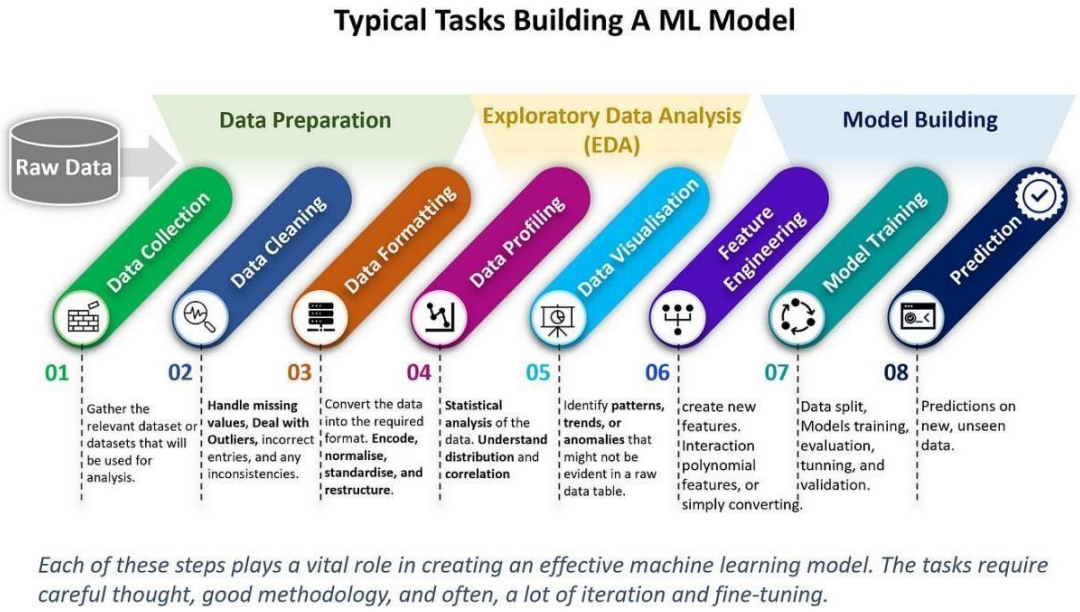


Figure 3.1 ML model, from data collection to prediction.

3.1 Data Cleaning and preprocessing

3.1.1 Data Collection

Gather all relevant data sources containing information about customer interactions, demographics, usage patterns, and more. This may include transaction logs, customerdatabases, survey results, and other relevant sources.

3.1.2 Data Cleaning

Clean the data to eliminate inconsistencies, errors, or missing values that could affectthe accuracy of the churn prediction model. Techniques such as imputation for missing values, outlier detection, and correcting data inconsistencies are commonly applied

3.1.3 Data Transformation

Transform the data into a format suitable for analysis. This process may involve encoding categorical variables, scaling numerical variables, and normalizing data toensure all features are on a comparable scale.

3.1.4 Balancing the Dataset

Churn prediction datasets frequently experience class imbalance, where the numberof churn instances is much smaller than the number of non-churn instances. To mitigate this issue, techniques such as oversampling (e.g., generating synthetic datapoints for the minority class) or under sampling (e.g., removing some data points from the majority class) can be applied to balance the dataset.

3.1.5 Feature Scaling

Scale the features to ensure that it’s on similar scale. It includes min scaling and maxscaling or standardization.

3.1.6 Feature Encoding

Encode categorical variables into a numerical format suitable for modeling. Techniques such as label encoding or one-hot encoding are commonly used.

3.1.7 Dimensionality Reduction

If the dataset contains many features, use dimensionality reduction techniques likePCA to reduce the number of features while retaining the most important information.

3.2 Feature Engineering

Create new features from existing ones to improve the model's predictive performance. This may involve combining existing features, generating interactionterms, or deriving new features based on domain expertise. The below Figure3.2 shows the Feature Engineering.

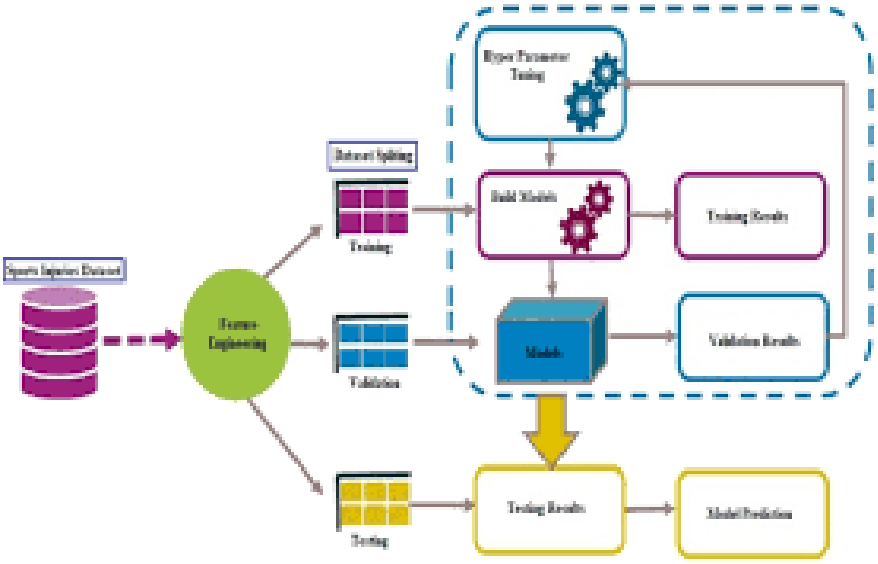


Figure 3.2 Feature Engineering

3.3 Pipeline for predictive modeling

Preparing the Dataset:

This dataset has 7043 records of attribute extracted from Telecom Customer data, it is classified into 2 classes:

- Churner
- Non-Churner

For the execution first we have to find the data. I have to collect the data form online source (kaggle). The data are in CSV format, using data science pipeline method I have to convert the csv file to simplified and understandable data. The below Figure 3.3 Shows the Pipeline in Data Science in simplified way.

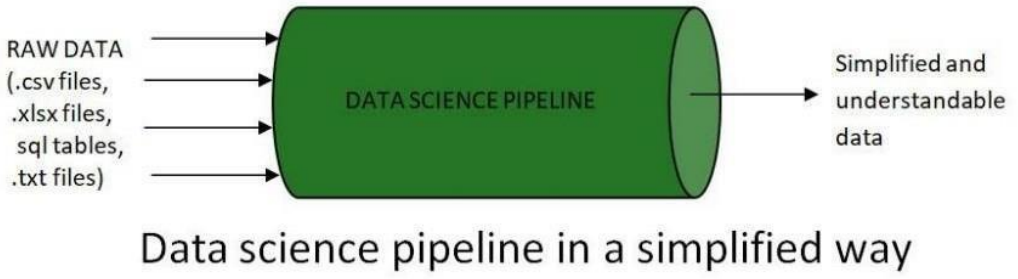


Figure 3.3 PIPELINING

importing some python libraries:

Pandas: Pandas is used for manipulation and preprocessing tasks.

Numpy: (Numerical Python) NumPy is essential for numerical computations and array operations

Seaborn: it's used for creating better visualization

Matplotlib: It is used for creating static, interactive, and animated visualizations in various formats.

Plotly: It is makes a interactive,publication-quality graphs online

Scikit-learn: It provides tools for preprocessing data, feature selection,model selection, and performance evaluation.

Imblearn: Provides various techniques for handling imbalanced datasets

TensorFlow with cuda: It is widely used libraries for building and deploying machine learning and deep learning models.

3.4 Problem Statement

This project aims to use the available information to build a machine learning model that will accurately predict which consumers will likely quit a company, enabling the business owner to make better-informed decisions.

To import a data set into python enviroment , the common library for handling data set is pandas.

```
**Load the data file **
[2]: telco_base_data = pd.read_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')
```

Now 'telco_base_data' is a Data Frame containing the dataset. The Figure 3.4 shows the entire dataset

[2]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	No
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	No
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	Yes
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	No
...	...	...	...	...	...	...	...	...	...	...	...	...	...
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	...	Yes	Yes
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	...	Yes	No
7040	4801-JZAZL	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	...	No	No
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No	...	No	No
7042	3186-AJIEK	Male	0	No	No	66	Yes	No	Fiber optic	Yes	...	Yes	Yes

7043 rows × 21 columns

Figure 3.4 Dataset

Here the dataset contains 7043 rows and 21 columns(Variable).We will checkit as (**telco_base_data.shape**)

There is no need of Customer_id in the code so we have to eraseit from thedataset.

telco_data=telco_base_data.drop(["customerID"],axis = 1)

After that we have to check what are the variables present in thedataset **telco_data.info()**

About the Variable

- **Customer-ID**: A unique identifier assigned to each customer.
- **Gender**: Represents the gender, either male or female.
- **SeniorCitizen**: Indicates if the customer is a senior citizen.
- **Partner**: Specifies whether the customer has a partner.
- **Dependents**: Shows whether the customer has any dependents.
- **Tenure**: The length of time (in months) the customer has been with the business.
- **Phone Service**: Indicates if the customer subscribes to phone service.
- **Multiple Lines**: Specifies if the customer has more than one phone line.
- **Internet Service**: The customers chosen internet service provider.
- **Online Security**: The customer has subscribed to online security.
- **Online Backup**: Indicates if the customer has an online backup service.
- **Device Protection**: Specifies whether the customer has protection for theirdevices.
- **Tech Support**: Indicates if the customer receives technical support.
- **Streaming TV**: Shows whether the customer subscribes to streaming TVservices.

- **Streaming Movies**: Specifies whether the customer subscribes to streamingmovie services.

- **Contract**: The type of contract the customer holds with the business.

- **Paperless Billing**: Indicates if the customer opts for paperless billing.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 20 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   gender            7043 non-null   object
 1   SeniorCitizen     7043 non-null   int64
 2   Partner           7043 non-null   object
 3   Dependents        7043 non-null   object
 4   tenure            7043 non-null   int64
 5   PhoneService      7043 non-null   object
 6   MultipleLines     7043 non-null   object
 7   InternetService   7043 non-null   object
 8   OnlineSecurity    7043 non-null   object
 9   OnlineBackup      7043 non-null   object
 10  DeviceProtection  7043 non-null   object
 11  TechSupport       7043 non-null   object
 12  StreamingTV       7043 non-null   object
 13  StreamingMovies   7043 non-null   object
 14  Contract          7043 non-null   object
 15  PaperlessBilling  7043 non-null   object
 16  PaymentMethod     7043 non-null   object
 17  MonthlyCharges    7043 non-null   float64
 18  TotalCharges      7043 non-null   object
 19  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(17)
memory usage: 1.1+ MB
```

Figure 3.4.1 Column with data type

In the above Figure 3.4.1 there are 20 columns with the data type of **int,float,object**. But in case of Total Charge it will show object**.** i.e Total charges contains some missing value, here we have to manage it

with Missingvalue Imputation

In the below in figure 3.4.2 it's shows the graph of missing value.

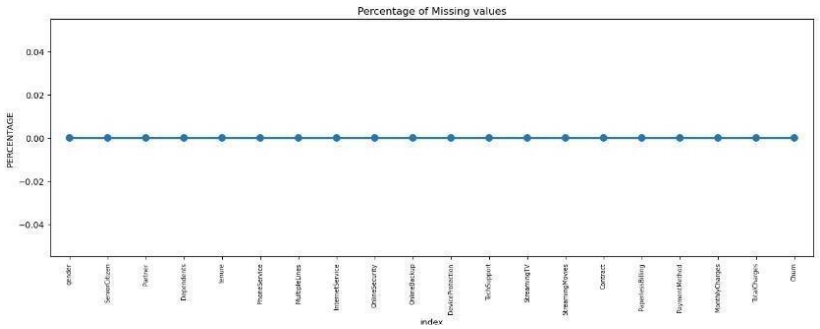


Figure 3.4.2 Graph of Missing Value

For checking null value in Total Charges

df.TotalCharges=pd.to_numeric(df.TotalCharges,errors='coerce')df.isnull ().sum() TotalCharges:11 (Thereare11nullvalues)

telco_data.loc[telco_data['TotalCharges'].isnull()==True]

The Figure3.4.3 displays a table with **customer information** for a telecom service.

[10]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService
488	Female	0	Yes	Yes	0	No	No phone service	DSL
753	Male	0	No	Yes	0	Yes	No	No
936	Female	0	Yes	Yes	0	Yes	No	DSL
1082	Male	0	Yes	Yes	0	Yes	Yes	No
1340	Female	0	Yes	Yes	0	No	No phone service	DSL
3331	Male	0	Yes	Yes	0	Yes	No	No
3826	Male	0	Yes	Yes	0	Yes	Yes	No
4380	Female	0	Yes	Yes	0	Yes	No	No
5218	Male	0	Yes	Yes	0	Yes	No	No
6670	Female	0	Yes	Yes	0	Yes	Yes	DSL
6754	Male	0	No	Yes	0	Yes	Yes	DSL

Figure 3.4.3 Null Value in Total Charges

telco_data.dropna(how= 'any',inplace=True)

By using this code we have to delete that 11 rows from the dataset. Then we will get the perfect dataset with (7032,20) values.

Then we will check the Churn Rate as per the dataset.

telco_data.churn.value_counts()

Yes: 1869 No:5163

The Figure 3.4.4 shows first few rows of the dataset with customer information of a telecomcompany.

```
[3]: telco_base_data.head()
```

[3]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	No
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	No
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	Yes
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	No

Figure 3.4.4 First five rows

We will check the tenure values **telco_data.tenure.value_counts()**

This figure3.4.5 depicts about the tenure value of the churned customers

```
[16]: tenure
1     613
72    362
2     238
3     200
4     176
     ...
38     59
28     57
39     56
44     51
36     50
Name: count, Length: 72, dtype: int64
```

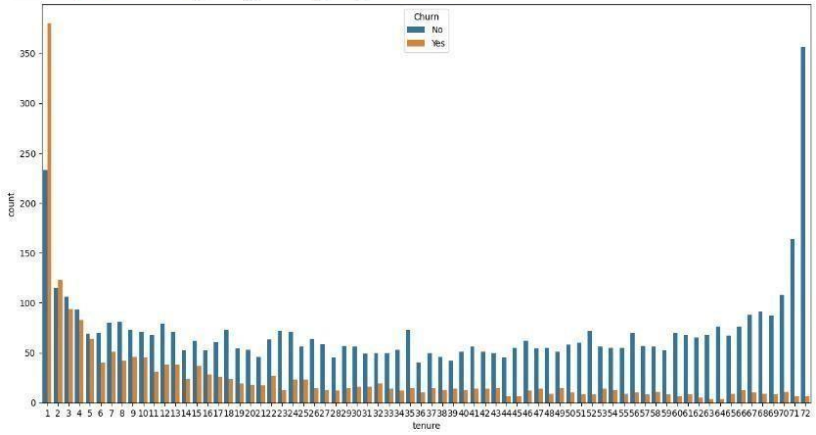


Figure 3.4.5Tenure Value

The count plot shows that “subscribe plan ” with less than1 year are churned more. For betterUnderstand we will divide it into groups of 12 Months. This Figure3.4.6 shows a Python script where customer tenure is grouped into intervals and theircounts are compute. The figure 3.4.6 it’s indicate the Tenure Group

```
[19]: labels = ["{0} - {1}".format(i, i + 11) for i in range(1, 72, 12)]

      df2['tenure_group'] = pd.cut(df.tenure, range(1, 80, 12), right=False, labels=labels)

[20]: df2.tenure_group.value_counts()

[20]: tenure_group
      1 - 12     2175
      61 - 72    1407
      13 - 24    1024
      25 - 36     832
```

Figure 3.4.6 Tenure group

3.5 DATAEXPLORATION

Here we will do some Univariate analysis with (one variable). The Figure 3.5 shows the code snippet related to **univariate analysis** using the Python library **Seaborn (sns)**. Here's a breakdown of the code. Relationship between Monthly Charges and TotalCharges.

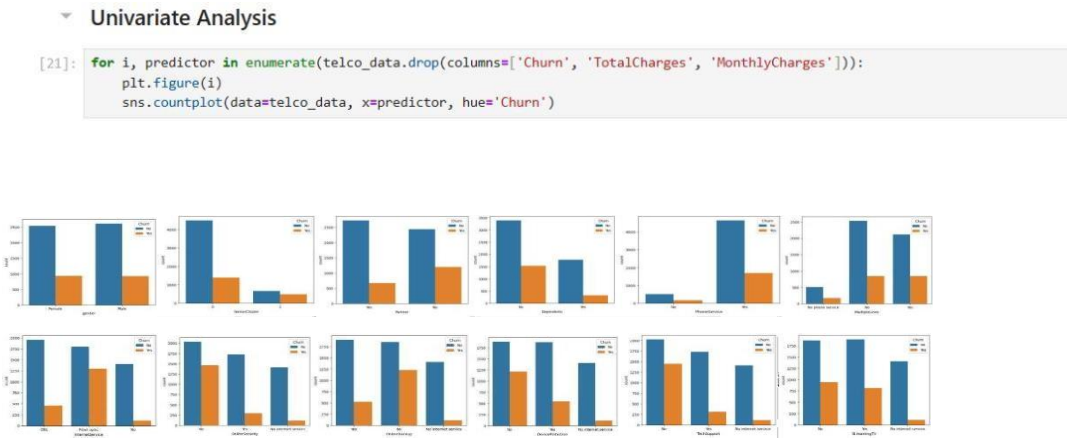
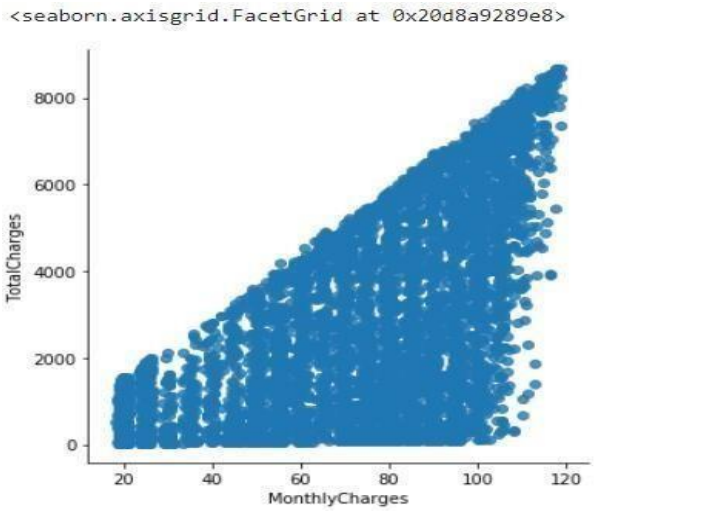


Figure 3.5 Uni variate analysis

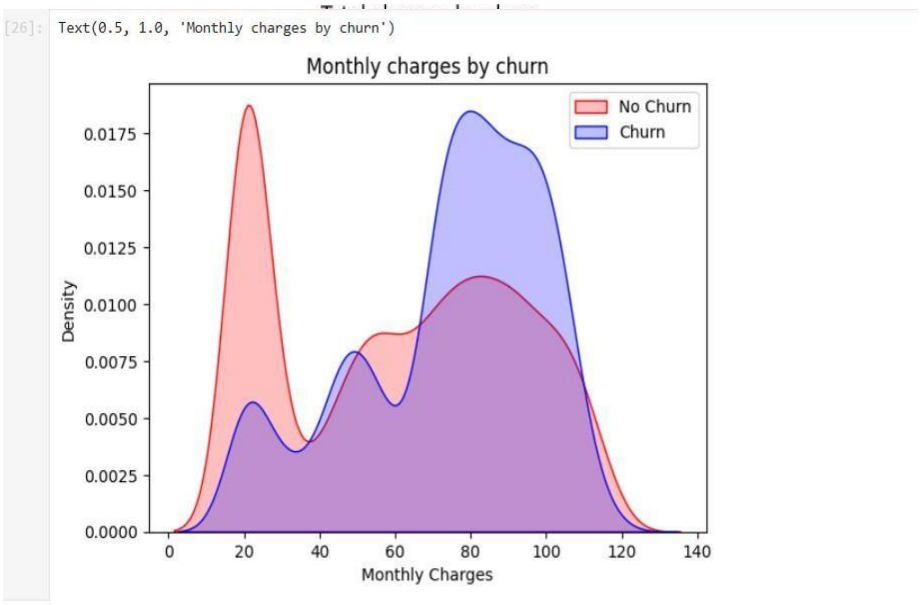
sns.lmplot(data=telco_data,x='MonthlyCharges',y='TotalCharges',fit_reg=False)

In the Figure3.5.1 and 3.5.2 it'sshows the distribution of monthly charges for customer who churned.



Total Charges increase as Monthly Charges increase - as expected.

Figure 3.5.1 Total Charges



Insight: Churn is high when Monthly Charges ar high

Figure 3.5.2 Monthly charges

3.6 Data Preprocessing

ML did not understand any value except binary value **(0,1)** , that's why we have to convert all the datainto binary value.The figure 3.6 showes the data preprocessing

From sk learn we have to import preprocessing

```
[38]: telco_data_dummies.to_csv('tel_churn.csv')

[39]: telco_data_dummies.head()
```

[39]:

	SeniorCitizen	MonthlyCharges	TotalCharges	Churn	gender_Female	gender_Male	Partner_No	Partner_Yes	Dependents_No	Dependents_Yes	...	PaymentMethod_Ba transfer (automa
0	0	29.85	29.85	0	1	0	0	1	1	0	...	
1	0	56.95	1889.50	0	0	1	1	0	1	0	...	
2	0	53.85	108.15	1	0	1	1	0	1	0	...	
3	0	42.30	1840.75	0	0	1	1	0	1	0	...	
4	0	70.70	151.65	1	1	0	1	0	1	0	...	

5 rows × 51 columns

Figure 3.6 Data Preprocessing

The Figure3.6 depicts the data preprocessing into CSV file for further analysis.

This will result in 7032 rows and 52 columns being obtained.

The Figure 3.6.1 shows the correlation matrix

```
[204]: corr_matrix
```

[204]:

	SeniorCitizen	MonthlyCharges	TotalCharges	Churn	gender_Female	gender_Male	Partner_No	Partner_Yes	Dependents_No	Dependent
SeniorCitizen	1.000000	0.219874	0.102411	0.150541	0.001819	-0.001819	-0.016957	0.016957	0.210550	-0.2
MonthlyCharges	0.219874	1.000000	0.651065	0.192858	0.013779	-0.013779	-0.097825	0.097825	0.112343	-0.1
TotalCharges	0.102411	0.651065	1.000000	-0.199484	-0.000048	0.000048	-0.319072	0.319072	-0.064653	0.0
Churn	0.150541	0.192858	-0.199484	1.000000	0.008545	-0.008545	0.149982	-0.149982	0.163128	-0.1
gender_Female	0.001819	0.013779	-0.000048	0.008545	1.000000	-1.000000	-0.001379	0.001379	0.010349	-0.0
gender_Male	-0.001819	-0.013779	0.000048	-0.008545	-1.000000	1.000000	0.001379	-0.001379	-0.010349	0.0
Partner_No	-0.016957	-0.097825	-0.319072	0.149982	-0.001379	0.001379	1.000000	-1.000000	0.452269	-0.4
Partner_Yes	0.016957	0.097825	0.319072	-0.149982	0.001379	-0.001379	-1.000000	1.000000	-0.452269	0.4
Dependents_No	0.210550	0.112343	-0.064653	0.163128	0.010349	-0.010349	0.452269	-0.452269	1.000000	-1.0
Dependents_Yes	-0.210550	-0.112343	0.064653	-0.163128	-0.010349	0.010349	-0.452269	0.452269	-1.000000	1.0
PhoneService_No	-0.008392	-0.248033	-0.113008	-0.011691	-0.007515	0.007515	0.018397	-0.018397	-0.001078	0.0
PhoneService_Yes	0.008392	0.248033	0.113008	0.011691	0.007515	-0.007515	-0.018397	0.018397	0.001078	-0.0
MultipleLines_No	-0.136377	-0.338514	-0.396765	-0.032654	-0.004335	0.004335	0.130028	-0.130028	-0.023388	0.0
MultipleLines_No phone service	-0.008392	-0.248033	-0.113008	-0.011691	-0.007515	0.007515	0.018397	-0.018397	-0.001078	0.0
MultipleLines_Yes	0.142996	0.490912	0.469042	0.040033	0.008883	-0.008883	-0.142561	0.142561	0.024307	-0.0
InternetService_DSL	-0.108276	-0.161368	-0.052190	-0.124141	-0.007584	0.007584	0.001043	-0.001043	-0.051593	0.0
InternetService_Fiber optic	0.254923	0.787195	0.360769	0.307463	0.011189	-0.011189	-0.001235	0.001235	0.164101	-0.1
InternetService_No	-0.182519	-0.763191	-0.374878	-0.227578	-0.004745	0.004745	0.000286	-0.000286	-0.138383	0.1
OnlineSecurity_No	0.185145	0.360220	-0.064515	0.342235	-0.010859	0.010859	0.129394	-0.129394	0.186979	-0.1

Figure 3.6.1 Correlation matrix

To understand easily we will show this in a graphical method called **heatmap**. This Figure3.6.2 shows the heatmap which is useful for visualizing the correlation between different variables in a dataset.

Figure 3.6.2 Heatmap

3.7 Algorithms

The model works by passing existing customer data through a machine learning model to establish the connection between features and targets and to make predictions about new customers.

ML models are powerful tools for Churn analysis, which helps business predict which customer are like to Churn (cancel their subscription or stop using a service). The Figure 3.7 depicts a flowchart illustrating the typical steps involved in a machine learningProcess.

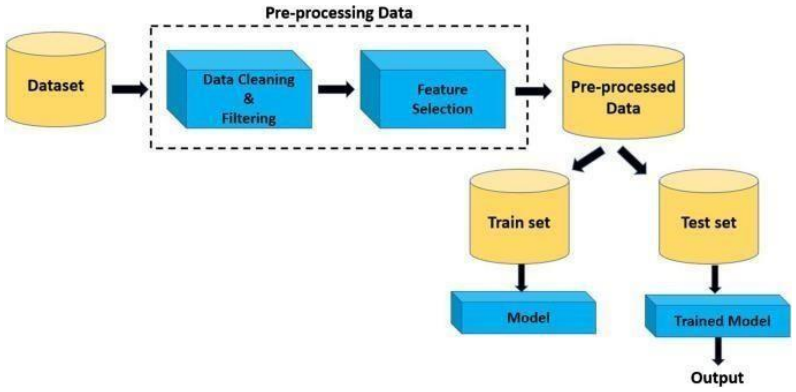


Figure 3.7 Pre-Processing Data

3.8 Benefits of ML for Churn analysis

ML models can analyze complex datasets and identify subtle patterns that might be missed bytraditional methods. This leads to more accurate churn predictions.

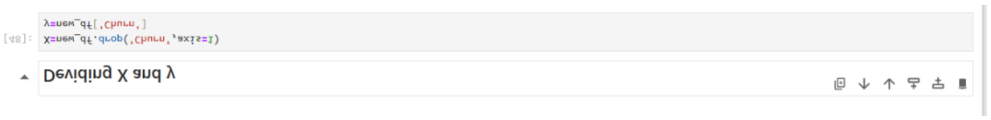
Scalability:

ML models can handle large volumes of data efficiently, making the deal for businesses with a large customerbase.

Customization:

ML models can be customized to specific industries and business models. This allows businesses to focus on the churn factors that matter most to them.

To work with ML models first we have to Split the dataset into two areas .From the two areas first oneis dependent variable and second one is independent variable.



Here we are predicting Churn that's why Churn is the dependent variable. and rest of all areindependent variable. Mostly Churn is depends on the independent variable.

After the splitting the data set into Variable (X,Y) we were get values in both the variable. Here we seethat the data set shows the imbalance ratio, for which we have to balance the dataset.

The Figure3.8.1 you sent shows a Python code snippet demonstrating the use of the SMOTE (SyntheticMinority Over-sampling Technique) method for handling imbalanced datasets. The figure 3.8.1 shows the Smote method

Figure 3.8.1 Smote Method

```
[49]: x_train,x_test,y_train,y_test = train_test_split (x_test,y_test,test_size = 0.2,random_state = 0)
```

Now store the value with test data of 20% in a variable. After that we will perform some scalingfeatures to find out a specific range. After that we will feeding the data into ML models.

The 2scaler function are Min Max scaler and Standard Scaler

MINMAXSCALER

A min-max scaler is to preprocess data by scaling features to a specific range.

Individual Feature Scaling:

Themin-maxscaleroperatesoneachfeature(column)ofyourdataindependently [5]. It finds theminimum and maximum values for each feature in the training data. The Figure3.8.2 shows a Python code snippet demonstrating how to use the Min-Max Scaler

```
[139]: from sklearn.preprocessing import MinMaxScaler
       model = MinMaxScaler()
       model.fit(x_train,y_train)
       model.fit_transform(x)

[139]: array([[0.        , 0.11542289, 0.0012751 , ..., 0.        , 1.        ,
               0.        ],
              [0.        , 0.38507463, 0.21586661, ..., 0.        , 0.        ,
               1.        ],
              [0.        , 0.35422886, 0.01031041, ..., 0.        , 0.        ,
               1.        ],
              ...,
              [0.        , 0.11293532, 0.03780868, ..., 0.        , 1.        ,
               0.        ],
              [1.        , 0.55870647, 0.03321025, ..., 0.        , 0.        ,
               1.        ],
              [0.        , 0.86965174, 0.78764136, ..., 0.        , 0.        ,
               0.        ]])
```

Figure 3.8.2 Feature Scaling

STANDARDSCALER

Target Distribution:

Standard scaler targets a standard normal distribution (mean0, standard deviation1)

Standardization:

Standard scaler aims to standardize your data by transforming features to have a zero mean(μ)and unit variance (σ)–essentially a standard normal distribution.

By standardizing the features standard scaler prevents features with larger scales fromdominating the model's learning. The figure 3.8.3 depicts how to use the Standard Scaler to standardize numerical features

```
[140]: from sklearn.preprocessing import StandardScaler
       scaler = StandardScaler()
       scaler.fit(x_train,y_train)
       scaler.fit_transform(x)

[140]: array([[-0.44032709, -1.16169394, -0.99419409, ..., -0.5253508 ,
                1.40476387, -0.54360352],
              [-0.44032709, -0.26087792, -0.17373982, ..., -0.5253508 ,
               -0.71186341,  1.83957601],
              [-0.44032709, -0.36392329, -0.95964911, ..., -0.5253508 ,
               -0.71186341,  1.83957601],
              ...,
              [-0.44032709, -1.17000405, -0.85451414, ..., -0.5253508 ,
                1.40476387, -0.54360352],
              [ 2.27103902,  0.31916782, -0.87209546, ..., -0.5253508 ,
               -0.71186341,  1.83957601],
              [-0.44032709,  1.35793167,  2.01234407, ..., -0.5253508 ,
               -0.71186341, -0.54360352]])
```

Figure 3.8.3 Standard Scaler

Chapter 4

PREDICTION MODEL

4.1 Logistic Regression:

Logistic regression is a statistical technique designed for binary classification tasks. It estimates the likelihood of an instance belonging to a specific category [6]. This likelihood is calculated using the logistic (or sigmoid) function, which transforms theoutput into values ranging between0and1. The figure 4.1 shows the accuracy using Logistic regression.

Figure 4.1 logistic regression

4.2 Confusion Matrix

Confusion matrix is best described as a method that helps evaluate the classification model by comparing the predicted and actual values with regards to values from testdata set. It is particularly useful in cases of binary classifications. The Figure 4.2 shows the accuracy using Confusion Matrix.

```
[212]: plt.figure(figsize=(5,3))
       sns.heatmap(cm,annot=True)
       plt.xlabel('Predicted')
       plt.ylabel('Truth')

[212]: Text(19.58333333333333, 0.5, 'Truth')
```

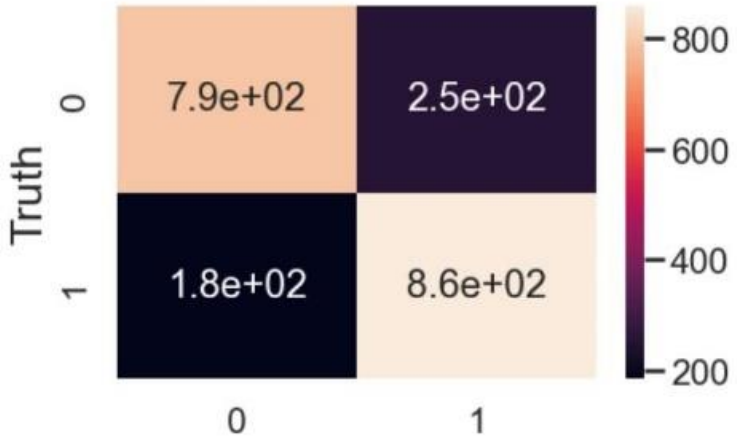


Figure 4.2 Accuracy using Confusion Matrix

4.3 Random Forest Classifier

Random Forests (RF) are a type of ensemble learning method where a collection of decision trees is trained. The Figure 4.3 shows the accuracy using random forest classifier.

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)}$$

```
[113]: grid_search1.best_estimator_
[113]: RandomForestClassifier(max_depth=20, min_samples_leaf=5, n_jobs=-1,
                              random_state=42)
[114]: rf_sm=RandomForestClassifier(max_depth=20, min_samples_leaf=5, n_estimators=200,n_jobs=-1, random_state=42)
[115]: rf_sm_model=rf_sm.fit(X1_train_sm,y1_train_sm)
[116]: rf_sm_pred=rf_sm_model.predict(X1_test_sm)
[117]: accuracy_score(y1_test_sm,rf_sm_pred)
[117]: 0.8259827420901247
```

Figure 4.3 Accuracy using Random Forest Classifier

4.4 SVM:

Support Vector Machine (SVM) is an algorithm that classifies **Hyperplane**: In SVM,the hyperplane serves as a boundary that divides data points into different classes. In 2D, it's a line, but in higher dimensions, it becomes a hyperplane. The Figure 4.4 shows the accuracy using Support Vector Machine.

```
Support Vector Machine (Smote)
[120]: svc_sm=SVC(C=1,gamma=0.1,kernel='rbf',decision_function_shape='ovo',random_state=1)
[121]: svc_sm_model=svc_sm.fit(X1_train_sm,y1_train_sm)
[122]: svc_sm_pred=svc_sm_model.predict(X1_test_sm)
[123]: accuracy_score(y1_test_sm,svc_sm_pred)
[123]: 0.7444870565675935
```

Figure 4.4 Accuracy using Support vector Machine

4.5 Naive Bayes:

The Naive Bayes algorithm is a supervised learning technique grounded in **Bayes'Theorem**, commonly used for classification tasks **Bayes'Theorem**

formula:

$$P(A/B)=P(B/A)\cdot P(A)P(B)P(A/B) = \frac{P(B/A) \cdot P(A)}{P(B)}P(A/B)=P(B)P(B/A)\cdot P(A)$$

Comparison of ML Models:

After evaluating various machine learning models, it was found that the **Random Forest classifier** achieved the highest accuracy, with a rate of **81%**. By using this model, we will get proper analysis of the Churn rate and how to work with it. Figure 4.5 shows the accuracy using naïve bayes model

```
[126]: clf_sm=BernoulliNB(alpha=0.5, fit_prior= True,)

[127]: clf_sm_model=clf_sm.fit(X1_train_sm,y1_train_sm)

[128]: clf_sm_pred=clf_sm_model.predict(X1_test_sm)

[129]: accuracy_score(y1_test_sm,clf_sm_pred)

[129]: 0.7622243528283796

[130]: print(classification_report(y1_test_sm,clf_sm_pred))

              precision    recall  f1-score   support

           0       0.78      0.73      0.75      1043
           1       0.75      0.79      0.77      1043

    accuracy                           0.76      2086
   macro avg       0.76      0.76      0.76      2086
weighted avg       0.76      0.76      0.76      2086

[131]: confusion_matrix(y1_test_sm,clf_sm_pred)

[131]: array([[764, 279],
       [217, 826]], dtype=int64)
```

Figure 4.5 Accuracy using Naive Bayes Model

4.6 ANN :

Now it's time to construct the architecture of the ANN model using the Keras library, which is a high-level API for building neural networks in TensorFlow. The figure 4.6 shows the accuracy using ANN Model and the figure 4.6.1 shows the Sequential Neural Networking.

```
]: from tensorflow import keras
   from tensorflow.keras import layers

]: from tensorflow.keras.callbacks import EarlyStopping

]: early_stopping = EarlyStopping(
       min_delta=0.001, # minimium amount of change to count as an improvement
       patience=20, # how many epochs to wait before stopping
       restore_best_weights=True,
   )

]: tf_model=keras.Sequential([
       layers.Dense(units=1024,activation='relu',input_shape=[43]),
       layers.Dropout(0.3),
       layers.BatchNormalization(),
       layers.Dense(units=1024,activation='relu',),
       layers.Dropout(0.3),
       layers.BatchNormalization(),
       layers.Dense(units=1024,activation='relu',),
       layers.Dropout(0.3),
       layers.BatchNormalization(),
       layers.Dense(units=1)
   ])
```

Figure 4.6 Accuracy using ANN

```
]: tf_model.summary()
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_4 (Dense)             (None, 1024)              45056

 dropout_3 (Dropout)         (None, 1024)              0

 batch_normalization_3 (Batc  (None, 1024)             4096
 hNormalization)

 dense_5 (Dense)             (None, 1024)              1049600

 dropout_4 (Dropout)         (None, 1024)              0

 batch_normalization_4 (Batc  (None, 1024)             4096
 hNormalization)

 dense_6 (Dense)             (None, 1024)              1049600

 dropout_5 (Dropout)         (None, 1024)              0

 batch_normalization_5 (Batc  (None, 1024)             4096
 hNormalization)

 dense_7 (Dense)             (None, 1)                 1025

=================================================================
Total params: 2,157,569
Trainable params: 2,151,425
Non-trainable params: 6,144
```

Figure 4.6.1 Sequential neural network

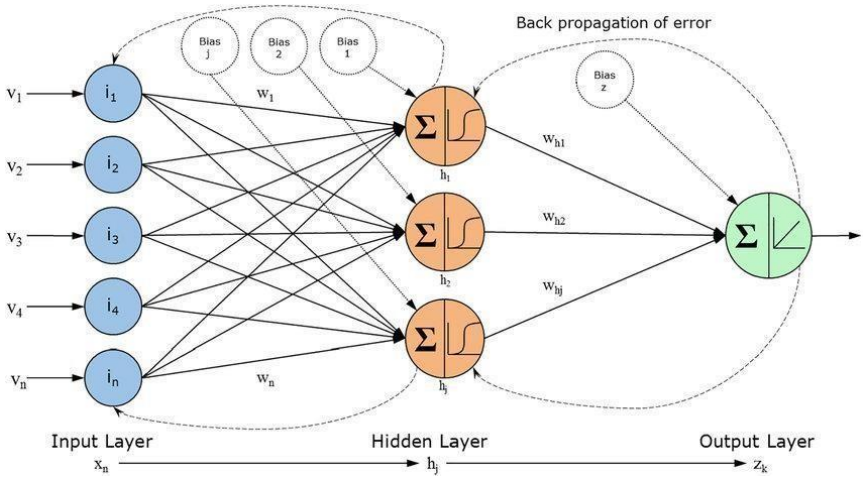


Figure 4.6.2 Back Propagation of error

Figure 4.6.2 indicates the back progression

Once the architecture of the Artificial Neural Network (ANN) is defined, the nextstep is to train the model using the training dataset

The model compile function is used to set the loss function, optimizer, and evaluation metrics. In the case of binary classification, **accuracy** is calculated as theratio of correct predictions to the total number of predictions.

The **'Adam'** optimizer refers to a popular optimization algorithm known for its adaptive learning rate. It's commonly used in deep learning because of its efficiencyand strong performance. In below figure the ml model is train by Tensor flow. The Figure 4.6.3 Indicates the Tensor flow

Figure 4.6.3 Tensor Flow to train a machine learning model

In the Below figure3.4.6 shows the calculation of loss decreases

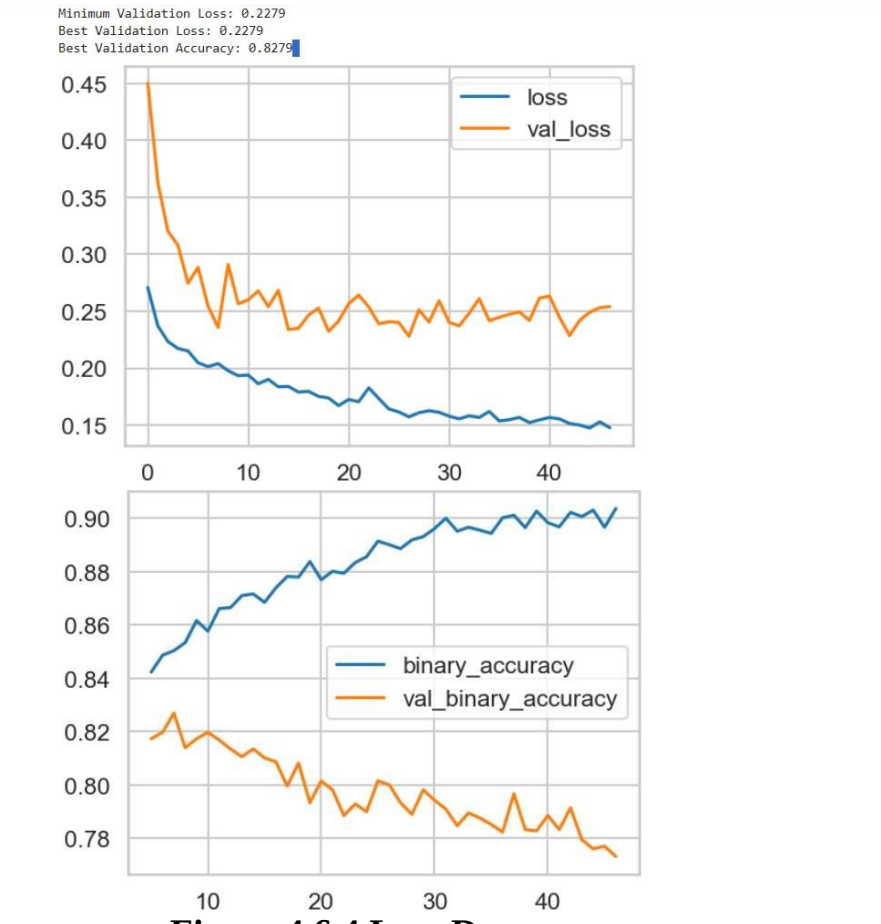


Figure 4.6.4 Loss Decreases

Chapter 5

RESULTS

Upon reviewing the results, we compare the accuracy of different models in predicting customer churn. The Random Forest Classification model stands out, achieving the highest accuracy at **81%**.

Managerial Insights and Practical Implications: The findings from this research demonstrate that it is feasible to achieve a solid level of accuracy in predicting customer churn. Managers in the telecom sector can leverage the model developed in this study to detect churn early and address underlying issues before they escalate. By applying the model to churn-related data, it becomes possible to pinpoint the key factors contributing to customer attrition.

With insights into these contributing factors, telecom management can prioritize service improvements that reduce churn. This model, although developed with a specific dataset, is versatile and can be adapted for different industries, especially when working with Persian text data. With further optimization and fine-tuning, this method offers a reliable tool for predicting customer churn and identifying ways to enhance customer retention strategies. The comparison between models is shown in Table 5.1.

Table 5.1 Comparison Between models

ML Models	Accuracy	Precision	Recall	F1-Score
Logistic Regression	80.41%	0.66	0.58	0.59
Confusion Matrix	80.59%	0.67	0.57	0.58
Random Forest Classifier	81.20%	0.66	0.45	0.56
SVM	79.98%	0.89	0.84	0.85
Naive Bayes	80.29%	0.68	0.56	0.60
ANN	81.10%	0.66	0.49	0.56

Chapter 6

CONCLUSION

In conclusion, telecom churn prediction is a critical strategy for reducing customer attrition and enhancing customer retention in a highly competitive industry. By leveraging machine learning models, businesses can identify at-risk customers and take proactive steps to address their concerns. Among the modelsimplemented, Random Forest emerged as the most effective, delivering superiorperformance in terms of accuracy, precision, recall, and F1-score. This highlightsits ability to handle complex relationships within the data while maintaining robustness against overfitting.

Additionally, addressing the class imbalance in the dataset through sampling methods proved to be beneficial. Balancing techniques improved the recall of themodels—enhancing their capability to identify potential churners—while slightly reducing precision. This trade-off is acceptable in churn prediction, as the cost of failing to identify a churner is typically higher than the cost of a false positive.

Overall, integrating machine learning models like Random Forest into telecom churn prediction systems allows companies to adopt a data-driven approach, anticipate customer needs, and improve customer retention, ultimately leading to enhanced profitability and competitive advantage. Future work could explorethe inclusion of deep learning models and advanced feature engineering techniques to further refine predictive performance.

REFERENCES

1. Saha, Somak, et al. "ChurnNet: Deep Learning Enhanced Customer Churn Prediction in Telecommunication Industry." *IEEE Access* (2024).

2. Wagh, Sharmila K., et al. "Customer churn prediction in telecom sector using machine learning techniques." *Results in Control and Optimization* 14 (2024): 100342.

3. Ribeiro, Hugo, et al. "Determinants of churn in telecommunication services: a systematic literature review." *Management Review Quarterly* 74.3 (2024): 1327-1364.

4. Chong, Angela Yi Wen, et al. "Customer churn prediction of telecom company using machine learning algorithms." *Journal of Soft Computing and Data Mining* 4.2 (2023): 1-22.

5. Melian, Denisa Maria, et al. "Customer churn prediction in telecommunication industry. A data analysis techniques approach." *Postmodern Openings* 13.1 Sup1 (2022): 78-104.

6. Ahmad, Abdelrahim Kasem, Assef Jafar, and Kadan Aljoumaa. "Customer churn prediction in telecom using machine learning in big data platform." *Journal Of Big Data* 6.1 (2019): 1-24.