TELECOM CHURN PREDICTION MODEL

Supervised by:
Mr.Durga Prasad

Done By:
NIKHITHA BEECHARAJU
ANJANA KAMANA
MELISSA MARY SMITH
MUGDHAMA KISHOR PATIL

INDEX

1.INTRODUCTION

1.1 OVERVIEW

In this report, we discuss the solution to the problem of customer churn. Telecom churn prediction model is the solution for the problem. We make use of machine learning to sove the problem. Different algorithms are used to find the best fit according to their accuracy.

1.2 PURPOSE

Customer churn 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. Churn prediction helps in identifying those customers who are likely to leave a company.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

Customer churn occurs when customers or subscribers stop doing business with a company or service. Also known as customer attrition, customer churn is a critical metric because it is much less expensive to retain existing customers than it is to acquire new customers – earning business from new customers means working leads all the way through the sales funnel, utilizing your marketing and sales resources throughout the process. Customer retention, on the other hand, is generally more cost-effective as you've already earned the trust and loyalty of existing customers. Customer churn impedes growth, so companies should have a defined method for calculating customer churn in a given period of time. By being aware of and monitoring churn rate, organizations are equipped to determine their customer retention success rates and identify strategies for improvement.
There are a multitude of issues that can lead customers to leave a business, but there are a few that are considered to be the leading causes of customer churn. The first is poor customer service. One study found that nearly nine out of ten customers have

abandoned a business due to a poor experience. We are living and working in the era of the customer, and customers are demanding exceptional customer service and experiences. When they don't receive it, they flock to competitors and share their negative experiences on social media: 59% of 25-34-year-olds share poor customer experiences online. Poor customer service, therefore, can result in many more customers churning than simply the one customer who had a poor service experience. Other causes of customer churn include a poor onboarding process, a lack of ongoing customer success, natural causes that occur for all businesses from time to time, a lack of value, low-quality communications, and a lack of brand loyalty.

2.2 PROPOSED SOLUTION

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 a churn prediction model which assists telecom operators to predict customers who are most likely subject to churn. The model developed in this work uses machine learning techniques on big data platform and builds a new way of features' engineering and selection. In order to measure the performance of the model, the Area Under Curve (AUC) standard measure is adopted.

3.THEORITICAL ANALYSIS

Classification in machine learning and statistics is a supervised learning approach in which the computer program learns from the data given to it and make new observations or classifications. In this article, we will learn about classification in machine learning in detail.

Classification is a process of categorizing a given set of data into classes, It can be performed on both structured or unstructured data. The process starts with predicting the class of given data points. The classes are often referred to as target, label or categories.

The classification predictive modeling is the task of approximating the mapping function from input variables to discrete output variables. The main goal is to identify which class/category the new data will fall into.

Classification Algorithms

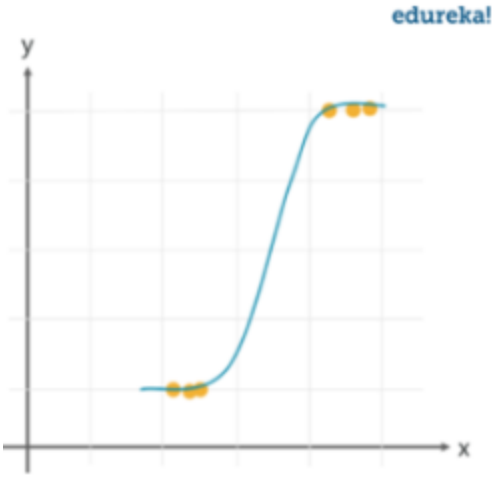
In machine learning, classification is a supervised learning concept which basically

categorizes a set of data into classes. The most common classification problems are – speech recognition, face detection, handwriting recognition, document classification, etc. It can be either a binary classification problem or a multi-class problem too. There are a bunch of machine learning algorithms for classification in machine learning. Let us take a look at those classification algorithms in machine learning.

Logistic Regression

It is a classification algorithm in machine learning that uses one or more independent variables to determine an outcome. The outcome is measured with a dichotomous variable meaning it will have only two possible outcomes.

The goal of logistic regression is to find a best-fitting relationship between the dependent variable and a set of independent variables. It is better than other binary classification algorithms like nearest neighbor since it quantitatively explains the factors leading to classification.



Naive Bayes Classifier

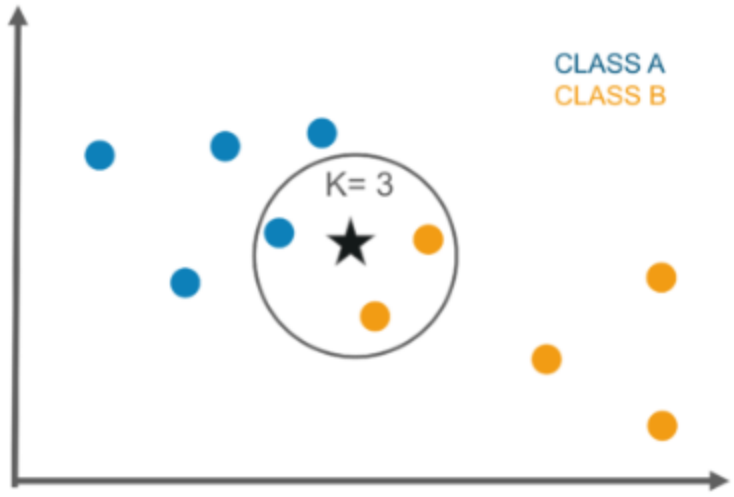
It is a classification algorithm based on Bayes's theorem which gives an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Even if the features depend on each other, all of these properties contribute to the probability independently. Naive Bayes model is easy to make and is particularly useful for comparatively large data sets. Even with a simplistic approach, Naive Bayes is known to outperform most of the classification methods in machine learning. Following is the Bayes theorem to implement the Naive Bayes Theorem.

$$P(C_i|\, x_1, x_2 \ldots, x_n) = \frac{P(x_1, x_2 \ldots, x_n|C_i).P(C_i)}{P(x_1, x_2 \ldots, x_n)} \; for\; 1 < i < k$$

K-Nearest Neighbor

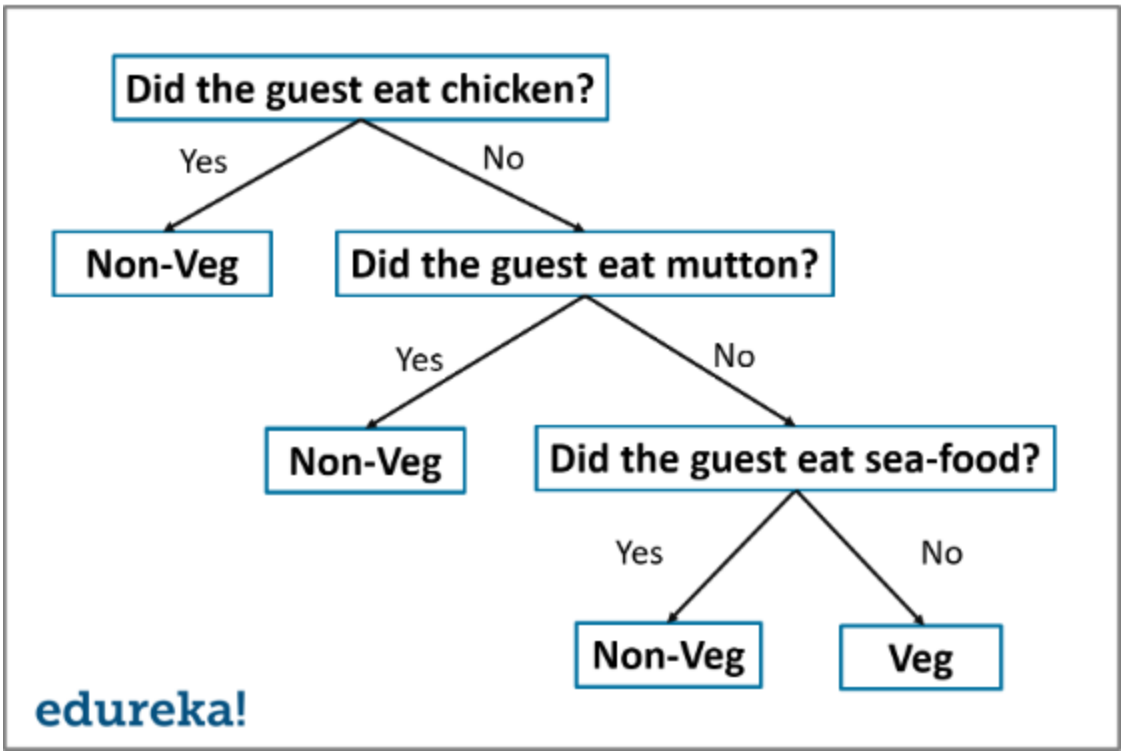
It is a lazy learning algorithm that stores all instances corresponding to training data in n-dimensional space. It is a lazy learning algorithm as it does not focus on constructing a general internal model, instead, it works on storing instances of training data. Classification is computed from a simple majority vote of the k nearest neighbors of each point. It is supervised and takes a bunch of labeled points and uses them to label other points. To label a new point, it looks at the labeled points closest to that new point also known as its nearest neighbors. It has those neighbors vote, so whichever label the most of the neighbors have is the label for the new point. The "k" is the number of neighbors it checks.



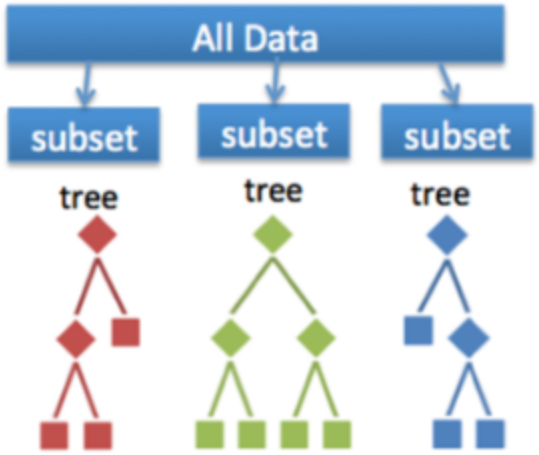
Decision Tree

The decision tree algorithm builds the classification model in the form of a tree structure. It utilizes the if-then rules which are equally exhaustive and mutually exclusive in classification. The process goes on with breaking down the data into smaller structures and eventually associating it with an incremental decision tree. The final structure looks like a tree with nodes and leaves. The rules are learned sequentially using the training data one at a time. Each time a rule is learned, the tuples covering the

rules are removed. The process continues on the training set until the termination point is met.
The tree is constructed in a top-down recursive divide and conquer approach. A decision node will have two or more branches and a leaf represents a classification or decision. The topmost node in the decision tree that corresponds to the best predictor is called the root node, and the best thing about a decision tree is that it can handle both categorical and numerical data.

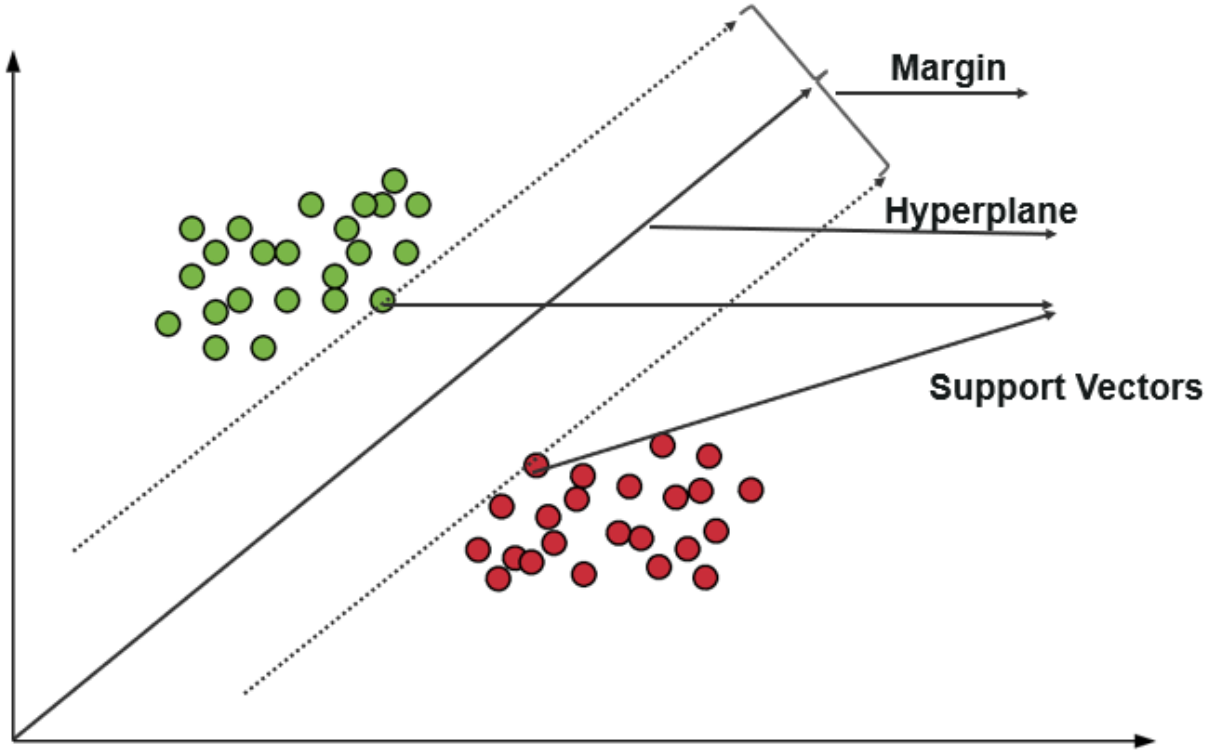


Random Forest
Random decision trees or random forest are an ensemble learning method for classification, regression, etc. It operates by constructing a multitude of decision trees at training time and outputs the class that is the mode of the classes or classification or mean prediction(regression) of the individual trees.
A random forest is a meta-estimator that fits a number of trees on various subsamples of data sets and then uses an average to improve the accuracy in the model's predictive nature. The sub-sample size is always the same as that of the original input size but the samples are often drawn with replacements.



Support Vector Machine

The support vector machine is a classifier that represents the training data as points in space separated into categories by a gap as wide as possible. New points are then added to space by predicting which category they fall into and which space they will belong to.



4.EXPERIMENTAL ANALYSIS

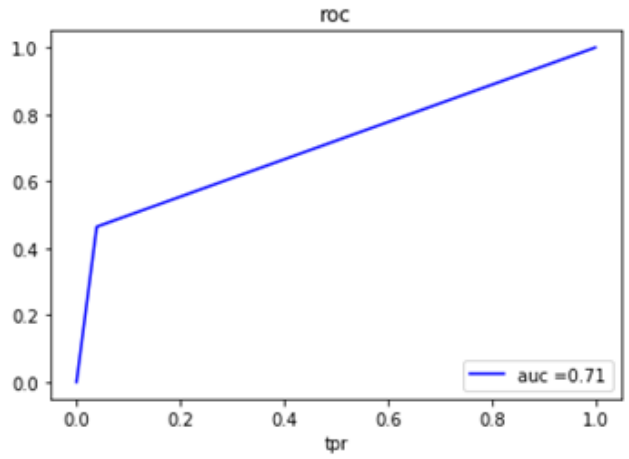
Training the Model using Machine Learning Algorithms
A number of algorithms were considered while deciding the final model
k-Nearest Neighbor, Logistic Regression, Naïve Bayes,Random Forest Classification,Decision Tree Classification.
After numerous trials, these models were compared to arrive at a suitable model for predicting adult income based on the given set of attributes

Random Tree Classifier

When we applied the random tree classifier to the data set , the accuracy obtained was 86.05% and AUC value was 0.71.

```
In [28]: from sklearn.metrics import accuracy_score

         rfcaccuracy=accuracy_score(y_test,rfcpred)

In [29]: rfcaccuracy

Out[29]: 0.8605
```

```
In [33]: plt.plot(fpr,tpr,"b",label="auc =%0.2f"%roc_auc)
         plt.legend(loc='lower right')
         plt.title("roc")
         plt.xlabel("fpr")
         plt.xlabel("tpr")

Out[33]: Text(0.5, 0, 'tpr')
```

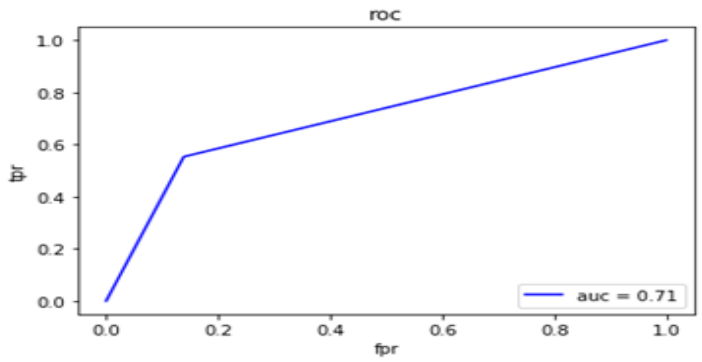


Decision Tree Classifier

When we applied the decision tree classifier to the data set , the accuracy obtained was 79.9% and AUC value was 0.71.

```
In [38]: dtcttacc = accuracy_score(y_test,dtcpred)
         dtcttacc

Out[38]: 0.799
```

```
In [41]: import matplotlib.pyplot as plt
         plt.plot(dtcfpr,dtctpr,"b",label = "auc = %0.2f"%dtcroc_auc)
         plt.legend(loc = "lower right")
         plt.title("roc")
         plt.xlabel("fpr")
         plt.ylabel("tpr")

Out[41]: Text(0, 0.5, 'tpr')
```

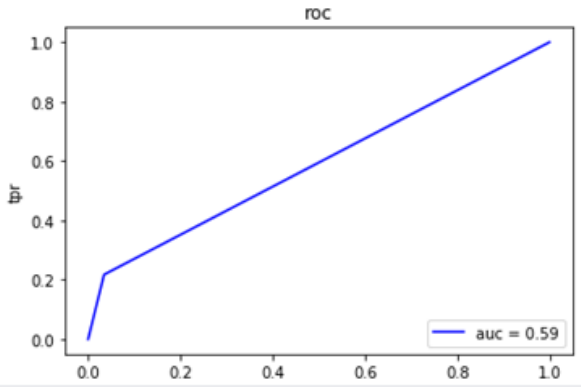


Logistic Regression

When we applied the logistic regression to the data set , the accuracy obtained was 81.4% and AUC value was 0.59.

```
In [45]: lracc = accuracy_score(y_test,lrpred)
         lracc

Out[45]: 0.814
```

```
In [47]: import sklearn.metrics as metrics
         lrfpr,lrtpr,threshold = metrics.roc_curve(y_test,lrpred)
         lrroc_auc = metrics.auc(lrfpr,lrtpr)

In [48]: import matplotlib.pyplot as plt
         plt.plot(lrfpr,lrtpr,"b",label = "auc = %0.2f"%lrroc_auc)
         plt.legend(loc = "lower right")
         plt.title("roc")
         plt.xlabel("fpr")
         plt.ylabel("tpr")

Out[48]: Text(0, 0.5, 'tpr')
```

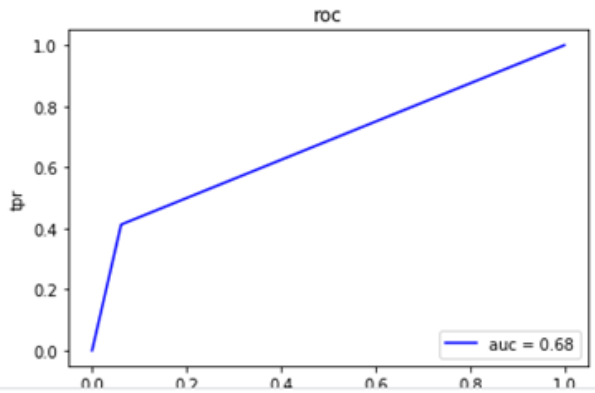


K-Neighbor Classifier

When we applied the k-neighbor classifier to the data set , the accuracy obtained was 83.2% and AUC value was 0.68.

```
In [51]: knnacc = accuracy_score(y_test,knnpred)
         knnacc

Out[51]: 0.832
```

```
In [53]: import sklearn.metrics as metrics
         knnfpr,knntpr,threshold = metrics.roc_curve(y_test,knnpred)
         knnroc_auc = metrics.auc(knnfpr,knntpr)

In [54]: import matplotlib.pyplot as plt
         plt.plot(knnfpr,knntpr,"b",label = "auc = %0.2f"%knnroc_auc)
         plt.legend(loc = "lower right")
         plt.title("roc")
         plt.xlabel("fpr")
         plt.ylabel("tpr")

Out[54]: Text(0, 0.5, 'tpr')
```

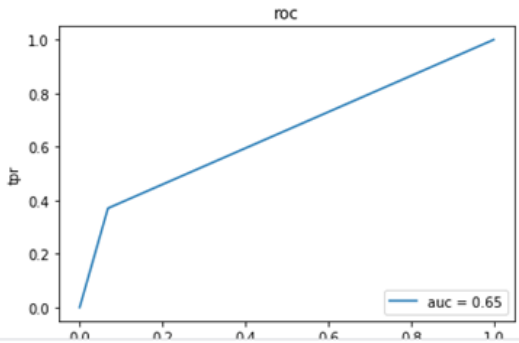


Naive Bayes

When we applied the Naive Bayes to the data set , the accuracy obtained was 81.8% and AUC value was 0.65.

```
In [59]: nbacc = accuracy_score(y_test,nbpred)
         nbacc

Out[59]: 0.818
```

```
In [61]: import sklearn.metrics as metrics
         nbfpr,nbtpr ,threshold = metrics.roc_curve(y_test,nbpred)
         nbroc_auc = metrics.auc(nbfpr,nbtpr)

In [62]: import matplotlib.pyplot as plt
         plt.plot(nbfpr,nbtpr,label = "auc = %0.2f"%nbroc_auc)
         plt.legend(loc = "lower right")
         plt.title("roc")
         plt.xlabel("fpr")
         plt.ylabel("tpr")

Out[62]: Text(0, 0.5, 'tpr')
```



Support Vector Machine

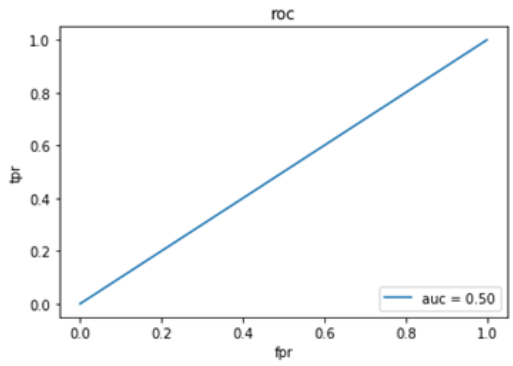
When we applied the Support Vector Machine to the data set , the accuracy obtained was 79.75% and AUC value was 0.5.

```
In [68]: svmacc = accuracy_score(y_test,svmpred)
         svmacc

Out[68]: 0.7975
```

In [66]:
```
import sklearn.metrics as metrics
svmfpr,svmtpr ,threshold = metrics.roc_curve(y_test,svmpred)
svmroc_auc = metrics.auc(svmfpr,svmtpr)
```

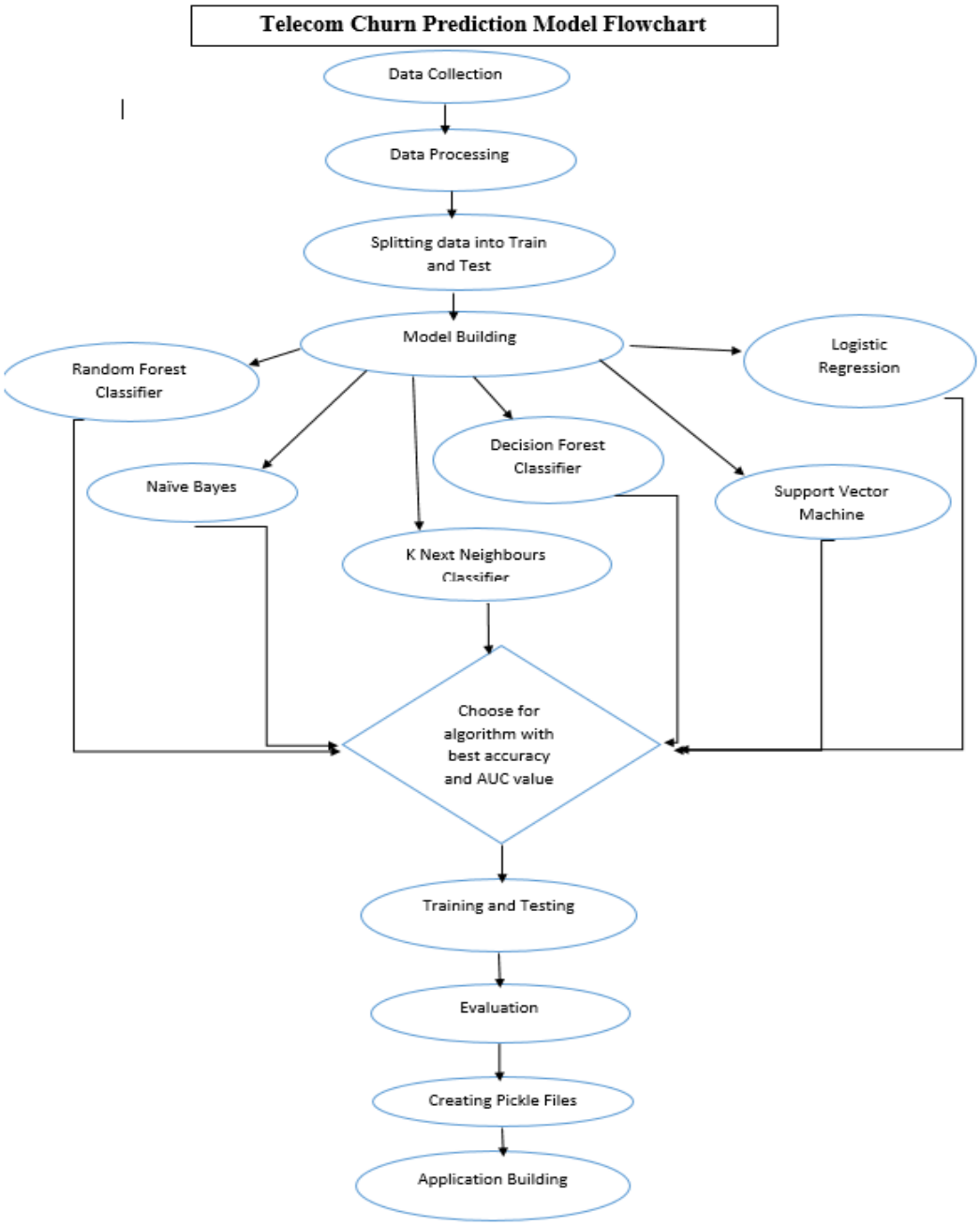
In [67]:
```
import matplotlib.pyplot as plt
plt.plot(svmfpr,svmtpr,label = "auc = %0.2f"%svmroc_auc)
plt.legend(loc = "lower right")
plt.title("roc")
plt.xlabel("fpr")
plt.ylabel("tpr")
```

Out[67]:
```
Text(0, 0.5, 'tpr')
```



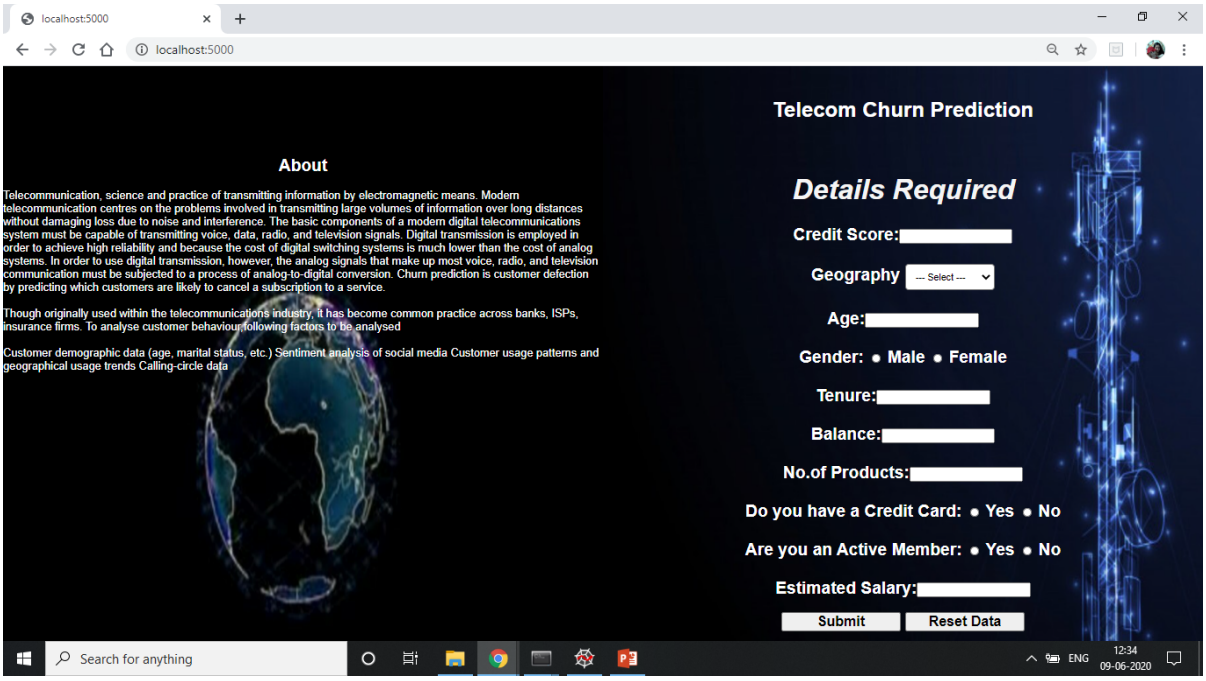
Hence, the Random Forest Classifier algorithm is to be used as it has the best accuracy and AUC value.

5.FLOWCHART

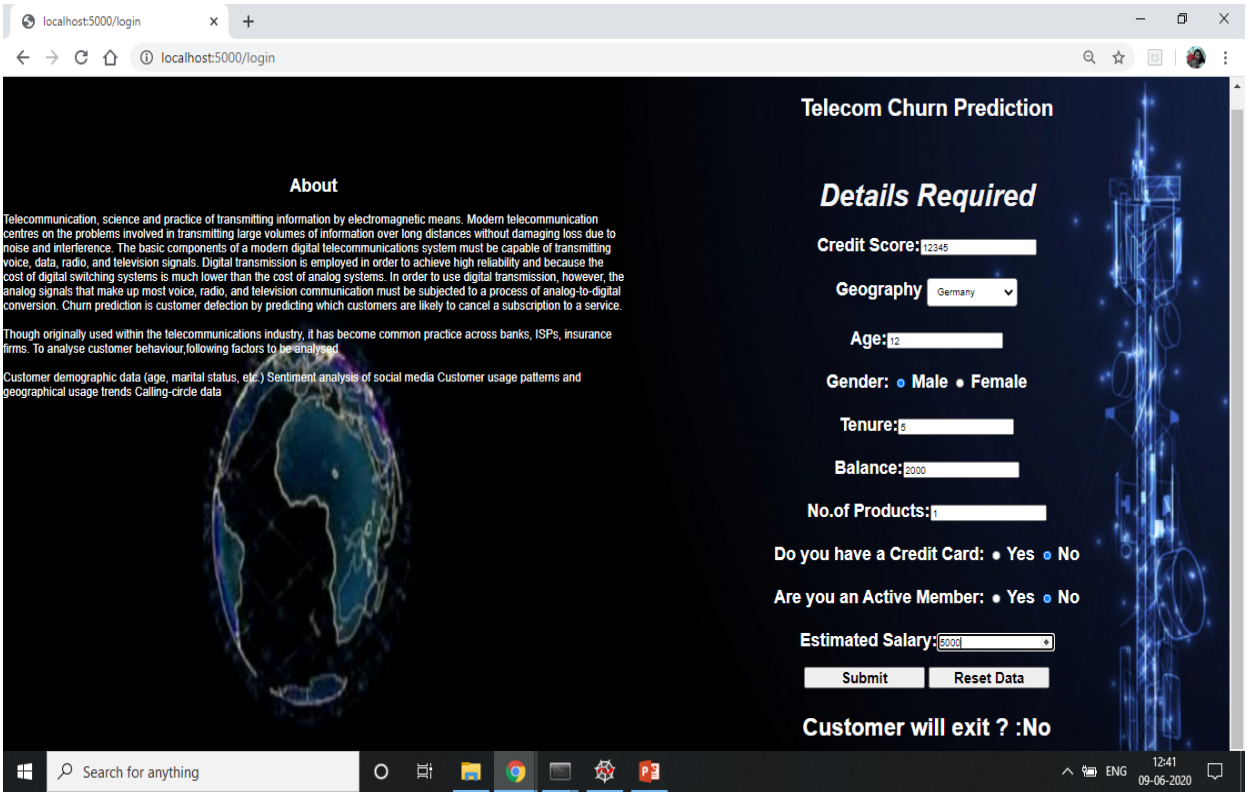

Telecom Churn Prediction Model Flowchart
Data Collection
Data Processing
Splitting data into Train and Test
Model Building
Random Forest Classifier
Logistic Regression
Naïve Bayes
Decision Forest Classifier
Support Vector Machine
K Next Neighbours Classifier
Choose for algorithm with best accuracy and AUC value
Training and Testing
Evaluation
Creating Pickle Files
Application Building

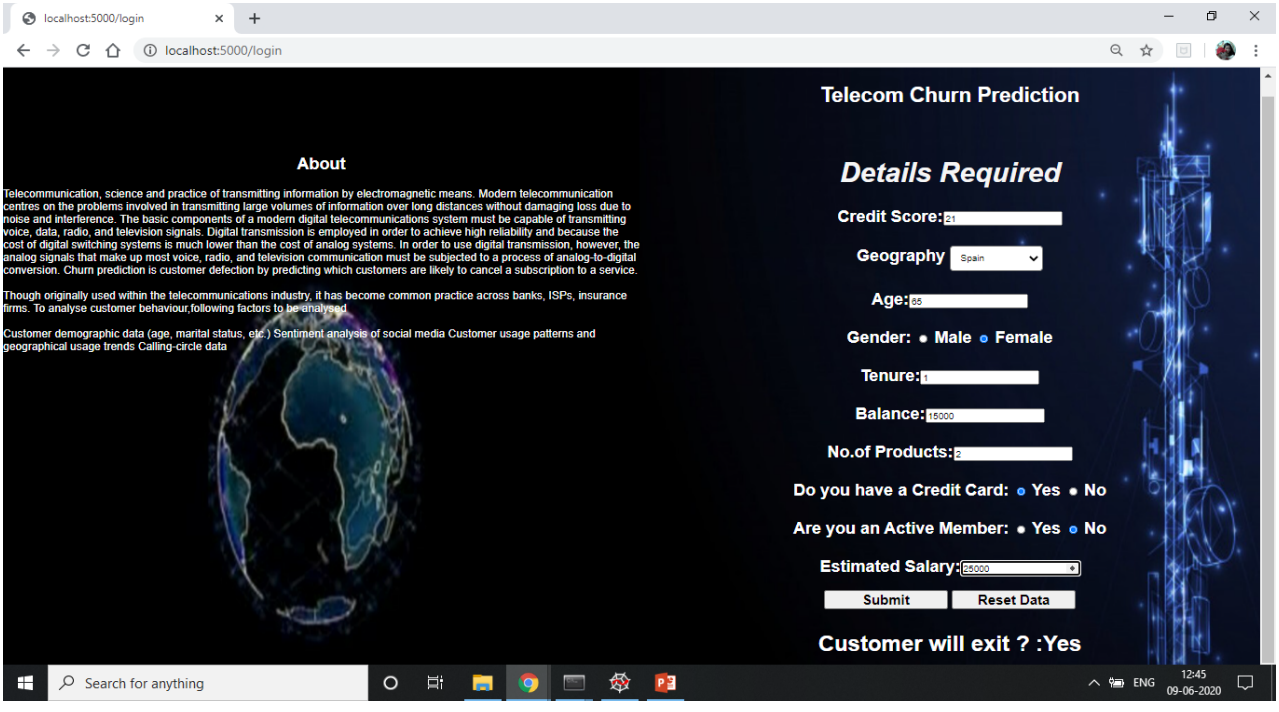
6.RESULT

The Application User Interface



The Prediction result in UI





7.ADVANTAGES AND DISADVANTAGES

7.1 ADVANTAGES:

- Increase Revenue - Upsell to existing customers is easier and more cost-effective rather than selling to new ones.
- Retain More Customers- Launch new strategies to increase loyalty to the product.
- Avoid Losses - Retain existing customers by stopping customer churn.
- Access all the relevant data quickly.
- Segment the customers based on behavior and demographics to improve retention
- Minimize acquisition costs and increase marketing efficiency
- Keep customers engaged and loyal.
- Improve Efficiency
- Inspire Collaboration
- Bring Flexibility to the Workplace
- Save Time

7.2 DISADVANTAGES:

- Eliminate Face to Face Contact - Most of the meetings held over video conference systems are mostly business or project oriented and do not allow for socialization, that increases bad relationship at work.
- Increase Vulnerability to Information Hacking and Attacks - With sensitive data being transmitted and shared over the Internet and other networks, there's a possibility people will try to hack it either for their own pleasure or for use by the competition. Communication through video conferences, teleconferences and video calls, unsafe and costly to a larger extent.
- Not fit for continuous variables: While working with continuous numeric variables decision tree losses information when it categorizes variables in different categories.

8.APPLICATIONS

- In Business to know the customers reviews
- In Multinational companies or Advertisement Agencies will the customer continue with the company or not.
- In telecom department if the customer is willing to buy there products the next time or not.

9.CONCLUSION

Machine Learning overall is a very powerful tool to revolutionize the way things work. It is very user-friendly and hence helps the user to predict the data easily and simultaneously and keeps improving the accuracy and efficiency by itself. Classification has been done to predict whether a telecommunication customer will exit or not based on a certain set of attributes. The Random Forest Classification Model was deployed which clocked the highest accuracy of 0.8605(86%) ,making it suitable to choose it over others.

10.FUTURE SCOPE

Having the ability to accurately predict future churn rates is necessary because it helps your business gain a better understanding of future expected revenue. Churn prediction modeling techniques attempt to understand the precise customer behaviors and attributes which signal the risk and timing of customer churn. As this can predict customers view towards the business so the respective company can persuade the customers to stay or can start the business with new customers when they feel the customer might exit no matter what. Predicting churn rates can also help your business identify and improve upon areas where customer service is lacking. And, by making those improvements, you can decrease churn and improve revenue numbers.

11. BIBLIOGRAPHY

- https://www.researchgate.net/publication/325419986_A_comparative_study_of_customer_churn_prediction_in_telecom_industry_using_ensemble_based_classifiers
- https://www.eztalks.com/unified-communications/advantages-and-disadvantages-of-telecommunication.html
- https://www.google.com/search?q=telecommunication+dark+images&tbm=isch&ved=2ahUKEwib0uLKiPPpAhXS0XMBHQYmCaQQ2-cCegQIABAA&oq=telecommunication+dar&gs_lcp=CgNpbWcQARgAMgQIIxAnOgIIADoECAAQGFCfF1jaNGDwQGgAcAB4AIABywKIAbQOkgEHMC42LjIuMpgBAKABAaoBC2d3cy13aXotaW1n&sclient=img&ei=0KDeXpuqK9Kjz7sPhsykoAo&rlz=1C1CHZL_enIN833IN833
- https://www.edureka.co/blog/classification-in-machine-learning/
- https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0191-6#:~:text=The%20main%20contribution%20of%20our,most%20likely%20subject%20to%20churn.&text=The%20dataset%20contained%20all%20customers,evaluate%20the%20system%20at%20SyriaTel
- https://www.qualtrics.com/experience-management/customer/customer-churn/

12. APPENDIX

A. SOURCE CODE

Project.ipynb

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
dataset=pd.read_csv("datasets_156197_358170_Churn_Modelling.csv")
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
dataset["Geography"]=le.fit_transform(dataset["Geography"])
dataset["Gender"]=le.fit_transform(dataset["Gender"])
dataset.drop("RowNumber",axis=1,inplace=True)
dataset.drop("Surname",axis=1,inplace=True)
x=dataset.iloc[:,1:11].values
y=dataset.iloc[:,11:12].values
from sklearn.preprocessing import OneHotEncoder
one=OneHotEncoder()
z=one.fit_transform(x[:,1:2]).toarray()
x=np.delete(x,1,axis=1)
x=np.concatenate((z,x),axis=1)
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.fit_transform(x_test)
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier(n_estimators=10,criterion='entropy',random_state=0)
rfc.fit(x_train,y_train)
import pickle
pickle.dump(rfc,open('exit.pkl','wb'))
pickle.dump(sc,open('scaler.pkl','wb'))
rfcpred=rfc.predict(x_test)
from sklearn.metrics import accuracy_score
```

```
rfcaccuracy=accuracy_score(y_test,rfcpred)
import sklearn.metrics as metrics
fpr,tpr,threshold=metrics.roc_curve(y_test,rfcpred)
roc_auc=metrics.auc(fpr,tpr)
plt.plot(fpr,tpr,"b",label="auc =%0.2f"%roc_auc)
plt.legend(loc='lower right')
plt.title("roc")
plt.xlabel("fpr")
plt.xlabel("tpr")
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier(random_state = 0,criterion = 'entropy')
dtc.fit(x_train,y_train)
dtcpred=dtc.predict(x_test)
dtcfpr,dtctpr,threshold = metrics.roc_curve(y_test,dtcpred)
dtcroc_auc = metrics.auc(dtcfpr,dtctpr)
import matplotlib.pyplot as plt
plt.plot(dtcfpr,dtctpr,"b",label = "auc = %0.2f"%dtcroc_auc)
plt.legend(loc = "lower right")
plt.title("roc")
plt.xlabel("fpr")
plt.ylabel("tpr")
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(x_train,y_train)
lrpred = lr.predict(x_test)
import sklearn.metrics as metrics
lrfpr,lrtpr,threshold = metrics.roc_curve(y_test,lrpred)
lrroc_auc = metrics.auc(lrfpr,lrtpr)
import matplotlib.pyplot as plt
plt.plot(lrfpr,lrtpr,"b",label = "auc = %0.2f"%lrroc_auc)
plt.legend(loc = "lower right")
plt.title("roc")
plt.xlabel("fpr")
plt.ylabel("tpr")
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 5, metric = "euclidean")
knn.fit(x_train,y_train)
```

```
knnpred = knn.predict(x_test)
import sklearn.metrics as metrics
knnfpr,knntpr,threshold = metrics.roc_curve(y_test,knnpred)
knnroc_auc = metrics.auc(knnfpr,knntpr)
import matplotlib.pyplot as plt
plt.plot(knnfpr,knntpr,"b",label = "auc = %0.2f"%knnroc_auc)
plt.legend(loc = "lower right")
plt.title("roc")
plt.xlabel("fpr")
plt.ylabel("tpr")
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(x_train,y_train)
nbpred = nb.predict(x_test)
import sklearn.metrics as metrics
nbfpr,nbtpr ,threshold = metrics.roc_curve(y_test,nbpred)
nbroc_auc = metrics.auc(nbfpr,nbtpr)
import matplotlib.pyplot as plt
plt.plot(nbfpr,nbtpr,label = "auc = %0.2f"%nbroc_auc)
plt.legend(loc = "lower right")
plt.title("roc")
plt.xlabel("fpr")
plt.ylabel("tpr")
from sklearn.svm import SVC
svm = SVC(kernel="linear",random_state=0)
svm.fit(x_train,y_train)
svmpred=svm.predict(x_test)
import sklearn.metrics as metrics
svmfpr,svmtpr ,threshold = metrics.roc_curve(y_test,svmpred)
svmroc_auc = metrics.auc(svmfpr,svmtpr)
import matplotlib.pyplot as plt
plt.plot(svmfpr,svmtpr,label = "auc = %0.2f"%svmroc_auc)
plt.legend(loc = "lower right")
plt.title("roc")
plt.xlabel("fpr")
plt.ylabel("tpr")
```

app1.py

```
from flask import Flask,render_template , request
app=Flask(__name__)

import pickle
model=pickle.load(open('exit.pkl','rb'))
scaler=pickle.load(open('scaler.pkl','rb'))
@app.route('/')
def helloworld():
    return render_template("tcp.html")
@app.route('/login',methods = ['POST'])
def admin():

    q=request.form['cs']
    s=request.form['s']
    if (s=="France"):
        s1,s2,s3=1,0,0
    if(s=='Germany'):
        s1,s2,s3=0,1,0
    if(s=='Spain'):
        s1,s2,s3=0,0,1
    g=request.form['g']
    if(g=="Male"):
        g1=1
    if(g=="Female"):
        g1=0
    r=request.form['age']
    a=request.form['t']
    b=request.form['b']
    c=request.form['p']
    d=request.form['c']
    if(d=="No"):
        d1=0
    if(d=="Yes"):
```

```
        d1=1
    e=request.form['ac']
    if(e=="No"):
        e1=0
    if(e=="Yes"):
        e1=1
    f=request.form['sal']
    t=[[int(s1),int(s2),int(s3),int(q),int(g1),int(r),int(a),int(b),int(c),int(d1),int(e1),int(f)]]
    y=model.predict(scaler.transform(t))
    index=["No","Yes"]
    o=index[y[0]]
    return render_template("tcp.html", y="Customer will exit ? :"+o)


if __name__== '__main__':
    app.run(debug=True)
```

tcp.html

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
  font-family: Arial;
  color: white;
}

.split {
  height: 100%;
  width: 50%;
  position: fixed;
  z-index: 1;
  top: 0;
  overflow-x: hidden;
```

```
  padding-top: 20px;
}

.left {
  left: 0;
  background:url({{ url_for ('static', filename = 'pic500.jpg') }});
}

.right {
  right: 0;
  background: url({{ url_for ('static', filename = 'finalpic1.jpg') }});
 background-size: 100% 100%;

}

.split right
{
background:url({{ url_for ('static', filename = 'finalpic1.jpg') }});
}
</style>
</head>
<body>

<div class="split left">
  <div class="container">
       <div id="content" style="margin-top:2em">


    <br><br><br>
    <h2  style="color:white;"> <center>About</center></h2>
    <p  style="color:white;font-family:Arial">
Telecommunication, science and practice of transmitting information by
electromagnetic means.
Modern telecommunication centres on the problems involved in transmitting large
volumes of information over
long distances without damaging loss due to noise and interference. The basic
components of a modern digital
```

telecommunications system must be capable of transmitting voice, data, radio, and television signals.
Digital transmission is employed in order to achieve high reliability and because the cost of digital switching
systems is much lower than the cost of analog systems. In order to use digital transmission, however,
the analog signals that make up most voice, radio, and television communication must be subjected
to a process of analog-to-digital conversion.

Churn prediction is customer defection by predicting which customers are likely to cancel a subscription to a service.

Though originally used within the telecommunications industry, it has become common practice across banks, ISPs, insurance firms.
To analyse customer behaviour,following factors to be analysed

Customer demographic data (age, marital status, etc.)
Sentiment analysis of social media
Customer usage patterns and geographical usage trends
Calling-circle data

<form action="/login" method="post">

<div class="split right"; style="background: url({{ url_for ('static', filename = 'finalpic1.jpg') }}) ;background-size:100% 100%; ">

<h2>Telecom Churn Prediction</h2>

<h1 style="color:#F5F5F5;font-size=150% ">Details Required</h1>
<font size="5"
font-family: "Times New Roman","bold";

```
        color="white">
<p style="text-align:center;">

<b>Credit Score:</b><input type="number" name="cs"><br><br>
<label for="Geography"><b>Geography</b></label>
<select name="s">
<option values="--Select--">---                Select                ---</option>
<option values="France">France</option>
<option values="Germany">Germany</option>
<option values="Spain">Spain</option>
</select>
<br>
<br>

<b>Age:</b><input type="number" name="age"><br><br>
<td><b>Gender:</b></td>

<input type="radio"  name="g" value="Male" />
<lable for="Male"><b>Male</b></label>
<input type="radio" name="g" value="Female" />
<b>Female</b>

<br><br>
<b>Tenure:</b><input type="number" name="t"><br><br>
<b>Balance:</b><input type="number" name="b"><br><br>
<b>No.of Products:<b/><input type="number" name="p"><br><br>
<tr>
<td><b>Do you have a Credit Card:</b> </td>
<td>
<input type="radio" name="c" value="Yes" />
Yes
<input type="radio" name="c" value="No" />
No
</td>
</tr>
<br><br>
<tr>
```

```
<td><b>Are you an Active Member:</b> </td>
<td>
<input type="radio" name="ac" value="Yes" />
Yes
<input type="radio" name="ac" value="No" />
No
</td>
</tr>
<br><br>
<b>Estimated Salary:</b><input type="number" name="sal"><br>
</font>
</p>
<tr>
<td colspan="4" align="center"><input type="submit" value="Submit"/>
<input type="reset" value="Reset Data"/>
</td>
</tr>
</form>
</center>
<style>
form{padding:20px;
text-align : center;
font-size : 20px;}
Gender {width : 100px;
height : 30px;
fontsize : 20px;}

input[type=submit]
{
width:20%;
cursor:pointer;
font-size:20px;
font-weight:bold;
}
select{
padding:10px;
width:15%;
```

```
border-radius:5px;
}
input[type=reset]
{
width:20%;
cursor:pointer;
font-size:20px;
font-weight:bold;
}
</style>
  </center>
  <font size="6"
        font-family: "Times New Roman","bold";
        color="white">
<p>{{y}}</p>
</div>

</body>
</html>
```