# INTRODUCTION

* 1. Overview

A brief description about your project

* 1. Purpose

The use of this project. What can be achieved using this.

1. **Problem Definition & Design Thinking**
   1. Empathy Map

Paste the empathy map screenshot

* 1. Ideation & Brainstorming Map

Paste the Ideation & brainstorming map screenshot

# RESULT

Final findings (Output) of the project along with screenshots.

# ADVANTAGES & DISADVANTAGES

List of advantages and disadvantages of the proposed solution

# APPLICATIONS

The areas where this solution can be applied

# CONCLUSION

Conclusion summarizing the entire work and findings.

# FUTURE SCOPE

Enhancements that can be made in the future.

# APPENDIX

A. Source Code

Attach the code for the solution built.

# 1 INTRODUCTION

* 1. **Overview**

Customer churn is often referred to as customer attrition, or customer defection which is the rate at which the customers are lost. 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. Looking at churn, different reasons trigger customers to terminate their contracts, for example better price offers, more interesting packages, bad service experiences or change of customers’ personal situations.

Customer churn has become highly important for companies because of increasing competition amongcompanies, increased importance of marketing strategies and conscious behaviour of customers in the recent years. Customers can easily trend toward alternative services. Companies must develop various strategies to prevent these possible trends, depending on the services they provide. During the estimation of possible churns, data from the previous churns might be used. An efficient churn predictive model benefits companies in many ways. Early identification of customers likely to leave may help to build cost effective ways in marketing strategies. Customer retention campaigns might be limited to selected customers but it should cover most of the customer. Incorrect predictions could result in a company losing profits because of the discounts offered to continuous subscribers.

Telecommunication industry always suffers from a very high churn rates when one industry offers a better plan than the previous there is a high possibility of the customer churning from the present due to a better plan in such a scenario it is very difficult to avoid losses but through prediction we can keep it to a minimal level.

Telecom companies often use customer churn as a key business metrics to predict the number of customers that will leave a telecom service provider. A machine learning model can be used to identity the probable churn customers and then makes the necessary business decisions.

**1.2 Purpose**

As mentioned above, due to the digitalization and the access to information in large proportions, companies are under constant threats of customer churning, in particular subscription-based service providers that rely on such incomes. Unlike the business-to-customer domain, business to business characteristics, such as higher transactional value of each customer, constitutes a greater negative impact on the revenue stream when customers churn . Hence, it is important to study churn prediction and to support companies in this problem. The purpose of our study is to contribute knowledge in the field of churn prediction in a B2B subscription based service context. In this study we employ different machine learning techniques, in order to investigate the effects on the performance of churn prediction models in the B2B subscription-based service context. In particular, the aim is to propose and evaluate different approaches and techniques of how machine learning can be used to predict churn; hence creating value for businesses offering subscription-based services. The objectives of this study are to:

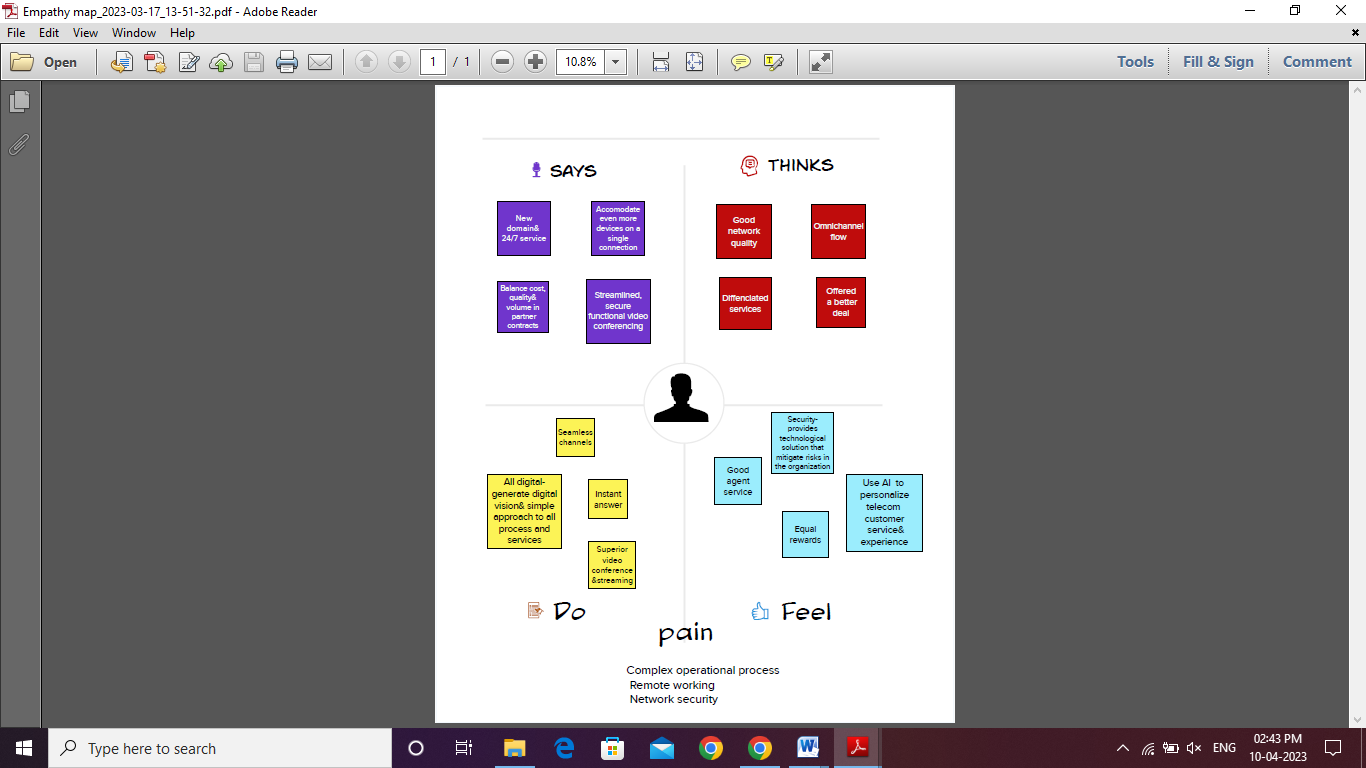
i) construct a binary churn prediction model,

ii) compare different types of algorithms for the built model, and

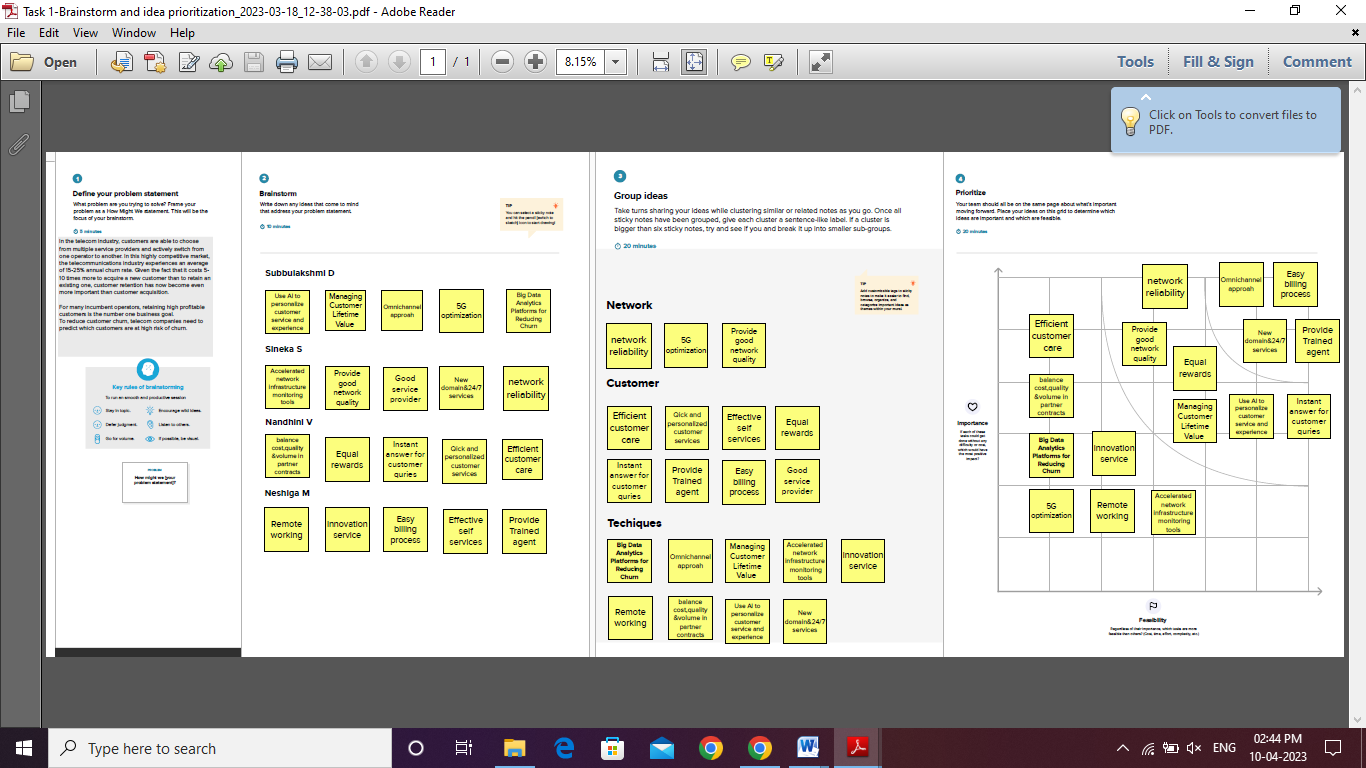
iii) study the effects of balancing the training dataset for the considered model

**2 Problem Definition & Design Thinking**

**2.1 Empathy map**



* 1. **Ideation & Brainstorming Map**



# RESULT

# E:\Pictures\Screenshots\Screenshot (496).png

# E:\Pictures\Screenshots\Screenshot (497).png

# E:\Pictures\Screenshots\Screenshot (498).png

# E:\Pictures\Screenshots\Screenshot (499).png

# ADVANTAGES & DISADVANTAGES

***“Communications technology is a crucial part of both*today’s*and tomorrow’s society.”***

In this system, we use various algorithms like Random Forest, XGBoost & Logistic Regression to find accurate values and which helps us to predict the churn of the customer. Here we implement the model by having a dataset that is trained and tested, which makes us have maximum correct values. In the Initial step, data preprocessing is performed in which we do filtering data and convert data into a similar form, and then we make feature selection.

In the further step prediction and classification is done using the algorithms like Random Forest, XGBoost, Logistic Regression(LR). Training and testing the model with the data set, we observe the behavior of the customer and analyze them. In the final step, we do analysis based on the results obtained and predict the customer churn.

**Advantages of Telecom customer churn prediction :**

* Quick and accessible communication
* Lack of time period
* Saves time
* Saves gasoline (do not need to drive distance)
* More than two people can communicate with at least one another at an equivalent time
* Next “best thing” to being there
* Easy to exchange ideas and knowledge via phone and/or fax
* Worldwide access
* Easy access to the people you would like to contact.
* Less effort in using transportation just to satisfy a private personally.
* You can just occupy your home and use a telephone or a cellphone if you would like to speak to someone.
* Enable end-users to speak electronically and share hardware, software, and data resources.
* This make corporation to do the transaction at the point only and in a very fast way from many remote locations, exchange business documents electronically with customers and suppliers, or remotely monitor and control production processes.
* Interconnect the pc systems of a business so their computing power is often shared by end-users throughout an enterprise.
* Make the organization work with collaboration and communication among the staff inside and out of doors a corporation.
* Speed
* Develops new products and inventions

**Disadvantages of Telecom customer churn prediction:**

* Cultural Barrier
* Misunderstanding
* Prank calls
* Sometimes expensive
* High electric bills
* Remote areas don’t have access
* Remote areas might not be ready to afford the necessary equipment
* Cannot see whom you’re speaking with
* Cannot see facial expressions, therefore results in misunderstandings
* Cultural barriers
* Poor connections or downed power lines during/after storms

1. **APPLICATIONS**

Customer churn is a common problem across businesses in many sectors. If you want to grow as a company, **you have to invest in acquiring new clients**. Every time a client leaves, it represents a significant investment lost. Both time and effort need to be channelled into replacing them. Being able to predict when a client is likely to leave, and offer them incentives to stay, can offer huge savings to a business. As a result, understanding what keeps customers engaged is extremely valuable knowledge, as it can help you to develop your retention strategies, and to roll out operational practices aimed at keeping customers from walking out the door. Predicting churn is a fact of life for any subscription business, and even slight fluctuations in churn can have a significant impact on your bottom line.

# 6 CONCLUSION

The importance of this type of research in the telecom market is to help companies make more profit. It has become known that predicting churn is one of the most important sources of income to telecom companies. Hence, this research aimed to build a system that predicts the churn of customers in SyriaTel telecom company. These prediction models need to achieve high AUC values. To test and train the model, the sample data is divided into 70% for training and 30% for testing. We chose to perform cross-validation with 10-folds for validation and hyperparameter optimization. We have applied feature engineering, effective feature transformation and selection approach to make the features ready for machine learning algorithms. In addition, we encountered another problem: the data was not balanced. Only about 5% of the entries represent customers’ churn. This problem was solved by under sampling or using trees algorithms not affected by this problem. Four tree based algorithms were chosen because of their diversity and applicability in this type of prediction. These algorithms are Decision Tree, Random Forest, GBM tree algorithm, and XGBOOST algorithm. The method of preparation and selection of features and entering the mobile social network features had the biggest impact on the success of this model, since the value of AUC in SyriaTel reached 93.301%. XGBOOST tree model achieved the best results in all measurements. The AUC value was 93.301%. The GBM algorithm comes in the second place and the random forest and Decision Tree came third and fourth regarding AUC values. We have evaluated the models by fitting a new dataset related to different periods and without any proactive action from marketing, XGBOOST also gave the best result with 89% AUC. The decrease in result could be due to the non-stationary data model phenomenon, so the model needs training each period of time.

# 7 FUTURE SCOPE

Some features such as Contract ID, MSISDN and other unique features for all customers were removed. They are not used in the training process because they have a direct correlation with the target output (specific to the customer itself). We deleted features with identical values or missing values, deleted duplicated features, and features that have few numeric values. We found that more than half of the features have more than 98% of missing values. We tried to delete all features that have at least one null value, but this method gave bad results.

Finally, we filled out the missing values with other values derived from either the same features or other features. This method is preferable so that it enables us to use the information in most features for the training process. We applied the following:

* Records that contain more than 90% of missing features were deleted.
* Features that have more than 70% of missing values were deleted.
* For the missing categories in categorical features, they were replaced by a new category called ‘Other’.
* The missing numerical values were replaced with the average of the feature.
* The number of categorical features were 78, the first 31 most frequent categories were chosen and the remaining categories were replaced with a new category, so the total number is 32 categories.
* There are some other features with a numeric character but they contain only a limited number of duplicate values in more than one record. This indicates that they are categorical so we have dealt with them as categorical features, but the experiment shows that they perform worse with the model, so that they have been deleted.
* We have also calculated the correlation between numerical features using Pearson and removed the correlated features. This removal had no effect on the final result. Many other methods were tested, but this applied approach gave the best performance of the four algorithms. The number of features after this operation exceeded 2000 features at the end.

# 8 APPENDIX

#import necessary libraries

import pandas as pd

import numpy as np

import pickle

import matplotlib. pyplot as plt

%matplotlib inline

import seaborn as sns

import sklearn

from sklearn.preprocessing import LabelEncoder, OneHotEncoder

from sklearn.linear\_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.svm import SVC

from sklearn.model\_selection import RandomizedSearchCV

import imblearn

from imblearn.over\_sampling import SMOTE

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import accuracy\_score,classification\_report, confusion\_matrix,

f1\_score

data=pd.read\_csv(r"/Churn\_Modelling.csv")

data

#checking for null values

data.TotalCharges= pd.to\_numeric(data.TotalCharges,errors= 'coerce')

data.isnull().any ()

data["TotalCharges"].fillna(data["TotalCharges"].median() ,inplace =True)

data.isnull().sum()

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

data["gender"] = le.fit\_transform(data["gender"])

data["Partner"] = le.fit\_transform(data["Partner"])

data["Dependents"] = le.fit\_transform(data["Dependents"])

data["PhoneService"] = le.fit\_transform(data[ "PhoneService"])

data["MultipleLines"] = le.fit\_transform(data["Multiplelines"])

data["InternetService"] = le.fit\_transform(data["InternetService"])

data["OnlineSecurity"] = le.fit\_transforn(data["OnlineSecurity"])

data["onl ineBackup"] = le.fit\_transform (data["OnlineBackup"])

data["DeviceProtection"] = le.fit\_transform(data["DeviceProtection"])

data["TechSupport"] = le.fit\_transform(data["TechSupport"])

data["StreamingTV"] = le.fit\_transform(data["StreamingTV"])

data["StreamingMovies"] = le.fit\_transform(data["StreamingMovies"])

data["Contract"] = le.fit\_transform(data["Contract"])

data["PaperlessBilling"] = le.fit\_transform(data["PaperlessBilling"])

data["PaymentMethod"] = le.fit\_transform(data["PaymentMethod"])

data["Churn"] = le.fit\_transfora (data["Churn"])

data.head()

x=data.iloc[:,0:19].values

y= data.iloc[:,19:20].values

from sklearn.preprocessing import OnetHotEncoder

one = OnetHotEncoder()

a= one.fit\_transform(x[:,6:7]).toarray()

b= one. fit\_transform(x[:,7:8]) .toarray()

c= one.fit\_transform(x[:,8:9]).toarray()

d= one.fit\_transform(x[:,9:10]).toarray ()

e=one.fit\_transform(x[:,10:11]). toarray()

f=one.fit\_transform(x[:,11:12]).toarray()

g=one.fit\_transform(x[:,12:13]).toarray()

h=one.fit\_transform(x[:,13:14]). toarray()

i=one. fit\_transform(x[:,14:15]).toarray()

j=one.fit\_transform(x[:,16:17]).toarray()

x=np.delete (x, [6,7, 8, 9, 10, 11, 12, 13, 14,16], axis=1)

x=np.concatenate((a,b, c,d, e, f,8, h, i,j,x), axis=1)

from imblearn.over\_sampling import SMOTE

smt=SMOTE()

x\_resample,y\_resample=smt.fit\_resample(x,y)

x\_resample

y\_resample

x.shape,x\_resample.shape

y.shape,y\_resample.shape

data.describe()

plt.figure (figsize=(12,5))

plt.subplot (1,2,1)

sns.distplot (data["tenure"])

plt.subplot (1,2,2)

sns.distplot (data["MonthlyCharges"])

plt.figure (figsize=(12,5))

plt.subplot(1,2,1)

sns.countplot(data["gender"])

plt.subplot(1,2,2)

sns.countplot(data["Dependents"])

sns.barplot(x="Churn",y="MonthlyCharges",data=data

sns.heatmap(data.corr(),annot=True)

sns.pairplot(data=data,markers=["^","v"],palette="inferno")

from sklearn.model\_selection import train\_test\_spit

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x\_resample,y\_resample,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)

#importing and building the Decision tree model

def logreg(x\_train,x\_test,y\_train,y\_test):

lr=LogisticRegression(random\_state=0)

lr.fit(x\_train,y\_train)

y\_lr\_tr= lr.predict(x\_train)

print(accuracy\_score(y\_lr\_ tr,y\_train))

yPred\_lr=lr.predict(x\_test)

print(accuracy\_score(yPred\_lr,y\_test))

print("\*\*\*Logistic Regression\*\*\*")

print("Confusion\_Matrix")

print(confusion\_matrix(y\_test,yPred\_lr))

print("Classification Report")

print(classification\_report(y\_test,yPred\_lr))

logreg(x\_train,x\_test,y\_train,y\_test)

#importing and building the Decision tree model

def decisionTree(x\_train,x\_test,y\_train,y\_test) :

  dtc=DecisionTreeClassifier(criterion="entropy", random\_state=0)

  dtc.fit(x\_train,y\_train)

  y\_dt\_tr=dtc.predict(x\_train)

  print(accuracy\_score(y\_dt\_tr,y\_train))

  yPred\_dt=dtc.predict(x \_test)

  print(accuracy\_score(yPred\_dt,y\_test))

  print ("\*\*\*Decision Tree\*\*\*")

  print ("Confusion\_Matrix")

  print (confusion\_matrix(y\_test,yPred\_dt))

  print("classification Report")

  print(classification\_report(y\_test,yPred\_dt))

#printing the train accuracy and test accuracy respectively

decisionrTree(x\_train,x\_test,y\_train,y\_test)

#importing and bui lding the random forest model

def RandomForest (x\_train, x\_test, y\_train,y\_test):

  rf=RandomForestclassifier(criterion="entropy",n\_estimators=10,random\_state=0)

  rf.fit(x\_train,y\_train)

  y\_rf\_tr = rf.predict(x\_train)

  print(accuracy\_score(y\_rf\_tr,y\_train))

  yPred\_rf = rf.predict(x\_test)

  print (accuracy\_score(yPred\_rf, y\_test))

  print("\*\*\*Random Forest\*\*\*")

  print("Confusion\_Matrix")

  print(confusion\_matrix(y\_test, yPred\_rf))

  print("Classification  Report")

  print(classification\_report(y\_test,yPred\_rf))

  #printing the train accuracy and test accuracy respectively

  RandomForest (x\_train,x\_test,y\_train,y\_test)

#importing and building the KNN model

def KNN(x\_train, x\_test,y\_train,y\_test) :

  knn = KNeighborsClassifier()

  knn.fit(x\_train,y\_train)

  y\_knn\_tr = knn.predict(x\_train)

  print(accuracy\_score (y\_knn\_tr,y\_train) )

  yPred\_knn = knn.predict(x\_test)

  print(accuracy\_score (yPred\_knn,y\_test))

  print("\*\*\*KNN\*\*\*")

  print("Confusion\_Matrix")

  print(confusion\_matrix(y\_test, yPred\_knn))

  print("Classification Report")

  print(classification\_report(y\_test,yPred\_knn))

#printing the train accuracy and test accuracy respectively

KNN (X\_train, x\_test,y\_train,y\_test)

#importing and building the random forest model

def svm(x\_tarin,x\_test,y\_train,y\_test):

  svm = SVC(kernel ="linear")

  svm.fit(x\_train,y\_train)

  y\_svm\_tr = svm. predict (x\_train)

  print(accuracy\_score(y\_svm\_tr,y\_train))

  yPred\_svm = svm. predict (x\_test)

  print(accuracy\_score(yPred\_svm,y\_test))

  print(""\*\*\*Support Vector Machine\*\*\*")

  print("Confusion\_Matrix")

  print(confusion\_matrix(y\_test,yPred\_svm))

  print("Classification Report")

  print(classification\_report(y\_test,yPred\_svm))

#printing the train accuracy and test accuracy respectively

svm(x\_train, x\_test,y\_train,y\_test)

from flask import flask, render\_template, request

import keras

from keras.models import load\_model

app = flask(\_name\_)

model = load\_model("telecom\_churn.h5")

@app.rute('/') # rendering the html template

def home():

return render\_template('home.html')

@app.route('/')

def helloworld():

return render render\_template("base.html")

@app.route('\assement')

def prediction():

return render render\_template(index.html")

@app.route('/predict', methods=['POST'])

def admin():

a=request.form["gender"]

if(a== 'f'):

a=0

if(a== 'm'):

a=1

b=request.form["srcitizen"]

if(b== 'n'):

b=0

if(a== 'y'):

b=1

c=request.form[" partner"]

if(c== 'n'):

c=0

if(c== 'y'):

c=1

d=request.form[" dependents"]

if(c== 'n'):

d=0

if(d== 'y'):

d=1

e=request.form[" tenure"]

d=request.form["phservices"]

if(f== 'n'):

f=0

if(f== 'y'):

f=1

e=request.form["multi"]

if(g== 'n'):

import keras

from keras.models import sequential

from keras.layers import Dense

classifier=Sequential()

classifier.add(Dence(

q=request.form{“plb”]

if (q==’n’):

q=0

if(q==’y’):

q=1

r=request.form[‘mcharges”]

s=request.form[“tcharges’]

t=[[int(g1),int(g2),int(g3),int(h1),int(h2),int(h3),int(i),int(i2),int(i3),int(j1),int(j2),int(j3)

print(t)

x=model.predict(t)

print(x[0])

if(x[[0]]<=0.5):

y=’no’

return render\_template(“predno.html”,z=y)

if(x[[0]]>=0.5)L

y=”Yes”

return render\_template(“predyes.html”,z=y)

l1, 12, 13=1,0,0

if (1 == 'nis'):

l1, 12, 13=0,1,0

if (1 == 'y):

l1,12, 13=0,0,1

m= request.form[ "stv"]

if (m =='n'):

m1, m2, m3=1,0,0

if (m =='nis'):

m1, m2, m3=0,1,o

if (m == 'y'):

m1, m2, m3=0,0,1

request.form["smv"]

if (n == 'n'):

n1, n2, n3=1,0,0

if (n =='nis') :

n1, n2, n3-0,1,0

if (n =='y'):

n1,n2, n3=0,O,1

O= request.form["contract"]

if (o== 'mtm):

01,02,03=1,0,0

if (o == oyr'):

o1,02 , o3=0,1,0

if (o == 'tyrs):

01,02, 03=0,0,1

p= request.form["pmt"]

if (p == 'ec'):

p1,p2, p3, p4=1,0, 0, 0

if (p == 'mail'):

p1, p2, p3, p4-0,1,0,0

if (p == ‘bt'):

p1, p2, p3, p4-0,0, 1,0

if (p == 'cc'):

p1, p2, p3, p4=0,0,0, 1

q= request. form[ "plb"]

if (g == "n'):

gl,g2,g3=1,0,0

if (g == 'nps'):

g1,g2,g3=0,1,o

if (g == 'y"):

g1,g2,g3-0,0,1

h= request. form[ "is"]

if (h == 'dsl '):

hi,h2, h3=1,0,o

if (h == 'fo'):

hi, h2, h3-0,1,o

if (h == 'n'):

h1,h2, h3-0,0,1

i= request.form[ "os"]

if (i == 'n'):

i1, i2, i3-1,0,o

if (i == 'nis'):

i1, i2, i3-0,1,0

if (i == 'y):

i1,i2, i3=0, e,1

j= request.form["ob" ]

if (Ở == 'n'):

j1,j2, j3=1,0,e

if (j == 'nis'):

j1,j2,j3-0,1,

if (j == 'y"):

j1,j2,j3-0,0, 1

k= request. form[ "dp"]

if (k == 'n):

k1,k2, k3=1,0,e

if (k == 'nis '):

k1,k2, k3=0,1,0

if (k == 'y'):

k1, k2, k3=0,0,1

l= request.form[ "ts"]

if (1 == 'n'):

11,12,l3=1,0,0