TELECOM CUSTOMER CHURN PREDICTION USING IBM WATSON

Submitted by:

1) Sanjai I, 20BIT0371, sanjai.i2020@vitstudent.ac.in
2) Pranav Sinha, 20BCE10217, pranav.sinha2020@vitbhopal.ac.in
3) P.Cherith Reddi, 20BIT0170,cherithreddi.p2020@vitstudent.ac.in
4) Adit Rajesh,20BCE2415, adit.rajesh2020@vitstudent.ac.in

INDEX OF THE PROJECT

Sl.no	TITLE	Page no
1.	INTRODUCTION	2
2.	LITERATURE SURVEY	4
3.	THEORETICAL ANALYSIS	7
4.	EXPERIMENTAL INVESTIGATIONS	8
5.	FLOWCHART	9
6.	RESULT	10
7.	ADVANTAGES & DISADVANTAGES	10
8.	APPLICATIONS	11
9.	CONCLUSION	11
10.	FUTURE SCOPE	11
11.	CODES	12
12.	BIBLIOGRAPHY	36

1. INTRODUCTION

1.1 Overview of the Project

In recent years, as business competition has increased, the significance of marketing techniques has grown, and customers' behaviour has gotten more aware, churn among customers has become a crucial issue for businesses. Alternative services can easily become more popular among consumers. Depending on the services they offer, businesses must build different prevention tactics for these potential trends. Data from the prior churns may be used to estimate potential churns. Companies can gain from an effective churn predicting model in a variety of ways. Early detection of clients who are likely to quit can aid in developing cost-effective marketing strategies. It is very difficult to avoid losses in the telecommunications industry because of the very high customer churn rates. However, through prediction, we can keep losses to a minimum. A machine learning model is created, which assists in identifying the customers who are likely to churn and then assists in making critical business decisions.

To identify customers who are likely to leave and take proactive actions to keep them, telecom companies must do the crucial work of telecom customer churn prediction. The two well-known machine learning methods utilized for this prediction are logistic regression and support vector machines (SVM). SVM seeks to identify an ideal hyperplane that, by increasing the margin between them, distinguishes churners from non-churners. It employs past customer data as input features and is capable of handling complex decision boundaries. Based on the input features, logistic regression calculates the probability of churn. Although it has a different name, it is a linear model that fits coefficients to establish the connection between characteristics and the likelihood of churn. SVM and logistic regression model performance can be evaluated using metrics like accuracy, precision, recall, and F1-score. Cross-validation methods can offer more reliable performance estimations. SVM and logistic regression models can assist telecom firms in predicting customer churn and implementing targeted retention tactics by utilizing previous customer data and appropriate input features. For the telecom business to continue to grow and remain competitive, proper model performance evaluation and monitoring are essential.

1.2 Purpose

Telecom customer churn prediction using Support Vector Machines (SVM) and Logistic Regression is a project intended at assisting telecom firms in proactively identifying consumers who are likely to abandon their services. Telecom firms may predict churn and take appropriate action to retain consumers, increase customer satisfaction, and promote company success by utilizing historical customer data and employing machine learning techniques, such as SVM and logistic regression. By precisely identifying clients at risk of churn, the project's main goal is to enable telecom businesses to reduce churn rates and maximise their resources. The ability to target high-value customers who are most likely to quit helps businesses allocate their retention efforts effectively. Telecom firms may solve pain areas, boost service quality, and improve the entire customer experience by knowing the churn-causing causes. Saving money is a major advantage of this project. In general, acquiring new clients is more expensive than keeping the ones you already have. Telecom firms can properly manage their resources and concentrate on keeping valuable clients by anticipating churn. This lowers the cost of customer acquisition and increases profitability.

Churn prediction models additionally enable targeted marketing strategies. Telecommunications firms can create individualized marketing strategies to re-engage client segments with a high tendency for turnover. Customized promotions, rewards, and communications can boost client loyalty and happiness, which will ultimately lead to revenue growth. Telecom companies can improve the performance of their businesses by deploying churn prediction algorithms. Companies can establish revenue stability, boost client loyalty, and gain a competitive advantage in the market by lowering turnover rates. Positive word-of-mouth recommendations and brand advocacy are more likely to result from satisfied and devoted customers recommending the products and services to others.

2. LITERATURE SURVEY

Sl.no	Title	Year	Summary
1.	A Review of Machine Learning Methods for Predicting Churn in the Telecom Sector	2023	In this study, there is a thorough evaluation of machine learning approaches used for telecom churn prediction. The authors perform a thorough review of the available literature and summarise various machine learning algorithms utilized in the telecoms industry for churn prediction. They're probably going to talk about popular algorithms like logistic regression, decision trees, random forests, support vector machines, neural networks, and ensemble approaches. Key issues such as data pretreatment approaches, feature engineering, model selection, and evaluation metrics utilized in prior studies may be covered in the paper. It most likely shows the strengths and weaknesses of various machine learning systems and indicates the most promising methods for telecom churn prediction.[3]
2.	Customer Churn Prediction Using Machine Learning Approaches	2023	The authors of this paper investigate the prediction of customer turnover using machine learning methodologies. Their goal is to create an accurate model that can successfully identify clients who are in danger of churning. The article will most likely include topics such as selecting an appropriate dataset, preprocessing techniques used, and the use of several machine learning algorithms for churn prediction. The authors may investigate techniques such as decision trees, logistic regression, support vector machines, and ensemble methods, assessing their effectiveness in predicting customer attrition. The study's findings and insights can be useful to telecom businesses looking to deploy churn prevention tactics based on machine learning approaches.[5]
3.	Customer churn prediction in telecom using machine learning in big data platform	2019	This paper focuses on predicting customer turnover in the telecom industry using machine learning approaches within a big data environment. Using the power of big data analytics, the authors hope to

			create an effective model that can reliably forecast client attrition. The article will most likely explore the usage of various machine learning methods and approaches designed specifically for dealing with large-scale telecom datasets. The authors may investigate feature engineering, model training and evaluation, as well as the integration of big data technologies for effective churn prediction. The findings of this study can be used to improve client retention efforts in the telecom industry by utilizing machine learning and big data analytics.[6]
4.	Customer Churn Prediction in Telecom Sector with Machine Learning and Information Gain Filter Feature Selection Algorithms	2021	The fundamental contribution of the research is the creation of a churn prediction model that allows telecom operators to predict which customers are likely to churn. This work employs machine learning (ML) methodologies such as the Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), Random Forest (RF), and Naive Bayes (NB). This research provides a novel feature selection methodology that combines the Information Gain and Ranker methods. The accuracy, precision, and F-measure standard metrics, as well as 10-fold cross-validation, are used to evaluate the model's performance. When feature selection is considered, the accuracy is 95.02%, while without feature selection, the accuracy is 92.92%. When the findings were compared to the existing approaches, our models performed well.
5.	Churn Prediction Using Machine Learning and Recommendations Plans for Telecoms	2019	In this study, three machine learning algorithms Naive Bayes, SVM, and decision trees have been used to predict churn using two benchmark datasets, the cell2cell dataset and the IBM Watson dataset, each of which contains 71,047 observations and 57 attributes. The models' effectiveness was assessed using the area under the curve (AUC), and they received scores of 0.82, 0.87, and 0.77 for the IBM dataset and 0.98, 0.99, and 0.98 for the cell2cell dataset, respectively. Additionally, the proposed models outperformed earlier research on the same datasets in terms of accuracy.

2.1 Existing System

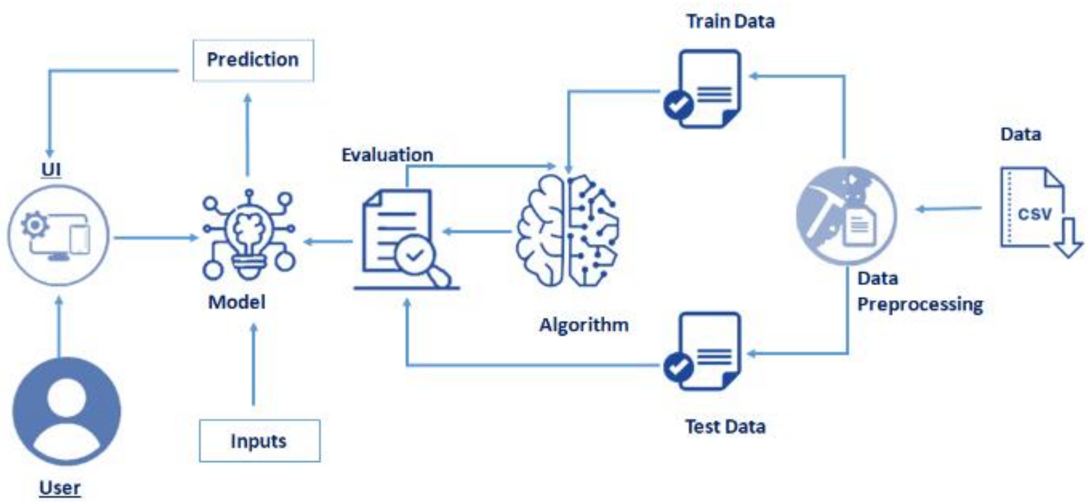
It has been possible to estimate customer churn using a variety of methods, including data mining, machine learning, and hybrid technologies. These methods make it possible for businesses to recognize, foresee, and keep churning customers. Additionally, they support CRM and decision-making in industries. The majority of them shared the usage of decision trees, which is one of the accepted techniques for determining customer turnover but is inappropriate for complicated issues [1]. However, the research demonstrates that decreasing the data enhances the decision tree's accuracy [2]. Data mining methods are sometimes used for historical analysis and consumer prediction. Regression tree techniques were addressed in conjunction with other widely used data mining techniques, including decision trees, rule-based learning, and neural networks [3].

2.2 Proposed System

To discover accurate values and anticipate customer turnover, we apply a variety of techniques in this system, including SVM and Logistic Regression. Here, we put the model into practice by using a dataset that has been trained and tested, giving us the highest number of accurate results. Data preprocessing is done in the first phase when we filter data and transform it into a similar shape before choosing features. Using algorithms like SVM and Logistic Regression (LR), prediction and classification are carried out in the following stage. After hyperparameter tuning, we observe and analyze consumer behaviour after training and testing the model using the data set. In the final step, we do an analysis based on the results obtained and predict the customer churn.

3. THEORETICAL ANALYSIS

3.1 Block Diagram



3.2 Hardware / Software Designing

HARDWARE REQUIREMENTS

1. Memory (RAM): A machine learning algorithm frequently needs a significant amount of RAM to store and handle data throughout the training and prediction stages, thus having enough RAM is crucial, especially when working with large datasets. Ensuring that the algorithms run smoothly and error-free depends on having enough RAM.

2. Processor (CPU): For large datasets and complicated models, a quick and powerful CPU can greatly speed up the training and prediction stages of machine learning algorithms. Additionally, parallel processing, which is frequently employed in machine learning applications, can be accelerated by multi-core computers.

3. Storage: Enough room to keep the trained models and the dataset.

4. Graphics Processing Unit (GPU): With large datasets in particular, a strong GPU may help train sophisticated machine learning models more quickly.

5. Network Connection: A steady and dependable internet connection is required for data upload, model training, and prediction when employing cloud-based solutions. For effective data transfer and distant access to cloud-based resources, high-speed internet may be necessary.

SOFTWARE REQUIREMENTS

1. Operating System: A compatible operating system, such as Windows, macOS, or Linux, depending on the preference and familiarity of the user.

2. Integrated Development Environment (IDE): An IDE, such as Python IDEs like PyCharm, and Jupyter Notebook, or R IDEs like RStudio, for coding, running, and debugging machine learning algorithms.

3. Programming Language: Proficiency in a programming language such as Python or R, as these are commonly used for implementing machine learning algorithms.

4. Libraries and Packages: Relevant machine learning libraries and packages such as sci-kit-learn, TensorFlow, Keras depending on the algorithms being used.

5. Data Visualization Tools: Tools such as Matplotlib, Seaborn, or Plotly for visualizing the data and the results of the predictions.

6. Data Processing Tools: Tools such as Pandas, NumPy for data preprocessing, feature engineering, and data manipulation tasks.

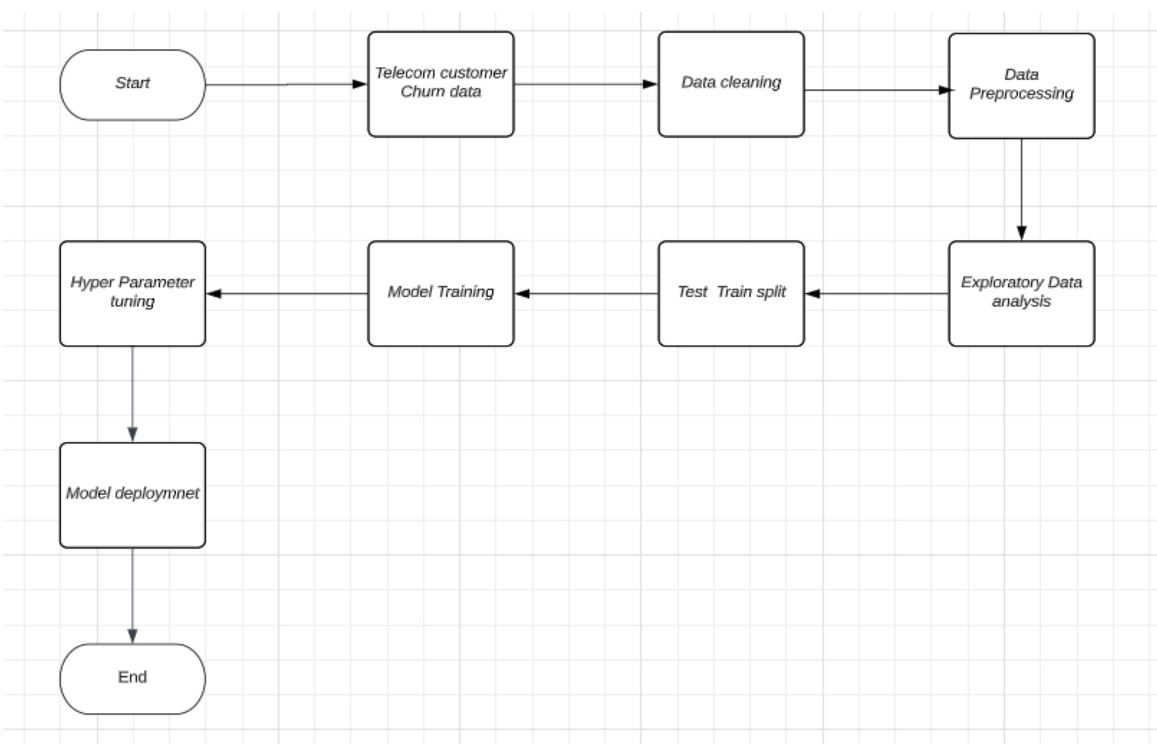
4. Experimental investigations

We will first download the dataset from Kaggle after eliminating all of the null values through data filtering. Then, we transformed all the data into a format that was comparable and easier to comprehend and process. We attempt to develop a predictor model for the Telecom company using logistic regression and a unique technique. Here, we have a customer data set that we divide into training and testing using preprocessing and feature selection. To get more accurate and efficient outcomes from this algorithm, we made some feature engineering. We can estimate the likelihood of happening event sites and create a discriminative probabilistic classification with the use of logistic regression. The dependent variable indicates whether an event occurs (it will be one if it does, 0 otherwise). By training the model, the data will receive results with their specifics, and we will test the model with the remaining data. As a result, we will be able to estimate customer churn with accuracy and provide a clear warning about the customer. This will enable the business to take the necessary precautions to prevent losing existing customers to competitors.

Support Vector Machines (SVM): SVM is a classification-focused supervised learning technique. SVM may examine customer data in the context of churn prediction and produce a decision boundary that distinguishes churned customers from non-churned customers. By taking into account the support vectors, which are the data points located closest to the decision boundary, SVM seeks to identify the ideal hyperplane that maximally divides the two classes.

Another extensively used method for tasks requiring binary classification is **logistic regression**. The logistic function is used to represent the relationship between the independent variables (client attributes) and the binary outcome (churn or not churn). Logistic regression determines the risk of turnover for each client by estimating the coefficients of the independent factors and gives a predicted class.

5. FLOWCHART



6. RESULT

7. ADVANTAGES & DISADVANTAGES

Advantages:

6. Enhanced Model Performance: By normalizing the numerical variables, the solution ensures that all features are on a similar scale, which can lead to better model performance and convergence.
7. Flexibility and Versatility: The solution employs multiple machine learning algorithms like Naive Bayes, Random Forest, SVM, Logistic Regression, and XGBoost. This allows for flexibility in selecting the most suitable algorithm for the problem at hand.
8. Hyperparameter Tuning: The inclusion of hyperparameter tuning using GridSearchCV helps in finding the optimal hyperparameters for the chosen algorithms, potentially improving their performance.
9. Evaluation Metrics: The solution provides evaluation metrics like accuracy, precision, recall, and F1-score, which give insights into the model's performance and can help in understanding its strengths and weaknesses.

Disadvantages:

- Time and Computational Resources: Some preprocessing steps like encoding and feature engineering can increase the dimensionality of the data, leading to increased computational requirements and longer training times, especially when dealing with large datasets.
- Model Selection Bias: The choice of machine learning algorithms in the solution may introduce bias and limitations. Different algorithms have different strengths and weaknesses, and relying solely on a predefined set of algorithms may not always yield the best results.
- Overfitting Risk: Without proper regularization or hyperparameter tuning, some algorithms, such as decision trees, SVM, and XGBoost, can be prone to overfitting, where the model performs well on the training data but poorly on unseen data.

8. APPLICATIONS

- Customer Retention: Spot clients who are at risk of leaving and take aggressive measures to keep them.
- Marketing campaign optimization: For targeted marketing activities, target client groupings with a high propensity for turnover.
- Service Quality Improvement: Address pain areas and improve the entire customer experience by using churn analytics.
- For specialized retention efforts, segment clients based on their propensity to leave.
- Resource Allocation: Concentrate retention efforts on clients who are at a high risk of leaving.
- Business forecasting: Examine churn patterns to produce more accurate projections and wiser choices.

9. CONCLUSION

The significance of churn prediction will assist numerous businesses, particularly those in the telecom sectors, in achieving a healthy income and strong revenue. In the telecom industry, customer churn forecast is a big problem, thus businesses are focusing more on retaining their current clientele than on attracting new ones. In comparison to previous techniques, we will get greater accuracy by applying SVM and logistic regression. Here, we are using a dataset of a few customers' service plans to assess their values and make a precise prediction that would aid in identifying the clients who will switch to other services offered by the business. By doing this, the telecom company can see everything clearly and offer them some tempting incentives to continue using the service. The findings gained demonstrated that our suggested churn model performed better and delivered better results when applying machine learning techniques.

10. FUTURE SCOPE

The future churn for predicting telecom customer churn using SVM and logistic regression includes developments like more sophisticated machine learning methods, incorporating big data and AI, real-time prediction, predictive analytics for customer lifetime value, customer segmentation refinement, upselling and cross-selling opportunities, explainable AI, and addressing ethical issues. These developments are meant to boost client retention efforts, increase the accuracy of churn prediction, and improve business success in the telecom sector.

11.CODES

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns


"""LOADING DATASET"""


data=pd.read_csv("DataSet.csv")


"""DATA ANALYTICS"""


data.info()


data.head()


data['TotalCharges']=pd.to_numeric(data['TotalCharges'],errors='coerce')


data.head()


data.describe()


data.isnull().any()
```

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

"""EXPLORATORY DATA ANALYSIS"""

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

import matplotlib.pyplot as plt
plt.hist(data['tenure'], bins=10)
plt.xlabel('Tenure')
plt.ylabel('Frequency')
plt.title('Histogram of Tenure')
plt.show()

plt.scatter(data['MonthlyCharges'], data['TotalCharges'])
plt.xlabel('Monthly Charges')
plt.ylabel('Total Charges')
plt.title('Scatter Plot of Monthly vs Total Charges')
plt.show()

sns.boxplot(x=data['Churn'], y=data['tenure'])
plt.xlabel('Churn')
plt.ylabel('Tenure')
```

```
plt.title('Box Plot of Tenure by Churn')

plt.show()


#multi-variate

sns.pairplot(data[['tenure', 'MonthlyCharges', 'TotalCharges']])

plt.title('Pairwise Relationships between Numerical Variables')

plt.show()


corr_matrix = data[['tenure', 'MonthlyCharges', 'TotalCharges']].corr()

sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')

plt.title('Correlation Matrix')

plt.show()


plt.pie(data['InternetService'].value_counts(),                labels=data['InternetService'].unique(),
autopct='%1.1f%%')

plt.title('Distribution of Internet Service')

plt.show()


table = pd.crosstab(data['PaymentMethod'], data['Churn'])

table.plot(kind='bar', stacked=True)

plt.xlabel('Payment Method')

plt.ylabel('Count')

plt.title('Stacked Bar Plot of Churn by Payment Method')
```

```
plt.legend(title='Churn')

plt.show()


sns.boxplot(x=data['Contract'], y=data['MonthlyCharges'])

plt.xlabel('Contract Type')

plt.ylabel('Monthly Charges')

plt.title('Box Plot of Monthly Charges by Contract Type')

plt.show()


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




""" ENCODING"""


data.corr()


data['gender'].unique()


data['Partner'].unique()


data['Dependents'].unique()
```

```
data['Dependents'].unique()

data['PhoneService'].unique()

data['MultipleLines'].unique()

data['InternetService'].unique()

data['OnlineSecurity'].unique()

data['OnlineBackup'].unique()

data['DeviceProtection'].unique()

data['TechSupport'].unique()

data['StreamingTV'].unique()

data['StreamingMovies'].unique()

data['Contract'].unique()

data['PaperlessBilling'].unique()
```

```
data['PaymentMethod'].unique()

data['MonthlyCharges'].unique()

data['Churn'].unique()

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['MultipleLines']=le.fit_transform(data['MultipleLines'])
data['PhoneService']=le.fit_transform(data['PhoneService'])
data['InternetService']=le.fit_transform(data['InternetService'])
data['OnlineSecurity']=le.fit_transform(data['OnlineSecurity'])
data['OnlineBackup']=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_transform(data['Churn'])

data

data.drop(columns='customerID', inplace=True)

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

data['Churn'].value_counts()

x.shape

y=data.iloc[:,19:20].values

y

from sklearn.preprocessing import OneHotEncoder
one=OneHotEncoder()
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,g,h,i,j,x), axis=1)


x.shape


from imblearn.over_sampling import SMOTE


smt = SMOTE()


x_resample, y_resample = smt.fit_resample(x,y)


x_resample.shape


y_resample
```

```
data

x_resample.shape

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x_resample,y_resample,test_size = 0.2, random_state=0)

print(x_train.shape)

print(x_test.shape)

print(y_train.shape)

print(y_test.shape)

print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)

from sklearn.preprocessing import StandardScaler
```

```
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.fit_transform(x_test)

x_train

x_test



"""MODEL BUILDING"""

from sklearn.svm import SVC
svm=SVC(kernel='linear')
svm.fit(x_train,y_train)

svm_pred=svm.predict(x_test)
svm_pred

from sklearn.metrics import accuracy_score
svm_acc=accuracy_score(svm_pred,y_test)
svm_acc
```

```
from sklearn.metrics import confusion_matrix
svm_cm=confusion_matrix(svm_pred,y_test)
svm_cm

import sklearn.metrics as metrics
fpr,tpr,threshold=metrics.roc_curve(y_test,svm_pred)
roc_auc=metrics.auc(fpr,tpr)

import matplotlib.pyplot as plt
plt.title("ROC_AUC CURVE for SVM")
plt.plot(fpr,tpr,'g',label='auc=%0.2f'%roc_auc)
plt.plot([0,1],[0,1],'r--')
plt.xlim([0,1])
plt.ylim([0,1])
plt.xlabel('tpr')
plt.ylabel('tpr')
plt.legend(loc='lower right')

import pickle
pickle.dump(svm,open('churnnew.pkl','wb'))

"""LOGISTIC REGRESSION"""
```

```
from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score, classification_report


# Instantiate the Logistic Regression classifier

logreg_classifier = LogisticRegression()


# Train the classifier on the training data

logreg_classifier.fit(x_train, y_train)


# Make predictions on the training and testing data

y_train_pred = logreg_classifier.predict(x_train)

y_test_pred = logreg_classifier.predict(x_test)


# Calculate accuracy

train_accuracy = accuracy_score(y_train, y_train_pred)

test_accuracy = accuracy_score(y_test, y_test_pred)

print("Training Accuracy:", train_accuracy)

print("Testing Accuracy:", test_accuracy)


# Calculate overall accuracy

overall_accuracy = accuracy_score(y_test, y_test_pred)

print("Overall Accuracy:", overall_accuracy)
```

```
# Print classification report
print(classification_report(y_test, y_test_pred))

"""DECISION TREE"""

from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report

# Create the Decision Tree classifier
dt = DecisionTreeClassifier()

# Train the classifier
dt.fit(x_train, y_train)

# Make predictions on the test set
y_pred = dt.predict(x_test)

# Generate the classification report
report = classification_report(y_test, y_pred)
print(report)

"""RANDOM FOREST"""
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
# Create the Random Forest classifier
rf = RandomForestClassifier()

# Train the classifier
rf.fit(x_train, y_train)

# Make predictions on the test set
y_pred = rf.predict(x_test)

# Generate the classification report
report = classification_report(y_test, y_pred)
print(report)

"""XGBOOST"""

from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import classification_report

# Create the Gradient Boosting classifier
gb = GradientBoostingClassifier()
```

```
# Train the classifier
gb.fit(x_train, y_train)

# Make predictions on the test set
y_pred = gb.predict(x_test)

# Generate the classification report
report = classification_report(y_test, y_pred)
print(report)




"""NAIVE BAYES"""

from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report



# Create the Naive Bayes classifier
nb = GaussianNB()

# Train the classifier
nb.fit(x_train, y_train)
```

```
# Make predictions on the test set
y_pred = nb.predict(x_test)

# Generate the classification report
report = classification_report(y_test, y_pred)
print(report)

"""KNN"""

from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report



# Create the KNN classifier
knn = KNeighborsClassifier()

# Train the classifier
knn.fit(x_train, y_train)

# Make predictions on the test set
y_pred = knn.predict(x_test)
```

```
# Generate the classification report

report = classification_report(y_test, y_pred)

print(report)


# import pickle

# pickle.dump(svm,open('churnnew.pkl','wb'))


"""HYPER PARAMETER TUNING ON KNN"""


from sklearn.neighbors import KNeighborsClassifier

from sklearn.model_selection import GridSearchCV

from sklearn.metrics import classification_report



# Define the parameter grid for GridSearchCV

param_grid = {

    'n_neighbors': [3, 5, 7, 9, 11],

    'weights': ['uniform', 'distance'],

    'p': [1, 2]

}


# Create the KNN classifier

knn = KNeighborsClassifier()
```

```
# Create the GridSearchCV object
grid_search = GridSearchCV(knn, param_grid, cv=5, scoring='accuracy')

# Fit the GridSearchCV object to the training data
grid_search.fit(x_train, y_train)

# Get the best parameters and best score
best_params = grid_search.best_params_
best_score = grid_search.best_score_
print("Best Parameters:", best_params)
print("Best Score:", best_score)

# Use the best estimator to make predictions on the test set
best_estimator = grid_search.best_estimator_
y_pred = best_estimator.predict(x_test)

# Generate the classification report
report = classification_report(y_test, y_pred)
print(report)
# Save the best estimator as a pickle file
filename = 'knn_model.pkl'
pickle.dump(best_estimator, open(filename, 'wb'))
```

FLASK CODES:

```
from flask import Flask, render_template, request

app = Flask(__name__)

import pickle

model = pickle.load(open(r'C:\Users\ilang\flask\knn_model.pkl','rb'))

@app.route('/')

def helloworld():

    return render_template("base.html")

@app.route('/assesment')

def prediction():

    return 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 (b == 'y'):
    b=1
c= request.form["partner"]
if (c == 'n'):
    c=0
if (c == 'y'):
    c=1
d= request.form["dependents"]
if (d == 'n'):
    d=0
if (d == 'y'):
    d=1
e= request.form["tenure"]
f= request.form["phservices"]
if (f == 'n'):
    f=0
if (f == 'y'):
    f=1
g= request.form["multi"]
if (g == 'n'):
    g1,g2,g3=1,0,0
if (g == 'nps'):
```

```
    g1,g2,g3=0,1,0
if (g == 'y'):
    g1,g2,g3=0,0,1
h= request.form["is"]
if (h == 'dsl'):
    h1,h2,h3=1,0,0
if (h == 'fo'):
    h1,h2,h3=0,1,0
if (h == 'n'):
    h1,h2,h3=0,0,1
i= request.form["os"]
if (i == 'n'):
    i1,i2,i3=1,0,0
if (i == 'nis'):
    i1,i2,i3=0,1,0
if (i == 'y'):
    i1,i2,i3=0,0,1
j= request.form["ob"]
if (j == 'n'):
    j1,j2,j3=1,0,0
if (j == 'nis'):
    j1,j2,j3=0,1,0
if (j == 'y'):
```

```
    j1,j2,j3=0,0,1
k= request.form["dp"]
if (k == 'n'):
    k1,k2,k3=1,0,0
if (k == 'nis'):
    k1,k2,k3=0,1,0
if (k == 'y'):
    k1,k2,k3=0,0,1
l= request.form["ts"]
if (l == 'n'):
    l1,l2,l3=1,0,0
if (l == 'nis'):
    l1,l2,l3=0,1,0
if (l == 'y'):
    l1,l2,l3=0,0,1
m= request.form["stv"]
if (m == 'n'):
    m1,m2,m3=1,0,0
if (m == 'nis'):
    m1,m2,m3=0,1,0
if (m == 'y'):
    m1,m2,m3=0,0,1
n= 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=0,0,1
o= request.form["contract"]
if (o == 'mtm'):
    o1,o2,o3=1,0,0
if (o == 'oyr'):
    o1,o2,o3=0,1,0
if (o == 'tyrs'):
    o1,o2,o3=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 (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(i1),int(i2),int(i3),int(j1),int(j2),int(j3),int(k
1),int(k2),int(k3),int(l1),int(l2),int(l3),int(m1),int(m2),int(m3),int(n1),int(n2),int(n3),int(o1),int
(o2),int(o3),int(p1),int(p2),int(p3),int(p4),int(a),int(b),int(c),int(d),int(e),int(f),int(q),float(r),flo
at(s)]]
    x = model.predict(t)
    if (x[0] == 0):
        y ="No"
        return render_template("predno.html", z = y)

    if (x[0] == 1):
        y ="Yes"
        return render_template("predyes.html", z = y)

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

12. BIBLIOGRAPHY

1. S. De, P. P and J. Paulose, "Effective ML Techniques to Predict Customer Churn," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2021, pp. 895-902, doi: 10.1109/ICIRCA51532.2021.9544785.
2. Anouar Dalli, “Impact of Hyperparameters on Deep Learning Model for Customer Churn Prediction in Telecommunication Sector”, Ecole Nationale des Sciences Appliqu´ees de Safi (ENSAS), Universit´e Cadi Ayyad, Marrakesh, Morocco, Volume 2022, Article ID 4720539, 11 pages, https://doi.org/10.1155/2022/4720539
3. F. A. Mohamed and A. K. Al-Khalifa, "A Review of Machine Learning Methods For Predicting Churn in the Telecom Sector," 2023 International Conference On Cyber Management And Engineering (CyMaEn), Bangkok, Thailand, 2023, pp. 164-170, doi: 10.1109/CyMaEn57228.2023.10051108.
4. V. C. Nwaogu and K. Dimililer, "Customer Churn Prediction For Business Intelligence Using Machine Learning," 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 2021, pp. 1-7, doi: 10.1109/HORA52670.2021.9461303.
5. R. Srinivasan, D. Rajeswari and G. Elangovan, "Customer Churn Prediction Using Machine Learning Approaches," 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), Chennai, India, 2023, pp. 1-6, doi: 10.1109/ICECONF57129.2023.10083813.
6. Ahmad, A.K., Jafar, A. & Aljoumaa, K. Customer churn prediction in telecom using machine learning in the big data platform. *J Big Data* **6**, 28 (2019). https://doi.org/10.1186/s40537-019-0191-6
7. Kavitha, V. & Kumar, G. & Kumar, S. Harish, (2020). Churn Prediction of Customer in Telecom Industry using Machine Learning Algorithms. International Journal of Engineering Research and. V9. 10.17577/IJERTV9IS050022.
8. Y. K. Saheed and M. A. Hambali, "Customer Churn Prediction in Telecom Sector with Machine Learning and Information Gain Filter Feature Selection Algorithms," 2021

International Conference on Data Analytics for Business and Industry (ICDABI), Sakheer, Bahrain, 2021, pp. 208-213, doi: 10.1109/ICDABI53623.2021.9655792.

9. Senthilnayaki, B. & M, Swetha & D, Nivedha. (2021). CUSTOMER CHURN PREDICTION. IARJSET. 8. 527-531. 10.17148/IARJSET.2021.8692.
10. Ebrah, K. and Elnasir, S. (2019) Churn Prediction Using Machine Learning and Recommendations Plans for Telecoms. Journal of Computer and Communications, 7, 33-53. https://doi.org/10.4236/jcc.2019.711003