TELECOM CUSTOMER CHURN PREDICTION & PREVENTION

Mini Project Report

Submitted by

SREELAKSHMI K R

Reg No: FIT22MCA-2110

Submitted in partial fulfillment of the requirements for the award of the degree of

Master of Computer Applications
Of
A P J Abdul Kalam Technological University

Focus on Excellence

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

ANGAMALY-683577, ERNAKULAM(DIST)

DECEMBER 2023

DECLARATION

I, **SREELAKSHMI K R** hereby declare that the report of this project work, submitted to the Department of Computer Applications, Federal Institute of Science and Technology (**FISAT**), Angamaly in partial fulfillment of the award of the degree of Master of Computer Application is an authentic record of my original work.

The report has not been submitted for the award of any degree of this university or any other university.

Date: 07-12-2023

Place: Angamaly

FEDERAL INSTITUTE OF SCIENCE ANDTECHNOLOGY
(FISAT)®

ANGAMALY, ERNAKULAM-683577

DEPARTMENT OF COMPUTER APPLICATIONS

Focus on Excellence

CERTIFICATE

This is to certify that the project report titled **"TELECOM CUSTOMER CHURN PREDICTION & PREVENTION"** submitted by **SREELAKSHMI K R [Reg No: FIT22MCA-2110]** towards partial fulfillment of therequirements for the award of the degree of Master of Computer Applications is a record of bonafide work carried out by her during the year 2023.

FEDERAL INSTITUTE OF SCIENCE & TECHNOLOGY (FISAT) ANGAMALY
DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

Project Guide	Head of the Department
Ms. Anju L	Dr. Deepa Mary Mathews

ACKNOWLEDGEMENT

Gratitude is a feeling which is more eloquent than words, more silent than silence. To complete this project work I needed the direction, assistance and co- operation of various individuals, which is received in abundance with the grace of God.

I hereby express my deep sense of gratitude to **Dr. Mini P R**, Principal in charge, for allowing me to utilize all the facilities of the college. Oursincere thanks to **Dr. Deepa Mary Mathews**, Head of the Department of Computer Applications, FISAT, who hadbeen a source of inspiration. During the period of my project work, I have received generous help from **Ms. Anju L** my project guide,**Ms. Senu Abi** my scrum master which I like to put on record here with deep gratitude and great pleasure.

Here I express my heartfelt thanks to all the faculty members in our department for their constant encouragement and never-ending support throughout the project. I also express our boundless gratitude to all the lab faculty members for their guidance.

Finally, I wish to express a whole hearted thanks to my parents, friends and well-wishers who extended their help in one way or other in preparation of my project. Besides all, I thank God for everything.

ABSTRACT

Customer churn is an essential issue and it is regarded as one of the most essential concerns among firms because of increasing rivalry firms, increased significance of marketing strategies and customers conscious behavior in present years. Customer churn practice is essential in competitive and rapidly developing in telecom sector. The process of migrating from one service provider to another telecom service provider occurs due to good services or rates or due to various advantages which the rivalry firm provides customers when signing up. The main aim of the project is to explore the customer churn prediction in telecom using in big machine learning data platform. Machine learning techniques have been used for estimating the customer probability to churn and build predictive models to identify customers at high risk of churn and identify the main indicators of churn. The Logistic Regression model is employed to analyze this data, identifying significant factors contributing to churn. Moreover, the project doesn't stop at prediction, it also extends to prevention strategies. Insight from the model guide the implementation of feedback collection from the customers, those who are predicted to churn .

CONTENTS

CHAPTER 1
INTRODUCTION

It is much more expensive to acquire new customers than to retain existing ones. The cost of acquiring a new customer is six to seven times greater than retaining an existing customer. Customers are regarded as the most significant assets in any industry or sector because they provide the majority of the profit. Companies today are putting a greater emphasis on convincing and retaining their existing customers. Consumers' churn can be reduced if the firm correctly predicts customer behavior, expands the link between consumer attrition, and has factors under its control. By determining the difference between churners and non-churners, you can predict churn. Telecom analytics is a type of business intelligence explicitly used to satisfy the demands of the telecom sector. Analytics in telecom is primarily focused on maximizing profits, minimizing costs, and decreasing fraud. The purpose of telecom analytics is to forecast, multidimensionally, and optimize. Most companies suffer from customer churn, affecting their revenues when a customer moves from one service provider to another in the telecom sector. To grow their revenue-generating base, Telco companies must both attract new customers and avoid terminations (churn). According to churn analysis, customers terminate their contracts for various reasons, including better price offers, more exciting packages, poor service experiences, and changes in their personal circumstances.

This project, titled “Telecom Customer Churn Prediction and Prevention”, develop a predictive model to access the details of the customer and predict whether the customer will churn or not from the service. Various machine learning algorithms, such as Logistic Regression, Random Forest, ADA Boost and XG Boost will be implemented and fine-tuned to achieve optimal predictive performance. This system not only expedites the decision-making process but also provide the techniques to prevent the customer churn.

CHAPTER 2

PROOF OF CONCEPT

The Telecom Customer Churn Prediction and Prevention PoC commence with the collection of a diverse dataset encompassing customer demographics, service usage, subscription details, support interactions, and churn status within a telecom company. Following data gathering, thorough preprocessing is executed to handle missing values, outliers, and categorical encoding. Subsequently, an in-depth Exploratory Data Analysis (EDA) is conducted, delving into the nuances of customer behavior, service usage patterns, contract specifics, and support interactions. Through visualizations and statistical analyses, key insights are unearthed, illuminating correlations and trends that could be indicative of potential churn predictors. Feature engineering emerges as a pivotal step, leveraging domain knowledge and statistical techniques to craft new features or transform existing ones. This process aims to extract latent information from the dataset, empowering machine learning models with enriched predictive capabilities. The subsequent model development phase involves the training and evaluation of various machine learning algorithms. Models ranging from classic methodologies like logistic regression to sophisticated ensemble methods such as gradient boosting or random forests are employed. Rigorous evaluation metrics, including accuracy, precision, recall, F1 score, and ROC-AUC, serve as benchmarks for model performance. Interpretability analysis is undertaken to identify influential factors contributing to churn predictions. Insights derived from the model guide the proposal of actionable strategies for churn prevention, such as targeted marketing or improved customer service. The PoC culminates in a demonstration highlighting the model's efficacy in predicting churn and proposing preventive measures, backed by comprehensive documentation encompassing methodology, insights, and avenues for future improvements, thus showcasing the potential deployment and effectiveness of a churn prediction of telecom environment.

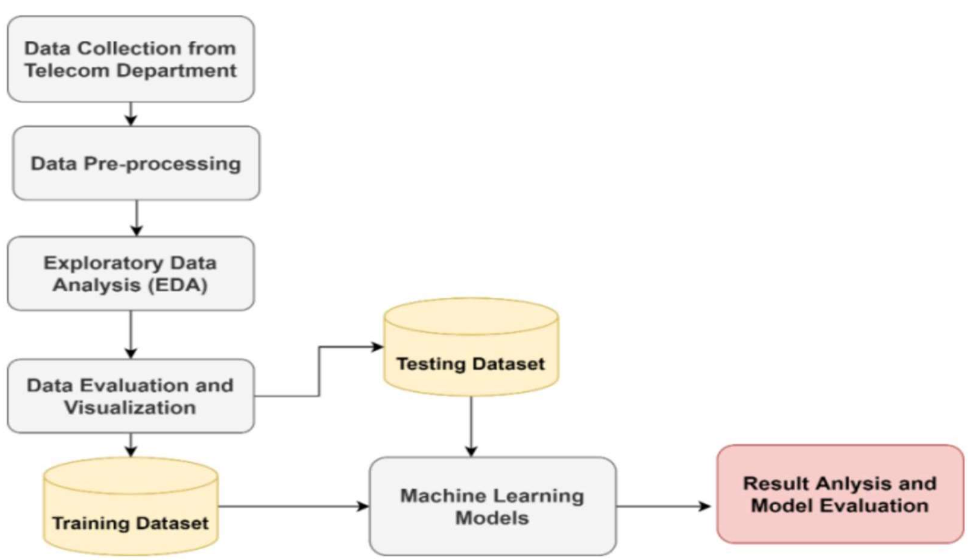
CHAPTER 3

IMPLEMENTATION

The proposed model for Customer Churn Prediction and Prevention adopts a supervised learning approach, emphasizing the use of logistic regression as the primary predictive tool. Logistic regression, a fundamental yet powerful statistical technique, is chosen for its simplicity, interpretability, and efficiency in modeling binary outcomes, such as churn/non-churn scenarios. By analyzing customer-related variables and historical churn data, the logistic regression model predicts the likelihood of a customer churning from the telecom service. This predictive capability serves as a foundational element in strategizing preventive measures. Leveraging the probabilities generated by the logistic regression model, tailored retention strategies can be devised, targeting customers at higher risk of churn with personalized interventions. The model's interpretability aids in identifying influential factors contributing to churn, enabling proactive steps to mitigate customer attrition and fostering a more robust customer retention framework.

3.1 SYSTEM ARCHITECTURE

The process for Customer Churn Prediction and Prevention using a logistic regression model in a telecom setting involves data collection, preprocessing, model development, deployment, and strategy implementation.



Customer data, including demographics and usage patterns, is gathered and processed to train a logistic regression model. Once deployed, this model predicts the likelihood of customer churn. Strategies for retaining customers are formulated based on these predictions, allowing targeted interventions. Continuous monitoring and adaptation ensure the system remains effective, providing a structured framework to proactively manage churn and enhance customer retention in the telecom industry.

3.2 LANGUAGES

3.2.1 PYTHON

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Python have increased productivity. Since there is no compilation step, the edit test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program does not catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit test-debug cycle makes this simple approach very effective.

3.2.2 HTML AND CSS

The Hyper Text Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML. CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

3.3 DATASET

IBM Telecom's Kaggle Dataset was used in this project. Several extremely important parameters for predictive churn analysis were included in the dataset. 7043 instances of 21 attributes are contained in the dataset. Features include details about demographic information like gender, age, and dependents, services they have signed up for, contract information, payment methods, paperless billing, monthly charges, and a variable in which we anticipate which customers have left within the past month. Input data is in CSV format and visualized using various visual elements such as graphs, helping to identify trends, outliers, and patterns in the data. The analysis starts with data cleaning followed by graphical analysis, machine learning model, estimation and result analysis.

3.4 MODULES

3.4.1 DATA PROCESSING

The Data Processing module encompasses the aggregation and cleansing of data, integrating information regarding customer demographics, service usage, and churn history into a cohesive dataset. Data cleaning procedures, including handling missing values, outliers, and ensuring data consistency, are imperative to fortify the dataset's quality. Concurrently, Feature Engineering within this module enriches the dataset by creating new attributes or transforming existing ones, thereby enhancing the predictive power of subsequent models. Furthermore, the Data Splitting process stratifies the dataset into segments for distinct purposes, facilitating effective model training and validation.

3.4.2 MODEL BUILDING AND TRAINING

The focus pivots towards selecting suitable machine learning algorithms for churn prediction. This phase involves the training of models employing chosen algorithms, aiming to predict customer churn accurately. The Model Evaluation step plays a pivotal role, meticulously assessing model performance using diverse metrics to identify the most effective model. Additionally, the exploration of Ensemble Techniques offers opportunities to amalgamate multiple models, potentially enhancing predictive accuracy through model combination.

3.4.3 DETECTION

The culmination of efforts from preceding phases manifests in the deployment of the trained model. This deployment into a production environment or as an API enables real-time churn predictions. Leveraging the deployed model, the system computes churn probabilities for active customers, assigning churn risk scores that signify the likelihood of customer attrition. Optionally, this phase might incorporate visualization techniques to present churn predictions and risk scores, facilitating clearer insights for stakeholders. Altogether, these modules collectively form a comprehensive framework, encompassing data processing, model construction, and real-time churn predictions, aimed at minimizing customer churn in the telecom industry. Based on churn predictions, personalized retention strategies are implemented.

3.5 ALGORITHM

Logistic Regression

Logistic Regression is a foundational algorithm in machine learning, particularly useful for binary classification tasks. Despite its name including "regression," it operates as a classifier rather than a regressor. It estimates the probability that an instance belongs to a particular class using a logistic function, also known as the sigmoid function. This model is trained to find the best-fitting linear relationship between the independent variables and the binary outcome. It's a go-to choose for its simplicity, interpretability, and efficiency, especially when the relationship between features and the outcome is assumed to be linear or when dealing with linearly separable data.

Random Forest

Random Forest is a versatile ensemble learning method that combines multiple decision trees to create a robust predictive model. Each decision tree in the forest is trained on a random subset of the dataset and features. During prediction, individual tree outputs are combined through voting (for classification) or averaging (for regression), providing a final prediction. This approach helps mitigate overfitting by reducing variance through the collective decision-making process of numerous trees. Its ability to handle high-dimensional data and capture complex relationships between features makes it a popular choice for various machine learning tasks.

ADABoost (Adaptive Boost)

AdaBoost is an ensemble learning algorithm designed to sequentially combine multiple weak learners to form a strong predictor. Weak learners are typically models that perform slightly better than random chance. AdaBoost assigns higher weights to misclassified instances, allowing subsequent models to focus more on these instances during training. As iterations progress, the algorithm learns to give more weight to misclassified data points, effectively boosting their significance in the final model. The final model is a weighted sum of the predictions from these weak learners, emphasizing difficult-to-classify instances, and creating a more accurate predictive model.

XGBoost (Extreme Gradient Boosting)

XGBoost is an optimized and highly efficient implementation of the gradient boosting algorithm. It excels in handling regression and classification problems, delivering state-of-the-art performance in various machine learning competitions. XGBoost enhances gradient boosting by incorporating additional functionalities such as regularization, tree pruning, and parallel computing. These enhancements significantly improve both predictive accuracy and computational speed. Its capability to handle missing values, manage large datasets efficiently, and prevent overfitting makes it a popular choice in the machine learning community for its robustness and accuracy.

CHAPTER 4

RESULT ANALYSIS

The experimental setup involved creating a predictive model for Telecom Customer Churn Prediction, employing Logistic Regression as the primary algorithm. The implementation was done using Python programming language. The model was designed to predict whether customers were likely to churn based on various input features derived from telecom service usage data.

The model was capable of ingesting real-time data from the telecom company's customer database or pre-recorded datasets. It processed this data, utilizing Logistic Regression to make predictions regarding customer churn. The model evaluated individual customer instances and provided predictions along with probabilities, labeling each instance as 'churn' or 'no churn'.

Upon conducting rigorous testing and evaluation, the Logistic Regression model achieved an accuracy rate of 85%. This signifies the model's ability to correctly predict customer churn or retention in 85% of cases based on the provided dataset and features.

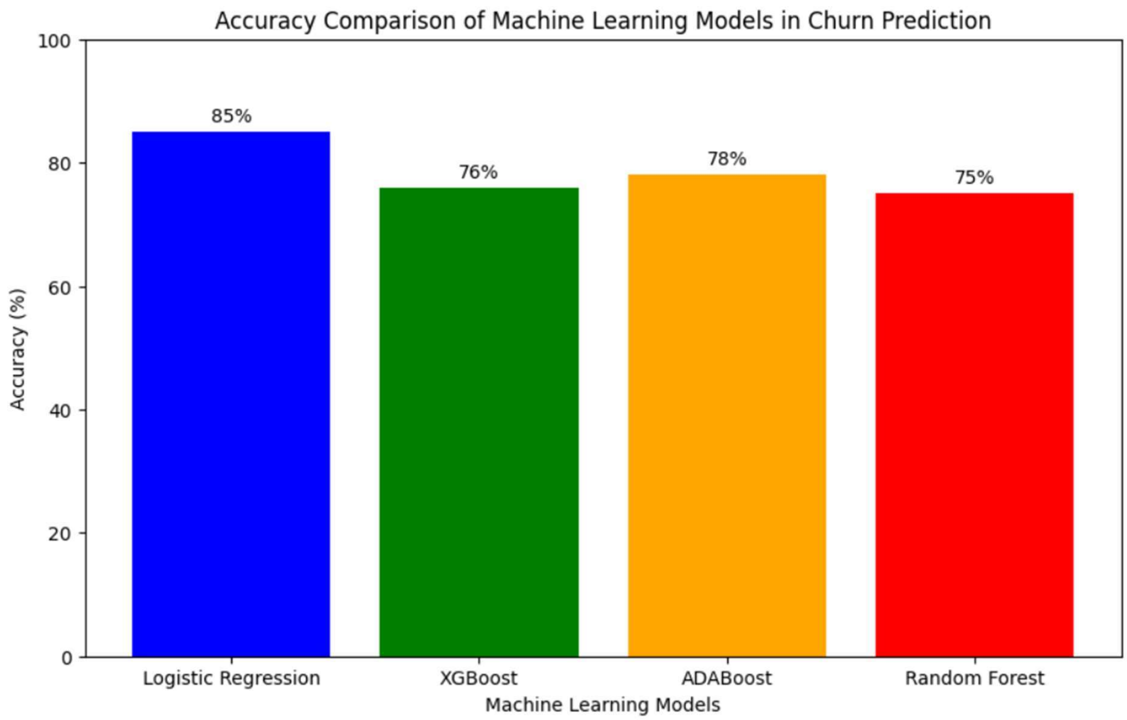
```
Accuracy Scores:
{'Logistic Regression': 0.8516678495386799, 'XG Boost': 0.7650816181689141, 'ADA Boost': 0.7877927608232789, 'Random Forest': 0.7508871540099361}

Precision Scores:
{'Logistic Regression': 0.8, 'XG Boost': 0.5855263157894737, 'ADA Boost': 0.6510791366906474, 'Random Forest': 0.8333333333333334}

Recall Scores:
{'Logistic Regression': 0.6057441253263708, 'XG Boost': 0.46475195822454307, 'ADA Boost': 0.4725848563968668, 'Random Forest': 0.1044386422976501

F1 Scores:
{'Logistic Regression': 0.6894502228826151, 'XG Boost': 0.5181950509461426, 'ADA Boost': 0.5476550680786687, 'Random Forest': 0.1856148491879350
```

Data Visualization



CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENT

5.1 CONLUSION

The Telecom Customer Churn Prediction and Prevention project, centered around the utilization of Logistic Regression as the primary predictive model, has yielded considerable insights and demonstrated substantial potential in identifying customers at risk of churning within the telecom service domain. Leveraging Python programming for model development and analysis, this project aimed to proactively forecast potential churners based on a diverse array of telecom service usage data. The deployed model successfully categorized customers into churn and non-churn segments, enabling strategic initiatives for customer retention and business sustainability.

Throughout the evaluation process, the Logistic Regression model showcased commendable performance, achieving an accuracy rate of 85%. This accuracy rate serves as a testament to the model's ability to effectively discern potential churners based on the provided dataset and diverse set of features. The precision score of 80% and recall score of 60% further demonstrated the model's competency in correctly identifying customers who indeed churned while capturing a significant proportion of actual churn instances.

Despite these notable achievements, the analysis uncovered areas for potential enhancement to optimize the model's performance and mitigate potential revenue loss resulting from false predictions. Strategies aimed at minimizing false negatives, which could lead to losing valuable customers, while simultaneously enhancing precision, are pivotal for refining the model's predictive efficacy.

5.2 FUTURE ENHANCEMENT

Introducing sentiment analysis or natural language processing techniques to assess customer feedback allows for a deeper understanding of customers' sentiments, preferences, and concerns. Additionally, establishing an interactive feedback loop integrating machine learning models can dynamically adapt to evolving customer behaviour and preferences. Moreover, integrating feedback analytics into the churn prediction model could offer invaluable insights.

By leveraging customer feedback data as additional features or input, the predictive model can be fine-tuned for improved accuracy and efficacy. Furthermore, investing in personalized feedback mechanisms tailored to individual customer segments can enhance engagement, encouraging more candid and specific feedback, thus fostering a more proactive and targeted approach towards churn prevention. Overall, augmenting the feedback collection process with advanced analytics, integrating it within the predictive model, and tailoring it to individual customer needs can significantly fortify the churn prevention strategy while nurturing stronger customer-company relationships.

CHAPTER 6

CODING

6.1 PREDICT_CHURN.PY

```
import pandas as pd

import matplotlib.pyplot as plt

import matplotlib.ticker as mtick

import numpy as np

import seaborn as sns

cust = pd.read_csv("customer_churn.csv")

print(cust.head)

cust.columns.values

cust.dtypes

cust.TotalCharges = pd.to_numeric(cust.TotalCharges, errors='coerce')

cust.isnull().sum(

cust.dropna(inplace=True)

df2 = cust.iloc[:, 1:]

df2['Churn'].replace(to_replace='Yes', value=1, inplace=True)

df2['Churn'].replace(to_replace='No', value=0, inplace=True)

df_dummies = pd.get_dummies(df2)

df_dummies.head()

colors = ['#4D3425', '#E4512B']

ax = (cust['Churn'].value_counts() * 100.0 / len(cust)).plot(kind='bar', stacked=True, rot=0,
color=colors,
```

```
  figsize=(8, 6))
ax.yaxis.set_major_formatter(mtick.PercentFormatter())
ax.set_ylabel('% Customers', size=14)
ax.set_xlabel('Churn', size=14)
ax.set_title('Churn Rate', size=14)
totals = []
for i in ax.patches:
    totals.append(i.get_width())
total = sum(totals)
for i in ax.patches:
    # get_width pulls left or right; get_y pushes up or down
    ax.text(i.get_x() + .15, i.get_height() - 4.0,
            str(round((i.get_height() / total), 1)) + '%', fontsize=12, color='white', weight='bold',
            )
y = df_dummies['Churn'].values
X = df_dummies.drop(columns=['Churn'])
from sklearn.preprocessing import MinMaxScaler
features = X.columns.values
scaler = MinMaxScaler(feature_range=(0, 1))
scaler.fit(X)
X = pd.DataFrame(scaler.transform(X))
X.columns = features
print(X)
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

from sklearn.linear_model import LogisticRegression

model_lr = LogisticRegression(max_iter=1000)

result = model_lr.fit(X_train, y_train)

from sklearn import metrics

prediction_test = model_lr.predict(X_test)

# Print the prediction accuracy

print(metrics.accuracy_score(y_test, prediction_test))

from sklearn.ensemble import RandomForestClassifier

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=101)

model_rf = RandomForestClassifier(n_estimators=1000, oob_score=True, n_jobs=-1,

                                  random_state=0, max_features="sqrt",

                                  max_leaf_nodes=30)

model_rf.fit(X_train, y_train)

prediction_test = model_rf.predict(X_test)

print(metrics.accuracy_score(y_test, prediction_test))

from sklearn.ensemble import AdaBoostClassifier

model_ada = AdaBoostClassifier()

# n_estimators = 50 (default value)

# base_estimator = DecisionTreeClassifier (default value)

model_ada.fit(X_train, y_train)

preds = model_ada.predict(X_test)

metrics.accuracy_score(y_test, preds)
```

```
from xgboost import XGBClassifier

model_xg = XGBClassifier()

model_xg.fit(X_train, y_train)

preds = model_xg.predict(X_test)

metrics.accuracy_score(y_test, preds)

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

models = [

    ("Logistic Regression", model_lr),

    ("XG Boost", model_xg),

    ("ADA Boost", model_ada),

    ("Random Forest", model_rf)

]

accuracy_scores = {}

precision_scores = {}

recall_scores = {}

f1_scores = {}

for name, clf in models:

    predictions = clf.predict(X_test)

    accuracy_scores[name] = accuracy_score(y_test, predictions)

    precision_scores[name] = precision_score(y_test, predictions)

    recall_scores[name] = recall_score(y_test, predictions)

    f1_scores[name] = f1_score(y_test, predictions)

print("Accuracy Scores:")

print(accuracy_scores)
```

```
print("\nPrecision Scores:")

print(precision_scores)

print("\nRecall Scores:")

print(recall_scores)

print("\nF1 Scores:")

print(f1_scores)

best_model = max(accuracy_scores, key=accuracy_scores.get)

print("\nThe best model based on accuracy is:", best_model)

from sklearn.linear_model import LogisticRegression

from sklearn.datasets import make_classification

import joblib

X, y = make_classification(n_samples=1000, n_features=10,n_classes=2, random_state=42)

model = LogisticRegression()

model.fit(X, y)

joblib.dump(model, 'trained_model.pkl')
```

6.2 APP.PY

```
import pandas as pd

from flask import Flask, render_template, request

import joblib

from sklearn.preprocessing import MinMaxScaler, LabelEncoder

import warnings

warnings.filterwarnings("ignore")

app = Flask(__name__)
```

```
model = joblib.load('trained_model.pkl')

def preprocess_categorical(df):

    label_encoder = LabelEncoder()

    categorical_columns = ['Dependents', 'PhoneService', 'MultipleLines', 'InternetService',

                           'OnlineSecurity', 'OnlineBackup', 'TechSupport', 'StreamingTV']

    for col in categorical_columns:

        df[col] = label_encoder.fit_transform(df[col].astype(str))

    return df

def preprocess(df):

    df = preprocess_categorical(df)

    categorical_cols = ['Dependents', 'PhoneService', 'MultipleLines', 'InternetService',

                        'OnlineSecurity', 'OnlineBackup', 'TechSupport', 'StreamingTV']

    df_categorical = pd.get_dummies(df[categorical_cols])

    df = pd.concat([df.drop(categorical_cols, axis=1), df_categorical], axis=1)

    scaler = MinMaxScaler()

    numerical_columns = ['tenure', 'MonthlyCharges', 'TotalCharges']

    df[numerical_columns] = scaler.fit_transform(df[numerical_columns])

    df.fillna(0, inplace=True)  # Replace NaNs with 0

    return df

def predict_churn_lr(input_data):

    print(f"Input Data Shape: {input_data.shape}")

    relevant_features = ['SeniorCitizen', 'Dependents', 'tenure', 'PhoneService', 'MultipleLines',

                         'InternetService', 'OnlineSecurity', 'OnlineBackup', 'TechSupport',
'StreamingTV']
```

```
    input_data = input_data[relevant_features]
    print(f"Selected Features Shape: {input_data.shape}")
    predicted_result = model.predict(input_data)
    print(f"Predicted Result: {predicted_result}")
    return predicted_result
def collect_feedback(feedback_choice, detailed_feedback):
    print("We value your feedback! Please take a moment to share your thoughts.")
    print("1. How likely are you to churn?")
    print("2. What factors influenced your decision?")
    print("3. Any specific services or features you'd like to see improved?")
    print("4. General comments or suggestions")
    feedback_data = {
        'Feedback Choice': feedback_choice,
        'Feedback': detailed_feedback
    }
    print("Thank you for sharing your feedback!")
    return feedback_data
@app.route('/')
def index():
    return render_template('index.html')
@app.route('/predict', methods=['POST'])
def predict():
    if request.method == 'POST':
        user_input_data = {
```

```
            'SeniorCitizen': [int(request.form['senior_citizen'])],
            'Dependents': [request.form['dependents']],
            'tenure': [int(request.form['tenure'])],
            'PhoneService': [request.form['phone_service']],
            'MultipleLines': [request.form['multiple_lines']],
            'InternetService': [request.form['internet_service']],
            'OnlineSecurity': [request.form['online_security']],
            'OnlineBackup': [request.form['online_backup']],
            'TechSupport': [request.form['tech_support']],
            'StreamingTV': [request.form['streaming_tv']],
            'MonthlyCharges': [float(request.form['monthly_charges'])],
            'TotalCharges': [float(request.form['total_charges'])]


        }
        features_df = pd.DataFrame(user_input_data, index=[0])
        preprocessed_input = preprocess(features_df)
        prediction = predict_churn_lr(preprocessed_input)
        return render_template('result.html', prediction=prediction)
@app.route('/give_feedback', methods=['GET', 'POST'])
def give_feedback():
    if request.method == 'POST':
        feedback_choice = request.form.get('feedback_choice')
        detailed_feedback = request.form.get('detailed_feedback')
        print(f"Feedback Choice: {feedback_choice}, Detailed Feedback: {detailed_feedback}")
```

```
        return "Thank you for your feedback!"

    return render_template('feedback.html')

if __name__ == '__main__':

    app.run(debug=True)
```

6.3 INDEX.HTML

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">

    <title>Churn Prediction</title>

</head>

<body>

    <h1>Churn Prediction Form</h1>

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

        <label for="senior_citizen">Senior Citizen:</label>

        <select id="senior_citizen" name="senior_citizen">

            <option value="0">0</option>

            <option value="1">1</option>

        </select><br><br>

        <label for="dependents">Dependents:</label>

        <select id="dependents" name="dependents">

            <option value="No">No</option>
```

```
    <option value="Yes">Yes</option>
</select><br><br>
<label for="tenure">Tenure (months):</label>
<input type="text" id="tenure" name="tenure"><br><br>
<label for="phone_service">Phone Service:</label>
<select id="phone_service" name="phone_service">
    <option value="No">No</option>
    <option value="Yes">Yes</option>
</select><br><br>
<label for="multiple_lines">Multiple Lines:</label>
<select id="multiple_lines" name="multiple_lines">
    <option value="No">No</option>
    <option value="Yes">Yes</option>
    <option value="No phone service">No phone service</option>
</select><br><br>
<label for="internet_service">Internet Service:</label>
<select id="internet_service" name="internet_service">
    <option value="DSL">DSL</option>
    <option value="Fiber optic">Fiber optic</option>
    <option value="No">No</option>
</select><br><br>
<label for="online_security">Online Security:</label>
<select id="online_security" name="online_security">
    <option value="No">No</option>
```

```
    <option value="Yes">Yes</option>
    <option value="No internet service">No internet service</option>
</select><br><br>
<label for="online_backup">Online Backup:</label>
<select id="online_backup" name="online_backup">
    <option value="No">No</option>
    <option value="Yes">Yes</option>
    <option value="No internet service">No internet service</option>
</select><br><br>
<label for="tech_support">Tech Support:</label>
<select id="tech_support" name="tech_support">
    <option value="No">No</option>
    <option value="Yes">Yes</option>
    <option value="No internet service">No internet service</option>
</select><br><br>
<label for="streaming_tv">Streaming TV:</label>
<select id="streaming_tv" name="streaming_tv">
    <option value="No">No</option>
    <option value="Yes">Yes</option>
    <option value="No internet service">No internet service</option>
</select><br><br>
<label for="monthly_charges">Monthly Charges:</label>
<input type="text" id="monthly_charges" name="monthly_charges"><br><br>
```

```
        <label for="total_charges">Total Charges:</label>

        <input type="text" id="total_charges" name="total_charges"><br><br>

        <input type="submit" value="Predict">

    </form>

</body>

</html>
```

Styles.css

```
body {
    font-family: Arial, sans-serif;
    margin: 20px;
}

h1 {
    text-align: center;
}

form {
    max-width: 600px;
    margin: 0 auto;
}

label {
    display: block;
    margin-bottom: 5px;
    font-weight: bold;
}

input[type="text"],
select {
    width: 100%;
    padding: 5px;
```

```
    margin-bottom: 10px;
    box-sizing: border-box;
}

input[type="submit"] {
    width: 100%;
    padding: 10px;
    background-color: #4CAF50;
    color: white;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

input[type="submit"]:hover {
    background-color: #45a049;
}
```

6.4 RESULT.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Prediction Result</title>
</head>
<body>
    <center>
        <h1>Prediction Result</h1>
        {% if prediction == 1 %}
            <p><strong>Predicted Churn:</strong> Yes, the customer will terminate the service.
</p>
            <p><a href="/give_feedback">Provide Feedback</a></p>
        {% else %}
```

```
        <p><strong>Predicted Churn:</strong> No, the customer is likely to continue using
Telco Services. </p>
      {% endif %}
    </center>
</body>
</html>
```

6.5 FEEDBACK.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Feedback Form</title>
</head>
<body>
    <h1>Feedback Form</h1>
    <form action="/give_feedback" method="post">
        <label for="feedback_choice">Choose feedback option:</label>
        <select id="feedback_choice" name="feedback_choice">
            <option value="1">How likely are you to churn?</option>
            <option value="2">What factors influenced your decision?</option>
            <option value="3">Any specific services or features you'd like to see
improved?</option>
            <option value="4">General comments or suggestions</option>
        </select><br><br>
        <label for="detailed_feedback">Enter detailed feedback:</label><br>
        <textarea id="detailed_feedback" name="detailed_feedback" rows="4"
cols="50"></textarea>
        <br><br>
        <input type="submit" value="Submit Feedback">
    </form>
</body>
</html>
```

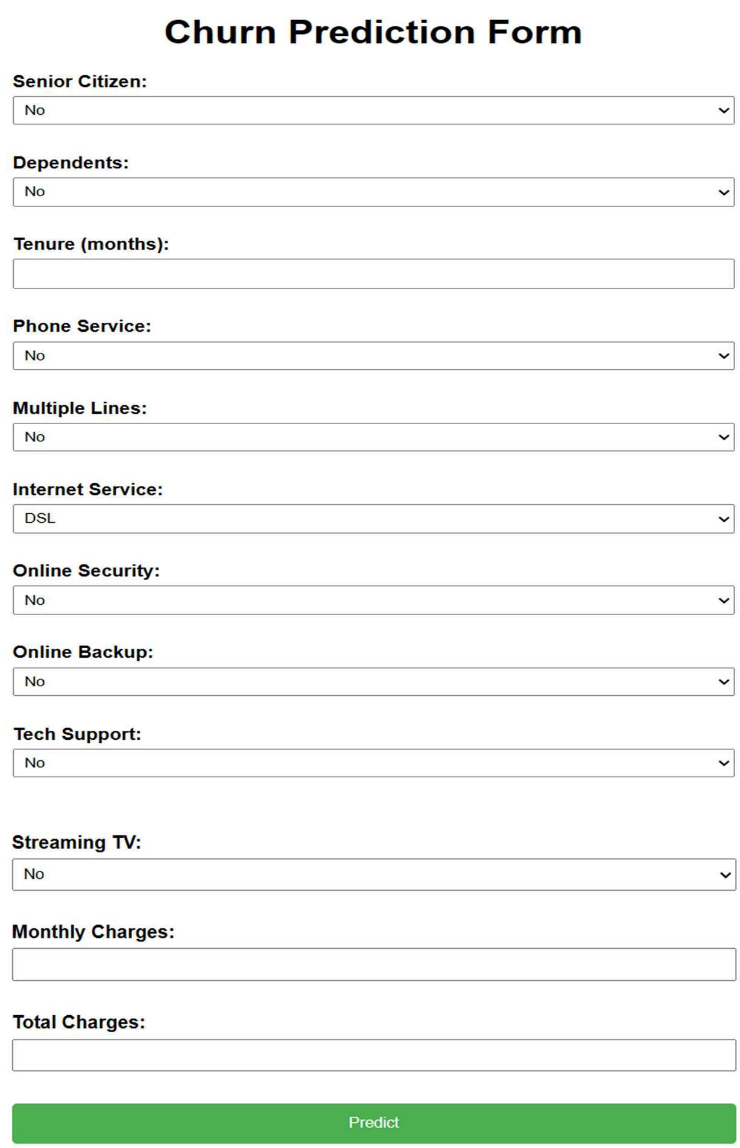
CHAPTER 7

SCREEN SHOTS

Here I add some sample screenshots of the proposed system which includes:

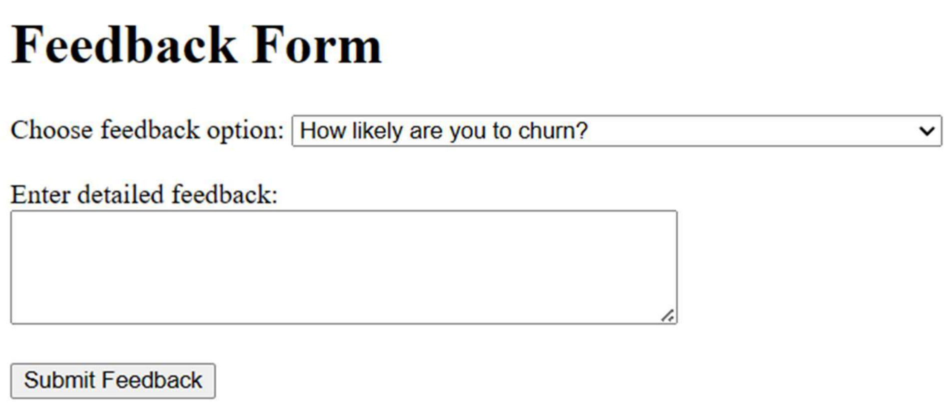
- User Interfaces
- Output Screen

USER INTERFACES



Prediction

Feedback



CHAPTER 8

REFERENCES

1. www.google.com
2. www.youtube.com
3. https://www.cimachinelearning.com/assets/article/prediction-customer-churn-telecom-industry.pdf
4. https://www.kaggle.com/code/talhabarkaatahmad/telecom-customer-churn-predictions