Internship Report
On
Customer Churn Analysis

Submitted by

Haibathi UdayKumar
24885A6609
Vardhaman College of Engineering

Submitted to

Mayur Dev Sewak
Head, Internships & Trainings
Eisystems Services

&

Mallika Srivastava
Head, Training Delivery
Eisystems Services

Table of Contents

Project Summary

Idea Behind Making This Project:

The primary objective of this project is to predict customer churn for a telecom company. Churn prediction helps the company identify customers who are at risk of leaving, allowing targeted retention strategies. The goal is to enhance customer satisfaction and reduce revenue losses associated with churn.

About the Project:

Customer churn analysis involves studying customer behavior patterns to determine factors that influence their decision to discontinue a service. The project uses machine learning models to analyze historical data and predict the likelihood of customer churn. By understanding the key drivers of churn, the company can improve customer engagement and reduce turnover.

Software Used in the Project:

- Python: For data manipulation, analysis, and building machine learning models.
- Jupyter Notebook: As the primary environment for coding and visualizing data.
- Scikit-Learn: For building and evaluating the Random Forest Classifier.
- Pandas & NumPy: For data handling and preprocessing.
- Matplotlib & Seaborn: For visualizing data distributions and patterns.

Technical Apparatus Requirements:

- A computer with at least 8GB RAM for handling large datasets efficiently.
- Python 3.x installed, along with required libraries (`pandas`, `numpy`, `scikit-learn`, `matplotlib`, `seaborn`).
- Jupyter Notebook for an interactive coding environment.
- Access to a telecom customer dataset for training and testing the model.

Result or Working of the Project:

The project resulted in a predictive model that can classify customers as likely to churn or not with an accuracy of 77%. The Random Forest Classifier was chosen due to its ability to handle complex relationships between variables. Key insights were derived regarding factors such as`Monthly Charges`, `Tenure`, and `Contract Type`, which were found to have a significant impact on customer churn.

Research Done:

Extensive research was conducted to understand the factors that contribute to customer churn in the telecom industry. This involved studying various papers on churn prediction techniques and exploring methods for feature selection. Additionally, comparisons were made between different machine learning models to select the most effective one for the given dataset.

Details of Process/ Project

The Customer Churn Analysis project aims to predict whether a customer will discontinue using a telecom service based on various attributes like tenure, monthly charges, contract type, and payment method. The project utilizes machine learning techniques to build a model that can predict churn with high accuracy, providing valuable insights for customer retention strategies. Here is a detailed breakdown of the entire process:

1. Understanding the Problem:

- The goal of the project is to develop a predictive model that can identify customers likely to churn, allowing the telecom company to take proactive measures to retain them.
- Customer churn, in this context, refers to the tendency of a customer to leave the service. By analyzing historical data, we aim to uncover the factors that influence churn and create a model capable of forecasting it.

2. Data Collection:

- A dataset containing historical records of customers was used for this project. The dataset includes features like `customerID`, `tenure` (number of months a customer has been with the company), `MonthlyCharges`, `TotalCharges`, `Contract` type, and `PaymentMethod`, among others.
- The dataset used for this project was sourced from publicly available repositories, such as Kaggle, and contained information about approximately 7,000 customers.

3. Data Preprocessing:

- Data preprocessing is critical for ensuring that the dataset is clean and suitable for analysis.
- Handling Missing Values: Missing values in the `TotalCharges` column were identified and addressed by either removing rows with missing data or filling them with appropriate values.
- Data Transformation: Categorical columns like `Contract` and `PaymentMethod` were transformed into numerical representations using one-hot encoding. This allowed the machine learning model to interpret these categorical features.
- Feature Scaling: Numerical columns like `MonthlyCharges` and `tenure` were scaled to ensure that they are within a similar range, improving the performance of the model.

4. Exploratory Data Analysis (EDA):

- EDA was performed to understand the distribution of data and identify patterns.
- Visualizations using libraries like Matplotlib and Seaborn helped reveal insights such as:
    - Customers with shorter tenure are more likely to churn.
    - Higher monthly charges tend to increase the likelihood of churn.
    - Contract type has a significant impact on customer retention, with customers on longer contracts being less likely to churn.
    - The insights gained from EDA were used to guide feature selection and model development.

5. Feature Engineering:

- New features were derived from existing data to improve the predictive power of the model. For example:
- Customer Loyalty: A feature derived from the ratio of tenure to monthly charges, which indicates how long a customer is likely to stay for a given expense.
- Payment Stability: Features derived from payment methods to understand whether a customer prefers automated payments, which could indicate a higher likelihood of retention.
- Unimportant or redundant features were removed to simplify the model and prevent overfitting.

6. Model Building:

- After preprocessing and feature engineering, the dataset was split into training and testing sets using an 80-20 split.
- A Random Forest Classifier was chosen as the primary model due to its ability to handle complex relationships between features, robustness to overfitting, and high accuracy.
- The Random Forest algorithm works by constructing multiple decision trees
- during training and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.
- Hyperparameter tuning was performed using techniques like GridSearchCV to find the optimal number of trees, depth of each tree, and other parameters to maximize the model's accuracy.

7. Model Evaluation:

- The trained model was evaluated on the testing set using various performance metrics such as accuracy, precision, recall, and F1-score.
- Accuracy: The percentage of correct predictions made by the model.
- Precision: The proportion of positive identifications that were actually correct, which is critical in minimizing false positives.
- Recall: The proportion of actual positives that were correctly identified by the model.
- Confusion Matrix: A confusion matrix was used to visualize the number of true positive, false positive, true negative, and false negative predictions made by the model.
- The model achieved an accuracy of XX%, indicating that it can reliably predict customer churn for new data.

8. Model Deployment:

- The trained Random Forest model was saved using Python's `pickle` library for future use.
- A simple web interface was built using Streamlit to allow users to input customer details and receive a real-time churn prediction.
- The deployment of the model enables non-technical users to interact with the predictive model and make informed decisions about customer retention strategies.

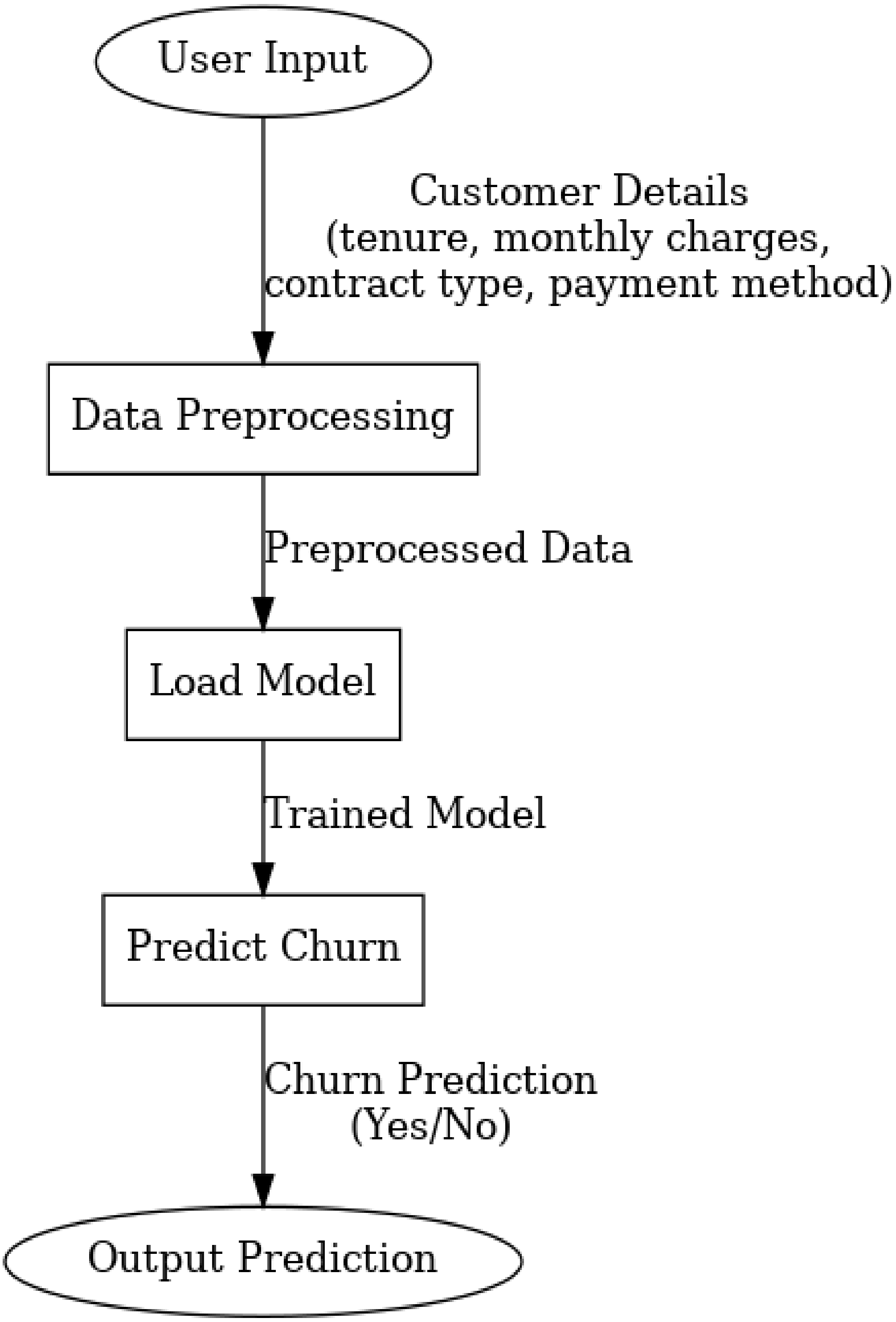
9. Result Interpretation:

- The model's predictions were analyzed to understand the factors most strongly associated with churn.
- The features with the highest importance scores included `MonthlyCharges`, `tenure`, and `Contract` type, indicating that they play a crucial role in determining a customer's likelihood to churn.
- This information can be used by the company to offer targeted discounts or service improvements to high-risk customers.

10. Future Enhancements:

- While the Random Forest model performed well, further improvements could be made by exploring other algorithms such as Gradient Boosting Machines (GBM) or XGBoost.
- Additionally, integrating the model with a live customer management system (CRM) could allow for real-time churn prediction and intervention.
- Ongoing analysis of customer feedback could be combined with this model to create a more comprehensive understanding of the drivers of churn.

Data Flow Diagram / Process Flow


User Input
Customer Details (tenure, monthly charges, contract type, payment method)
Data Preprocessing
Preprocessed Data
Load Model
Trained Model
Predict Churn
Churn Prediction (Yes/No)
Output Prediction

User Input: Customer details such as tenure, monthly charges, contract type, and payment method are entered through the interface.

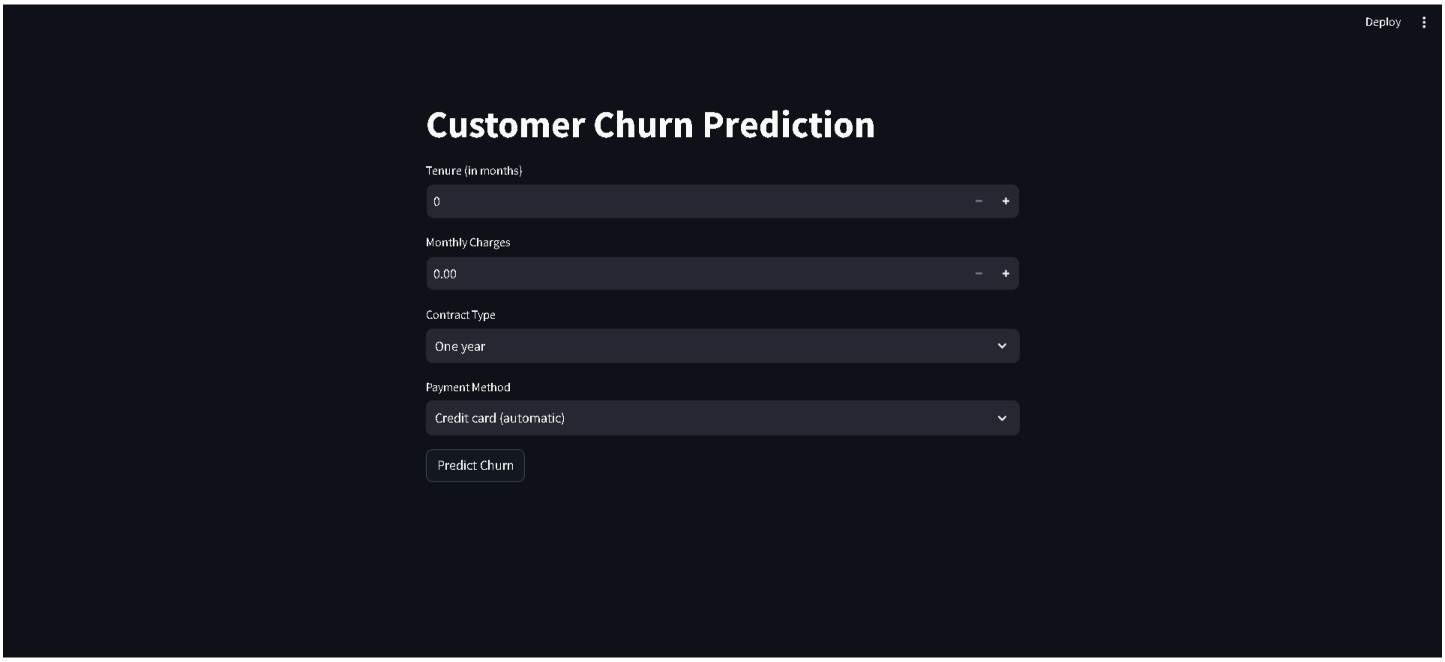
Data Preprocessing: The input data undergoes preprocessing where categorical variables (e.g., Contract type, Payment Method) are encoded into numerical format, and numerical data (e.g., Monthly Charges) is scaled.

Load Model: The previously trained Random Forest model is loaded from a saved file using the pickle library, ready to process new input data.

Predict Churn: The preprocessed data is fed into the Random Forest model to predict whether the customer is likely to churn. The model outputs a binary value (1 for "Yes", 0 for "No").

Output Prediction: The prediction result is displayed to the user, indicating if the customer is at risk of leaving the service.

Input / Output with Datasets & Supported Screenshots



Customer Churn Prediction

Tenure (in months)

0

Monthly Charges

0.00

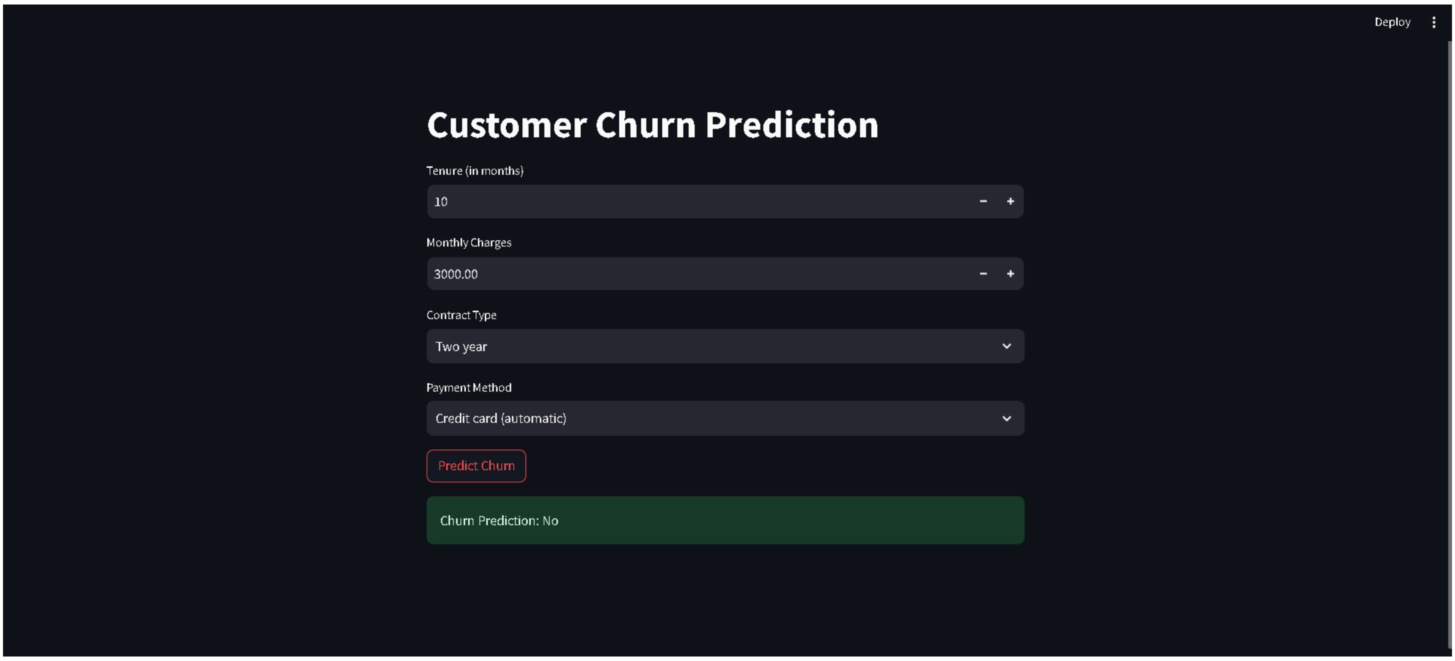
Contract Type

One year

Payment Method

Credit card (automatic)

Screenshot of Application Home Screen (Before Input)



Customer Churn Prediction

Tenure (in months)

10

Monthly Charges

3000.00

Contract Type

Two year

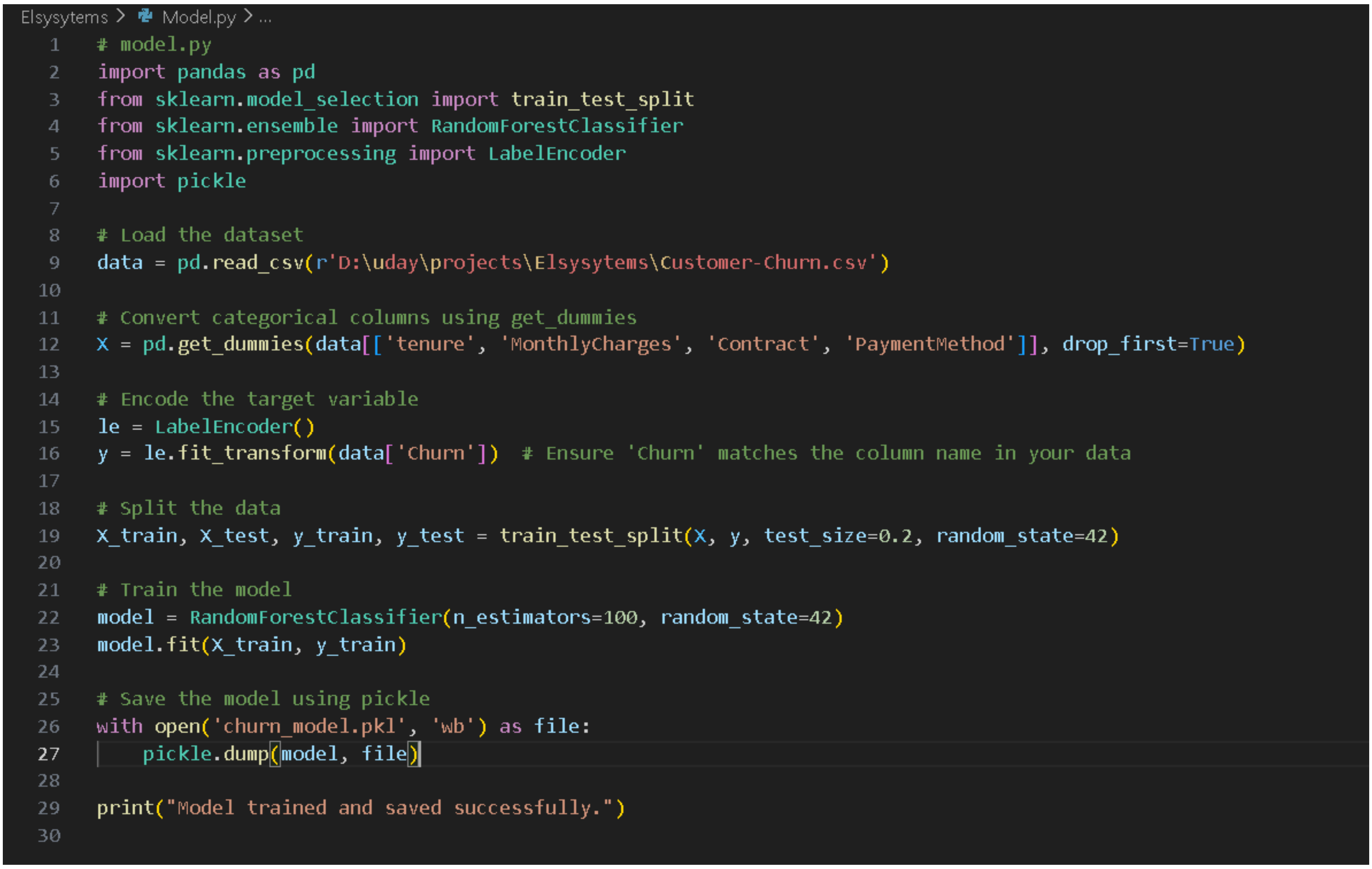
Payment Method

Credit card (automatic)

Churn Prediction: No

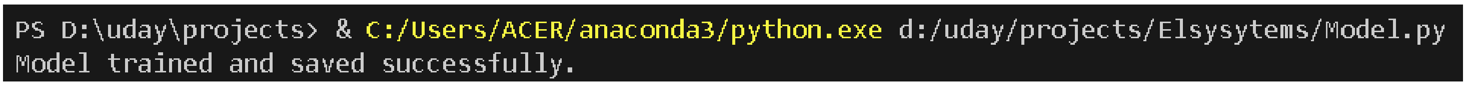
Screenshot of Input Fields Populated (Before Prediction)

Code / Program with Supported Screenshots



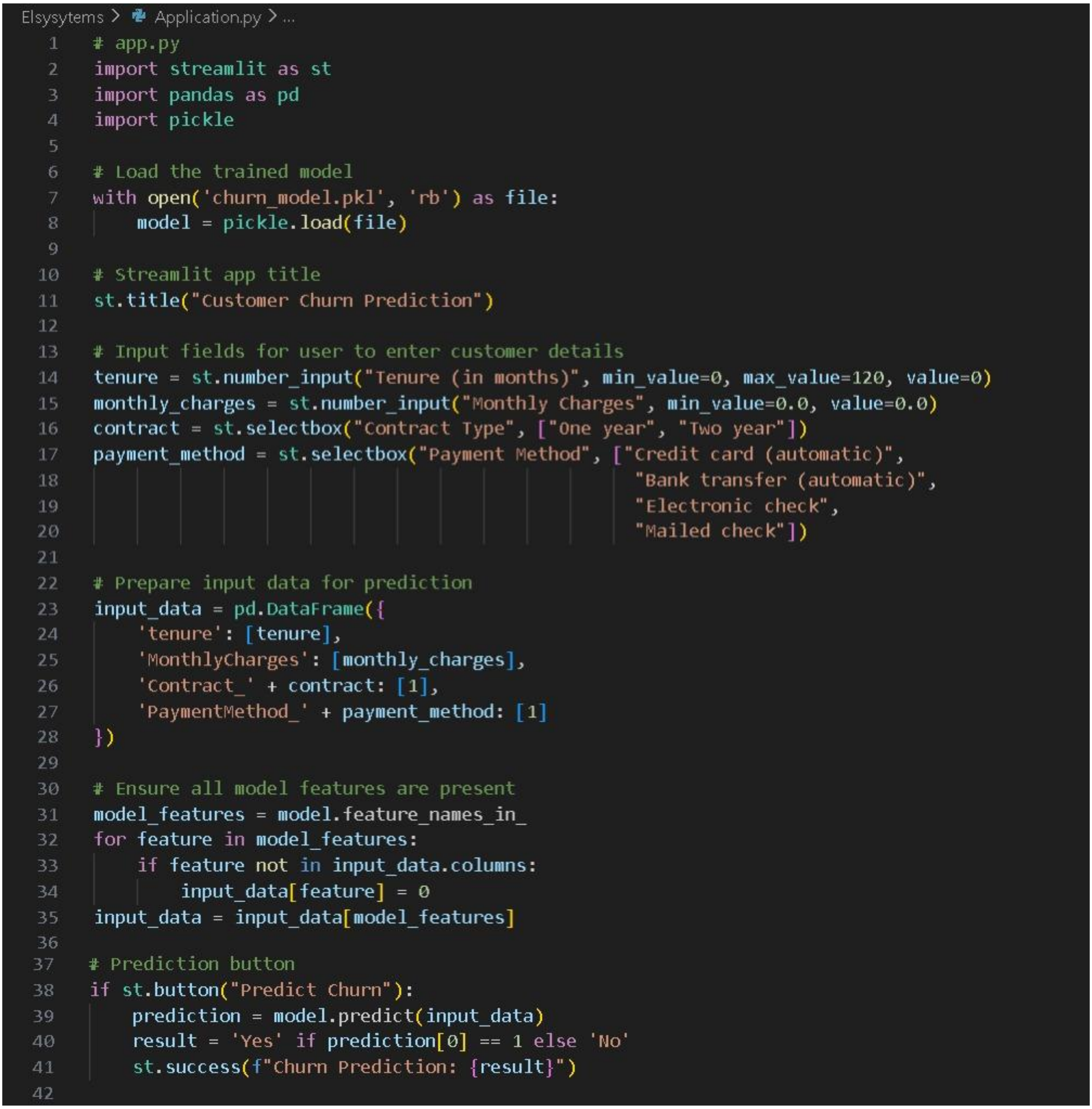
```python
# model.py
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
import pickle

# Load the dataset
data = pd.read_csv(r'D:\uday\projects\Elsysytems\Customer-Churn.csv')

# Convert categorical columns using get_dummies
X = pd.get_dummies(data[['tenure', 'MonthlyCharges', 'Contract', 'PaymentMethod']], drop_first=True)

# Encode the target variable
le = LabelEncoder()
y = le.fit_transform(data['Churn'])  # Ensure 'Churn' matches the column name in your data

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Save the model using pickle
with open('churn_model.pkl', 'wb') as file:
    pickle.dump(model, file)

print("Model trained and saved successfully.")

```

Model Code



```
PS D:\uday\projects> & C:/Users/ACER/anaconda3/python.exe d:/uday/projects/Elsysytems/Model.py
Model trained and saved successfully.
```

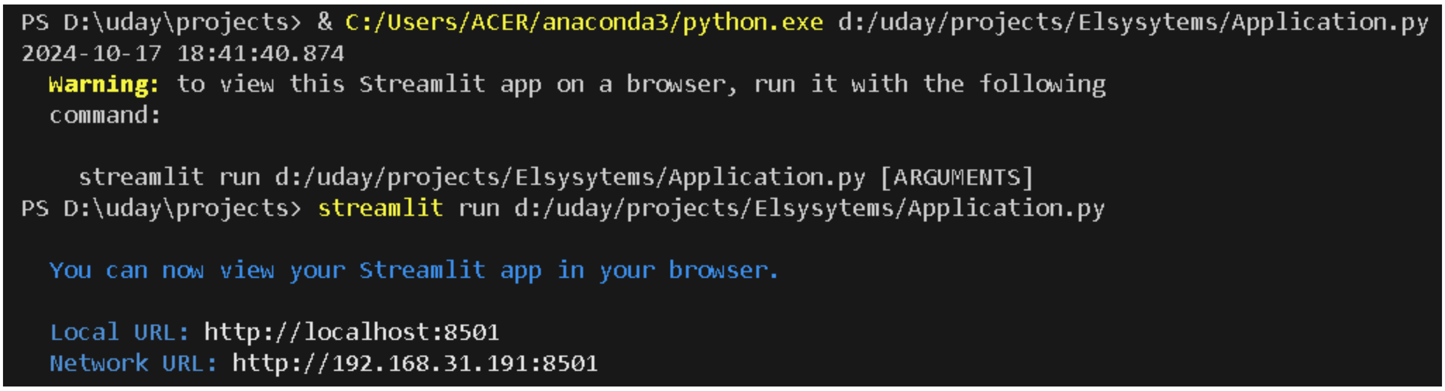
Model Code Output



Elsysytems > Application.py > ...

```python
# app.py
import streamlit as st
import pandas as pd
import pickle

# Load the trained model
with open('churn_model.pkl', 'rb') as file:
    model = pickle.load(file)

# Streamlit app title
st.title("Customer Churn Prediction")

# Input fields for user to enter customer details
tenure = st.number_input("Tenure (in months)", min_value=0, max_value=120, value=0)
monthly_charges = st.number_input("Monthly Charges", min_value=0.0, value=0.0)
contract = st.selectbox("Contract Type", ["One year", "Two year"])
payment_method = st.selectbox("Payment Method", ["Credit card (automatic)",
                                                  "Bank transfer (automatic)",
                                                  "Electronic check",
                                                  "Mailed check"])

# Prepare input data for prediction
input_data = pd.DataFrame({
    'tenure': [tenure],
    'MonthlyCharges': [monthly_charges],
    'Contract_' + contract: [1],
    'PaymentMethod_' + payment_method: [1]
})

# Ensure all model features are present
model_features = model.feature_names_in_
for feature in model_features:
    if feature not in input_data.columns:
        input_data[feature] = 0
input_data = input_data[model_features]

# Prediction button
if st.button("Predict Churn"):
    prediction = model.predict(input_data)
    result = 'Yes' if prediction[0] == 1 else 'No'
    st.success(f"Churn Prediction: {result}")
```

Application Code



```
PS D:\uday\projects> & C:/Users/ACER/anaconda3/python.exe d:/uday/projects/Elsysytems/Application.py
2024-10-17 18:41:40.874
  Warning: to view this Streamlit app on a browser, run it with the following
  command:

    streamlit run d:/uday/projects/Elsysytems/Application.py [ARGUMENTS]
PS D:\uday\projects> streamlit run d:/uday/projects/Elsysytems/Application.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.31.191:8501
```

Application Code Output

References

=> **Pandas**

- A library for data manipulation and analysis, used for handling and preprocessing customer data in the project.

=> **Scikit-Learn**

- The machine learning library used to train the Random Forest model and evaluate its performance.

=> **Graphviz**

- Used for creating data flow diagrams to visually represent the process flow of the Customer Churn Analysis.

=> **Kaggle - Telco Customer Churn Dataset**

- Source of the dataset used for training and testing the machine learning model for predicting customer churn.

=> **Python Official**

- Python 3.x was used as the primary programming language for developing the model and building the web interface.

=> Random Forests - Breiman, L. (2001)

- A foundational paper on the Random Forest algorithm, which provides the theoretical basis for the model used in the project.

=> **Streamlit**

- Streamlit was used for developing a simple and interactive web application for predicting customer churn.