

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

# Load the dataset
def load_data(file_path):
    return pd.read_excel(file_path, engine="xlrd") # Specify x

# Preprocess the dataset
def preprocess_data(df):
    df = df.dropna()
    label_encoders = {}
    for column in df.select_dtypes(include=['object']).columns:
        label_encoders[column] = LabelEncoder()
        df[column] = label_encoders[column].fit_transform(df[column])

    X = df.iloc[:, :-1]
    y = df.iloc[:, -1]

    scaler = StandardScaler()
    X = scaler.fit_transform(X)

    return X, y, scaler, label_encoders

# Train the model
def train_model(X, y):
    X_train, X_test, y_train, y_test = train_test_split(X, y,
    model = RandomForestClassifier(n_estimators=100, random_st
    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)
    print("Accuracy: ", accuracy_score(y_test, y_pred))
```

```
print( Accuracy. , accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_

return model
```

```
pip install xlrd
```



Requirement already satisfied: xlrd in /usr/local/lib/python3.11/dist-packages (2.0.1

```
# Make predictions
def predict(model, scaler, input_data):
    input_data = scaler.transform(input_data)
    return model.predict(input_data)

if __name__ == "__main__":
    file_path = "cm.xls"
    df = load_data(file_path)
    X, y, scaler, label_encoders = preprocess_data(df)
    model = train_model(X, y)

    # Example prediction
    sample_input = np.array([X[0]])
    prediction = predict(model, scaler, sample_input)
    print("Predicted Churn:", prediction)
```

Start coding or [generate](#) with AI.

