Only for Reference and basic understanding of flow

Imagine a telecommunications company that wants to predict whether a customer will churn (leave the service) based on various features such as age, account length, and monthly charges. The company has historical data on customers, including whether they churned or not. Age: (in years), Account_Length: (in months), Monthly_Charges: (in dollars) Churn: Target variable (1 if the customer churned, 0 otherwise) 'Age': [25, 34, 45, 29, 50, 38, 42, 35, 48, 55], 'Account_Length': [12, 24, 36, 18, 48, 30, 42, 24, 36, 60], 'Monthly_Charges': [70, 90, 80, 100, 60, 80, 90, 70, 80, 100], 'Churn': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1] build a logistic regression model using python that can predict the probability of a customer churning by spliting 80% of given data for training. Predict targets on trained model of test data. Print accuracy

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

# Data
data = {
    'Age': [25, 34, 45, 29, 50, 38, 42, 35, 48, 55],
    'Account_Length': [12, 24, 36, 18, 48, 30, 42, 24, 36, 60],
    'Monthly_Charges': [70, 90, 80, 100, 60, 80, 90, 70, 80, 100],
    'Churn': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1]
}

df = pd.DataFrame(data)
X = df[['Age', 'Account_Length', 'Monthly_Charges']]
y = df['Churn']

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

# Model
model = LogisticRegression()
```

```
model.fit(X_train, y_train)


# Predictions
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)


print(accuracy)
```

Build a logistic regression model that can predict whether a customer is likely to churn based on the features in file named customer_churn.csv customer_id: Unique customer ID age: Customer age gender: Customer gender (male/female) account_length: Length of the customer's account (in months) international_plan: Whether the customer has an international plan (yes/no) voice_mail_plan: Whether the customer has a voice mail plan (yes/no) number_vmail_messages: Number of voice mail messages total_day_calls: Total day calls total_night_calls: Total night calls total_intl_calls: Total international calls churn: Whether the customer churned (yes/no)

```
# Load data
df = pd.read_csv("customer_churn.csv")


# Preprocess data
df = pd.get_dummies(df, drop_first=True)


# Split features and target
X = df.drop('churn', axis=1)
y = df['churn']


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


# Model
model = LogisticRegression()
model.fit(X_train, y_train)
```

```
# Predictions and accuracy
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)

print(accuracy)
```