# Import necessary libraries

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.linear\_model import LogisticRegression

from sklearn.ensemble import GradientBoostingClassifier

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

# Load the dataset

# Replace 'your\_dataset.csv' with your actual dataset file

dataset = pd.read\_csv('your\_dataset.csv')

# Separate features (X) and target variable (y)

X = dataset.drop('Churn', axis=1) # Assuming 'Churn' is the target column

y = dataset['Churn']

# Split the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Train a Random Forest classifier

random\_forest\_classifier = RandomForestClassifier(n\_estimators=100, random\_state=42)

random\_forest\_classifier.fit(X\_train, y\_train)

# Make predictions on the test set

predictions\_rf = random\_forest\_classifier.predict(X\_test)

# Train a Logistic Regression classifier

logistic\_regression\_classifier = LogisticRegression()

logistic\_regression\_classifier.fit(X\_train, y\_train)

# Make predictions on the test set

predictions\_lr = logistic\_regression\_classifier.predict(X\_test)

# Train a Gradient Boosting classifier

gradient\_boosting\_classifier = GradientBoostingClassifier(n\_estimators=100, random\_state=42)

gradient\_boosting\_classifier.fit(X\_train, y\_train)

# Make predictions on the test set

predictions\_gb = gradient\_boosting\_classifier.predict(X\_test)

# Evaluate the models

def evaluate\_model(predictions, model\_name):

accuracy = accuracy\_score(y\_test, predictions)

conf\_matrix = confusion\_matrix(y\_test, predictions)

classification\_rep = classification\_report(y\_test, predictions)

print(f'\nResults for {model\_name}:')

print(f'Accuracy: {accuracy:.2f}')

print('\nConfusion Matrix:')

print(conf\_matrix)

print('\nClassification Report:')

print(classification\_rep)

# Evaluate Random Forest model

evaluate\_model(predictions\_rf, 'Random Forest')

# Evaluate Logistic Regression model

evaluate\_model(predictions\_lr, 'Logistic Regression')

# Evaluate Gradient Boosting model

evaluate\_model(predictions\_gb, 'Gradient Boosting')