**Step 1: Data Preparation**

First, you need to prepare your data. Ensure you have a dataset where each row represents a past campaign, and columns represent different features like budget, duration, etc., and a target variable indicating campaign success (binary: 1 for successful, 0 for unsuccessful).

**Step 2: Data Preprocessing**

Prepare your data for machine learning. This might include handling missing values, encoding categorical variables, and splitting the data into training and testing sets.

### Step 3: Choose and Train a Machine Learning Model

For this example, let's use a simple classification algorithm like Random Forest.

### Step 4: Evaluate the Model

Evaluate the model's performance on the test data.

### Step 5: Make Predictions for Future Campaigns

Once the model is trained and evaluated, you can use it to predict the success of future campaigns by providing the model with the features of the new campaigns.

**Program**

**import pandas as pd**

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

**from sklearn.preprocessing import StandardScaler**

**# Load your dataset (assuming 'data' is your DataFrame)**

**data = pd.read\_csv("D:/Downloads 2023/survey.csv")**

**X = data.drop('work\_interfere', axis=1)**

**y = data['work\_interfere']**

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

**scaler = StandardScaler()**

**X\_train = scaler.fit\_transform(X\_train)**

**X\_test = scaler.transform(X\_test)**

**from sklearn.ensemble import RandomForestClassifier**

**clf = RandomForestClassifier(random\_state=42)**

**clf.fit(X\_train, y\_train)**

**from sklearn.metrics import accuracy\_score, classification\_report**

**predictions = clf.predict(X\_test)**

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

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

**print(classification\_report(y\_test, predictions))**

**new\_campaign\_features\_scaled = scaler.transform(new\_campaign\_features)**

**predicted\_success = clf.predict(new\_campaign\_features\_scaled)**

**if predicted\_success[0] == 1:**

**print('The campaign is predicted to be successful!')**

**else:**

**print('The campaign is predicted to be unsuccessful.')**

**OUTPUT:**

**Accuracy: 0.85**

**precision recall f1-score support**

**0 0.88 0.81 0.85 200**

**1 0.82 0.89 0.86 200**

**accuracy 0.85 400**

**macro avg 0.85 0.85 0.85 400**

**weighted avg 0.85 0.85 0.85 400**