By Nitish Adhikari

Email id :nitishbuzzpro@gmail.com (mailto:nitishbuzzpro@gmail.com), +91-9650740295

Linkedin : https://www.linkedin.com/in/nitish-adhikari-6b2350248 (https://www.linkedin.com/in/nitish-adhikari-6b2350248)

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.preprocessing import LabelEncoder
```

In [90]:
```
# load the data
df = pd.read_excel('Data_Science_Internship.xlsx')
df.head()
```

Out[90]:

	Unnamed: 0	Agent_id	status	lost_reason	budget	lease	movein	
0	0.0	1deba9e96f404694373de9749ddd1ca8aa7bb823145a6f...	LOST	Not responding	NaN	NaN	NaT	9b2d5b4678781e53038e91ea532453(
1	1.0	299ae77a4ef350ae0dd37d6bba1c002d03444fb1edb236...	LOST	Low budget	NaN	NaN	NaT	9b2d5b4678781e53038e91ea532453(
2	2.0	c213697430c006013012dd2aca82dd9732aa0a1a6bca13...	LOST	Not responding	£121 - £180 Per Week	Full Year Course Stay 40 - 44 weeks	2022-08-31	7aae3e886e89fc1187a5c47d6cea1c2
3	3.0	eac9815a500f908736d303e23aa227f0957177b0e6756b...	LOST	Low budget	0-0	0.0	NaT	ba2d0a29556ac20f86f45e4543c0825
4	4.0	1deba9e96f404694373de9749ddd1ca8aa7bb823145a6f...	LOST	Junk lead	NaN	NaN	NaT	9b2d5b4678781e53038e91ea532453(

In [91]:
```
# drop unnecessary columns
df.drop(['lead_id', 'Unnamed: 0'], axis=1, inplace=True)
```

In [92]:
```
# replace unique NaN value with actual NaN
df.replace('9b2d5b4678781e53038e91ea5324530a03f27dc1d0e5f6c9bc9d493a23be9de0', np.nan, inplace=True)
```

In [93]:
```
# drop rows with missing status values
df.dropna(subset=['status'], inplace=True)
```

In [94]:
```
# encode categorical columns
le = LabelEncoder()
for col in df.columns:
    if df[col].dtype == 'object':
        df[col] = df[col].astype(str)
        df[col] = le.fit_transform(df[col])
```

In [95]:
```
# select relevant features
X = df.drop(['status', 'movein'], axis=1)
y = df['status']
```

In [96]:
```
# split the data 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)
```

In [97]:
```
# train a logistic regression model
clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)
```

Out[97]:
```
▾ DecisionTreeClassifier

DecisionTreeClassifier()
```

In [98]:
```
# make predictions on the test set
y_pred = clf.predict(X_test)
```

In [102]:
```
# evaluate the model performance
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Precision:", precision_score(y_test, y_pred, average='weighted'))
print("F1 score:", f1_score(y_test, y_pred, average='weighted'))
```

```
Accuracy: 0.9908817850246728
Precision: 0.9906241934569052
F1 score: 0.9907208292143868
```