```python
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as ply
data = pd.read_csv("/content/adult_dataset.csv")
print(data)
```

```
       age workclass  fnlwgt     education  education.num
marital.status  \
0       90         ?   77053       HS-grad              9
Widowed
1       82   Private  132870       HS-grad              9
Widowed
2       66         ?  186061  Some-college             10
Widowed
3       54   Private  140359       7th-8th              4
Divorced
4       41   Private  264663  Some-college             10
Separated
...    ...       ...     ...           ...            ...
...
32556   22   Private  310152  Some-college             10       Never-
married
32557   27   Private  257302    Assoc-acdm             12  Married-
civ-spouse
32558   40   Private  154374       HS-grad              9  Married-
civ-spouse
32559   58   Private  151910       HS-grad              9
Widowed
32560   22   Private  201490       HS-grad              9       Never-
married

              occupation   relationship   race     sex
capital.gain  \
0                      ?  Not-in-family  White  Female             0

1        Exec-managerial  Not-in-family  White  Female             0

2                      ?      Unmarried  Black  Female             0

3      Machine-op-inspct      Unmarried  White  Female             0

4         Prof-specialty      Own-child  White  Female             0

...                  ...            ...    ...     ...           ...

32556    Protective-serv  Not-in-family  White    Male             0

32557       Tech-support           Wife  White  Female             0
```

```
32558  Machine-op-inspct        Husband  White    Male             0

32559       Adm-clerical      Unmarried  White  Female             0

32560       Adm-clerical      Own-child  White    Male             0


       capital.loss  hours.per.week native.country income
0              4356              40  United-States  <=50K
1              4356              18  United-States  <=50K
2              4356              40  United-States  <=50K
3              3900              40  United-States  <=50K
4              3900              40  United-States  <=50K
...             ...             ...            ...    ...
32556             0              40  United-States  <=50K
32557             0              38  United-States  <=50K
32558             0              40  United-States   >50K
32559             0              40  United-States  <=50K
32560             0              20  United-States  <=50K

[32561 rows x 15 columns]
```

```python
from sklearn.preprocessing import LabelEncoder
for column in data:
    encoder = LabelEncoder()
    data[column] = encoder.fit_transform(data[column])

print(data)
```

```
       age  workclass  fnlwgt  education  education.num
marital.status  \
0        72          0    2649         11              8
6
1        65          4    6514         11              8
6
2        49          0   11175         15              9
6
3        37          4    7009          5              3
0
4        24          4   16850         15              9
5
...    ...        ...     ...        ...            ...             ..
.
32556    5          4   18560         15              9
4
32557   10          4   16528          7             11
2
32558   23          4    8080         11              8
2
32559   41          4    7883         11              8
```

```
6
32560    5          4   12881         11              8
4

       occupation  relationship  race  sex  capital.gain  capital.loss
\
0               0             1     4    0             0            91

1               4             1     4    0             0            91

2               0             4     2    0             0            91

3               7             4     4    0             0            90

4              10             3     4    0             0            90

...           ...           ...   ...  ...           ...           ...

32556          11             1     4    1             0             0

32557          13             5     4    0             0             0

32558           7             0     4    1             0             0

32559           1             4     4    0             0             0

32560           1             3     4    1             0             0


       hours.per.week  native.country  income
0                  39              39       0
1                  17              39       0
2                  39              39       0
3                  39              39       0
4                  39              39       0
...               ...             ...     ...
32556              39              39       0
32557              37              39       0
32558              39              39       1
32559              39              39       0
32560              19              39       0

[32561 rows x 15 columns]

X = data.drop('income', axis=1)
y = data['income']

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)
```

```python
# Initialize and train the Decision Tree model
clf = DecisionTreeClassifier(random_state=42,max_depth=15)
clf.fit(X_train, y_train)

# Make predictions and evaluate the model
y_pred = clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
```

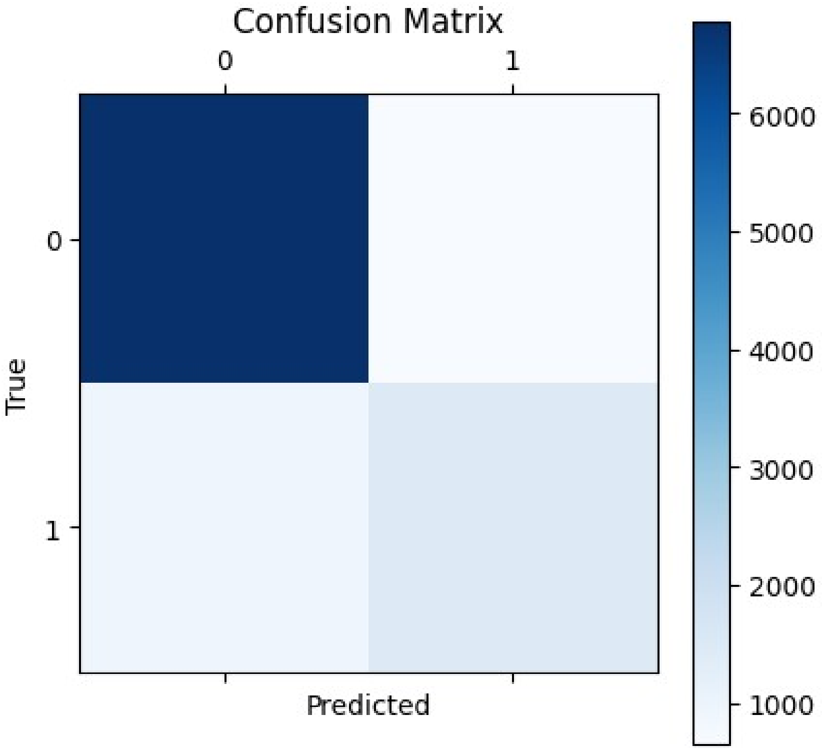
```
Accuracy: 0.84
```

```python
from sklearn.metrics import accuracy_score, confusion_matrix,
precision_score, recall_score, f1_score
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

# Print the results
print(f"Accuracy: {accuracy}")
print(f"Confusion Matrix:\n{conf_matrix}")
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"F1 Score: {f1}")
```

```
Accuracy: 0.8405159176988433
Confusion Matrix:
[[6781  648]
 [ 910 1430]]
Precision: 0.8353258505260778
Recall: 0.8405159176988433
F1 Score: 0.8371687607011491
```

```python
import matplotlib.pyplot as plt
plt.matshow(conf_matrix, cmap=plt.cm.Blues)
plt.title("Confusion Matrix")
plt.colorbar()
plt.xlabel("Predicted")
plt.ylabel("True")
plt.show()
```


Confusion Matrix
0
1
0
1
True
Predicted
6000
5000
4000
3000
2000
1000