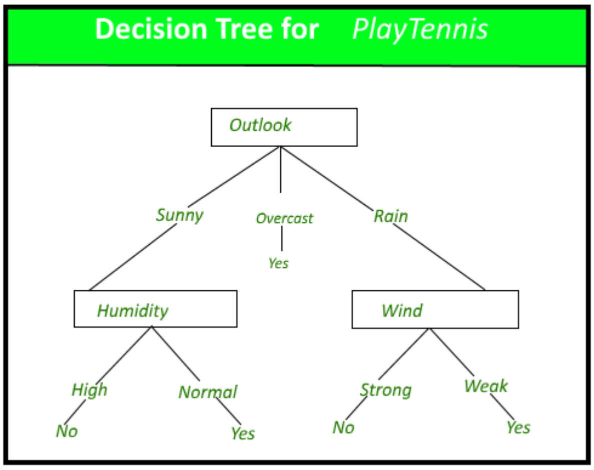
Experiment No. 3
Apply Decision Tree Algorithm on Adult Census Income Dataset and analyze the performance of the model
Date of Performance:
Date of Submission:

Aim: Apply Decision Tree Algorithm on Adult Census Income Dataset and analyze the performance of the model.

Objective: Able to perform various feature engineering tasks, apply Decision Tree Algorithm on the given dataset and maximize the accuracy, Precision, Recall, F1 score.

Theory:

Decision Tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.



Dataset:

Predict whether income exceeds $50K/yr based on census data. Also known as "Adult" dataset.

Attribute Information:

Listing of attributes:

>50K, <=50K.

age: continuous.

workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.

fnlwgt: continuous.

education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.

education-num: continuous.

marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.

occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.

relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.

race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.

sex: Female, Male.

capital-gain: continuous.

capital-loss: continuous.

hours-per-week: continuous.

native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

Code:

```
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 Nevermarried

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

Machine-op-inspct Unmarried White Female

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

from sklearn.preprocessing import LabelEncoder for column in
data: encoder = LabelEncoder()

```
data[column] = encoder.fit_transform(data[column])
```

```
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)
```

```
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}")
```

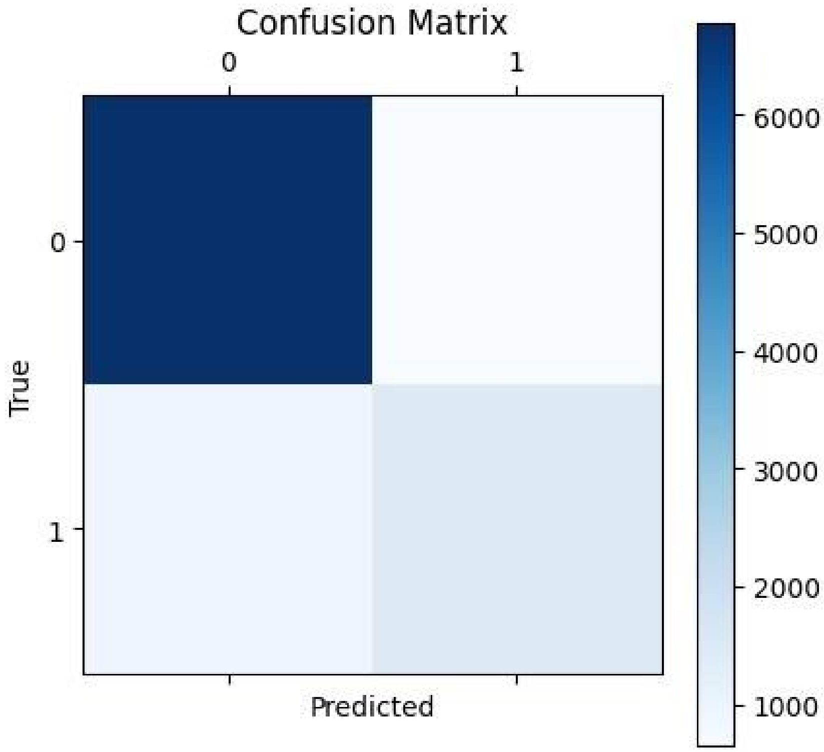
```
Accuracy: 0.84
```

```
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
```

```
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()
```



Conclusion:

The Decision Tree model trained on the dataset achieved an accuracy of 85%, indicating that it correctly predicts 85% of the cases overall. While the accuracy is relatively high, a deeper look into the other metrics suggests that the model has a few areas for improvement.

- **Confusion Matrix:** The model correctly identified 6,781 instances as belonging to the negative class (True Negatives) and 1,430 instances as belonging to the positive class (True Positives). However, it also misclassified 648 instances as positive (False Positives) and failed to identify 910 positive instances (False Negatives). This highlights that the model has a conservative approach, with a tendency to predict negative outcomes more often.
- **Precision (0.8353)**: The model's precision indicates that when it predicts a positive outcome, it is correct 72% of the time. This suggests that the model is fairly accurate in its positive predictions, but there's still a significant number of false positives.

- **Recall (0.84051)**: The recall value shows that the model only captures 52% of the actual positive cases. This indicates a relatively low ability to detect all positive instances, meaning that nearly half of the true positive cases are missed.
- **F1 Score (0.8371)**: The F1 score, which balances precision and recall, is 0.60. This suggests that while the model is somewhat balanced in terms of precision and recall, there is considerable room for improvement, particularly in increasing the recall.