```
# Lab 6: Decision Tree Algorithm
# 0. Installation and Imports

# Uncomment to install ucimlrepo if not already installed
!pip install ucimlrepo

from ucimlrepo import fetch_ucirepo
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import matplotlib.pyplot as plt
```

```
Collecting ucimlrepo
  Downloading ucimlrepo-0.0.7-py3-none-any.whl.metadata (5.5 kB)
Requirement already satisfied: pandas>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from ucimlrepo) (2.2.2)
Requirement already satisfied: certifi>=2020.12.5 in /usr/local/lib/python3.12/dist-packages (from ucimlrepo) (2025.1
Requirement already satisfied: numpy>=1.26.0 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.0.0->ucimlrep
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.0.0-
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.0.0->ucimlrepo
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.0.0->ucimlre
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->panda
Downloading ucimlrepo-0.0.7-py3-none-any.whl (8.0 kB)
Installing collected packages: ucimlrepo
Successfully installed ucimlrepo-0.0.7
```

```
# Lab 6: Decision Tree Algorithm
# 1. Load Dataset and Inspect

contraceptive_method_choice = fetch_ucirepo(id=30)

X = contraceptive_method_choice.data.features
y = contraceptive_method_choice.data.targets

print("Metadata:\n", contraceptive_method_choice.metadata)
print("\nVariables:\n", contraceptive_method_choice.variables)

print("Sample features:\n", X.head())
print("Sample targets:\n", y.head())
```

```
Metadata:
 {'uci_id': 30, 'name': 'Contraceptive Method Choice', 'repository_url': 'https://archive.ics.uci.edu/dataset/30/cont

Variables:
                         name     role         type      demographic  \
0                  wife_age  Feature      Integer              Age
1                  wife_edu  Feature  Categorical  Education Level
2               husband_edu  Feature  Categorical  Education Level
3              num_children  Feature      Integer            Other
4             wife_religion  Feature       Binary            Other
5              wife_working  Feature       Binary       Occupation
6        husband_occupation  Feature  Categorical       Occupation
7  standard_of_living_index  Feature  Categorical             None
8            media_exposure  Feature       Binary             None
9      contraceptive_method   Target  Categorical             None

  description units missing_values
0        None  None             no
1        None  None             no
2        None  None             no
3        None  None             no
4        None  None             no
5        None  None             no
6        None  None             no
7        None  None             no
8        None  None             no
9        None  None             no
Sample features:
    wife_age  wife_edu  husband_edu  num_children  wife_religion  wife_working  \
0        24         2            3             3              1             1
1        45         1            3            10              1             1
2        43         2            3             7              1             1
3        42         3            2             9              1             1
4        36         3            3             8              1             1

   husband_occupation  standard_of_living_index  media_exposure
0                   2                         3               0
1                   3                         4               0
2                   3                         4               0
3                   3                         3               0
4                   3                         2               0
Sample targets:
    contraceptive_method
0                      1
1                      1
2                      1
3                      1
4                      1
```

```
# Lab 6: Decision Tree Algorithm
# 2. Train-Test Split

X_train, X_test, y_train, y_test = train_test_split(
    X, y.values.ravel(), test_size=0.2, random_state=42
)

print("Training set size:", X_train.shape)
print("Testing set size:", X_test.shape)
```

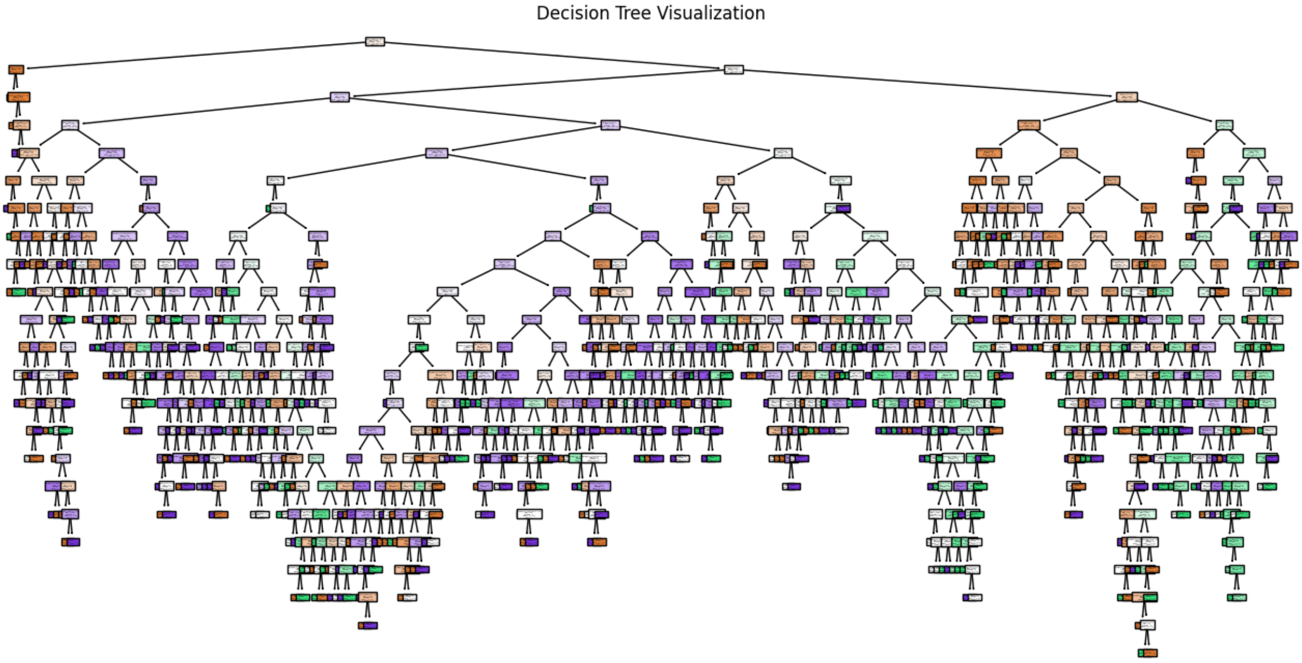
```
Training set size: (1178, 9)
Testing set size: (295, 9)
```

```
# Lab 6: Decision Tree Algorithm
# 3. Decision Tree Training

dtree = DecisionTreeClassifier(random_state=42)
dtree.fit(X_train, y_train)
y_pred = dtree.predict(X_test)
```

```
# Lab 6: Decision Tree Algorithm
# 4. Evaluation and Output

print("Accuracy Score:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

```
Accuracy Score: 0.511864406779661
Classification Report:
               precision    recall  f1-score   support

           1       0.62      0.62      0.62       130
           2       0.42      0.38      0.40        71
           3       0.44      0.47      0.45        94

    accuracy                           0.51       295
   macro avg       0.49      0.49      0.49       295
weighted avg       0.51      0.51      0.51       295

Confusion Matrix:
 [[80 19 31]
 [18 27 26]
 [32 18 44]]
```

```
# Lab 6: Decision Tree Algorithm
# 5. Visualize the Decision Tree

plt.figure(figsize=(16, 8))
plot_tree(dtree, feature_names=X.columns, class_names=[str(c) for c in dtree.classes_], filled=True)
plt.title("Decision Tree Visualization")
plt.show()
```



Start coding or generate with AI.