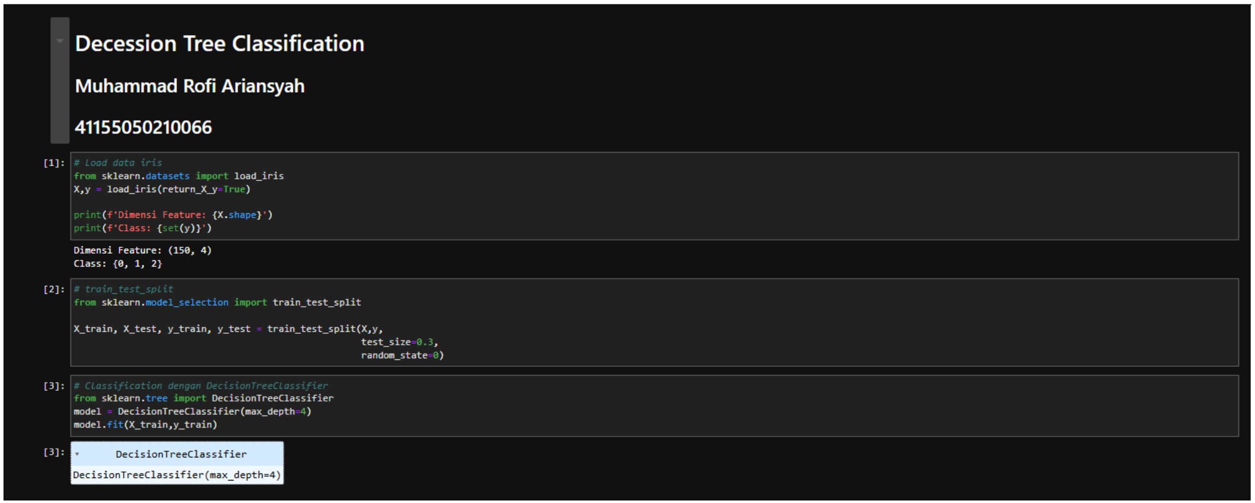


Decession Tree Classification

Muhammad Rofi Ariansyah

41155050210066

[1]:
```
# Load data iris
from sklearn.datasets import load_iris
X,y = load_iris(return_X_y=True)

print(f'Dimensi Feature: {X.shape}')
print(f'Class: {set(y)}')
```

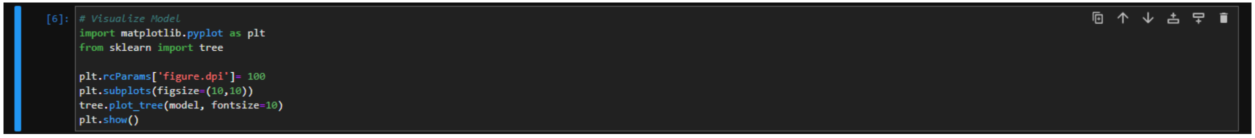
```
Dimensi Feature: (150, 4)
Class: {0, 1, 2}
```

[2]:
```
# train_test_split
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,y,
                                                    test_size=0.3,
                                                    random_state=0)
```

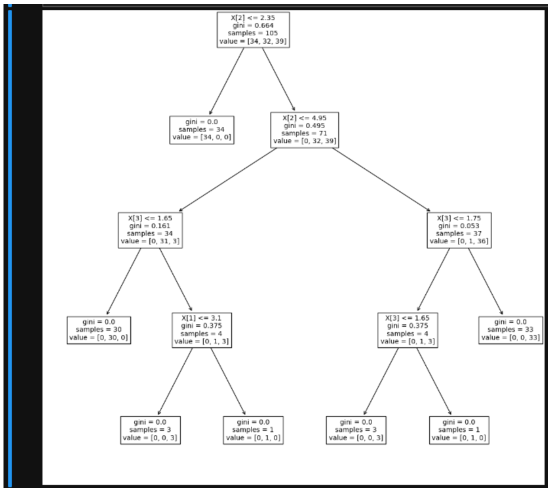
[3]:
```
# Classification dengan DecisionTreeClassifier
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier(max_depth=4)
model.fit(X_train,y_train)
```

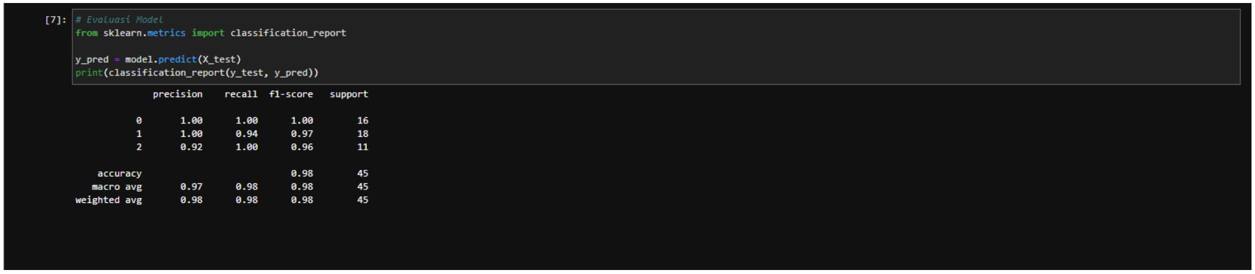
[3]:
```
▾      DecisionTreeClassifier
DecisionTreeClassifier(max_depth=4)
```



[6]:
```
# Visualize Model
import matplotlib.pyplot as plt
from sklearn import tree

plt.rcParams['figure.dpi']= 100
plt.subplots(figsize=(10,10))
tree.plot_tree(model, fontsize=10)
plt.show()
```





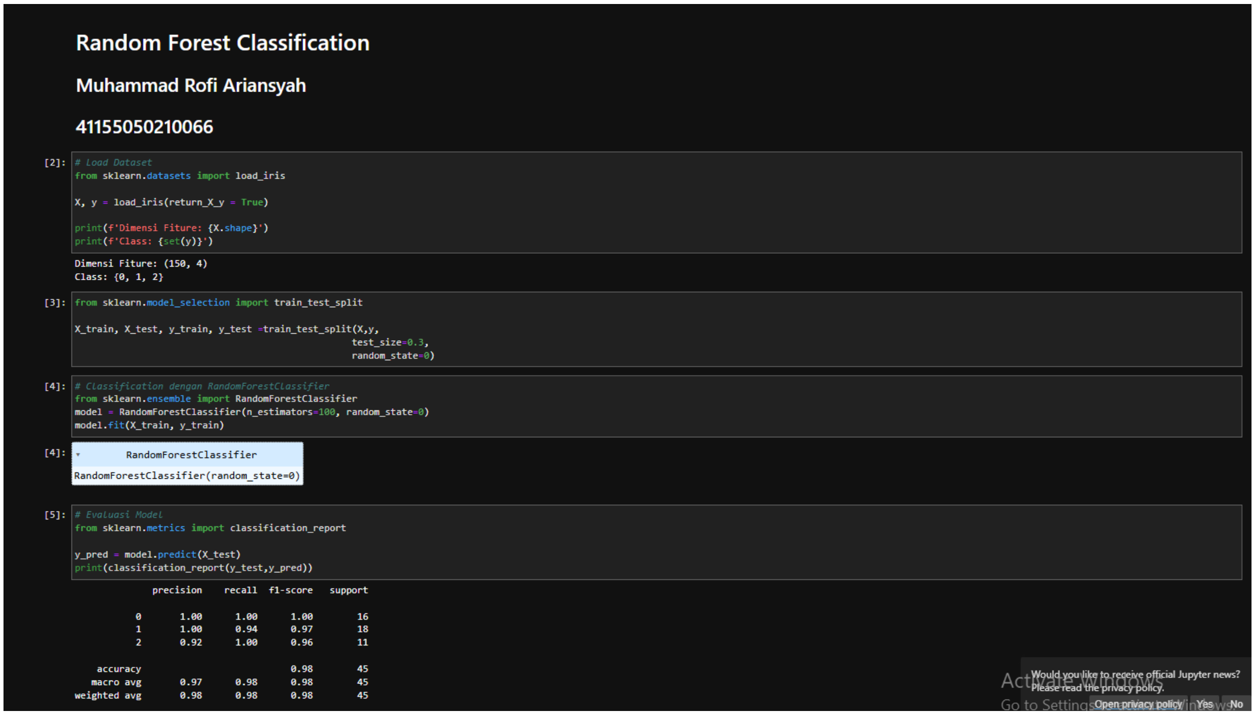
[7]:
```
# Evaluasi Model
from sklearn.metrics import classification_report

y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        16
           1       1.00      0.94      0.97        18
           2       0.92      1.00      0.96        11

    accuracy                           0.98        45
   macro avg       0.97      0.98      0.98        45
weighted avg       0.98      0.98      0.98        45
```



Random Forest Classification

Muhammad Rofi Ariansyah

41155050210066

[2]:

```
# Load Dataset
from sklearn.datasets import load_iris

X, y = load_iris(return_X_y = True)

print(f'Dimensi Fiture: {X.shape}')
print(f'Class: {set(y)}')
```

```
Dimensi Fiture: (150, 4)
Class: {0, 1, 2}
```

[3]:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test =train_test_split(X,y,
                                                   test_size=0.3,
                                                   random_state=0)
```

[4]:

```
# Classification dengan RandomForestClassifier
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=100, random_state=0)
model.fit(X_train, y_train)
```

[4]:

```
▾        RandomForestClassifier
RandomForestClassifier(random_state=0)
```

[5]:

```
# Evaluasi Model
from sklearn.metrics import classification_report

y_pred = model.predict(X_test)
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        16
           1       1.00      0.94      0.97        18
           2       0.92      1.00      0.96        11

    accuracy                           0.98        45
   macro avg       0.97      0.98      0.98        45
weighted avg       0.98      0.98      0.98        45
```

Would you like to receive official Jupyter news?
Please read the privacy policy.