



# SIMPLE CLASSIFICATION OF IRIS PLANTS USING DECISION TREE CLASSIFIER

DIMAS TRIPANGESTU

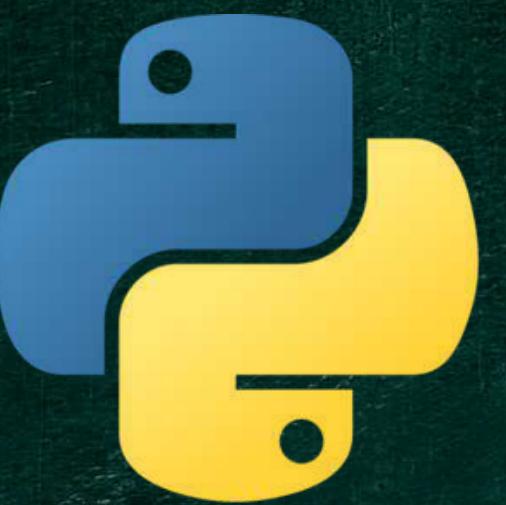


# INTRODUCTION AND OBJECTIVE



The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

# TOOLS USED:



matplotlib

# CONTENT

01.

Input Data

02.

EXPLORATORY  
DATA ANALYSIS  
(EDA)

03.

Data Modelling

04.

Data  
Visualization

# INPUT DATA

```
import pandas as pd
from sklearn import datasets

# memuat datasets iris dari scikit-learn dan mengonversinya menjadi dataframe
iris = datasets.load_iris()

x = iris.data    # inputan untuk machine learning
y = iris.target  # output yang diinginkan dari machine learning

# mengonversi datafitur dan target menjadi dataframe
df_x = pd.DataFrame(x, columns=iris.feature_names)
df_y = pd.Series(y, name='target')

#gabungkan fitur dan target dalam satu frame
df = pd.concat([df_x, df_y], axis=1)

df.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	grid icon	bar chart icon
0	5.1	3.5	1.4	0.2	0		
1	4.9	3.0	1.4	0.2	0		
2	4.7	3.2	1.3	0.2	0		
3	4.6	3.1	1.5	0.2	0		
4	5.0	3.6	1.4	0.2	0		

# EXPLORATORY DATA ANALYSIS (EDA)

```
▶ df.info()
```

```
→ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   sepal length (cm)    150 non-null   float64
 1   sepal width (cm)     150 non-null   float64
 2   petal length (cm)    150 non-null   float64
 3   petal width (cm)     150 non-null   float64
 4   target              150 non-null   int64  
dtypes: float64(4), int64(1)
memory usage: 6.0 KB
```

```
[ ] df['target'].unique()
```

```
→ array([0, 1, 2])
```

# EXPLORATORY DATA ANALYSIS (EDA)

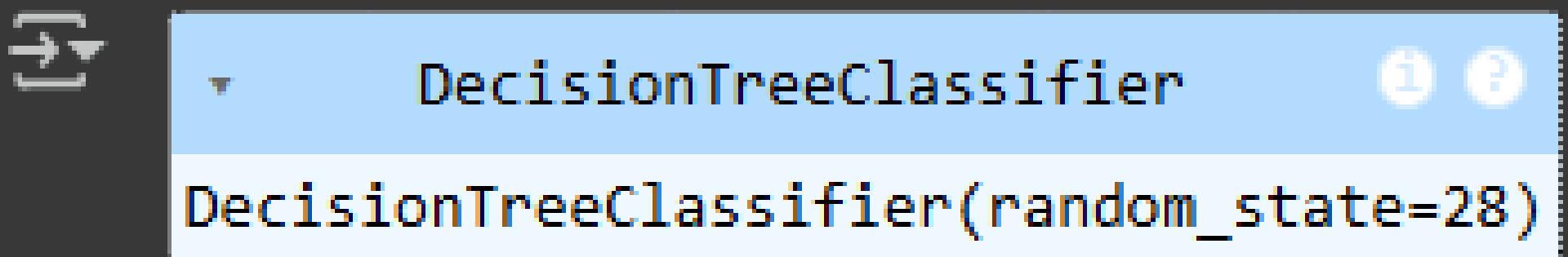
```
[ ] df.describe()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333	1.000000
std	0.828066	0.435866	1.765298	0.762238	0.819232
min	4.300000	2.000000	1.000000	0.100000	0.000000
25%	5.100000	2.800000	1.600000	0.300000	0.000000
50%	5.800000	3.000000	4.350000	1.300000	1.000000
75%	6.400000	3.300000	5.100000	1.800000	2.000000
max	7.900000	4.400000	6.900000	2.500000	2.000000

# DATA MODELLING

```
[ ] from sklearn.model_selection import train_test_split  
  
# membagi data menjadi data train dan test  
x_train, x_test, y_train, y_test = train_test_split(df_x, df_y, test_size=0.2, random_state=28)
```

```
[ ] from sklearn.tree import DecisionTreeClassifier  
  
#membuat dan melatih model decission tree  
model = DecisionTreeClassifier(random_state=28)  
model.fit(x_train, y_train)
```



# DATA MODELLING

```
▶ from sklearn.metrics import accuracy_score  
  
#memprediksi dan mengevaluasi  
y_pred = model.predict(x_test)  
  
accuracy = accuracy_score(y_test, y_pred)  
  
print("laporan klasifikasi")  
print(f"akurasi: {accuracy * 100:.2f}%")
```

```
→ laporan klasifikasi  
akurasi: 96.67%
```

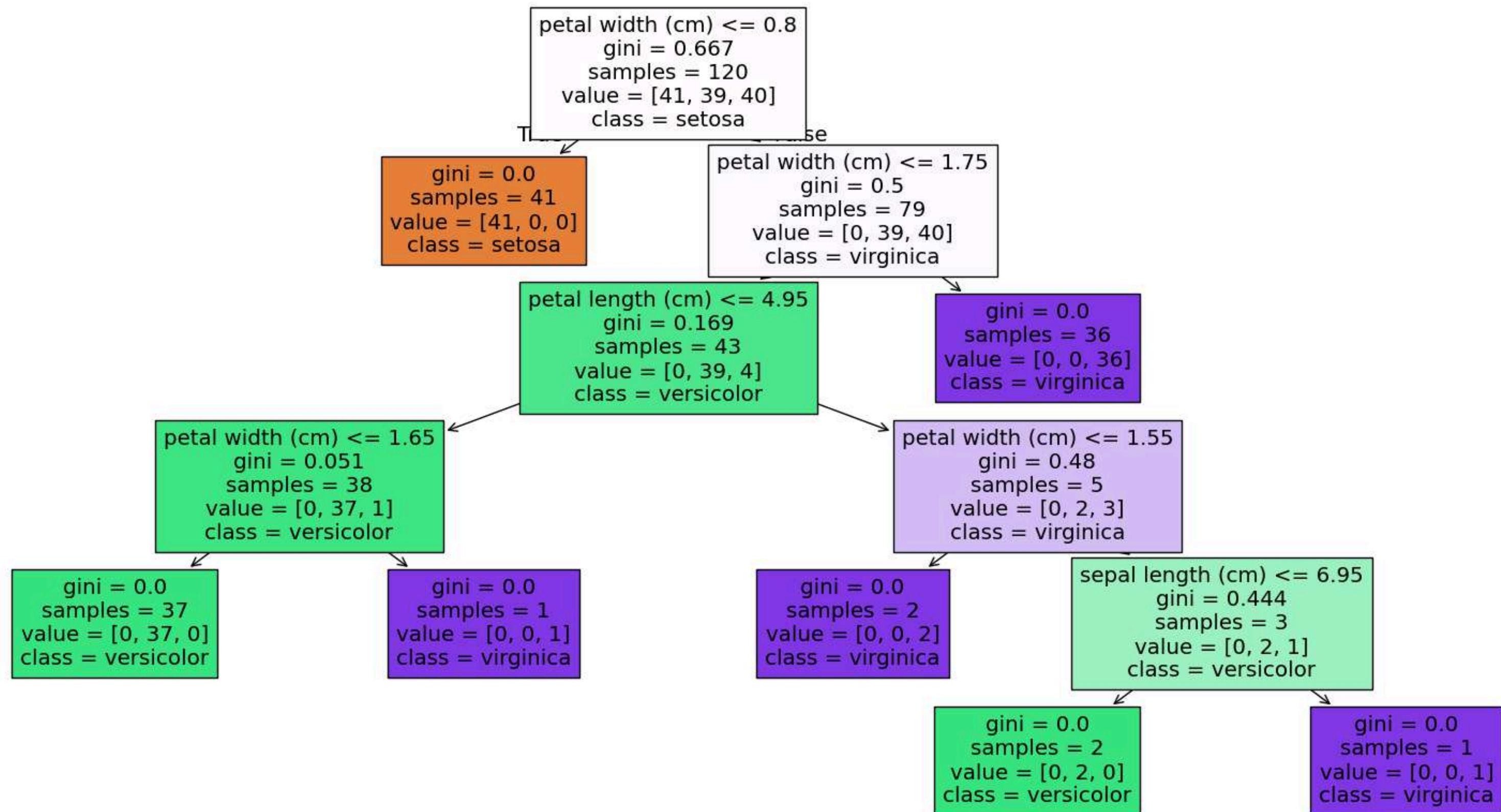
# DATA VISUALIZATION

```
[ ] import matplotlib.pyplot as plt
from sklearn import tree

#visualisasi decission tree
plt.figure(figsize=(20,10))
tree.plot_tree(model,
                feature_names=iris.feature_names,
                class_names=iris.target_names,
                filled = True)

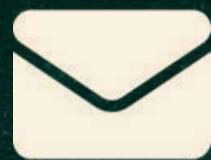
plt.show()
```

# DATA VISUALIZATION



# THANK YOU VERY MUCH!

IF YOU HAVE ANY QUESTIONS, SUGGESTIONS OR FEEDBACKS,  
PLEASE DO NOT HESITATE TO REACH ME THROUGH THE  
CONTACTS BELOW:



dimatripangestu2803@gmail.com



<https://github.com/DimasTripangestu>



[www.linkedin.com/in/dimas-tripangestu-6b75b4331](https://www.linkedin.com/in/dimas-tripangestu-6b75b4331)