

iris_dt-classification

February 18, 2024

1 Decision Tree Classification using Iris Dataset

1.0.1 By Matindi Steve - github.com/stevemats/DT_Classification

```
[1]: # Essential libs import
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.tree import plot_tree
```

```
[ ]:
```

```
[ ]:
```

```
[2]: # Loading the iris dataset to a variable
iris = load_iris()
```

```
[ ]:
```

1.0.2 Data exploration point

```
[3]: df = pd.DataFrame(iris.data, columns=iris.feature_names) # Dataset conversion
      ↪ into a pandas DF
```

```
[4]: df['target'] = iris.target # Target variable addition to the DataFrame
```

```
[ ]:
```

```
[5]: print("Summary of the Iris Dataset:")
      print(df.describe())
```

Summary of the Iris Dataset:

	sepal length (cm)	sepal width (cm)	petal length (cm)	\
count	150.000000	150.000000	150.000000	
mean	5.843333	3.057333	3.758000	

std	0.828066	0.435866	1.765298
min	4.300000	2.000000	1.000000
25%	5.100000	2.800000	1.600000
50%	5.800000	3.000000	4.350000
75%	6.400000	3.300000	5.100000
max	7.900000	4.400000	6.900000

	petal width (cm)	target
count	150.000000	150.000000
mean	1.199333	1.000000
std	0.762238	0.819232
min	0.100000	0.000000
25%	0.300000	0.000000
50%	1.300000	1.000000
75%	1.800000	2.000000
max	2.500000	2.000000

```
[ ]:
```

```
[6]: # Display the first few rows of the dataset
print("\nFirst few rows of the Iris Dataset:")
print(df.head())
```

First few rows of the Iris Dataset:

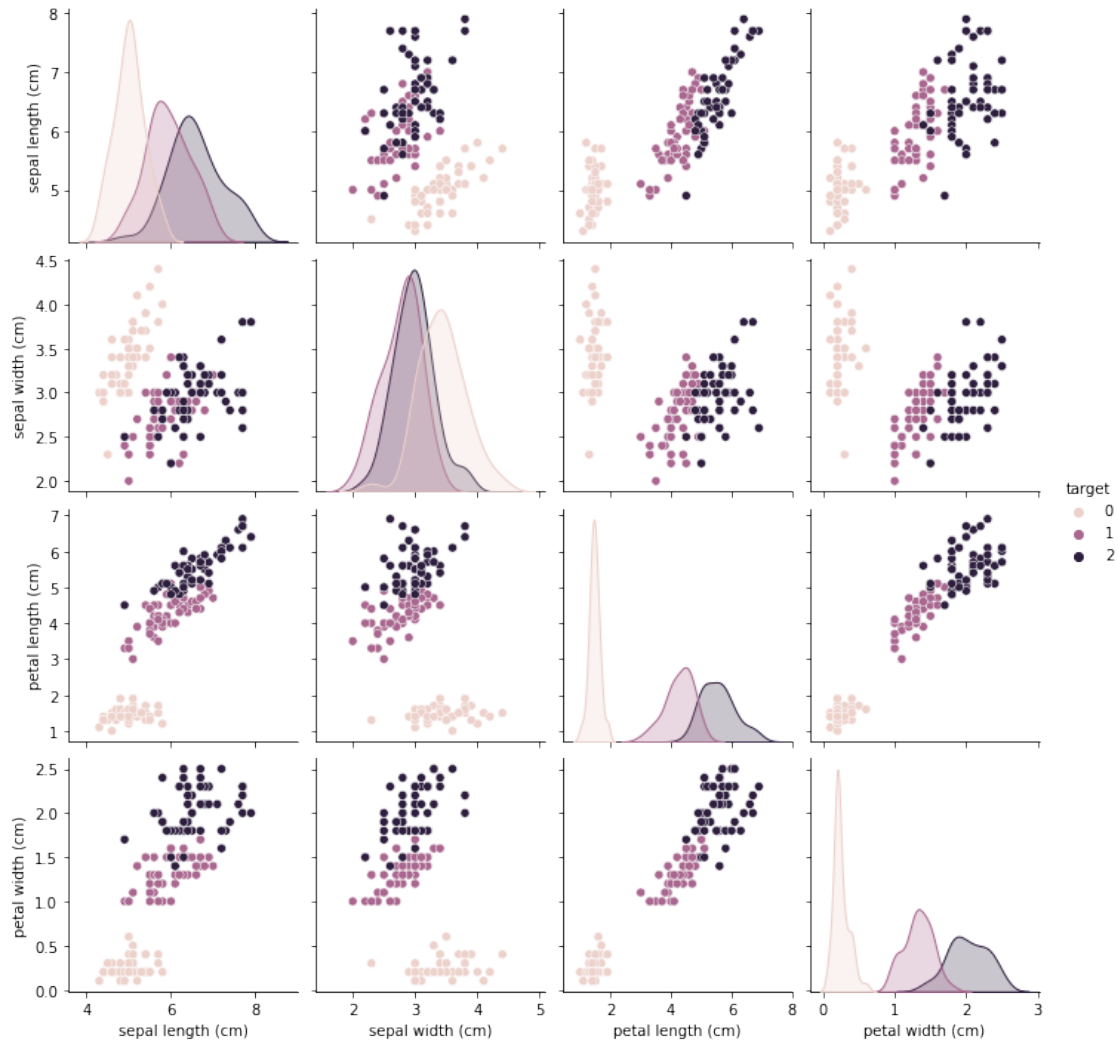
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	\
0	5.1	3.5	1.4	0.2	
1	4.9	3.0	1.4	0.2	
2	4.7	3.2	1.3	0.2	
3	4.6	3.1	1.5	0.2	
4	5.0	3.6	1.4	0.2	

	target
0	0
1	0
2	0
3	0
4	0

```
[ ]:
```

```
[7]: sns.pairplot(df, hue='target') # Dataset visualization using pair plots
```

```
[7]: <seaborn.axisgrid.PairGrid at 0x1adc5b39550>
```



[]:

1.0.3 Splitting the dataset into training and testing sets

```
[8]: X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target,
↳ test_size=0.2, random_state=42)
```

[]:

1.0.4 Building & Training DT

```
[9]: clf = DecisionTreeClassifier(random_state=42)
      clf.fit(X_train, y_train)
```

```
[9]: DecisionTreeClassifier(random_state=42)
```

```
[ ]:
```

1.0.5 Predicting the target values for the test set

```
[10]: y_pred = clf.predict(X_test)
```

```
[ ]:
```

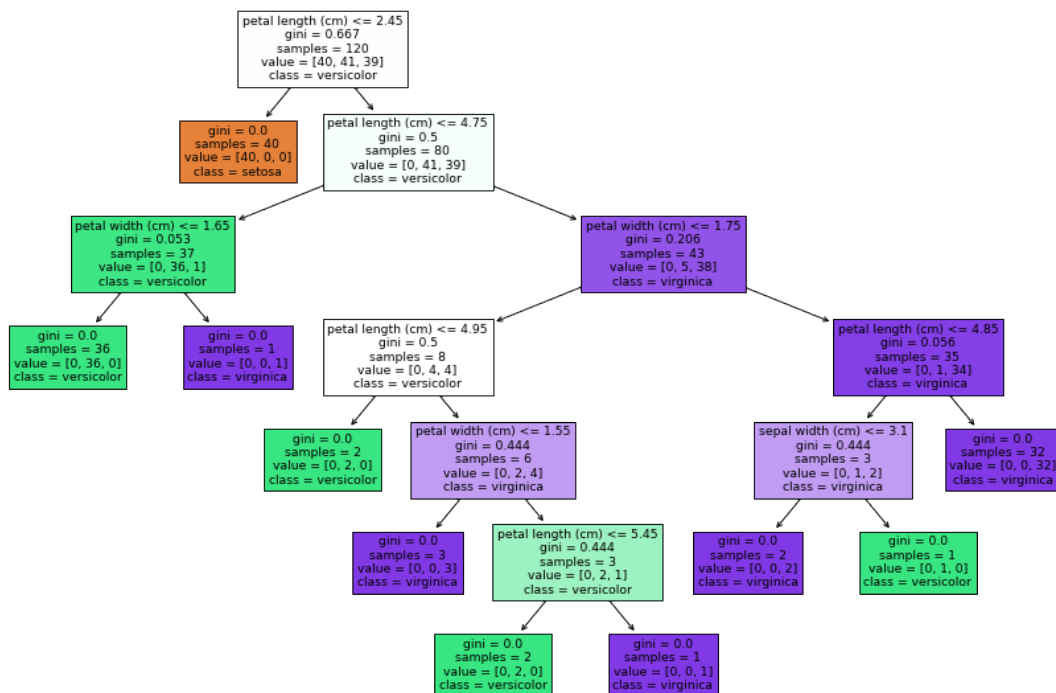
1.0.6 Calculating accuracy score of the model

```
[11]: accuracy = accuracy_score(y_test, y_pred)
```

```
[ ]:
```

1.0.7 Visualizing the Decision Tree

```
[12]: plt.figure(figsize=(15,10))  
plot_tree(clf, feature_names=iris.feature_names, class_names=iris.target_names,  
         ↪filled=True)  
plt.show()
```



```
[ ]:
```

1.0.8 Accuracy of the model

```
[13]: print(f"Accuracy: {accuracy*100:.2f}%")
```

Accuracy: 100.00%

```
[ ]:
```