

Network Login exp2.ipynb (auto-D) - Jupyter ChatGPT 21AIC301J-Lab AI-A Welcome To Colab - Colab

Not secure 10.1.38.19/user/ra2311047010033/lab/workspaces/auto-D/tree/DTL/exp2.ipynb

File Edit View Run Kernel Tabs Settings Help

exp2.ipynb +

Filter files by name / DLT /

Name	Last Modified
exp2.ipynb	last month
exp3.ipynb	last month
exp4.ipynb	last month
exp5.ipynb	14 days ago
exp6.ipynb	13 days ago
exp7.ipynb	28 minutes ago
Untitled.ipynb	13 days ago

```
[2]: 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, confusion_matrix, classification_report
from sklearn import tree
import matplotlib.pyplot as plt

iris = load_iris()
X = iris.data
y = iris.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)

dtc = DecisionTreeClassifier(criterion='gini', random_state=1)
dtc.fit(X_train, y_train)

y_pred = dtc.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred, target_names=iris.target_names))

plt.figure(figsize=(12, 8))
tree.plot_tree(dtc, filled=True, feature_names=iris.feature_names, class_names=iris.target_names)
plt.title("Decision Tree Visualization")
plt.show()
```

Accuracy: 0.9666666666666667

Simple 0 \$ 1 Python 3 (ipykernel) | Idle Mem: 150.20 MB Mode: Command ✓ Ln 1, Col 1 exp2.ipynb 0 Type here to search ENG US 10:28 23-09-2025

Network Login exp2.ipynb (auto-D) Inbox (3,125) - ds107 ChatGPT 21AIC301J- Lab AI-A Welcome To Colab -

Not secure 10.1.38.19/user/ra2311047010033/lab/workspaces/auto-D/tree/DLT/exp2.ipynb

File Edit View Run Kernel Tabs Settings Help

+

Filter files by name

/ DLT /

Name	Last Modified
exp2.ipynb	last month
exp3.ipynb	last month
exp4.ipynb	last month
exp5.ipynb	14 days ago
exp6.ipynb	13 days ago
exp7.ipynb	29 minutes ago
Untitled.ipynb...	13 days ago

exp2.ipynb

Accuracy: 0.9666666666666667

Confusion Matrix:

```
[[11  0  0]
 [ 0 12  1]
 [ 0  0  6]]
```

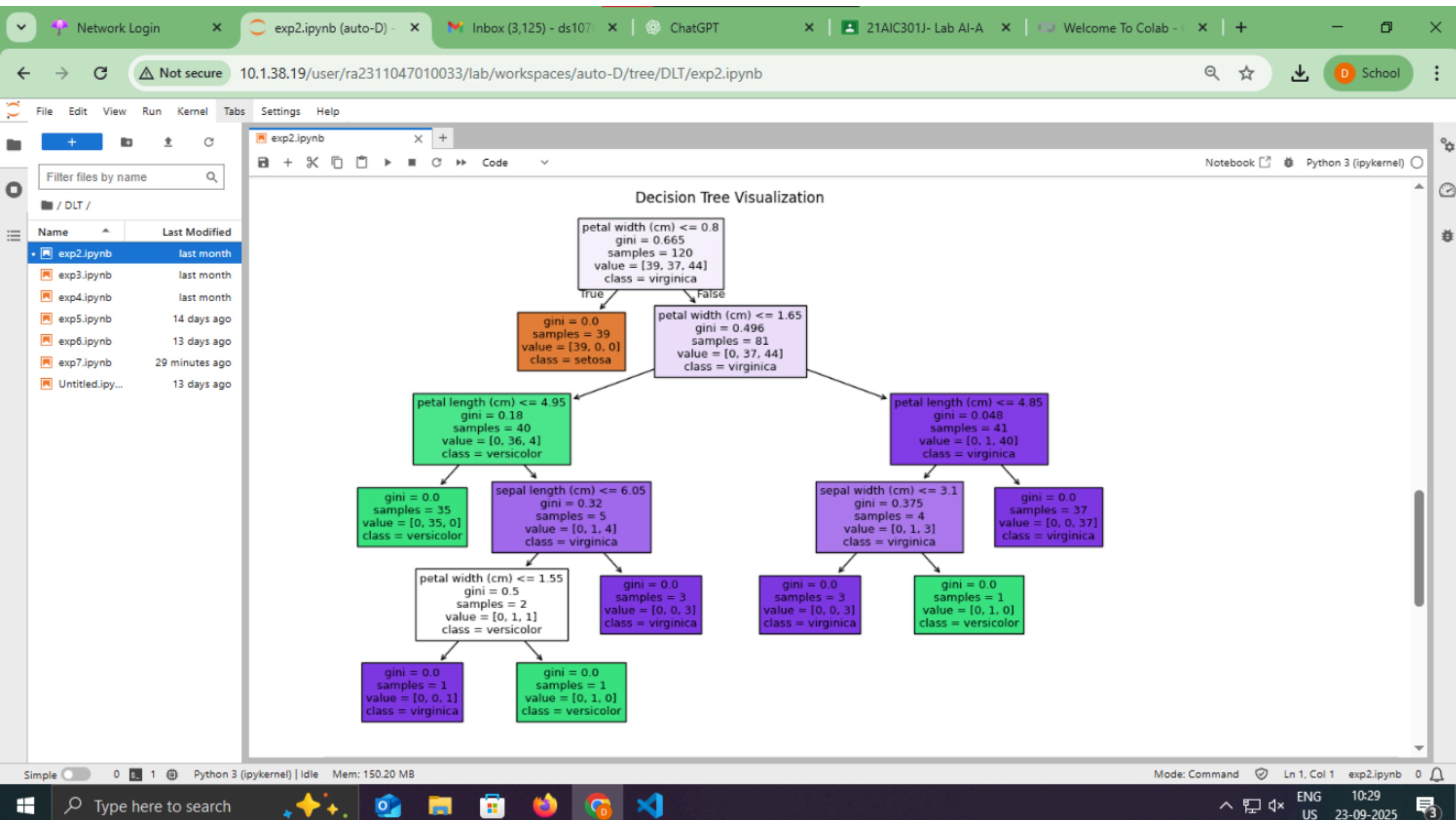
Classification Report:

	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	11
versicolor	1.00	0.92	0.96	13
virginica	0.86	1.00	0.92	6
accuracy			0.97	30
macro avg	0.95	0.97	0.96	30
weighted avg	0.97	0.97	0.97	30

Decision Tree Visualization

```
graph TD
    Root["petal width (cm) <= 0.8  
gini = 0.665  
samples = 120  
value = [39, 37, 44]  
class = virginica"] -- True --> L["gini = 0.0  
samples = 39  
value = [39, 0, 0]  
class = setosa"]
    Root -- False --> R["petal width (cm) <= 1.65  
gini = 0.496  
samples = 81  
value = [0, 37, 44]  
class = virginica"]
```

Simple 0 \$ 1 Python 3 (ipykernel) | Idle Mem: 150.20 MB Mode: Command 0 Ln 1, Col 1 exp2.ipynb 0 Type here to search ENG US 10:28 23-09-2025



31092

2. Implement a classifier using open-source data set.
⇒ with Decision tree.

Aim:- To implement and evaluate a Decision tree classifier using the Iris data set.

Objective:-

- to understand the concept of decision tree classifier.
- to load and preprocess the Iris data set.
- to split the dataset into training and testing sets.
- to implement the Decision tree algorithm.
- to evaluate model performance using metrics such as accuracy, confusion matrix and classification report.

Pseudo code:-

1. Import necessary libraries
2. Load the Iris data set
3. Split the dataset into training and testing sets.
4. Train a Decision Tree classifier.
5. Predict the output for the test data
6. Evaluate the model using accuracy and confusion matrix

```
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,  
confusion_matrix, classification_report  
from sklearn import tree  
import matplotlib.pyplot as plt  
import matplotlib.pyplot as plt
```

#1. Load the dataset

```
iris = load_iris()  
X = iris.data  
y = iris.target
```

#2. Split the data into training and test sets.

```
x_train, x_test, y_train, y_test = train_test_split(X, y,  
test_size=0.2, random_state=1)
```

#3. Train the decision tree classifier.

```
dtc = DecisionTreeClassifier(criterion='gini',  
random_state=1)
```

dtc.fit(x_train, y_train)

#4. Predict on test data.

```
y_pred = dtc.predict(x_test)
```

#5. Evaluate.

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

classification report (X-test, X-pred,
target_names = iris.target - names))

6. optional:- visualize the decision tree.

plt.figure(figsize=(12, 8))

tree.plot_tree(dt, filled=True)

feature_names = iris.feature_names,

class_names = iris.target_names)

plt.title("decision tree visualization")

plt.show()

Observation:-

Test Size.

Observation:- 30 samples (20%)

Accuracy :- ~96.67% (may vary slightly)

misclassified samples :- 1 sample (if any)

Tree Splits :- Based on petal length and width

Visualization :- Tree is easy to interpret.

Result :-

→ Decision tree classifier correctly classified the iris flower species with an accuracy of 96.67% on the test data.

→ The model is interpretable and effective for small datasets like Iris.

→ Tree visualization helps understand the decision-making process.

accuracy :- 96.67

confusion matrix

$$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 12 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 6 \end{bmatrix}$$

classification Report:-

	Precision	recall	f1score	support
setosa	1.00	1.00	1.00	11
Versicolor	1.00	0.92	0.96	13
virginica	0.86	1.00	0.92	6

	Precision	recall	f1score	support
accuracy	1.00	1.00	0.97	30
macroavg	0.95	0.97	0.96	30
weightedavg	0.97	0.97	0.97	30