

Library 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, plot_tree
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
```

Load Dataset:

```
# Load dataset
iris = load_iris()

# Features
print(iris.feature_names)

# Labels
print(iris.target_names)
```

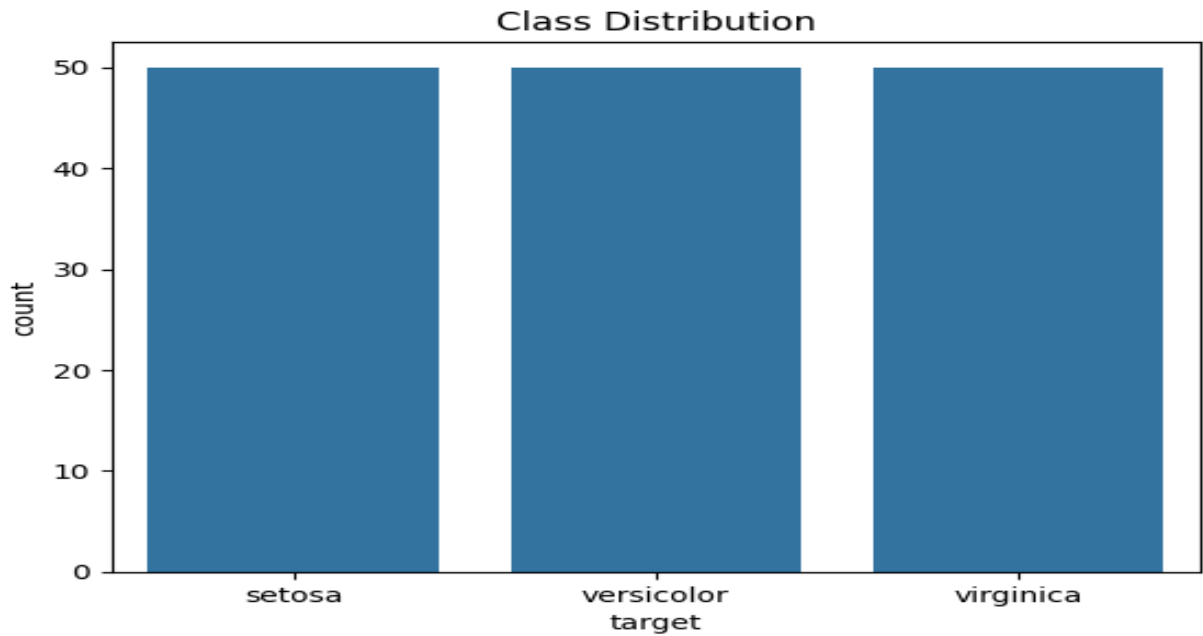
Convert dataframe:

```
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target
df.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
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

Visualization dataset:

```
sns.countplot(x='target', data=df)
plt.title("Class Distribution")
plt.xticks(ticks=[0, 1, 2], labels=iris.target_names)
plt.show()
```



Split Dataset into Train & Test:

```
X = df.drop('target', axis=1)
y = df['target']

# Split into train and test sets (80/20 split)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

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

Training size: (120, 4)
Testing size: (30, 4)
```

Train the Model:

```
model = DecisionTreeClassifier(random_state=42)
model.fit(X_train, y_train)
```

Evaluate Model:

```
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))

# Classification Report
print("\nClassification Report:")
print(classification_report(y_test, y_pred,
target_names=iris.target_names))
```

Accuracy: 0.9333333333333333

Classification Report:

	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	10
versicolor	0.90	0.90	0.90	10
virginica	0.90	0.90	0.90	10
accuracy			0.93	30
macro avg	0.93	0.93	0.93	30
weighted avg	0.93	0.93	0.93	30

Confusion Metrics:

```
cm = confusion_matrix(y_test, y_pred)

# Plot
sns.heatmap(cm, annot=True, cmap="Blues", fmt="d",
            xticklabels=iris.target_names,
            yticklabels=iris.target_names)
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

