✅ Dataset Loaded Successfully!

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

```
# Features and target
X = data.drop("species", axis=1)
y = data["species"]

# Split dataset: 80% training, 20% testing
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

print("Training samples:", X_train.shape[0])
print("Testing samples:", X_test.shape[0])
```

```
Training samples: 120
Testing samples: 30
```

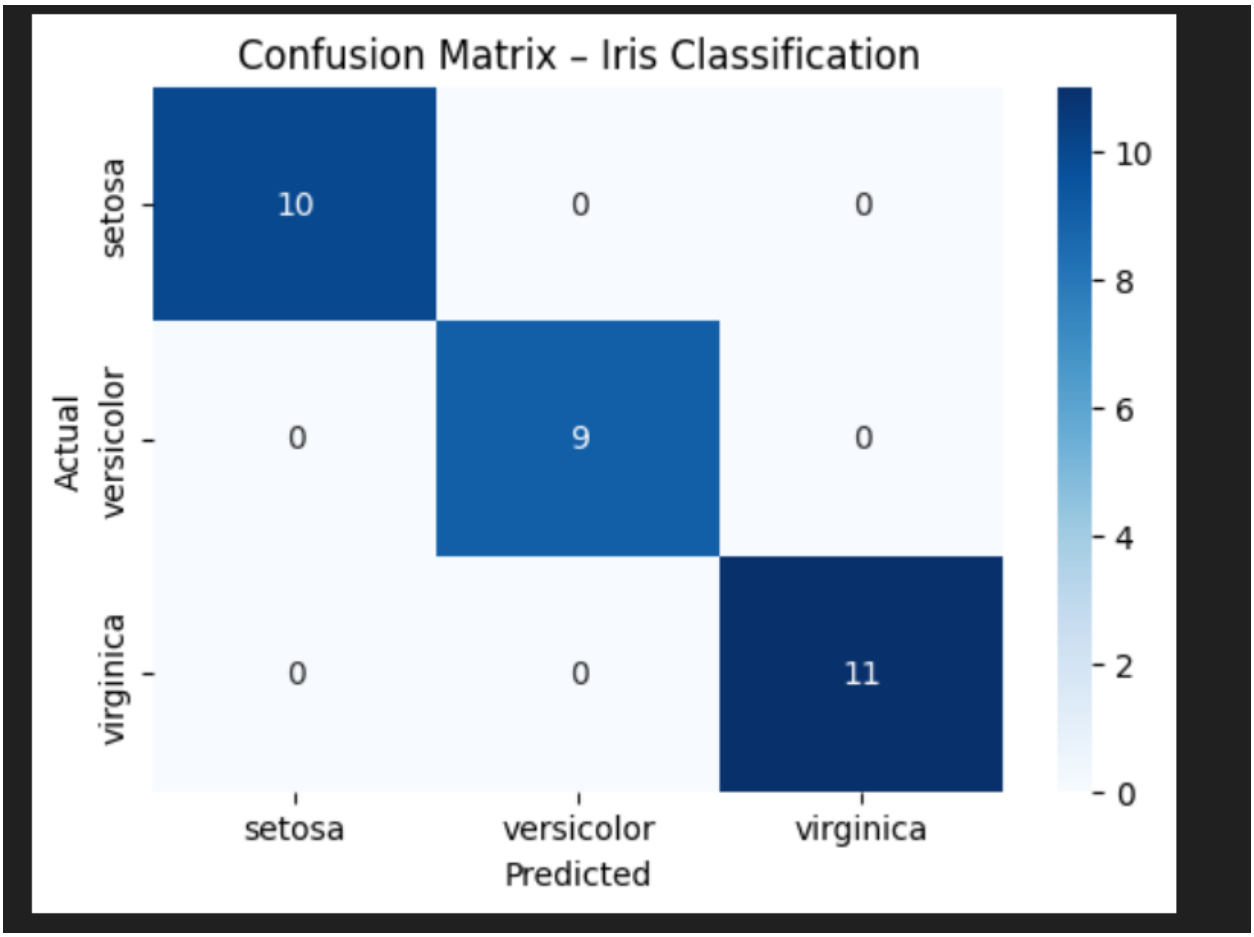
```
# Initialize Decision Tree Classifier
model = DecisionTreeClassifier(criterion="entropy", random_state=42)

# Train the model
model.fit(X_train, y_train)

print("🌳 Model training complete!")
```

```
🌳 Model training complete!
```

✅ Model Accuracy: 100.00%

📊 Classification Report:

	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	10
versicolor	1.00	1.00	1.00	9
virginica	1.00	1.00	1.00	11
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30



```
# Save predictions for reporting
results = X_test.copy()
results["Actual"] = y_test.values
results["Predicted"] = y_pred
results.to_csv("iris_predictions.csv", index=False)

print("💾 Predictions saved to iris_predictions.csv")
```

```
💾 Predictions saved to iris_predictions.csv
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
Training samples: (60000, 28, 28)
Testing samples: (10000, 28, 28)
✅ Data normalized and reshaped successfully!
```

```
/usr/local/lib/python3.12/dist-packages/keras/src/layers/convolutional/base_conv.py:113: Us
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_1 (Conv2D)	(None, 11, 11, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 5, 5, 64)	0
flatten (Flatten)	(None, 1600)	0
dense (Dense)	(None, 128)	204,928
dense_1 (Dense)	(None, 10)	1,290

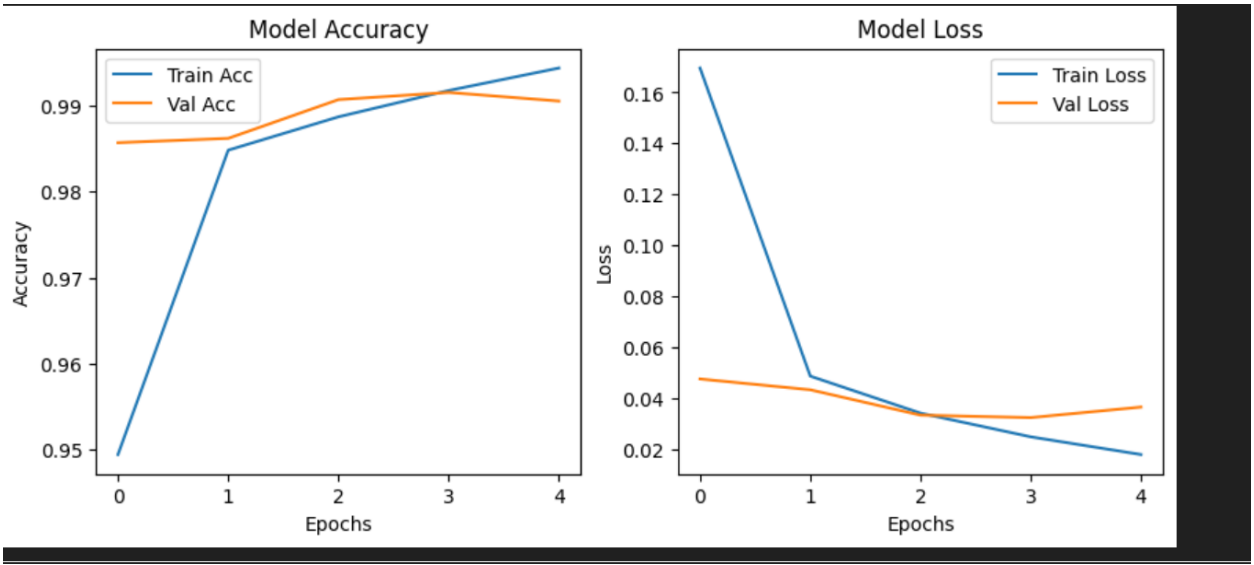
```
Total params: 225,034 (879.04 KB)
Trainable params: 225,034 (879.04 KB)
Non-trainable params: 0 (0.00 B)
```

```
history = model.fit(X_train, y_train, epochs=5, batch_size=64, validation_split=0.1)
```

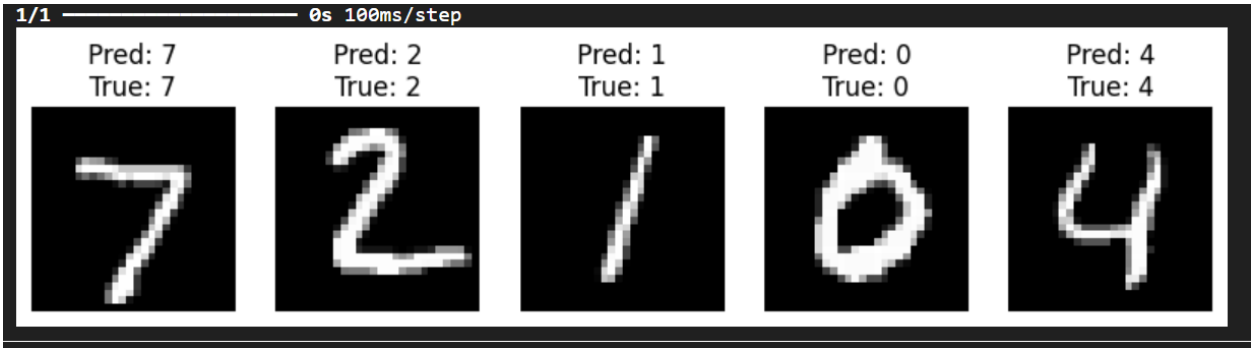
```
Epoch 1/5
844/844 ━━━━━━━━━━━━━━━━━━━━ 43s 49ms/step - accuracy: 0.8815 - loss: 0.3955 - val_accuracy: 0.9857 - val_loss: 0.0476
Epoch 2/5
844/844 ━━━━━━━━━━━━━━━━━━━━ 41s 48ms/step - accuracy: 0.9834 - loss: 0.0527 - val_accuracy: 0.9862 - val_loss: 0.0433
Epoch 3/5
844/844 ━━━━━━━━━━━━━━━━━━━━ 41s 49ms/step - accuracy: 0.9882 - loss: 0.0347 - val_accuracy: 0.9907 - val_loss: 0.0334
Epoch 4/5
844/844 ━━━━━━━━━━━━━━━━━━━━ 81s 48ms/step - accuracy: 0.9924 - loss: 0.0222 - val_accuracy: 0.9915 - val_loss: 0.0324
Epoch 5/5
844/844 ━━━━━━━━━━━━━━━━━━━━ 40s 48ms/step - accuracy: 0.9944 - loss: 0.0170 - val_accuracy: 0.9905 - val_loss: 0.0365
```

```
test_loss, test_acc = model.evaluate(X_test, y_test)
print(f"✅ Test Accuracy: {test_acc * 100:.2f}%")
```

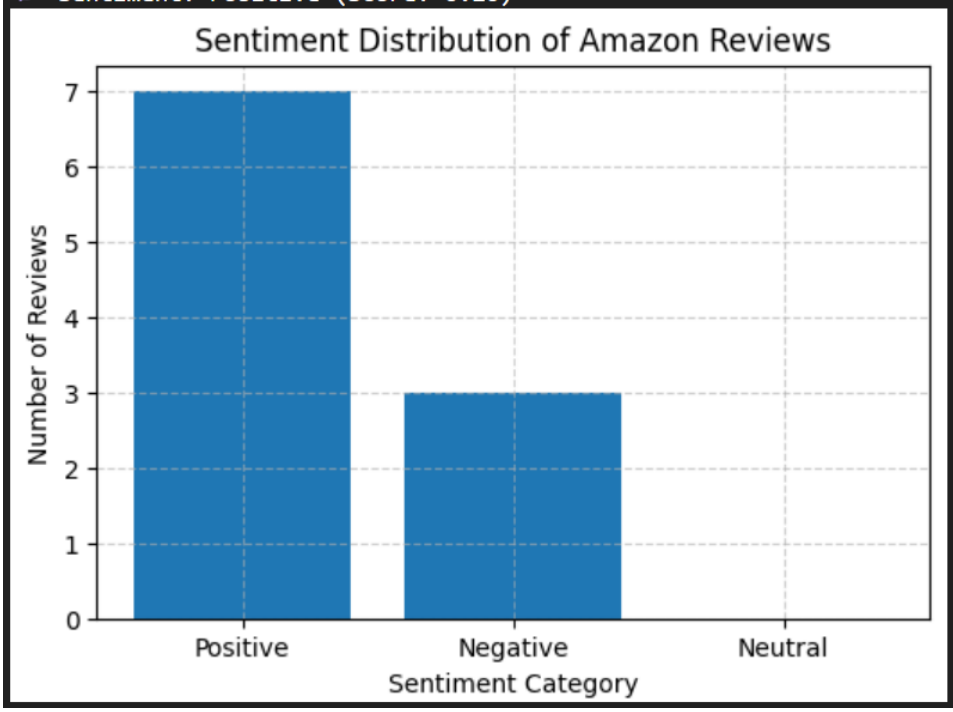
```
313/313 ━━━━━━━━━━━━━━━━━━━━ 2s 8ms/step - accuracy: 0.9884 - loss: 0.0361
✅ Test Accuracy: 99.08%
```




```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 100ms/step
```



Review: Amazon delivery was fast, but the Logitech mouse stopped working after a week.
Named Entities: [('Amazon', 'ORG'), ('Logitech', 'ORG'), ('a week', 'DATE')]
Sentiment: Positive (Score: 0.20)



Review: I absolutely love the new Samsung Galaxy S23! The camera quality is outstanding.
Named Entities: [('Samsung Galaxy S23', 'ORG')]
Sentiment: Positive 😊 (Score: 0.39)

Review: The battery of my Lenovo laptop dies too quickly, very disappointed.
Named Entities: [('Lenovo', 'ORG')]
Sentiment: Negative 😞 (Score: -0.98)

Review: Apple AirPods are great but a bit overpriced for the sound quality.
Named Entities: [('Apple AirPods', 'ORG')]
Sentiment: Positive 😊 (Score: 0.60)

Review: I bought a Sony headphone last week and it’s simply amazing!
Named Entities: [('Sony', 'ORG'), ('last week', 'DATE')]
Sentiment: Positive 😊 (Score: 0.38)

Review: This HP printer is the worst purchase I’ve ever made. Constant paper jams!
Named Entities: [('HP', 'ORG')]
Sentiment: Negative 😞 (Score: -0.50)