

AP23110010750

naïve Bayesian classifier

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

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

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)
```

```
model = GaussianNB()
```

```
model.fit(X_train, y_train)
```

```
* GaussianNB ⓘ ⓘ
GaussianNB()
```

```
y_pred = model.predict(X_test)
```

```
accuracy = accuracy_score(y_test, y_pred)
print(f"Model Accuracy: {accuracy*100:.2f}%\n")
```

```
Model Accuracy: 97.78%
```

```
print("== Prediction Results ==")
for i in range(len(y_test)):
    result = "Correct" if y_pred[i] == y_test[i] else "Wrong"
    print(f"Sample {i}: True = {iris.target_names[y_test[i]}}, Predicted = {iris.target_names[y_pred[i]}] → {result}")

== Prediction Results ==
Sample 0: True = versicolor, Predicted = versicolor → Correct
Sample 1: True = setosa, Predicted = setosa → Correct
Sample 2: True = virginica, Predicted = virginica → Correct
Sample 3: True = versicolor, Predicted = versicolor → Correct
Sample 4: True = versicolor, Predicted = versicolor → Correct
Sample 5: True = setosa, Predicted = setosa → Correct
Sample 6: True = versicolor, Predicted = versicolor → Correct
Sample 7: True = virginica, Predicted = virginica → Correct
Sample 8: True = versicolor, Predicted = versicolor → Correct
Sample 9: True = versicolor, Predicted = versicolor → Correct
Sample 10: True = virginica, Predicted = virginica → Correct
Sample 11: True = setosa, Predicted = setosa → Correct
Sample 12: True = setosa, Predicted = setosa → Correct
Sample 13: True = setosa, Predicted = setosa → Correct
Sample 14: True = setosa, Predicted = setosa → Correct
Sample 15: True = versicolor, Predicted = virginica → Wrong
Sample 16: True = virginica, Predicted = virginica → Correct
Sample 17: True = versicolor, Predicted = versicolor → Correct
Sample 18: True = versicolor, Predicted = versicolor → Correct
Sample 19: True = virginica, Predicted = virginica → Correct
Sample 20: True = setosa, Predicted = setosa → Correct
Sample 21: True = virginica, Predicted = virginica → Correct
Sample 22: True = setosa, Predicted = setosa → Correct
Sample 23: True = virginica, Predicted = virginica → Correct
Sample 24: True = virginica, Predicted = virginica → Correct
Sample 25: True = virginica, Predicted = virginica → Correct
Sample 26: True = virginica, Predicted = virginica → Correct
Sample 27: True = virginica, Predicted = virginica → Correct
Sample 28: True = setosa, Predicted = setosa → Correct
Sample 29: True = setosa, Predicted = setosa → Correct
```

```
Sample 30: True = setosa, Predicted = setosa → Correct
Sample 31: True = setosa, Predicted = setosa → Correct
Sample 32: True = versicolor, Predicted = versicolor → Correct
Sample 33: True = setosa, Predicted = setosa → Correct
Sample 34: True = setosa, Predicted = setosa → Correct
Sample 35: True = virginica, Predicted = virginica → Correct
Sample 36: True = versicolor, Predicted = versicolor → Correct
Sample 37: True = setosa, Predicted = setosa → Correct
Sample 38: True = setosa, Predicted = setosa → Correct
Sample 39: True = setosa, Predicted = setosa → Correct
Sample 40: True = virginica, Predicted = virginica → Correct
Sample 41: True = versicolor, Predicted = versicolor → Correct
Sample 42: True = versicolor, Predicted = versicolor → Correct
Sample 43: True = setosa, Predicted = setosa → Correct
Sample 44: True = setosa, Predicted = setosa → Correct
```

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

print("==== Confusion Matrix ===")
print(confusion_matrix(y_test, y_pred))
```

```
==== Classification Report ===
      precision    recall  f1-score   support
setosa       1.00     1.00     1.00      19
versicolor   1.00     0.92     0.96      13
virginica    0.93     1.00     0.96      13

accuracy         0.98     0.98     0.98      45
macro avg       0.98     0.97     0.97      45
weighted avg    0.98     0.98     0.98      45

==== Confusion Matrix ===
[[19  0  0]
 [ 0 12  1]
 [ 0  0 13]]
```

Start coding or [generate](#) with AI.