



DIGITAL SKILL FAIR 37.0
DATA SCIENCE

MACHINE LEARNING CLASSIFICATION

ANNISA NURLAILI AULIA SAFITRI

TABLE OF CONTENTS

1

DEFINITION

What is the definition of Machine Learning Classification?

2

TOOLS

What tools are needed?

3

STEPS

How to run the program?

4

RESULT

What are the results are obtained?

DEFINITION

Machine learning classification is a technique in machine learning that categorizes or classifies data into predefined labels based on patterns learned from previous data (training data). The classification model takes features from input data and uses an algorithm to determine the most appropriate category or label.

TOOLS



Python is an open-source programming language used for application development, data analysis, artificial intelligence, and more.

LIBRARY



Pandas is a Python library for data manipulation and analysis, providing powerful data structures like DataFrame and Series to handle structured data efficiently.



Matplotlib is a Python library for data visualization, allowing users to create various plots such as line charts, bar charts, scatter plots, and histograms.



Scikit-learn is a Python library for machine learning, providing tools for classification, regression, clustering, and data preprocessing.

DATASET

Iris Plants Dataset

A classic dataset in machine learning, often used for classification tasks

Source:

https://scikit-learn.org/1.5/datasets/toy_dataset.html

Data Set Characteristics:

Number of Instances: 150 (50 in each of three classes)

Number of Attributes: 4 numeric, predictive attributes and the class

Attribute Information:

- sepal length in cm
- sepal width in cm
- petal length in cm
- petal width in cm
- class:
 - Iris-Setosa
 - Iris-Versicolour
 - Iris-Virginica

DATASET

Summary Statistics:

sepal length:	4.3	7.9	5.84	0.83	0.7826
sepal width:	2.0	4.4	3.05	0.43	-0.4194
petal length:	1.0	6.9	3.76	1.76	0.9490 (high!)
petal width:	0.1	2.5	1.20	0.76	0.9565 (high!)

STEPS

1

**Collect and prepare
data**

2

Data preprocessing

3

**Exploratory Data
Analysis (EDA)**

4

Data Splitting

5

Model Training

6

Model Evaluation

DATA PREPROCESSING

```
[89] import pandas as pd
      from sklearn import datasets
```

```
iris = datasets.load_iris()
```

```
x = iris.data
y = iris.target
```

```
x = pd.DataFrame(x, columns=iris.feature_names)
df_y = pd.DataFrame(y, columns=['target'])
```

```
df = pd.concat([df_x, df_y], axis=1)
```

```
df
```

	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
...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

150 rows × 5 columns

This dataset has 150 rows and 4 columns, represented by the variable `x`. Meanwhile, the variable `y` represents the target with 150 rows and 1 column.

EDA

```
✓ 1s df['target'].unique()
array([0, 1, 2])
```

The data from y (target) has 3 arrays.

```
[7] df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
#   Column                Non-Null Count  Dtype
---  -
0   sepal length (cm)      150 non-null   float64
1   sepal width (cm)       150 non-null   float64
2   petal length (cm)      150 non-null   float64
3   petal width (cm)       150 non-null   float64
4   target                 150 non-null   int64
dtypes: float64(4), int64(1)
memory usage: 6.0 KB
```

The Iris dataset contains 150 samples with 4 numerical features (float64) and 1 categorical column (int64). There are no missing values (NaN).

EDA

✓ [77] df.describe()
0s

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333	1.000000
std	0.828066	0.435866	1.765298	0.762238	0.819232
min	4.300000	2.000000	1.000000	0.100000	0.000000
25%	5.100000	2.800000	1.600000	0.300000	0.000000
50%	5.800000	3.000000	4.350000	1.300000	1.000000
75%	6.400000	3.300000	5.100000	1.800000	2.000000
max	7.900000	4.400000	6.900000	2.500000	2.000000

Sepal length and width are more variable than petals.

Petal width and petal length have a wider distribution than sepal width.

DATA SPLITTING

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

```
[79] round(150*0.2)
```

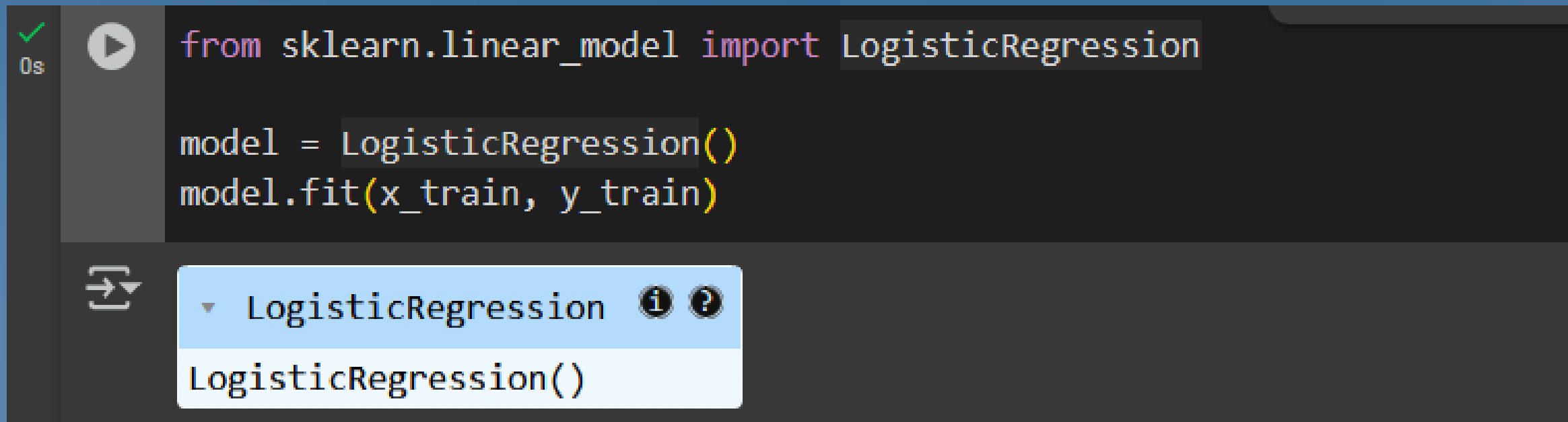
```
30
```

```
[94] round(150*0.8)
```

```
120
```

The model is trained on 120 samples and tested on 30 samples, so that the evaluation of model performance is more accurate.

MODEL TRAINING



```
✓ 0s ▶ from sklearn.linear_model import LogisticRegression

model = LogisticRegression()
model.fit(x_train, y_train)
```

↔

- LogisticRegression ⓘ ⓘ
 - LogisticRegression()

The algorithm used is Logistic Regression. Logistic Regression is a machine learning algorithm used for binary and multiclass classification. Although called regression, this algorithm is used to predict categories or classes, not continuous values.

MODEL EVALUATION

```

▶ from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

y_pred = model.predict(x_test)
accuracy = accuracy_score(y_test, y_pred)
print("Laporan Klasifikasi:")
print(f'Accuracy: {accuracy * 100:.1f}%')

```

```

⇒ Laporan Klasifikasi:
Accuracy: 100.0%

```

```

[83] import seaborn as sns
import matplotlib.pyplot as plt

cm = confusion_matrix(y_test, y_pred)

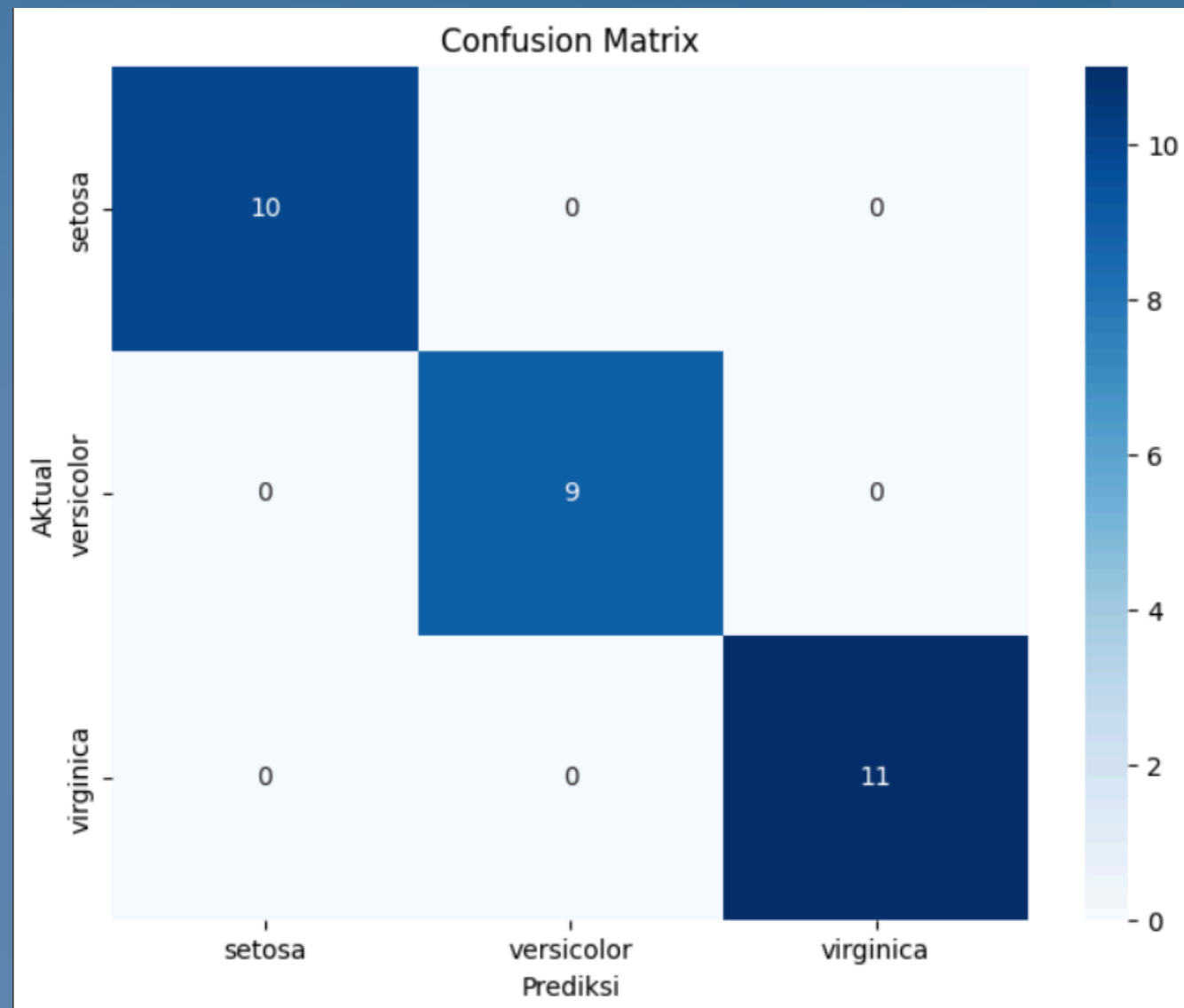
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=iris.target_names, yticklabels=iris.target_names)
plt.xlabel('Prediksi')
plt.ylabel('Aktual')
plt.title('Confusion Matrix')
plt.show()

```

The model correctly classifies all test samples.

Further discussion of accuracy values is explained in the Results section.

RESULT



The confusion matrix illustrates the model's classification performance on the test dataset. Each row represents the actual class, while each column represents the predicted class. The results show that all Setosa (10), Versicolor (9), and Virginica (11) samples were correctly classified, with no misclassifications.

The model perfectly classified all test samples, but 100% accuracy is suspicious. It may indicate an easy dataset, a small test set with no difficult cases, or data leakage between training and testing.

CONCLUSION

The Iris dataset was classified using Logistic Regression, achieving 100% accuracy with no misclassifications. While this suggests a well-structured dataset, it may also indicate a small test set or data leakage. Future steps include cross-validation, testing on new datasets, and verifying data integrity to ensure model reliability.



THANK YOU

Check for more on my GitHub:
[https://github.com/annisanass/
MachineLearningClassification.git](https://github.com/annisanass/MachineLearningClassification.git)

