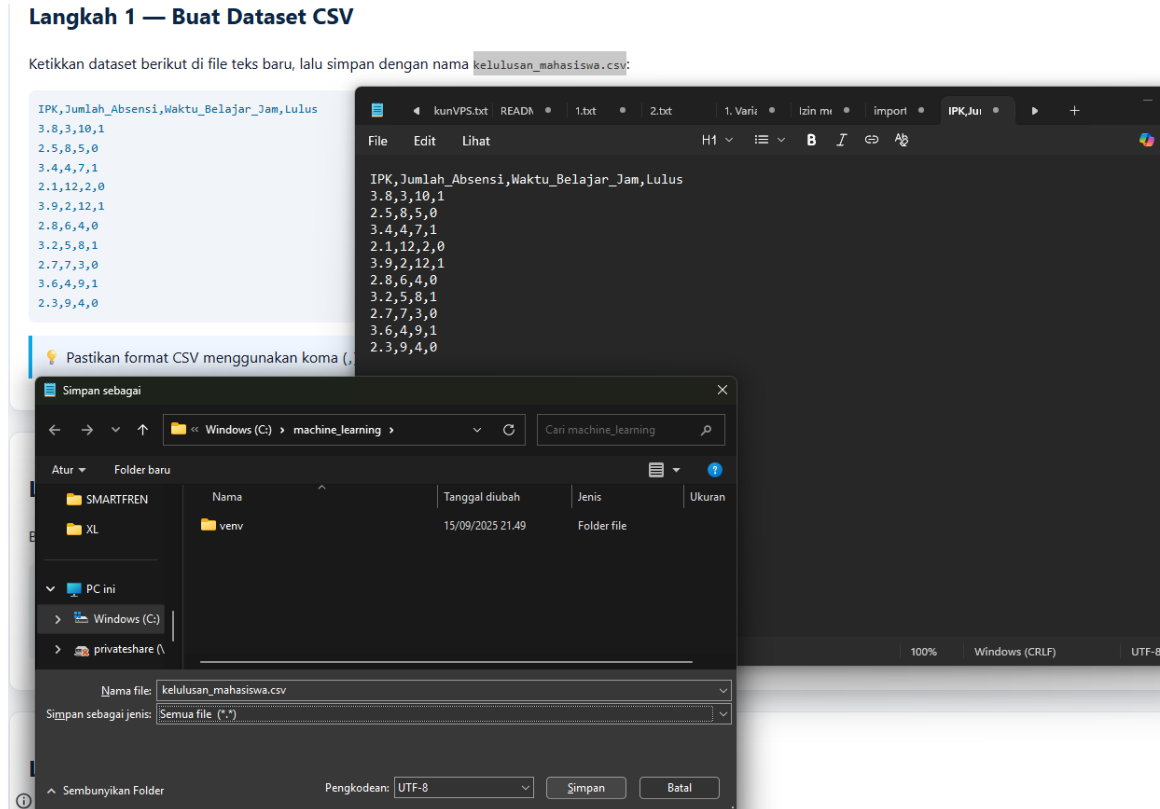


NAMA : SEFTIA DELLA FIISYATIR RODHIAH
NIM : 231011401012
KELAS : TI.05TPLE016

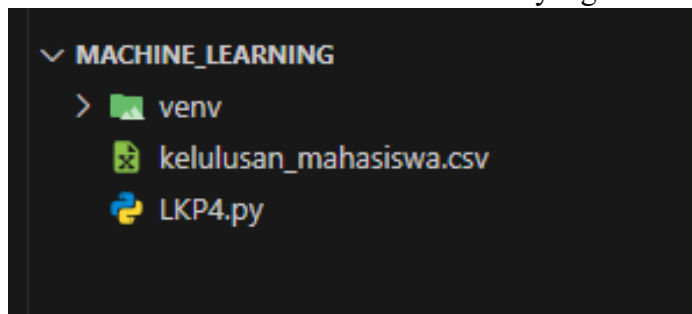
Lembar Kerja Pertemuan 4 – Machine Learning

1. Langkah 1 – Membuat Dataset CSV

Simpan terlebih dahulu dataset dalam format CSV ke folder yang di pilih.



Kemudian salin file tersebut ke direktori yang sudah memiliki environment Python.



2. Langkah 2 – Collection

Masukkan potongan kode berikut, lalu jalankan.

```
LKP4.py X
LKP4.py > ...
1 # Langkah 2 – Collection
2 import pandas as pd
3 df = pd.read_csv("kelulusan_mahasiswa.csv")
4 print(df.info())
5 print(df.head())
```

Hasil yang muncul akan terlihat seperti berikut:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
• (venv) PS C:\machine_learning> python LKP4.py
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   IPK                    10 non-null    float64
1   Jumlah_Absensi        10 non-null    int64
2   Waktu_Belajar_Jam     10 non-null    int64
3   Lulus                  10 non-null    int64
dtypes: float64(1), int64(3)
memory usage: 448.0 bytes
None
   IPK  Jumlah_Absensi  Waktu_Belajar_Jam  Lulus
0  3.8               3                10     1
1  2.5               8                 5     0
2  3.4               4                 7     1
3  2.1              12                 2     0
4  3.9               2                12     1
(venv) PS C:\machine_learning>
```

Penjelasan:

- Fungsi `pandas.read_csv()` digunakan untuk membaca file CSV bernama **kelulusan_mahasiswa.csv**.
- Perintah `df.info()` menampilkan informasi mengenai struktur DataFrame, seperti jumlah kolom, tipe data, serta apakah terdapat nilai kosong.
- Sedangkan `df.head()` digunakan untuk menampilkan **lima baris pertama** dari dataset.

3. Langkah 3 – Data Cleaning

Masukkan kode lanjutan berikut untuk melakukan proses pembersihan data, lalu jalankan program tersebut.

```
LKP4.py X
LKP4.py > ...
1 # Langkah 2 – Collection
2 import pandas as pd
3 df = pd.read_csv("kelulusan_mahasiswa.csv")
4 print(df.info())
5 print(df.head())
6
7 # Langkah 3 – Cleaning
8 print(df.isnull().sum())
9 df = df.drop_duplicates()
10
11 import seaborn as sns
12 sns.boxplot(x=df['IPK'])
```

Output yang dihasilkan akan tampak seperti ini:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
• (venv) PS C:\machine_learning> python LKP4.py
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype
---  -
0   IPK                    10 non-null    float64
1   Jumlah_Absensi        10 non-null    int64
2   Waktu_Belajar_Jam     10 non-null    int64
3   Lulus                  10 non-null    int64
dtypes: float64(1), int64(3)
memory usage: 448.0 bytes
None
   IPK  Jumlah_Absensi  Waktu_Belajar_Jam  Lulus
0  3.8                3                  10     1
1  2.5                8                   5     0
2  3.4                4                   7     1
3  2.1               12                   2     0
4  3.9                2                  12     1
IPK          0
Jumlah_Absensi  0
Waktu_Belajar_Jam  0
Lulus          0
dtype: int64
• (venv) PS C:\machine_learning>
```

Penjelasan:

- Mengecek apakah terdapat nilai kosong (NaN) pada setiap kolom.
- Menghapus **baris duplikat** jika ditemukan data yang sama.
- Membuat **boxplot** pada kolom **IPK** untuk mendeteksi adanya **outlier**.

4. Langkah 4 – Exploratory Data Analysis (EDA)

Selanjutnya, tambahkan potongan kode berikut untuk melakukan analisis eksploratif terhadap data

```
LKP4.py > ...
1 # Langkah 2 – Collection
2 import pandas as pd
3 df = pd.read_csv("kelulusan_mahasiswa.csv")
4 print(df.info())
5 print(df.head())
6
7 # Langkah 3 – Cleaning
8 print(df.isnull().sum())
9 df = df.drop_duplicates()
10
11 import seaborn as sns
12 sns.boxplot(x=df['IPK'])
13
14 # Langkah 4 – Exploratory Data Analysis (EDA)
15 print(df.describe())
16 sns.histplot(df['IPK'], bins=10, kde=True)
17 sns.scatterplot(x='IPK', y='Waktu_Belajar_Jam', data=df, hue='Lulus')
18 sns.heatmap(df.corr(), annot=True, cmap="coolwarm")
```

Hasil analisis akan muncul setelah dijalankan.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Data columns (total 4 columns):
# Column Non-Null Count Dtype
---
0 IPK 10 non-null float64
Data columns (total 4 columns): ...
0 IPK 10 non-null float64
1 Jumlah_Absensi 10 non-null int64
2 Waktu_Belajar_Jam 10 non-null int64
3 Lulus 10 non-null int64
dtypes: float64(1), int64(3)
memory usage: 448.0 bytes
None
IPK Jumlah_Absensi Waktu_Belajar_Jam Lulus
0 3.8 3 10 1
1 2.5 8 5 0
2 3.4 4 7 1
3 2.1 12 2 0
4 3.9 2 12 1
IPK 0
Jumlah_Absensi 0
Waktu_Belajar_Jam 0
Lulus 0
dtype: int64
IPK Jumlah_Absensi Waktu_Belajar_Jam Lulus
count 10.000000 10.000000 10.000000 10.000000
mean 3.030000 6.000000 6.400000 0.500000
std 0.639531 3.05505 3.306559 0.527046
min 2.100000 2.00000 2.000000 0.000000
25% 2.550000 4.00000 4.000000 0.000000
50% 3.000000 5.50000 6.000000 0.500000
75% 3.550000 7.75000 8.750000 1.000000
max 3.900000 12.00000 12.000000 1.000000
```

Penjelasan:

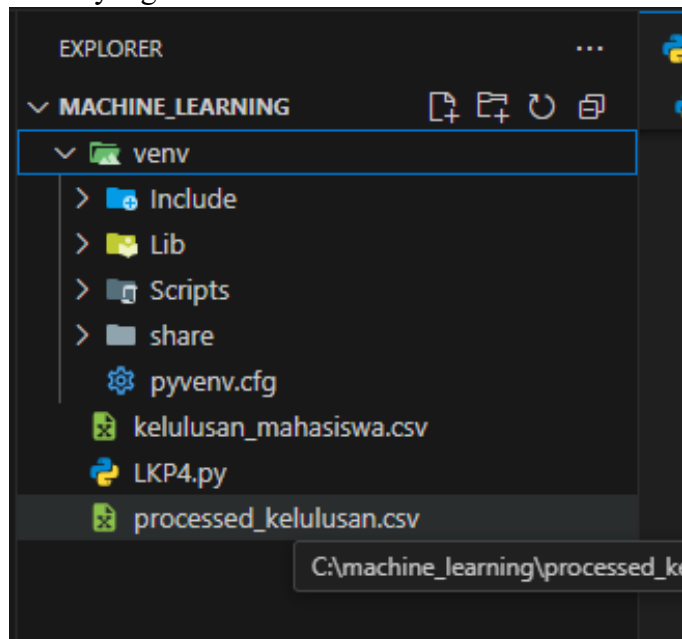
- `df.describe()` menampilkan **statistik deskriptif** seperti nilai rata-rata, standar deviasi, nilai minimum, maksimum, dan kuartil.
- `sns.histplot()` digunakan untuk melihat **distribusi nilai IPK**.
- `sns.scatterplot()` menampilkan **hubungan antara IPK dan waktu belajar**, dengan pewarnaan berdasarkan status kelulusan.
- `sns.heatmap()` menunjukkan **tingkat korelasi antar variabel** di dalam dataset

5. Langkah 5 – Feature Engineering

Masukkan kode berikut untuk membuat fitur baru pada dataset, lalu jalankan.

```
LKP4.py x
LKP4.py > ...
1 # Langkah 2 – Collection
2 import pandas as pd
3 df = pd.read_csv("kelulusan_mahasiswa.csv")
4 print(df.info())
5 print(df.head())
6
7 # Langkah 3 – Cleaning
8 print(df.isnull().sum())
9 df = df.drop_duplicates()
10
11 import seaborn as sns
12 sns.boxplot(x=df['IPK'])
13
14 # Langkah 4 – Exploratory Data Analysis (EDA)
15 print(df.describe())
16 sns.histplot(df['IPK'], bins=10, kde=True)
17 sns.scatterplot(x='IPK', y='Waktu_Belajar_Jam', data=df, hue='Lulus')
18 sns.heatmap(df.corr(), annot=True, cmap="coolwarm")
19
20 # Langkah 5 – Feature Engineering
21 df['Rasio_Absensi'] = df['Jumlah_Absensi'] / 14
22 df['IPK_x_Study'] = df['IPK'] * df['Waktu_Belajar_Jam']
23 df.to_csv("processed_kelulusan.csv", index=False)
```

setelah berhasil, akan terbentuk file baru bernama “**processed_kelulusan.csv**” di folder yang sama



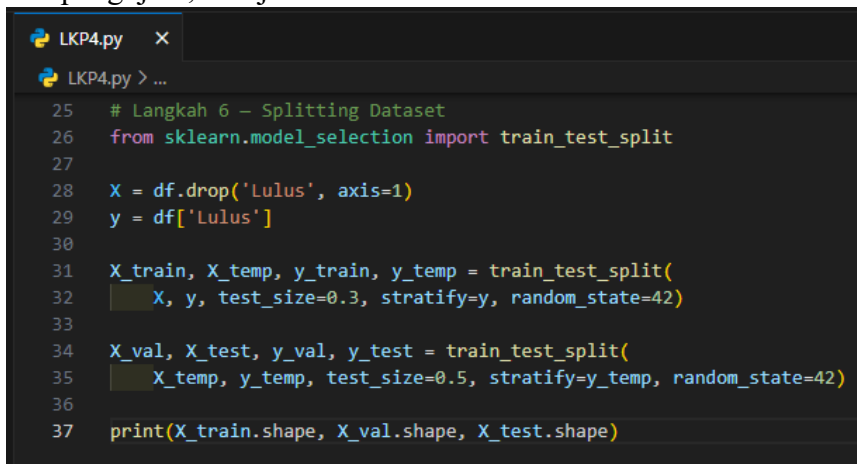
Penjelasan:

Dibuat dua fitur tambahan, yaitu:

- **Rasio_Absensi** → menunjukkan perbandingan antara jumlah kehadiran mahasiswa dengan total **14 pertemuan**.
- **IPK_x_Study** → hasil perkalian antara **IPK dan waktu belajar**, yang menggambarkan kombinasi antara prestasi dan usaha mahasiswa.

Dataset hasil modifikasi kemudian disimpan ke dalam file **processed_kelulusan.csv**, dengan pengaturan agar **index baris tidak ikut disimpan**.

6. Langkah 6 – Splitting Dataset
- tambahkan kode berikut untuk membagi dataset menjadi bagian pelatihan, validasi, dan pengujian, lalu jalankan.



```
25 # Langkah 6 – Splitting Dataset
26 from sklearn.model_selection import train_test_split
27
28 X = df.drop('Lulus', axis=1)
29 y = df['Lulus']
30
31 X_train, X_temp, y_train, y_temp = train_test_split(
32     X, y, test_size=0.3, stratify=y, random_state=42)
33
34 X_val, X_test, y_val, y_test = train_test_split(
35     X_temp, y_temp, test_size=0.5, stratify=y_temp, random_state=42)
36
37 print(X_train.shape, X_val.shape, X_test.shape)
```

Penjelasan:

Proses ini bertujuan untuk memisahkan:

- Fitur (X) dan target (y),
- Kemudian membagi data menjadi:
 - **70% untuk training**,
 - **15% untuk validation**,
 - **15% untuk testing**.

Parameter **stratify=y** berfungsi agar proporsi kelas *Lulus (1)* dan *Tidak Lulus (0)* tetap seimbang pada setiap subset data.

namun, ketika dijalankan, muncul pesan error.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Waktu_Belajar_Jam    0
Lulus                0
Waktu_Belajar_Jam    0
Lulus                0
Lulus                0 ...

Jumlah_Absensi        0
Waktu_Belajar_Jam    0
Lulus                0
dtype: int64

   IPK  Jumlah_Absensi  Waktu_Belajar_Jam  Lulus
count  10.000000      10.000000      10.000000  10.000000
mean    3.030000       6.000000       6.400000   0.500000
std     0.639531       3.055050       3.306559   0.527046
min     2.100000       2.000000       2.000000   0.000000
25%     2.550000       4.000000       4.000000   0.000000
50%     3.000000       5.500000       6.000000   0.500000
75%     3.550000       7.750000       8.750000   1.000000
max      3.900000      12.000000      12.000000   1.000000

Traceback (most recent call last):
  File "C:\machine_learning\LKP4.py", line 34, in <module>
    X_val, X_test, y_val, y_test = train_test_split(
  File "C:\machine_learning\venv\lib\site-packages\sklearn\utils\_param_validation.py", line 218, in
wrapper
    return func(*args, **kwargs)
  File "C:\machine_learning\venv\lib\site-packages\sklearn\model_selection\_split.py", line 2940, in
train_test_split
    train, test = next(cv.split(X=arrays[0], y=stratify))
  File "C:\machine_learning\venv\lib\site-packages\sklearn\model_selection\_split.py", line 1927, in
split
    for train, test in self._iter_indices(X, y, groups):
  File "C:\machine_learning\venv\lib\site-packages\sklearn\model_selection\_split.py", line 2342, in
_iter_indices
    raise ValueError(
ValueError: The least populated class in y has only 1 member, which is too few. The minimum number of
groups for any class cannot be less than 2.
(venv) PS C:\machine_learning>
```

Penyebab error:

Kelas dengan jumlah data paling sedikit di variabel y hanya memiliki satu anggota. Pada pembagian data dengan `stratify`, setiap kelas minimal harus memiliki dua data agar proses berjalan.

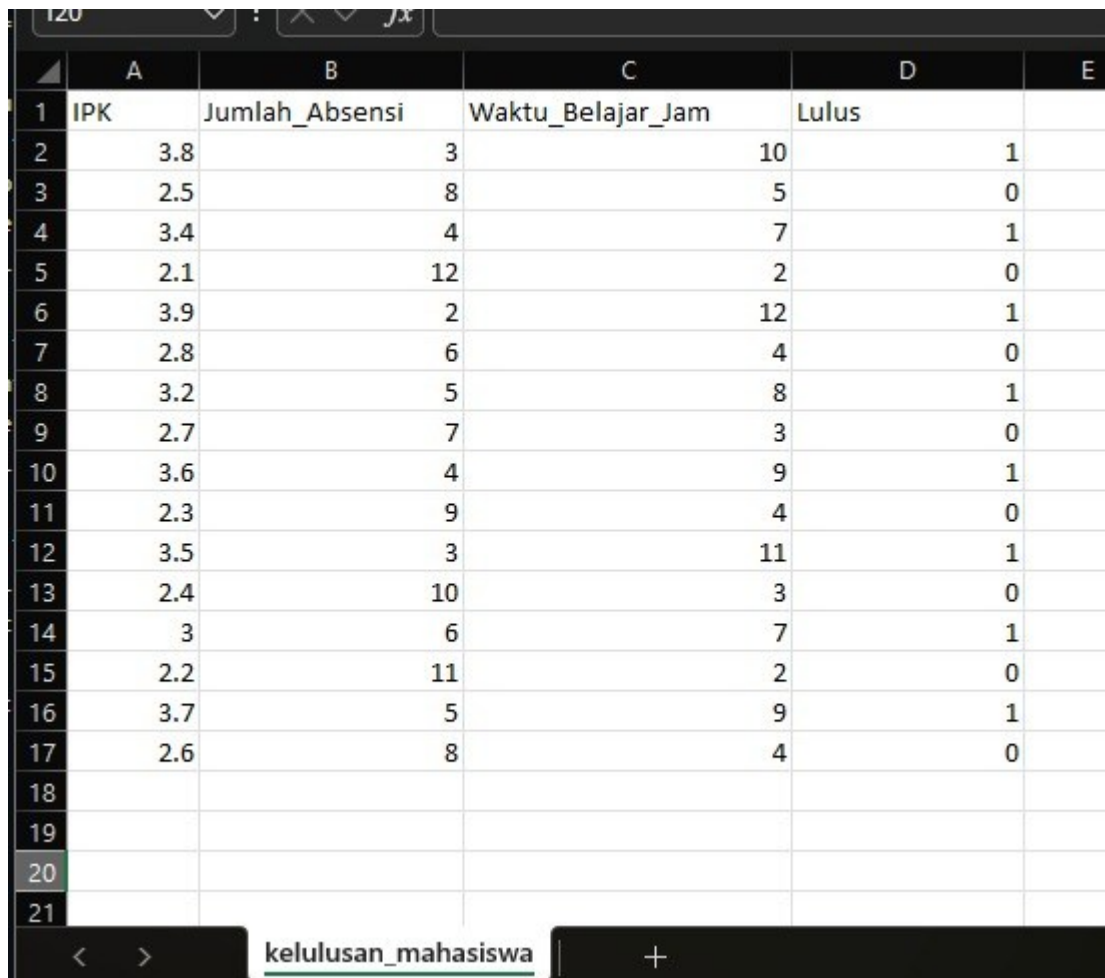
```
LKP4.py  X
LKP4.py > ...

25  # Langkah 6 - Splitting Dataset
26  from sklearn.model_selection import train_test_split
27
28  X = df.drop('Lulus', axis=1)
29  y = df['Lulus']
30
31  X_train, X_temp, y_train, y_temp = train_test_split(
32      X, y, test_size=0.3, stratify=y, random_state=42)
33
34  X_val, X_test, y_val, y_test = train_test_split(
35      X_temp, y_temp, test_size=0.5, stratify=y_temp, random_state=42)
36
37  print(X_train.shape, X_val.shape, X_test.shape)
```

Masalah ini muncul di bagian **train_test_split kedua**, karena jumlah data total hanya **10 baris**. Saat proses pembagian dua tahap dilakukan, salah satu kelas di subset data (**y_temp**) hanya memiliki **1 data saja**, sehingga scikit-learn tidak bisa melakukan stratifikasi.

Dengan kata lain, saat pembagian pertama dilakukan, salah satu label (misalnya *Lulus = 1* atau *Tidak Lulus = 0*) tersisa satu baris saja, sehingga pembagian kedua gagal karena **stratify membutuhkan minimal 2 data per kelas**.

Untuk mengatasi hal tersebut, dilakukan **penambahan data baru** di file **kelulusan_mahasiswa.csv**, dari **10 baris menjadi 16 baris**.



	A	B	C	D	E
1	IPK	Jumlah_Absensi	Waktu_Belajar_Jam	Lulus	
2	3.8	3	10	1	
3	2.5	8	5	0	
4	3.4	4	7	1	
5	2.1	12	2	0	
6	3.9	2	12	1	
7	2.8	6	4	0	
8	3.2	5	8	1	
9	2.7	7	3	0	
10	3.6	4	9	1	
11	2.3	9	4	0	
12	3.5	3	11	1	
13	2.4	10	3	0	
14	3	6	7	1	
15	2.2	11	2	0	
16	3.7	5	9	1	
17	2.6	8	4	0	
18					
19					
20					
21					

Setelah disimpan dan dijalankan kembali, program berhasil berjalan dengan **output** yang sesuai.

```
(venv) PS C:\machine_learning> python LKP4.py
RangeIndex: 16 entries, 0 to 15
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype
---  ---                ---
0   IPK                    16 non-null     float64
1   Jumlah_Absensi        16 non-null     int64
2   Waktu_Belajar_Jam     16 non-null     int64
3   Lulus                  16 non-null     int64
dtypes: float64(1), int64(3)
memory usage: 640.0 bytes
None
   IPK  Jumlah_Absensi  Waktu_Belajar_Jam  Lulus
0  3.8                3                10      1
1  2.5                8                 5      0
2  3.4                4                 7      1
3  2.1               12                 2      0
4  3.9                2                12      1
IPK                0
Jumlah_Absensi      0
Waktu_Belajar_Jam  0
Lulus               0
dtype: int64
   IPK  Jumlah_Absensi  Waktu_Belajar_Jam  Lulus
count  16.000000      16.000000      16.000000  16.000000
mean   2.981250       6.437500       6.250000   0.500000
std    0.610157       3.010399       3.296463   0.516398
min    2.100000       2.000000       2.000000   0.000000
25%    2.475000       4.000000       3.750000   0.000000
50%    2.900000       6.000000       6.000000   0.500000
75%    3.525000       8.250000       9.000000   1.000000
max     3.900000      12.000000      12.000000   1.000000
(11, 5) (2, 5) (3, 5)
(venv) PS C:\machine_learning>
```