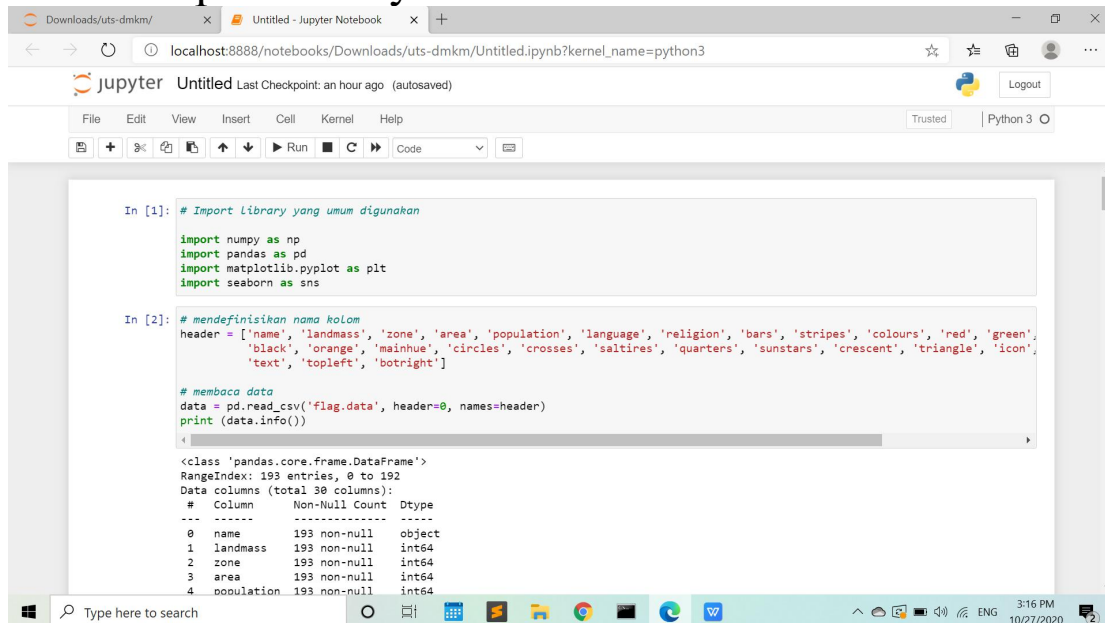


Nama : Zahlul Fuadi
NIM : 221810676
Kelas: 3SI1

UTS Data Mining

Proses import library dan membaca data



```
In [1]: # Import library yang umum digunakan

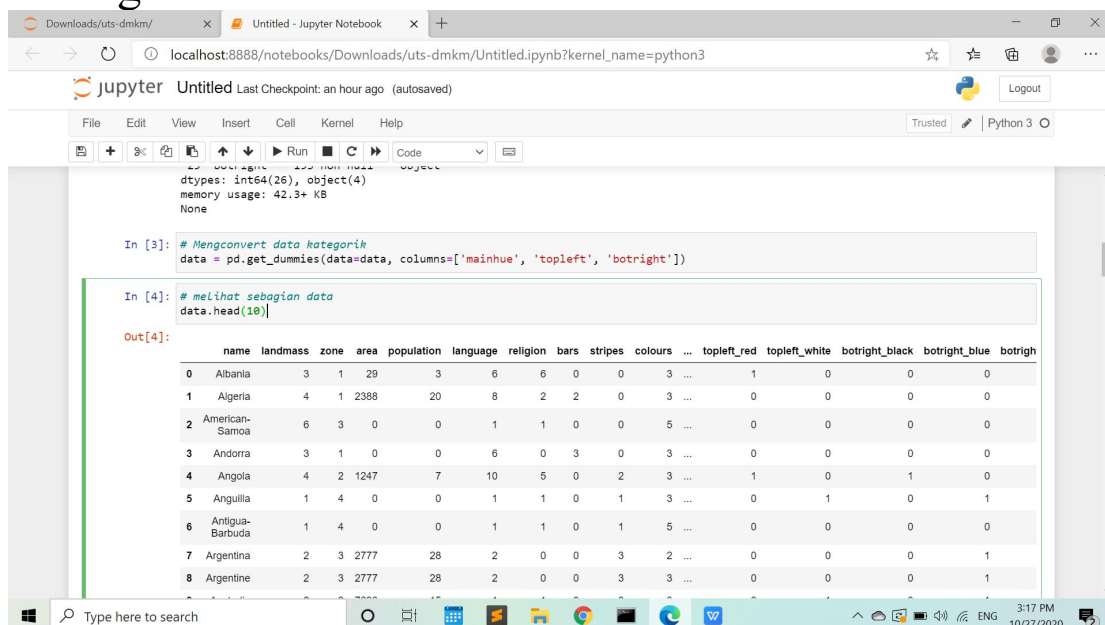
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

In [2]: # mendefinisikan nama kolom
header = ['name', 'landmass', 'zone', 'area', 'population', 'language', 'religion', 'bars', 'stripes', 'colours', 'red', 'green',
          'black', 'orange', 'mainhue', 'circles', 'crosses', 'saltires', 'quarters', 'sunstars', 'crescent', 'triangle', 'icon',
          'text', 'topleft', 'botright']

# membaca data
data = pd.read_csv('flag.data', header=0, names=header)
print(data.info())

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 193 entries, 0 to 192
Data columns (total 30 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   name        193 non-null      object
1   landmass    193 non-null      int64
2   zone        193 non-null      int64
3   area        193 non-null      int64
4   population  193 non-null      int64
```

Melakukan convert terhadap data kategorik dan melihat sebagian data

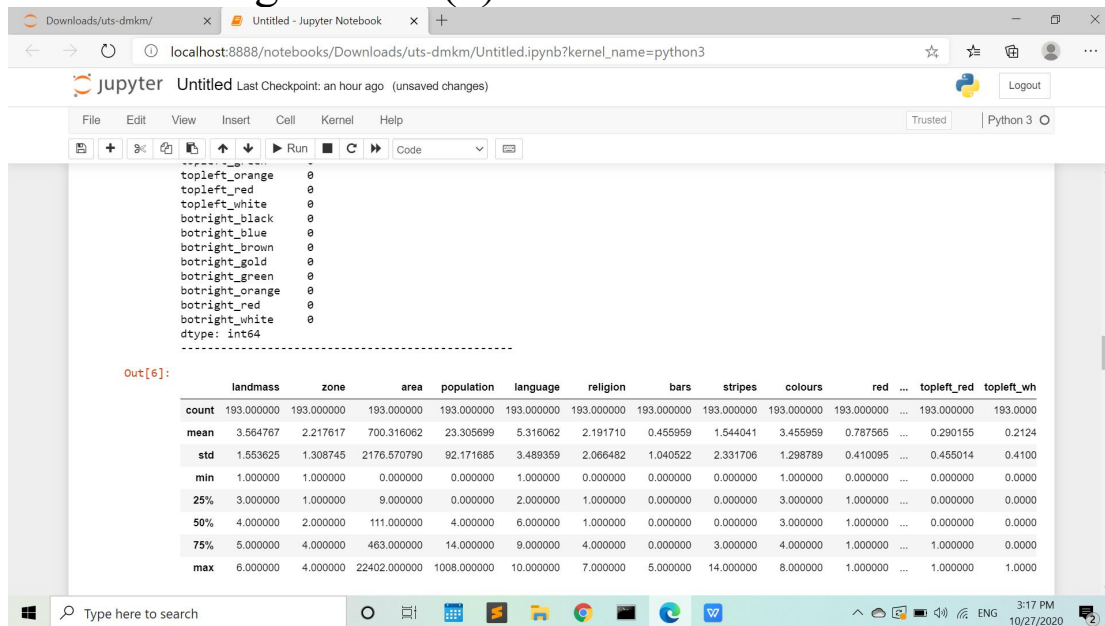


```
In [3]: # Mengconvert data kategorik
data = pd.get_dummies(data=data, columns=['mainhue', 'topleft', 'botright'])

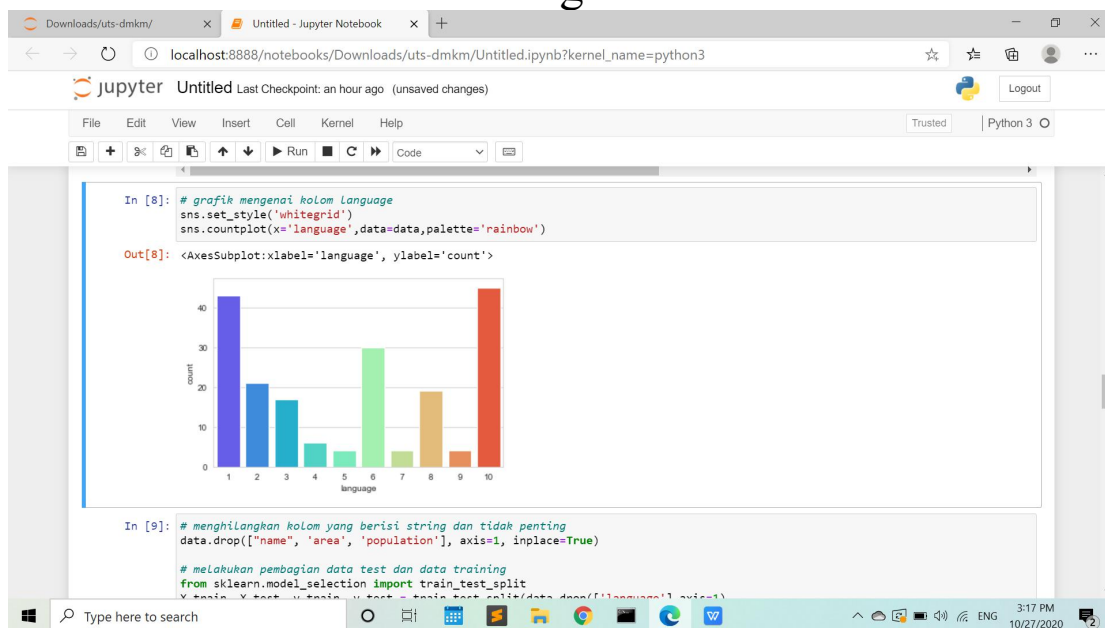
In [4]: # melihat sebagian data
data.head(10)
```

	name	landmass	zone	area	population	language	religion	bars	stripes	colours	...	topleft_red	topleft_white	botright_black	botright_blue	botright
0	Albania	3	1	29	3	6	6	0	0	3	...	1	0	0	0	0
1	Algeria	4	1	2388	20	8	2	2	0	3	...	0	0	0	0	0
2	American-Samoa	6	3	0	0	1	1	0	0	5	...	0	0	0	0	0
3	Andorra	3	1	0	0	6	0	3	0	3	...	0	0	0	0	0
4	Angola	4	2	1247	7	10	5	0	2	3	...	1	0	1	0	0
5	Anguilla	1	4	0	0	1	1	0	1	3	...	0	1	0	0	1
6	Antigua-Barbuda	1	4	0	0	1	1	0	1	5	...	0	0	0	0	0
7	Argentina	2	3	2777	28	2	0	0	3	2	...	0	0	0	0	1
8	Argentine	2	3	2777	28	2	0	0	3	3	...	0	0	0	0	1

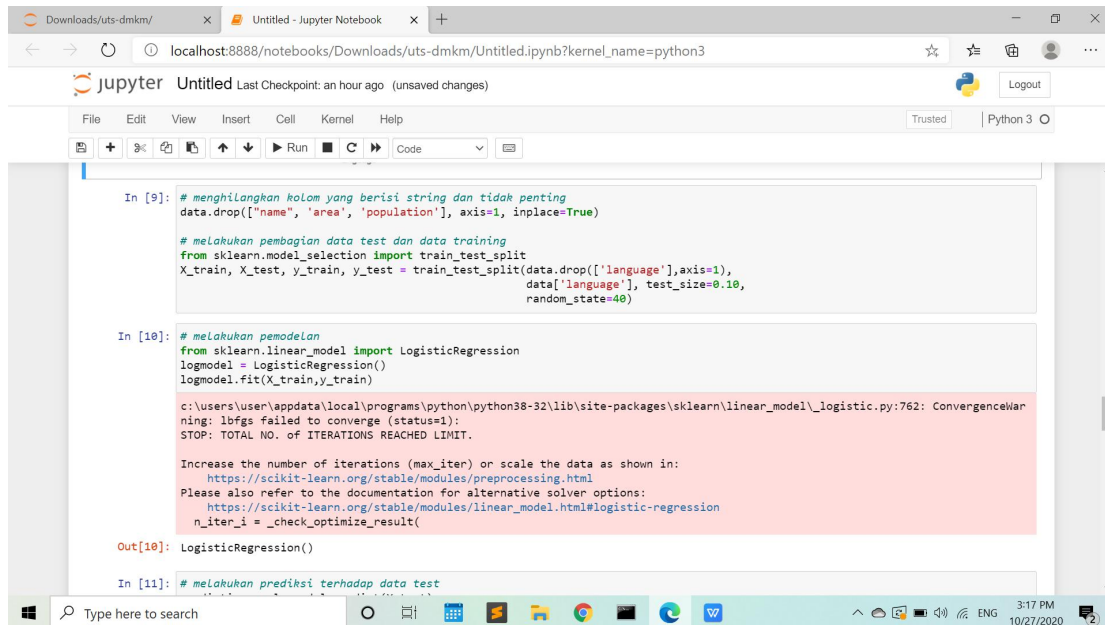
Melihat sebagian data (2)



Grafik data bahasa semua negara



Melakukan pembagian data training dan dataset dan pemodelan



```
In [9]: # menghilangkan kolom yang berisi string dan tidak penting
data.drop(["name", "area", "population"], axis=1, inplace=True)

# melakukan pembagian data test dan data training
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(data.drop(["language"], axis=1),
                                                    data["language"], test_size=0.10,
                                                    random_state=40)

In [10]: # melakukan pemodelan
from sklearn.linear_model import LogisticRegression
logmodel = LogisticRegression()
logmodel.fit(X_train, y_train)

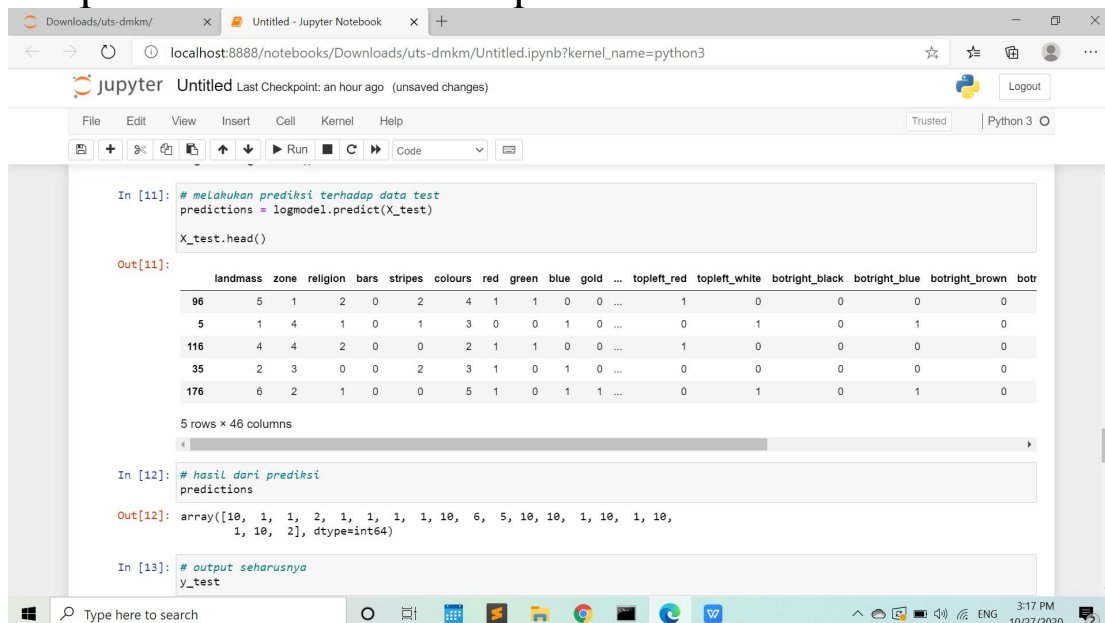
c:\users\user\appdata\local\programs\python\python38-32\lib\site-packages\sklearn\linear_model\_logistic.py:762: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
n_iter_i = _check_optimize_result(

Out[10]: LogisticRegression()

In [11]: # melakukan prediksi terhadap data test
```

Output dari model terhadap data test



```
In [11]: # melakukan prediksi terhadap data test
predictions = logmodel.predict(X_test)
X_test.head()

Out[11]:
```

	landmass	zone	religion	bars	stripes	colours	red	green	blue	gold	...	toleft_red	toleft_white	botright_black	botright_blue	botright_brown	botr
96	5	1	2	0	2	4	1	1	0	0	...	1	0	0	0	0	
5	1	4	1	0	1	3	0	0	1	0	...	0	1	0	0	1	0
116	4	4	2	0	0	2	1	1	0	0	...	1	0	0	0	0	0
35	2	3	0	0	2	3	1	0	1	0	...	0	0	0	0	0	0
176	6	2	1	0	0	5	1	0	1	1	...	0	1	0	1	0	

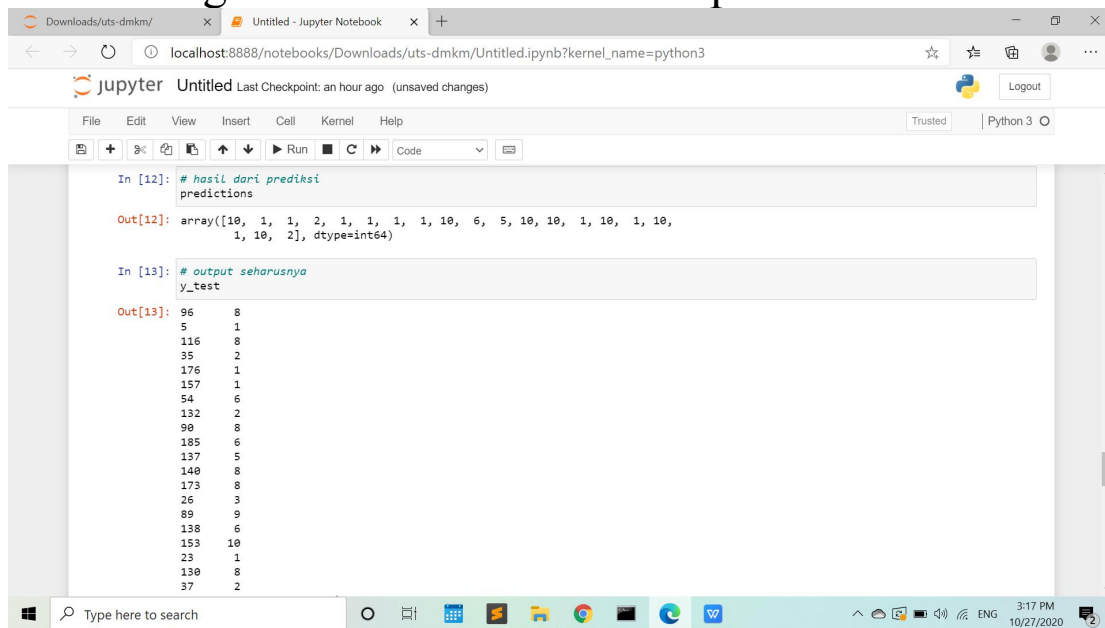
5 rows x 46 columns

```
In [12]: # hasil dari prediksi
predictions

Out[12]: array([10,  1,  1,  2,  1,  1,  1,  1,  1,  6,  5, 10, 10,  1, 10,  1, 10,
        1, 10,  2], dtype=int64)

In [13]: # output seharusnya
y_test
```

Perbandingan secara manual terhadap data asli



The screenshot shows a Jupyter Notebook interface with the following code and output:

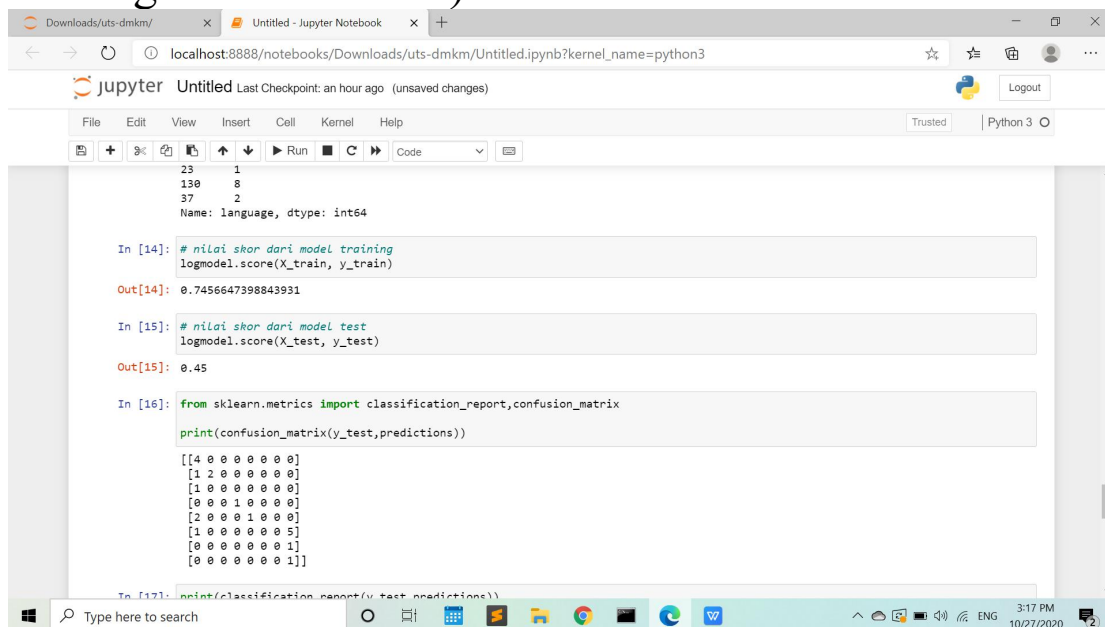
```
In [12]: # hasil dari prediksi
predictions

Out[12]: array([[10, 1, 1, 2, 1, 1, 1, 1, 10, 6, 5, 10, 10, 1, 10, 1, 10,
                1, 10, 2], dtype=int64)
```

```
In [13]: # output seharusnya
y_test

Out[13]: 96      8
          5      1
          116    8
          35     2
          176    1
          157    1
          54     6
          132    2
          90     8
          185    6
          137    5
          140    8
          173    8
          26     3
          89     9
          138    6
          153   10
          23     1
          130    8
          37     2
```

Skor dari model yang telah dibuat (skor terhadap model training dan model test) + confusion matrix



The screenshot shows a Jupyter Notebook interface with the following code and output:

```
23      1
130     8
37      2
Name: language, dtype: int64

In [14]: # nilai skor dari model training
logmodel.score(X_train, y_train)

Out[14]: 0.7456647398843931

In [15]: # nilai skor dari model test
logmodel.score(X_test, y_test)

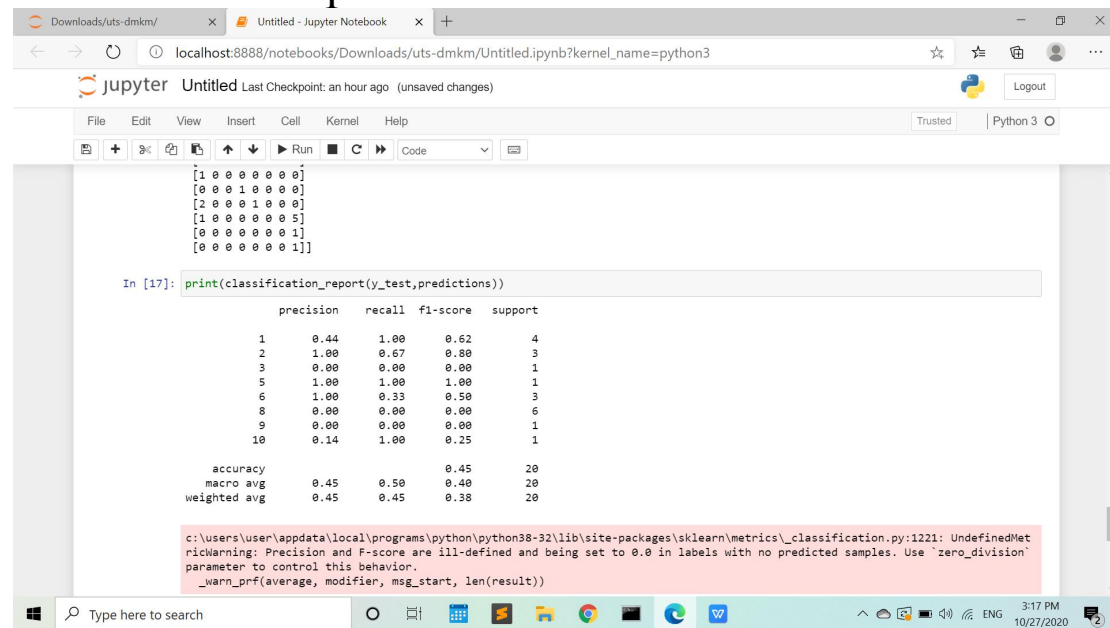
Out[15]: 0.45

In [16]: from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, predictions))

[[4 0 0 0 0 0 0]
 [1 2 0 0 0 0]
 [1 0 0 0 0 0]
 [0 0 1 0 0 0]
 [2 0 0 1 0 0]
 [1 0 0 0 0 5]
 [0 0 0 0 0 1]
 [0 0 0 0 0 1]]

In [17]: print(classification_report(y_test, predictions))
```

Classification report



The screenshot shows a Jupyter Notebook interface with a code cell containing a classification report. The report is a table with columns for precision, recall, f1-score, and support. The rows represent different classes (1, 2, 3, 5, 6, 8, 9, 10) and summary metrics (accuracy, macro avg, weighted avg). A warning message is displayed below the report, indicating that precision and f1-score are ill-defined for labels with no predicted samples.

```
In [17]: print(classification_report(y_test, predictions))
```

	precision	recall	f1-score	support
1	0.44	1.00	0.62	4
2	1.00	0.67	0.80	3
3	0.00	0.00	0.00	1
5	1.00	1.00	1.00	1
6	1.00	0.33	0.50	3
8	0.00	0.00	0.00	6
9	0.00	0.00	0.00	1
10	0.14	1.00	0.25	1
accuracy			0.45	20
macro avg	0.45	0.50	0.40	20
weighted avg	0.45	0.45	0.38	20

c:\users\user\appdata\local\programs\python\python38-32\lib\site-packages\sklearn\metrics_classification.py:1221: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use 'zero_division' parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))

- Accuracy artinya rasio prediksi Benar (positif dan negatif) dengan keseluruhan data yaitu sebesar 0,45
- Presicion artinya rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif pada setiap pilihan output
- Recall artinya rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif pada setiap pilihan output
- Specificity artinya kebenaran memprediksi negatif dibandingkan dengan keseluruhan data negatif
- F1 Score yaitu perbandingan rata-rata presicion dan recall yang dibobotkan