

## Hands-On

Hands-On digunakan pada kegiatan Microcredential Associate Data Scientist 2021

### Tugas Mandiri Pertemuan 16

Pertemuan 16 (enam belas) pada Microcredential Associate Data Scientist 2021 menyampaikan materi mengenai Membangun model: Evaluasi. silakan Anda kerjakan Latihan 1 s/d 5. Output yang anda lihat merupakan panduan yang dapat Anda ikuti dalam penulisan code :)

#### Soal 1: Pemahaman Tentang Model Evaluasi

Jawab pertanyaan di bawah ini dengan bahasa masing-masing?

1. Apa perbedaan antara data latih, data validasi, dan data test?
2. Bagaimana cara kita memilih performa suatu model?
3. Apa itu Confusion Matrix? Jelaskan secara lengkap!
4. Apa itu Classification Report dan sklearn?

Jawab:

1. Perbedaan antara data latih, data validasi, dan data test:
  - Data Latih atau yang biasa disebut juga data training yaitu sebuah data yang akan digunakan untuk melakukan training pada model yang akan dibuat.
  - Data Validasi yaitu sebuah data yang akan digunakan untuk melakukan proses validasi model dimana juga digunakan untuk mengecek agar tidak terjadi overfitting namanya. proses ini dilakukan setelah selesai melakukan training dan data test yaitu sebuah data yang digunakan untuk melakukan testing atau uji pada model. Jadi ini akan menjadi simulasu untuk pengujian dari model yang telah dibuat sebelumnya
2. Beberapa cara dapat dilakukan melalui evaluasi metrik. Metrik yang digunakan berbeda sesuai model yang dibuat. Metrik dapat berakurasi (paling omum digunakan), ataupun MAE atau MSE. Dari metrik-metrik tersebut, dipilih metrik yang sesuai dengan kasus yang dikerjakan (klasifikasi menggunakan akurasi, MSE untuk regresi, dan beberapa contoh lainnya). Biaya akurasi tinggi stupuan yang rendah, maka model dapat dikatakan memiliki performa yang baik. Sebaliknya, model memiliki performa yang kurang baik.
3. Confusion Matrix yaitu salah satu cara pengukuran performa pada masalah klasifikasi machine learning dimana output bisa berupa dua kelas atau lebih dengan label yang terdiri 4 kombinasi berbeda dari nilai prediksi dan nilai aktual. Confusion matrix bermanfaat untuk melihat seberapa besar percobaan atau untuk mengkuantifikasi biaya karena terjadinya kesalahan. selain itu bisa juga untuk memahami perbedaan antar kelas (klasifikasi). Untuk karakteristiknya diantaranya yaitu :
  - Ada sumbu data aktual dan sumbu data prediksi
  - Setiap kelas terdapat salah satu nama lainnya
  - Percobaan valid berada pada diagonal utama
  - Matrice berbentuk bujur sangkar
4. Classification report yaitu salah satu library yang ada pada sklearn yang digunakan untuk evaluasi matrix performance pada machine learning. dengan classification report maka kita akan mengetahui hasil dari precision, recall, F1 Score, dan support dari trained classification model kita

#### Soal 2: Aplikasi Model Evaluasi

Kali ini kita akan menggunakan data untuk memprediksi kelangsungan hidup pasien yang telah mengalami operasi payudara. Dengan informasi yang dimiliki terhadap pasien, kita akan membuat model untuk memprediksi apakah pasien akan bertahan hidup dalam waktu lebih dari 5 tahun atau tidak.

Lebih Lengkapnya kalau kita membaca informasi tentang dataset di link berikut:

<https://www.rithhubercontent.com/brownlee/Datasets/master/haberman.names>

Buat model Klasifikasi (Model Algoritma Bebas) untuk memprediksi status pasien dengan ketentuan sebagai berikut:

1. Bagi kedua data ini menjadi data training dan data test dengan `test_size=0.25`.
2. Pelajari tentang metrics `roc_auc` pada sklearn module dan evaluasi dengan menggunakan teknik cross-validation dengan scoring `'roc_auc'`. Baca [https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.cross\\_val\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html) untuk menggunakan metric `roc_auc` saat cross-validation.
3. Berapa score rata-rata dengan teknik cross-validation tersebut?
4. Prediksi hasil test dengan teknik cross-validation yang telah kalian buat!
5. Hitung jumlah korusus matrics dan hasil prediksi tersebut?
6. Bagaimana classification report dari hasil prediksi tersebut?
7. Seberapa baik model anda dalam memprediksi seorang pasien mempunyai status positive?
8. Seberapa baik model anda dalam memprediksi seorang pasien mempunyai status negatif?

#### Load Dataset

```
In [2]: # import library pandas
import pandas as pd

# Load dataset
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/haberman.csv"
list_cols = ['Age', 'Patient's Years', 'N_positive_ax', 'survival_status']
df = pd.read_csv(url, names=list_cols)
```

```
In [3]: # tampilkan 5 baris awal dataset dengan function head()
df.head()
```

```
Out[3]: 
   Age Patient's Years N_positive_ax survival_status
0   30          64           1            1
1   30          62           3            1
2   30          65           0            1
3   31          59           2            1
4   31          65           4            1
```

```
In [4]: # hitung jumlah masing" data pada kolom survival_status
df['survival_status'].value_counts()
```

```
Out[4]: 
Name: survival_status, dtype: int64
```

#### Build Model

```
In [5]: #import library train test split dan cross val
from sklearn.model_selection import train_test_split, cross_val_score

#Import library Logistic regression
from sklearn.linear_model import LogisticRegression

#Import library roc auc score
from sklearn.metrics import roc_auc_score
```

```
#Import library scale
from sklearn.preprocessing import scale
```

```
#Import library numpy
import numpy as np
```

```
In [6]: ## pemisahan feature dan target (data target : 'survival_status')
X = df.drop(['survival_status'], axis = 1)
X = scale(X)
Y = df['survival_status']
```

#### NO 1

```
In [7]: ## pemisahan variabel test dan train dari data xs dan y
# test size: 25%, random state = 42, dan stratify = y
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42, s
tratify=y)
```

```
In [8]: ## pembuatan objek model
model_logReg = LogisticRegression(random_state = 42)
```

## latih model

model\_logReg.fit(X\_train, y\_train)

## prediksi.

y\_predict = model\_logReg.predict(X\_test)

#### NO 2

```
In [9]: ## menghitung cross_val_score dengan scoring = 'roc_auc'
## parameter cv = 10
score = cross_val_score(model_logReg, X, y, scoring = 'roc_auc', cv = 10)
print(score)
```

[0.44021739 0.48978261 0.6731304 0.69021739 0.70380435 0.79292929

0.875 0.62784091 0.67613636 0.61363636]

#### NO 3

```
In [10]: # cetak rata-rata nilai rata-rata auc score
score.mean()
```

```
Out[10]: 0.693477711901624
```

#### NO 4

```
In [11]: # Prediksi data test dengan model yang telah kalian buat
auc_score = roc_auc_score(y_test, y_predict)
auc_score
```

```
Out[11]: 0.5311403589771929
```

#### NO 5

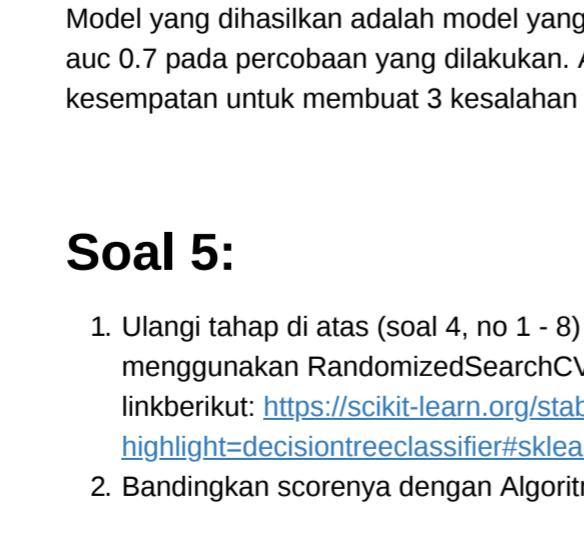
```
In [12]: # import library confusion matrix dan classification report
from sklearn.metrics import confusion_matrix, classification_report
```

```
In [13]: # apply confusion matrix dan cetak nilai confusion matrix
cm = confusion_matrix(y_test, y_predict, labels = [1,2])
```

```
Out[13]: array([[52,  5],
               [17,  3]], dtype=int64)
```

```
In [14]: # visualisasikan nilai confusion matrix ke dalam diagram heatmap
import seaborn as sns
sns.heatmap(cm, annot=True)
```

<AxesSubplot: >



#### NO 6

```
In [15]: # cetak nilai rata-rata probabilitas data test
y_predict.mean()
```

```
Out[15]: 0.5
```

#### NO 7

```
In [16]: # import library confusion matrix dan classification report
from sklearn.metrics import confusion_matrix, classification_report
```

```
In [17]: # cetak nilai rata-rata nilai confusion matrix
cm = confusion_matrix(y_test, y_predict, labels = [1,2])
```

```
Out[17]: array([[52,  5],
               [17,  3]], dtype=int64)
```

```
In [18]: # visualisasikan nilai confusion matrix ke dalam diagram heatmap
import seaborn as sns
sns.heatmap(cm, annot=True)
```

<AxesSubplot: >



#### NO 8

```
In [19]: # 8. score validasi terbaik
gscv.best_score_
```

```
Out[19]: 0.7289653361344539
```

#### NO 9

```
In [20]: # 9. prediksi nilai probabilitas masing-masing data test
y_predict = gscv.predict_proba(X_test)
```

```
Out[20]: array([[0.49, 0.51],
               [0.51, 0.49], ...]]
```

#### NO 10

```
In [21]: # 10. nilai score roc_auc pada data test
y_predict.mean()
```

```
Out[21]: 0.5
```

#### NO 11

```
In [22]: # cetak nilai rata-rata probabilitas data test
y_predict.mean()
```

```
Out[22]: 0.5
```

#### NO 12

```
In [23]: # 1. import library DecisionTreeClassifier
from sklearn.tree import DecisionTreeClassifier
```

```
In [24]: # 2. import library GridSearchCV
from sklearn.model_selection import GridSearchCV
```

```
In [25]: # 3. tuning hyperparameter dengan GridSearchCV (parameter cv=10)
# build model KNN
model_knn = KNeighborsClassifier()
param_knn = {'n_neighbors': np.arange(3,51), 'weights' : ['uniform', 'distance']}
gscv = GridSearchCV(model_knn, param_knn, scoring='roc_auc', cv = 10)
```

```
In [26]: # fit model
gscv.fit(X_train, y_train)
```

## prediksi.

y\_predict = gscv.predict(X\_test)

#### NO 13

```
In [27]: # 4. cetak nilai rata-rata nilai roc_auc pada data test
y_predict.mean()
```

```
Out[27]: 0.5311403589771929
```

#### NO 14

```
In [28]: # 5. cetak nilai rata-rata nilai roc_auc pada data test
y_predict.mean()
```

```
Out[28]: 0.5311403589771929
```

#### NO 15

```
In [29]: # 6. cetak nilai rata-rata nilai roc_auc pada data test
y_predict.mean()
```

```
Out[29]: 0.5311403589771929
```

#### NO 16

```
In [30]: # 7. cetak nilai rata-rata nilai roc_auc pada data test
y_predict.mean()
```

```
Out[30]: 0.5311403589771929
```

#### NO 17

```
In [31]: # 8. cetak nilai rata-rata nilai roc_auc pada data test
y_predict.mean()
```

```
Out[31]: 0.5311403589771929
```

#### NO 18

```
In [32]: # 9. cetak nilai rata-rata nilai roc_auc pada data test
y_predict.mean()
```

```
Out[32]: 0.5311403589771929
```

#### NO 19

```
In [33]: # 10. cetak nilai rata-rata nilai roc_auc pada data test
y_predict.mean()
```

```
Out[33]: 0.5311403589771929
```

#### NO 20

```
In [34]: # 11. cetak nilai rata-rata nilai roc_auc pada data test
y_predict.mean()
```

```
Out[34]: 0.5311403589771929
```

#### NO 21

```
In [35]: # 12. cetak nilai rata-rata nilai roc_auc pada data test
y_predict.mean()
```

```
Out[35]: 0.5311403589771929
```

#### NO 22

```
In [36]: # 13. cetak nilai rata-rata nilai roc_auc pada data test
y_predict.mean()
```

```
Out[36]: 0.5311403589771929
```

#### NO 23

```
In [37]: # 14. cetak nilai rata-rata nilai roc_auc pada data test
y_predict.mean()
```

```
Out[37]: 0.5311403589771929
```

#### NO 24

```
In [38]: # 15. cetak nilai rata-rata nilai roc_auc pada data test
y_predict.mean()
```

```
Out[38]: 0.5311403589771929
```

#### NO 25

```
In [39]: # 16. cetak nilai rata-rata nilai roc_auc pada data test
y_predict.mean()
```

```
Out[39]: 0.5311403589771929
```

#### NO 26

```
In [40]: # 17. cetak nilai rata-rata nilai roc_auc pada data test
y_predict.mean()
```

```
Out[40]: 0.5311403589771929
```

#### NO 27

```
In [41]: # 18. cetak nilai rata-rata nilai roc_auc pada data test
y_predict.mean()
```

```
Out[41]: 0.5311403589771929
```

## &lt;