



Hands-On

Hands-On ini digunakan pada kegiatan Microcredential Associate Data Scientist 2021

Tugas Mandiri Pertemuan 16

Pertemuan 16 (enambelas) 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 menilai performa suatu model?
3. Apa itu Confusion Matrix? Jelaskan secara lengkap!
4. Apa itu Classification Report dari sklearn?

Jawab:

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 terkait pasien, kita akan membuat model untuk memprediksi apakah pasien akan bertahan hidup dalam waktu lebih dari 5 tahun atau tidak.

Lebih Lengkapnya kalian bisa membaca informasi tentang dataset di link berikut:

<https://raw.githubusercontent.com/jbrownlee/Datasets/master/haberman.names>
[\(https://raw.githubusercontent.com/jbrownlee/Datasets/master/haberman.names\)](https://raw.githubusercontent.com/jbrownlee/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. Pelajar tentang metrics `roc_auc_score` kemudian buatlah model 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 rata2 dari model dengan teknik cross-validation tersebut?
4. Prediksi data test dengan model yang telah kalian buat!
5. Bagaimana hasil confusion matrix dari 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 [1]: # 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 [2]: # tampilkan 5 baris awal dataset dengan function head()
df.head()
```

Out[2]:

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

Out[3]:

1	225
2	81

Name: survival_status, dtype: int64

Build Model

```
In [4]: #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 [5]: ## pemisahan feature dan target (data target : 'survival_status')
X = df.drop('survival_status', axis = 1)
Xs = scale(X)
y = df['survival_status']
```

NO 1

```
In [6]: ## 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=0.33, random_state=42)
```

```
In [22]: ## 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 [10]: ## 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 0.80978261 0.67391304 0.69021739 0.70380435 0.79292929
 0.875      0.62784091 0.67613636 0.61363636]
```

NO 3

```
In [11]: # cetak rata-rata nilai rata-rata auc score
score.mean()

Out[11]: 0.6903477711901624
```

NO 4

```
In [28]: # Prediksi data test dengan model yang telah kalian buat
auc_score = y_predict.fit(y_test, y_predict)
auc_score

-----
AttributeError Traceback (most recent call last)
<ipython-input-28-8bddec90921c> in <module>
      1 # Prediksi data test dengan model yang telah kalian buat
----> 2 auc_score = y_predict.fit(y_test, y_predict)
      3 auc_score

AttributeError: 'numpy.ndarray' object has no attribute 'fit'
```

NO 5

```
In [29]: # import library confusion matrix dan classification report
from sklearn.metrics import classification_report
```

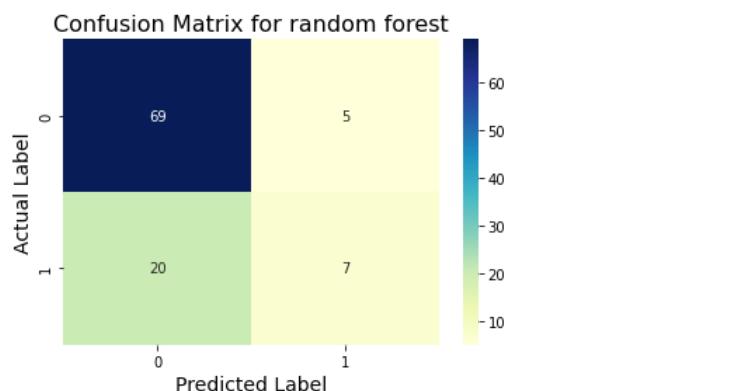
```
In [43]: # apply confusion matrix dan cetak nilai confusion matrix
cm = classification_report(y_test, y_predict, labels = (1,2))
cm
```

```
Out[43]: '          precision    recall   f1-score   support\n          1         0.78      0.93      0.85     74\n          0         0.75      0.75      0.75    101\nweighted avg       0.72      0.72      0.72    101\nmacro avg       0.75      0.75      0.75     74'
```

```
In [48]: # visualisasikan nilai confusion matrix ke dalam diagram heatmap
from sklearn import metrics
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

cm = confusion_matrix(y_test,y_predict)
p = sns.heatmap(pd.DataFrame(cm), annot=True, cmap="YlGnBu" ,fmt='g')
plt.title('Confusion Matrix for random forest', fontsize=16)
plt.xlabel('Predicted Label', fontsize=14)
plt.ylabel('Actual Label', fontsize=14)
```

```
Out[48]: Text(33.0, 0.5, 'Actual Label')
```

**NO 6**

```
In [49]: # cetak nilai classification_report
print(classification_report(y_test, y_predict))
```

	precision	recall	f1-score	support
1	0.78	0.93	0.85	74
2	0.58	0.26	0.36	27
accuracy			0.75	101
macro avg	0.68	0.60	0.60	101
weighted avg	0.72	0.75	0.72	101

NO 7

- Bagaimana hasil confusion matrix dari hasil prediksi tersebut?
Jawab disini
- Bagaimana classification report dari hasil prediksi tersebut?
Jawab disini
- Seberapa baik model anda dalam memprediksi seorang pasien mempunyai status positive? dari hasil classification_report diatas
Jawab disini
- Seberapa baik model anda dalam memprediksi seorang pasien mempunyai status negatif? dari hasil classification_report diatas
Jawab disini

Soal 3: Pemahaman Tentang Model Selection

Jelaskan dengan bahasa sendiri!

1. Apa itu Bias dan Variance?
2. Apa itu Overfitting dan Underfitting?
3. Apa yang bisa kita lakukan untuk mengatur kompleksitas dari model?
4. Bagaimana model yang baik?
5. Kapan kita menggunakan GridSearchCV dan kapan menggunakan RandomizedSearchCV?

Jawab

Soal 4: Aplikasi Model Selection

1. Bagi kedua data berikut ini menjadi data training dan data test dengan test_size=0.25.
2. Import library KNN dan GridSearchCV.
3. Gunakan algoritma KNN dan fungsi GridSearchCV untuk hyperparameter tuning dan model selection.
4. jumlah fold bebas!, gunakan scoring 'roc_auc'
5. Definisikan kombinasi hyperparameter untuk model selection dengan GridSearchCV. Kombinasi Hyperparameter bebas, baca lagi dokumentasi KNN di link berikut <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html> (<https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>) untuk memahami lagi jenis2 hyperparameter di algoritma KNN.
6. Latih model terhadap data training.
7. Apa hyperparameter terbaik untuk kombinasi hyperparameter kalian?
8. Berapa score validasi terbaik dari model tersebut?
9. Prediksi probabilitasi output dari model yang telah dibuat terhadap data test. note : gunakan method .predict_proba() untuk menghasilkan output probabilitas
10. Berapa nilai score roc_auc untuk data test? (y_predict)
11. Apakah model anda termasuk baik, overfitting, atau underfitting?

Load Dataset

```
In [50]: # 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"]
df2 = pd.read_csv(url, names=list_cols)
```

```
In [51]: # tampilkan 5 baris awal dataset dengan function head()
df2.head()
```

Out[51]:

	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 [52]: # hitung jumlah masing" data pada kolom survival_status
df2['survival_status'].value_counts()
```

Out[52]:

1	225
2	81

Name: survival_status, dtype: int64

NO 1

```
In [53]: # 1. pembagian variabel train dan test
# test size= 25%, random state = 42, dan stratify = y
X = df2.drop('survival_status', axis = 1)
y = df2['survival_status']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=25, random_state=42,stratify = y)
```

NO 2

```
In [74]: # 2. import Library KNN dan GridSearchCV
from sklearn.neighbors import KNeighborsClassifier

from sklearn.model_selection import GridSearchCV
```

NO 3 - 6

```
In [62]: # 3. tuning hyperparameter dengan GridSearchCV (parameter cv=10)
## build model KNN
model_knn = KNeighborsClassifier()
param_grid = {'n_neighbors' : np.arange(3,51), 'weights' : ['uniform','distance']}
gscv = GridSearchCV(model_knn, param_grid, scoring='roc_auc', cv = 10)
gscv.fit(X_train, y_train)
```

```
Out[62]: GridSearchCV(cv=10, estimator=KNeighborsClassifier(),
param_grid={'n_neighbors': array([ 3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50]),
'weights': ['uniform', 'distance']},
scoring='roc_auc')
```

NO 7

```
In [66]: # 7. parameter terbaik
gscv.best_params_

Out[66]: {'n_neighbors': 50, 'weights': 'uniform'}
```

NO 8

```
In [68]: # 8. score validasi terbaik
gscv.best_score_

Out[68]: 0.7126509353741496
```

NO 9

```
In [72]: # 9. prediksi probabilitas masing-masing data test
y_predict = gscv.fit(X_test)
y_predict

C:\Users\dwiah\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:548: FitFailedWarning: Estimator fit failed. The score on this train-test partition for these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\dwiah\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py", line 529, in _fit_and_
    estimator.fit(X_train, **fit_params)
TypeError: fit() missing 1 required positional argument: 'y'

    warnings.warn("Estimator fit failed. The score on this train-test"
-----
TypeError                                     Traceback (most recent call last)
<ipython-input-72-1ebf3d64ba50> in <module>
      1 # 9. prediksi probabilitas masing-masing data test
----> 2 y_predict = gscv.fit(X_test)
      3 y_predict

~/anaconda3/lib/site-packages\sklearn\utils\validation.py in inner_f(*args, **kwargs)
    71         FutureWarning)
    72     kwargs.update({k: arg for k, arg in zip(sig.parameters, args)})
--> 73     return f(**kwargs)
    74
    75

~/anaconda3/lib/site-packages\sklearn\model_selection\_search.py in fit(self, X, y, groups, **fit_params)
    765         self.best_estimator_.fit(X, y, **fit_params)
    766     else:
--> 767         self.best_estimator_.fit(X, **fit_params)
    768     refit_end_time = time.time()
    769     self.refit_time_ = refit_end_time - refit_start_time

TypeError: fit() missing 1 required positional argument: 'y'
```

```
In [80]: # nilai rata-rata probabilitas data test
y_predict.mean()
```

```
Out[80]: 1.118811881188119
```

NO 10

```
In [153]: # 10. nilai score roc_auc
kurang_5th = roc_auc[:,1]
print(kurang_5th)

[0.16756916 0.17051611 0.16345985 0.11362437 0.20646919 0.14235942
 0.          0.08940655 0.          0.59208121 0.25152363 0.14148435
 0.25618281 0.60656564 0.12407537 0.16972843 0.18108432 0.15210734
 0.18027431 0.45610922 0.22527062 0.18097357 0.          0.123134
 0.51750434 0.54996576 0.42456436 0.          0.18471835 0.08182622
 0.          0.12407537 0.          0.32365645 0.18510126 0.
 0.08598644 0.16299863 0.34549723 0.28399464 0.13570151 0.32815
 0.29270146 0.67340073 0.14847957 0.12473245 0.1342232 0.
 0.14412323 0.18529338 1.          0.41369782 0.21534365 0.1726462
 0.06934707 0.21228668 0.40228092 0.14180065 0.13197082 0.15725626
 0.20849651 0.18027431 0.15104146 0.12348324 0.19489063 0.
 0.38934982 0.2209358 0.60105079 0.25547443 0.13157023 0.34982587
 0.21589415 0.0855198 0.22196137 0.23216426 0.13983004]
```

NO 11

Jawab

Soal 5:

1. Ulangi tahap di atas (soal 4, no 1 - 8) namun kali ini menggunakan algoritma DecisionTreeClassifier dan kalian bisa menggunakan RandomizedSearchCV apabila process training lama. pelajari algoritma DecisionTreeClassifier di linkberikut: <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html?highlight=decisiontreeclassifier#sklearn.tree.DecisionTreeClassifier>
2. Bandingkan scorenya dengan Algoritma KNN, mana yang lebih baik?

Note : Data Science adalah experiment, sangat di dimungkinkan memerlukan beberapa kali percobaan untuk mendapatkan hasil yang terbaik! Happy Coding :)

NO 1

```
In [73]: # 1. import algoritma DecisionTreeClassifier
from sklearn.tree import DecisionTreeClassifier

In [76]: # Build model decision tree classifier
model_tree = DecisionTreeClassifier()
params = {'criterion' : ['entropy','gini'], 'splitter' : ['best', 'random'],
          'min_samples_split' : np.arange(2,50)}
gscv = GridSearchCV(model_tree, param_grid = params, cv = 10, scoring ='roc_auc')
gscv.fit(X_train, y_train)

Out[76]: GridSearchCV(cv=10, estimator=DecisionTreeClassifier(),
param_grid={'criterion': ['entropy', 'gini'],
'min_samples_split': array([ 2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 1
7, 18,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49]),
'splitter': ['best', 'random']},
scoring='roc_auc')

In [77]: # parameter terbaik
gscv.best_params_

Out[77]: {'criterion': 'entropy', 'min_samples_split': 24, 'splitter': 'random'}

In [79]: # score validasi terbaik
gscv.best_score_

Out[79]: 0.7288265306122449
```

NO 2

Jawab