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:

1.
2. data latih adalah data untuk melatih model biasanya kisaran 70% dari dataset

- data validasi adalah data untuk proses validasi model guna mencegah overfitting biasanya 20 %
- data test adalah data untuk testing model biasanya kisaran 10 % dari dataset dan tidak boleh menggunakan dari data yang sudah dipake di data latih ataupun data validasi

1. dilihat dari akurasinya , bisa melalui nilai confusion matrix
2. confusion matrix untuk memberikan perbandingan hasil klasifikasi yang dilakukan oleh model dengan hasil sebenarnya. ada 4 klasifikasi yang dilakuakan yaitu True Positive(TP) , True Negative(TN) , False Positive (FP) , False Negative (FN)
3. Classification Report adalah untuk mempermudah melakukan laporan seperti menampilkan precision, recall, f1, dan support sukses dari suatu model

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

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 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

import seaborn as sns
```

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(Xs,y,test_size=0.25,random_state=42,stratify=y)
```

In [7]:
```
## 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 [8]:
```
## 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 [9]:
```
# cetak rata-rata nilai rata-rata auc score
score.mean()
```

Out[9]: 0.6903477711901624

NO 4

In [10]:
```
# Prediksi data test dengan model yang telah kalian buat
auc_score = roc_auc_score(y_test, y_predict)
auc_score
```

Out[10]: 0.5399122807017543

NO 5

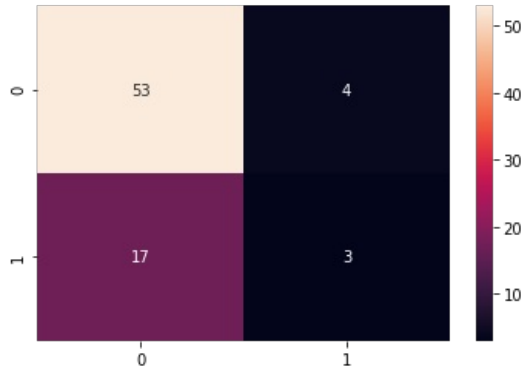
In [11]:
```
# import library confusion matrix dan classification report
from sklearn.metrics import confusion_matrix ,classification_report
```

In [12]:
```
# apply confusion matrix dan cetak nilai confusion matrix
cm = confusion_matrix(y_test, y_predict, labels = (1,2))
cm
```

Out[12]:
```
array([[53,  4],
       [17,  3]], dtype=int64)
```

In [13]:
```
# visualisasikan nilai confusion matrix ke dalam diagram heatmap
sns.heatmap(pd.DataFrame(cm), annot=True)
```

Out[13]: <AxesSubplot:>



NO 6

In [14]:
```
# cetak nilai classification_report
print(classification_report(y_test, y_predict))
```

```
              precision    recall  f1-score   support

           1       0.76      0.93      0.83        57
           2       0.43      0.15      0.22        20
```

```
    accuracy                           0.73        77
   macro avg       0.59      0.54      0.53        77
weighted avg       0.67      0.73      0.68        77
```

NO 7

- Bagaimana hasil confusion matrix dari hasil prediksi tersebut?
  jawab disini dari gambar confusion model ada 53 orang sudah operasi ,model meprediksi 4 orang tidak bisa hidup lebih dari 5 tahun

  dan dari 17 belum operasi model memprediksi 3 orang yang bisa hidup lebih dari 5 tahun
- Bagaimana classification report dari hasil prediksi tersebut?
  jawab disini precision1 pasien yg hidup lebih dari 5 tahun sangat bagus untuk prediksi
- Seberapa baik model anda dalam memprediksi seorang pasien mempunyai status positive? dari hasil classification_report diatas
  jawab disini bagus nilainya seperti precision 0.76
- Seberapa baik model anda dalam memprediksi seorang pasien mempunyai status negatif? dari hasil classification_report diatas
  jawab disini agak kurang nilainya seperti precision 0.43

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

1.
2. Bias kesalahan sistematis yang terjadi pada model karena asumsi yang salah dalam proses pemodelan misal prediksi model dengan data sebenarnya salah

- Variance adalah variabel dari prediksi model misal model bekerja baik pada data pengujian atau validasi tapi saat pada data test akurasinya kurang baik

1.
2. Overfitting adalah model ML sangat bergantung pada data pelatihan , disebut overfitting misal saat Model dengan varian tinggi dan bias rendah

- Underfitting adalah Ketidakmampuan model untuk memahami / menganalisis tren / struktur yang mendasari data, tidak dapat memperoleh informasi penting dari data pelatihan. Ini biasanya terjadi jika data pelatihan tidak mencukup , disebut underfitting saat Model dengan bias tinggi dan varians rendah

1. menggunakan metode evaluasi model dan seleksi fitur
2. model yang meminimalisir terjadinya tidak overfiting dan tidak underfitting
3. Grid Search metode yang efektif dalam supervised learning untuk menyesuaikan parameter meningkatkan performa generalisasi model , RandomizedSearchCV digunakan ketika kita memiliki banyak parameter untuk dicoba dan waktu pelatihannya sangat lama

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 untuk memahami lagi jenis2 hyperparameter di algorithma 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 di buat 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, overtting, atau underfitting?

Load Dataset

In [15]:
```
# 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 [16]:
```
# tampilkan 5 baris awal dataset dengan function head()
df2.head()
```

Out[16]:

	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 [17]:
```
# hitung jumlah masing" data pada kolom survival_status
df2['survival_status'].value_counts()
```

Out[17]:
```
1    225
2     81
Name: survival_status, dtype: int64
```

NO 1

In [18]:
```
# 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=0.25,random_state=42,stratify=y)
```

NO 2

In [19]:
```
# 2. import library KNN dan GridSearchCv
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import GridSearchCV
```

NO 3 - 6

In [20]:
```
# 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[20]:
```
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 [21]:
```
# 7. parameter terbaik
gscv.best_params_
```

Out[21]:
```
{'n_neighbors': 44, 'weights': 'distance'}
```

NO 8

In [22]:
```
# 8. score validasi terbaik
gscv.best_score_
```

Out[22]:
```
0.7319065126050421
```

NO 9

In [23]:
```
# 9. prediksikan probabilitas masing-masing data test
y_predict = gscv.predict_proba(X_test)
y_predict
```

Out[23]:
```
array([[0.83243084, 0.16756916],
       [0.82948389, 0.17051611],
       [0.83654015, 0.16345985],
       [0.88637563, 0.11362437],
       [0.79353081, 0.20646919],
       [0.85764058, 0.14235942],
       [1.        , 0.        ],
       [0.91059345, 0.08940655],
       [1.        , 0.        ],
       [0.40791879, 0.59208121],
       [0.74847637, 0.25152363],
       [0.85851565, 0.14148435],
       [0.74381719, 0.25618281],
       [0.39343436, 0.60656564],
       [0.87592463, 0.12407537],
       [0.83027157, 0.16972843],
       [0.81891568, 0.18108432],
       [0.84789266, 0.15210734],
       [0.81972569, 0.18027431],
       [0.54389078, 0.45610922],
       [0.77472938, 0.22527062],
       [0.81902643, 0.18097357],
       [1.        , 0.        ],
       [0.876866  , 0.123134  ],
       [0.48249566, 0.51750434],
       [0.45003424, 0.54996576],
       [0.57543564, 0.42456436],
       [1.        , 0.        ],
       [0.81528165, 0.18471835],
       [0.91817378, 0.08182622],
       [1.        , 0.        ],
       [0.87592463, 0.12407537],
       [1.        , 0.        ],
       [0.67634355, 0.32365645],
       [0.81489874, 0.18510126],
       [1.        , 0.        ],
       [0.91401356, 0.08598644],
       [0.83700137, 0.16299863],
       [0.65450277, 0.34549723],
       [0.71600536, 0.28399464],
       [0.86429849, 0.13570151],
       [0.67185   , 0.32815   ],
       [0.70729854, 0.29270146],
       [0.32659927, 0.67340073],
       [0.85152043, 0.14847957],
       [0.87526755, 0.12473245],
       [0.8657768 , 0.1342232 ],
```

```
[1.        , 0.        ],
[0.85587677, 0.14412323],
[0.81470662, 0.18529338],
[0.        , 1.        ],
[0.58630218, 0.41369782],
[0.78465635, 0.21534365],
[0.8273538 , 0.1726462 ],
[0.93065293, 0.06934707],
[0.78771332, 0.21228668],
[0.59771908, 0.40228092],
[0.85819935, 0.14180065],
[0.86802918, 0.13197082],
[0.84274374, 0.15725626],
[0.79150349, 0.20849651],
[0.81972569, 0.18027431],
[0.84895854, 0.15104146],
[0.87651676, 0.12348324],
[0.80510937, 0.19489063],
[1.        , 0.        ],
[0.61065018, 0.38934982],
[0.7790642 , 0.2209358 ],
[0.39894921, 0.60105079],
[0.74452557, 0.25547443],
[0.86842977, 0.13157023],
[0.65017413, 0.34982587],
[0.78410585, 0.21589415],
[0.9144802 , 0.0855198 ],
[0.77803863, 0.22196137],
[0.76783574, 0.23216426],
[0.86016996, 0.13983004]])
```

In [24]:
```
# nilai rata-rata probabilitas data test
y_predict.mean()
```

Out[24]: 0.5

NO 10

In [32]:
```
# 10. nilai score roc_auc
kurang_5th = roc_auc_score(y_test, y_predict[:,1])
print(kurang_5th)
```

```
0.5557017543859649
```

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 [33]:
```
# 1. import algoritma DecisionTreeClassifier
from sklearn.tree import DecisionTreeClassifier
```

In [34]:
```
# 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[34]:
```
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,
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]),
                         'splitter': ['best', 'random']},
             scoring='roc_auc')
```

In [35]:
```
# parameter terbaik
gscv.best_params_
```

Out[35]:
```
{'criterion': 'gini', 'min_samples_split': 33, 'splitter': 'random'}
```

In [36]:
```
# score validasi terbaik
gscv.best_score_
```

Out[36]:
```
0.7411983543417366
```

NO 2

Jawab

Score Decision Tree Classifier 0.74 sedangkan Score yang pake knn Classifier 0.73 jadi lebih bagus menggunakan Desicion tree classifier