

Modul Praktikum

Pembelajaran Mesin (*Machine Learning*)



Pertemuan 2 Evaluasi Algoritma

Evaluasi Algoritma

Pada praktikum minggu lalu telah membahas mengenai python dan melakukan summary dataset yang terdiri dari menentukan dimensi dari dataset, melihat isi dataset, distribusi kelas data, melihat ringkasan statistik dan cara visualisasi data menggunakan plot univariat dan multivariat.

Pada praktikum minggu ini akan membahas mengenai evaluasi algoritma. Evaluasi algoritma digunakan untuk memperkirakan akurasi dari data yang belum diketahui sebelumnya.

Terdapat beberapa model yang dapat digunakan pada evaluasi algoritma antara lain :

1. K-Nearest Neighbors (KNN).
2. Gaussian Naive Bayes (NB).
3. Support Vector Machines (SVM).

Untuk memperkirakan akurasi pada data yang tidak diketahui sebelumnya terdapat beberapa data yang dapat dilakukan :

1. Membuat Validasi Dataset
2. K-Folds cross Validation
3. Membangun Model
4. Memilih model terbaik

I. Membuat Validasi Dataset

Validasi ini dilakukan untuk mengetahui bahwa model yang dibuat itu bagus. Kita akan menggunakan metode statistik untuk memperkirakan keakuratan model yang dibuat pada data yang tidak terlihat. Juga diinginkan perkiraan yang lebih konkret mengenai keakuratan model terbaik pada data yang tidak terlihat dengan mengevaluasi data aktual yang tidak terlihat.

Artinya, kita akan menahan beberapa data yang tidak dapat dilihat oleh algoritma dan akan menggunakan data ini untuk mendapatkan informasi tentang seberapa akurat model terbaik sebenarnya.

Kita akan membagi datateset yang telah dimuat menjadi dua, 80% diantaranya akan digunakan untuk melatih model dan 20% digunakan untuk

data validasi. (Dataset yang digunakan berupa 150 Bunga Iris pada praktikum sebelumnya)

```
# Split-out validation dataset

array = dataset.values

X = array[:,0:4]

Y = array[:,4]

validation_size = 0.20

seed = 7

X_train, X_validation, Y_train, Y_validation =
model_selection.train_test_split(X, Y, test_size=validation_size,
random_state=seed)
```

Setelah perintah di atas dieksekusi, kita sudah memiliki dua data yaitu X_train dan Y_train untuk mempersiapkan model dan rangkaian X_validation dan Y_validation yang dapat digunakan selanjutnya.

II. K-Folds Cross Validation

Kita akan menggunakan validasi silang 10 kali lipat untuk memperkirakan akurasi. Untuk itu dataset dibagi menjadi 10 bagian, 9 untuk latihan dan 1 untuk pengujian dan ulangi untuk semua kombinasi.

```
# Test options and evaluation metric

seed = 7

scoring = 'accuracy'
```

III. Membangun Model

Untuk mengetahui algoritma yang cocok dengan studi kasus ini maka kita harus mengevaluasi dengan beberapa algoritma. Pada praktikum kali ini hanya akan membahas tiga model yaitu K-Nearest Neighbors (KNN), Gaussian Naive Bayes (NB) dan Support Vector Machines (SVM).

```
# Spot Check Algorithms

models = []

models.append(('KNN', KNeighborsClassifier()))

models.append(('NB', GaussianNB()))
```

```

models.append(('SVM', SVC()))

# evaluate each model in turn
results = []
names = []

for name, model in models:

    kfold = model_selection.KFold(n_splits=10, random_state=seed)
    cv_results = model_selection.cross_val_score(model, X_train, Y_train,
    cv=kfold, scoring=scoring)

    results.append(cv_results)
    names.append(name)

    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)

```

IV. Memilih Model Terbaik

Jika sudah memiliki hasil evaluasi dari ketiga model diatas untuk memilih model terbaik dilakukan dari perbandingan satu sama lainnya dan dipilih yang paling akurat.

Dari eksekusi script di atas, kita mendapatkan hasil mentah sebagai berikut:

1. KNN: 0.983333 (0.033333)
2. NB: 0.975000 (0.053359)
3. SVM: 0.991667 (0.025000)

Dari hasil output di atas, kita dapat melihat bahwa SVM memiliki nilai akurasi perkiraan terbesar. Setelah mengetahui model yang paling akurat yaitu SVM, selanjutnya kita dapat mencoba melakukan pengujian tentang keakuratan model SVM terhadap data yang ada.

```

# Make predictions on validation dataset
svm = SVC()

svm.fit(X_train, Y_train)

predictions = svm.predict(X_validation)

print(accuracy_score(Y_validation, predictions))

```

```
print(confusion_matrix(Y_validation, predictions))  
print(classification_report(Y_validation, predictions))
```

Menghasilkan output :

```
0.9333333333333333  
[[ 7  0  0]  
 [ 0 10  2]  
 [ 0  0 11]]  
  
              precision    recall  f1-score   support  
  
   Iris-setosa           1.00        1.00        1.00         7  
Iris-versicolor           1.00        0.83        0.91        12  
   Iris-virginica          0.85        1.00        0.92        11  
  
avg / total             0.94        0.93        0.93        30
```

Tugas!

1. Lakukan Eksplorasi terhadap algoritma klasifikasi lain yang ada!
2. Buatlah evaluasi algoritma dengan dataset yang telah dicoba pada tugas praktikum sebelumnya (dengan menggunakan 3 model yaitu KNN, NB dan SVM)!