EVALUASI ALGORITMA
PRAKTIKUM PEMBELAJARAN MESIN

Disusun Oleh:

Nashirudin Baqiy 24060119130045

Lab A2

DEPARTEMEN ILMU KOMPUTER / INFORMATIKA

FAKULTAS SAINS DAN MATEMATIKA

UNIVERSITAS DIPONEGORO

2021

BAB I PENDAHULUAN

1.1. Rumusan Masalah

1.1.1. Lakukan Eksplorasi terhadap algoritma klasifikasi lain yang ada!

1.1.2. Buatlah evaluasi algoritma dengan dataset yang telah dicoba pada tugas praktikum sebelumnya (dengan menggunakan 3 model yaitu KNN, NB dan SVM)!

1.2. Tujuan

1.2.1. Melakukan eksplorasi algoritma klasifikasi machine learning, selain k-NN, Naive bayes, dan SVM.

1.2.2. Melakukan evaluasi algoritma, menggunakan k-NN, Naive Bayes, dan SVM, dengan dataset yang telah di preprocessing pada praktikum sebelumnya.

1.3. Dasar Teori

Pembelajaran mesin (ML) adalah studi tentang algoritma komputer yang dapat ditingkatkan secara otomatis melalui pengalaman dan penggunaan data. Itu dilihat sebagai bagian dari kecerdasan buatan. Algoritma pembelajaran mesin membangun model berdasarkan data sampel, yang dikenal sebagai "data pelatihan", untuk membuat prediksi atau keputusan tanpa diprogram secara eksplisit untuk melakukannya. Algoritma pembelajaran mesin digunakan dalam berbagai macam aplikasi, seperti dalam kedokteran, penyaringan email, pengenalan suara, dan visi komputer, di mana sulit atau tidak mungkin untuk mengembangkan algoritma konvensional untuk melakukan tugas-tugas yang diperlukan.

Bagian dari pembelajaran mesin terkait erat dengan statistik komputasi, yang berfokus pada pembuatan prediksi menggunakan komputer; tetapi tidak semua pembelajaran mesin adalah pembelajaran statistik. Studi

optimasi matematika memberikan metode, teori, dan domain aplikasi ke bidang pembelajaran mesin. Data mining adalah bidang studi terkait, dengan fokus pada analisis data eksplorasi melalui pembelajaran tanpa pengawasan. Beberapa implementasi pembelajaran mesin menggunakan data dan jaringan saraf dengan cara yang meniru kerja otak biologis. Dalam penerapannya di seluruh masalah bisnis, machine learning juga disebut sebagai analitik prediktif.

BAB II PEMBAHASAN

Eksplorasi Algoritma

2.1. Deskripsi dataset

Dataset yang digunakan adalah dataset bunga iris yang sudah sangat terkenal.

2.2. Splitting data train dan test

Sebelum splitting dataset, perlu dilakukan memisahkan fitur dengan kelasnya. Setelah itu dilakukan baru kemudian splitting dataset dilakukan.

```
[5]  array = dataset.values

     X = array[:,0:4]
     y = array[:,4]

     X_train, X_validation, y_train, y_validation = train_test_split(X, y, test_size=0.2, random_state=42)

     # print(X_train)
     # print(y_train)
```

2.3. Naive Bayes

Berikut ini hasil dari model naive bayes.

```
[('Naive Bayes', GaussianNB(priors=None, var_smoothing=1e-09)), ('Random Forest', RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)), ('Logistic Regression', LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                    intercept_scaling=1, l1_ratio=None, max_iter=100,
                    multi_class='auto', n_jobs=None, penalty='l2',
                    random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                    warm_start=False))]
```

2.4. Random Forest

Random forest menghasilkan akurasi 100% dengan detail sebagai berikut.

```
1.0
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
                precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        10
Iris-versicolor       1.00      1.00      1.00         9
 Iris-virginica       1.00      1.00      1.00        11

       accuracy                           1.00        30
      macro avg       1.00      1.00      1.00        30
   weighted avg       1.00      1.00      1.00        30
```

2.5. Logistic Regression

Logistic regression juga menghasilkan akurasi 100%.

```
1.0
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
                precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        10
Iris-versicolor       1.00      1.00      1.00         9
 Iris-virginica       1.00      1.00      1.00        11

       accuracy                           1.00        30
      macro avg       1.00      1.00      1.00        30
   weighted avg       1.00      1.00      1.00        30
```

Evaluasi dengan dataset sendiri

2.1. Deskripsi dataset

Dataset yang akan digunakan adalah dataset dari UCI yang berisi data mengenai prediksi pasar saham dengan 9472 baris dan 15 kolom. File dataset yang digunakan bernama AirQualityUCI.csv yang telah di preprocessing yang mebedakan praktikum pertama dengan praktikum kali ini, fitur yang digunakan sebanyak 5 fitur dengan total 15 kolom beserta dengan class/label.

2.2. Preprocessing dataset

Preprocessing yang dilakukan sama seperti pada praktikum sebelumnya perbedaannya hanya saja pada di praktikum kali ini,

fitur yang dihilangkan adalah fitur ‘Name’ dan ‘Date’ sehingga sisa total fitur ada 5 fitur.

```
                                                Date  ...  AH
0  Date;Time;CO(GT);PT08.S1(CO);NMHC(GT);C6H6(GT)...  ... NaN
1                              10/03/2004;18.00.00;2  ... NaN
2                  10/03/2004;19.00.00;2;1292;112;9  ... NaN
3                              10/03/2004;20.00.00;2  ... NaN
4                              10/03/2004;21.00.00;2  ... NaN

[5 rows x 15 columns]
================================================================================
                                                Date  ... True hourly averaged NO2
0    Date;Time;CO(GT);PT08.S1(CO);NMHC(GT);C6H6(GT)...  ...                      NaN
1                              10/03/2004;18.00.00;2  ...                      NaN
2                  10/03/2004;19.00.00;2;1292;112;9  ...                      NaN
3                              10/03/2004;20.00.00;2  ...                      NaN
4                              10/03/2004;21.00.00;2  ...                      NaN
...                                              ...  ...                      ...
9467                                ;;;;;;;;;;;;;;;;  ...                      NaN
9468                                ;;;;;;;;;;;;;;;;  ...                      NaN
9469                                ;;;;;;;;;;;;;;;;  ...                      NaN
9470                                ;;;;;;;;;;;;;;;;  ...                      NaN
9471                                ;;;;;;;;;;;;;;;;  ...                      NaN

[9472 rows x 10 columns]
================================================================================
0      NaN
1      NaN
2      NaN
3      NaN
4      NaN
       ..
9467   NaN
9468   NaN
9469   NaN
9470   NaN
9471   NaN
Name: AH, Length: 9472, dtype: float64
```

2.3. Splitting data train dan data test

Langkah berikutnya adalah memisahkan data train dan data test dari dataset ini. Proses splitting dilakukan dengan bantuan library sklearn, yaitu train_test_split dengan porsi data test sebesar 20%. Perlu di garis bawahi, karena dataset sudah balance, maka tidak perlu dilakukan proses imbalancing dataset.

```
 [7.2 3.0 5.8 1.6]
 [4.9 3.1 1.5 0.1]
 [6.7 3.1 5.6 2.4]
 [4.9 3.0 1.4 0.2]
 [6.9 3.1 4.9 1.5]
 [7.4 2.8 6.1 1.9]
 [6.3 2.9 5.6 1.8]
 [5.7 2.8 4.1 1.3]
 [6.5 3.0 5.5 1.8]
 [6.3 2.3 4.4 1.3]
 [6.4 2.9 4.3 1.3]
 [5.6 2.8 4.9 2.0]
 [5.9 3.0 5.1 1.8]
 [5.4 3.4 1.7 0.2]
 [6.1 2.8 4.0 1.3]
 [4.9 2.5 4.5 1.7]
 [5.8 4.0 1.2 0.2]
 [5.8 2.6 4.0 1.2]
 [7.1 3.0 5.9 2.1]]
========================================================================
['Iris-setosa' 'Iris-setosa' 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa'
 'Iris-virginica' 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa'
 'Iris-setosa' 'Iris-virginica' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica'
 'Iris-virginica' 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor'
 'Iris-virginica' 'Iris-versicolor' 'Iris-setosa' 'Iris-virginica'
 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa' 'Iris-setosa'
 'Iris-setosa' 'Iris-versicolor' 'Iris-setosa' 'Iris-versicolor'
 'Iris-virginica' 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica'
 'Iris-setosa' 'Iris-virginica' 'Iris-virginica' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor' 'Iris-setosa'
 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa' 'Iris-setosa'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa' 'Iris-virginica'
 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-virginica' 'Iris-versicolor' 'Iris-virginica' 'Iris-virginica'
 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa' 'Iris-virginica'
 'Iris-virginica' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa' 'Iris-virginica'
 'Iris-virginica' 'Iris-setosa' 'Iris-versicolor' 'Iris-versicolor'
```

2.4. Naive Bayes

Model pertama yang digunakan untuk pengklasifikasian kali ini adalah naive bayes. Pada saat training, model ini memberikan nilai rata-rata cross validation score sebesar 0.95 dan pada saat testing/validation hasil yang diberikan adalah sebesar 0.055277. Dengan hasil yang sangat mirip, maka bisa dipastikan model ini tidak overfit ataupun underfit.

```
Accuracy  1.0

Confussion Matrix
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]

Report
                precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        10
Iris-versicolor       1.00      1.00      1.00         9
 Iris-virginica       1.00      1.00      1.00        11

       accuracy                           1.00        30
      macro avg       1.00      1.00      1.00        30
   weighted avg       1.00      1.00      1.00        30
```

2.5. kNN

Model yang kedua adalah kNN. Pada saat training, model ini memberikan nilai rata-rata cross validation score sebesar 0.95 dan pada saat testing/validation hasil yang diberikan adalah sebesar 0.055277. Dengan hasil yang sangat mirip, maka bisa dipastikan model ini tidak overfit ataupun underfit.

```
Accuracy  1.0

Confussion Matrix
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]

Report
                precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        10
Iris-versicolor       1.00      1.00      1.00         9
 Iris-virginica       1.00      1.00      1.00        11

       accuracy                           1.00        30
      macro avg       1.00      1.00      1.00        30
   weighted avg       1.00      1.00      1.00        30
```

2.6. Support Vector Machine (SVM)

Model terakhir yang digunakan pada praktikum ini adalah SVM. Pada saat training, model ini memberikan nilai rata-rata cross validation score sebesar 0.958333 dan pada saat testing/validation hasil yang diberikan adalah sebesar 0.041667. Dengan hasil yang sangat mirip, maka bisa dipastikan model ini tidak overfit ataupun underfit.

```
Accuracy  1.0

Confussion Matrix
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]

Report
                precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        10
Iris-versicolor       1.00      1.00      1.00         9
 Iris-virginica       1.00      1.00      1.00        11

       accuracy                           1.00        30
      macro avg       1.00      1.00      1.00        30
   weighted avg       1.00      1.00      1.00        30
```

BAB III PENUTUP

3.1. Kesimpulan

Pada praktikum ini mempelajari mengenai algoritma machine learning untuk melakukan klasifikasi, yang terdiri dari kNN, Naive Bayes, dan SVM. Hasil dari ketiga algoritma tersebut bervariasi tergantung dari dataset yang digunakan, preprocessing yang dilakukan, dan pengaturan dari algoritma itu sendiri. Pada praktikum kali ini SVM dan kNN memberikan hasil yang sama baiknya untuk dataset *Air Quality Data Set.*

.

DAFTAR PUSTAKA

S. De Vito, E. Massera, M. Piga, L. Martinotto, G. Di Francia, On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario, Sensors and Actuators B: Chemical, Volume 129, Issue 2, 22 February 2008, Pages 750-757, ISSN 0925-4005

Saverio De Vito, Marco Piga, Luca Martinotto, Girolamo Di Francia, CO, NO2 and NOx urban pollution monitoring with on-field calibrated electronic nose by automatic bayesian regularization, Sensors and Actuators B: Chemical, Volume 143, Issue 1, 4 December 2009, Pages 182-191, ISSN 0925-4005

S. De Vito, G. Fattoruso, M. Pardo, F. Tortorella and G. Di Francia, 'Semi-Supervised Learning Techniques in Artificial Olfaction: A Novel Approach to Classification Problems and Drift Counteraction,' in IEEE Sensors Journal, vol. 12, no. 11, pp. 3215-3224, Nov. 2012.
doi: 10.1109/JSEN.2012.2192425