EVALUASI ALGORITMA

LAPORAN PRAKTIKUM
Disusun untuk Memenuhi Tugas Laporan Praktikum Pertemuan 2
Mata Kuliah Pembelajaran Mesin

DISUSUN OLEH:
LINGGAR MARETVA CENDANI
24060117120031

PROGRAM STUDI STRATA 1 INFORMATIKA
DEPARTEMEN ILMU KOMPUTER/INFORMATIKA
FAKULTAS SAINS DAN MATEMATIKA
UNIVERSITAS DIPONEGORO
SEMARANG
2019

BAB I
RUMUSAN MASALAH DAN TUJUAN

1.1 Rumusan Masalah

1. Bagaimana cara meng-Import Library dan memuat data Haberman's Survival Data Set?
2. Bagaimana membuat validasi dataset pada Haberman's Survival Data Set?
3. Bagaimana cara melakukan K-Folds Cross Validation pada Haberman's Survival Data Set?
4. Bagaimana membangun model K-Nearest Neighbors (KNN), Gaussian Naive Bayes (NB) Dan Support Vector Machines (SVM) dari Haberman's Survival Data Set?
5. Bagaimana memilih model terbaik dari Haberman's Survival Data Set?

1.2 Tujuan

1. Mengetahui cara meng-Import Library dan memuat data Haberman's Survival Data Set.
2. Mengetahui cara membuat validasi dataset pada Haberman's Survival Data Set.
3. Mengetahui cara melakukan K-Folds Cross Validation pada Haberman's Survival Data Set.
4. Mengetahui cara membangun model K-Nearest Neighbors (KNN), Gaussian Naive Bayes (NB) Dan Support Vector Machines (SVM) dari Haberman's Survival Data Set.
5. Mengetahui cara memilih model terbaik dari Haberman's Survival Data Set.

BAB II
DASAR TEORI

2.1 Pandas Python Library

Pandas adalah sebuah librari berlisensi BSD dan open source yang menyediakan struktur data dan analisis data yang mudah digunakan dan berkinerja tinggi untuk bahasa pemrograman Python.

Dengan kata lain, Pandas adalah librari analisis data yang memiliki struktur data yang diperlukan untuk membersihkan data mentah ke dalam sebuah bentuk yang cocok untuk analisis (yaitu tabel). Pandas melakukan tugas penting seperti menyelaraskan data untuk perbandingan dan penggabungan set data, penanganan data yang hilang, dll, itu telah menjadi sebuah librari de facto untuk pemrosesan data tingkat tinggi dalam Python (yaitu statistik).

2.2 Matplotlib

Matplotlib adalah library Python 2D yang dapat menghasilkan plot dengan kualitas tinggi dalam berbagai format dan dapat digunakan di banyak platform.

Matplotlib dapat digunakan sebagai pembuat grafik dalam berbagai platform, seperti Python dan Jupyter. Grafik yang dapat dibuat beragam, seperti grafik garis, batang, lingkaran, histogram, dsb.

2.3 K-Nearest Neighbors

Algoritma K-Nearest Neighbor (K-NN) adalah sebuah metode klasifikasi terhadap sekumpulan data berdasarkan pembelajaran data yang sudah terklasifikasikan sebelumya. Termasuk dalam supervised learning, dimana hasil query instance yang baru diklasifikasikan berdasarkan mayoritas kedekatan jarak dari kategori yang ada dalam K-NN.

2.4 Gaussian Naive Bayes

Algoritma Naive Bayes merupakan sebuah metoda klasifikasi menggunakan metode probabilitas dan statistik yg dikemukakan oleh

ilmuwan Inggris Thomas Bayes. Algoritma Naive Bayes memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai Teorema Bayes. Ciri utama dr Naïve Bayes Classifier ini adalah asumsi yg sangat kuat (naïf) akan independensi dari masing-masing kondisi / kejadian.

2.5 Support Vector Machines

SVM merupakan salah satu metode klasifikasi dalam data mining. SVM juga dapat melakukan prediksi baik pada klasifikasi maupun regresi. Pada dasarnya SVM memiliki prinsip linear, akan tetapi kini SVM telah berkembang sehingga dapat bekerja pada masalah non-linear. Cara kerja SVM pada masalah non-linear adalah dengan memasukkan konsep kernel pada ruang berdimensi tinggi.

Pada ruang yang berdimensi ini, nantinya akan dicari pemisah atau yang sering disebut hyperplane. Hyperplane dapat memaksimalkan jarak atau margin antara kelas data. Hyperplane terbaik antara kedua kelas dapat ditemukan dengan mengukur margin dan kemudian mencari titik maksimalnya. Usaha dalam mencari hyperplane yang terbaik sebagai pemisah kelas-kelas adalah inti dari proses pada metode SVM.

BAB III
PEMBAHASAN

3.1 Import Library dan Memuat Data

Pertama, kita import terlebih dahulu library - library yang akan kita gunakan pada program python kita, untuk mengolah dataset yang kita miliki.

```
In [1]: import sys
        import scipy
        import numpy
        import matplotlib
        import sklearn
        import pandas
```

Seperti yang ditunjukkan pada gambar di atas, library yang kita import adalah Scipy, Numpy, Matplotlib, Pandas, dan Sklearn.

Selain itu, kita import juga dataset yang digunakan evaluasi algoritma ini. Disini saya menggunakan Haberman's Survival Data Set. Dataset ini berisi kasus-kasus dari penelitian yang dilakukan antara tahun 1958 dan 1970 di Rumah Sakit Billings University of Chicago tentang kelangsungan hidup pasien yang telah menjalani operasi untuk kanker payudara.

```
In [2]: url = "https://archive.ics.uci.edu/ml/machine-learning-databases/haberman/haberman.data"
        names = ['age', 'year', 'positive-axilliary', 'class']
        dataset = pandas.read_csv(url, names=names)
```

Dataset ini memiliki 4 attribut. Yang pertama adalah attribut age, yaitu usia pasien saat melakukan operasi, kemudian attribut year, yaitu tahun ketika operasi dilakukan, kemudian yang ketiga attribut positive-axilliary, yaitu jumlah noda positive-axilliary yang terdeteksi. Kemudian yang keempat adalah attribut kelas, yaitu attribut survival, yaitu status waktu bertahan hidup pasien, nilai 1 untuk yang pasien bertahan 5 tahun

atau lebih dan nilai 2 untuk pasien yang meninggal dalam 5 tahun. Semua attribut ini berjenis numerical.

3.2 Membuat Validasi Dataset

Validasi dilakukan untuk mengetahui bahwa model yang dibuat bagus. Akan dinggunakan 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, akan ditahan beberapa data yang tidak dapat dilihat oleh algoritma dan akan menggunakan data ini untuk mendapatkan informasi tentang seberapa akurat model terbaik sebenarnya.

```
In [4]: array = dataset.values
X = array[:,0:4]
Y = array[:,3]
validation_size = 0.20
seed = 7
from sklearn import model_selection
X_train, X_validation, Y_train, Y_validation = model_selection.train_test_split(X, Y, test_size=validation_size, random_state=se
```

Data dibagi menjadi dua, 80% untuk melatih model, sedangkan 20% untuk data validasi 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..

3.3 K-Folds Cross Validation

Disini digunakan 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.

```
In [5]: seed = 7
        scoring = 'accuracy'
```

3.4 Membangun Model

Untuk mengetahui algoritma yang cocok dengan studi kasus ini maka harus dilakukan evaluasi dengan beberapa algoritma. Pada

praktikum kali ini, hanya akan dibahas tiga model yaitu K-Nearest Neighbors (KNN), Gaussian Naive Bayes (NB) dan Support Vector Machines (SVM).

Sebelum itu, lakukan import library dari sklearn untuk penggunaan KNN, NB, dan SVM sebagai berikut

```
In [7]: from sklearn.neighbors import KNeighborsClassifier
        from sklearn.naive_bayes import GaussianNB
        from sklearn.svm import SVC
```

Kemudian, setelah itu, masukkan kode sebagai berikut.

```
In [8]: # 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)

        from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
```

Maka akan tercetak hasil evaluasi dengan metode KNN, NB, dan SVM. Berikut hasilnya.

```
KNN: 0.750333 (0.075854)
NB: 1.000000 (0.000000)
SVM: 0.738167 (0.085653)
```

3.5 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 ketiga model yang telah kita gunakan tadi, terlihat bahwa Naive Bayes memiliki nilai akurasi terbesar (1.000000). Kemudian

selanjutnya dapat kita lakukan pengujian keakuratan model Naive Bayes dengan dataset yang kita miliki.

Maka dari itu, masukkan kode sebagai berikut (Menggunakan GaussianNB()).

```
In [19]: # Make predictions on validation dataset
         from sklearn.naive_bayes import GaussianNB
         nb = GaussianNB()
         nb.fit(X_train, Y_train)
         predictions = nb.predict(X_validation)
         print(accuracy_score(Y_validation, predictions))
         print(confusion_matrix(Y_validation, predictions))
         print(classification_report(Y_validation, predictions))
```

Maka akan tercetak hasil sebagai berikut.

```
1.0
[[45  0]
 [ 0 17]]
              precision    recall  f1-score   support

           1       1.00      1.00      1.00        45
           2       1.00      1.00      1.00        17

    accuracy                           1.00        62
   macro avg       1.00      1.00      1.00        62
weighted avg       1.00      1.00      1.00        62
```

BAB IV
KESIMPULAN

4. 1 Kesimpulan

Dari tiga model evaluasi yang telah digunakan, yaitu model K-Nearest Neighbors (KNN), Gaussian Naive Bayes (NB) Dan Support Vector Machines (SVM), menunjukkan hasil yang berbeda - beda. Disinilah kita dapat menentukan model mana yang terbaik yang dapat kita gunakan.

Dari data yang telah didapatkan, setelah dilakukan evaluasi algoritma ketiga model pada Haberman's Survival Dataset, Gaussian Naive Bayes merupakan model terbaik yang dapat digunakan. Dengan tingkat akurasi mencapai 1.000000.

DAFTAR PUSTAKA

Informatics (21 Oktober 2019). "*Pertemuan 2 - Evaluasi Algoritma*". Makalah disajikan dalam Praktikum Pembelajaran Mesin di laboratorium komputer gedung E Universitas Diponegoro. Semarang, 21 Oktober 2019.

INFORMATIKALOGI (8 April 2017). "*Algoritma K-Nearest Neighbor (K-NN)*". Retrieved 21 Oktober 2019. from informatikalogi.com : https://informatikalogi.com/algoritma-k-nn-k-nearest-neighbor/

INFORMATIKALOGI (8 April 2017). "*Algoritma Naive Bayes*". Retrieved 21 Oktober 2019. from informatikalogi.com : https://informatikalogi.com/algoritma-naive-bayes/

NUR ANNISSA TAJMAKAL OHORELLA (10 Juli 2018). "*Support Vector Machine (SVM) dalam R*". Retrieved 21 Oktober 2019. from Medium : https://medium.com/@16611094/support-vector-machine-svm-dalam-r-932c759aedb2