NAMA : ADITYA
NIM : F55119076

Laporan

Pengenalan Pola

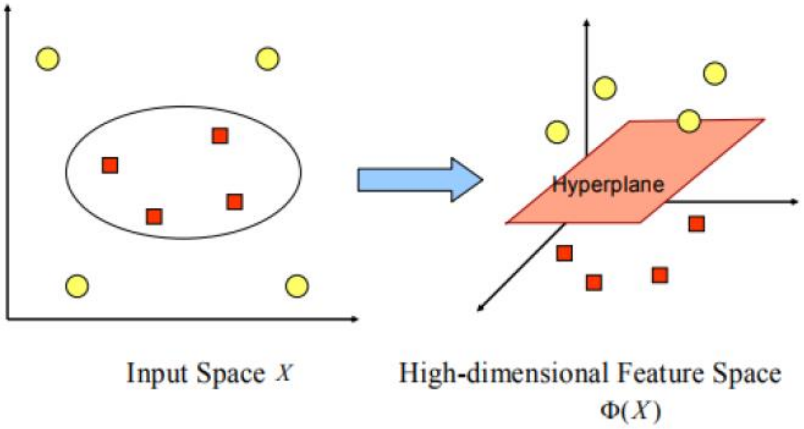
Support Vector Machine Classification with Python

A. Tujuan

1. Dapat mengetahui penggunaa Naive Bayes pada Pengenalan Pola
2. Dapat mengetahui tahapan dalam metode Naive Bayes.
3. Dapat mampu melakukan pemrograman dasar Naive Bayes.

B. Teori Dasar

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.



Berikut ini merupakan beberapa fungsi kernel pada umumnya:

- Kernel Polynomial dengan Variabel Bebas q

$$K(\vec{X}_i, \vec{X}_j) = (\vec{X}_i, \vec{X}_j + 1)^q$$

- Kernel Gaussian atau RBF

$$K(\vec{X}_i, \vec{X}_j) = \exp\left(\frac{\|\vec{X} - \vec{X}_i\|^2}{2\sigma^2}\right)$$

C. Praktikum

langsung saja kita masuk ke penerapan SVM untuk klasifikasi data pasien Penyakit Kanker Payudara. Datanya berupa rata-rata dari karakteristik yang diambil sampelnya dari tubuh pasien yang sedang didiagnosis.

Oke, langkah pertama yang kita lakukan adalah memuat data penyakit kanker payudara ke dalam notebook menggunakan sintaks berikut:

```
#Import scikit-learn dataset library
from sklearn import datasets#Load dataset
cancer = datasets.load_breast_cancer()
```

Kemudian kita eksplor data untuk mengetahui variabel features/independen dan nama target.

```
# print the names of the 13 features
print("Features: ", cancer.feature_names)# print the label type of cancer('malignant' 'benign')
print("Labels: ", cancer.target_names)
```

```
Features:  ['mean radius' 'mean texture' 'mean perimeter' 'mean area'
 'mean smoothness' 'mean compactness' 'mean concavity'
 'mean concave points' 'mean symmetry' 'mean fractal dimension'
 'radius error' 'texture error' 'perimeter error' 'area error'
 'smoothness error' 'compactness error' 'concavity error'
 'concave points error' 'symmetry error' 'fractal dimension error'
 'worst radius' 'worst texture' 'worst perimeter' 'worst area'
 'worst smoothness' 'worst compactness' 'worst concavity'
 'worst concave points' 'worst symmetry' 'worst fractal dimension']
Labels:  ['malignant' 'benign']
```

Setelah itu kita akan cek dimensi datanya menggunakan sintaks berikut:

```
# print data(feature)shape
cancer.data.shape
```

```
Out[3]: (569, 30)
```

Data memiliki 569 baris dan 30 kolom. Kemudian kita akan lihat 5 data pertama dari variabel features.

```
print(cancer.data[0:5])
```

```
[[1.799e+01 1.038e+01 1.228e+02 1.001e+03 1.184e-01 2.776e-01 3.001e-01
  1.471e-01 2.419e-01 7.871e-02 1.095e+00 9.053e-01 8.589e+00 1.534e+02
  6.399e-03 4.904e-02 5.373e-02 1.587e-02 3.003e-02 6.193e-03 2.538e+01
  1.733e+01 1.846e+02 2.019e+03 1.622e-01 6.656e-01 7.119e-01 2.654e-01
  4.601e-01 1.189e-01]
 [2.057e+01 1.777e+01 1.329e+02 1.326e+03 8.474e-02 7.864e-02 8.690e-02
  7.017e-02 1.812e-01 5.667e-02 5.435e-01 7.339e-01 3.398e+00 7.408e+01
  5.225e-03 1.308e-02 1.860e-02 1.340e-02 1.389e-02 3.532e-03 2.499e+01
  2.341e+01 1.588e+02 1.956e+03 1.238e-01 1.866e-01 2.416e-01 1.860e-01
  2.750e-01 8.902e-02]
 [1.969e+01 2.125e+01 1.300e+02 1.203e+03 1.096e-01 1.599e-01 1.974e-01
  1.279e-01 2.069e-01 5.999e-02 7.456e-01 7.869e-01 4.585e+00 9.403e+01
  6.150e-03 4.006e-02 3.832e-02 2.058e-02 2.250e-02 4.571e-03 2.357e+01
  2.553e+01 1.525e+02 1.709e+03 1.444e-01 4.245e-01 4.504e-01 2.430e-01
  3.613e-01 8.758e-02]
 [1.142e+01 2.038e+01 7.758e+01 3.861e+02 1.425e-01 2.839e-01 2.414e-01
  1.052e-01 2.597e-01 9.744e-02 4.956e-01 1.156e+00 3.445e+00 2.723e+01
  9.110e-03 7.458e-02 5.661e-02 1.867e-02 5.963e-02 9.208e-03 1.491e+01
  2.650e+01 9.887e+01 5.677e+02 2.098e-01 8.663e-01 6.869e-01 2.575e-01
  6.638e-01 1.730e-01]
 [2.029e+01 1.434e+01 1.351e+02 1.297e+03 1.003e-01 1.328e-01 1.980e-01
  1.043e-01 1.809e-01 5.883e-02 7.572e-01 7.813e-01 5.438e+00 9.444e+01
  1.149e-02 2.461e-02 5.688e-02 1.885e-02 1.756e-02 5.115e-03 2.254e+01
  1.667e+01 1.522e+02 1.575e+03 1.374e-01 2.050e-01 4.000e-01 1.625e-01
  2.364e-01 7.678e-02]]
```

```
# print the cancer labels (0:malignant, 1:benign)
print(cancer.target)
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 0 1 0 0 1 1 1 1 0 1 0 0 1 1 1 1 0 1 0 0
 1 0 1 0 0 1 1 1 0 0 1 0 0 0 1 1 1 0 1 1 0 0 1 1 1 0 0 1 1 1 1 0 1 1 0 1 1
 1 1 1 1 1 1 0 0 0 1 0 0 1 1 1 0 0 1 0 1 0 0 1 0 0 1 1 0 1 1 0 1 1 1 1 0 1
 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 1 1 0 0 0 1 0
 1 0 1 1 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 1 0 1 0 1 1 0 1 0 0 0 0 1 1 0 0 1 1
 1 0 1 1 1 1 1 0 0 1 1 0 1 1 0 0 1 0 1 1 1 1 0 1 1 1 1 1 0 1 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 0 1 1 0 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1
 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 0 0 0 1 1
 1 1 0 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0
 0 1 0 0 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1
 1 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 0 1 1 1 1 1 0 1 1
 0 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1
 1 1 1 1 1 1 0 1 0 1 1 0 1 1 1 1 1 1 0 0 1 0 1 0 1 1 1 1 1 1 0 1 1 0 1 0 1 0 0
 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 1]
```

Selanjutnya, kita akan membagi dataset menjadi data training dan data testing. Data training yang digunakan adalah sebanyak 75% dan data testing sebanyak 25%.

```
# Import train_test_split function
from sklearn.model_selection import train_test_split# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(cancer.data, cancer.target,
test_size=0.25,random_state=123) # 75% training and 25% test
```

Kemudian, kita akan membuat model yang akan kita gunakan untuk melakukan klasifikasi.

```
#Import svm model
from sklearn import svm#Create a svm Classifier
clf = svm.SVC(kernel='linear') # Linear Kernel#Train the model using the training sets
clf.fit(X_train, y_train)#Predict the response for test dataset
y_pred = clf.predict(X_test)
```

```
In [8]: y_pred

Out[8]: array([1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0,
               1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1,
               0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0,
               1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1,
               0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
               1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1,
               1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0])

In [9]: y_test

Out[9]: array([1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0,
               1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1,
               0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0,
               1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1,
               0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
               1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1,
               1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0])
```

Langkah berikutnya adalah melihat confusion matrix untuk memudahkan dalam mengetahui apakah terdapat kesalahan dalam pengklasifikasian.

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred)
```

```
Out[10]: array([[52,  2],
                [ 0, 89]], dtype=int64)
```

Berdasarkan output, diketahui bahwa terdapat 2 kesalahan dalam pengklasifikasian menggunakan algoritma SVM, yaitu 2 pasien diklasifikasikan dalam pasien pengidap kanker jinak, tetapi dalam keadaan sebenarnya, pasien mengalami kanker ganas.

Kemudian kita akan melihat akurasi dari hasil pengklasifikasian menggunakan algoritma SVM. Akurasi merupakan proporsi jumlah prediksi benar.

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      0.96      0.98        54
           1       0.98      1.00      0.99        89

    accuracy                           0.99       143
   macro avg       0.99      0.98      0.99       143
weighted avg       0.99      0.99      0.99       143
```

Akurasi yang diperoleh adalah sebesar 99%. Hasil ini bisa dikatakan sebagai hasil yang sangat bagus.

Daftar Pustaka

Bhumika M. Jadav, & Vimalkumar B. Vaghela, ―Sentiment Analysis using Support Vector Machine based on Feature Selection and Semantic Analysis‖, International Journal of Computer Applications (0975 – 8887) Volume 146 – No.13, July 2016

Geetika Gautam & Divakar Yadav, ― Sentiment Analysis of Twitter Data Using Machine Learning Approaches and Semantic Analysis‖, Conference: Conference: 7th International Conference on Contemporary Computing, pp. 437-442, At Noida(India), Volume: IEEE Xplore.