

Tugas Mata Kuliah Kecerdasan Buatan

March 19, 2022

Muhammad Nazhim Maulana (1194025)

Definisi Kecerdasan Buatan

Berdasarkan Pengertian nya kecerdasan buatan atau yang lebih dikenal dalam Bahasa Inggris sebagai *Artificial Intelligence*. Secara sederhana, sebuah kecerdasan buatan itu bisa diartikan sebagai sebuah sistem kecerdasan yang manusia tanamkan ke dalam sebuah teknologi dan sistem kecerdasan ini juga bergantung terhadap banyaknya data yang di berikan. Maksudnya disini adalah semakin banyak data yang dimiliki maka akan semakin akurat juga program yang berjalan pada teknologi tersebut.

Sejarah dan Perkembangan Kecerdasan Buatan

Berdasarkan sejarah, istilah *Artificial Intelligence* itu pertama kali ditemukan pada tahun 1956 dan sampai sekarang ini masih populer. Untuk penelitian atau riset awalnya dimulai pada tahun 1950an kemudian muncul keinginan dari Departemen Pertahanan Amerika Serikat untuk melatih komputer agar memiliki kecerdasan layaknya manusia.

Kemudian terus terjadi perkembangan program AI yang dibuat oleh manusia mulai dari tahun 1959 dikeluarkan program yang dapat mengeluarkan teorema dengan menggunakan aksioma-aksioma yang ada. Perkembangan itu terus terjadi hingga sekarang ini dimana sudah banyak dilakukan pengembangan program-program yang dapat membantu kehidupan sehari-hari dari manusia.

Supervised dan Unsupervised Learning

Supervised Learning adalah salah satu jenis dari algoritma yang digunakan untuk mengajar sebuah *Machine Learning*, untuk pembelajarannya ini akan diawasi langsung oleh seorang *Supervisor* atau pengawas. Algoritma ini akan memerlukan sebuah data berlabel untuk membangun model yang memiliki tingkat akurasi yang terus menerus dapat ditingkatkan.

Selanjutnya ada *Unsupervised Learning* yang merupakan kebalikan dari *Supervised Learning*. Jikalau *Supervised Learning* itu memiliki pengawas, maka *Unsupervised Learning* sendiri tidak diawasi. Dengan begitu algoritma ini cenderung lebih bebas dalam proses eksplorasi data karena tiap data yang ada tidak memiliki label sehingga lebih mudah dalam melakukan eksplorasi dan menemukan data yang tersembunyi.

Regresi dan Klasifikasi

Regresi merupakan sebuah proses untuk menemukan korelasi antara variabel *Dependent* dan *Independent*. Regresi ini biasanya sangat membantu untuk melakukan prediksi yang bersifat *Continuous* seperti prediksi Tren Pasar, atau bahkan hingga ke perkiraan cuaca juga dapat digunakan.

Sedangkan klasifikasi merupakan sebuah proses untuk menemukan fungsi yang dapat membantu untuk melakukan pembagian terhadap dataset ke dalam kelas-kelas tertentu berdasarkan dengan parameter-parameter yang berbeda-beda. Biasanya sebuah program komputer akan dilatih dengan mengkategorikan data yang telah diberikan menjadi berbagai kelas yang berbeda-beda.

Dataset

Secara sederhana dataset itu merupakan kumpulan dari data yang berasal dari informasi-informasi yang ada pada masa lampau. Informasi yang diperoleh ini nantinya akan diolah kembali untuk menghasilkan sebuah informasi yang baru. Dataset sendiri bisa diubah (baik ditambahkan, dihapus atau diperbarui baris yang ada di dalamnya)

Training Set dan Testing Set

Setelah membahas mengenai Dataset maka selanjutnya adalah bagian dari dataset yakni *Training Set* dan *Testing Set*. *Training Set* merupakan bagian dari dataset yang dilatih untuk membuat prediksi atau fungsi lain yang nantinya akan ada di dalam sebuah *Machine Learning*, biasanya dalam *Training Set* ini diberikan algoritma untuk melatih sebuah mesin.

Selain *Testing Set*, ada juga yang namanya *Test Set* yang merupakan bagian dari Dataset juga yang di tes untuk melihat tingkat ke akuratan atau bisa dikata untuk melihat performa dari mesin yang sudah dilatih menggunakan *Training Set*.

1. Instalasi Library Scikit Learn

Untuk instalasi dari scikit learn sendiri hasil nya dapat dilihat pada gambar berikut ini:

Figure 1: Hasil Instalasi

```
(base) C:\Users\hpc>conda install scikit-learn
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: D:\Anaconda

added / updated specs:
- scikit-learn

The following packages will be downloaded:

package | build | size
-----|-----|-----
conda-4.12.0 | py38haa95532_0 | 14.5 MB
-----|-----|-----
Total: 14.5 MB

The following packages will be UPDATED:

conda | 4.9.2-py38haa95532_0 --> 4.12.0-py38haa95532_0

Proceed [y]/n/Y

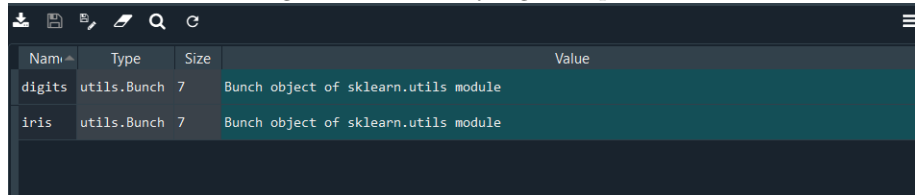
Downloading and Extracting Packages
conda-4.12.0 | 14.5 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(base) C:\Users\hpc>conda list scikit-learn
# packages in environment at D:\Anaconda:
#
# Name          Version      Build Channel
scikit-learn    0.23.2       py38ha7e9c7a_0
```

2. Loading Example Dataset

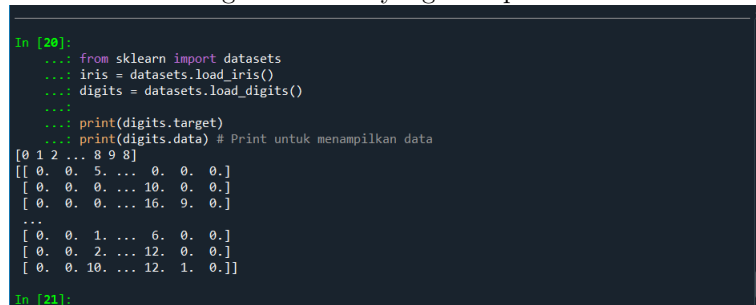
Selanjutnya adalah loading dataset yang telah ada pada dokumentasi scikit learn maka akan diperoleh beberapa hasil seperti pada gambar dibawah ini:

Figure 2: Variable yang di dapatkan



Name	Type	Size	Value
digits	utils.Bunch	7	Bunch object of sklearn.utils module
iris	utils.Bunch	7	Bunch object of sklearn.utils module

Figure 3: Hasil yang di dapatkan



```
In [20]:
....: from sklearn import datasets
....: iris = datasets.load_iris()
....: digits = datasets.load_digits()
....:
....: print(digits.target)
....: print(digits.data) # Print untuk menampilkan data
[0 1 2 ... 8 9 8]
[[ 0.  0.  5. ...  0.  0.  0.]
 [ 0.  0.  0. ... 10.  0.  0.]
 [ 0.  0.  0. ... 16.  9.  0.]
 ...
 [ 0.  0.  1. ...  6.  0.  0.]
 [ 0.  0.  2. ... 12.  0.  0.]
 [ 0.  0. 10. ... 12.  1.  0.]]

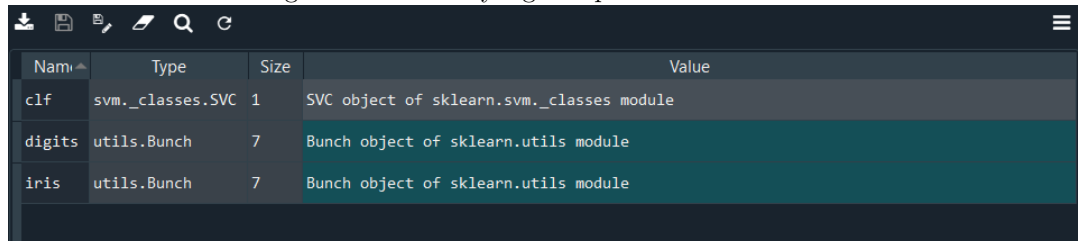
In [21]:
```

Seperti pada gambar diatas dapat dilihat hasil yang di dapatkan untuk loading example dataset.

3. Learning and Predicting

Untuk praktikum yang selanjutnya hasil yang diperoleh adalah sebagai berikut:

Figure 4: Variable yang didapatkan



Name	Type	Size	Value
clf	svm._classes.SVC	1	SVC object of sklearn.svm._classes module
digits	utils.Bunch	7	Bunch object of sklearn.utils module
iris	utils.Bunch	7	Bunch object of sklearn.utils module

Untuk praktikum yang kedua ini nantinya akan diperoleh hasil prediksi berdasarkan sampel dataset yang ada. Hasil yang diperoleh dari consolenya adalah sebagai berikut:

Figure 5: Hasil Akhir pada Console

```
In [21]: from sklearn import svm
...: clf = svm.SVC(gamma=0.001, C=100.)
...: clf.fit(digits.data[:-1], digits.target[:-1])
...: print(clf.predict(digits.data[-1:]))
[8]
```

4. Model Persistence

Untuk praktikum yang selanjutnya hasil yang diperoleh adalah sebagai berikut untuk variabel dan juga hasil akhir pada consolenya:

Figure 6: Variable yang Dihasilkan

Name	Type	Size	Value
clf	svm._classes.SVC	1	SVC object of sklearn.svm._classes module
digits	utils.Bunch	7	Bunch object of sklearn.utils module
iris	utils.Bunch	7	Bunch object of sklearn.utils module
x	Array of float64 (150, 4)		[[5.1 3.5 1.4 0.2] [4.9 3. 1.4 0.2]
y	Array of int32 (150,)		[0 0 0 ... 2 2 2]

Figure 7: Hasil Akhir pada Console

```
In [21]: from sklearn import svm
...: clf = svm.SVC(gamma=0.001, C=100.)
...: clf.fit(digits.data[:-1], digits.target[:-1])
...: print(clf.predict(digits.data[-1:]))
[8]

In [22]: clf = svm.SVC()
...: x, y = datasets.load_iris(return_X_y=True)
...: print(clf.fit(x, y))
SVC()
```

5. Conventions

Untuk praktikum yang selanjutnya hasil yang diperoleh adalah sebagai berikut untuk variabel dan juga hasil akhir pada consolenya:

Figure 8: Variable yang Di dapatkan

Name	Type	Size	Value
clf	svm._classes.SVC	1	SVC object of sklearn.svm._classes module
digits	utils.Bunch	7	Bunch object of sklearn.utils module
iris	utils.Bunch	7	Bunch object of sklearn.utils module
rng	random.mtrand.RandomState	1	RandomState object of numpy.random.mtrand module
transformer	random_projection.GaussianRandomProjection	1	GaussianRandomProjection object of sklearn.random_projection module
x	Array of float64	(150, 4)	[[5.1 3.5 1.4 0.2] [4.9 3. 1.4 0.2]
X	Array of float32	(10, 2000)	[[0.5488135 0.71518934 0.60276335 ... 0.4801078 0.64386404 0.5017731 ...
X_new	Array of float64	(10, 1973)	[[0.3773693 -0.95099789 -0.28294577 ... -0.27... 0 ...
y	Array of int32	(150,)	[0 0 0 ... 2 2 2]

Figure 9: Hasil Akhir pada Console

```

SVC()

In [23]: import numpy as np
        ....: from sklearn import random_projection
        ....:
        ....: rng = np.random.RandomState(0)
        ....: X = rng.rand(10, 2000)
        ....: X = np.array(X, dtype='float32')
        ....: print(X.dtype)
        ....:
        ....: transformer = random_projection.GaussianRandomProjection()
        ....: X_new = transformer.fit_transform(X)
        ....: print(X_new.dtype)
float32
float64

In [24]:

```