

Tugas Kecerdasan Buatan Section 1

Ahmad Agung Tawakkal

1184015

D4 TI 3 B

Maret 2021

1 Chapter 1

1.1 Teori

1. Definisi, Sejarah, dan Perkembangan Kecerdasan Buatan

- Definisi

Kecerdasan Buatan merupakan sebuah sistem yang sangat cerdas, dimana sistem ini dapat digunakan pada sebuah teknologi, misalnya robot yang dapat diajak ngobrol atau robot yang bisa menyelesaikan rubik dengan cepat. Sistem ini memiliki proses berfikir layaknya manusia.

- Sejarah dan Perkembangan

Riset Kecerdasan Buatan awalnya dilakukan pada tahun 1950, namun Kecerdasan Buatan pertama kali dibuat pada tahun 1956. Kecerdasan Buatan bertujuan untuk membangun sebuah komputer yang memiliki penalaran bagaikan manusia. Pada tahun 1960 Amerika ingin mengembangkan dan melatih komputer agar dapat memiliki penalaran bagaikan manusia. Kemudian pada tahun 1970 Proyek DARPA berhasil menyelesaikan pemetaan jalan, pada tahun 2003 DARPA berhasil menghasilkan asisten pribadi yang cerdas. John McCarthy merupakan bapak AI mendirikan sebuah lembaga AI (Stanford Artificial Intelligence Laboratory dan MIT Artificial Intelligence Laboratory). Lembaga inilah pengembangan inovasi pada bidang AI semakin pesat juga pada lembaga ini memiliki bidang Human Skill, Vision, Listening, Reasoning, dan Moment of Limbs.

2. Definisi Supervised Learning, Klasifikasi, Regresi dan Unsupervised learning. Data set, Training set dan Testing set

- Supervised Learning

Supervised Learning mempunyai sebuah variable inputan dan variable output serta menggunakan satu algoritma atau banyak untuk memahami fungsi pemetaan dari inputan ke output. Supervised Learning bertujuan untuk memperkirakan fungsi pemetaan, saat kita memiliki sebuah inputan kita dapat memprediksi hasil output dari inputan tersebut.

- Unsupervised Learning

Unsupervised Learning mempunyai sebuah variable inputan namun tidak memiliki variable output. Unsupervised Learning berfungsi untuk memodelkan struktur dasar agar dapat mempelajari data lebih

jauh atau menyimpulkan fungsi, mendeskripsikan atau menjelaskan data.

- Klasifikasi

Klasifikasi atau penggolongan data untuk menebak suatu kelas tertentu dari sebuah kumpulan data (data set). Pada sekumpulan data ini biasanya memiliki sebuah atribut yang menjadi tujuan klasifikasi. Atribut ini merupakan kolom dari tabel, kemudian diklasifikasi serta juga membutuhkan atribut pendukung agar atribut dapat berpengaruh signifikan terhadap atribut tujuan.

- Regresi

Regresi hampir sama dengan klasifikasi namun yang membedakan adalah regresi memberikan hasil yang tidak terbatas sedangkan klasifikasi memberikan hasil yang terbatas.

1.2 Instalasi

1. Instalasi library scikit dari anaconda

Membuka anaconda prompt kemudian masukan perintah berikut *pip install -U scikit-learn*.

```
(base) C:\Users\ahmad>pip install -U scikit-learn
Collecting scikit-learn
  Downloading https://files.pythonhosted.org/packages/16/33/e0b09b2810e355b667cd3
.8MB)
  100% |#####| 6.8MB 473kB/s
Collecting threadpoolctl>=2.0.0 (from scikit-learn)
  Downloading https://files.pythonhosted.org/packages/f7/12/ec3f2e203afa394a14991
Requirement already satisfied, skipping upgrade: numpy>=1.13.3 in c:\users\ahmad\
Collecting joblib>=0.11 (from scikit-learn)
  Downloading https://files.pythonhosted.org/packages/55/85/70c6602b078bd9e6f3da4
  100% |#####| 307kB 789kB/s
Requirement already satisfied, skipping upgrade: scipy>=0.19.1 in c:\users\ahmad\
Installing collected packages: threadpoolctl, joblib, scikit-learn
  Found existing installation: scikit-learn 0.20.3
    Uninstalling scikit-learn-0.20.3:
      Successfully uninstalled scikit-learn-0.20.3
Successfully installed joblib-1.0.1 scikit-learn-0.24.1 threadpoolctl-2.1.0

(base) C:\Users\ahmad>sn
```

2. Loading an example dataset

```
PS D:\KULIAH\SEMESTER 6\KECERDASAN BUATAN AI\1\program> python
Type "help", "copyright", "credits" or "license" for more information.
>>> from sklearn import datasets
>>> iris = datasets.load_iris()
>>> digits = datasets.load_digits()
>>> print(digits.data)
[[ 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.]]
>>> digits.target
array([0, 1, 2, ..., 8, 9, 8])
>>>
```

- from sklearn import datasets

Mengimport datasets dari library sklearn

- iris = datasets.load_iris()

Membuat sebuah variable iris dengan mengisi fungsi load_iris() dari file datasets yang telah di import

- digits = datasets.load_digits()

Membuat sebuah variable digits dengan mengisi fungsi load_digits() dari file datasets yang telah di import

- print(digits.data)

Menampilkan isi dari variable digits

- digits.target

Menampilkan array angka yang sesuai dengan setiap gambar digit

3. Learning and predicting

```
>>> from sklearn import svm
>>> clf = svm.SVC(gamma=0.001, C=100.)
>>> clf.fit(digits.data[:-1], digits.target[:-1])
SVC(C=100.0, gamma=0.001)
>>> clf.predict(digits.data[-1:])
array([8])
>>>
```

- from sklearn import svm

Mengimport svm dari library sklearn

- clf = svm.SVC(gamma=0.001, C=100.)

Membuat variable dengan nama clf, dan valuenya dari fungsi SVC yang telah kita import (svm)

- clf.fit(digits.data[:-1], digits.target[:-1])
- clf.predict(digits.data[-1:])

4. Model persistence

```
>>> from sklearn import svm
>>> from sklearn import datasets
>>> clf = svm.SVC()
>>> X, y = datasets.load_iris(return_X_y=True)
>>> clf.fit(X, y)
SVC()
>>> import pickle
>>> s = pickle.dumps(clf)
>>> clf2 = pickle.loads(s)
>>> clf2.predict(X[0:1])
array([0])
>>> y[0]
0
```

- from sklearn import svm

Mengimport svm dari library sklearn

- from sklearn import datasets

Mengimport datasets dari library sklearn

- `clf = svm.SVC()`

Membuat variable dengan nama `clf` dengan valuenya dari fungsi `SVC` dari file `svm`

- `X, y = datasets.load_iris(return_X_y=True)`
- `clf.fit(X, y)`
- `import pickle`
- `s = pickle.dumps(clf)`
- `clf2 = pickle.loads(s)`
- `clf2.predict(X[0:1])`
- `y[0]`

5. Conventions

- Type Castign

```
>>> 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')
>>> X.dtype
dtype('float32')
>>> transformer = random_projection.GaussianRandomProjection()
>>> X_new = transformer.fit_transform(X)
>>> X_new.dtype
dtype('float64')
```

- `import numpy as np`

Mengimport numpy dengan menggunakan alias `np`

- `from sklearn import random_projection`

Menimport `random_projection` dari library sklearn

- `rng = np.random.RandomState(0)`
- `X = rng.rand(10, 2000)`
- `X = np.array(X, dtype='float32')`
- `X.dtype`
- `transformer = random_projection.GaussianRandomProjection()`
- `X_new = transformer.fit_transform(X)`
- `X_new.dtype`
- Refitting and updating parameters
 - `from sklearn import datasets`

```

>>> from sklearn import datasets
>>> from sklearn.svm import SVC
>>> iris = datasets.load_iris()
>>> clf = SVC()
>>> clf.fit(iris.data, iris.target)
SVC()
>>> list(clf.predict(iris.data[:3]))
[0, 0, 0]
>>> clf.fit(iris.data, iris.target_names[iris.target])
SVC()
>>> list(clf.predict(iris.data[:3]))
['setosa', 'setosa', 'setosa']
>>>

```

Mengimport datasets dari library sklearn

- from sklearn.svm import SVC

Menimport fungsi SVC dari file svm dan library sklearn

- iris = datasets.load_iris()
- clf = SVC()
- clf.fit(iris.data, iris.target)
- list(clf.predict(iris.data[:3]))
- clf.fit(iris.data, iris.target_names[iris.target])
- list(clf.predict(iris.data[:3]))
-

- Multiclass vs. multilabel fitting

- import numpy as np

Menimport numpy dengan alias np

- from sklearn.datasets import load_iris

Menimport load_iris dari file datasets dan library sklearn

- from sklearn.svm import SVC

Menimport SVC dengan dari file svm pada library sklearn

- X, y = load_iris(return_X_y=True)
- clf = SVC()
- clf.set_params(kernel='linear').fit(X, y)
- clf.predict(X[:5])
- clf.set_params(kernel='rbf').fit(X, y)
- clf.predict(X[:5])

```

>>> import numpy as np
>>> from sklearn.datasets import load_iris
>>> from sklearn.svm import SVC
>>> X, y = load_iris(return_X_y=True)
>>> clf = SVC()
>>> clf.set_params(kernel='linear').fit(X, y)
SVC(kernel='linear')
>>> clf.predict(X[:5])
array([0, 0, 0, 0, 0])
>>> clf.set_params(kernel='rbf').fit(X, y)
SVC()
>>> clf.predict(X[:5])
array([0, 0, 0, 0, 0])
>>>

```

1.3 Error

-