

CERDAS MENGUASAI GIT

CERDAS MENGUASAI GIT

Dalam 24 Jam

Rolly M. Awangga
Informatics Research Center



Kreatif Industri Nusantara

Penulis:

Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

Editor:

M. Yusril Helmi Setyawan

Penyunting:

Syafrial Fachrie Pane

Khaera Tunnisa

Diana Asri Wijayanti

Desain sampul dan Tata letak:

Deza Martha Akbar

Penerbit:

Kreatif Industri Nusantara

Redaksi:

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

Distributor:

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara
apapun tanpa ijin tertulis dari penerbit

*‘Jika Kamu tidak dapat
menahan lelahnya
belajar, Maka kamu harus
sanggup menahan
perihnya Kebodohan.’
Imam Syafi’i*

CONTRIBUTORS

ROLLY MAULANA AWANGGA, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia

CONTENTS IN BRIEF

1 Chapter 1	1
2 Chapter 2	3
3 Chapter 3	5
4 Chapter 4	7
5 Chapter 5	15
6 Chapter 6	17
7 Chapter 7	19

DAFTAR ISI

Foreword	xi
Kata Pengantar	xiii
Acknowledgments	xv
Acronyms	xvii
Glossary	xix
List of Symbols	xxi
Introduction	xxiii
<i>Rolly Maulana Awangga, S.T., M.T.</i>	
1 Chapter 1	1
2 Chapter 2	3
3 Chapter 3	5
4 Chapter 4	7
4.1 Evietania - 1174051	7
	ix

4.1.1	Teori	7
4.1.2	Praktek	9
4.1.3	Penanganan Error	13
5	Chapter 5	15
6	Chapter 6	17
7	Chapter 7	19

FOREWORD

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa

KATA PENGANTAR

Buku ini diciptakan bagi yang awam dengan git sekalipun.

R. M. AWANGGA

Bandung, Jawa Barat
Februari, 2019

ACKNOWLEDGMENTS

Terima kasih atas semua masukan dari para mahasiswa agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk team IRC yang telah fokus untuk belajar dan memahami bagaimana buku ini mendampingi proses Intership.

R. M. A.

ACRONYMS

ACGIH	American Conference of Governmental Industrial Hygienists
AEC	Atomic Energy Commission
OSHA	Occupational Health and Safety Commission
SAMA	Scientific Apparatus Makers Association

GLOSSARY

git	Merupakan manajemen sumber kode yang dibuat oleh linus torvald.
bash	Merupakan bahasa sistem operasi berbasiskan *NIX.
linux	Sistem operasi berbasis sumber kode terbuka yang dibuat oleh Linus Torvald

SYMBOLS

- A Amplitude
- $\&$ Propositional logic symbol
- a Filter Coefficient

- \mathcal{B} Number of Beats

INTRODUCTION

ROLLY MAULANA AWANGGA, S.T., M.T.

Informatics Research Center
Bandung, Jawa Barat, Indonesia

Pada era disruptif saat ini. git merupakan sebuah kebutuhan dalam sebuah organisasi pengembangan perangkat lunak. Buku ini diharapkan bisa menjadi penghantar para programmer, analis, IT Operation dan Project Manajer. Dalam melakukan implementasi git pada diri dan organisasinya.

Rumusnya cuman sebagai contoh aja biar keren[?].

$$ABCDEF\alpha\beta\Gamma\Delta\sum_{def}^{abc} \tag{I.1}$$

BAB 1

CHAPTER 1

BAB 2

CHAPTER 2

BAB 3

CHAPTER 3

BAB 4

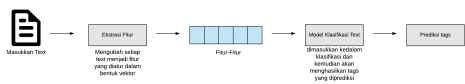
CHAPTER 4

4.1 Evietania - 1174051

4.1.1 Teori

1. Jelaskan apa itu klasifikasi teks, sertakan gambar ilustrasi buatan sendiri.

klasifikasi teks adalah cara untuk memilah-milah teks berdasarkan parameter tertentu baik itu jenis teks atau jenis dari dokumen yang terdapat kumpulan teks didalamnya, sedangkan teks itu sndiri merupakan sekumpulan kata yang dapat dibaca. bisa berupa buku, majalah, rambu-rambu dan lain sebagainya.



Gambar 4.1 contoh klasifikasi teks

2. Jelaskan mengapa klasifikasi bunga tidak bisa menggunakan machine learning, sertakan ilustrasi sendiri.

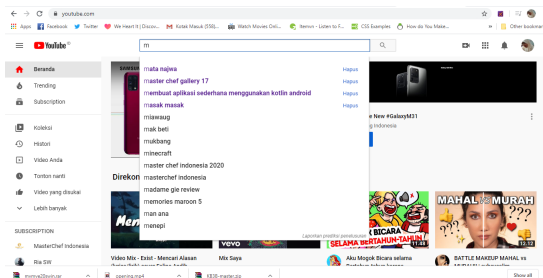
Klasifikasi bunga tidak dapat menggunakan mesin learning dikarenakan jenis-jenis bunga banyak yang mirip bahkan banyak bunga yang serupa tetapi tidak sama. oleh karena itu klasifikasi bunga tidak bisa di gunakan oleh mesin learning dikarenakan jika salah satu inputan ciri-ciri dari suatu bunga di inputkan kemungkinan jawaban dari mesin learning itu tidak tepat contoh dimasukan inputan ciri ciri bunga mawar putih kemudian mesin learning menjawab bahwa itu bunga mawar merah.



Gambar 4.2 contoh klasifikasi bunga

3. Jelaskan bagaimana teknik pembelajaran mesin pada teks pada kata-kata yang digunakan di youtube, jelaskan arti per atribut data csv dan sertakan ilustrasi buatan sendiri.

cara pembelajaran teks yang di gunakan youtube yaitu dengan cara merekam data yang sering di inputkan oleh user pada menu pencarian youtube. sehingga pada saat user akan mencari data yang serupa seringkali youtube menyediakan opsi atau rekomendasi-rekomendasi dari pencaharian. contoh saya menuliskan m maka muncul opsi pilihan master chef dan lainnya yang berawalan m rekomendasi yang muncul merupakan kata-kata yang sering di cari oleh banyak user atau sering di buka oleh user itu sendiri.

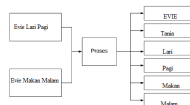


Gambar 4.3 contoh teknik pembelajaran mesin

4. Jelaskan apa yang dimaksud vektorisasi data.

vektorisasi data merupakan pemecahan data menjadi bagian-bagian yang lebih sederhana contoh pada satu paragraf terdiri dari 200 kata kemudian dilakukan vektorisasi dengan cara membagi-bagi kata dalam paragraf tersebut ke dalam kalimat-kalimat yang terpisah kemudian dipecah lagi menjadi data dalam perkata selanjutnya kata-kata tersebut di terjemahkan.

5. Jelaskan apa itu bag of words dengan kata-kata yang sederhana dan ilustrasi sendiri. bag of words merupakan proses penyederhanaan kata-kata yang asalnya tersiri dalam satu kalimat atau satu paragraf di ubah menjadi perkata kemudian kata-kata tersebut di kumpulkan menjadi satu kelompok tanpa ada arti dari kata-kata yang telah di kumpulkan tersebut lalu di hitung frekuensi kemunculan dari kata tersebut.



Gambar 4.4 contoh bag of words

6. Jelaskan apa itu TF-IDF, ilustrasikan dengan gambar sendiri. TF-IDF merupakan metode untuk menghitung bobot dari kata yang sering muncul pada suatu kalimat. metode ini menghitung nilai TF atau Term Frequency dan IDF atau Inverse Document Frequency pada setiap kata pada kalimat yang dijadikan acuan kata pada metode ini sering di sebut token adapun rumus dari metode ini.

	Erie	Malam	Laut	Pagi	Malam	Malam
Doc1	1		1			
Doc2	1		1	1	1	
Doc3		1				1

Gambar 4.5 contoh TF-IDF

4.1.2 Praktek

1. Buat aplikasi sederhana menggunakan pandas, buat data dummy format csv sebanyak 500 baris dan melakukan load ke dataframe pandas. Jelaskan arti setiap baris kode yang dibuat.

```

1 # In[1]: Import library dari pandas
2 import pandas as pan
3 #Membaca file csv menggunakan pandas
4 data_forest = pan.read_csv('praktek1.csv')
5 # In[2]: untuk melihat jumlah dari baris data yang telah di
   import
6 print(len(data_forest))
7 # In[3]: untuk melihat lima baris pertama data yang telah di
   import
8 print(data_forest.head())
9 # In[4]: untuk mengetahui banyak baris dan kolom dari data yang

```

```

10 # telah di import.
11 print(data_forest.shape)

```

2. Dari dataframe tersebut, dipecah menjadi dua dataframe yaitu 450 row pertama dan 50 row sisanya(Harus beda dengan teman sekelas)

```

1 #%%
2 data_450 = data_forest[:450]
3 #%%
4 data_50 = data_forest[450:]

```

3. Praktekkan vektorisasi dan klasifikasi dari data Shakira dengan decision tree. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud dari setiap luaran yang didapatkan.

```

1 npm = 1174039%4
2 print(npm)
3 #%%
4 #Import pandas dan meload file Youtube05-Shakira
5 import pandas as pan
6 data_komentar = pan.read_csv('Youtube05-Shakira.csv')
7
8 #Mengelompokkan spam dan bukan spam ke 2 variabel yang berbeda
9 spam=data_komentar.query('CLASS == 1')
10 no_spam=data_komentar.query('CLASS == 0')
11
12 #Melakukan fungsi bag of word dengan cara menghitung semua kata
   yang ada pada file
13 from sklearn.feature_extraction.text import CountVectorizer
14 vectorizer = CountVectorizer()
15
16 #Membuat variabel dengan perintah untuk memproses kolom CONTENT
   pada dataset, lalu membaca variable yang ada
17 data_vektor = vectorizer.fit_transform(data_komentar['CONTENT'])
18 data_vektor
19
20 #Menampilkan sampel komentar spam yang didapat
21 print(data_komentar['CONTENT'][304])
22
23 #Menampilkan daftar nama kolom
24 dk=vectorizer.get_feature_names()
25
26 #Untuk melakukan randomisasi dari setiap komentar
27 train_test = data_komentar.sample(frac=1)
28 #Memuat data training dan testing dari data yang sudah ada
29 dk_train=train_test[:300]
30 dk_test=train_test[300:]
31
32 #Melakukan training pada data training dan memvektorisasi data
   tersebut
33 dk_train_att = vectorizer.fit_transform(dk_train['CONTENT'])
34 dk_train_att
35
36 #Melakukan Testing pada data testing dan memvektorisasi data
   tersebut

```

[illegible]

4. Cobalah klasifikasikan dari data vektorisasi yang ditentukan di nomor sebelumnya dengan klasifikasi SVM. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

```
In [88]: from sklearn import svm
...: c2fcm = svm.SVC()
...: c2fcm.fit(dk_train_att, dk_train_label)
...: c2fcm.score(dk_test_att, dk_test_label)
C:\Users\jagun\Anaconda2\lib\site-packages\sklearn\svm\base.py:193: FutureWarning: The
default value of gamma will change from 'auto' to 'scale' in version 0.22 to account
better for unequal features. Set gamma explicitly to 'auto' or 'scale' to avoid this
warning.
  "avoid this warning.", FutureWarning)
Out[88]: 0.7289714289714289

In [89]:
```

5. Coba klasifikasikan dari data vektorisasi yang ditentukan di nomor sebelumnya dengan klasifikasi Decision Tree. Tunjukkan kelaurannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

```
In [11]: from sklearn import tree
...: clf = tree.DecisionTreeClassifier()
...: clf.fit(X_train_att, X_train_label)
...: clf.score(X_test_att, X_test_label)
Out[11]: 0.9420571428571428
```

- Plotlah confusion matrix dari praktek modul ini menggunakan matplotlib. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan

```

1  ###
2  #Melakukan confusion matrix
3  from sklearn.metrics import confusion_matrix
4  pred_labels = clftree.predict(dk_test_att)
5  cm = confusion_matrix(dk_test_label, pred_labels)
6  cm

```

```

In [12]: from sklearn.metrics import confusion_matrix
...: pred_labels = clftree.predict(dk_test_att)
...: cm = confusion_matrix(dk_test_label, pred_labels)
...:
Out[12]:
array([[ 4,  0],
       [ 0, 15]), dtype=int)
In [13]:

```

Gambar 4.9 Penggunaan confusion matrix

7. Jalankan program cross validation pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

```

1  from sklearn.model_selection import cross_val_score
2  scores = cross_val_score(clftee, dk_train_att, dk_train_label, cv
    =5)
3  scorerata2=scores.mean()
4  scorersd=scores.std()
5
6  from sklearn.model_selection import cross_val_score
7  scores = cross_val_score(clftee, dk_train_att, dk_train_label,
    cv=5)
8  # show average score and +/- two standard deviations away (
    covering 95
9  #% of scores)
10 print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std
    () * 2))
11
12 scoretree = cross_val_score(clftee, dk_train_att,
    dk_train_label, cv=5)
13 print("Accuracy: %0.2f (+/- %0.2f)" % (scoretree.mean(),
    scoretree.std() * 2))
14
15 scoressvm = cross_val_score(clfsvm, dk_train_att, dk_train_label,
    cv=5)
16 print("Accuracy: %0.2f (+/- %0.2f)" % (scoressvm.mean(),
    scoressvm.std() * 2))

```

```

In [14]: from sklearn.model_selection import cross_val_score
...: scores = cross_val_score(clftee, dk_train_att, dk_train_label, cv=5)
...: scorerata2=scores.mean()
...: scorersd=scores.std()
...:
...: from sklearn.model_selection import cross_val_score
...: scores = cross_val_score(clfsvm, dk_train_att, dk_train_label, cv=5)
...: scoressvm = cross_val_score(clfsvm, dk_train_att, dk_train_label, cv=5)
...: print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
...:
...: print("Accuracy: %0.2f (+/- %0.2f)" % (scoretree.mean(), scoretree.std() * 2))
...:
...: scoressvm = cross_val_score(clfsvm, dk_train_att, dk_train_label, cv=5)
...: print("Accuracy: %0.2f (+/- %0.2f)" % (scoressvm.mean(), scoressvm.std() * 2))
...:
Accuracy: 0.80 (+/- 0.00)
Accuracy: 0.80 (+/- 0.00)
Accuracy: 0.80 (+/- 0.00)
Out[14]:
0.8000000000000001
0.8000000000000001
0.8000000000000001
Warning: The
default value of gamma will change from 'auto' to 'scale' in version 0.16 to account
better for scaled features. We guess explicitly to 'auto' or 'scale' to avoid this
warning.
"avoid this warning": Parameter(gamma)

```

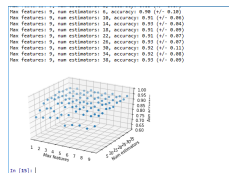
Gambar 4.10 Mengambil tingkat akurasi dari klasifikasi data dan prediksi

8. Buatlah program pengamatan komponen informasi pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan

```

1  ###
2  from sklearn.ensemble import RandomForestClassifier
3  import numpy as np
4  max_features_opts = range(1, 10, 1)
5  n_estimators_opts = range(2, 40, 4)
6  rf_params = np.empty((len(max_features_opts)*len(
7      n_estimators_opts),4), float)
8  i = 0
9  for max_features in max_features_opts:
10     for n_estimators in n_estimators_opts:
11         clf = RandomForestClassifier(max_features=max_features,
12             n_estimators=n_estimators)
13         scores = cross_val_score(clf, dk_train_att,
14             dk_train_label, cv=5)
15         rf_params[i,0] = max_features
16         rf_params[i,1] = n_estimators
17         rf_params[i,2] = scores.mean()
18         rf_params[i,3] = scores.std() * 2
19         i += 1
20     print("Max features: %d, num estimators: %d, accuracy:
21         %0.2f (+/- %0.2f)"
22         % (max_features, n_estimators, scores.mean(), scores.std() * 2)
23         )
24
25 import matplotlib.pyplot as plt
26 from mpl_toolkits.mplot3d import Axes3D
27 from matplotlib import cm
28 fig = plt.figure()
29 fig.clf()
30 ax = fig.gca(projection='3d')
31 x = rf_params[:,0]
32 y = rf_params[:,1]
33 z = rf_params[:,2]
34 ax.scatter(x, y, z)
35 ax.set_zlim(0.6, 1)
36 ax.set_xlabel('Max features')
37 ax.set_ylabel('Num estimators')
38 ax.set_zlabel('Avg accuracy')
39 plt.show()

```



Gambar 4.11 Melakukan regresi data dengan numpy dan mengambil fitur - fitur yang ada

4.1.3 Penanganan Error

4.1.3.1 Sreenshoot Error

1. Error 1

```
FileNotFoundError: [Errno 2] File b'praktek.csv' does not exist: b'prak
```

```
- -
```

Gambar 4.12 error 1

4.1.3.2 Penanganan untuk Error

1. penanganan 1, kesalahan pada nama file yang dituju, dimana cara penanganannya adalah memasukkan nama file dengan benar

```
# In[1]: Import library dari pandas
import pandas as pan
#Membaca file csv menggunakan pandas
data_forest = pan.read_csv('praktek1.csv')
# In[2]: untuk melihat jumlah dari baris data yang t
```

Gambar 4.13 penanganan1

BAB 5

CHAPTER 5

BAB 6

CHAPTER 6

BAB 7

CHAPTER 7
