

classification

March 19, 2019

0.1 Klasifikasi Text - Tugas 3 Pengenalan Pola

0.2 Baskara - 16/398499/PA/17460

0.2.1 Import Library

```
In [1]: import pandas as pd
import numpy as np
import nltk
import math
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
```

0.2.2 Data Preparation (Pembagian Menjadi Training & Test)

```
In [2]: df = pd.read_csv('data_penyakit.csv', names=['tanda_gejala', 'diagnosis_penyakit'])
training = df[df['diagnosis_penyakit'].notnull()]
test = df[df['diagnosis_penyakit'].isnull()]
```

```
In [3]: training.reset_index(inplace=True, drop=True)
training.head()
```

```
Out[3]:
```

	tanda_gejala \
0	menggigil, demam, sakit kepala
1	Kaku kuduk, penurunan kesadaran, muntah proyek...
2	Pipi bengkak, nyeri saat mengunyah, nyeri testis
3	Sakit gigi, gigi sensitif pada makanan dingin ...
4	Hidung tersumbat, bersin, batuk, sakit tenggor...

	diagnosis_penyakit
0	Malaria (bentuk benigna)
1	Meningitis + perdarahan subarachnoid
2	Parotitis
3	Karies dentis
4	Common cold

```
In [4]: test.reset_index(inplace=True, drop=True)
test.head()
```

```
Out[4]:
```

	tanda_gejala	diagnosis_penyakit
0	Mata lengket, mata berair, pandangan sedikit k...	NaN

1	Gusi bengkak, gusi kemerahan, gusi berdarah	NaN
2	Batuk lebih dari tiga minggu, sesak napas atau...	NaN
3	Demam, menggigil, suhu tubuh meningkat, batuk ...	NaN
4	Demam, muntah, diare cair, ampas sedikit seper...	NaN

0.2.3 Tokenization & Stemming Data Training

```
In [5]: factory = StemmerFactory()
        stemmer = factory.create_stemmer()
        tokenizer = nltk.RegexpTokenizer(r'\w+')

        for index, row in training.iterrows():
            # Stemming
            stemmed = stemmer.stem(row[0])
            #Tokenization
            tokens = tokenizer.tokenize(row[0])
            #Case Folding
            words = [w.lower() for w in tokens]
            training.at[index, 'tanda_gejala'] = words
```

```
In [6]: training.head()
```

```
Out[6]:
```

	tanda_gejala \	diagnosis_penyakit
0	[menggigil, demam, sakit, kepala]	Malaria (bentuk benigna)
1	[kaku, kuduk, penurunan, kesadaran, muntah, pr...	Meningitis + perdarahan subarachnoid
2	[pipi, bengkak, nyeri, saat, mengunyah, nyeri,...	Parotitis
3	[sakit, gigi, gigi, sensitif, pada, makanan, d...	Karies dentis
4	[hidung, tersumbat, bersin, batuk, sakit, teng...	Common cold

0.2.4 Membuat Kolom Untuk Setiap Kata

```
In [7]: columnlist = []
        for index, row in training.iterrows():
            columnlist = np.concatenate((columnlist, row[0]))
        columnlist = np.unique(columnlist)

In [8]: for index in range(len(columnlist)):
        training.insert(2, str(columnlist[index]), 0)

In [9]: training.head()
```

```

Out[9]:
          tanda_gejala \
0          [menggigil, demam, sakit, kepala]
1  [kaku, kuduk, penurunan, kesadaran, muntah, pr...
2  [pipi, bengkak, nyeri, saat, mengunyah, nyeri,...
3  [sakit, gigi, gigi, sensitif, pada, makanan, d...
4  [hidung, tersumbat, bersin, batuk, sakit, teng...

          diagnosis_penyakit yang warna wajah vulva volume \
0          Malaria (bentuk benigma)      0      0      0      0      0
1  Meningitis + perdarahan subarachnoid      0      0      0      0      0
2          Parotitis      0      0      0      0      0
3          Karies dentis      0      0      0      0      0
4          Common cold      0      0      0      0      0

          vesikuler vesikul vesikel ... ampas amis amandel alis aksila \
0          0      0      0 ...      0      0      0      0      0
1          0      0      0 ...      0      0      0      0      0
2          0      0      0 ...      0      0      0      0      0
3          0      0      0 ...      0      0      0      0      0
4          0      0      0 ...      0      0      0      0      0

          akibat akan agak ada abdomen
0          0      0      0      0      0
1          0      0      0      0      0
2          0      0      0      0      0
3          0      0      0      0      0
4          0      0      0      0      0

[5 rows x 427 columns]

```

0.2.5 Menghitung Jumlah Frekuensi Setiap Kata

```

In [10]: for index, row in training.iterrows():
          for columnindex in range(len(columnlist)):
              training.at[index, columnlist[columnindex]] = row[0].count(str(columnlist[columnindex]))

In [11]: training.head()

```

```

Out[11]:
          tanda_gejala \
0          [menggigil, demam, sakit, kepala]
1  [kaku, kuduk, penurunan, kesadaran, muntah, pr...
2  [pipi, bengkak, nyeri, saat, mengunyah, nyeri,...
3  [sakit, gigi, gigi, sensitif, pada, makanan, d...
4  [hidung, tersumbat, bersin, batuk, sakit, teng...

          diagnosis_penyakit yang warna wajah vulva volume \
0          Malaria (bentuk benigma)      0      0      0      0      0
1  Meningitis + perdarahan subarachnoid      0      0      0      0      0

```

2	Parotitis	0	0	0	0	0
3	Karies dentis	0	0	0	0	0
4	Common cold	0	0	0	0	0

	vesikuler	vesikul	vesikel	...	ampas	amis	amandel	alis	aksila	\
0	0	0	0	...	0	0	0	0	0	
1	0	0	0	...	0	0	0	0	0	
2	0	0	0	...	0	0	0	0	0	
3	0	0	0	...	0	0	0	0	0	
4	0	0	0	...	0	0	0	0	0	

	akibat	akan	agak	ada	abdomen
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

[5 rows x 427 columns]

0.2.6 Prepare Test Data

```
In [12]: test.insert(2, 'jarak', 0.0)
test.head()
```

```
Out[12]:
```

	tanda_gejala	diagnosis_penyakit	jarak
0	Mata lengket, mata berair, pandangan sedikit k...		NaN
1	Gusi bengkak, gusi kemerahan, gusi berdarah		NaN
2	Batuk lebih dari tiga minggu, sesak napas atau...		NaN
3	Demam, menggigil, suhu tubuh meningkat, batuk ...		NaN
4	Demam, muntah, diare cair, ampas sedikit seper...		NaN

0.2.7 Preprocessing dan Penghitungan Jumlah Frekuensi Setiap Kata

```
In [13]: for index, row in test.iterrows():
# Stemming
stemmed = stemmer.stem(row[0])
#Tokenization
tokens = tokenizer.tokenize(row[0])
#Case Folding
words = [w.lower() for w in tokens]
test.at[index, 'tanda_gejala'] = words
for index in range(len(columnlist)):
test.insert(3, str(columnlist[index]), 0)
for index, row in test.iterrows():
for columnindex in range(len(columnlist)):
test.at[index, columnlist[columnindex]] = row[0].count(str(columnlist[columnindex]))
```

0.2.8 Penghitungan Jarak (Menggunakan Cosine similarity)

```
In [14]: for test_index, test_row in test.iterrows():
        distance = []
        for train_index, train_row in training.iterrows():
            temp_a = 0
            temp_b = 0
            temp_c = 0
            for columnindex in range(len(columnlist)):
                temp_a += test_row[3+columnindex] * train_row[2+columnindex]
                temp_b += test_row[3+columnindex]**2
                temp_c += train_row[2+columnindex]**2
            distance += [((temp_a)/(math.sqrt(temp_b)*math.sqrt(temp_c)))]
        test.at[test_index, 'jarak'] = (np.min(distance))
        test.at[test_index, 'diagnosis_penyakit'] = training.at[np.argmin(distance), 'diagnosis_penyakit']
```

0.2.9 Hasil Prediksi

```
In [15]: test.iloc[:, : 3]
```

```
Out[15]:
```

	tanda_gejala \	
0	[mata, lengket, mata, berair, pandangan, sedikit...]	
1	[gusi, bengkak, gusi, kemerahan, gusi, berdarah]	
2	[batuk, lebih, dari, tiga, minggu, sesak, napa...]	
3	[demam, menggigil, suhu, tubuh, meningkat, bat...]	
4	[demam, muntah, diare, cair, ampas, sedikit, s...]	
5	[nyeri, kolik, daerah, pinggang, malaise, mual...]	
6	[ruam, yang, gatal, terdri, dari, macula, maku...]	
	diagnosis_penyakit jarak	
0	Malaria (bentuk benigma)	0.0
1	Malaria (bentuk benigma)	0.0
2	Malaria (bentuk benigma)	0.0
3	Meningitis + perdarahan subarachnoid	0.0
4	Parotitis	0.0
5	Malaria (bentuk benigma)	0.0
6	Meningitis + perdarahan subarachnoid	0.0