Basic Practice for Regex Libraray

Regular expressions (called REs, or regexes, or regex patterns) are essentially a tiny, highly specialized programming language embedded inside Python and made available through the re module. Using this little language, you specify the rules for the set of possible strings that you want to match; this set might contain English sentences, or e-mail addresses, or TeX commands, or anything you like. You can then ask questions such as “Does this string match the pattern?”, or “Is there a match for the pattern anywhere in this string?”. You can also use REs to modify a string or to split it apart in various ways

In [1]:

```
import re
```

In [2]:

```
text = "there is pain in again and you should be in a spain"
```

In [3]:

```
x = re.findall("ai", text) #Find "ai" in the above text
print(x)
```

```
['ai', 'ai', 'ai']
```

In [4]:

```
x = re.findall("india", text) #Find india in the above text the answer will be empty.
x
```

Out[4]:

```
[]
```

In [5]:

```
x = re.findall("in", text) #Find "in" in the above text
x
```

Out[5]:

```
['in', 'in', 'in', 'in', 'in']
```

In [6]:

```
x = re.search("there", text) #search "there" in the above text
x
```

Out[6]:

```
<re.Match object; span=(0, 5), match='there'>
```

In [7]:

```
text = "The rain in spain"
x = re.search('^The.*Spain$',text )
print(x)
```

```
None
```

In [8]:

```
text = "The rain in Spain"
x = re.search('^The.*Spain$',text )
print(x)
```

```
<re.Match object; span=(0, 17), match='The rain in Spain'>
```

In [9]:

```
text = "The rain in spain"
x = re.split('\s', text) #It show the split in the text
print(x)
```

```
['The', 'rain', 'in', 'spain']
```

In [10]:

```
text = "The rain in spain"
x = re.split('\s', text, 2) #Show two place has split, we can show one place has split,
print(x)
```

```
['The', 'rain', 'in spain']
```

Case study for Naive Bayes

News Categorization using Multinomial Naive Bayes

The objective of this site is to show how to use Multinomial Naive Bayes method to classify news according to some predefined classes.

The News Aggregator Data Set comes from the UCI Machine Learning Repository.

This dataset contains headlines, URLs, and categories for 422,937 news stories collected by a web aggregator between March 10th, 2014 and August 10th, 2014. News categories in this dataset are labelled:

b: business; t: science and technology; e: entertainment; and m: health.

In [11]:

```
import pandas as pd
```

In [12]:

```
df = pd.read_csv("uci-news-aggregator.csv")
df.head(20)
```

Out[12]:

	ID	TITLE	URL	PUBLISHER	CATEGORY	
0	1	Fed official says weak data caused by weather,...	http://www.latimes.com/business/money/la-fi-mo...	Los Angeles Times	b	ddUyU
1	2	Fed's Charles Plosser sees high bar for change...	http://www.livemint.com/Politics/H2EvwJSK2VE6O...	Livemint	b	ddUyU
2	3	US open: Stocks fall after Fed official hints ...	http://www.ifamagazine.com/news/us-open-stocks...	IFA Magazine	b	ddUyU

	ID	TITLE	URL	PUBLISHER	CATEGORY	
3	4	Fed risks falling 'behind the curve', Charles ...	http://www.ifamagazine.com/news/fed-risks-fall...	IFA Magazine	b	ddUyU
4	5	Fed's Plosser: Nasty Weather Has Curbed Job Gr...	http://www.moneynews.com/Economy/federal-reser...	Moneynews	b	ddUyU
5	6	Plosser: Fed May Have to Accelerate Tapering Pace	http://www.nasdaq.com/article/plosser-fed-may-...	NASDAQ	b	ddUyU
6	7	Fed's Plosser: Taper pace may be too slow	http://www.marketwatch.com/story/feds-plosser-...	MarketWatch	b	ddUyU
7	8	Fed's Plosser expects US unemployment to fall ...	http://www.fxstreet.com/news/forex-news/articl...	FXstreet.com	b	ddUyU
8	9	US jobs growth last month hit by weather:Fed P...	http://economictimes.indiatimes.com/news/inter...	Economic Times	b	ddUyU
9	10	ECB unlikely to end sterilisation of SMP purch...	http://www.iii.co.uk/news-opinion/reuters/news...	Interactive Investor	b	dPhGU5
10	11	ECB unlikely to end sterilization of SMP purch...	http://in.reuters.com/article/2014/03/10/us-ec...	Reuters India	b	dPhGU5
11	12	EU's half-baked bank union could work	http://blogs.reuters.com/hugo-dixon/2014/03/10...	Reuters UK *blog*	b	dPhGU5
12	13	Europe reaches crunch point on banking union	http://in.reuters.com/article/2014/03/10/eu-ba...	Reuters	b	dPhGU5
13	14	ECB FOCUS-Stronger euro drowns out ECB's messa...	http://in.reuters.com/article/2014/03/10/ecb-p...	Reuters	b	dPhGU5
14	15	EU aims for deal on tackling failing banks	http://main.omanobserver.om/\?p=63376	Oman Daily Observer	b	dPhGU5

	ID	TITLE	URL	PUBLISHER	CATEGORY	
15	16	Forex - Pound drops to one-month lows against ...	http://www.nasdaq.com/article/forex-pound-drop...	NASDAQ	b	dPhGU!
16	17	Noyer Says Strong Euro Creates Unwarranted Eco...	http://www.sfgate.com/business/bloomberg/artic...	San Francisco Chronicle	b	dPhGU!
17	18	EU Week Ahead March 10-14: Bank Resolution, Tr...	http://blogs.wsj.com/brussels/2014/03/10/eu-we...	Wall Street Journal *blog*	b	dPhGU!
18	19	ECB member Noyer is 'very open to all kinds of...	http://www.ifamagazine.com/news/ecb-member-noy...	IFA Magazine	b	dPhGU!
19	20	Euro Anxieties Wane as Bunds Top Treasuries, S...	http://www.businessweek.com/news/2014-03-10/ge...	Businessweek	b	dPhGU!

In [13]:

```
df.CATEGORY.value_counts() #To show the values in one column which is Category
```

Out[13]:

```
e    152469
b    115971
t    108344
m     45640
Name: CATEGORY, dtype: int64
```

In [14]:

```
x= "Hi, How are you?"
```

In [15]:

```
y = "Hi, how are you?"
```

In [16]:

```
x==y
```

Out[16]:

```
False
```

In [17]:

```
t= "Hi, How are you?"
```

In [18]:

```
x==t
```

Out[18]:

```
True
```

In [19]:

```
#Python is sensitive case
```

To do EDA .... Normalize the text

In [20]:

```
#Will create function

def normalize_text(s):
    s = s.lower() #To convert all letter to small.

    #Remove all punctuations that is not word internal (ex: Hyphen, Apostrophes)
    s = re.sub('\s\w', '',s)
    s = re.sub('\w\s', '',s)

    #just make sure that we did not introduce douple spaces
    s = re.sub('\s+', '',s)

    return s
```

In [21]:

```
df['text'] = [normalize_text(s) for s in df['TITLE']]
df['text']
```

Out[21]:

```
0            fedfficialayseakataausedyeather,houldotlowaper
1              fed'sharleslossereesigharorhangenacefapering
2          uspen:tocksallfteredfficialintstccleratedapering
3                fedisksallin'behindheurve',harleslosserays
4                      fed'slosser:astyeatherasurbedobrowth
                                ...
422419                                          imalesma...
422420                                       imusinessma...
422421                                       salesusines...
422422                                      sickotelleve...
422423                                          imalesma...
Name: text, Length: 422424, dtype: object
```

Feature Extraction

In [22]:

```
from sklearn.feature_extraction.text import CountVectorizer #countvectorizer will conve
```

In [23]:

```
#Pull the function in one variable

vectorizer = CountVectorizer()
```

In [24]:

```
x = vectorizer.fit_transform(df['text'])
x
```

Out[24]:

```
<422424x557016 sparse matrix of type '<class 'numpy.int64'>'
        with 914437 stored elements in Compressed Sparse Row format>
```

In [25]:

```
#422424 is the number of rows # X is independent variable
```

In [26]:

```
# Let us deal with dependent variable (y) which is CATEGORY column
df['CATEGORY']
```

Out[26]:

```
0         b
1         b
2         b
3         b
4         b
         ..
422419    b
422420    b
422421    b
422422    m
422423    b
Name: CATEGORY, Length: 422424, dtype: object
```

In [27]:

```
#We will use label encoder to convert polynomic data

from sklearn.preprocessing import LabelEncoder

#Let us pull this one to one variable

encoder = LabelEncoder()
```

In [28]:

```
y = encoder.fit_transform(df['CATEGORY'])
y
```

Out[28]:

```
array([0, 0, 0, ..., 0, 2, 0])
```

In [29]:

```
#Let us split data into train and test

from sklearn.model_selection import train_test_split

#split into train and test

x_train, x_test, y_train, y_test = train_test_split(x,y, random_state=123)
```

In [30]:

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(316818, 557016)
(105606, 557016)
(316818,)
(105606,)
```

Let us apply Naive bayes model

In [31]:

```
from sklearn.naive_bayes import MultinomialNB

nb = MultinomialNB()
```

In [32]:

```
nb.fit(x_train, y_train)
```

Out[32]:

```
MultinomialNB()
```

In [33]:

```
nb.score(x_test, y_test)
```

Out[33]:

```
0.5967369278260705
```

In [34]:

```
#59% of unit  the model is identified #The accuracy
```

small application function which can predict the category of the news¶

In [35]:

```
def predict_cat(title):
    cat_names = {'b': 'Business', 't': 'Technology', 'e': 'Entertainment', 'm': 'Health
    cod = nb.predict(vectorizer.transform([title]))
    return cat_names[encoder.inverse_transform(cod)[0]]
```

In [36]:

```
print("MIUI 13 introduces Optimized File Storage System, a new system-level method of m
```

```
MIUI 13 introduces Optimized File Storage System, a new system-level method of managing
the way files are stored on devices. The Optimized File Storage System on MIUI 13 reduce
s fragmentation and actively manages stored data.To further boost performance, MIUI 13's
RAM Optimization, brings RAM efficiency. This feature analyzes how apps use memory and d
ivides a single app's RAM usage processes into important and unimportant tasks. Then it
closes all unimportant tasks, allowing apps to use memory only for what's important to y
ou right now.
```

In [37]:

```
print(predict_cat("MIUI 13 introduces Optimized File Storage System, a new system-level
```

```
Entertainment
```

In [38]:

```
print(predict_cat("Carabao Cup final: Liverpool's win against Chelsea could turbocharge
```

```
Business
```

In [39]:

```
print(predict_cat("he entire sports industry is changing, from being a manufacturing in
```

```
Technology
```

In [ ]:

```
Learning from Naive bayes:
    1- Naive Bayes assuming IV are independent of each other that why easy to deal with
    2- easy to deal with multiple dimenstion data etc
```