In [1]:
```
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.recommendation import ALS
from pyspark.sql import Row
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
data = spark.read.format("csv").option("header",True).option("inferSchem
a",True).load("RCdata/rating_final.csv")
data.show()
```

```
+------+-------+------+-----------+--------------+
|userID|placeID|rating|food_rating|service_rating|
+------+-------+------+-----------+--------------+
|     1| 135085|     2|          2|             2|
|     2| 135038|     2|          2|             1|
|     3| 132825|     2|          2|             2|
|     4| 135060|     1|          2|             2|
|     5| 135104|     1|          1|             2|
|     6| 132740|     0|          0|             0|
|     7| 132663|     1|          1|             1|
|     8| 132732|     0|          0|             0|
|     9| 132630|     1|          1|             1|
|    10| 132584|     2|          2|             2|
|    11| 132733|     1|          1|             1|
|    12| 132732|     1|          2|             2|
|    13| 132630|     1|          0|             1|
|    14| 135104|     0|          0|             0|
|    15| 132560|     1|          0|             0|
|    16| 132584|     1|          2|             1|
|    17| 132732|     0|          0|             2|
|    18| 132630|     1|          2|             0|
|    19| 132613|     2|          2|             2|
|    20| 132667|     1|          2|             2|
+------+-------+------+-----------+--------------+
only showing top 20 rows
```

In [4]:
```
(training, test) = data.randomSplit([0.8, 0.2])
als = ALS(maxIter=5, regParam=0.01, implicitPrefs=True,userCol="userID",
 itemCol="placeID", ratingCol="rating",
          coldStartStrategy="drop")
model = als.fit(training)
```

In [7]:
```
predictions = model.transform(test)
evaluator = RegressionEvaluator(metricName="rmse", labelCol="rating",
                                predictionCol="prediction")
#rmse = evaluator.evaluate(predictions)
#print("Root-mean-square error = " + str(rmse))
userRecs = model.recommendForAllUsers(10)
movieRecs = model.recommendForAllItems(10)
userRecs.show(5,False)
```

```
+------+--------------------------------------------------------------------
--------------------------------------------------------------------------
--------------------------------------------------------------------------
-----------+
|userID|recommendations


           |
+------+--------------------------------------------------------------------
--------------------------------------------------------------------------
--------------------------------------------------------------------------
-----------+
|463   |[[132862,0.48100093], [135060,0.36393434], [135032,0.25766575],
[132754,0.18146174], [135051,0.17885107], [135072,0.1247983], [132723,
0.12071887], [135057,0.104022026], [132872,0.102057114], [135058,0.0910
4104]]     |
|833   |[[135038,0.09330819], [132825,0.090056315], [135039,0.0592330
1], [135057,0.057779774], [135079,0.049921643], [135045,0.048617616],
 [132921,0.04774813], [135075,0.035188816], [135058,0.031516954], [1350
59,0.03065281]]|
|496   |[[132560,0.0], [132630,0.0], [132660,0.0], [132740,0.0], [13283
0,0.0], [132870,0.0], [135000,0.0], [135030,0.0], [135040,0.0], [13505
0,0.0]]
             |
|148   |[[135051,0.051854], [132862,0.05014171], [135026,0.03888851],
 [132572,0.038523607], [135060,0.035370983], [135030,0.031751085], [135
072,0.03148463], [135038,0.030215243], [135076,0.028989723], [134976,0.
0251938]]    |
|1088  |[[135075,0.39391476], [135076,0.23548545], [135030,0.18134171],
[135057,0.15644243], [135066,0.15440075], [135041,0.13692331], [132754,
0.1352996], [135047,0.13422821], [132723,0.121487096], [135062,0.116676
53]]      |
+------+--------------------------------------------------------------------
--------------------------------------------------------------------------
--------------------------------------------------------------------------
-----------+
only showing top 5 rows
```

In [2]:
```
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.recommendation import ALS
from pyspark.sql import Row
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
data = spark.read.format("csv").option("header",True).option("inferSchem
a",True).load("ml-20m/ratings.csv")
data.show()
```

```
+------+-------+------+----------+
|userId|movieId|rating| timestamp|
+------+-------+------+----------+
|     1|      2|   3.5|1112486027|
|     1|     29|   3.5|1112484676|
|     1|     32|   3.5|1112484819|
|     1|     47|   3.5|1112484727|
|     1|     50|   3.5|1112484580|
|     1|    112|   3.5|1094785740|
|     1|    151|   4.0|1094785734|
|     1|    223|   4.0|1112485573|
|     1|    253|   4.0|1112484940|
|     1|    260|   4.0|1112484826|
|     1|    293|   4.0|1112484703|
|     1|    296|   4.0|1112484767|
|     1|    318|   4.0|1112484798|
|     1|    337|   3.5|1094785709|
|     1|    367|   3.5|1112485980|
|     1|    541|   4.0|1112484603|
|     1|    589|   3.5|1112485557|
|     1|    593|   3.5|1112484661|
|     1|    653|   3.0|1094785691|
|     1|    919|   3.5|1094785621|
+------+-------+------+----------+
only showing top 20 rows
```

In [3]:
```
(training, test) = data.randomSplit([0.8, 0.2])
als = ALS(maxIter=5, regParam=0.01, implicitPrefs=True,userCol="userId",
 itemCol="movieId", ratingCol="rating",
          coldStartStrategy="drop")
model = als.fit(training)
```

In [4]:
```
predictions = model.transform(test)
evaluator = RegressionEvaluator(metricName="rmse", labelCol="rating",
                                predictionCol="prediction")
#rmse = evaluator.evaluate(predictions)
#print("Root-mean-square error = " + str(rmse))
userRecs = model.recommendForAllUsers(10)
movieRecs = model.recommendForAllItems(10)
userRecs.show(5,False)
```

```
+------+--------------------------------------------------------------------
--------------------------------------------------------------------------
-----------------------------------------------+
|userId|recommendations

                                                |
+------+--------------------------------------------------------------------
--------------------------------------------------------------------------
-----------------------------------------------+
|148   |[[17,0.7760857], [539,0.76959646], [1307,0.71515596], [62,0.698
13776], [597,0.6962341], [1035,0.6932603], [1393,0.6854005], [357,0.673
3391], [11,0.66974914], [708,0.6490281]]       |
|463   |[[590,0.9840783], [150,0.9808516], [457,0.92620337], [454,0.912
4953], [296,0.91238725], [339,0.911694], [356,0.9104003], [380,0.904721
5], [597,0.9004352], [592,0.8948771]]          |
|471   |[[1721,1.1610202], [2028,1.1132988], [1610,1.1045119], [1961,1.
0844686], [1580,1.0815619], [2396,1.0779964], [1307,1.0614403], [2268,
1.0577246], [1270,1.0558075], [3578,1.0534228]] |
|496   |[[1196,0.90157634], [1197,0.8774836], [1270,0.87396485], [1198,
0.87006056], [1097,0.8494378], [1210,0.8274293], [260,0.825099], [919,
0.8202874], [1136,0.81852674], [1214,0.8003226]]|
|833   |[[592,0.9677862], [590,0.9576383], [380,0.9545622], [150,0.9510
537], [457,0.94851166], [480,0.9285179], [349,0.9152355], [165,0.910314
44], [356,0.9014194], [153,0.88558245]]        |
+------+--------------------------------------------------------------------
--------------------------------------------------------------------------
-----------------------------------------------+
only showing top 5 rows
```

In [ ]:
```
import nltk
nltk.download('punkt')
import csv
```

Read a dataset that contains news_headlines for clustering

In [50]:
```
from pyspark.sql import Row
from pyspark.sql import SparkSession
from pyspark.ml.feature import HashingTF, IDF, Tokenizer
spark = SparkSession.builder.getOrCreate()
data = spark.read.format("csv").option("header",True).option("inferSchem
a",True).load("abcnews-date-text.csv")
data.show()
data = data.limit(500000)
```

```
+------------+--------------------+
|publish_date|       headline_text|
+------------+--------------------+
|    20030219|aba decides again...|
|    20030219|act fire witnesse...|
|    20030219|a g calls for inf...|
|    20030219|air nz staff in a...|
|    20030219|air nz strike to ...|
|    20030219|ambitious olsson ...|
|    20030219|antic delighted w...|
|    20030219|aussie qualifier ...|
|    20030219|aust addresses un...|
|    20030219|australia is lock...|
|    20030219|australia to cont...|
|    20030219|barca take record...|
|    20030219|bathhouse plans m...|
|    20030219|big hopes for lau...|
|    20030219|big plan to boost...|
|    20030219|blizzard buries u...|
|    20030219|brigadier dismiss...|
|    20030219|british combat tr...|
|    20030219|bryant leads lake...|
|    20030219|bushfire victims ...|
+------------+--------------------+
only showing top 20 rows
```

In [51]:
```
tokenizer = Tokenizer(inputCol="headline_text", outputCol="words")
wordsData = tokenizer.transform(data)
wordsData.show()
wordsData.count()
```

```
+------------+--------------------+--------------------+
|publish_date|       headline_text|               words|
+------------+--------------------+--------------------+
|    20030219|aba decides again...|[aba, decides, ag...|
|    20030219|act fire witnesse...|[act, fire, witne...|
|    20030219|a g calls for inf...|[a, g, calls, for...|
|    20030219|air nz staff in a...|[air, nz, staff, ...|
|    20030219|air nz strike to ...|[air, nz, strike,...|
|    20030219|ambitious olsson ...|[ambitious, olsso...|
|    20030219|antic delighted w...|[antic, delighted...|
|    20030219|aussie qualifier ...|[aussie, qualifie...|
|    20030219|aust addresses un...|[aust, addresses,...|
|    20030219|australia is lock...|[australia, is, l...|
|    20030219|australia to cont...|[australia, to, c...|
|    20030219|barca take record...|[barca, take, rec...|
|    20030219|bathhouse plans m...|[bathhouse, plans...|
|    20030219|big hopes for lau...|[big, hopes, for,...|
|    20030219|big plan to boost...|[big, plan, to, b...|
|    20030219|blizzard buries u...|[blizzard, buries...|
|    20030219|brigadier dismiss...|[brigadier, dismi...|
|    20030219|british combat tr...|[british, combat,...|
|    20030219|bryant leads lake...|[bryant, leads, l...|
|    20030219|bushfire victims ...|[bushfire, victim...|
+------------+--------------------+--------------------+
only showing top 20 rows
```

Out[51]: 500000

TF-IDF

In [52]:
```
hashingTF = HashingTF(inputCol="words", outputCol="rawFeatures",numFeatu
res=300)
featurizedData = hashingTF.transform(wordsData)
```

In [53]:
```
featurizedData.show()
```

```
+------------+--------------------+--------------------+--------------------+
|publish_date|       headline_text|               words|         rawFeatures|
+------------+--------------------+--------------------+--------------------+
|    20030219|aba decides again...|[aba, decides, ag...|(300,[42,57,122,1...|
|    20030219|act fire witnesse...|[act, fire, witne...|(300,[23,43,72,11...|
|    20030219|a g calls for inf...|[a, g, calls, for...|(300,[46,66,144,1...|
|    20030219|air nz staff in a...|[air, nz, staff, ...|(300,[83,116,117,...|
|    20030219|air nz strike to ...|[air, nz, strike,...|(300,[45,88,109,1...|
|    20030219|ambitious olsson ...|[ambitious, olsso...|(300,[18,110,192,...|
|    20030219|antic delighted w...|[antic, delighted...|(300,[50,81,105,1...|
|    20030219|aussie qualifier ...|[aussie, qualifie...|(300,[0,39,51,110...|
|    20030219|aust addresses un...|[aust, addresses,...|(300,[79,83,111,1...|
|    20030219|australia is lock...|[australia, is, l...|(300,[2,62,100,18...|
|    20030219|australia to cont...|[australia, to, c...|(300,[2,42,88,111...|
|    20030219|barca take record...|[barca, take, rec...|(300,[81,95,138,1...|
|    20030219|bathhouse plans m...|[bathhouse, plans...|(300,[107,139,197...|
|    20030219|big hopes for lau...|[big, hopes, for,...|(300,[44,168,225,...|
|    20030219|big plan to boost...|[big, plan, to, b...|(300,[88,123,204,...|
|    20030219|blizzard buries u...|[blizzard, buries...|(300,[11,105,154,...|
|    20030219|brigadier dismiss...|[brigadier, dismi...|(300,[35,129,195,...|
|    20030219|british combat tr...|[british, combat,...|(300,[35,95,117,1...|
|    20030219|bryant leads lake...|[bryant, leads, l...|(300,[84,88,200,2...|
|    20030219|bushfire victims ...|[bushfire, victim...|(300,[15,88,158,1...|
+------------+--------------------+--------------------+--------------------+
only showing top 20 rows
```

IDF features

In [54]:
```
idf = IDF(inputCol="rawFeatures", outputCol="features")
idfModel = idf.fit(featurizedData)
rescaledData = idfModel.transform(featurizedData)

rescaledData.select("headline_text", "features").show()
rescaledData = rescaledData.select("features")
```

```
+--------------------+--------------------+
|       headline_text|            features|
+--------------------+--------------------+
|aba decides again...|(300,[42,57,122,1...|
|act fire witnesse...|(300,[23,43,72,11...|
|a g calls for inf...|(300,[46,66,144,1...|
|air nz staff in a...|(300,[83,116,117,...|
|air nz strike to ...|(300,[45,88,109,1...|
|ambitious olsson ...|(300,[18,110,192,...|
|antic delighted w...|(300,[50,81,105,1...|
|aussie qualifier ...|(300,[0,39,51,110...|
|aust addresses un...|(300,[79,83,111,1...|
|australia is lock...|(300,[2,62,100,18...|
|australia to cont...|(300,[2,42,88,111...|
|barca take record...|(300,[81,95,138,1...|
|bathhouse plans m...|(300,[107,139,197...|
|big hopes for lau...|(300,[44,168,225,...|
|big plan to boost...|(300,[88,123,204,...|
|blizzard buries u...|(300,[11,105,154,...|
|brigadier dismiss...|(300,[35,129,195,...|
|british combat tr...|(300,[35,95,117,1...|
|bryant leads lake...|(300,[84,88,200,2...|
|bushfire victims ...|(300,[15,88,158,1...|
+--------------------+--------------------+
only showing top 20 rows
```

I only use the headline of news and that maybe the reason why the Error is so huge

In [59]:
```
from pyspark.ml.clustering import KMeans

kmeans = KMeans().setK(300).setSeed(1)
model = kmeans.fit(rescaledData)


wssse = model.computeCost(rescaledData)
print("Within Set Sum of Squared Errors = " + str(wssse))


centers = model.clusterCenters()
```

```
Within Set Sum of Squared Errors = 33112470.9293
```

In [94]:
```
import wikipedia
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

t = wikipedia.search("google", results=30)
print t
plist = []
for p in t:
    print p
    page = wikipedia.page(p, auto_suggest=False)
    plist.append(page)
tokenized_docs_list = []
from nltk.tokenize import word_tokenize
for i in plist:
    tokenized_docs = word_tokenize(i.content.encode('ascii','ignore'))
    tokenized_docs_list.append(tokenized_docs)
```

```
[u'Google', u'Google Search', u'Google+', u'Google Play', u'Google Doc
s, Sheets, and Slides', u'Google Books', u'Google Translate', u'.googl
e', u'Google hacking', u'Google Account', u'Google Chrome', u'Google Ma
ps', u'Gmail', u'Google Glass', u'Google Hangouts', u'Thuppakki', u'G S
uite', u'AdWords', u'Google Classroom', u'Google Doodle', u'Google Eart
h', u'Google Analytics', u'List of Google products', u'Google Drive',
 u'Googleplex', u'Vikas Gupta', u'Google Brain', u'Google Traffic', u'G
oogle Dashboard', u'Motorola']
Google
Google Search
Google+
Google Play
Google Docs, Sheets, and Slides
Google Books
Google Translate
.google
Google hacking
Google Account
Google Chrome
Google Maps
Gmail
Google Glass
Google Hangouts
Thuppakki
G Suite
AdWords
Google Classroom
Google Doodle
Google Earth
Google Analytics
List of Google products
Google Drive
Googleplex
Vikas Gupta
Google Brain
Google Traffic
Google Dashboard
Motorola
```

In [106]:
```
print len(tokenized_docs_list)
```

```
29
```

Remove the stopwords

In [107]:
```
from nltk.corpus import stopwords
nltk.download('stopwords')
print "and" in stopwords.words()
t_nsw_list = []
for p in tokenized_docs_list:
    tokenized_nsw = []
    for i in p:
        if not i in stopwords.words():
            tokenized_nsw.append(i)
    t_nsw_list.append(tokenized_nsw)
print len(t_nsw_list)
```

```
[nltk_data] Downloading package stopwords to /Users/han/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
29
```

Stem the words

In [101]:
```
from nltk.stem.porter import PorterStemmer
from nltk.stem.snowball import SnowballStemmer
from nltk.stem.wordnet import WordNetLemmatizer

porter = PorterStemmer()
#snowball = SnowballStemmer("english")
wordnet = WordNetLemmatizer()
after_stem_list = []
for p in t_nsw_list:
    words = []
    for i in p:
        words.append(porter.stem(i))
    after_stem_list.append(words)
```

[u'unilev', '(', ')', 'dutch-british', u'transnat', u'consum', u'good',
u'compani', u'co-headquart', 'rotterdam', ',', u'netherland', 'london',
',', u'unit', 'kingdom', '.', u'it', u'product', u'includ', 'food',
',', u'beverag', ',', u'clean', u'agent', u'person', u'product', '.',
'It', 'world', "'s", 'largest', u'consum', u'good', u'compani', u'meas
ur', '2012', u'revenu', ',', 'world', "'s", 'largest', u'produc', 'foo
d', u'spread', ',', u'margarin', '.', 'It', u'europ', '7th-most', u'val
uabl', u'compani', '.', u'unilev', 'one', 'oldest', u'multin', u'compan
i', ';', u'product', u'avail', 'around', '190', u'countri', '.', u'unil
ev', u'own', '400', u'brand', ',', u'focus', 'thirteen', u'brand', u'sa
le', 'one', 'billion', u'euro', ':', 'axe/lynx', ',', 'dove', ',', 'om
o', ',', 'becel/flora', ',', 'heartbrand', 'ice', u'cream', ',', 'hellm
ann', "'s", ',', 'knorr', ',', 'lipton', ',', 'lux', ',', 'magnum',
',', 'rama', ',', 'rexona', ',', 'sunsilk', 'surf', '.', 'It', u'dual-
list', u'compani', u'consist', u'unilev', 'n.v.', ',', u'base', 'rotter
dam', ',', u'unilev', 'plc', ',', u'base', 'london', '.', 'the', 'two',
u'compani', u'oper', u'singl', u'busi', ',', 'common', 'board', u'direc
tor', '.', u'unilev', u'organis', 'four', 'main', u'divis', u'food',
',', u'refresh', '(', u'beverag', 'ice', 'cream', ')', ',', 'home', 'c
are', ',', u'person', 'care', '.', 'It', 'research', u'develop', u'faci
l', u'unit', 'kingdom', '(', 'two', ')', ',', u'netherland', ',', 'chin
a', ',', 'india', u'unit', u'state', '.', u'unilev', u'found', '1930',
'merger', 'dutch', u'margarin', u'produc', u'margarin', u'uni', 'briti
sh', u'soapmak', 'lever', u'brother', '.', u'dure', 'second', 'half',
'20th', u'centuri', u'compani', u'increasingli', u'diversifi', 'make
r', u'product', 'made', u'oil', u'fat', ',', u'expand', u'oper', u'worl
dwid', '.', 'It', 'made', u'numer', u'corpor', u'acquisit', ',', u'incl
ud', 'lipton', '(', '1971', ')', ',', u'brook', 'bond', '(', '1984',
')', ',', u'chesebrough-pond', '(', '1987', ')', ',', 'best', u'food',
'(', '2000', ')', ',', 'ben', '&', u'jerri', "'s", '(', '2000', ')',
',', u'alberto-culv', '(', '2010', ')', ',', 'dollar', 'shave', 'clu
b', '(', '2016', ')', '.', u'unilev', u'divest', u'special', u'chemic',
u'busi', 'ici', '1997', '.', 'In', '2015', ',', 'leadership', 'paul',
'polman', ',', u'compani', u'gradual', u'shift', u'focu', u'toward',
'health', u'beauti', u'brand', 'away', 'food', u'brand', u'show', 'slo
w', 'growth', '.', u'unilev', 'n.v.', u'primari', u'list', 'euronext',
'amsterdam', u'constitu', 'aex', 'index', '.', u'unilev', 'plc', u'pri
mari', u'list', 'london', 'stock', u'exchang', u'constitu', 'ftse', '10
0', 'index', '.', 'the', u'compani', 'lux', 'euro', 'stoxx', '50', 'sto
ck', 'market', 'index', '.', '==', u'histori', '==', '===', u'1870s191
0', '===', 'In', '1872', ',', 'antoon', u'jurgen', ',', u'found', 'firs
t', u'margarin', u'factori', 'world', u'oss', ',', u'netherland', '.',
'then', ',', '1888', ',', 'samuel', 'bergh', ',', u'oss', ',', u'ope
n', u'margarin', u'factori', 'kleve', '.', 'these', 'two', u'compani',
u'merg', '1927', 'form', u'margarin', u'uni', '.', '===', u'1910s192
0', '===', 'the', u'initi', u'harvest', 'palm', 'oil', 'british', 'wes
t', 'africa', ',', u'news', u'report', 'seen', 'back', 'england', u'sho
w', u'worker', 'abroad', u'favour', u'condit', '.', 'In', '1911', ',',
u'compani', u'receiv', u'concess', '750,000', u'hectar', 'forest', 'be
lgian', 'congo', ',', u'mostli', 'south', 'bandundu', ',', 'system',
u'forc', 'labour', u'oper', '.', '===', u'1920s1930', '===', 'In', '19
22', 'lever', u'brother', u'acquir', 'mac', u'fisheri', ',', 'owner',
'T.', 'wall', '&', u'son', '.', 'In', u'septemb', '1929', ',', u'unile
v', u'form', 'merger', u'oper', 'dutch', u'margarin', u'uni', 'britis
h', u'soapmak', 'lever', u'brother', ',', 'name', u'result', u'compan
i', 'portmanteau', 'name', u'compani', '.', '===', u'1930s1940', '===',
'In', u'1930', u'busi', 'grew', 'new', u'ventur', u'launch', 'africa',

```
 'latin', 'america', '.', 'the', 'nazi', u'occup', u'europ', 'second',
 'world', 'war', 'meant', u'unilev', u'unabl', 'reinvest', u'capit',
 u'europ', ',', 'instead', u'acquir', 'new', u'busi', 'UK', 'US', '.',
 'In', '1943', u'acquir', 'T.', 'J.', 'lipton', ',', u'major', 'stake',
u'frost', u'food', '(', 'owner', u'bird', 'eye', 'brand', ')', u'batche
lor', u'pea', ',', 'one', 'largest', u'veget', u'canner', 'UK', '.', 'I
n', '1944', ',', u'pepsod', u'acquir', '.', 'after', '1945', u'unilev',
"'s", u'success', 'US', u'busi', '(', 'lever', u'brother', 't.j.', 'lip
ton', ')', 'began', u'declin', '.', 'As', 'result', ',', u'unilev', 'be
gan', u'oper', '``', u'hand', "''", u'polici', u'toward', u'subsidiar
i', ',', 'left', 'american', u'manag', u'devic', '.', '===', u'1950s196
0', '===', 'sunsilk', 'first', u'launch', 'UK', '1954', '.', 'dove', 'f
irst', u'launch', 'US', '1957', '.', u'unilev', 'took', 'full', 'owners
hip', u'frost', u'food', '1957', ',', u'renam', u'bird', 'eye', '.', 't
he', u'us-bas', 'good', 'humor', 'ice', 'cream', u'busi', u'acquir', '1
961', '.', 'By', u'mid-1960', u'laundri', 'soap', u'edibl', u'fat', 'st
ill', u'contribut', 'around', 'half', u'unilev', "'s", u'corpor', u'pro
fit', '.', u'howev', 'stagnant', 'market', 'yellow', u'fat', u'increa
s', u'competit', u'deterg', u'soap', 'procter', '&', u'gambl', u'forc',
u'unilev', u'diversifi', '.', 'In', '1971', ',', u'unilev', u'acquir',
 u'british-bas', 'lipton', 'ltd', u'alli', u'supplier', '.', 'In', '197
8', ',', u'nation', 'starch', u'acquir', '$', '487', 'million', ',',
 u'mark', 'largest', 'ever', u'foreign-acquisit', 'US', u'compani', 'po
int', '.', '===', u'1970s1980', '===', 'By', u'1970', ',', u'acquisit',
',', u'unilev', u'gain', '30', 'cent', 'western', 'european', 'ice', 'c
ream', 'market', '.', 'In', '1982', u'unilev', u'manag', u'decid', u're
posit', u'unwieldi', u'conglomer', u'concentr', 'fmcg', u'compani',
 '.', 'In', '1984', u'unilev', u'acquir', u'brook', 'bond', '(', 'make
r', 'PG', u'tip', 'tea', ')', '390', 'million', u'compani', "'s", 'firs
t', u'success', u'hostil', u'takeov', '.', 'In', '1986', u'unilev', u's
trengthen', u'posit', 'world', 'skin', 'market', u'acquir', u'chesebrou
gh-pond', '(', u'merg', 'chesebrough', u'manufactur', 'pond', "'s", u'c
ream', ')', ',', 'maker', 'rag', ',', 'pond', "'s", ',', 'aqua-net',
 ',', 'cutex', ',', u'vaselin', u'anoth', u'hostil', u'takeov', '.', 'I
n', '1989', ',', u'unilev', 'bought', 'calvin', 'klein', u'cosmet',
 ',', 'faberg', ',', 'elizabeth', 'arden', ',', 'latter', 'later', 'sol
d', '(', '2000', ')', 'ffi', u'fragranc', '.', '===', u'1990', '===',
 'In', '1993', u'unilev', u'acquir', u'breyer', 'kraft', ',', 'made',
 u'compani', 'largest', 'ice', 'cream', u'manufactur', u'unit', u'stat
e', '.', 'In', '1996', u'unilev', u'merg', 'elida', u'gibb', 'lever',
 u'brother', 'UK', u'oper', '.', 'It', u'purchas', u'helen', u'curti',
 ',', u'significantli', u'expand', u'presenc', u'unit', u'state', 'sham
poo', u'deodor', 'market', '.', 'the', u'purchas', 'brought', u'unile
v', u'suav', u'finess', u'hair-car', 'product', u'brand', u'degre', u'd
eodor', 'brand', '.', 'In', '1997', u'unilev', 'sold', u'special', u'ch
emic', u'divis', ',', u'includ', u'nation', 'starch', '&', u'chemic',
 ',', 'quest', ',', 'unichema', 'crosfield', u'imperi', u'chemic', u'in
dustri', '4.9', 'billion', '.', u'unilev', u'establish', u'sustain',
 u'agricultur', u'programm', '1998', '.', '===', u'2000', '===', 'In',
 'april', '2000', u'unilev', 'bought', 'ben', '&', u'jerri', "'s", 'sli
m', 'fast', '1.63', 'billion', '.', 'later', 'year', ',', u'compani',
 u'acquir', 'best', u'food', '13.4', 'billion', '.', 'the', u'bestfoo
d', u'acquisit', u'increas', u'unilev', "'s", 'scale', u'food', 'americ
a', ',', u'ad', u'brand', 'knorr', 'hellmann', "'s", 'portfolio', '.',
 'the', u'transact', 'second', 'largest', 'cash', u'acquisit', 'world',
u'busi', u'histori', '.', 'In', u'exchang', 'european', u'regulatori',
 u'approv', 'deal', ',', u'unilev', u'divest', 'well-known', u'brand',
```

```
 'oxo', ',', 'royco', u'batchelor', '.', 'the', 'year', u'compani', 'bo
ught', u'worldwid', 'mustard', u'product', 'firm', u'maill', '.', u'mai
ll', 'three', u'boutiqu', u'europ', ',', 'sell', 'mustard', 'pump', u't
radit', u'maill', 'fashion', '.', u'pari', ',', 'dijon', ',', u'franc',
'london', ',', 'UK', '.', 'the', u'merg', 'best', u'food', u'unilev',
 u'approv', u'isra', 'anti', 'trust', u'agenc', '.', 'In', '2001', u'un
ilev', 'split', 'two', u'divis', ':', 'one', u'food', 'one', 'home',
 u'person', 'care', '.', 'In', 'UK', u'merg', 'lever', u'brother', 'eli
da', u'faberg', u'busi', 'lever', u'faberg', u'januari', '2001', '.',
 'In', u'septemb', '2002', ',', u'compani', 'sold', u'specialti', u'oi
l', u'fat', u'divis', ',', u'loder', 'croklaan', ',', 'rm814', 'millio
n', '(', '218.5', 'million', ')', 'ioi', u'corpor', ',', 'kuala', 'lump
ur', ',', u'malaysia-bas', 'oil', 'palm', u'compani', '.', 'As', 'par
t', 'deal', ',', u'loder', 'croklaan', 'brand', u'maintain', '.', 'als
o', '2002', u'unilev', 'sold', 'mazola', ',', 'argo', '&', u'kingsfor
d', ',', 'karo', ',', 'golden', u'griddl', ',', 'henri', "'s", u'bran
d', ',', 'along', u'sever', 'canadian', u'brand', ',', 'ach', 'food',
 u'compani', ',', 'american', u'subsidiari', u'associ', 'british', u'fo
od', '.', 'In', '2004', u'unilev', 'sold', 'share', 'rushdi', 'food',
 u'industri', 'bashir', u'famili', u'start', u'use', u'barack', 'bran
d', 'name', '.', 'As', '2014', 'roshadi', 'food', u'industri', 'one',
 'three', 'largest', 'tahini', u'produc', 'israel', 'one', 'largest',
 u'produc', 'tahini', u'worldwid', '.', 'In', 'may', '2007', u'unilev',
u'becam', 'first', u'large-scal', u'compani', 'commit', u'sourc', 'te
a', u'sustain', 'manner', ',', u'employ', 'rainforest', u'allianc',
 ',', u'intern', u'environment', 'ngo', ',', u'certifi', 'tea', u'esta
t', 'east', 'africa', ',', 'well', u'third-parti', u'supplier', 'afric
a', u'part', 'world', '.', 'It', u'declar', 'aim', 'lipton', 'yellow',
 'label', 'PG', u'tip', 'tea', u'bag', 'sold', 'western', u'europ', u'c
ertifi', '2010', ',', u'follow', 'lipton', 'tea', u'bag', u'global', '2
015', '.', 'In', u'septemb', '2009', u'unilev', u'agre', u'acquir', u'p
erson', u'busi', 'sara', 'lee', u'corpor', ',', u'includ', u'brand', 'r
adox', ',', u'badeda', u'duschda', ',', u'strengthen', u'categori', 'le
adership', 'skin', u'cleans', u'deodor', '.', 'the', 'sara', 'lee', u'a
cquisit', u'complet', '6', u'decemb', '2010', '.', '===', '20102014',
 '===', 'On', '9', 'august', '2010', u'unilev', u'sign', 'asset', u'pur
chas', 'agreement', 'norwegian', u'dairi', 'group', 'tine', ',', u'acqu
ir', u'activ', u'diplom-i', 'denmark', '.', 'On', '24', u'septemb', '20
10', u'unilev', u'announc', u'enter', u'definit', 'agreement', 'sell',
 u'consum', 'tomato', u'product', u'busi', 'brazil', u'cargil', '.', 'O
n', '27', u'septemb', '2010', u'unilev', u'purchas', u'alberto-culv',
 ',', 'maker', u'person', 'household', u'product', u'includ', u'simpl',
',', 'vo5', ',', u'nexxu', ',', 'tresemm', ',', u'mr', '.', 'dash',
 ',', 'US', '$', '3.7', 'billion', '.', 'On', '28', u'septemb', '2010',
u'unilev', 'evga', u'announc', u'sign', 'agreement', u'unilev', 'woul
d', u'acquir', 'evga', "'s", 'ice', 'cream', u'brand', '(', 'amongst',
 u'other', ',', 'scandal', ',', u'variet', 'karabola', ')', u'distribu
t', 'network', u'greec', ',', u'undisclos', 'amount', '.', 'In', u'febr
uari', '2011', u'unilev', u'announc', 'switch', '100', '%', u'cage-fr
e', u'egg', u'product', u'produc', u'worldwid', '.', 'In', 'march', '20
11', u'announc', u'unilev', u'enter', u'bind', 'agreement', 'sell', 'sa
nex', 'brand', u'colgate-palmol', '672', 'million', ',', u'unilev', 'wo
uld', u'acquir', u'colgate-palmol', "'s", u'laundri', u'deterg', u'bran
d', 'colombia', '(', 'fab', ',', u'lavomat', 'vel', ')', 'US', '$', '21
5', 'million', '.', 'In', 'april', '2011', u'unilev', u'fine', '104',
 'million', 'european', u'commiss', u'establish', u'price-fix', 'carte
l', u'europ', 'along', 'P', '&', 'G', ',', u'fine', '211.2', 'million',
```

',', 'henkel', '(', u'fine', ')', '.', 'though', 'fine', 'set', 'highe
r', 'first', ',', u'discount', '10', '%', u'unilev', 'P', '&', 'G', u'a
dmit', u'run', 'cartel', '.', 'As', u'provid', 'tip-off', u'lead', u'in
vestig', ',', 'henkel', u'fine', '.', 'On', '24', 'august', '2011', u'a
nnounc', u'unilev', u'agre', 'sell', 'alberto', 'vo5', 'brand', u'uni
t', u'state', 'puerto', 'rico', ',', 'rave', 'brand', u'global', ',',
 'brynwood', u'partner', 'VI', 'l.p.', 'On', '14', u'octob', '2011',
 u'announc', u'unilev', u'agre', u'acquir', '82', '%', u'russia-bas',
 u'beauti', u'compani', 'kalina', '.', 'On', '27', u'decemb', '2012',
 u'announc', u'unilev', 'would', 'phase', 'use', u'microplast', 'form',
u'microbead', u'person', u'product', '2015', '.', 'In', u'januari', '20
13', ',', u'unilev', u'agre', 'sell', u'skippi', 'peanut', 'butter', 'b
rand', ',', u'togeth', u'relat', u'manufactur', u'facil', u'littl', 'ro
ck', ',', u'arkansa', ',', u'unit', u'state', 'weifang', ',', 'shandon
g', ',', 'china', ',', 'hormel', u'food', u'approxim', '$', '700', 'mil
lion', '(', '433', 'million', ',', u'approxim', '540', 'million', ')',
 'cash', '.', 'In', u'juli', '2013', u'unilev', u'increas', 'stake', 'i
ndian', 'unit', ',', 'hindustan', u'unilev', ',', '67', '%', 'around',
 '2.45', 'billion', '.', 'On', '12', 'august', '2013', u'unilev', u'ann
ounc', u'sign', 'agreement', u'wish-bon', 'western', u'dress', u'bran
d', u'pinnacl', u'food', 'inc.', 'total', 'cash', u'consider', u'approx
im', 'US', '$', '580', 'million', ',', 'subject', u'regulatori', u'appr
ov', '.', 'On', '6', u'septemb', '2013', u'unilev', u'enter', u'defini
t', 'agreement', u'acquir', 'premium', 'australian', 'tea', 'brand', 'T
2', '.', 'On', '21', u'februari', '2014', u'unilev', u'sign', u'defini
t', 'agreement', 'meat', u'snack', u'busi', ',', u'includ', 'peperami',
'(', 'uk/ireland', ')', 'bifi', '(', u'continent', u'europ', ')', 'jac
k', 'link', "'s", ',', u'undisclos', 'amount', '.', 'In', 'march', '201
4', u'unilev', u'agre', u'acquir', u'major', 'stake', u'china-bas', 'wa
ter', u'purif', u'compani', 'qinyuan', ',', u'make', 'water', u'purif
i', ',', u'drink', 'water', u'equip', 'water', 'treatment', u'membran',
',', u'undisclos', 'price', '.', 'On', '22', 'may', '2014', u'compani',
u'announc', 'sold', 'north', 'america', 'pasta', u'sauc', u'busi', u'in
clud', 'rag', 'bertolli', u'brand', u'japanes', u'compani', 'mizkan',
 'deal', 'worth', '$', '2.15', 'billion', '.', 'On', '10', u'juli', '20
14', '.', u'unilev', u'announc', 'sold', 'slim-fast', 'brand', u'kain
o', u'capit', ',', 'yet', u'retain', u'minor', 'stake', u'busi', '.',
 'On', '2', u'decemb', '2014', ',', u'unilev', u'announc', u'acquir',
 'talenti', 'gelato', '&', 'sorbetto', ':', u'minneapolis-bas', 'talent
i', ',', u'found', '2003', ',', 'grown', u'best-sel', u'packag', 'gelat
o', u'unit', u'state', '.', 'On', '22', u'decemb', '2014', ',', u'unile
v', u'announc', u'purchas', 'camay', 'brand', u'global', 'zest', 'bran
d', u'outsid', 'north', 'america', 'caribbean', 'procter', '&', u'gamb
l', '.', '====', 'hampton', 'creek', 'lawsuit', '====', 'In', u'novem
b', '2014', ',', u'unilev', 'subject', 'media', 'backlash', 'due', 'law
suit', 'rival', 'hampton', 'creek', '.', 'In', 'suit', ',', u'unilev',
 u'reveal', 'hampton', 'creek', '``', u'seiz', 'market', 'share', "''",
u'loss', u'caus', u'unilev', '``', u'irrepar', 'harm', '.', "''", u'uni
lev', u'use', 'standard', u'ident', u'regul', u'claim', 'hampton', 'cre
ek', "'s", '``', 'just', 'mayo', "''", u'product', u'fals', u'adverti
s', "n't", 'contain', u'egg', '.', 'the', 'washington', 'post', u'headl
in', 'suit', 'read', ',', '``', 'big', 'food', "'s", 'weird', 'war', 'o
ver', 'the', u'mean', u'mayonnais', '.', "''", 'the', u'lo', u'angel',
 u'time', 'began', u'stori', '``', 'big', 'tobacco', ',', 'big', 'oil',
',', 'big', 'mayo', '?', "''", 'A', 'wall', 'street', 'journal', 'write
r', u'describ', '``', 'giant', u'corpor', u'gener', 'huge', u'quantit
i', 'free', u'advertis', 'brand', u'equiti', u'tini', 'rival', u'su',

```
 '.', "'", 'eat', 'drink', u'polit', u'headlin', u'controversi', '``',
u'unilev', "'s", u'bulli', u'backfir', ',', u'boost', 'hampton', 'cree
k', "'", '.', u'neg', 'media', u'coverag', 'big', 'mayo', 'lawsuit',
 u'goe', 'viral', 'case', u'studi', 'PR', 'blunder', "'", '.', '===',
 '2015present', '===', 'In', 'march', '2015', ',', u'unilev', u'confir
m', u'reach', 'agreement', u'acquir', 'ren', u'skincar', ',', 'britis
h', u'nich', u'skincar', 'brand', '.', u'thi', u'follow', 'may', '201
5', u'acquisit', u'prestig', u'skincar', 'brand', 'kate', u'somervil',
 u'skincar', 'llc', '.', 'In', u'juli', '2015', ',', u'compani', u'sepa
r', u'spread', u'busi', ',', u'includ', 'flora', 'I', 'Ca', "n't", u'be
liev', 'It', "'s", 'not', 'butter', '!', u'brand', ',', u'standalon',
 u'entiti', u'name', u'unilev', u'bake', ',', u'cook', u'spread', '.',
 'the', u'separ', 'first', u'announc', u'decemb', '2014', 'made', u'res
pons', u'declin', u'worldwid', u'sale', 'product', u'categori', '.', 'I
n', u'octob', '2015', ',', u'unilev', u'agre', u'acquir', 'italian', 'p
remium', 'ice', 'cream', 'maker', 'grom', u'undisclos', 'fee', '.', 'I
n', u'juli', '2016', ',', u'unilev', 'bought', 'US', 'start-up', 'dolla
r', 'shave', 'club', u'report', '$', '1bn', '(', '764m', ')', 'cash',
 u'compet', 'male', u'groom', 'market', '.', 'In', u'septemb', u'unile
v', u'acquir', 'seventh', u'gener', 'inc.', '$', '700', 'million', '.',
'On', u'februari', '17', ',', '2017', ',', u'significantli', 'smaller',
'kraft', 'heinz', 'made', '$', '143', 'billion', 'bid', 'food', u'consu
m', u'product', 'giant', u'unilev', 'the', 'deal', u'declin', u'unile
v', u'abandon', u'februari', '19', 'UK', 'prime', u'minist', 'theresa',
'may', u'order', u'scrutini', 'deal', '.', '==', u'oper', '==', u'unile
v', u'organis', 'four', 'main', u'divis', ':', u'person', 'care', '(',
 u'product', 'skin', 'hair', u'product', ',', u'deodor', 'oral', u'prod
uct', ')', ';', u'food', '(', u'product', u'soup', ',', u'bouillon',
 ',', u'sauc', ',', u'snack', ',', u'mayonnais', ',', 'salad', u'dres
s', ',', u'margarin', u'spread', ')', ';', u'refresh', '(', u'product',
'ice', 'cream', ',', u'tea-bas', u'beverag', ',', u'weight-manag', u'pr
oduct', u'nutrit', u'enhanc', u'stapl', 'sold', u'develop', u'market',
 ')', ';', 'home', 'care', '(', u'product', 'home', u'product', u'inclu
d', u'powder', ',', u'liquid', u'capsul', ',', 'soap', u'bar', u'clea
n', u'product', ')', '.', 'In', u'financi', 'year', u'end', '31', u'dec
emb', '2013', ',', u'unilev', 'total', u'turnov', '49.797', 'billion',
 '36', '%', u'person', 'care', ',', '27', '%', u'food', ',', '19', '%',
u'refresh', '18', '%', 'home', 'care', '.', u'unilev', u'invest', 'tota
l', '1.04', 'billion', 'research', u'develop', '2013', '.', u'unilev',
 'one', 'largest', 'media', u'buyer', 'world', ',', u'invest', 'aroun
d', '6', 'billion', '(', 'US', '$', '8', 'billion', ')', u'advertis',
 u'promot', '2010', '.', u'unilev', "'s", 'largest', u'intern', u'compe
titor', 'nestl', 'procter', '&', u'gambl', '.', 'It', u'face', u'compet
it', 'local', u'market', u'specif', 'product', u'rang', u'numer', u'com
pani', ',', u'includ', 'beiersdorf', ',', 'conagra', ',', u'danon',
 ',', 'henkel', ',', u'mar', ',', 'pepsico', ',', 'reckitt', u'bencki
s', 'S.', 'C.', 'johnson', '&', 'son', '.', u'unilev', u'fine', 'autori
t', u'concurr', u'franc', '2016', u'price-fix', u'person', u'hygien',
 u'product', '.', '==', u'product', '==', u'unilev', "'s", u'product',
 u'includ', u'food', ',', u'beverag', ',', u'clean', u'agent', u'perso
n', u'product', '.', 'the', u'compani', u'own', '400', u'brand', ',',
 u'organis', 'four', 'main', u'categori', '-', u'food', ',', u'refres
h', ',', 'home', 'care', ',', u'person', 'care', '.', u'unilev', "'s",
 'current', u'largest-sel', u'brand', u'includ', ':', 'axe/lynx', ';',
 'ben', '&', u'jerri', "'s", ';', 'dove', ';', 'flora/becel', ';', 'hea
rtbrand', ';', "hellmann's/best", u'food', ';', 'knorr', ';', 'lipton',
';', 'lux/radox', ';', 'omo/surf', ';', u'rexona/sur', ';', 'sunsilk',
```

```
 ';', 'tresemm', ';', 'magnum', ';', u'vaselin', 'vo5', '.', u'unilev',
"'s", 'standard', u'industri', u'classif', u'code', '10890', ':', u'man
ufactur', 'food', u'product', 'n.e.c.', ',', '10410', ':', u'manufactu
r', u'oil', u'fat', ',', '10420', ':', u'manufactur', u'margarin', 'sim
ilar', u'edibl', u'fat', '.', '==', u'corpor', u'affair', '==', '===',
 'legal', u'structur', '===', u'unilev', 'two', u'hold', u'compani',
 ':', u'unilev', 'n.v.', ',', u'regist', 'head', u'offic', 'rotterdam',
',', u'netherland', u'unilev', 'plc', ',', u'regist', u'offic', 'port',
'sunlight', u'merseysid', ',', u'unit', 'kingdom', 'head', u'offic',
 u'unilev', u'hous', 'london', ',', u'unit', 'kingdom', '.', u'unilev',
'plc', u'unilev', 'n.v.', u'subsidiari', u'compani', u'oper', u'nearl
i', u'practic', u'singl', u'econom', u'entiti', ',', 'whilst', u'remai
n', u'separ', 'legal', u'entiti', u'differ', u'sharehold', u'separ', 's
tock', u'exchang', u'list', '.', 'there', u'seri', 'legal', u'agreemen
t', 'parent', u'compani', ',', u'togeth', 'special', u'provis', u'respe
ct', u'articl', u'associ', ',', 'known', u'foundat', u'agreement', '.',
'A', 'key', u'requir', u'agreement', u'peopl', u'board', 'two', 'paren
t', u'compani', '.', 'An', u'equalis', 'agreement', u'regul', 'mutual',
u'right', u'sharehold', u'unilev', 'plc', u'unilev', 'n.v.', u'object',
u'ensur', ',', u'principl', ',', 'make', u'financi', u'differ', 'hold',
u'share', u'unilev', 'plc', 'rather', u'unilev', 'n.v.', '(', 'vice',
 'versa', ')', '.', '===', 'senior', u'manag', '===', u'unilev', "'s",
 'highest', u'execut', u'bodi', u'unilev', 'leadership', u'execut',
 ',', 'led', 'chief', u'execut', '(', u'current', 'paul', 'polman',
 ')', '.', '==', 'logo', '==', 'In', '1930', ',', 'logo', u'unilev',
 u'use', 'helvetica', 'font', u'cap', '.', 'In', '1969', ',', u'typefa
c', 'logo', u'chang', ',', 'basic', 'design', u'remain', '.', 'the', 'c
urrent', u'unilev', u'corpor', 'logo', u'introduc', '2004', u'design',
 'wolff', u'olin', ',', 'brand', u'consult', u'agenc', '.', 'the',
 "'U", "'", 'shape', 'made', '25', 'distinct', u'symbol', ',', 'icon',
 u'repres', 'one', u'compani', "'s", u'sub-brand', u'corpor', u'valu',
 '.', 'the', 'brand', u'ident', u'develop', 'around', 'idea', '``', u'a
d', u'vital', 'life', '.', "'", '==', u'advertis', '==', 'dove', ':',
 'dove', u'describ', u'dedic', '``', 'help', '...', 'women', 'develop',
u'posit', 'relationship', 'way', 'look', u'help', u'rais', 'self-estee
m', u'realiz', 'full', u'potenti', "'", '.', '(', 'dove', ',', '``',
 'our', 'vision', "'", ')', 'dove', u'employ', 'use', u'advertis', u'p
roduct', 'display', u'messag', u'posit', 'self-esteem', '.', 'In', u'se
ptemb', '2004', 'dove', u'creat', 'real', u'beauti', 'campaign', ',',
 u'focus', u'predomin', 'women', u'shape', 'colour', '.', 'later', '200
7', 'campaign', u'further', u'includ', 'women', u'age', '.', u'thi', 'c
ampaign', u'consist', u'mostli', u'advertis', ',', 'shown', u'televis',
u'popularis', 'internet', '.', 'dove', 'fell', u'scrutini', u'gener',
 'public', 'felt', 'dove', u'advertis', u'describ', 'opinion', u'cellul
it', 'still', u'unsightli', ',', 'women', "'s", u'age', u'process', u's
ometh', u'asham', '.', u'lynx/ax', ':', 'axe', ',', 'known', 'lynx',
 u'unit', 'kingdom', ',', u'republ', 'ireland', ',', 'australia', 'ne
w', 'zealand', ',', u'toiletri', 'brand', u'market', u'toward', 'youn
g', 'age', '16', '24', '.', u'it', u'market', '``', 'tongue-in-cheek',
 'take', u"'mate", 'game', "'", "'", ',', u'suggest', 'women', u'insta
ntli', 'drawn', u'use', u'product', '.', u'unlik', 'dove', "'s", 'lon
g', u'run', u'beauti', 'campaign', 'lynx', u'advertis', 'often', u'crea
t', 'mini', u'seri', "'", u'advertis', u'base', 'around', 'singular',
 'product', 'rather', u'commun', u'overarch', 'idea', '.', u'thi', u'ad
vertis', 'campaign', u'thrive', u'controversi', '.', u'use', u'imag',
 u'compani', u'know', u'receiv', u'complaint', u'garner', 'brand', 'fre
e', u'public', u'notorieti', '.', 'A', 'wide', u'varieti', u'advert',
```

```
 u'ban', u'countri', 'around', 'world', '.', 'In', '2012', 'lynx',
 "'s", "'clean", u'ball', "'", 'advert', u'ban', '.', u'thi', 'advert',
u'design', u'televis', ',', u'show', u'attract', 'young', 'woman', u'cl
ean', u'variou', 'sport', u'ball', '.', 'In', '2011', 'UK', 'lynx',
 "'s", 'shower', 'gel', 'campaign', u'ban', '.', 'the', 'poster', 'lyn
x', 'shower', 'gel', u'show', 'woman', u'undon', 'bikini', 'shower', 'b
each', ',', u'headlin', ':', '``', 'the', 'cleaner', 'dirtier', 'get',
 '.', "'", 'both', u'advertis', u'campaign', 'make', 'stark', u'compar
ison', 'women', u'sexual', u'portray', u'advertis', u'sale', u'effici',
'.', 'lynx', u'commonli', u'portray', 'women', 'visual', u'advertis',
 'hyper', 'sexual', ',', u'flawless', u'stereotyp', u'attract', u'arou
s', ',', u'age', u'statur', ',', 'use', 'lynx', 'product', '.', 'thei
r', 'target', u'audienc', u'age', '16-24', u'singl', '.', '==', u'envir
onment', 'record', '==', u'unilev', u'declar', 'goal', u'decoupl', u'en
vironment', 'impact', 'growth', ',', ':', u'halv', u'environment', 'foo
tprint', u'product', 'next', '10', u'year', ';', u'help', '1', 'billio
n', u'peopl', u'improv', 'health', u'well-b', ';', u'sourc', u'agricult
ur', 'raw', u'materi', u'sustain', '.', '===', 'palm', 'oil', '===',
 u'unilev', u'criticis', u'greenpeac', u'caus', u'deforest', ',', u'uni
lev', u'target', '2008', u'greenpeac', 'UK', ',', u'criticis', u'compan
i', u'buy', 'palm', 'oil', u'supplier', u'damag', 'indonesia', "'s",
 u'rainforest', '.', 'By', '2008', ',', 'indonesia', u'lose', '2', '%',
u'remain', 'rainforest', 'year', ',', 'fastest', u'deforest', 'rate',
 u'countri', '.', 'the', u'unit', u'nation', u'environment', u'program
m', u'state', 'palm', 'oil', u'plantat', u'lead', u'caus', u'deforest',
'indonesia', '.', u'furthermor', ',', 'indonesia', 'fourteenth', 'large
st', u'emitt', u'greenhous', u'gase', u'larg', 'due', u'destruct', u'ra
inforest', 'palm', 'oil', u'industri', ',', u'contribut', '4', '%', 'gl
obal', 'green', u'hous', u'ga', u'emiss', '.', u'accord', u'greenpeac',
',', 'palm', 'oil', u'expans', u'take', 'place', u'littl', 'oversight',
'central', 'local', u'govern', u'procedur', u'environment', 'impact',
 u'assess', ',', u'land-us', u'plan', u'ensur', 'proper', u'process',
 u'develop', u'concess', u'neglect', '.', u'plantat', u'off-limit',
 ',', 'law', ',', 'palm', 'oil', u'plantat', u'establish', 'well', u'il
leg', 'use', 'fire', 'clear', 'forest', u'area', u'commonplac', '.',
 u'unilev', ',', u'found', 'member', u'roundtabl', u'sustain', 'palm',
 'oil', '(', 'rspo', ')', ',', u'respond', u'publicis', 'plan', 'obtai
n', 'palm', 'oil', u'sourc', u'certifi', u'sustain', '2015', '.', 'It',
u'claim', 'goal', '2012', u'encourag', 'rest', u'industri', u'becom',
 '100', '%', u'sustain', '2020', '.', 'In', 'cte', u"d'ivoir", '(', u'i
vori', 'coast', ')', ',', 'one', u'unilev', "'s", 'palm', 'oil', u'supp
lier', u'accus', u'clear', 'forest', u'plantat', ',', u'activ', u'threa
ten', u'primat', u'speci', ',', u'miss', 'waldron', "'s", 'red', u'colo
bu', '.', u'unilev', u'interven', 'halt', u'clearanc', u'pend', u'resul
t', u'environment', u'assess', '.', u'accord', u'amnesti', u'intern',
 '2016', u'unilev', 'palm', 'oil', u'provid', 'wilmar', u'intern', u'pr
ofit', '8', '14', 'year', 'old', 'child', 'labor', u'forc', 'labor',
 '.', 'some', u'worker', u'extort', ',', u'threaten', 'paid', 'work',
 '.', 'some', u'worker', u'suffer', u'sever', u'injuri', 'toxic', u'ba
n', u'chemic', '.', 'In', '2016', u'singapore-bas', 'wilmar', u'inter
n', u'world', 'biggest', 'palm', 'oil', 'grower', '.', '===', 'paper',
 'use', '===', 'for', u'year', ',', u'unilev', u'purchas', 'paper', u'p
ackag', 'asia', 'pulp', '&', 'paper', ',', 'third', 'largest', 'paper',
u'produc', 'world', ',', u'label', '``', 'forest', u'crimin', "'", u'd
estroy', '``', u'preciou', 'habitat', "'", 'indonesia', "'s", 'rainfor
est', '.', 'In', '2011', ',', u'unilev', u'cancel', 'contract', 'asia',
'pulp', '&', 'paper', ',', u'greenpeac', u'execut', 'director', 'phil',
```

```
'radford', u'commend', u'compani', u'effort', 'made', u'toward', 'fores
t', u'protect', ',', '``', u'take', 'rainforest', u'conserv', u'seriou
s', '.', "'", '===', 'rainforest', u'allianc', '===', u'unilev', u'cer
tifi', 'tea', u'product', 'rainforest', u'allianc', 'scheme', '.', 'th
e', u'compani', u'state', 'least', '50', '%', 'tea', u'product', u'orig
in', u'certifi', u'farm', ',', u'compar', u'allianc', "'s", '30', '%',
 'minimum', u'entri', 'point', '.', u'unilev', u'decid', 'scheme', u'fa
irtrad', ',', u'accord', u'compani', "'s", u'analysi', ',', u'fairtra
d', 'might', '``', 'lack', 'scale', u'organiz', u'flexibl', u'certifi',
u'industri', 'tea', u'estat', "'", '.', '===', u'critic', '===', 'th
e', 'rainforest', u'allianc', u'certif', 'scheme', u'criticis', u'offe
r', u'produc', 'minimum', u'guarante', 'price', ',', u'therefor', u'lea
v', u'vulner', 'market', 'price', u'variat', '.', 'the', u'altern', u'c
ertif', ',', u'fairtrad', ',', u'howev', u'receiv', 'similar', u'criti
c', 'well', '.', 'the', 'rainforest', u'allianc', u'certif', u'furtherm
or', u'criticis', u'allow', 'use', 'seal', u'product', 'contain', 'mini
mum', '30', '%', u'certifi', 'content', ',', u'accord', u'endang', u'in
tegr', u'certif', '.', '===', u'juli', '-', u'septemb', '2016', 'salmon
ella', 'affair', '===', '====', 'salmonella', 'affair', u'cereal', 'isr
ael', '====', 'In', u'juli', '2016', ',', u'rumour', 'salmonella', u'co
ntamin', u'cereal', 'spread', 'among', u'isra', u'consum', '.', u'init
i', ',', u'unilev', u'provid', 'public', u'inform', 'subject', u'quer
i', 'matter', u'initi', u'rebuf', u'compani', u'non-stori', u'nonsens',
'.', 'On', 'night', '26', u'juli', '2016', ',', u'unilev', u'stop', u't
ransfer', u'cornflak', u'retail', u'chain', '.', 'On', '28', u'juli',
 ',', 'yediot', 'ahronot', u'report', u'ten', u'thousand', u'box', 'bre
akfast', 'cereal', u'destroy', '.', 'By', '28', u'juli', ',', u'despi
t', u'compani', "'s", u'assur', u'noth', u'contamin', u'releas', u'cons
umpt', ',', u'mani', u'custom', u'stop', u'buy', u'unilev', u'product',
u'start', 'throw', 'away', u'cornflak', 'made', u'unilev', '.', 'the',
 u'compani', 'withheld', u'inform', u'affect', u'product', u'date',
 '.', 'On', '2', 'august', '2016', ',', u'globe', u'report', u'compan
i', u'publish', u'inform', 'telma', u'cereal', u'handl', u'packag', 'li
ne', u'contamin', u'discov', 'telma', u'announc', 'made', ':', '``', 'W
e', u'stress', 'telma', u'product', u'store', u'home', 'safe', 'eat',
 '.', u'accord', u'compani', "'s", 'strict', u'procedur', ',', u'ever
i', u'product', 'batch', u'check', 'put', 'hold', '.', 'these', u'produ
ct', u'market', 'test', u'result', u'product', u'seri', u'return', ',',
u'confirm', 'well', '.', 'If', 'flaw', u'discov', ',', 'batch', u'marke
t', u'store', ',', 'case', '.', "'", 'In', u'follow', u'day', 'healt
h', u'minist', ',', 'yakov', 'litzman', ',', u'threaten', 'pull', u'uni
lev', "'s", u'licenc', 'israel', '.', 'He', u'accus', u'unilev', u'li
e', u'ministri', u'regard', u'salmonella-infect', 'breakfast', u'cerea
l', '.', 'On', '7', 'august', ',', u'globe', u'report', u'contamin', 'm
ay', u'sourc', 'pigeon', u'faec', ',', 'health', u'ministri', 'said',
 'might', u'sourc', u'contamin', 'pigeon', u'faec', u'possibl', u'sour
c', '.', u'globe', 'said', u'product', 'line', u'automat', '(', '``',
 'without', 'human', u'hand', "'", ')', u'possibl', u'sourc', 'human',
'slim', u'chanc', '.', 'On', '8', 'august', '2016', ',', u'isra', 'heal
th', u'minist', u'suspend', u'manufactur', u'licenc', u'unilev', u'carr
i', 'number', u'correct', ';', 'action', 'came', u'inspect', 'arad', 'p
lant', ',', u'state', '``', u'thi', u'seri', u'neglig', u'mistak', ',',
u'incid', u'malici', 'intent', 'firm', "'s", u'manag', u'qualiti', 'con
trol', u'procedur', '.', "'", 'An', u'investig', 'led', 'prof.', 'itam
r', 'grutto', 'eli', 'gordon', u'conclud', 'event', u'caus', u'neglig',
'.', 'On', '23', u'septemb', u'report', u'cereal', u'produc', '18th',
 '20th', 'arad', 'plant', u'trace', 'salmonella', '.', '=====', u'clas
```

```
s', u'action', '=====', 'A', u'file', u'class', 'action', 'must', 'firs
t', u'approv', u'isra', 'court', ',', u'approv', 'case', 'held', '.',
 'for', 'sum', '1.2', 'million', u'ni', '(', '~', '$', '329k', 'usd',
 ')', u'unilev', u'hide', u'contamin', u'mislead', 'public', 'for', 'su
m', '76', 'million', u'ni', '(', '~', '$', '23m', 'usd', ')', u'unile
v', '15-year-old', 'teen', u'hospitalis', u'salmonellosi', u'allegedl
i', u'contract', u'unilev', u'product', '====', 'salmonella', 'affair',
'tehnia', '====', 'On', '31', 'august', ',', u'unilev', u'state', 'tehn
ia', u'product', u'produc', 'rjm', u'contamin', 'salmonella', '.', '==
=', u'kodaikan', '===', 'In', '2015', ',', 'indian', 'rapper', 'sofia',
'ashraf', u'releas', 'music', 'video', '``', u'kodaikan', 'Wo', "n't",
 ',', "'", 'set', 'beat', 'nicki', 'minaj', "'s", '``', 'anaconda',
 ',', "'", u'criticis', u'unilev', u'dump', u'mercuri', u'wast', 'grou
nd', 'indian', 'town', u'kodaikan', '.', u'unilev', u'acknowledg', 'too
k', u'thermomet', u'factori', u'acquisit', u'chesebrough-pond', '.',
 u'accord', u'unilev', "'s", 'statement', ',', u'factori', 'sold', u'me
rcury-contamin', 'scrap', u'glass', 'local', 'dealer', ',', u'prompt',
 'hindustan', u'unilev', u'immedi', 'close', u'factori', ',', 'plan',
 'clean-up', u'affect', u'site', 'monitor', 'health', u'worker', '.',
 u'unilev', "'s", u'websit', u'state', u'wait', u'sinc', '2010', 'loca
l', u'govern', 'tamil', 'nadu', u'pollut', 'control', 'board', 'give',
 u'permiss', 'clean-up', '.', '==', 'see', '==', 'list', 'food', u'comp
ani', '==', u'refer', '==', '==', u'extern', u'link', '==', u'offici',
 u'websit']
```

Put the cleaned word into a string

In [108]:
```
w_str_list=[]
for p in after_stem_list:
    word_str = ''
    for i in p:
        if (i!=p[0]):
            word_str += ' '
        word_str += i
    w_str_list.append(word_str)
```

In [109]:
```
import csv
f= open("wiki_result.csv","w")
writer = csv.writer(f, delimiter='|')
for i in w_str_list:
    p = ['0',i]
    writer.writerow(p)
f.close()
```

Find a different sort of wiki pages

In [98]:
```
t = wikipedia.search('asia',results=30)
plist = []
for p in t:
    print p
    if p =="XXX": continue
    page = wikipedia.page(p, auto_suggest=False)
    plist.append(page)

tokenized_docs_list = []

for i in plist:
    tokenized_docs = word_tokenize(i.content.encode('ascii','ignore'))
    tokenized_docs_list.append(tokenized_docs)
#-----------------------#
t_nsw_list = []
for p in tokenized_docs_list:
    tokenized_nsw = []
    for i in p:
        if not i in stopwords.words():
            tokenized_nsw.append(i)
    t_nsw_list.append(tokenized_nsw)
#-----------------------#
after_stem_list = []
for p in t_nsw_list:
    words = []
    for i in p:
        words.append(porter.stem(i))
    after_stem_list.append(words)
#-----------------------#
w_str_list2 = []
for p in after_stem_list:
    word_str = ''
    for i in p:
        if (i!=p[0]):
            word_str += ' '
        word_str += i
    w_str_list2.append(word_str)
#-----------------------#
```

```
Asia
Unilever
Eurasia
Eric Hoffer
Shamanism
Asian giant hornet
List of Asian pornographic actors
Georgia (country)
Calligraphy
Hornet
Asia Argento
Time (magazine)
Pacific War
Boundaries between the continents of Earth
Demographics of Russia
List of most common surnames in Asia
Humid subtropical climate
Aedes albopictus
Buddhism
Lists of World Heritage Sites
XXX
Shanghai SIPG F.C.
Alcohol flush reaction
Asian Boyz
KLM
XXX (Asia album)
Pre-Indo-European languages
Tatars
Diospyros kaki
Indian subcontinent
```

In [110]:
```
f= open("wiki_result.csv","a")
writer = csv.writer(f, delimiter='|')
for i in w_str_list2:
    p = ['1',i]
    writer.writerow(p)
f.close()
```

In [96]:
```

```

```
[u'Asia', u'Unilever', u'Eurasia', u'Eric Hoffer', u'Shamanism', u'Asia
n giant hornet', u'List of Asian pornographic actors', u'Georgia (count
ry)', u'Calligraphy', u'Hornet', u'Asia Argento', u'Time (magazine)',
 u'Pacific War', u'Boundaries between the continents of Earth', u'Demog
raphics of Russia', u'List of most common surnames in Asia', u'Humid su
btropical climate', u'Aedes albopictus', u'Buddhism', u'Lists of World
 Heritage Sites', u'XXX', u'Shanghai SIPG F.C.', u'Alcohol flush reacti
on', u'Asian Boyz', u'KLM', u'XXX (Asia album)', u'Pre-Indo-European la
nguages', u'Tatars', u'Diospyros kaki', u'Indian subcontinent']
```

Open the file that recorded clean wiki pages

In [40]:
```
from pyspark.sql import Row
from pyspark.sql import SparkSession
import csv
spark = SparkSession.builder.getOrCreate()
f = open("wiki_result.csv")
reader = csv.reader(f,delimiter='|')
ww = []
for w in reader:
    ww.append(w)
ww= map(lambda p: Row(label=int(p[0]), text=str(p[1])),ww)
wikies = spark.createDataFrame(ww)
wikies.show()
f.close
```

```
+-----+--------------------+
|label|                text|
+-----+--------------------+
|    0|asia ( ) earth 's...|
|    0|unilev ( ) dutch-...|
|    0|eurasia combin co...|
|    0|eric hoffer ( jul...|
|    0|shaman ( shah-men...|
|    0|the asian giant h...|
|    0|list asian pornog...|
|    0|georgia ( ; georg...|
|    0|calligraphi ( gre...|
|    0|hornet ( insect g...|
|    0|asia argento ( it...|
|    0|time american wee...|
|    0|the pacif war , s...|
|    0|the boundari cont...|
|    0|the demograph rus...|
|    0|thi list common s...|
|    0|A humid subtrop c...|
|    0|aed albopictu ( s...|
|    0|buddhism ( ) reli...|
|    0|thi list list wor...|
+-----+--------------------+
only showing top 20 rows
```

Out[40]: <function close>

Do tokenize

In [41]:
```
from pyspark.ml.feature import HashingTF, IDF, Tokenizer
tokenizer = Tokenizer(inputCol="text", outputCol="words")
wordsData = tokenizer.transform(wikies)
wordsData.show()
```

```
+-----+--------------------+--------------------+
|label|                text|               words|
+-----+--------------------+--------------------+
|    0|asia ( ) earth 's...|[asia, (, ), eart...|
|    0|unilev ( ) dutch-...|[unilev, (, ), du...|
|    0|eurasia combin co...|[eurasia, combin,...|
|    0|eric hoffer ( jul...|[eric, hoffer, (,...|
|    0|shaman ( shah-men...|[shaman, (, shah-...|
|    0|the asian giant h...|[the, asian, gian...|
|    0|list asian pornog...|[list, asian, por...|
|    0|georgia ( ; georg...|[georgia, (, ;, g...|
|    0|calligraphi ( gre...|[calligraphi, (, ...|
|    0|hornet ( insect g...|[hornet, (, insec...|
|    0|asia argento ( it...|[asia, argento, (...|
|    0|time american wee...|[time, american, ...|
|    0|the pacif war , s...|[the, pacif, war,...|
|    0|the boundari cont...|[the, boundari, c...|
|    0|the demograph rus...|[the, demograph, ...|
|    0|thi list common s...|[thi, list, commo...|
|    0|A humid subtrop c...|[a, humid, subtro...|
|    0|aed albopictu ( s...|[aed, albopictu, ...|
|    0|buddhism ( ) reli...|[buddhism, (, ), ...|
|    0|thi list list wor...|[thi, list, list,...|
+-----+--------------------+--------------------+
only showing top 20 rows
```

Do tokenize

In [42]:
```
hashingTF = HashingTF(inputCol="words", outputCol="rawFeatures",numFeatu
res=300)
featurizedData = hashingTF.transform(wordsData)
featurizedData.show()
```

```
+-----+--------------------+--------------------+--------------------+
|label|                text|               words|         rawFeatures|
+-----+--------------------+--------------------+--------------------+
|    0|asia ( ) earth 's...|[asia, (, ), eart...|(300,[0,1,2,3,4,5...|
|    0|unilev ( ) dutch-...|[unilev, (, ), du...|(300,[0,1,2,3,4,6...|
|    0|eurasia combin co...|[eurasia, combin,...|(300,[0,1,2,3,4,5...|
|    0|eric hoffer ( jul...|[eric, hoffer, (,...|(300,[0,1,2,3,4,5...|
|    0|shaman ( shah-men...|[shaman, (, shah-...|(300,[0,1,2,3,4,5...|
|    0|the asian giant h...|[the, asian, gian...|(300,[0,1,2,3,4,5...|
|    0|list asian pornog...|[list, asian, por...|(300,[1,4,8,10,12...|
|    0|georgia ( ; georg...|[georgia, (, ;, g...|(300,[0,1,2,3,4,5...|
|    0|calligraphi ( gre...|[calligraphi, (, ...|(300,[0,1,2,3,4,5...|
|    0|hornet ( insect g...|[hornet, (, insec...|(300,[0,1,3,4,5,6...|
|    0|asia argento ( it...|[asia, argento, (...|(300,[0,1,2,3,4,5...|
|    0|time american wee...|[time, american, ...|(300,[0,1,2,3,4,5...|
|    0|the pacif war , s...|[the, pacif, war,...|(300,[0,1,2,3,4,5...|
|    0|the boundari cont...|[the, boundari, c...|(300,[0,1,2,3,4,5...|
|    0|the demograph rus...|[the, demograph, ...|(300,[0,1,2,3,4,5...|
|    0|thi list common s...|[thi, list, commo...|(300,[1,4,5,7,9,1...|
|    0|A humid subtrop c...|[a, humid, subtro...|(300,[0,1,2,3,4,5...|
|    0|aed albopictu ( s...|[aed, albopictu, ...|(300,[0,1,2,3,4,5...|
|    0|buddhism ( ) reli...|[buddhism, (, ), ...|(300,[0,1,2,3,4,5...|
|    0|thi list list wor...|[thi, list, list,...|(300,[1,2,3,5,6,1...|
+-----+--------------------+--------------------+--------------------+
only showing top 20 rows
```

In [43]:
```
idf = IDF(inputCol="rawFeatures", outputCol="features")
idfModel = idf.fit(featurizedData)
rescaledData = idfModel.transform(featurizedData)

rescaledData.select("text", "features").show()
rescaledData = rescaledData.select("label","features")
```

```
+--------------------+--------------------+
|                text|            features|
+--------------------+--------------------+
|asia ( ) earth 's...|(300,[0,1,2,3,4,5...|
|unilev ( ) dutch-...|(300,[0,1,2,3,4,6...|
|eurasia combin co...|(300,[0,1,2,3,4,5...|
|eric hoffer ( jul...|(300,[0,1,2,3,4,5...|
|shaman ( shah-men...|(300,[0,1,2,3,4,5...|
|the asian giant h...|(300,[0,1,2,3,4,5...|
|list asian pornog...|(300,[1,4,8,10,12...|
|georgia ( ; georg...|(300,[0,1,2,3,4,5...|
|calligraphi ( gre...|(300,[0,1,2,3,4,5...|
|hornet ( insect g...|(300,[0,1,3,4,5,6...|
|asia argento ( it...|(300,[0,1,2,3,4,5...|
|time american wee...|(300,[0,1,2,3,4,5...|
|the pacif war , s...|(300,[0,1,2,3,4,5...|
|the boundari cont...|(300,[0,1,2,3,4,5...|
|the demograph rus...|(300,[0,1,2,3,4,5...|
|thi list common s...|(300,[1,4,5,7,9,1...|
|A humid subtrop c...|(300,[0,1,2,3,4,5...|
|aed albopictu ( s...|(300,[0,1,2,3,4,5...|
|buddhism ( ) reli...|(300,[0,1,2,3,4,5...|
|thi list list wor...|(300,[1,2,3,5,6,1...|
+--------------------+--------------------+
only showing top 20 rows
```

In [44]:
```
from pyspark.ml.clustering import KMeans

kmeans = KMeans().setK(2).setSeed(1)
model = kmeans.fit(rescaledData)


wssse = model.computeCost(rescaledData)
print("Within Set Sum of Squared Errors = " + str(wssse))


centers = model.clusterCenters()
print("Cluster Centers: ")
for center in centers:
    print(center)
```

```
Within Set Sum of Squared Errors = 53447.3276304
Cluster Centers:
[  3.64005606   0.          6.9126512   2.73938205   3.9037373
   2.33138898   7.42197374   4.86614262   7.00360714   3.52862577   0.
   3.47475518   5.25497411   2.3896737   2.76852441   0.71731246
   4.4073333   1.77148904   4.51706615   5.45735621   6.33847393   0.
   6.18741141   2.22842045   2.76852441   2.78838379   2.8098329
   2.76852441   2.42906735   7.18551901   6.67360095   1.72702476
   4.41136294   5.95761387   6.32143761   4.320407   1.47998832
   3.52862577   3.04577306   4.95709856   3.20289532   6.69696294
   3.60291263   1.39883339   3.1250631   4.17147429   1.76914732
   3.32222929   3.06722217   0.58626499   6.32674656   5.42294066
   4.95709856   1.93145832   4.42006807   2.89562932   0.          3.4
9406762
   7.41290886   5.77570199   3.95477433   3.14214947   2.65968915
   4.07533015   5.25497411   0.          4.7132242   2.71023969
   7.55923641   3.00861776   1.63013206   2.25215614   2.56452788
   1.58723385   2.42374899   0.          3.98585034   1.79722902
   1.58661625   4.1172993   2.9343309   2.33138898   6.57438696
   5.25503099   3.68371544   2.9796245   3.98585034   3.46719941
   5.1844884   2.9254495   4.42872259   5.52584906   1.77315422
   6.98857382   2.46743116   2.31674054   4.43996496   2.97147433   0.
   1.51049451   2.3896737   2.62094938   7.25944877   7.03758856
   2.31650346   4.03243194   1.58661625   5.3486538   3.73807437
   2.6743269   2.18780882   4.36545513   3.23147834   1.30357746
   3.96863931   6.64308389   8.95156643   3.64279528   4.4004967
   5.82117996   3.90059934   2.35940167   2.76852441   4.19720749
   3.61365292   4.72970872   2.78008864   4.80832739   9.36358492
   3.49148234   4.90293265   1.80172491  10.0223733   1.73120604   0.
   3.78850709   6.16580924   3.50180357   2.09825008   2.85273111
   5.63926809   4.58596375   8.45129857   1.51641199   2.04158162
   6.06759897   1.89551498   4.4572115   2.38085077   0.          4.2
2199838
   3.25049945  11.96018419   0.          3.00861776   2.63968013
   3.32222929   4.01149035   2.63824005   3.16337328   1.73359971
   2.89718747   3.63823748   3.15719148   2.91058998   2.08782637   0.
   3.95477433   3.34606055   4.13967747   3.32222929   3.1959168
   3.15719148   5.82117996   2.23070703   4.42963906   2.50954541
   1.64277965   3.46794111   1.89612468   4.72970872   3.84679182
   3.15719148   3.03683835   4.83763213   3.81764945   3.8257732
  14.93461634   3.67193764   2.57389273   3.3198917   4.73176703
   3.08898738   3.12004805  10.24932943   5.05150637   5.75733937
   1.96860175   1.89261996   3.93082607   3.38005205   6.33847393
   4.7315499   2.27360524   2.50954541   4.60449323   5.11087003
  11.24895182   2.80817035   4.55525753   3.97280748   3.26026412
   2.36053134   1.01389358   6.39001403   8.14055635   2.01621597
   3.26394457   3.45404953   4.36545513   2.4143229   2.70258736
   0.92858573   3.19433491   2.6743269   6.63978339   5.02272221
   2.68109733   5.53227373   1.36196262   2.42374899   6.57981643
   1.65533645   3.46794111   4.63875278   3.8593154   4.31127032
   2.54676623   3.52862577   2.59396609   1.66223369   1.92339591
   4.60487422   4.38639081   0.92231156   2.13814292   2.19146232
   3.79604794   2.21845457   0.39314241   5.79125864   5.38168537
   2.95997664   4.38292464   2.91058998   8.23151229   5.07077103
   4.43690915   1.13114658   1.15825173   1.7776841   11.70448114
   7.83929544   0.          2.55543502   2.42374899   1.95253827
   2.04650858   2.87418021   3.15719148   5.50283418   4.61155781
```

```
   3.84679182   5.67785988   2.23249543   8.41457293   1.76914732
   3.90006006   3.95477433   4.06312431   7.31362376   7.74702369
   2.8559515    3.96336126   3.72578218   3.00154592   2.23249543
   4.36545513   3.4563256    4.35416853   3.38005205   3.00861776
   0.67592905   2.08056329   8.98398012   3.64005606   3.00861776
   1.41812603   3.04702389   1.84462312]
[ 0.8589418   0.          0.86218648  0.53427664  0.92052414  0.7771296
6
  1.35437477  0.78638987  1.02325429  0.71965394  0.          0.6792216
9
  0.97063525  1.00784003  0.60713255  0.30175404  1.22569617  0.7776626
  0.54034797  2.42549165  1.04964045  0.          1.04640046  0.4547796
8
  0.38856483  0.63006749  0.33514228  0.78320099  0.97408409  1.1085254
8
  1.40854751  0.31591916  1.14642379  1.43066109  1.18432209  1.1180000
6
  0.40217074  0.97501502  0.49601058  0.99483056  1.44069628  0.8795829
9
  0.95180037  0.52213399  0.65105481  0.6207551   0.46510321  0.7042737
5
  0.74178158  0.17099396  1.27853003  1.17620859  0.84323733  1.9190771
7
  0.51846037  1.26907211  0.          1.28904404  1.07062717  1.0137797
1
  0.85777069  0.80133841  0.53622765  0.57197616  0.99320817  0.
  1.68168201  0.74070171  0.8943541   0.82025073  0.7149702   0.6345360
5
  0.55249062  0.49601058  0.67028456  0.          0.83038549  0.4124500
2
  0.38027307  1.09478627  0.46429286  0.6678458   1.06013537  0.9652097
7
  0.82428818  1.10607273  0.83038549  0.53046345  1.44013567  0.8351977
8
  0.79084332  2.79904119  1.16654883  1.70425492  0.94901198  0.5528585
4
  0.94286695  0.97501502  0.          0.43395105  0.65570315  0.5690219
1
  1.0609269   1.17293143  0.46026207  0.64347318  0.21646313  0.7660832
3
  0.59818219  0.50298394  0.69262738  0.84356621  0.79703608  0.3118125
1
  0.97910509  0.98855415  1.13751752  0.6192752   0.78320099  0.9758814
1
  0.58689573  0.55410191  0.93498412  0.76608323  0.97141207  0.8527119
1
  1.00333586  0.7776626   1.79257819  1.33871109  0.72739215  0.7641244
  0.82391132  0.2816371   0.          1.14140919  0.85120358  0.9095593
7
  0.74070171  0.77306153  0.75796614  2.51742719  1.21893729  0.4387766
2
  0.31324943  0.68847384  0.41830037  0.68096287  0.65241031  0.
  1.36487016  0.48531763  1.1297793   0.          0.5881043   1.1817716
9
  0.83177159  1.0214443   0.58091329  0.59313249  0.76747904  1.3387110
9
  0.85271191  0.61131894  0.57794918  0.67221682  0.          0.6207551
```

```
  0.75518728  0.78646722  1.05641063  0.57644473  0.42560179  1.4211865
1
  0.56303903  0.62534652  0.70156451  0.3890486   0.81962894  0.4966040
8
  0.91903395  0.73463038  1.0214443   0.93583126  1.86389692  0.5646332
7
  1.08335002  1.89293651  0.84998557  0.91605557  0.63479664  1.0384379
9
  0.84830146  1.45478431  1.04127704  1.00596787  0.68539754  0.4642928
6
  0.66726986  0.77342643  0.58036608  1.5010987   0.7582612   0.6881588
2
  0.50494771  1.14140919  0.6230628   0.6192752   0.62891315  0.9094698
2
  0.74432663  0.74178158  0.51606266  0.40952334  0.8962891   1.0327288
7
  0.51388483  0.49177736  0.92728125  1.29171076  0.65774822  0.5719761
6
  0.93632394  0.68096287  1.02918252  1.20327125  1.03123524  0.8317715
9
  0.66726986  0.2720415   0.54963334  0.71175899  0.21122783  1.1899797
9
  0.66322037  0.76448188  0.65241031  0.58742376  0.88215644  2.9788431
7
  0.33480329  0.54034797  0.82391132  0.69625251  0.37089079  0.5704588
3
  0.6422718   1.12036137  0.5060557   0.16237241  1.59527726  0.8969475
6
  1.27800924  1.15299395  0.72006783  1.07062717  0.78927231  1.2695270
1
  0.27876326  0.29939377  0.63748917  0.94901198  1.11712389  0.
  0.81904968  0.65687887  0.41892146  0.72006783  0.43345069  1.0137060
9
  1.37381363  0.91605557  0.80141496  1.09189613  0.44755215  1.0412770
4
  0.62013762  0.92084751  0.82391132  0.81262486  1.21893729  0.9254894
3
  0.57070459  1.01998268  0.68995966  0.85271191  0.26911632  0.9094698
2
  0.88113564  0.95627265  0.71965394  0.70417751  0.20116936  0.5004791
4
  1.1994457   1.11430287  1.16847038  0.33054505  0.73901699  0.7417815
8]
```

Summary

At first I took 10 pages for each category. The Error is around 2000. When I finally used 30 pages for each category. The Error increased to 50000. So the model is not effective to cluster.

next we perform the classification to the wikipages. Question3-wiki:

In [45]:
```
splits = rescaledData.select("label", "features").randomSplit([0.8,
0.2], 1234)
train = splits[1]
test = splits[0]
```

In [46]:
```
from pyspark.ml.classification import NaiveBayes
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

nb = NaiveBayes()
model = nb.fit(train)
predictions = model.transform(test)

evaluator = MulticlassClassificationEvaluator(labelCol="label", predicti
onCol="prediction",
                                              metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Test set accuracy = " + str(accuracy))
```

```
Test set accuracy = 0.382978723404
```

In [47]:
```
from pyspark.ml.classification import DecisionTreeClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

dt = DecisionTreeClassifier()

model = dt.fit(train)

predictions = model.transform(test)

evaluator = MulticlassClassificationEvaluator(labelCol="label", predicti
onCol="prediction",metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Test set accuracy = " + str(accuracy))
```

```
Test set accuracy = 0.382978723404
```

In [48]:
```
from pyspark.ml.classification import RandomForestClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

rf = RandomForestClassifier()

model = rf.fit(train)

predictions = model.transform(test)

evaluator = MulticlassClassificationEvaluator(labelCol="label",predictio
nCol="prediction",metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Test set accuracy of RandomForest= " + str(accuracy))
```

```
Test set accuracy of RandomForest= 0.382978723404
```

PIMA DATASET

already preprocessed the dataset.

In [3]:
```
from pyspark.ml.feature import HashingTF, IDF, Tokenizer
from pyspark.sql import SparkSession
from pyspark.sql.functions import monotonically_increasing_id
spark = SparkSession.builder.getOrCreate()

data = spark.read.format("csv").option("header",True).option("inferSchem
a",True).\
    load("pima/pima-indians-diabetes.data")
```

Vertorize the Data into feature

In [4]:
```
from pyspark.ml.feature import VectorAssembler
label = ["label"]
assembler = VectorAssembler(
    inputCols=[x for x in data.columns if x not in label],
    outputCol='features')
data = assembler.transform(data)
```

Split the Data

In [5]:
```
splits = data.select("label", "features").randomSplit([0.8, 0.2], 1234)
train = splits[1]
test = splits[0]
```

Use NaiveBayes method to build a model

In [6]:
```
from pyspark.ml.classification import NaiveBayes
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

nb = NaiveBayes()
model = nb.fit(train)
predictions = model.transform(test)

evaluator = MulticlassClassificationEvaluator(labelCol="label", predicti
onCol="prediction",
                                              metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Test set accuracy = " + str(accuracy))
```

```
Test set accuracy = 0.61648177496
```

Use DecisionTree method to build a model

In [7]:
```
from pyspark.ml.classification import DecisionTreeClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

dt = DecisionTreeClassifier()

model = dt.fit(train)

predictions = model.transform(test)

evaluator = MulticlassClassificationEvaluator(labelCol="label", predicti
onCol="prediction",metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Test set accuracy = " + str(accuracy))
```

```
Test set accuracy = 0.698890649762
```

Use RandomForest method to build a model

In [10]:
```
from pyspark.ml.classification import RandomForestClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

rf = RandomForestClassifier()

model = rf.fit(train)

predictions = model.transform(test)

evaluator = MulticlassClassificationEvaluator(labelCol="label",predictio
nCol="prediction",metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Test set accuracy of RandomForest= " + str(accuracy))
```

```
Test set accuracy of RandomForest= 0.765451664025
```

Statlog DATASET

already preprocessed the dataset.

In [13]:
```
from pyspark.ml.feature import HashingTF, IDF, Tokenizer
from pyspark.sql import SparkSession
from pyspark.sql.functions import monotonically_increasing_id
from pyspark.sql import Row
import csv
spark = SparkSession.builder.getOrCreate()
f = open("pima/australian.dat")
reader = csv.reader(f,delimiter=' ')
ww = []
for w in reader:
    ww.append(w)

data = map(lambda p: Row(label=int(p[0]),
a_1=float(p[1]),a_2=float(p[2]),
                                label_2=int(p[3]),label_3=int(p[4]),label
_4=int(p[5]),
                                a_3=float(p[6]),label_5 =int(p[7]),label_
6=int(p[8]),
                                a_4=float(p[9]),label_7=int(p[10]),label_
8=int(p[11]),

a_5=float(p[12]),a_6=float(p[13]),label_9=int(p[14]))
                               ,ww)
data = spark.createDataFrame(data)
f.close()
```

In [14]:
```
data.show()
```

```
+-----+-----+-----+----+-----+------+-----+-------+-------+-------+----
---+-------+-------+-------+-------+
|  a_1|  a_2|  a_3| a_4|  a_5|   a_6|label|label_2|label_3|label_4|labe
l_5|label_6|label_7|label_8|label_9|
+-----+-----+-----+----+-----+------+-----+-------+-------+-------+----
---+-------+-------+-------+-------+
|22.08|11.46|1.585| 0.0|100.0|1213.0|    1|      2|      4|      4|
  0|      0|      1|      2|      0|
|22.67|  7.0|0.165| 0.0|160.0|   1.0|    0|      2|      8|      4|
  0|      0|      0|      2|      0|
|29.58| 1.75| 1.25| 0.0|280.0|   1.0|    0|      1|      4|      4|
  0|      0|      1|      2|      0|
|21.67| 11.5|  0.0|11.0|  0.0|   1.0|    0|      1|      5|      3|
  1|      1|      1|      2|      1|
|20.17| 8.17| 1.96|14.0| 60.0| 159.0|    1|      2|      6|      4|
  1|      1|      0|      2|      1|
|15.83|0.585|  1.5| 2.0|100.0|   1.0|    0|      2|      8|      8|
  1|      1|      0|      2|      1|
|17.42|  6.5|0.125| 0.0| 60.0| 101.0|    1|      2|      3|      4|
  0|      0|      0|      2|      0|
|58.67| 4.46| 3.04| 6.0| 43.0| 561.0|    0|      2|     11|      8|
  1|      1|      0|      2|      1|
|27.83|  1.0|  3.0| 0.0|176.0| 538.0|    1|      1|      2|      8|
  0|      0|      0|      2|      0|
|55.75| 7.08| 6.75| 3.0|100.0|  51.0|    0|      2|      4|      8|
  1|      1|      1|      2|      0|
| 33.5| 1.75|  4.5| 4.0|253.0| 858.0|    1|      2|     14|      8|
  1|      1|      1|      2|      1|
|41.42|  5.0|  5.0| 6.0|470.0|   1.0|    1|      2|     11|      8|
  1|      1|      1|      2|      1|
|20.67| 1.25|1.375| 3.0|140.0| 211.0|    1|      1|      8|      8|
  1|      1|      1|      2|      0|
|34.92|  5.0|  7.5| 6.0|  0.0|1001.0|    1|      2|     14|      8|
  1|      1|      1|      2|      1|
|58.58| 2.71|2.415| 0.0|320.0|   1.0|    1|      2|      8|      4|
  0|      0|      1|      2|      0|
|48.08| 6.04| 0.04| 0.0|  0.0|2691.0|    1|      2|      4|      4|
  0|      0|      0|      2|      1|
|29.58|  4.5|  7.5| 2.0|330.0|   1.0|    1|      2|      9|      4|
  1|      1|      1|      2|      1|
|18.92|  9.0| 0.75| 2.0| 88.0| 592.0|    0|      2|      6|      4|
  1|      1|      0|      2|      1|
| 20.0| 1.25|0.125| 0.0|140.0|   5.0|    1|      1|      4|      4|
  0|      0|      0|      2|      0|
|22.42|5.665|2.585| 7.0|129.0|3258.0|    0|      2|     11|      4|
  1|      1|      0|      2|      1|
+-----+-----+-----+----+-----+------+-----+-------+-------+-------+----
---+-------+-------+-------+-------+
only showing top 20 rows
```

Vertorize the Data into feature

In [15]:
```
from pyspark.ml.feature import VectorAssembler
label = ["label"]
assembler = VectorAssembler(
    inputCols=[x for x in data.columns if x not in label],
    outputCol='features')
data = assembler.transform(data)
data.show()
```

```
+-----+-----+-----+----+-----+------+-----+-------+-------+-------+----
---+-------+-------+-------+-------+--------------------+
|  a_1|  a_2|  a_3| a_4|  a_5|   a_6|label|label_2|label_3|label_4|labe
l_5|label_6|label_7|label_8|label_9|            features|
+-----+-----+-----+----+-----+------+-----+-------+-------+-------+----
---+-------+-------+-------+-------+--------------------+
|22.08|11.46|1.585| 0.0|100.0|1213.0|    1|      2|      4|      4|
  0|      0|      1|      2|      0|[22.08,11.46,1.58...|
|22.67|  7.0|0.165| 0.0|160.0|   1.0|    0|      2|      8|      4|
  0|      0|      0|      2|      0|[22.67,7.0,0.165,...|
|29.58| 1.75| 1.25| 0.0|280.0|   1.0|    0|      1|      4|      4|
  0|      0|      1|      2|      0|[29.58,1.75,1.25,...|
|21.67| 11.5|  0.0|11.0|  0.0|   1.0|    0|      1|      5|      3|
  1|      1|      1|      2|      1|[21.67,11.5,0.0,1...|
|20.17| 8.17| 1.96|14.0| 60.0| 159.0|    1|      2|      6|      4|
  1|      1|      0|      2|      1|[20.17,8.17,1.96,...|
|15.83|0.585|  1.5| 2.0|100.0|   1.0|    0|      2|      8|      8|
  1|      1|      0|      2|      1|[15.83,0.585,1.5,...|
|17.42|  6.5|0.125| 0.0| 60.0| 101.0|    1|      2|      3|      4|
  0|      0|      0|      2|      0|[17.42,6.5,0.125,...|
|58.67| 4.46| 3.04| 6.0| 43.0| 561.0|    0|      2|     11|      8|
  1|      1|      0|      2|      1|[58.67,4.46,3.04,...|
|27.83|  1.0|  3.0| 0.0|176.0| 538.0|    1|      1|      2|      8|
  0|      0|      0|      2|      0|[27.83,1.0,3.0,0....|
|55.75| 7.08| 6.75| 3.0|100.0|  51.0|    0|      2|      4|      8|
  1|      1|      1|      2|      0|[55.75,7.08,6.75,...|
| 33.5| 1.75|  4.5| 4.0|253.0| 858.0|    1|      2|     14|      8|
  1|      1|      1|      2|      1|[33.5,1.75,4.5,4....|
|41.42|  5.0|  5.0| 6.0|470.0|   1.0|    1|      2|     11|      8|
  1|      1|      1|      2|      1|[41.42,5.0,5.0,6....|
|20.67| 1.25|1.375| 3.0|140.0| 211.0|    1|      1|      8|      8|
  1|      1|      1|      2|      0|[20.67,1.25,1.375...|
|34.92|  5.0|  7.5| 6.0|  0.0|1001.0|    1|      2|     14|      8|
  1|      1|      1|      2|      1|[34.92,5.0,7.5,6....|
|58.58| 2.71|2.415| 0.0|320.0|   1.0|    1|      2|      8|      4|
  0|      0|      1|      2|      0|[58.58,2.71,2.415...|
|48.08| 6.04| 0.04| 0.0|  0.0|2691.0|    1|      2|      4|      4|
  0|      0|      0|      2|      1|[48.08,6.04,0.04,...|
|29.58|  4.5|  7.5| 2.0|330.0|   1.0|    1|      2|      9|      4|
  1|      1|      1|      2|      1|[29.58,4.5,7.5,2....|
|18.92|  9.0| 0.75| 2.0| 88.0| 592.0|    0|      2|      6|      4|
  1|      1|      0|      2|      1|[18.92,9.0,0.75,2...|
| 20.0| 1.25|0.125| 0.0|140.0|   5.0|    1|      1|      4|      4|
  0|      0|      0|      2|      0|[20.0,1.25,0.125,...|
|22.42|5.665|2.585| 7.0|129.0|3258.0|    0|      2|     11|      4|
  1|      1|      0|      2|      1|[22.42,5.665,2.58...|
+-----+-----+-----+----+-----+------+-----+-------+-------+-------+----
---+-------+-------+-------+-------+--------------------+
only showing top 20 rows
```

Split the Data

In [17]:
```
splits = data.select("label", "features").randomSplit([0.8, 0.2], 1234)
train = splits[1]
test = splits[0]
```

Use NaiveBayes method to build a model

In [18]:
```
from pyspark.ml.classification import NaiveBayes
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

nb = NaiveBayes()
model = nb.fit(train)
predictions = model.transform(test)

evaluator = MulticlassClassificationEvaluator(labelCol="label", predicti
onCol="prediction",
                                              metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Test set accuracy = " + str(accuracy))
```

```
Test set accuracy = 0.388791593695
```

Use DecisionTree method to build a model

In [19]:
```
from pyspark.ml.classification import DecisionTreeClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

dt = DecisionTreeClassifier()

model = dt.fit(train)

predictions = model.transform(test)

evaluator = MulticlassClassificationEvaluator(labelCol="label", predicti
onCol="prediction",metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Test set accuracy = " + str(accuracy))
```

```
Test set accuracy = 0.647985989492
```

Use RandomForest method to build a model

In [20]:
```
from pyspark.ml.classification import RandomForestClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

rf = RandomForestClassifier()

model = rf.fit(train)

predictions = model.transform(test)

evaluator = MulticlassClassificationEvaluator(labelCol="label",predictio
nCol="prediction",metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Test set accuracy of RandomForest= " + str(accuracy))
```

```
Test set accuracy of RandomForest= 0.647985989492
```

Summary

Naive Bayes model fails when the input data is very independent. And the classifiers used in this experiment give out poor performance on continuous data.