771219/Models/Te...

Markdown styles

FINISHED

Took 1 sec. Last updated by 771219 at November 14 2018, 1:19:08 PM. (outdated)

FINISHED

Sentiment Analysis example #2

Classify some complaints according to products they relate to (predefined, given as column)

Useful links:

- https://towardsdatascience.com/multi-class-text-classification-with-scikit-learn-12f1e60e0a9f (https://towardsdatascience.com/multi-class-text-classification-with-scikit-learn-12f1e60e0a9f)
- https://towardsdatascience.com/multi-class-text-classification-with-pyspark-7d78d022ed35 (https://towardsdatascience.com/multi-class-text-classification-with-pyspark-7d78d022ed35)

Data:

- A 631MB file: https://catalog.data.gov/dataset/consumer-complaint-database (https://catalog.data.gov/dataset/consumer-complaint-database) with 155,335 complaints

Tools:

- sk-learn

Took 0 sec. Last updated by 771219 at November 14 2018, 3:14:40 PM. (outdated)

READY

0. Set-Up

Keytab

READY

```
Ticket cache: FILE:/tmp/krb5cc_1443748855
Default principal: 776859@BNZNAG.NZ.THENATIONAL.COM
Valid starting       Expires              Service principal
10/31/2018 10:45:21  10/31/2018 20:45:21  krbtgt/BNZNAG.NZ.THENATIONAL.COM@BNZNAG.NZ.THENATIONAL.CO
M
        renew until 11/07/2018 10:45:21
```

Pypark

FINISHED

```
['seaborn-darkgrid', 'Solarize_Light2', 'seaborn-notebook', 'classic', 'seaborn-ticks', 'grayscal
e', 'bmh', 'seaborn-talk', 'dark_background', 'ggplot', 'fivethirtyeight', '_classic_test', 'seabor
n-colorblind', 'seaborn-deep', 'seaborn-whitegrid', 'seaborn', 'seaborn-poster', 'seaborn-bright',
 'seaborn-muted', 'seaborn-paper', 'seaborn-white', 'fast', 'seaborn-pastel', 'seaborn-dark', 'tabl
eau-colorblind10', 'seaborn-dark-palette']
```

Took 1 sec. Last updated by 771219 at November 14 2018, 1:52:49 PM. (outdated)

Python 3

FINISHED

```
['Solarize_Light2', '_classic_test', 'bmh', 'classic', 'dark_background', 'fast', 'fivethirtyeigh
t', 'ggplot', 'grayscale', 'seaborn-bright', 'seaborn-colorblind', 'seaborn-dark-palette', 'seaborn
-dark', 'seaborn-darkgrid', 'seaborn-deep', 'seaborn-muted', 'seaborn-notebook', 'seaborn-paper',
 'seaborn-pastel', 'seaborn-poster', 'seaborn-talk', 'seaborn-ticks', 'seaborn-white', 'seaborn-whi
tegrid', 'seaborn', 'tableau-colorblind10']
```

Took 3 sec. Last updated by 771219 at November 14 2018, 1:52:56 PM. (outdated)

READY

1. Import data

FINISHED

```
%sh
kinit -kt /etc_cloudera/user_home/$USER/$USER.keytab $USER
klist
```

```
Ticket cache: FILE:/tmp/krb5cc_1443739293
Default principal: 771219@BNZNAG.NZ.THENATIONAL.COM
Valid starting       Expires              Service principal
11/14/2018 13:51:37  11/14/2018 23:51:37  krbtgt/BNZNAG.NZ.THENATIONAL.COM@BNZNAG.NZ.THENATIONAL.CO
M
        renew until 11/21/2018 13:51:37
```

Took 1 sec. Last updated by 771219 at November 14 2018, 1:51:37 PM. (outdated)

Load from Hue

FINISHED

```
%pyspark

data_sp = spark.read.format('csv').option('header', 'true').option('mode', 'DROPMALFORMED').load("
```

Took 36 sec. Last updated by 771219 at November 14 2018, 1:52:18 PM. (outdated)

FINISHED

```
%pyspark
cn.show(data_sp.limit(10).toPandas())
```

Date received ▼	Product ▼	Sub-product ▼	Issue
03/12/2014	Mortgage	Other mortgage	Loan modific
01/19/2017	Student loan	Federal student loan servicing	Dealing with

▼	▼	▼	
04/06/2018	Credit card or prepaid card	General-purpose credit card or charge card	Other feature
06/08/2014	Credit card	None	Bankruptcy
09/13/2014	Debt collection	Credit card	Communicati

Took 1 sec. Last updated by 771219 at November 14 2018, 1:53:02 PM. (outdated)

```
%pyspark
data_sp.printSchema()

data_sp\
    .limit(10)\
    .select('Complaint ID', 'Date received', 'Product', 'Sub-product', 'Issue', 'Consumer complain
    .show()
```

FINISHED

```
 |-- Company response to consumer: string (nullable = true)
 |-- Timely response?: string (nullable = true)
 |-- Consumer disputed?: string (nullable = true)
 |-- Complaint ID: string (nullable = true)
+------------+-------------+--------------------+--------------------+--------------------+-----
-----------------------+
|Complaint ID|Date received|             Product|         Sub-product|               Issue|Consu
mer complaint narrative|
+------------+-------------+--------------------+--------------------+--------------------+-----
-----------------------+
|       87065|   05/25/2012|Bank account or s...|Other bank produc...|Account opening, ...|
                      null|
|      902552|   06/19/2014|    Credit reporting|                null|Incorrect informa...|
                      null|
|         361|   12/17/2011|         Credit card|                null|APR or interest rate|
                      null|
|     2416367|   04/03/2017|         Credit card|                null|    Billing disputes|
                      null|
```

Took 13 sec. Last updated by 771219 at November 14 2018, 1:54:41 PM. (outdated)

READY

2. Prepare data

READY

2.1 Clean data

READY

We need to remove the rows with no Customer complaint narrative

```
%pyspark

from pyspark.sql.functions import col, length
```

FINISHED

```
data_sp.where(length(col("Date received")) <> 10)\
    .limit(10)\
    .select('Complaint ID', 'Date received', 'Product', 'Sub-product', 'Issue', 'Consumer complain
    .show()
```

```
+------------+-------------------+-------------------+-------------------+-------------------
-+----------------------------+
|Complaint ID|      Date received|            Product|        Sub-product|               Issu
e|Consumer complaint narrative|
+------------+-------------------+-------------------+-------------------+-------------------
-+----------------------------+
|     2207648|On XXXX XXXX we r...| we had already s...| and this form wa...| we asked how th
e...|         they even charge...|
|     3001896|I have also conta...| which it is not....| even when I have...| and submitted a
 ...|         the entry should...|
|     1524001| Please help me find| where on those l...| says what are th...| 36 and 48. Why
 i...|         this would save ...|
|     1617341|        Subsequently|        on XXXX XXXX| 2015 XXXX I rece...| did the caller
 a...|         and that I would...|
|     1871979|             Equifax | on the other han...| even though I ha...|         the paymen
t|         which has droppe...|
|     2999137|They are not prov...| even if we reach...| they refuse to d...| which is not li
s...|         XXXX ID # XXXX a...|
```

Took 4 sec. Last updated by 771219 at November 14 2018, 1:54:53 PM. (outdated)

READY

We need to have a proper date and cast the Complaint ID

Keep only well formed dates

FINISHED

```
%pyspark

data2_sp = data_sp\
            .filter(col('Consumer complaint narrative').isNotNull())\
            .filter(length(col("Date received")) == 10)
```

Took 0 sec. Last updated by 771219 at November 14 2018, 1:55:26 PM. (outdated)

Check data

FINISHED

```
%pyspark
data2_sp\
    .limit(10)\
    .select('Complaint ID', 'Date received', 'Product', 'Sub-product', 'Issue', 'Consumer complain
    .show()
```

```
+------------+-------------+--------------------+--------------------+--------------------+-----
-----------------------+
|Complaint ID|Date received|             Product|         Sub-product|               Issue|Consu
mer complaint narrative|
+------------+-------------+--------------------+--------------------+--------------------+-----
-----------------------+
|     2796097|   01/28/2018|Credit reporting,...|    Credit reporting|Problem with a cr...|
   i have been dispu...|
|     2500238|   06/02/2017|            Mortgage|Conventional home...|Struggling to pay...|
   The lender has fa...|
|     2140886|   09/30/2016|Bank account or s...|    Checking account|Problems caused b...|
   on XX/XX/2016 ACH...|
|     2343047|   02/14/2017|    Credit reporting|               null|Incorrect informa...|
   "This account was...|
|     2283724|   01/11/2017|            Mortgage|        FHA mortgage|Loan servicing, p...|
   "I made 2 payment...|
|     1636969|   11/03/2015|Bank account or s...|     Savings account|Deposits and with...|
   On at least XXXX    |
```

Took 4 sec. Last updated by 771219 at November 14 2018, 1:55:37 PM. (outdated)

Reformat Date

FINISHED

```
%pyspark
import pyspark.sql.functions as F

data2_sp\
    .select(F.to_date(F.unix_timestamp(col('Date received'), "mm/dd/yyyy").cast("timestamp")).alia
            *list(set(data2_sp.columns)))\
        # F.to_date(F.unix_timestamp(col('Date received'), "dd/mm/yyyy").cast("timestamp")),
        # *list(set(data2_sp.columns)))\
    .select('Complaint ID', 'Date_received', 'Date received', 'Product', 'Sub-product', 'Issue', '
    .limit(100)\
    .show()
```

```
forma...|        This is my Second...|
|     2015698|   2016-01-16|   07/16/2016|     Debt collection|Other (i.e. phone...|False statem
ents ...|        this account is o...|
|     2999866|   2018-01-23|   08/23/2018|Credit reporting,...|    Credit reporting|Incorrect in
forma...|        On XX/XX/2018, I ...|
|     2694471|   2017-01-06|   10/06/2017|Credit card or pr...|   Store credit card|Closing your
 account|        Alleged company u...|
|     2139485|   2016-01-29|   09/29/2016|    Credit reporting|                null|Incorrect in
forma...|        On the credit rep...|
|     2846089|   2018-01-17|   03/17/2018|Vehicle loan or l...|                Loan|Managing the
 loan...|        I Filed a complai...|
|     2597783|   2017-01-07|   08/07/2017|Credit card or pr...|General-purpose c...|Other featur
es, t...|        In late XXXX of 2...|
|     2558021|   2017-01-24|   06/24/2017|Credit reporting,...|    Credit reporting|Incorrect in
forma...|        XXXX XXXX XXXX XX...|
|     2680423|   2017-01-21|   09/21/2017|Money transfer, v...|Domestic (US) mon...|Money was no
t ava...|        I transferred ( W...|
|     2982154|   2018-01-05|   08/05/2018|Credit reporting,...|    Credit reporting|Problem with
```

Took 1 sec. Last updated by 771219 at November 14 2018, 1:55:52 PM. (outdated)

Cast complaintID

FINISHED

```
%pyspark

data3_sp = data2_sp\
    .select(F.to_date(F.unix_timestamp(col('Date received'), "mm/dd/yyyy").cast("timestamp")).alia
            'Complaint ID', 'Product', 'Sub-product', 'Issue', 'Consumer complaint narrative')\
    .withColumn("Complaint ID", col("Complaint ID").cast("integer"))

data3_sp.printSchema()
```

```
root
 |-- Date_Received: date (nullable = true)
 |-- Complaint ID: integer (nullable = true)
 |-- Product: string (nullable = true)
 |-- Sub-product: string (nullable = true)
 |-- Issue: string (nullable = true)
 |-- Consumer complaint narrative: string (nullable = true)
```

Took 0 sec. Last updated by 771219 at November 14 2018, 1:56:21 PM. (outdated)

All good

FINISHED

```
%pyspark
data3_sp.limit(100).show()
```

```
|  2018-01-18|     2966321|Credit reporting,...|    Credit reporting|Problem with a cr...|
   I have been dispu...|
|  2018-01-16|     2817119|     Debt collection|       I do not know|Communication tac...|
   I have gotten 7 c...|
|  2018-01-22|     2998714|     Debt collection|          Other debt|Communication tac...|
   I APPARENTLY HAVE...|
|  2018-01-23|     2854122|Credit reporting,...|    Credit reporting|Incorrect informa...|
   This is my Second...|
|  2016-01-16|     2015698|     Debt collection|Other (i.e. phone...|False statements ...|
   this account is o...|
|  2018-01-23|     2999866|Credit reporting,...|    Credit reporting|Incorrect informa...|
   On XX/XX/2018, I ...|
|  2017-01-06|     2694471|Credit card or pr...|   Store credit card|Closing your account|
   Alleged company u...|
|  2016-01-29|     2139485|    Credit reporting|                null|Incorrect informa...|
   On the credit rep...|
|  2018-01-17|     2846089|Vehicle loan or l...|                Loan|Managing the loan...|
   I Filed a complai...|
```

Took 0 sec. Last updated by 771219 at November 14 2018, 1:56:25 PM. (outdated)

READY

How come that we have 205k complaints after filtering 155K according to Hue?

We will run into memory issues unless using mllib

To Pandas

FINISHED

```
%pyspark

data_pd = data3_sp.toPandas()
data_pd.shape
```

```
(205639, 6)
```

Took 9 sec. Last updated by 771219 at November 14 2018, 1:56:41 PM. (outdated)

READY

We replace the label strings (Product) with a category_id number

We cut at 20,000 (or correlation does not work - could still train & test though)

FINISHED

```
%pyspark
# Select a subset
df         = data_pd[['Product', 'Consumer complaint narrative']][:10]
df.columns = ['Product', 'Consumer_complaint_narrative']
```

```
# Create a numerical target
df['category_id'] = df['Product'].factorize()[0]

df
```

```
                                             Product     ...     category_id
0                                       Student loan     ...               0
1                        Credit card or prepaid card     ...               1
2                                   Credit reporting     ...               2
3  Credit reporting, credit repair services, or o...     ...               3
4  Credit reporting, credit repair services, or o...     ...               3
5                                    Debt collection     ...               4
6                                    Debt collection     ...               4
7  Credit reporting, credit repair services, or o...     ...               3
8                                    Debt collection     ...               4
9  Credit reporting, credit repair services, or o...     ...               3
[10 rows x 3 columns]
```

Took 0 sec. Last updated by 771219 at November 14 2018, 2:12:20 PM. (outdated)

Get features, label, drop duplicates

FINISHED

```
%pyspark

from io import StringIO

# Take 20,000 or we may run out of memory in the correlatio analysis
max_input = 10

# Select a subset
df         = data_pd[['Product', 'Consumer complaint narrative']][:max_input]
df.columns = ['Product', 'Consumer_complaint_narrative']

# Create a numerical target
df['category_id'] = df['Product'].factorize()[0]

# Remove duplicates
category_id_df = df[['Product', 'category_id']].drop_duplicates().sort_values('category_id')
category_to_id = dict(category_id_df.values)

id_to_category = dict(category_id_df[['category_id', 'Product']].values)
cn.show(df.head(5), type='st')
```

	Product	Consumer_complaint_narrative	category_id
0	Student loan	When my loan was switched over to Navient i was never told that i had a deliquint balance because with XXXX i did not. When going to purchase a vehicle i discovered my credit score had been dropped from the XXXX into the XXXX. I have been faithful at paying my student loan. I was told that Navient was the company i had delinquency with. I contacted Navient to resolve this issue you and kept being told to just contact the credit bureaus and expalin the situation and maybe they could help me. I was so angry that i just hurried and paid the balance off and then after tried to dispute the delinquency with the credit bureaus. I have had so much trouble bringing my credit score back up.	0

	Product	Consumer_complaint_narrative	category_id
1	Credit card or prepaid card	I tried to sign up for a spending monitoring program and Capital One will not let me access my account through them	1
2	Credit reporting	My credit score has gone down XXXX points in the last month - from the XXXX 's to the XXXX 's. I requested and reviewed reports from all XXXX credit reporting agencies and I can not find a significant reason for the significant decrease in my score. Please help me.	2
3	Credit reporting, credit repair services, or other personal consumer reports	I few months back I contacted XXXX in regards to fraudulent accounts one being XXXX XXXX. I sent in the necessary documents affidavit, police report. They removed the account now the account has appeared back on my report under XXXX XXXX and XXXX allowed that. Dropping my score tremendously	3
4	Credit reporting, credit repair services, or other personal consumer reports	I have been disputing a Bankruptcy on my credit report i have written to the bureau for the past 3 years i call them on the phone 2 dozen times i wrote to the XXXX district Court about a Bankruptcy that both Experian and XXXX have been reporting in Accurately on my Bureau the court sent me a letter indicating that they DID NOT AND DO NOT report to any Credit Bureaus i sent these letters to the Bureaus and they say that they have verify this as a accurate item reported by the court so I went to the XXXX District court with my ID and they said to me that we don't have a BANKRUPTCY in my name they then told me that the Bureau gets this information from XXXX XXXX i contacted them by phone and in writing they have never responded by phone or mail so i sent a letter to the Bureau explaining that they show me their Method of Verification they responded that the info was verified by XXXX XXXX i asked them then why isn't XXXX XXXX showing on my credit report they said that i should contact them bottom line is they don't give a XXXX about me and my credit and how this effects my life so i have spoken with a attorney he said that i should fit a complaint with you so i am complain that these Bureaus are ignoring the Fair Credit act and have no fear of your organization the attorney general or the federal Trade commission so I am asking for your help in this matter if they have verified this then i have a right to know who they verified this information with NAME EMAIL PHONE NUMBER etc.	3

Took 0 sec. Last updated by 771219 at November 14 2018, 2:12:46 PM. (outdated)

```
%pyspark
```

FINISHED

```
abc=category_to_id.items(),len(category_to_id)
abc
```

```
([(u'Student loan', 0), (u'Debt collection', 4), (u'Credit card or prepaid card', 1), (u'Credit rep
orting', 2), (u'Credit reporting, credit repair services, or other personal consumer reports', 3)],
 5)
```

Took 0 sec. Last updated by 771219 at November 14 2018, 2:12:50 PM. (outdated)

Badly umbalanced classes

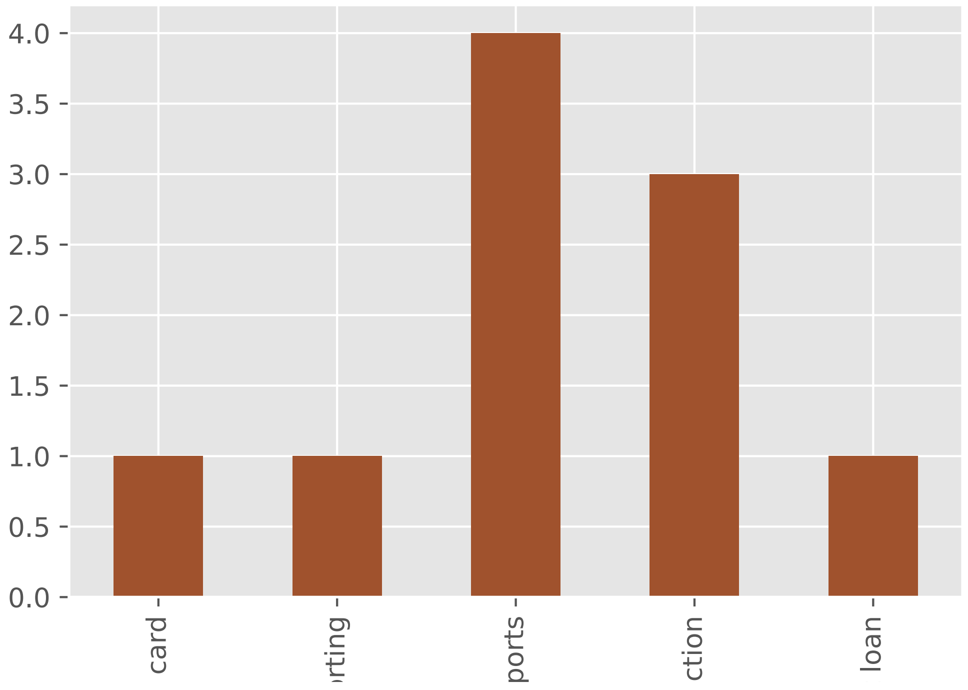
FINISHED

```
%pyspark

fig1 = plt.figure(figsize=(6, 4))
ax1  = fig1.add_subplot(1, 1, 1)

df.groupby('Product').Consumer_complaint_narrative.count().plot.bar(ax=ax1, ylim=0, color='sienna'

cn.show(fig1)
```



Took 0 sec. Last updated by 771219 at November 14 2018, 2:12:54 PM. (outdated)

READY

2.3 Create features and targets sets:

READY

Explore these:

- Remove odd characters and numbers
- Text length as feature
- Stemmer (NLTK)
- Lemmatization (TextBlob)
- Part of Speech (NLTK)
- Sentiment as feature (For sentiment analytics, not topics classification)
    - SentiWordNet

- TextBlob .sentiment

Vectorizer / Transfomer options: BOW, unigrams and bigrams, stop-words, NO stemmer

FINISHED

```
%pyspark

from sklearn.feature_extraction.text import TfidfVectorizer

# TFIDF transformer options
trans_options = {
            "sublinear_tf"      : True,
            "norm"              : 'l2'
            }

# WordCounter options
# We remove numbers with tohen_pattern. See https://stackoverflow.com/questions/45981037/sklearn-t
vect_options = {
            "token_pattern"     : u'(?ui)\\b\\w*[a-z]+\\w*\\b',
            "analyzer"          : 'word',
            "preprocessor"      : cta.preprocessor,
            "min_df"            : 4,
            "encoding"          : 'utf-8',
            "stop_words"        : 'english',
            "lowercase"         : True,
            "ngram_range"       : (1, 2),
            }
```

Took 0 sec. Last updated by 771219 at November 14 2018, 2:13:13 PM. (outdated)

Define Stemmer

FINISHED

```
%pyspark
import nltk.stem
from sklearn.feature_extraction.text    import TfidfTransformer, TfidfVectorizer, CountVectorizer

english_stemmer = nltk.stem.SnowballStemmer('english')

class StemmedTfidfVectorizer(TfidfVectorizer):
    def build_analyzer(self):
        analyzer = super(TfidfVectorizer, self).build_analyzer()

        return lambda doc: (english_stemmer.stem(w) for w in analyzer(doc))
```

Took 0 sec. Last updated by 771219 at November 14 2018, 2:14:27 PM. (outdated)

Define Lemmatization

FINISHED

```
%pyspark
from nltk import word_tokenize
from nltk.stem import WordNetLemmatizer

class LemmaTokenizer(object):
    def __init__(self):
        self.wnl = WordNetLemmatizer()
    def __call__(self, doc):
        return [self.wnl.lemmatize(t) for t in word_tokenize(doc)]
```

Took 0 sec. Last updated by 771219 at November 14 2018, 2:14:24 PM. (outdated)

READY

Add the text length as a possible feature

READY

3 Do classifications

Features:

- TFIDF, Note that we allow bow and bigrams
- Text len (1)
- SentiWorNet (3)
- Part Of SPeech (7)

Features selection

- SelectKBest
- see https://github.com/scikit-learn/scikit-learn/blob/master/doc/modules/feature_selection.rst#id93 (https://github.com/scikit-learn/scikit-learn/blob/master/doc/modules/feature_selection.rst#id93)

Choice of model:

- Good pointers:
  - http://streamhacker.com/2012/11/22/text-classification-sentiment-analysis-nltk-scikitlearn/ (http://streamhacker.com/2012/11/22/text-classification-sentiment-analysis-nltk-scikitlearn/)
  - https://streamhacker.com/2010/06/16/text-classification-sentiment-analysis-eliminate-low-information-features/ (https://streamhacker.com/2010/06/16/text-classification-sentiment-analysis-eliminate-low-information-features/)
- I setlled on SVM with SGD after a bit of experiment
- The code can easily be modified to try other models (BernoulliNB or MultinomialNB)

READY

3.1 Warm-up: Experiment with a quick classifier

- no n-fold
- no steemer or lemmanizer
- no POS or text length

Split sets

FINISHED

```
%pyspark
from sklearn.model_selection import train_test_split

# Split
X_train, X_test, y_train, y_test= train_test_split(df['Consumer_complaint_narrative'], df['Product
```

Took 0 sec. Last updated by 771219 at November 14 2018, 2:19:36 PM. (outdated)

FINISHED

```
%pyspark
from sklearn.model_selection import train_test_split

# Split
X_train, X_test, y_train, y_test= train_test_split(df['Consumer_complaint_narrative'], df['Product
```

Took 0 sec. Last updated by 771219 at November 14 2018, 2:23:11 PM. (outdated)

FINISHED

```
%pyspark


 y_train
# X_test
```

```
9    Credit reporting, credit repair services, or o...
1                          Credit card or prepaid card
6                                      Debt collection
7    Credit reporting, credit repair services, or o...
3    Credit reporting, credit repair services, or o...
0                                         Student loan
5                                      Debt collection
Name: Product, dtype: object
```

Took 0 sec. Last updated by 771219 at November 14 2018, 2:27:33 PM. (outdated)

Train a quick classifier

FINISHED

```
%pyspark

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.naive_bayes import MultinomialNB

# BOW
count_vect          = CountVectorizer(**vect_options)
X_train_counts      = count_vect.fit_transform(X_train)

# TFIDF
tfidf_transformer   = TfidfTransformer(**trans_options)
X_train_tfidf       = tfidf_transformer.fit_transform(X_train_counts)

# Model
clf                 = MultinomialNB().fit(X_train_tfidf, y_train)
```

Took 0 sec. Last updated by 771219 at November 14 2018, 2:27:47 PM. (outdated)

ERROR

```
%pyspark



# X_train_counts.vocabulary()
```

```
Traceback (most recent call last):
  File "/tmp/zeppelin_pyspark-3166545167771040100.py", line 367, in <module>
    raise Exception(traceback.format_exc())
Exception: Traceback (most recent call last):
  File "/tmp/zeppelin_pyspark-3166545167771040100.py", line 360, in <module>
    exec(code, _zcUserQueryNameSpace)
  File "<stdin>", line 1, in <module>
  File "/usr/lib64/python2.7/site-packages/scipy/sparse/base.py", line 647, in __getattr__
    raise AttributeError(attr + " not found")
```

```
AttributeError: collect not found
```

Took 0 sec. Last updated by 771219 at November 14 2018, 2:34:02 PM. (outdated)

Make a quick prediction

READY

```
%pyspark

cnr = "This company refuses to provide me verification and validation of debt per my right under t

print(clf.predict(count_vect.transform([cnr])))
```

```
[u'Debt collection']
```

READY

```
%pyspark

df[df['Consumer_complaint_narrative'] == cnr]
```

```
             Product     ...     category_id
702  Debt collection     ...               4
[1 rows x 3 columns]
```

READY

```
%pyspark

cnr = "I am disputing the inaccurate information the Chex-Systems has on my credit report. I initi
    Systems only deleted the items that I mentioned in the letter and not all the items that were
    wanted me to say word for word to them what items were fraudulent. The total disregard of the
    fraudulent. If they just had paid a little closer attention to the police report I would not b
    once again. I would like the reported information to be removed : XXXX XXXX XXXX"

print(clf.predict(count_vect.transform([cnr])))
```

```
[u'Credit reporting, credit repair services, or other personal consumer reports']
```

READY

```
%pyspark

df[df['Consumer_complaint_narrative'] == cnr]
```

```
              Product     ...     category_id
745  Credit reporting     ...               2
[1 rows x 3 columns]
```

Check hit rate on Training subset

READY

```
%pyspark

pred_train = clf.predict(count_vect.transform(X_train[:1000]))

np.mean(pred_train == y_train[:1000])
```

```
0.662
```

Check hit rate on Testing subset

READY

```
%pyspark

pred_test = clf.predict(count_vect.transform(X_test[:1000]))

np.mean(pred_test == y_test[:1000])
```

```
0.62
```

READY

Rather mild performance: we need to do better than that

READY

3.2 Model Selection using n-folds

We are not using the lingistic features as they slow us down

Preparation

READY

```
%pyspark

from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import LinearSVC
from sklearn.model_selection import cross_val_score

# NB Folds
CV      = 5

tfidf_transformer  = TfidfVectorizer(**cio.merge2Dicts(trans_options, vect_options))         # Pla
#tfidf_transformer  = StemmedTfidfVectorizer(**cio.merge2Dicts(trans_options, vect_options))  # St

## Tfidf pipeline
tfidf_stats         = Pipeline([('tfidf',   tfidf_transformer)])

## Text Len feature
def get_text_length(x):
    return np.array([len(t) for t in x]).reshape(-1, 1)

len_stats            = Pipeline([('count',   FunctionTransformer(get_text_length, validate=False))

featpipe = FeatureUnion([
        ('text',   tfidf_stats),
#        ('length', len_stats),
        ])

## Featres filter
model_filter        = SelectKBest(chi2, k=nbfeats)

## Models
models = [
    RandomForestClassifier(n_estimators=600, max_depth=8, random_state=0),  # Check parametrisatio
    LinearSVC(),
    MultinomialNB(),
    LogisticRegression(random_state=0),
    ]
```

This can take up to 12min - be patient

READY

```
%pyspark

## Prepare output
cv_df   = pd.DataFrame(index=range(CV * len(models)))
entries = []

# Whether you want to run this or not
doCVModels = True

if doCVModels:
    # Loop over candidate models
    for model in models:
        model_name = model.__class__.__name__
        print("Doing: {}".format(model_name))

        # A fairly simple Pipeline
        text_clf = Pipeline([
            ('allf',   featpipe),   # Feature preparation
            ('chi2',   model_filter),
            ('clf',    model),
            ])

        accuracies = cross_val_score(estimator=text_clf, X=df['Consumer_complaint_narrative'], y=d

        for fold_idx, accuracy in enumerate(accuracies):
            entries.append((model_name, fold_idx, accuracy))

    cv_df = pd.DataFrame(entries, columns=['model_name', 'fold_idx', 'accuracy'])

    # Save to pickle
    pk.dump(cv_df, open(modelpath + 'TextAnalysis_Ex2_cv_df.data', "w"))
else:
    pass
    # Just read from pickle file
    cv_df = pk.load(open(modelpath + 'TextAnalysis_Ex2_cv_df.data', "r"))
```

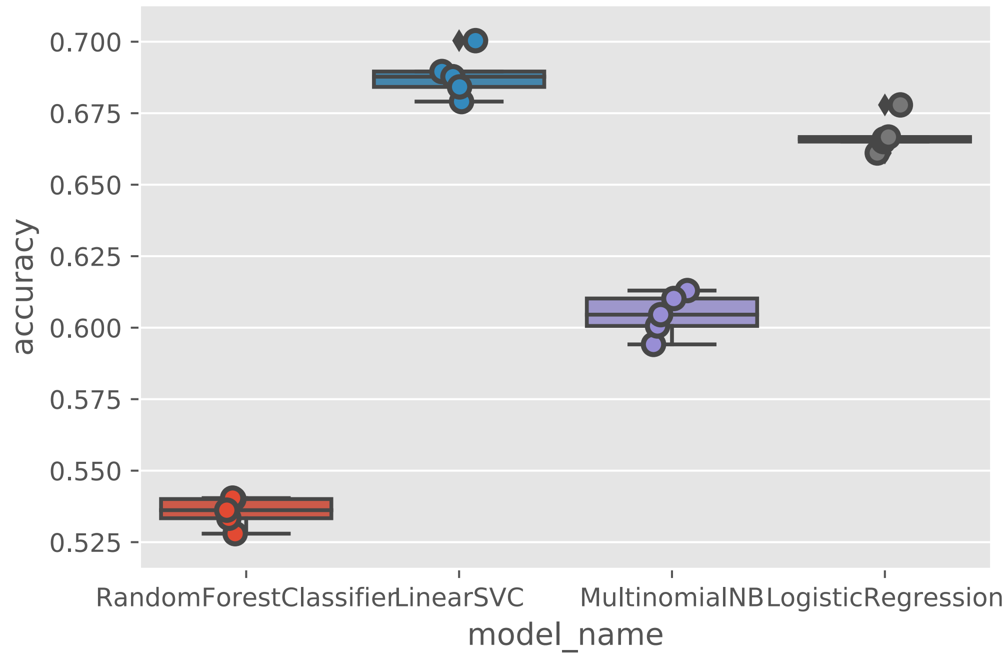
```
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:  2.8min remaining:  4.2min
[CV] .......................... , score=0.594155844156, total= 2.8min
[CV] .......................... , score=0.610207655742, total= 2.8min
[CV] .......................... , score=0.604557976459, total= 2.8min
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:  2.9min finished
Doing: LogisticRegression
[CV]  ................................................................
[CV]  ................................................................
[CV]  ................................................................
[CV]  ................................................................
[CV]  ................................................................
[CV] .......................... , score=0.665999499625, total= 2.7min
[CV] .......................... , score=0.666666666667, total= 2.7min
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:  2.8min remaining:  4.2min
[CV] .......................... , score=0.677902621723, total= 2.8min
[CV] .......................... , score=0.665083729068, total= 2.8min
[CV] .......................... , score=0.661088911089, total= 2.8min
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:  2.9min finished
```

READY

```
%pyspark

fig1, ax1 = plt.subplots(figsize=(6, 4))

sns.boxplot(ax=ax1, x='model_name', y='accuracy', data=cv_df)
sns.stripplot(ax=ax1, x='model_name', y='accuracy', data=cv_df, size=8, jitter=True, edgecolor="gr
cn.show(fig1, align='l')
```



READY

```
%pyspark

cv_df.groupby('model_name').accuracy.mean()
```

```
model_name
LinearSVC                 0.688197
LogisticRegression        0.667348
MultinomialNB             0.604501
RandomForestClassifier    0.535601
Name: accuracy, dtype: float64
```

READY

LinearSVC and LogisticRegression are the best ones

Note: no len feature, no linguistic feature

READY

3.3 Selected model with better features

- steemer or lemmanizer
- POS and text length
- no n-fold

NB Feats

READY

```
%pyspark

nbfeats  = 4000   # Experiment with this -> 4,000 for a large corpus seems reasonnable, but check
myoffset = 11     # Offset from end for purely tfidf features
```

Subsets (if required)

READY

```
%pyspark

tr_subset = 10000
ts_subset = 1000
```

Get features and labels - Standard, Stemmer or Lemmization

READY

```
%pyspark

from sklearn.pipeline import Pipeline, FeatureUnion
from sklearn.multiclass import OneVsRestClassifier
from sklearn.preprocessing import FunctionTransformer

##
## TFIDF features
##
## Can be customized for Lemmatization, Stemmer or plain

# Lemmatization
count_vect      = CountVectorizer(tokenizer=LemmaTokenizer(), **vect_options)

# Transformer: Switch for Stemmer
tfidf_transformer   = TfidfTransformer(**trans_options)                                       # 2n
#tfidf_transformer  = TfidfVectorizer(**cio.merge2Dicts(trans_options, vect_options))         # Pl
#tfidf_transformer  = StemmedTfidfVectorizer(**cio.merge2Dicts(trans_options, vect_options))  # St

# Full tfidf pipeline
tfidf_stats         = Pipeline([
                                ('vect',    count_vect),
                                ('tfidf',   tfidf_transformer),
                                ])

##
## Text Len feature
##
def get_text_length(x):
    return np.array([len(t) for t in x]).reshape(-1, 1)

len_stats            = Pipeline([
                                ('count',   FunctionTransformer(get_text_length, validate=False))
                                ])

##
## Part-of-speech + Sentiment features
##
ling_stats          = Pipeline([
                                ('ta',      cta.LinguisticVectorizer())
                                ])

##
## Al in one pipeline
##

featpipe = FeatureUnion([
        ('text',   tfidf_stats),
        ('length', len_stats),
        ('ling',   ling_stats),
        ])
```

Create pipeline with features filtering and model

READY

```
%pyspark

from sklearn.pipeline                   import Pipeline
from sklearn.feature_selection          import SelectKBest, chi2
from sklearn.naive_bayes                import MultinomialNB, BernoulliNB
from sklearn.linear_model               import SGDClassifier, LogisticRegression
from sklearn.svm                        import NuSVC

# Choice of model
model_choice    = LogisticRegression(random_state=0)
#model_choice   = SGDClassifier(loss='hinge', penalty='l2') # Relying on good sci-kit defaults!
#model_choice   = BernoulliNB()       # Unigrams mostly
#model_choice   = MultinomialNB()     # Bigrams mostly
#model_choice   = NuSVC(gamma='scale')


# Filter: main tfidf features (based on correlation)
model_filter        = SelectKBest(chi2, k=nbfeats)

# A fairly simple Pipeline
text_clf = Pipeline([
            ('allf',   featpipe),   # Feature preparation
            ('chi2',   model_filter),
            ('clf',    model_choice),
            ])
```

Split sets

READY

```
%pyspark

X_train, X_test, y_train, y_test, indices_train, indices_test = train_test_split(df['Consumer_comp
```

Performance on Training

READY

```
%pyspark

myclf       = text_clf.fit(X_train[:tr_subset], y_train[:tr_subset])
pred_train  = text_clf.predict(X_train[:tr_subset])

np.mean(pred_train == y_train[:tr_subset])
```

```
0.6961
```

Performance on Testing

READY

```
%pyspark

pred_test = text_clf.predict(X_test[:ts_subset])

np.mean(pred_test == y_test[:ts_subset])
```

```
0.673
```

READY

```
%pyspark

from sklearn.metrics import confusion_matrix

conf_mat = confusion_matrix(y_test[:ts_subset], pred_test)
```

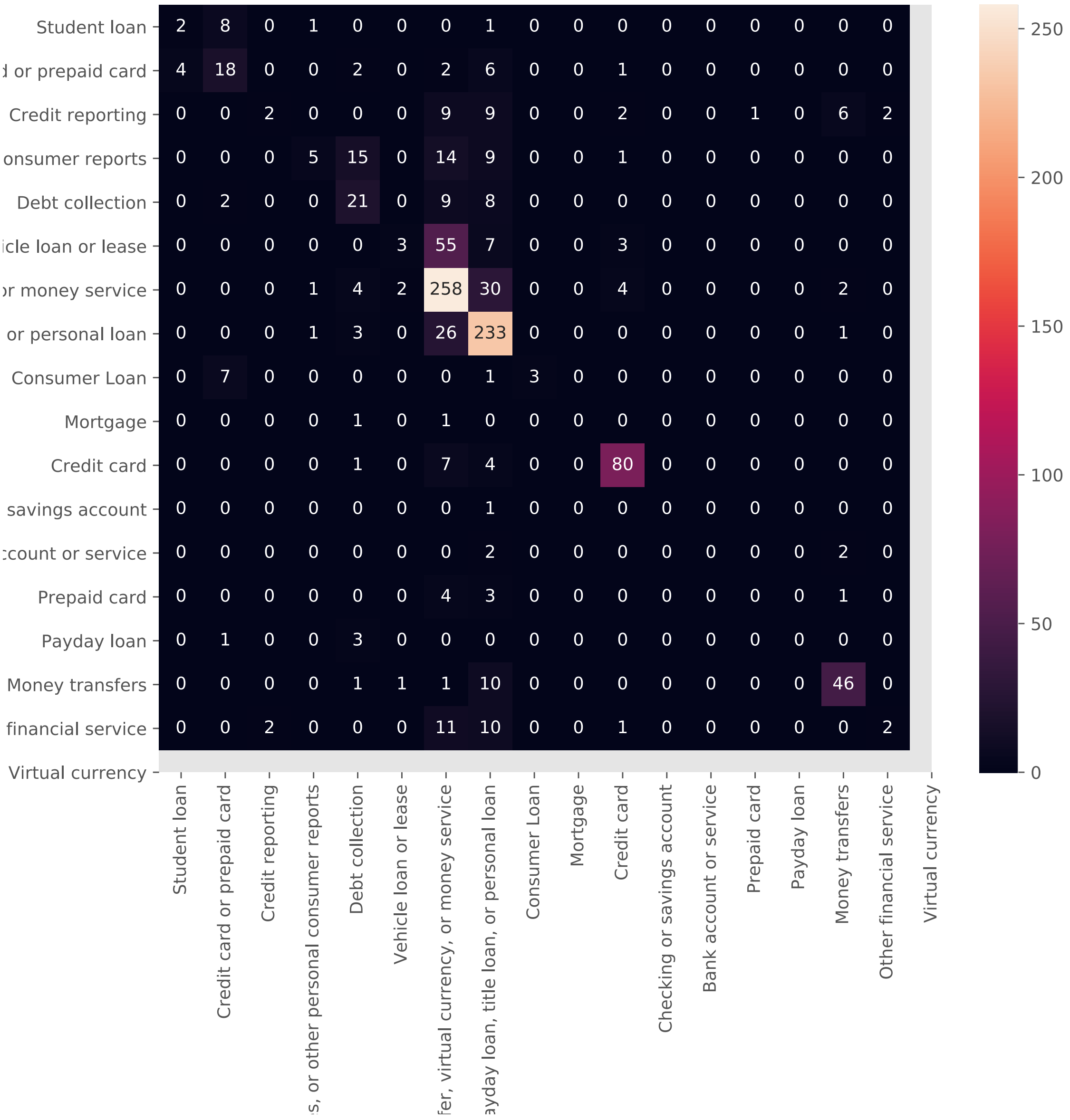
```
fig2, ax2 = plt.subplots(figsize=(10, 8))

import seaborn as sns

sns.heatmap(conf_mat,
            ax=ax2,
            annot=True,
            fmt='d',
            xticklabels=category_id_df.Product.values,
            yticklabels=category_id_df.Product.values,
            )

# Not working
ax2.set(xlabel='common xlabel', ylabel='common ylabel')

cn.show(fig2)
```



READY

Issues:

- We should rebalance the classes: support is stoo small for some categories
- Some label missing on Testing (especially if using subset)

Get classification report

READY

```
%pyspark

from sklearn import metrics

res = metrics.classification_report(y_test[:ts_subset], pred_test, target_names=df['Product'].uniq

print(res)
```

```
/usr/lib64/python2.7/site-packages/sklearn/metrics/classification.py:1428: UserWarning: labels s
ize, 17, does not match size of target_names, 18
  .format(len(labels), len(target_names))
/usr/lib64/python2.7/site-packages/sklearn/metrics/classification.py:1135: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted sample
s.
  'precision', 'predicted', average, warn_for)
                                                                        precision    recal
l  f1-score   support
                                                           Student loan       0.33      0.1
7      0.22        12
                                               Credit card or prepaid card       0.50      0.5
5      0.52        33
                                                      Credit reporting       0.50      0.0
6      0.11        31
Credit reporting, credit repair services, or other personal consumer reports       0.62      0.1
1      0.19        44
                                                       Debt collection       0.41      0.5
```

READY

No precision/recall as not a binary classification problem

READY

4 Audit features

READY

4.1 Check features correlation

We are explicitly recreating the features - before Chi2 selection

Takes a few minutes - otherwise done in the pipeline

READY

```
%pyspark
```

```
# There must be a better way to do that, as this was effectiovely done in the pipeline

allfeatures    = featpipe.fit_transform(X_train[:tr_subset]).toarray()
tfidffeatures  = tfidf_stats.fit_transform(X_train[:tr_subset]).toarray()

# Check dimensions
(allfeatures.shape, tfidffeatures.shape)
```

```
((10000, 23518), (10000, 23507))
```

READY

```
%pyspark

count_vect.get_feature_names()[:100]
```

```
['aadvantage', 'aargon', u'aargon agency', 'ab', 'abandoned', 'abide', 'abiding', 'ability', u'abil
ity obtain', u'ability open', u'ability pay', u'ability purchase', u'ability recovery', u'ability s
ecure', u'ability use', 'able', u'able access', u'able afford', u'able assist', u'able buy', u'able
 contact', u'able continue', u'able credit', u'able determine', u'able dispute', u'able fix', u'abl
e help', u'able home', u'able information', u'able loan', u'able locate', u'able make', u'able obta
in', u'able open', u'able pay', u'able payment', u'able prove', u'able provide', u'able pull', u'ab
le qualify', u'able reach', u'able receive', u'able remove', u'able repay', u'able report', u'able
 resolve', u'able send', u'able set', u'able speak', u'able talk', u'able tell', u'able use', u'abl
e verify', u'able work', 'abruptly', 'absent', u'absent proof', 'absolute', 'absolutely', u'absolut
ely ridiculous', 'absurd', 'abuse', 'abused', 'abusing', 'abusive', 'ac', 'acc', 'accelerate', 'acc
elerated', 'acceleration', 'accent', 'accept', u'accept offer', u'accept payment', 'acceptable', 'a
cceptance', u'acceptance corporation', u'acceptance wa', 'accepted', u'accepted payment', 'acceptin
g', u'accepting payment', 'accepts', 'access', u'access account', u'access blocked', u'access credi
t', u'access equifax', u'access file', u'access fund', u'access information', u'access medical',
 u'access money', u'access online', u'access report', u'access subchapter', u'access wa', u'access
 website', 'accessed', 'accessible']
```

Check features most correlated to labels

READY

```
%pyspark

from sklearn.feature_selection import chi2

nlevel    = 10
mylist1   = []
mylist2   = []

# Labels
labels          = np.unique(y_train[:ts_subset])

for label in labels:
  features_chi2 = chi2(tfidffeatures, y_train[:tr_subset] == label)

  indices       = np.argsort(features_chi2[0])

  feature_names = np.array(count_vect.get_feature_names())[indices]
  pvals         = features_chi2[1][indices]

  unigrams      = [v for v in feature_names if len(v.split(' ')) == 1]
  bigrams       = [v for v in feature_names if len(v.split(' ')) == 2]

  # Most correlated
  mylist1.append(
        pd.DataFrame(
            [[
            label,
            "{}".format('\n.'.join(unigrams[-nlevel:])),
            "{}".format('\n.'.join(bigrams[-nlevel:])),
            pvals[-nlevel:]
            ]])
        )
```

```
    # Least correlated
    mylist2.append(
          pd.DataFrame(
              [[
              label,
              "{}".format('\n.'.join(unigrams[:nlevel])),
              "{}".format('\n.'.join(bigrams[:nlevel])),
              pvals[:nlevel]
              ]])
          )

# One pass
topcors1         = pd.concat(mylist1, ignore_index=True)
topcors1.columns = ["label", "MC unigrams", "MC bigrams", "PVals"]

botcors1         = pd.concat(mylist2, ignore_index=True)
botcors1.columns = ["label", "MC unigrams", "MC bigrams", "PVals"]
```

Top correlation -> should make sense

READY

```
%pyspark

cn.show(topcors1, type='st', fs=90)
```

	label	MC unigrams	MC bigrams	PVals
0	Bank account or service	teller .promotion .citigold .checking .check .deposited .branch .overdraft .bank .deposit	charging overdraft .citi gold .pnc bank .account citibank .account cash .debit card .fargo bank .direct deposit .overdraft fee .checking account	[2.84780428e-10 2.79910993e-10 2.40968129e-11 8.34434922e-12 6.67995035e-12 3.06421051e-12 7.28313273e-13 2.03404023e-16 1.93540554e-16 1.81865614e-21]
1	Bank account or service	accessing .destroyed .fraudulent .retained .dealing .obvious .fear .lied .single .valid	proper documentation .requested copy .wa file .regarding account .account social .called stated .told contact .removed wa .did use .received card	[0.99995459 0.99969002 0.9996888 0.99924673 0.99922965 0.99922583 0.9991713 0.99910336 0.99892159 0.99887671]
2	Checking or savings account	branch .saving .deposited .checking .transaction .deposit .atm .fund .bank .overdraft	td bank .bank america .account bank .account chase .external application .deposited check .charged overdraft .saving account .checking account .overdraft fee	[2.45864411e-15 1.96690023e-15 4.71324080e-16 2.90210713e-16 3.44593390e-18 3.76625093e-19 1.10861595e-19 1.14288654e-20 1.04654466e-20 4.61216389e-29]
3	Checking or savings account	cooperate .sitting .travel .additional .owes .pa .unlawful .federal .maximum .supposed	action taken .bank credit .told need .letter chase .reported fraud .payment going .hour day .paper work .representative phone .used card	[0.99988406 0.99981425 0.99977922 0.99976564 0.99974025 0.99972535 0.99972155 0.99969876 0.99969581 0.99962881]
4	Consumer Loan	regional .repossessed .truck .dealership .nissan .santander .honda .ally .car .vehicle	car loan .payment car .consumer usa .car wa .car having .toyota financial .ally financial .santander consumer .loan santander .truck loan	[1.59817968e-07 1.07429205e-07 3.71277090e-08 5.57463843e-09 4.36009810e-10 9.97064917e-12 2.54706690e-12 5.01673725e-16 2.49196496e-16 1.14798447e-17]

	label	MC unigrams	MC bigrams	PVals
5	Consumer Loan	conducted .held .random .company .worked .touch .reasonable .essentially .decided .settled	getting credit .noticed wa .ha sent .couple day .late day .year paid .correct address .debt just .fair debt .applied loan	[0.99994793 0.99977671 0.99976648 0.99950936 0.99907476 0.9988021 0.99877297 0.99866632 0.99857468 0.99838282]
6	Credit card	merchandise .signup .express .lowes .merchant .amazon .citi .macy .synchrony .card	cancelled card .canceled card .care credit .buy credit .cancel card .american express .annual fee .synchrony bank .best buy .credit card	[2.30045103e-07 1.81807511e-07 1.53593807e-08 1.43479541e-08 7.72679501e-10 2.13829376e-10 2.13668383e-11 8.51997655e-12 4.57497478e-14 1.35004969e-22]
7	Credit card	f .fax .woman .giving .reversed .incident .ct .deliver .obligated .partial	credit union .said payment .did include .ago wa .asked speak .wait day .saying wa .account status .told pay .wa filed	[0.99983259 0.99933342 0.99909975 0.99908289 0.99908127 0.99863566 0.9985073 0.99838836 0.99836792 0.99798209]
8	Credit card or prepaid card	mastercard .platinum .minimum .charge .capital .american .purchase .express .reward .card	completed research .research determined .established verifying .compliant removing .ha non .removing unverified .verifying evidence .evidence confirm .american express .credit card	[1.12817003e-09 1.12817003e-09 1.12817003e-09 1.12817003e-09 1.12817003e-09 1.10890464e-10 2.58730105e-11 3.82504126e-13 2.20421504e-18 8.91637999e-32]
9	Credit card or prepaid card	follow .adjust .distress .awful .suffer .joke .tracking .conference .violated .defined	owe balance .account hold .notice letter .number changed .time correct .said n .report negatively .information did .proof company .called representative	[0.99972104 0.99955139 0.99948369 0.99946133 0.99939878 0.99927856 0.99923864 0.99917547 0.99904769 0.99902513]
10	Credit reporting	delete .reinserted .judgement .disputed .annualcreditreport .trans .report .transunion .experian .equifax	disputed resolve .possible thank .promptly delete .credit report .required promptly .trans union .manner soon .resolve manner .unauthorized fraudulent .experian ha	[1.70869326e-05 1.65898461e-05 2.82189847e-06 2.82189847e-06 1.64370516e-06 2.46241010e-07 1.83638604e-07 3.40983355e-09 4.17810225e-14 1.79138969e-14]
11	Credit reporting	dealt .mailed .status .supposed .tenant .invoice .measure .discrepancy .recommended .included	representative phone .approved credit .wa low .high credit .account causing .did account .payment current .loan forgiven .failed provide .judgement wa	[0.99962706 0.99959234 0.99954296 0.99953539 0.99948625 0.99948438 0.99940449 0.99938283 0.9992756 0.99925899]

	label	MC unigrams	MC bigrams	PVals
12	Credit reporting, credit repair services, or other personal consumer reports	remove .inquires .bureau .experian .transunion .credit .reporting .report .equifax .inquiry	mistake appear .report understanding .credit inquiry .reporting agency .identity theft .credit file .credit bureau .inquiry credit .hard inquiry .credit report	[3.89786583e-09 8.36515878e-10 7.00291363e-10 3.62513726e-10 3.54619166e-10 6.28775598e-11 2.86471526e-11 2.64199599e-15 2.86513358e-17 3.71190296e-28]
13	Credit reporting, credit repair services, or other personal consumer reports	contract .protecting .pertinent .emotionally .okay .wear .befor .ding .numerous .mon	related account .victim identify .know happened .loan past .information company .request proof .point ha .informed representative .scam help .refused acknowledge	[0.99999522 0.99999281 0.99998402 0.99996839 0.9999563 0.99992188 0.99988381 0.99983992 0.99978986 0.99977892]
14	Debt collection	loan .medical .validation .calling .owe .collector .recovery .collect .collection .debt	stop calling .owe debt .alleged debt .portfolio recovery .trying collect .debt collection .debt collector .debt wa .collect debt .collection agency	[1.15815899e-11 1.12808632e-11 8.23835997e-12 1.42342743e-12 6.25531628e-13 4.13375959e-14 1.59967253e-14 8.46509556e-17 6.68354027e-27 4.65272161e-56]
15	Debt collection	printing .lengthy .eye .pointless .resume .patience .register .recognized .dmv .separated	approval letter .t loan .bureau credit .number request .called just .score negatively .party credit .wa did .fine wa .stated sent	[0.99999333 0.99994284 0.99994111 0.9998762 0.99985254 0.99979762 0.99979025 0.99977753 0.99975857 0.99965208]
16	Money transfer, virtual currency, or money service	coin .usd .paypal .currency .moneygram .wire .transfer .cryptocurrency .bitcoin .coinbase	transaction day .buy sell .transfer bank .customer support .wire transfer .money coinbase .support ticket .coinbase com .account coinbase .coinbase account	[1.23276033e-015 9.57992440e-016 1.48856995e-016 1.04946557e-018 1.26069635e-022 6.70655525e-026 6.70326232e-031 8.22937257e-033 1.80944317e-058 9.76603581e-163]
17	Money transfer, virtual currency, or money service	bought .old .match .reverse .cause .posted .identification .meet .country .directly	paid month .called number .mark credit .loan did .trade commission .account time .month later .supervisor wa .wa completed .told account	[0.99993569 0.99992095 0.99953017 0.99942124 0.99925782 0.99916608 0.99913447 0.99825411 0.99814991 0.99735049]
18	Money transfers	receiver .shipping .manipulated .ph .park .cardmember .recipient .western .gram .paypal	saying money .fraud want .cancel transaction .money deposited .using paypal .send money .money transfer .money paypal .western union .money gram	[2.34117144e-13 1.77210293e-16 2.65211708e-17 1.77949712e-18 8.98505201e-20 3.48911000e-22 1.76231617e-24 3.72440459e-38 3.72440459e-38 3.23806508e-49]
19	Money transfers	claiming .billing .came .damage .happened .tax .investigation .caused .did .account	wa paid .wa wa .company wa .wa charged .account ha .appropriate proof .apply consumer .apply check .company shall .company provision	[0.99889819 0.9985256 0.99851285 0.99842131 0.99820066 0.99704121 0.99680317 0.99679875 0.99550659 0.9954255]

	label	MC unigrams	MC bigrams	PVals
20	Mortgage	house .nationstar .sale .home .ocwen .property .foreclosure .escrow .modification .mortgage	nationstar mortgage .foreclosure sale .mortgage wa .property tax .sale date .short sale .escrow account .mortgage payment .mortgage company .loan modification	[5.31252172e-21 2.65543367e-23 1.54591931e-23 6.12072313e-24 7.03030320e-28 1.26876455e-28 1.14593347e-39 1.39716686e-42 5.45052580e-54 1.87579643e-89]
21	Mortgage	click .goal .mobile .age .targeted .protect .rip .additionally .odd .accruing	went online .thing happen .available loan .u year .correct problem .sent received .pay check .contacted told .ask question .law regulation	[0.99991539 0.99982952 0.99980736 0.99977306 0.99976337 0.99975924 0.99966628 0.99964352 0.99964347 0.99963217]
22	Payday loan	payed .big .threating .cash .indiana .speedy .picture .percentage .ace .payday	speedy cash .loan ace .percentage rate .ace cash .cash express .finance charge .took payday .picture loan .payday loan .big picture	[3.90498770e-15 5.02805968e-18 4.76486889e-19 2.48288218e-21 1.95068666e-21 3.86756184e-22 8.17419633e-25 7.86080879e-31 3.31877264e-31 8.42185979e-32]
23	Payday loan	ended .month .resolve .afford .refund .statement .recorded .ocwen .didnt .number	did authorize .wa paid .information credit .block ha .notified promptly .agency subsequent .manner consumer .section p .exception resellers .exception verification	[0.99938199 0.99873821 0.99840028 0.99773663 0.99739746 0.99660654 0.99578609 0.99379129 0.99233652 0.99203759]
24	Payday loan, title loan, or personal loan	cashnet .lending .club .georgia .loan .payday .usury .borrowed .pls .mobiloans	state georgia .make loan .got loan .wa easy .code state .took loan .lending club .illegal state .usury law .loan state	[1.46116044e-07 1.23089860e-07 4.86646277e-08 4.07796487e-09 3.60543926e-09 1.90380288e-09 2.61210279e-10 1.28168016e-10 1.61062554e-11 2.05445051e-12]
25	Payday loan, title loan, or personal loan	single .poor .surprise .chapter .yesterday .form .trade .lawsuit .promptly .led	past month .account reported .pulled credit .synchrony bank .know wa .debit card .card payment .payment history .called time .copy letter	[0.99998327 0.99901227 0.99897172 0.99887595 0.99871123 0.9986854 0.9977833 0.99768195 0.99764112 0.99729796]
26	Prepaid card	pst .expedited .moneygram .access .ur .trailer .gift .rescind .prepaid .rushcard	express serve .receive loan .came went .rush card .locked account .pre paid .able reach .prepaid card .t access .access fund	[3.47841266e-010 7.24809207e-012 2.01057255e-013 6.30313950e-014 4.14202641e-015 1.05522745e-023 1.31360844e-025 1.44457490e-030 9.37681765e-044 3.84941059e-150]
27	Prepaid card	calling .mailed .authorize .document .set .practice .close .told .cell .respond	account account .did n .subsequent use .agency rescinds .issue authorization .identifies consumer .requiring consumer .b time .authority decline .identified information	[0.99825784 0.9975646 0.99556022 0.9951412 0.994963 0.99470984 0.99423171 0.99391009 0.99262117 0.99214283]

	label	MC unigrams	MC bigrams	PVals
28	Student loan	college .mae .private .forgiveness .forbearance .repayment .school .loan .student .navient	called navient .income driven .private student .income based .federal loan .loan navient .sallie mae .private loan .loan forgiveness .student loan	[3.46042517e-020 5.34019506e-021 8.39228307e-023 1.26236486e-028 1.01294242e-029 1.61051573e-033 9.22188552e-035 5.20010906e-055 5.43761857e-058 3.91078462e-103]
29	Student loan	slow .want .tomorrow .dozen .defamation .xxxxi .complain .box .record .word	got information .issue received .company make .stating needed .statement balance .able receive .record credit .fee pay .representative called .thank time	[0.99978811 0.99975863 0.99960032 0.99951965 0.99946714 0.99938097 0.99936345 0.99932826 0.99927676 0.99917413]
30	Vehicle loan or lease	toyota .gap .acceptance .gm .santander .westlake .repo .warranty .vehicle .car	gm financial .vehicle wa .sold auction .car worth .car payment .paid like .gap insurance .car wa .warranty wa .credit acceptance	[4.77357213e-11 4.66862121e-11 3.99271722e-11 2.78220009e-12 3.81111428e-13 2.67533894e-13 1.60026815e-15 1.05997893e-16 8.31189807e-19 2.11509595e-32]
31	Vehicle loan or lease	hospital .correspondence .informed .ordered .rectify .area .reach .hr .attaching .mailed	remove late .letter called .company trying .wa talking .called ask .attempted contact .owe money .late credit .payment plan .day passed	[0.99988348 0.99984562 0.99983023 0.99953466 0.9995081 0.99940901 0.99933308 0.9991575 0.99907062 0.99891947]

Bottom correlation -> should be neutral

READY

```
%pyspark
cn.show(botcors1, type='st', fs=90)
```

	label	MC unigrams	MC bigrams	PVals
0	Bank account or service	teller .promotion .citigold .checking .check .deposited .branch .overdraft .bank .deposit	charging overdraft .citi gold .pnc bank .account citibank .account cash .debit card .fargo bank .direct deposit .overdraft fee .checking account	[2.84780428e-10 2.79910993e-10 2.40968129e-11 8.34434922e-12 6.67995035e-12 3.06421051e-12 7.28313273e-13 2.03404023e-16 1.93540554e-16 1.81865614e-21]
1	Bank account or service	accessing .destroyed .fraudulent .retained .dealing .obvious .fear .lied .single .valid	proper documentation .requested copy .wa file .regarding account .account social .called stated .told contact .removed wa .did use .received card	[0.99995459 0.99969002 0.9996888 0.99924673 0.99922965 0.99922583 0.9991713 0.99910336 0.99892159 0.99887671]
2	Checking or savings account	branch .saving .deposited .checking .transaction .deposit .atm .fund .bank .overdraft	td bank .bank america .account bank .account chase .external application .deposited check .charged overdraft .saving account .checking account .overdraft fee	[2.45864411e-15 1.96690023e-15 4.71324080e-16 2.90210713e-16 3.44593390e-18 3.76625093e-19 1.10861595e-19 1.14288654e-20 1.04654466e-20 4.61216389e-29]

	label	MC unigrams	MC bigrams	PVals
3	Checking or savings account	cooperate .sitting .travel .additional .owes .pa .unlawful .federal .maximum .supposed	action taken .bank credit .told need .letter chase .reported fraud .payment going .hour day .paper work .representative phone .used card	[0.99988406 0.99981425 0.99977922 0.99976564 0.99974025 0.99972535 0.99972155 0.99969876 0.99969581 0.99962881]
4	Consumer Loan	regional .repossessed .truck .dealership .nissan .santander .honda .ally .car .vehicle	car loan .payment car .consumer usa .car wa .car having .toyota financial .ally financial .santander consumer .loan santander .truck loan	[1.59817968e-07 1.07429205e-07 3.71277090e-08 5.57463843e-09 4.36009810e-10 9.97064917e-12 2.54706690e-12 5.01673725e-16 2.49196496e-16 1.14798447e-17]
5	Consumer Loan	conducted .held .random .company .worked .touch .reasonable .essentially .decided .settled	getting credit .noticed wa .ha sent .couple day .late day .year paid .correct address .debt just .fair debt .applied loan	[0.99994793 0.99977671 0.99976648 0.99950936 0.99907476 0.9988021 0.99877297 0.99866632 0.99857468 0.99838282]
6	Credit card	merchandise .signup .express .lowes .merchant .amazon .citi .macy .synchrony .card	cancelled card .canceled card .care credit .buy credit .cancel card .american express .annual fee .synchrony bank .best buy .credit card	[2.30045103e-07 1.81807511e-07 1.53593807e-08 1.43479541e-08 7.72679501e-10 2.13829376e-10 2.13668383e-11 8.51997655e-12 4.57497478e-14 1.35004969e-22]
7	Credit card	f .fax .woman .giving .reversed .incident .ct .deliver .obligated .partial	credit union .said payment .did include .ago wa .asked speak .wait day .saying wa .account status .told pay .wa filed	[0.99983259 0.99933342 0.99909975 0.99908289 0.99908127 0.99863566 0.9985073 0.99838836 0.99836792 0.99798209]
8	Credit card or prepaid card	mastercard .platinum .minimum .charge .capital .american .purchase .express .reward .card	completed research .research determined .established verifying .compliant removing .ha non .removing unverified .verifying evidence .evidence confirm .american express .credit card	[1.12817003e-09 1.12817003e-09 1.12817003e-09 1.12817003e-09 1.12817003e-09 1.10890464e-10 2.58730105e-11 3.82504126e-13 2.20421504e-18 8.91637999e-32]
9	Credit card or prepaid card	follow .adjust .distress .awful .suffer .joke .tracking .conference .violated .defined	owe balance .account hold .notice letter .number changed .time correct .said n .report negatively .information did .proof company .called representative	[0.99972104 0.99955139 0.99948369 0.99946133 0.99939878 0.99927856 0.99923864 0.99917547 0.99904769 0.99902513]
10	Credit reporting	delete .reinserted .judgement .disputed .annualcreditreport .trans .report .transunion .experian .equifax	disputed resolve .possible thank .promptly delete .credit report .required promptly .trans union .manner soon .resolve manner .unauthorized fraudulent .experian ha	[1.70869326e-05 1.65898461e-05 2.82189847e-06 2.82189847e-06 1.64370516e-06 2.46241010e-07 1.83638604e-07 3.40983355e-09 4.17810225e-14 1.79138969e-14]

	label	MC unigrams	MC bigrams	PVals
11	Credit reporting	dealt .mailed .status .supposed .tenant .invoice .measure .discrepancy .recommended .included	representative phone .approved credit .wa low .high credit .account causing .did account .payment current .loan forgiven .failed provide .judgement wa	[0.99962706 0.99959234 0.99954296 0.99953539 0.99948625 0.99948438 0.99940449 0.99938283 0.9992756 0.99925899]
12	Credit reporting, credit repair services, or other personal consumer reports	remove .inquires .bureau .experian .transunion .credit .reporting .report .equifax .inquiry	mistake appear .report understanding .credit inquiry .reporting agency .identity theft .credit file .credit bureau .inquiry credit .hard inquiry .credit report	[3.89786583e-09 8.36515878e-10 7.00291363e-10 3.62513726e-10 3.54619166e-10 6.28775598e-11 2.86471526e-11 2.64199599e-15 2.86513358e-17 3.71190296e-28]
13	Credit reporting, credit repair services, or other personal consumer reports	contract .protecting .pertinent .emotionally .okay .wear .befor .ding .numerous .mon	related account .victim identify .know happened .loan past .information company .request proof .point ha .informed representative .scam help .refused acknowledge	[0.99999522 0.99999281 0.99998402 0.99996839 0.9999563 0.99992188 0.99988381 0.99983992 0.99978986 0.99977892]
14	Debt collection	loan .medical .validation .calling .owe .collector .recovery .collect .collection .debt	stop calling .owe debt .alleged debt .portfolio recovery .trying collect .debt collection .debt collector .debt wa .collect debt .collection agency	[1.15815899e-11 1.12808632e-11 8.23835997e-12 1.42342743e-12 6.25531628e-13 4.13375959e-14 1.59967253e-14 8.46509556e-17 6.68354027e-27 4.65272161e-56]
15	Debt collection	printing .lengthy .eye .pointless .resume .patience .register .recognized .dmv .separated	approval letter .t loan .bureau credit .number request .called just .score negatively .party credit .wa did .fine wa .stated sent	[0.99999333 0.99994284 0.99994111 0.9998762 0.99985254 0.99979762 0.99979025 0.99977753 0.99975857 0.99965208]
16	Money transfer, virtual currency, or money service	coin .usd .paypal .currency .moneygram .wire .transfer .cryptocurrency .bitcoin .coinbase	transaction day .buy sell .transfer bank .customer support .wire transfer .money coinbase .support ticket .coinbase com .account coinbase .coinbase account	[1.23276033e-015 9.57992440e-016 1.48856995e-016 1.04946557e-018 1.26069635e-022 6.70655525e-026 6.70326232e-031 8.22937257e-033 1.80944317e-058 9.76603581e-163]
17	Money transfer, virtual currency, or money service	bought .old .match .reverse .cause .posted .identification .meet .country .directly	paid month .called number .mark credit .loan did .trade commission .account time .month later .supervisor wa .wa completed .told account	[0.99993569 0.99992095 0.99953017 0.99942124 0.99925782 0.99916608 0.99913447 0.99825411 0.99814991 0.99735049]

	label	MC unigrams	MC bigrams	PVals
18	Money transfers	receiver .shipping .manipulated .ph .park .cardmember .recipient .western .gram .paypal	saying money .fraud want .cancel transaction .money deposited .using paypal .send money .money transfer .money paypal .western union .money gram	[2.34117144e-13 1.77210293e-16 2.65211708e-17 1.77949712e-18 8.98505201e-20 3.48911000e-22 1.76231617e-24 3.72440459e-38 3.72440459e-38 3.23806508e-49]
19	Money transfers	claiming .billing .came .damage .happened .tax .investigation .caused .did .account	wa paid .wa wa .company wa .wa charged .account ha .appropriate proof .apply consumer .apply check .company shall .company provision	[0.99889819 0.9985256 0.99851285 0.99842131 0.99820066 0.99704121 0.99680317 0.99679875 0.99550659 0.9954255]
20	Mortgage	house .nationstar .sale .home .ocwen .property .foreclosure .escrow .modification .mortgage	nationstar mortgage .foreclosure sale .mortgage wa .property tax .sale date .short sale .escrow account .mortgage payment .mortgage company .loan modification	[5.31252172e-21 2.65543367e-23 1.54591931e-23 6.12072313e-24 7.03030320e-28 1.26876455e-28 1.14593347e-39 1.39716686e-42 5.45052580e-54 1.87579643e-89]
21	Mortgage	click .goal .mobile .age .targeted .protect .rip .additionally .odd .accruing	went online .thing happen .available loan .u year .correct problem .sent received .pay check .contacted told .ask question .law regulation	[0.99991539 0.99982952 0.99980736 0.99977306 0.99976337 0.99975924 0.99966628 0.99964352 0.99964347 0.99963217]
22	Payday loan	payed .big .threating .cash .indiana .speedy .picture .percentage .ace .payday	speedy cash .loan ace .percentage rate .ace cash .cash express .finance charge .took payday .picture loan .payday loan .big picture	[3.90498770e-15 5.02805968e-18 4.76486889e-19 2.48288218e-21 1.95068666e-21 3.86756184e-22 8.17419633e-25 7.86080879e-31 3.31877264e-31 8.42185979e-32]
23	Payday loan	ended .month .resolve .afford .refund .statement .recorded .ocwen .didnt .number	did authorize .wa paid .information credit .block ha .notified promptly .agency subsequent .manner consumer .section p .exception resellers .exception verification	[0.99938199 0.99873821 0.99840028 0.99773663 0.99739746 0.99660654 0.99578609 0.99379129 0.99233652 0.99203759]
24	Payday loan, title loan, or personal loan	cashnet .lending .club .georgia .loan .payday .usury .borrowed .pls .mobiloans	state georgia .make loan .got loan .wa easy .code state .took loan .lending club .illegal state .usury law .loan state	[1.46116044e-07 1.23089860e-07 4.86646277e-08 4.07796487e-09 3.60543926e-09 1.90380288e-09 2.61210279e-10 1.28168016e-10 1.61062554e-11 2.05445051e-12]
25	Payday loan, title loan, or personal loan	single .poor .surprise .chapter .yesterday .form .trade .lawsuit .promptly .led	past month .account reported .pulled credit .synchrony bank .know wa .debit card .card payment .payment history .called time .copy letter	[0.99998327 0.99901227 0.99897172 0.99887595 0.99871123 0.9986854 0.9977833 0.99768195 0.99764112 0.99729796]

	label	MC unigrams	MC bigrams	PVals
26	Prepaid card	pst .expedited .moneygram .access .ur .trailer .gift .rescind .prepaid .rushcard	express serve .receive loan .came went .rush card .locked account .pre paid .able reach .prepaid card .t access .access fund	[3.47841266e-010 7.24809207e-012 2.01057255e-013 6.30313950e-014 4.14202641e-015 1.05522745e-023 1.31360844e-025 1.44457490e-030 9.37681765e-044 3.84941059e-150]
27	Prepaid card	calling .mailed .authorize .document .set .practice .close .told .cell .respond	account account .did n .subsequent use .agency rescinds .issue authorization .identifies consumer .requiring consumer .b time .authority decline .identified information	[0.99825784 0.9975646 0.99556022 0.9951412 0.994963 0.99470984 0.99423171 0.99391009 0.99262117 0.99214283]
28	Student loan	college .mae .private .forgiveness .forbearance .repayment .school .loan .student .navient	called navient .income driven .private student .income based .federal loan .loan navient .sallie mae .private loan .loan forgiveness .student loan	[3.46042517e-020 5.34019506e-021 8.39228307e-023 1.26236486e-028 1.01294242e-029 1.61051573e-033 9.22188552e-035 5.20010906e-055 5.43761857e-058 3.91078462e-103]
29	Student loan	slow .want .tomorrow .dozen .defamation .xxxxi .complain .box .record .word	got information .issue received .company make .stating needed .statement balance .able receive .record credit .fee pay .representative called .thank time	[0.99978811 0.99975863 0.99960032 0.99951965 0.99946714 0.99938097 0.99936345 0.99932826 0.99927676 0.99917413]
30	Vehicle loan or lease	toyota .gap .acceptance .gm .santander .westlake .repo .warranty .vehicle .car	gm financial .vehicle wa .sold auction .car worth .car payment .paid like .gap insurance .car wa .warranty wa .credit acceptance	[4.77357213e-11 4.66862121e-11 3.99271722e-11 2.78220009e-12 3.81111428e-13 2.67533894e-13 1.60026815e-15 1.05997893e-16 8.31189807e-19 2.11509595e-32]
31	Vehicle loan or lease	hospital .correspondence .informed .ordered .rectify .area .reach .hr .attaching .mailed	remove late .letter called .company trying .wa talking .called ask .attempted contact .owe money .late credit .payment plan .day passed	[0.99988348 0.99984562 0.99983023 0.99953466 0.9995081 0.99940901 0.99933308 0.9991575 0.99907062 0.99891947]

READY

4.2 Audit of some of the bad classifications

READY

Many of the unigrams and bigrams look bad without a chi2 filter

Simple check

READY

```
%pyspark

iscorrect = [pred_test == y_test[:ts_subset]]
badones   = [(X_test.iloc[i], y_test.iloc[i], pred_test[i]) for i in range(np.size(iscorrect)) if

for badone in badones[:20]:
    print('[Actual: %s, Predicted: %s]\n%s\n' % (badone[2], badone[1], badone[0]))
    print("---\n")
```

```
 sure who else they call. XXXX from Ally denies this happened. Ally also told my friend they wer
e calling other people too. She left her phone # and extension with my friend. Sadly and most fr
ighteningly, Ally knows my phone number and should not have been calling other people, telling t
hem my business, and putting out my personal information like that - its unlawful, and goes agai
nst the FTC rules and regulations. I 'm not sure what other information they leaked about me, bu
t it needs to come to end. And fast!
---
[Actual: Credit card or prepaid card, Predicted: Credit reporting, credit repair services, or ot
her personal consumer reports]
XXXX has been non-compliant with removing the unverified account WELLS FARGO DLR SVC/ which has
 been deleted by XXXX and XXXX . XXXX   and XXXX have both completed their research and determin
ed that WELLS FARGO DLR SVC/ was not established by myself but XXXX keeps verifying this account
s. Also, when I called the company they responded that XXXX did not send them all of the verifyi
ng evidence to confirm each account is unverified.
---
[Actual: Credit card or prepaid card, Predicted: Credit card]
I suspect that Citibank from which I have a credit card, shares my credit card transaction data
```

READY

4.3 Show log-regression coefficients

Dig out used features

READY

```
%pyspark

# Get support
support =  text_clf.named_steps['chi2'].get_support()

a =    text_clf.named_steps['allf'].transformer_list[0][1].named_steps['vect'].get_feature_names()
b =    "len"
c =    text_clf.named_steps['allf'].transformer_list[2][1].named_steps['ta'].get_feature_names()

# Get feature names
allfeats = np.append(np.append(a, b), c)
```

Look at regression coefficient for each category

READY

```
%pyspark
topcats         = sorted(category_to_id.items())
coefs           = pd.DataFrame(columns=['name', 'coef'])

coefs['name']   = allfeats[support]

for cattext, catnum in topcats[:3]:
    print("---  Doing: {}".format(cattext))
    coefs['coef']   = text_clf.named_steps['clf'].coef_[catnum]

    coefs.sort_values(by='coef', ascending=False, inplace=True)
    cn.show(coefs, mr=10, type='st', fs=90)
    coefs.sort_values(by='coef', ascending=True, inplace=True)
    cn.show(coefs, mr=10, type='st', fs=90)

    coefs.sort_values(by='name', ascending=True, inplace=True)
    cn.show(coefs, mr=10, type='st', fs=90)
```

READY

5. Save classifier

READY

No pickle support for LemmaTokenizer

READY

```
%pyspark

# Save to pickle
pk.dump(text_clf, open(modelpath + 'TextAnalysis_Ex2_myclf.data', "w"))
```

```
 File "/usr/lib64/python2.7/pickle.py", line 663, in _batch_setitems
   save(v)
 File "/usr/lib64/python2.7/pickle.py", line 286, in save
   f(self, obj) # Call unbound method with explicit self
 File "/usr/lib64/python2.7/pickle.py", line 600, in save_list
   self._batch_appends(iter(obj))
 File "/usr/lib64/python2.7/pickle.py", line 615, in _batch_appends
   save(x)
 File "/usr/lib64/python2.7/pickle.py", line 286, in save
   f(self, obj) # Call unbound method with explicit self
 File "/usr/lib64/python2.7/pickle.py", line 562, in save_tuple
   save(element)
 File "/usr/lib64/python2.7/pickle.py", line 331, in save
   self.save_reduce(obj=obj, *rv)
 File "/usr/lib64/python2.7/pickle.py", line 419, in save_reduce
   save(state)
 File "/usr/lib64/python2.7/pickle.py", line 286, in save

   f(self, obj) # Call unbound method with explicit self
```

READY

```
%pyspark
```

READY