sms spam or ham

August 6, 2020

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
sms = pd.read_csv("datasets_483_982_spam.csv", encoding='ISO-8859-1')
sms.head()
```

[2]:
```
     v1                                                 v2 Unnamed: 2  \
0   ham  Go until jurong point, crazy.. Available only ...        NaN
1   ham                      Ok lar... Joking wif u oni...        NaN
2  spam  Free entry in 2 a wkly comp to win FA Cup fina...        NaN
3   ham  U dun say so early hor... U c already then say...        NaN
4   ham  Nah I don't think he goes to usf, he lives aro...        NaN

  Unnamed: 3 Unnamed: 4
0        NaN        NaN
1        NaN        NaN
2        NaN        NaN
3        NaN        NaN
4        NaN        NaN
```

[3]:
```
sms.dropna(how="any", inplace=True, axis=1)
sms = sms[['v1', 'v2']]
sms = sms.rename(columns = {'v1': 'label', 'v2':'message'})
sms.head()
```

[3]:
```
  label                                            message
0   ham  Go until jurong point, crazy.. Available only ...
1   ham                      Ok lar... Joking wif u oni...
2  spam  Free entry in 2 a wkly comp to win FA Cup fina...
3   ham  U dun say so early hor... U c already then say...
4   ham  Nah I don't think he goes to usf, he lives aro...
```

[4]:
```
sms.describe()
```

[4]:
```
       label                 message
count   5572                    5572
unique     2                    5169
```

```
top      ham  Sorry, I'll call later
freq    4825                    30
```

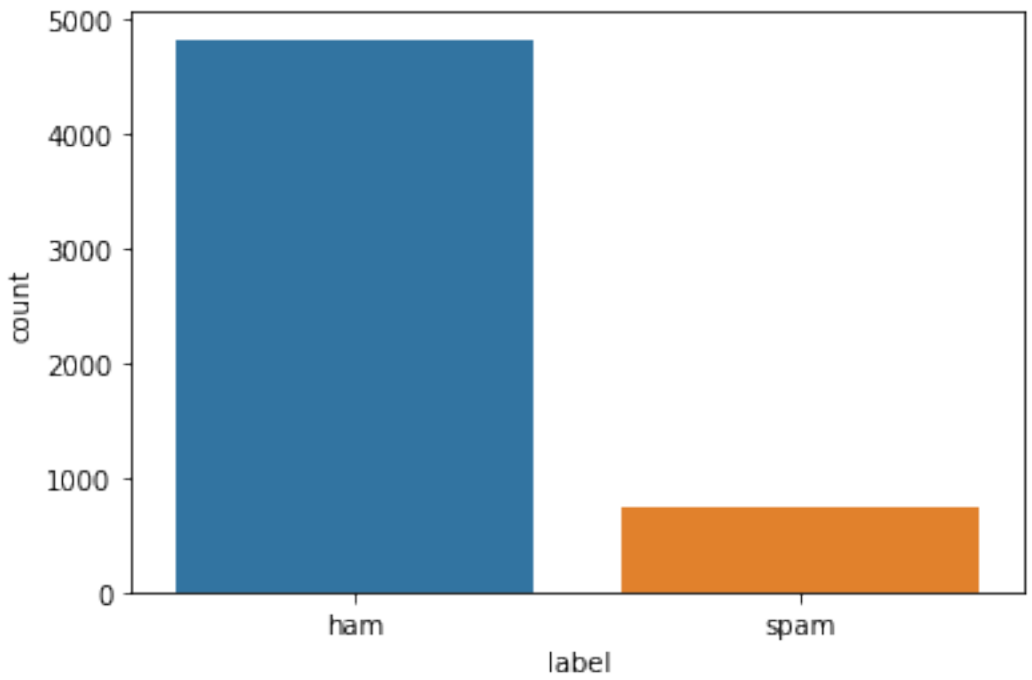
[5]:
```
sms['label_num'] = sms['label'].map({'ham':0, 'spam':1})
sms['message_len'] = sms['message'].apply(len)
sms.head()
```

[5]:
```
  label                                            message  label_num  \
0   ham  Go until jurong point, crazy.. Available only ...          0
1   ham                      Ok lar... Joking wif u oni...          0
2  spam  Free entry in 2 a wkly comp to win FA Cup fina...          1
3   ham  U dun say so early hor... U c already then say...          0
4   ham  Nah I don't think he goes to usf, he lives aro...          0

   message_len
0          111
1           29
2          155
3           49
4           61
```

[6]:
```
sns.countplot(sms['label'])
```

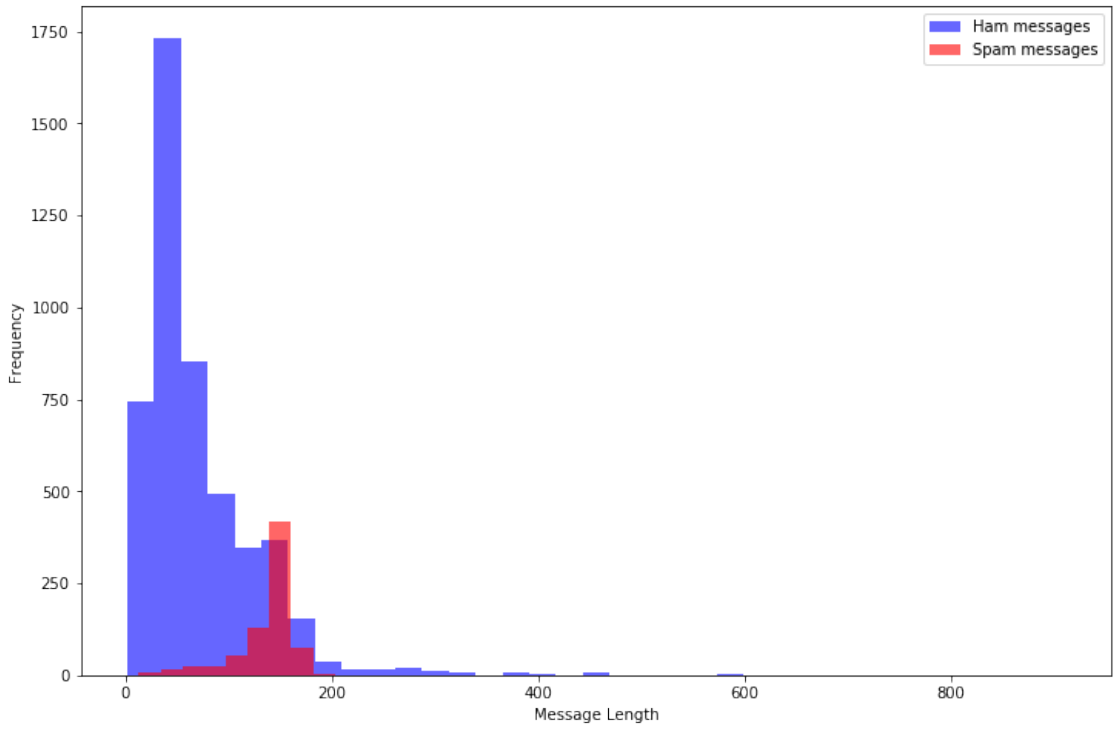
[6]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe4326fa5c0>
```



[7]:
```
plt.figure(figsize=(12, 8))
```

```
sms[sms.label=='ham'].message_len.plot(bins=35, kind='hist',␣
 ↪color='blue',label='Ham messages', alpha=0.6)
sms[sms.label=='spam'].message_len.plot(kind='hist', color='red',label='Spam␣
 ↪messages', alpha=0.6)
plt.legend()
plt.xlabel("Message Length")
```

[7]:
```
Text(0.5, 0, 'Message Length')
```



[8]:
```
sms[sms['label'] == 'ham'].describe()
```

[8]:

	label_num	message_len
count	4825.0	4825.000000
mean	0.0	71.023627
std	0.0	58.016023
min	0.0	2.000000
25%	0.0	33.000000
50%	0.0	52.000000
75%	0.0	92.000000
max	0.0	910.000000

[9]:
```
sms[sms['label'] == 'spam'].describe()
```

[9]:

	label_num	message_len
count	747.0	747.000000
mean	1.0	138.866131

```
std          0.0    29.183082
min          1.0    13.000000
25%          1.0   132.500000
50%          1.0   149.000000
75%          1.0   157.000000
max          1.0   224.000000
```

[10]:
```
###let's take a look at the message with length of 910
sms[sms['message_len'] == 910]['message'].iloc[0]
```

[10]:
```
"For me the love should start with attraction.i should feel that I need her
every time around me.she should be the first thing which comes in my thoughts.I
would start the day and end it with her.she should be there every time I
dream.love will be then when my every breath has her name.my life should happen
around her.my life will be named to her.I would cry for her.will give all my
happiness and take all her sorrows.I will be ready to fight with anyone for
her.I will be in love when I will be doing the craziest things for her.love will
be when I don't have to proove anyone that my girl is the most beautiful lady on
the whole planet.I will always be singing praises for her.love will be when I
start up making chicken curry and end up makiing sambar.life will be the most
beautiful then.will get every morning and thank god for the day because she is
with me.I would like to say a lot..will tell later.."
```

[11]:
```
# text preprocessing
import string
from nltk.corpus import stopwords
```

[12]:
```
### Takes in a string of text, then performs the following:
#    1. Remove all punctuation
#    2. Remove all stopwords
#    3. Returns a list of the cleaned text

def text_process(mess):
    stops = stopwords.words('english') + ['u', 'ü', 'ur', '4', '2', 'im',␣
 ↪'dont', 'doin', 'ure']
    nopunc = [w for w in mess if w not in string.punctuation]
    nopunc = ''.join(nopunc)

    return ' '.join([w for w in nopunc.split() if w.lower() not in stops])
```

[13]:
```
sms['clean_msg'] = sms.message.apply(text_process)
```

[14]:
```
sms.head()
```

[14]:
```
  label                                            message  label_num  \
0   ham  Go until jurong point, crazy.. Available only ...          0
1   ham                      Ok lar... Joking wif u oni...          0
2  spam  Free entry in 2 a wkly comp to win FA Cup fina...          1
3   ham  U dun say so early hor... U c already then say...          0
4   ham  Nah I don't think he goes to usf, he lives aro...          0
```

```
   message_len                                          clean_msg
0          111  Go jurong point crazy Available bugis n great ...
1           29                              Ok lar Joking wif oni
2          155  Free entry wkly comp win FA Cup final tkts 21s...
3           49                  dun say early hor c already say
4           61             Nah think goes usf lives around though
```

[15]:
```
from collections import Counter

words = sms[sms.label=='ham'].clean_msg.apply(lambda x: [word.lower() for word␣
 ↪in x.split()])
ham_words = Counter()

for msg in words:
    ham_words.update(msg)

print(ham_words.most_common(50))
```

```
[('get', 303), ('ltgt', 276), ('ok', 272), ('go', 247), ('ill', 236), ('know',
232), ('got', 231), ('like', 229), ('call', 229), ('come', 224), ('good', 222),
('time', 189), ('day', 187), ('love', 185), ('going', 167), ('want', 163),
('one', 162), ('home', 160), ('lor', 160), ('need', 156), ('sorry', 153),
('still', 146), ('see', 137), ('n', 134), ('later', 134), ('da', 131), ('r',
131), ('back', 129), ('think', 128), ('well', 126), ('today', 125), ('send',
123), ('tell', 121), ('cant', 118), ('ì', 117), ('hi', 117), ('take', 112),
('much', 112), ('oh', 111), ('night', 107), ('hey', 106), ('happy', 105),
('great', 100), ('way', 100), ('hope', 99), ('pls', 98), ('work', 96), ('wat',
95), ('thats', 94), ('dear', 94)]
```

[16]:
```
words = sms[sms.label=='spam'].clean_msg.apply(lambda x: [word.lower() for word␣
 ↪in x.split()])
spam_words = Counter()

for msg in words:
    spam_words.update(msg)

print(spam_words.most_common(50))
```

```
[('call', 347), ('free', 216), ('txt', 150), ('mobile', 123), ('text', 120),
('claim', 113), ('stop', 113), ('reply', 101), ('prize', 92), ('get', 83),
('new', 69), ('send', 67), ('nokia', 65), ('urgent', 63), ('cash', 62), ('win',
60), ('contact', 56), ('service', 55), ('please', 52), ('guaranteed', 50),
('customer', 49), ('16', 49), ('week', 49), ('tone', 48), ('per', 46), ('phone',
45), ('18', 43), ('chat', 42), ('awarded', 38), ('draw', 38), ('latest', 36),
('åč1000', 35), ('line', 35), ('150ppm', 34), ('mins', 34), ('receive', 33),
('camera', 33), ('1', 33), ('every', 33), ('message', 32), ('holiday', 32),
```

```
('landline', 32), ('shows', 31), ('åč2000', 31), ('go', 31), ('box', 30),
('number', 30), ('apply', 29), ('code', 29), ('live', 29)]
```

[17]:
```
sms.head()
```

[17]:
```
  label                                            message  label_num  \
0   ham  Go until jurong point, crazy.. Available only ...          0
1   ham                      Ok lar... Joking wif u oni...          0
2  spam  Free entry in 2 a wkly comp to win FA Cup fina...          1
3   ham  U dun say so early hor... U c already then say...          0
4   ham  Nah I don't think he goes to usf, he lives aro...          0

   message_len                                        clean_msg
0          111  Go jurong point crazy Available bugis n great ...
1           29                              Ok lar Joking wif oni
2          155  Free entry wkly comp win FA Cup final tkts 21s...
3           49                  dun say early hor c already say
4           61             Nah think goes usf lives around though
```

[18]:
```
X = sms.clean_msg
y = sms.label_num
print(X.shape)
print(y.shape)
```

```
(5572,)
(5572,)
```

[19]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = 1)
```

[20]:
```
from sklearn.feature_extraction.text import CountVectorizer

vect = CountVectorizer()
vect.fit(X_train)
X_train_dtm = vect.fit_transform(X_train)
```

[21]:
```
X_test_dtm  = vect.transform(X_test)
```

[22]:
```
#### tfidf
from sklearn.feature_extraction.text import TfidfTransformer

tfidf_transformer = TfidfTransformer()
tfidf_transformer.fit(X_train_dtm)
tfidf_transformer.transform(X_train_dtm)
```

[22]:
```
<4179x7996 sparse matrix of type '<class 'numpy.float64'>'
        with 34796 stored elements in Compressed Sparse Row format>
```

[23]:
```
### model
from sklearn.naive_bayes import MultinomialNB
nb = MultinomialNB()
```

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

[23]:
```
MultinomialNB()
```

[24]:
```
y_pred_class = nb.predict(X_test_dtm)
```

[25]:
```
from sklearn import metrics
metrics.accuracy_score(y_test, y_pred_class)
```

[25]:
```
0.9827709978463748
```

[26]:
```
metrics.confusion_matrix(y_test, y_pred_class)
```

[26]:
```
array([[1205,    8],
       [  16,  164]])
```

[27]:
```
### take a look at false positive
X_test[(y_pred_class == 1)&(y_test == 0)]
```

[27]:
```
2418    Madamregret disturbancemight receive reference...
4598                                laid airtel line rest
386                                   Customer place call
1289    HeyGreat dealFarm tour 9am 5pm 95pax 50 deposi...
5094    Hi ShanilRakhesh herethanksi exchanged uncut d...
494                                      free nowcan call
759     Call youcarlos isare phones vibrate acting mig...
3140                                  Customer place call
Name: clean_msg, dtype: object
```

[28]:
```
sms.message.iloc[[2418]].iloc[0]
```

[28]:
```
'Madam,regret disturbance.might receive a reference check from DLF
Premarica.kindly be informed.Rgds,Rakhesh,Kerala.'
```

[29]:
```
### take a look at false negative
X_test[(y_pred_class == 0)&(y_test == 1)]
```

[29]:
```
4674    Hi babe Chloe r smashed saturday night great w...
3528    Xmas New Years Eve tickets sale club day 10am ...
3417    LIFE never much fun great came made truly spec...
2773    come takes little time child afraid dark becom...
1960    Guess Somebody know secretly fancies Wanna fin...
5       FreeMsg Hey darling 3 weeks word back Id like ...
2078                         85233 FREERingtoneReply REAL
1457    CLAIRE havin borin time alone wanna cum 2nite ...
190     unique enough Find 30th August wwwareyouunique...
2429    Guess IThis first time created web page WWWASJ...
3057    unsubscribed services Get tons sexy babes hunk...
1021    Guess Somebody know secretly fancies Wanna fin...
4067    TBSPERSOLVO chasing us since Sept foråč38 defi...
3358         Sorry missed call lets talk time 07090201529
2821    ROMCAPspam Everyone around responding well pre...
2247    Back work 2morro half term C 2nite sexy passio...
```

```
Name: clean_msg, dtype: object
```

[30]:
```
sms.message.iloc[[4674]].iloc[0]
```

[30]:
```
'Hi babe its Chloe, how r u? I was smashed on saturday night, it was great! How
was your weekend? U been missing me? SP visionsms.com Text stop to stop
150p/text'
```

[31]:
```
### use tfidf and pipeline
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.pipeline import Pipeline

pipe = Pipeline([
    ('bow', CountVectorizer()),
    ('tfidf', TfidfTransformer()),
    ('model', MultinomialNB())
])

pipe.fit(X_train, y_train)
```

[31]:
```
Pipeline(steps=[('bow', CountVectorizer()), ('tfidf', TfidfTransformer()),
                ('model', MultinomialNB())])
```

[32]:
```
y_pred = pipe.predict(X_test)
metrics.accuracy_score(y_test, y_pred)
```

[32]:
```
0.9669777458722182
```

[33]:
```
### comparing different models
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression(solver = 'liblinear')
logreg.fit(X_train_dtm, y_train)
```

[33]:
```
LogisticRegression(solver='liblinear')
```

[34]:
```
y_pred_class = logreg.predict(X_test_dtm)

metrics.accuracy_score(y_test, y_pred_class)
```

[34]:
```
0.9842067480258435
```

[ ]: