stack overflow

In [1]:

```
import warnings
warnings.filterwarnings("ignore")
import pandas as pd
pd.set_option('display.max_colwidth',100)
import sqlite3
import csv
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from wordcloud import WordCloud
import re
import os
os.chdir('./Dataset/')
from sqlalchemy import create_engine # database connection
import datetime as dt
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
nltk.download('punkt')
from nltk.tokenize import word_tokenize
from nltk.stem.snowball import SnowballStemmer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.multiclass import OneVsRestClassifier
from sklearn.linear_model import SGDClassifier
from sklearn import metrics
from sklearn.metrics import f1_score,precision_score,recall_score
from sklearn import svm
from sklearn.linear_model import LogisticRegression
# from skmultilearn.adapt import mlknn
# from skmultilearn.problem_transform import ClassifierChain
# from skmultilearn.problem_transform import BinaryRelevance
# from skmultilearn.problem_transform import LabelPowerset
from sklearn.naive_bayes import GaussianNB
from datetime import datetime
from sklearn.model_selection import GridSearchCV
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /home/rushabh6792/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     /home/rushabh6792/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Stack Overflow: Tag Prediction

1. Business Problem

1.1 Description

Description

Stack Overflow is the largest, most trusted online community for developers to learn, share their programming knowledge, and build their careers.

Stack Overflow is something which every programmer use one way or another. Each month, over 50 million developers come to Stack Overflow to learn, share their knowledge, and build their careers. It features questions and answers on a wide range of topics in computer programming. The website serves as a platform for users to ask and answer questions, and, through membership and active participation, to vote questions and answers up or down and edit questions and answers in a fashion similar to a wiki or Digg. As of April 2014 Stack Overflow has over 4,000,000 registered users, and it exceeded 10,000,000 questions in late August 2015. Based on the type of tags assigned to questions, the top eight most discussed topics on the site are: Java, JavaScript, C#, PHP, Android, jQuery, Python and HTML.

Problem Statemtent

Suggest the tags based on the content that was there in the question posted on Stackoverflow.

Source: https://www.kaggle.com/c/facebook-recruiting-iii-keyword-extraction/

1.2 Source / useful links

Data Source : https://www.kaggle.com/c/facebook-recruiting-iii-keyword-extraction/data
Youtube : https://youtu.be/nNDqbUhtIRg
Research paper : https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tagging-1.pdf
Research paper : https://dl.acm.org/citation.cfm?id=2660970&dl=ACM&coll=DL

1.3 Real World / Business Objectives and Constraints

1. Predict as many tags as possible with high precision and recall.

2. Incorrect tags could impact customer experience on StackOverflow.
3. No strict latency constraints.

2. Machine Learning problem

2.1 Data

2.1.1 Data Overview

Refer: https://www.kaggle.com/c/facebook-recruiting-iii-keyword-extraction/data
All of the data is in 2 files: Train and Test.

Train.csv contains 4 columns: Id,Title,Body,Tags.

Test.csv contains the same columns but without the Tags, which you are to predict.

Size of Train.csv - 6.75GB

Size of Test.csv - 2GB

Number of rows in Train.csv = 6034195

The questions are randomized and contains a mix of verbose text sites as well as sites related to math and programming. The number of questions from each site may vary, and no filtering has been performed on the questions (such as closed questions).

Data Field Explaination

Dataset contains 6,034,195 rows. The columns in the table are:

Id - Unique identifier for each question

Title - The question's title

Body - The body of the question

Tags - The tags associated with the question in a space-seperated format (all lowercase, should not contain tabs '\t' or ampersands '&')

2.1.2 Example Data point

Title: Implementing Boundary Value Analysis of Software Testing in a C++ program?
Body :

```
            #include<
            iostream>\n
            #include<
            stdlib.h>\n\n
            using namespace std;\n\n
            int main()\n
            {\n
                     int n,a[n],x,c,u[n],m[n],e[n][4];\n
```

```
                cout<<"Enter the number of variables";\n
       cin>>n;\n\n
                cout<<"Enter the Lower, and Upper Limits
of the variables";\n
                for(int y=1; y<n+1; y++)\n
                {\n
                   cin>>m[y];\n
                   cin>>u[y];\n
                }\n
                for(x=1; x<n+1; x++)\n
                {\n
                   a[x] = (m[x] + u[x])/2;\n
                }\n
                c=(n*4)-4;\n
                for(int a1=1; a1<n+1; a1++)\n
                {\n\n
                   e[a1][0] = m[a1];\n
                   e[a1][1] = m[a1]+1;\n
                   e[a1][2] = u[a1]-1;\n
                   e[a1][3] = u[a1];\n
                }\n
                for(int i=1; i<n+1; i++)\n
                {\n
                   for(int l=1; l<=i; l++)\n
                   {\n
                       if(l!=1)\n
                       {\n
                           cout<<a[l]<<"\\t";\n

                       }\n
                   }\n
                   for(int j=0; j<4; j++)\n
                   {\n
                       cout<<e[i][j];\n
                       for(int k=0; k<n-(i+1); k++)\n

                       {\n
                           cout<<a[k]<<"\\t";\n

                       }\n
                       cout<<"\\n";\n
                   }\n
                }    \n\n
                system("PAUSE");\n
                return 0;    \n
       }\n
```

\n\n

The answer should come in the form of a table like
\n\n

```
1            50              50\n
2            50              50\n
99           50              50\n
100          50              50\n
50           1               50\n
50           2               50\n
50           99              50\n
50           100             50\n
50           50              1\n
50           50              2\n
50           50              99\n
```

```
           50                     50                100\n


\n\n


if the no of inputs is 3 and their ranges are\n
        1,100\n
        1,100\n
        1,100\n
        (could be varied too)
\n\n


The output is not coming,can anyone correct the code or tell me
 what\'s wrong?
\n'
Tags : 'c++ c'
```

2.2 Mapping the real-world problem to a Machine Learning Problem

2.2.1 Type of Machine Learning Problem

It is a multi-label classification problem
Multi-label Classification: Multilabel classification assigns to each sample a set of target labels. This can be thought as predicting properties of a data-point that are not mutually exclusive, such as topics that are relevant for a document. A question on Stackoverflow might be about any of C, Pointers, FileIO and/or memory-management at the same time or none of these.
__Credit__: http://scikit-learn.org/stable/modules/multiclass.html

2.2.2 Performance metric

Micro-Averaged F1-Score (Mean F Score) : The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is:

*F1 = 2 * (precision * recall) / (precision + recall)*

In the multi-class and multi-label case, this is the weighted average of the F1 score of each class.

'Micro f1 score':
Calculate metrics globally by counting the total true positives, false negatives and false positives. This is a better metric when we have class imbalance.

'Macro f1 score':
Calculate metrics for each label, and find their unweighted mean. This does not take label imbalance into account.

https://www.kaggle.com/wiki/MeanFScore
http://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

Hamming loss : The Hamming loss is the fraction of labels that are incorrectly predicted.
https://www.kaggle.com/wiki/HammingLoss

3. Exploratory Data Analysis

3.1 Data Loading and Cleaning

3.1.1 Using Pandas with SQLite to Load the data

In [2]:

```python
#Creating db file from csv
#Learn SQL: https://www.w3schools.com/sql/default.asp
if not os.path.isfile('train.db'):
    start = datetime.now()
    disk_engine = create_engine('sqlite:///train.db')
    start = dt.datetime.now()
    chunksize = 180000
    j = 0
    index_start = 1
    for df in pd.read_csv('Train.csv', names=['Id', 'Title', 'Body', 'Tags'], chunksize=chunksize, iterator=True, encoding='utf-8', ):
        df.index += index_start
        j+=1
#         print('{} rows'.format(j*chunksize))
        df.to_sql('data', disk_engine, if_exists='append')
        index_start = df.index[-1] + 1
    print("Time taken to run this cell :", datetime.now() - start)
```

3.1.2 Counting the number of rows

In [3]:

```python
if os.path.isfile('train.db'):
    start = datetime.now()
    con = sqlite3.connect('train.db')
    num_rows = pd.read_sql_query("""SELECT count(*) FROM data""", con)
    #Always remember to close the database
    print("Number of rows in the database :","\n",num_rows['count(*)'].values[0])
    con.close()
    print("Time taken to count the number of rows :", datetime.now() - start)
else:
    print("Please download the train.db file from drive or run the above cell to genarate train.db file")
```

```
Number of rows in the database :
 6034196
Time taken to count the number of rows : 0:01:41.252723
```

3.1.3 Checking for duplicates

In [4]:

```
#Learn SQl: https://www.w3schools.com/sql/default.asp
if os.path.isfile('train.db'):
    start = datetime.now()
    con = sqlite3.connect('train.db')
    df_no_dup = pd.read_sql_query('SELECT Title, Body, Tags, COUNT(*) a
s cnt_dup FROM data GROUP BY Title, Body, Tags', con)
    con.close()
    print("Time taken to run this cell :", datetime.now() - start)
else:
    print("Please download the train.db file from drive or run the firs
t to genarate train.db file")
```

```
Time taken to run this cell : 0:02:49.410350
```

In [5]:

```
df_no_dup.head()
# we can observe that there are duplicates
```

Out[5]:

	Title	Bod
0	Implementing Boundary Value Analysis of Software Testing in a C++ program?	<pre <code>#include&lt;iostream&gt;\n#include&lt;stdlib.h&gt;\n\nusin namespace std;\n\nint mai.
1	Dynamic Datagrid Binding in Silverlight?	<p>I should do binding for datagrid dynamically at code. I wrot the code as below. When I debug.
2	Dynamic Datagrid Binding in Silverlight?	<p>I should do binding for datagrid dynamically at code. I wrot the code as below. When I debug.
3	java.lang.NoClassDefFoundError: javax/servlet/jsp/tagext/TagLibraryValidator	<p>I followed the guide in < href="http://stackoverflow.com/tags/jstl/info">this link</a> to in.
4	java.sql.SQLException:[Microsoft][ODBC Driver Manager] Invalid descriptor index	<p>I use the following code</p>\n\n<pre><code>try {\ Class.forName("sun.jdbc.odbc.JdbcOdbcDr.

In [6]:

```
print("number of duplicate questions :", num_rows['count(*)'].values[0]
- df_no_dup.shape[0], "(",(1-((df_no_dup.shape[0])/(num_rows['count(*)'
].values[0])))*100,"% )")
```

```
number of duplicate questions : 1827881 ( 30.292038906260256 % )
```

In [7]:

```
# number of times each question appeared in our database
df_no_dup.cnt_dup.value_counts()
```

Out[7]:

```
1    2656284
2    1272336
3     277575
4         90
5         25
6          5
Name: cnt_dup, dtype: int64
```

In [8]:

```
df_no_dup[df_no_dup.isnull().any(axis=1)]
```

Out[8]:

	Title	Body	Tags	cnt_dup
777547	Do we really need NULL?	<blockquote>\n <p><strong>Possible Duplicate:</strong><br>\n <a href="http://stackoverflow.com...	None	1
962680	Find all values that are not null and not in another table	<p>I am running into a problem which results in an ORA-01722 error. I have a table that contain...	None	1
1126558	Handle NullObjects	<p>I have done quite a bit of research on best ways to deal with null objects.\nThe best I came ...	None	1
1256102	How do Germans call null	<p>In german null means 0, so how do they call null (like null reference) ?</p>\n	None	1
2430668	Page cannot be null. Please ensure that this operation is being performed in the context of an A...	<p>I get this error when i remove dynamically telerik raddock and raddock zone controls and add...	None	1

	Title	Body	Tags	cnt_dup
3329908	What is the difference between NULL and "0"?	<p>What is the difference from NULL and "0"?</p>\n\n<p>Example:</p>\n\n<pre> <code>return NULL;\n...	None	1
3551595	a bit of difference between null and space	<p>I was just reading this quote</p>\n\n<blockquote>\n <p>And don't tell me there isn't one bit...	None	2

In [9]:
```
df_no_dup = df_no_dup[~df_no_dup.isnull().any(axis=1)]
```

In [10]:
```
start = datetime.now()
df_no_dup["tag_count"] = df_no_dup["Tags"].apply(lambda text: len(text.
split(" ")))
# adding a new feature number of tags per question
print("Time taken to run this cell :", datetime.now() - start)
df_no_dup.head()
```

```
Time taken to run this cell : 0:00:03.162973
```

Out[10]:

	Title	Bod
0	Implementing Boundary Value Analysis of Software Testing in a C++ program?	<pre <code>#include&lt;iostream&gt;\n#include&lt;stdlib.h&gt;\n\nusin namespace std;\n\nint mai.
1	Dynamic Datagrid Binding in Silverlight?	<p>I should do binding for datagrid dynamically at code. I wrot the code as below. When I debug.
2	Dynamic Datagrid Binding in Silverlight?	<p>I should do binding for datagrid dynamically at code. I wrot the code as below. When I debug.
3	java.lang.NoClassDefFoundError: javax/servlet/jsp/tagext/TagLibraryValidator	<p>I followed the guide in < href="http://stackoverflow.com/tags/jstl/info">this link</a> to in.
4	java.sql.SQLException:[Microsoft][ODBC Driver Manager] Invalid descriptor index	<p>I use the following code</p>\n\n<pre><code>try {\ Class.forName("sun.jdbc.odbc.JdbcOdbcDr.

In [11]:

```
# distribution of number of tags per question
df_no_dup.tag_count.value_counts()
```

Out[11]:

```
3    1206157
2    1111706
4     814996
1     568291
5     505158
Name: tag_count, dtype: int64
```

In [12]:

```
#Creating a new database with no duplicates
if not os.path.isfile('train_no_dup.db'):
    disk_dup = create_engine("sqlite:///train_no_dup.db")
    no_dup = pd.DataFrame(df_no_dup, columns=['Title', 'Body', 'Tags'])
    no_dup.to_sql('no_dup_train',disk_dup)
```

In [13]:

```
#This method seems more appropriate to work with this much data.
#creating the connection with database file.
if os.path.isfile('train_no_dup.db'):
    start = datetime.now()
    con = sqlite3.connect('train_no_dup.db')
    tag_data = pd.read_sql_query("""SELECT Tags FROM no_dup_train""", c
on)
    #Always remember to close the database
    con.close()

    # Let's now drop unwanted column.
    tag_data.drop(tag_data.index[0], inplace=True)
    #Printing first 5 columns from our data frame
    tag_data.head()
    print("Time taken to run this cell :", datetime.now() - start)
else:
    print("Please download the train.db file from drive or run the abov
e cells to genarate train.db file")
```

```
Time taken to run this cell : 0:00:50.063605
```

3.2 Analysis of Tags

3.2.1 Total number of unique tags

In [14]:
```
# Importing & Initializing the "CountVectorizer" object, which
#is scikit-learn's bag of words tool.

#by default 'split()' will tokenize each tag using space.
vectorizer = CountVectorizer(tokenizer = lambda x: x.split())
# fit_transform() does two functions: First, it fits the model
# and learns the vocabulary; second, it transforms our training data
# into feature vectors. The input to fit_transform should be a list of strings.
tag_dtm = vectorizer.fit_transform(tag_data['Tags'])
```

In [15]:
```
print("Number of data points :", tag_dtm.shape[0])
print("Number of unique tags :", tag_dtm.shape[1])
```

```
Number of data points : 4206307
Number of unique tags : 42048
```

In [16]:
```
#'get_feature_name()' gives us the vocabulary.
tags = vectorizer.get_feature_names()
#Lets look at the tags we have.
print("Some of the tags we have :", tags[:10])
```

```
Some of the tags we have : ['.a', '.app', '.asp.net-mvc', '.aspxauth',
'.bash-profile', '.class-file', '.cs-file', '.doc', '.drv', '.ds-stor
e']
```

3.2.3 Number of times a tag appeared

In [17]:
```
# https://stackoverflow.com/questions/15115765/how-to-access-sparse-matrix-elements
#Lets now store the document term matrix in a dictionary.
```

```
freqs = tag_dtm.sum(axis=0).A1
result = dict(zip(tags, freqs))
```

In [18]:

```
#Saving this dictionary to csv files.
if not os.path.isfile('tag_counts_dict_dtm.csv'):
    with open('tag_counts_dict_dtm.csv', 'w') as csv_file:
        writer = csv.writer(csv_file)
        for key, value in result.items():
            writer.writerow([key, value])
tag_df = pd.read_csv("tag_counts_dict_dtm.csv", names=['Tags', 'Counts'
])
tag_df.head()
```

Out[18]:

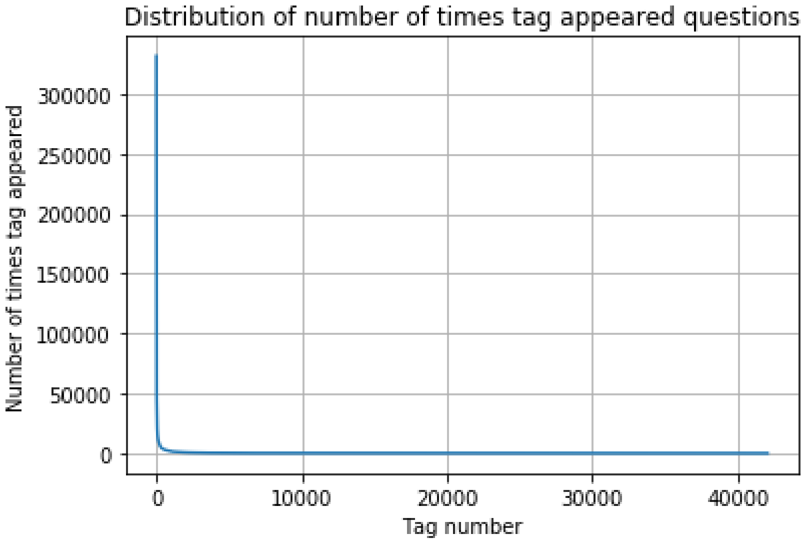
	Tags	Counts
0	hyper-v-server-2008-r2	223
1	rmail	2
2	django-generic-views	109
3	asm.js	1
4	menhir	1

In [19]:

```
tag_df_sorted = tag_df.sort_values(['Counts'], ascending=False)
tag_counts = tag_df_sorted['Counts'].values
```

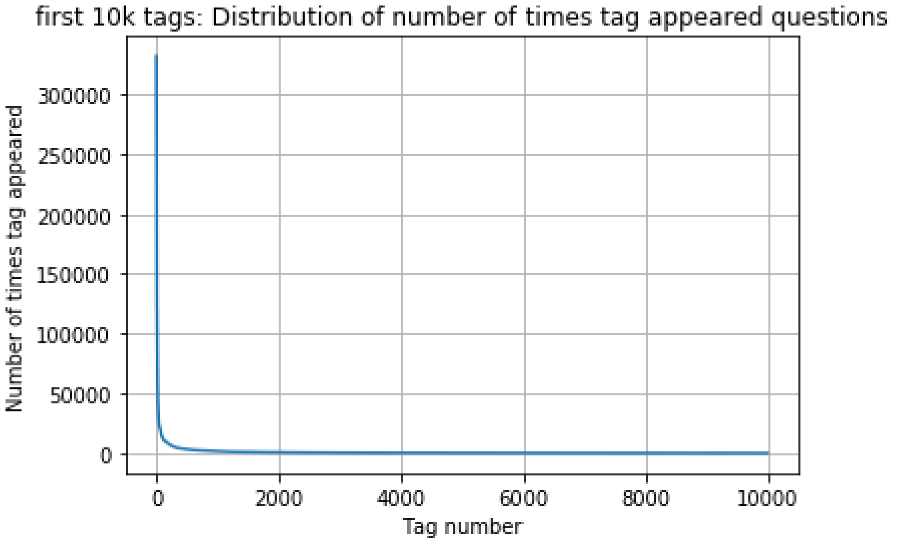
In [20]:

```
plt.plot(tag_counts)
plt.title("Distribution of number of times tag appeared questions")
plt.grid()
plt.xlabel("Tag number")
plt.ylabel("Number of times tag appeared")
plt.show()
```



In [21]:

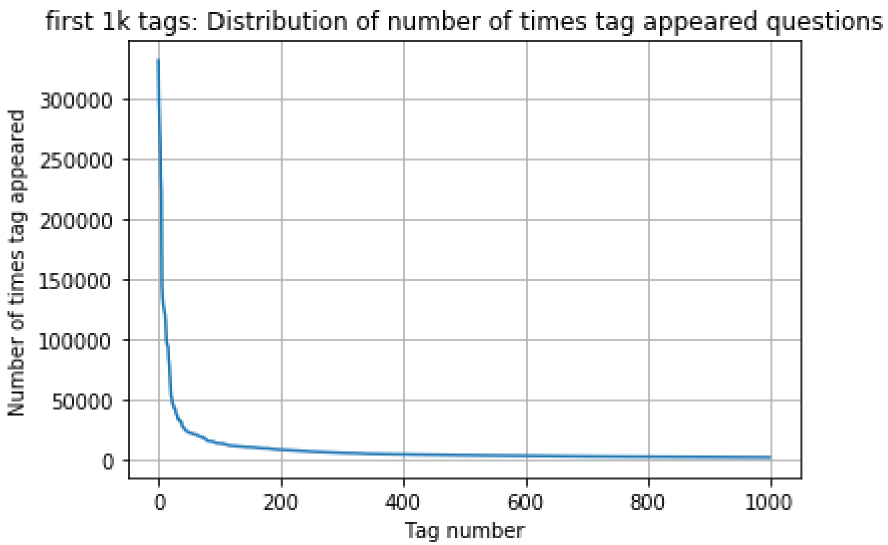
```
plt.plot(tag_counts[0:10000])
plt.title('first 10k tags: Distribution of number of times tag appeared questions')
plt.grid()
plt.xlabel("Tag number")
plt.ylabel("Number of times tag appeared")
plt.show()
print(len(tag_counts[0:10000:25]), tag_counts[0:10000:25])
```



```
400 [331505  44829  22429  17728  13364  11162  10029   9148   8054   7
151
   6466   5865   5370   4983   4526   4281   4144   3929   3750   3593
   3453   3299   3123   2986   2891   2738   2647   2527   2431   2331
   2259   2186   2097   2020   1959   1900   1828   1770   1723   1673
   1631   1574   1532   1479   1448   1406   1365   1328   1300   1266
   1245   1222   1197   1181   1158   1139   1121   1101   1076   1056
   1038   1023   1006    983    966    952    938    926    911    891
    882    869    856    841    830    816    804    789    779    770
    752    743    733    725    712    702    688    678    671    658
    650    643    634    627    616    607    598    589    583    577
    568    559    552    545    540    533    526    518    512    506
    500    495    490    485    480    477    469    465    457    450
    447    442    437    432    426    422    418    413    408    403
    398    393    388    385    381    378    374    370    367    365
    361    357    354    350    347    344    342    339    336    332
    330    326    323    319    315    312    309    307    304    301
    299    296    293    291    289    286    284    281    278    276
    275    272    270    268    265    262    260    258    256    254
    252    250    249    247    245    243    241    239    238    236
    234    233    232    230    228    226    224    222    220    219
    217    215    214    212    210    209    207    205    204    203
    201    200    199    198    196    194    193    192    191    189
    188    186    185    183    182    181    180    179    178    177
    175    174    172    171    170    169    168    167    166    165
    164    162    161    160    159    158    157    156    156    155
    154    153    152    151    150    149    149    148    147    146
    145    144    143    142    142    141    140    139    138    137
    137    136    135    134    134    133    132    131    130    130
    129    128    128    127    126    126    125    124    124    123
    123    122    122    121    120    120    119    118    118    117
    117    116    116    115    115    114    113    113    112    111
    111    110    109    109    108    108    107    106    106    106
    105    105    104    104    103    103    102    102    101    101
    100    100     99     99     98     98     97     97     96     96
     95     95     94     94     93     93     93     92     92     91
     91     90     90     89     89     88     88     87     87     86
     86     86     85     85     84     84     83     83     83     82
     82     82     81     81     80     80     80     79     79     78
     78     78     78     77     77     76     76     76     75     75
     75     74     74     74     73     73     73     73     72     72]
```

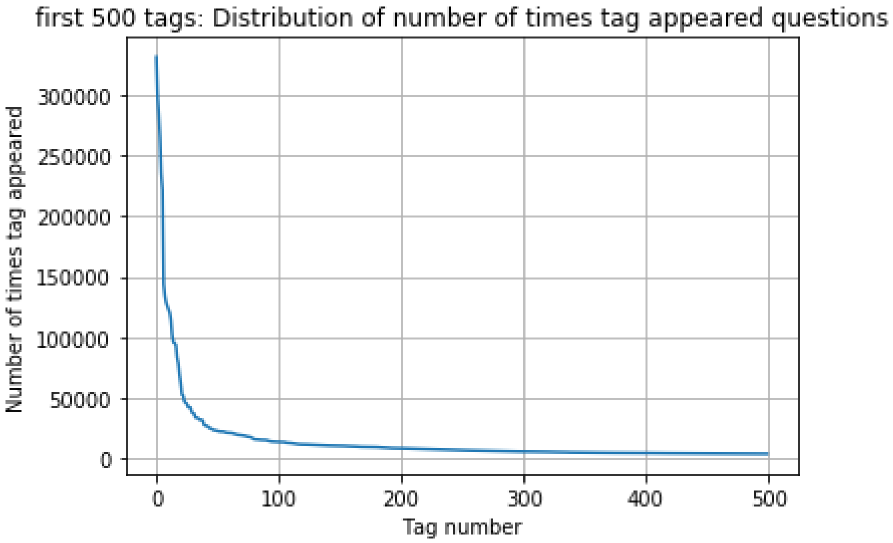
In [22]:

```
plt.plot(tag_counts[0:1000])
plt.title('first 1k tags: Distribution of number of times tag appeared questions')
plt.grid()
plt.xlabel("Tag number")
plt.ylabel("Number of times tag appeared")
plt.show()
print(len(tag_counts[0:1000:5]), tag_counts[0:1000:5])
```



```
200 [331505 221533 122769  95160  62023  44829  37170  31897  26925  24
537
  22429  21820  20957  19758  18905  17728  15533  15097  14884  13703
  13364  13157  12407  11658  11228  11162  10863  10600  10350  10224
  10029   9884   9719   9411   9252   9148   9040   8617   8361   8163
   8054   7867   7702   7564   7274   7151   7052   6847   6656   6553
   6466   6291   6183   6093   5971   5865   5760   5577   5490   5411
   5370   5283   5207   5107   5066   4983   4891   4785   4658   4549
   4526   4487   4429   4335   4310   4281   4239   4228   4195   4159
   4144   4088   4050   4002   3957   3929   3874   3849   3818   3797
   3750   3703   3685   3658   3615   3593   3564   3521   3505   3483
   3453   3427   3396   3363   3326   3299   3272   3232   3196   3168
   3123   3094   3073   3050   3012   2986   2983   2953   2934   2903
   2891   2844   2819   2784   2754   2738   2726   2708   2681   2669
   2647   2621   2604   2594   2556   2527   2510   2482   2460   2444
   2431   2409   2395   2380   2363   2331   2312   2297   2290   2281
   2259   2246   2222   2211   2198   2186   2162   2142   2132   2107
   2097   2078   2057   2045   2036   2020   2011   1994   1971   1965
   1959   1952   1940   1932   1912   1900   1879   1865   1855   1841
   1828   1821   1813   1801   1782   1770   1760   1747   1741   1734
   1723   1707   1697   1688   1683   1673   1665   1656   1646   1639]
```

In [23]:
```
plt.plot(tag_counts[0:500])
plt.title('first 500 tags: Distribution of number of times tag appeared
 questions')
plt.grid()
plt.xlabel("Tag number")
plt.ylabel("Number of times tag appeared")
plt.show()
print(len(tag_counts[0:500:5]), tag_counts[0:500:5])
```



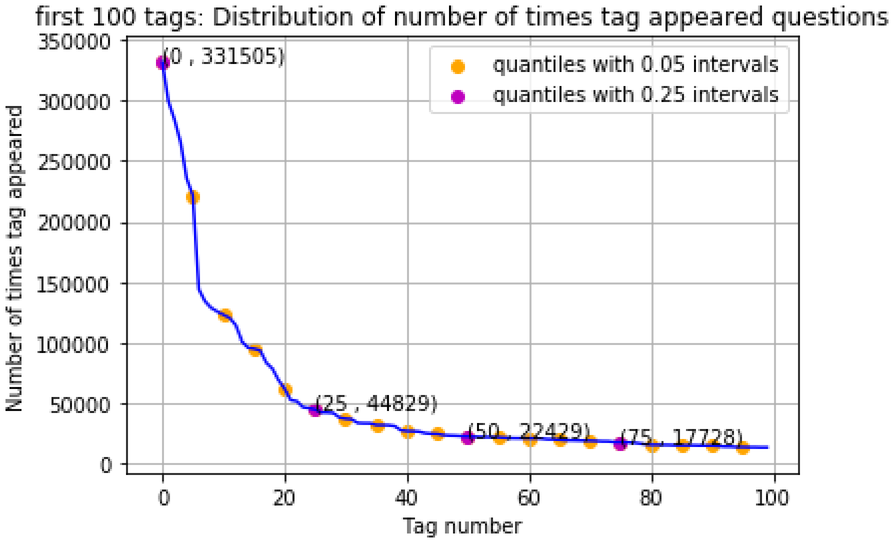
```
100 [331505 221533 122769  95160  62023  44829  37170  31897  26925  24
537
  22429  21820  20957  19758  18905  17728  15533  15097  14884  13703
  13364  13157  12407  11658  11228  11162  10863  10600  10350  10224
  10029   9884   9719   9411   9252   9148   9040   8617   8361   8163
   8054   7867   7702   7564   7274   7151   7052   6847   6656   6553
   6466   6291   6183   6093   5971   5865   5760   5577   5490   5411
   5370   5283   5207   5107   5066   4983   4891   4785   4658   4549
   4526   4487   4429   4335   4310   4281   4239   4228   4195   4159
   4144   4088   4050   4002   3957   3929   3874   3849   3818   3797
   3750   3703   3685   3658   3615   3593   3564   3521   3505   3483]
```

In [24]:

```
plt.plot(tag_counts[0:100], c='b')
plt.scatter(x=list(range(0,100,5)), y=tag_counts[0:100:5], c='orange',
label="quantiles with 0.05 intervals")
# quantiles with 0.25 difference
plt.scatter(x=list(range(0,100,25)), y=tag_counts[0:100:25], c='m', lab
el = "quantiles with 0.25 intervals")

for x,y in zip(list(range(0,100,25)), tag_counts[0:100:25]):
    plt.annotate(s="({} , {})".format(x,y), xy=(x,y), xytext=(x-0.05, y
+500))

plt.title('first 100 tags: Distribution of number of times tag appeared
 questions')
plt.grid()
plt.xlabel("Tag number")
plt.ylabel("Number of times tag appeared")
plt.legend()
plt.show()
print(len(tag_counts[0:100:5]), tag_counts[0:100:5])
```



```
20 [331505 221533 122769  95160  62023  44829  37170  31897  26925  245
37
  22429  21820  20957  19758  18905  17728  15533  15097  14884  13703]
```

```
 22429  21820  20957  19758  18905  17728  15533  15097  14884  13703]
```

In [25]:

```
# Store tags greater than 10K in one list
lst_tags_gt_10k = tag_df[tag_df.Counts>10000].Tags
#Print the length of the list
print ('{} Tags are used more than 10000 times'.format(len(lst_tags_gt_
10k)))
# Store tags greater than 100K in one list
lst_tags_gt_100k = tag_df[tag_df.Counts>100000].Tags
#Print the length of the list.
print ('{} Tags are used more than 100000 times'.format(len(lst_tags_gt
_100k)))
```

```
153 Tags are used more than 10000 times
14 Tags are used more than 100000 times
```

Observations:

1. There are total 153 tags which are used more than 10000 times.
2. 14 tags are used more than 100000 times.
3. Most frequent tag (i.e. c#) is used 331505 times.
4. Since some tags occur much more frequenctly than others, Micro-averaged F1-score is the appropriate metric for this probelm.

3.2.4 Tags Per Question

In [26]:

```
#Storing the count of tag in each question in list 'tag_count'
tag_quest_count = tag_dtm.sum(axis=1).tolist()
#Converting list of lists into single list, we will get [[3], [4], [2],
 [2], [3]] and we are converting this to [3, 4, 2, 2, 3]
tag_quest_count=[int(j) for i in tag_quest_count for j in i]
print ('We have total {} datapoints.'.format(len(tag_quest_count)))

print(tag_quest_count[:5])
```

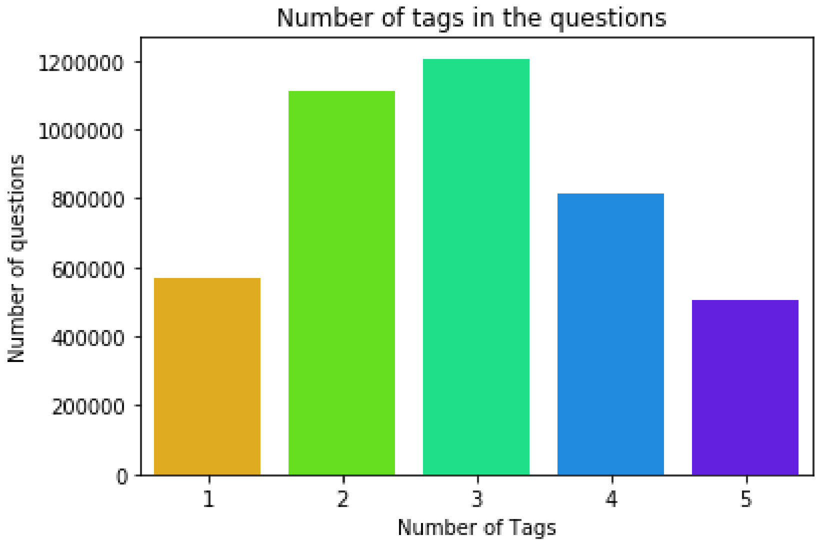
```
We have total 4206307 datapoints.
```

```
[3, 4, 2, 2, 3]
```

In [27]:
```
print( "Maximum number of tags per question: %d"%max(tag_quest_count))
print( "Minimum number of tags per question: %d"%min(tag_quest_count))
print( "Avg. number of tags per question: %f"% ((sum(tag_quest_count)*
1.0)/len(tag_quest_count)))
```

```
Maximum number of tags per question: 5
Minimum number of tags per question: 1
Avg. number of tags per question: 2.899443
```

In [28]:
```
sns.countplot(tag_quest_count, palette='gist_rainbow')
plt.title("Number of tags in the questions ")
plt.xlabel("Number of Tags")
plt.ylabel("Number of questions")
plt.show()
```



Observations:

1. Maximum number of tags per question: 5
2. Minimum number of tags per question: 1

3. Avg. number of tags per question: 2.899
4. Most of the questions are having 2 or 3 tags

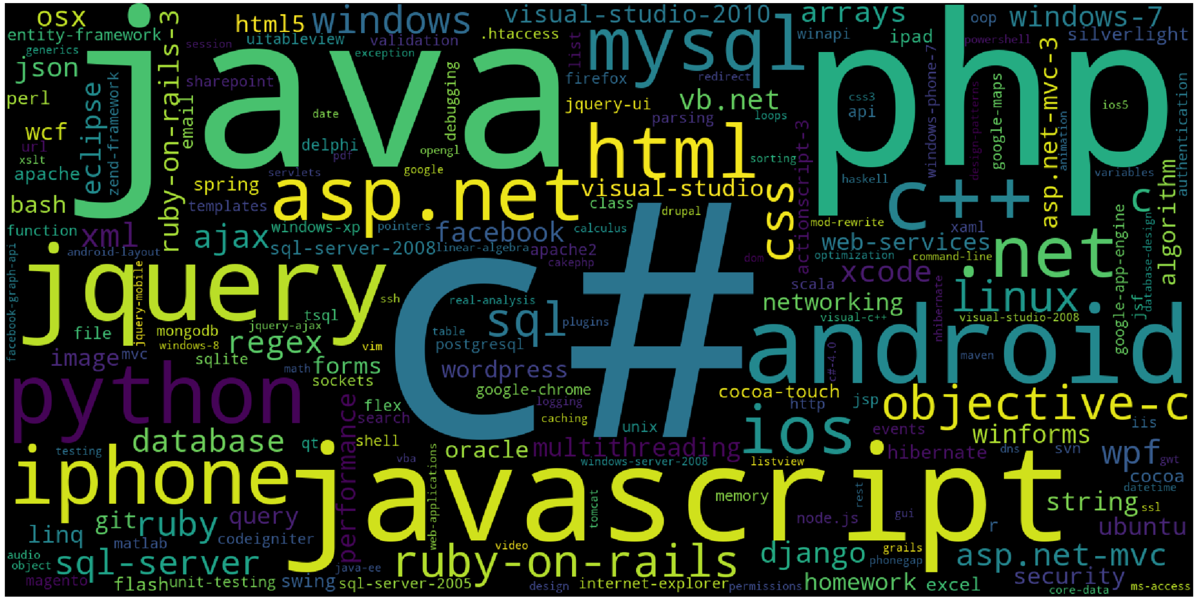
3.2.5 Most Frequent Tags

In [29]:

```
# Ploting word cloud
start = datetime.now()

# Lets first convert the 'result' dictionary to 'list of tuples'
tup = dict(result.items())
#Initializing WordCloud using frequencies of tags.
wordcloud = WordCloud(    background_color='black',
                          width=1600,
                          height=800,
                    ).generate_from_frequencies(tup)

fig = plt.figure(figsize=(30,20))
plt.imshow(wordcloud)
plt.axis('off')
plt.tight_layout(pad=0)
fig.savefig("tag.png")
plt.show()
print("Time taken to run this cell :", datetime.now() - start)
```

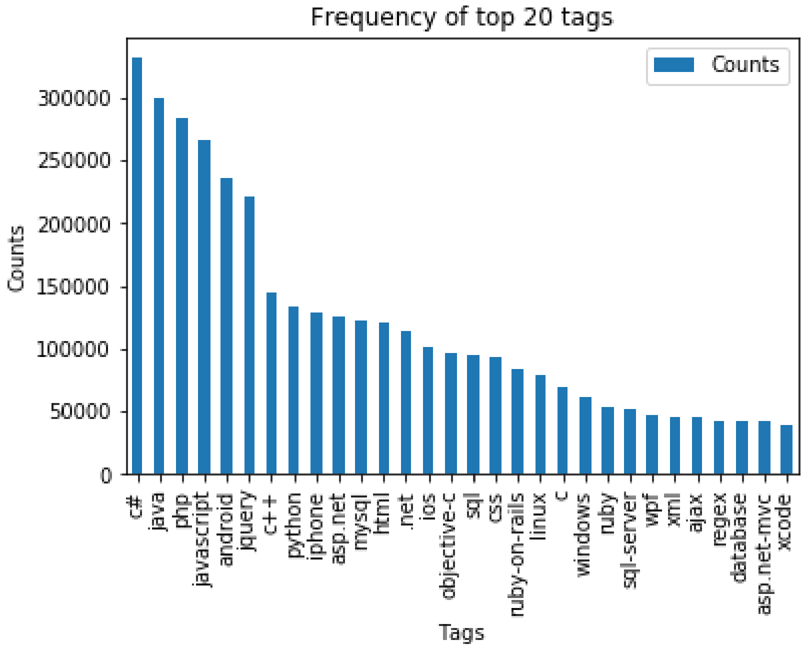


```
Time taken to run this cell : 0:00:04.672499
```

Observations:
A look at the word cloud shows that "c#", "java", "php", "asp.net", "javascript", "c++" are some of the most frequent tags.

3.2.6 The top 20 tags

In [30]:
```
i=np.arange(30)
tag_df_sorted.head(30).plot(kind='bar')
plt.title('Frequency of top 20 tags')
plt.xticks(i, tag_df_sorted['Tags'])
plt.xlabel('Tags')
plt.ylabel('Counts')
plt.show()
```



Observations:

1. Majority of the most frequent tags are programming language.
2. C# is the top most frequent programming language.
3. Android, IOS, Linux and windows are among the top most frequent operating systems.

3.3 Cleaning and preprocessing of Questions

3.3.1 Preprocessing

1. Sample 1M data points
2. Separate out code-snippets from Body
3. Remove Spcial characters from Question title and description (not in code)

4. Remove stop words (Except 'C')
5. Remove HTML Tags
6. Convert all the characters into small letters
7. Use SnowballStemmer to stem the words

In [31]:

```python
def striphtml(data):
    cleanr = re.compile('<.*?>')
    cleantext = re.sub(cleanr, ' ', str(data))
    return cleantext
stop_words = set(stopwords.words('english'))
stemmer = SnowballStemmer("english")
```

In [32]:

```python
#http://www.sqlitetutorial.net/sqlite-python/create-tables/
def create_connection(db_file):
    """ create a database connection to the SQLite database
        specified by db_file
    :param db_file: database file
    :return: Connection object or None
    """
    try:
        conn = sqlite3.connect(db_file)
        return conn
    except Error as e:
        print(e)

    return None

def create_table(conn, create_table_sql):
    """ create a table from the create_table_sql statement
    :param conn: Connection object
    :param create_table_sql: a CREATE TABLE statement
    :return:
    """
    try:
        c = conn.cursor()
        c.execute(create_table_sql)
    except Error as e:
        print(e)

def checkTableExists(dbcon):
    cursr = dbcon.cursor()
    str = "select name from sqlite_master where type='table'"
    table_names = cursr.execute(str)
    print("Tables in the databse:")
    tables =table_names.fetchall()
    print(tables[0][0])
    return(len(tables))

def create_database_table(database, query):
    conn = create_connection(database)
    if conn is not None:
        create_table(conn, query)
        checkTableExists(conn)
    else:
        print("Error! cannot create the database connection.")
    conn.close()

sql_create_table = """CREATE TABLE IF NOT EXISTS QuestionsProcessed (qu
estion text NOT NULL, code text, tags text, words_pre integer, words_po
st integer, is_code integer);"""
create_database_table("Processed.db", sql_create_table)
```

```
Tables in the databse:
QuestionsProcessed
```

In [33]:

```
# http://www.sqlitetutorial.net/sqlite-delete/
# https://stackoverflow.com/questions/2279706/select-random-row-from-a-
sqlite-table
start = datetime.now()
read_db = 'train_no_dup.db'
write_db = 'Processed.db'
if os.path.isfile(read_db):
    conn_r = create_connection(read_db)
    if conn_r is not None:
        reader =conn_r.cursor()
        reader.execute("SELECT Title, Body, Tags From no_dup_train ORDE
R BY RANDOM() LIMIT 100000;")

if os.path.isfile(write_db):
    conn_w = create_connection(write_db)
    if conn_w is not None:
        tables = checkTableExists(conn_w)
        writer =conn_w.cursor()
        if tables != 0:
            writer.execute("DELETE FROM QuestionsProcessed WHERE 1")
            print("Cleared All the rows")
print("Time taken to run this cell :", datetime.now() - start)
```

```
Tables in the databse:
QuestionsProcessed
Cleared All the rows
Time taken to run this cell : 0:00:30.674447
```

we create a new data base to store the sampled and preprocessed questions

In [34]:

```
#http://www.bernzilla.com/2008/05/13/selecting-a-random-row-from-an-sql
ite-table/

start = datetime.now()
preprocessed_data_list=[]
reader.fetchone()
questions_with_code=0
len_pre=0
len_post=0
questions_proccesed = 0
for row in reader:

    is_code = 0

    title, question, tags = row[0], row[1], row[2]

    if '<code>' in question:
        questions_with_code+=1
        is_code = 1
    x = len(question)+len(title)
    len_pre+=x

    code = str(re.findall(r'<code>(.*?)</code>', question, flags=re.DOT
ALL))

    question=re.sub('<code>(.*?)</code>', '', question, flags=re.MULTIL
INE|re.DOTALL)
    question=striphtml(question.encode('utf-8'))

    title=title.encode('utf-8')

    question=str(title)+" "+str(question)
    question=re.sub(r'[^A-Za-z]+',' ',question)
    words=word_tokenize(str(question.lower()))

    #Removing all single letter and and stopwords from question exceptt
 for the letter 'c'
    question=' '.join(str(stemmer.stem(j)) for j in words if j not in s
top_words and (len(j)!=1 or j=='c'))

    len_post+=len(question)
    tup = (question,code,tags,x,len(question),is_code)
    questions_proccesed += 1
    writer.execute("insert into QuestionsProcessed(question,code,tags,w
ords_pre,words_post,is_code) values (?,?,?,?,?,?)",tup)
    if (questions_proccesed%100000==0):
        print("number of questions completed=",questions_proccesed)

no_dup_avg_len_pre=(len_pre*1.0)/questions_proccesed
no_dup_avg_len_post=(len_post*1.0)/questions_proccesed

print( "Avg. length of questions(Title+Body) before processing: %d"%no_
dup_avg_len_pre)
print( "Avg. length of questions(Title+Body) after processing: %d"%no_d
up_avg_len_post)
print ("Percent of questions containing code: %d"%((questions_with_code
*100.0)/questions_proccesed))
```

```
print("Time taken to run this cell :", datetime.now() - start)
```

```
Avg. length of questions(Title+Body) before processing: 1173
Avg. length of questions(Title+Body) after processing: 327
Percent of questions containing code: 57
Time taken to run this cell : 0:02:25.423838
```

In [35]:

```
# dont forget to close the connections, or else you will end up with lo
cks
conn_r.commit()
conn_w.commit()
conn_r.close()
conn_w.close()
```

In [36]:

```
#Taking 1 Million entries to a dataframe.
write_db = 'Processed.db'
if os.path.isfile(write_db):
    conn_r = create_connection(write_db)
    if conn_r is not None:
        reader =conn_r.cursor()
        reader.execute("SELECT question From QuestionsProcessed LIMIT 1
0")
        print("Questions after preprocessed")
        print('='*100)
        reader.fetchone()
        for row in reader:
            print(row)
            print('-'*100)
    conn_r.commit()
    conn_r.close()
```

```
Questions after preprocessed
=======================================================================
=============================
('check variabl type primit mayb question bit stupid know check variabl
primit java like thank',)
-----------------------------------------------------------------------
-----------------------------
('not missingmethodexcept occur one thousand and user machin insight is
('net missingmethodexcept occur one thousand end user machin insight is
su baffl affect singl user knowledg reproduc us user receiv missingmeth
odexcept trace file indic occur creat new instanc compon call initi set
up method prepar work initializeworkerbyargu exampl method specifi erro
r interfac method base class implement class deriv base class overrid n
eed user latest releas applic provid code ship within singl assembl dis
til version compon',)
-----------------------------------------------------------------------
-----------------------------
('updat listview delet oper display data pull android os sqlite databas
success get item delet click howev issu refresh updat listview oper cod
e delet contact updatelist method includ method three way tri refresh n
one work idea might achiev edit chang code think would solut work eithe
r dbadapt class delet method getallcontactslist thought get new cursor
delet contact would updat list accord unfortun made differ idea',)
-----------------------------------------------------------------------
-----------------------------
('specifi visual studio instal condit visual studio instal project want
instal creat specif folder check box checkbox form ad project ui check
name properti checkbox checkboxa idea put condit properti folder get cr
eat checkbox check',)
-----------------------------------------------------------------------
-----------------------------
('stop black box oper use asynchron deleg invok method load xml file xp
athdocu xml big fit memori never finish load code work xml file success
load xpathdocu abl use timer event execut asyncxpath endinvok result st
atement work end createdocu method stop xpathdocu load conclus thing is
su applic end statement kill applic anyon know stop blackbox oper load
xpathdocu',)
-----------------------------------------------------------------------
-----------------------------
('mirror drive extent found plex instal window ultim system two disk dr
ive window partit disk mb system reserv rest c incl boot page file cras
h dump want mirror disk setup disk mirror disk window softwar raid chan
g disk dynam disk add mirror disk surpris grab hunk disk mirror c mirro
r mb system well good disk die want abl boot disk put new disk resync b
ack busi minimum fuss click system reserv partit say add mirror ask sel
ect disk select disk give error extent found plex disk look like though
t mayb round issu perhap even shrink system reserv mb still give error
disk complet mirror disk disk die disk readi run',)
```

```
----------------------------------------------------------------------
-----------------------------
('jqueri pagin plugin asp net gridview implement pagin asp gridview num
ber record retriev databas use jqueri pagin plug implement',)
----------------------------------------------------------------------
-----------------------------
('apach ubuntu return known caus note question resolv pleas see solut c
ase similar issu background php mysql base databas driven applic make h
eavi use ajax near request server occur way request get request instanc
server respond via json user work inward data set return finer finer re
sult applic stabl product abil debug platform develop laptop osx hous u
buntu server product ubuntu server ubuntu server run mysql server copi
apach develop laptop connect mysql server run hous ubuntu server run ap
ach server ubuntu server fulli date via offici sourc apach php fun part
applic return result correct case except one subsect data set request d
ata within subsect alway return instead result set subsect question mer
e queri return result base known id record known exist contain corrupt
data debug log turn apach set php log file occur see follow apach error
log mask impli lack data howev error log php error log whatsoev time re
turn result elsewher data set issu might think problem must databas how
ev use develop laptop copi apach connect mysql server hous network run
request generat error run applic apach server laptop everyth cool perfo
rm request data within databas databas server instead use server local
copi apach fail utter baffl help would great appreci point',)
----------------------------------------------------------------------
-----------------------------
('remov work copi creat via git new workdir without hose origin repo us
e git new workdir script present contrib section git codebas https gith
ub com git git tree master contrib workdir work multipl branch code bas
e simultan window use msysgit repo git svn repo pure git ni problem cre
at work copi use command say longer interest branch want get rid work c
opi also hose origin work copi hind sight kinda obvious given git new w
orkdir use hard link share git repo multipl work copi good way clean wo
rk copi creat way longer want machin',)
----------------------------------------------------------------------
-----------------------------
```

```
In [37]: if os.path.isfile(write_db):
    conn_r = create_connection(write_db)
    if conn_r is not None:
        preprocessed_data = pd.read_sql_query("""SELECT question, Tags
FROM QuestionsProcessed""", conn_r)
    conn_r.commit()
    conn_r.close()
```

In [38]:

```
preprocessed_data.head()
```

Out[38]:

	question	tags
0	inherit parent constructor argument brows discuss similar topic find situat tri call parent cons...	javascript inheritance constructor arguments
1	check variabl type primit mayb question bit stupid know check variabl primit java like thank	python
2	net missingmethodexcept occur one thousand end user machin insight issu baffl affect singl user ...	.net exception missingmethodexception
3	updat listview delet oper display data pull android os sqlite databas success get item delet cli...	android sqlite android-listview
4	specifi visual studio instal condit visual studio instal project want instal creat specif folder...	visual-studio-2008 windows-installer

In [39]:

```
print("number of data points in sample :", preprocessed_data.shape[0])
print("number of dimensions :", preprocessed_data.shape[1])
```

```
number of data points in sample : 99999
number of dimensions : 2
```

4. Machine Learning Models

In [40]:

```
global_report = pd.DataFrame(columns=['Vectorizer', 'Model', 'NGram',
'Parameter', 'Precision', 'Recall', 'F1_Score_Micro'])
```

4.1 Converting tags for multilabel problems

X	y1	y2	y3	y4
x1	0	1	1	0
x1	1	0	0	0
x1	0	1	0	0

In [41]:

```
# binary='true' will give a binary vectorizer
vectorizer = CountVectorizer(tokenizer = lambda x: x.split(), binary='true')
multilabel_y = vectorizer.fit_transform(preprocessed_data['tags'])
```

We will sample the number of tags instead considering all of them (due to limitation of computing power)

In [42]:

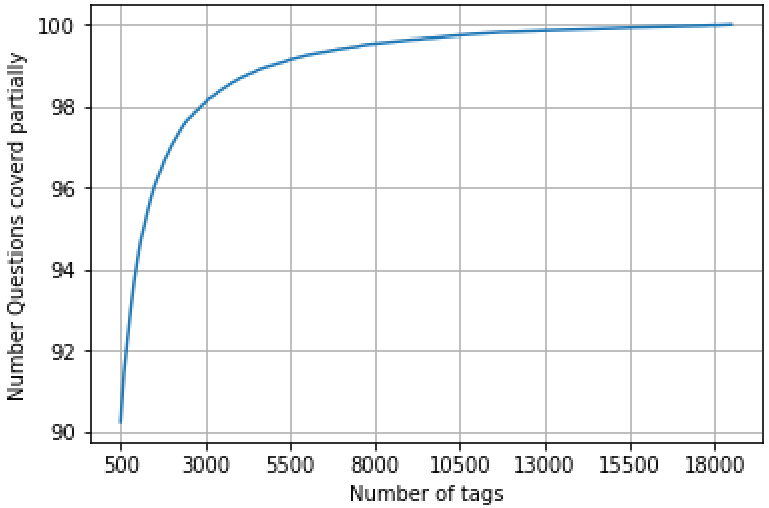
```
def tags_to_choose(n):
    t = multilabel_y.sum(axis=0).tolist()[0]
    sorted_tags_i = sorted(range(len(t)), key=lambda i: t[i], reverse=True)
    multilabel_yn=multilabel_y[:,sorted_tags_i[:n]]
    return multilabel_yn

def questions_explained_fn(n):
    multilabel_yn = tags_to_choose(n)
    x= multilabel_yn.sum(axis=1)
    return (np.count_nonzero(x==0))
```

In [43]:

```
questions_explained = []
total_tags=multilabel_y.shape[1]
total_qs=preprocessed_data.shape[0]
for i in range(500, total_tags, 100):
    questions_explained.append(np.round(((total_qs-questions_explained_fn(i))/total_qs)*100,3))
```

In [44]:
```
fig, ax = plt.subplots()
ax.plot(questions_explained)
xlabel = list(500+np.array(range(-50,450,50))*50)
ax.set_xticklabels(xlabel)
plt.xlabel("Number of tags")
plt.ylabel("Number Questions coverd partially")
plt.grid()
plt.show()
# you can choose any number of tags based on your computing power, mini
mun is 50(it covers 90% of the tags)
print("with ",5500,"tags we are covering ",questions_explained[50],"% o
f questions")
```



```
with  5500 tags we are covering  99.154 % of questions
```

In [45]:
```
multilabel_yx = tags_to_choose(5500)
print("number of questions that are not covered :", questions_explained
_fn(5500),"out of ", total_qs)
```

```
number of questions that are not covered : 846 out of  99999
```

In [46]:
```
print("Number of tags in sample :", multilabel_y.shape[1])
print("number of tags taken :", multilabel_yx.shape[1],"(",(multilabel_
yx.shape[1]/multilabel_y.shape[1])*100,"%)")
```

```
Number of tags in sample : 18587
number of tags taken : 5500 ( 29.59057405713671 %)
```

We consider top 15% tags which covers 99% of the questions

4.2 Split the data into test and train (80:20)

In [47]:

```
total_size=preprocessed_data.shape[0]
train_size=int(0.80*total_size)

x_train=preprocessed_data.head(train_size)
x_test=preprocessed_data.tail(total_size - train_size)

y_train = multilabel_yx[0:train_size,:]
y_test = multilabel_yx[train_size:total_size,:]
```

In [48]:

```
print("Number of data points in train data :", y_train.shape)
print("Number of data points in test data :", y_test.shape)
```

```
Number of data points in train data : (79999, 5500)
Number of data points in test data : (20000, 5500)
```

4.3 Featurizing data

In [49]:

```
# start = datetime.now()
# vectorizer = TfidfVectorizer(min_df=0.00009, max_features=200000, smo
oth_idf=True, norm="l2", \
#                               tokenizer = lambda x: x.split(), subline
ar_tf=False, ngram_range=(1,3))
# x_train_multilabel = vectorizer.fit_transform(x_train['question'])
# x_test_multilabel = vectorizer.transform(x_test['question'])
# print("Time taken to run this cell :", datetime.now() - start)
```

In [50]:
```
# print("Dimensions of train data X:",x_train_multilabel.shape, "Y :",y_train.shape)
# print("Dimensions of test data X:",x_test_multilabel.shape,"Y:",y_test.shape)
```

In [51]:
```
# https://www.analyticsvidhya.com/blog/2017/08/introduction-to-multi-label-classification/
#https://stats.stackexchange.com/questions/117796/scikit-multi-label-classification
# classifier = LabelPowerset(GaussianNB())
"""
from skmultilearn.adapt import MLkNN
classifier = MLkNN(k=21)

# train
classifier.fit(x_train_multilabel, y_train)

# predict
predictions = classifier.predict(x_test_multilabel)
print(accuracy_score(y_test,predictions))
print(metrics.f1_score(y_test, predictions, average = 'macro'))
print(metrics.f1_score(y_test, predictions, average = 'micro'))
print(metrics.hamming_loss(y_test,predictions))

"""
# we are getting memory error because the multilearn package
# is trying to convert the data into dense matrix
# -------------------------------------------------------------------------
#MemoryError                              Traceback (most recent call last)
#<ipython-input-170-f0e7c7f3e0be> in <module>()
#----> classifier.fit(x_train_multilabel, y_train)
```

Out[51]:
```
"\nfrom skmultilearn.adapt import MLkNN\nclassifier = MLkNN(k=21)\n\n# train\nclassifier.fit(x_train_multilabel, y_train)\n\n# predict\npredictions = classifier.predict(x_test_multilabel)\nprint(accuracy_score(y_test,predictions))\nprint(metrics.f1_score(y_test, predictions, average = 'macro'))\nprint(metrics.f1_score(y_test, predictions, average = 'mic
ro'))\nprint(metrics.hamming_loss(y_test,predictions))\n\n"
```

4.4 Applying Logistic Regression with OneVsRest Classifier

In [52]:

```
# # this will be taking so much time try not to run it, download the lr_with_equal_weight.pkl file and use to predict
# # This takes about 6-7 hours to run.
# classifier = OneVsRestClassifier(SGDClassifier(loss='log', alpha=0.00001, penalty='l1'), n_jobs=-1)
# classifier.fit(x_train_multilabel, y_train)
# predictions = classifier.predict(x_test_multilabel)

# print("accuracy :",metrics.accuracy_score(y_test,predictions))
# print("macro f1 score :",metrics.f1_score(y_test, predictions, average = 'macro'))
# print("micro f1 scoore :",metrics.f1_score(y_test, predictions, average = 'micro'))
# print("hamming loss :",metrics.hamming_loss(y_test,predictions))
# print("Precision recall report :\n",metrics.classification_report(y_test, predictions))
```

In [53]:

```
# from sklearn.externals import joblib
# joblib.dump(classifier, 'lr_with_equal_weight.pkl')
```

4.5 Modeling with less data points (0.15M data points) and more weight to title and 500 tags only.

In [54]:

```
sql_create_table = """CREATE TABLE IF NOT EXISTS QuestionsProcessed (question text NOT NULL, code text, tags text, words_pre integer, words_post integer, is_code integer);"""
create_database_table("Titlemoreweight.db", sql_create_table)
```

```
Tables in the databse:
QuestionsProcessed
```

In [55]:
```python
# http://www.sqlitetutorial.net/sqlite-delete/
# https://stackoverflow.com/questions/2279706/select-random-row-from-a-sqlite-table

read_db = 'train_no_dup.db'
write_db = 'Titlemoreweight.db'
# train_datasize = 100000
if os.path.isfile(read_db):
    conn_r = create_connection(read_db)
    if conn_r is not None:
        reader =conn_r.cursor()
        # for selecting first 0.5M rows
        reader.execute("SELECT Title, Body, Tags From no_dup_train LIMIT 150001;")
        # for selecting random points
        #reader.execute("SELECT Title, Body, Tags From no_dup_train ORDER BY RANDOM() LIMIT 500001;")

if os.path.isfile(write_db):
    conn_w = create_connection(write_db)
    if conn_w is not None:
        tables = checkTableExists(conn_w)
        writer =conn_w.cursor()
        if tables != 0:
            writer.execute("DELETE FROM QuestionsProcessed WHERE 1")
            print("Cleared All the rows")
```

```
Tables in the databse:
QuestionsProcessed
Cleared All the rows
```

4.5.1 Preprocessing of questions

1. Separate Code from Body
2. Remove Spcial characters from Question title and description (not in code)
3. **Give more weightage to title : Add title three times to the question**

4. Remove stop words (Except 'C')
5. Remove HTML Tags
6. Convert all the characters into small letters
7. Use SnowballStemmer to stem the words

In [56]:

```
#http://www.bernzilla.com/2008/05/13/selecting-a-random-row-from-an-sqlite-table/
start = datetime.now()
preprocessed_data_list=[]
reader.fetchone()
questions_with_code=0
len_pre=0
len_post=0
questions_proccesed = 0
for row in reader:

    is_code = 0

    title, question, tags = row[0], row[1], str(row[2])

    if '<code>' in question:
        questions_with_code+=1
        is_code = 1
    x = len(question)+len(title)
    len_pre+=x

    code = str(re.findall(r'<code>(.*?)</code>', question, flags=re.DOTALL))

    question=re.sub('<code>(.*?)</code>', '', question, flags=re.MULTILINE|re.DOTALL)
    question=striphtml(question.encode('utf-8'))

    title=title.encode('utf-8')

    # adding title three time to the data to increase its weight
    # add tags string to the training data
    question=str(title)+" "+str(title)+" "+str(title)+" "+question

#     if questions_proccesed<=train_datasize:
#         question=str(title)+" "+str(title)+" "+str(title)+" "+questio
n+" "+str(tags)
#     else:
#         question=str(title)+" "+str(title)+" "+str(title)+" "+questio
n

    question=re.sub(r'[^A-Za-z0-9#+.\-]+',' ',question)
    words=word_tokenize(str(question.lower()))

    #Removing all single letter and and stopwords from question exceptt
 for the letter 'c'
    question=' '.join(str(stemmer.stem(j)) for j in words if j not in s
top_words and (len(j)!=1 or j=='c'))

    len_post+=len(question)
    tup = (question,code,tags,x,len(question),is_code)
    questions_proccesed += 1
    writer.execute("insert into QuestionsProcessed(question,code,tags,w
ords_pre,words_post,is_code) values (?,?,?,?,?,?)",tup)
    if (questions_proccesed%100000==0):
        print("number of questions completed=",questions_proccesed)

no_dup_avg_len_pre=(len_pre*1.0)/questions_proccesed
no_dup_avg_len_post=(len_post*1.0)/questions_proccesed

print( "Avg. length of questions(Title+Body) before processing: %d"%no_
dup_avg_len_pre)
print( "Avg. length of questions(Title+Body) after processing: %d"%no_d
up_avg_len_post)
print ("Percent of questions containing code: %d"%((questions_with_code
*100.0)/questions_proccesed))

print("Time taken to run this cell :", datetime.now() - start)
```

```
number of questions completed= 100000
Avg. length of questions(Title+Body) before processing: 1222
Avg. length of questions(Title+Body) after processing: 430
```

```
Percent of questions containing code: 57
Time taken to run this cell : 0:05:33.648006
```

In [57]:

```
# never forget to close the conections or else we will end up with data
base locks
conn_r.commit()
conn_w.commit()
conn_r.close()
conn_w.close()
```

Sample quesitons after preprocessing of data

In [58]:

```
if os.path.isfile(write_db):
    conn_r = create_connection(write_db)
    if conn_r is not None:
        reader =conn_r.cursor()
        reader.execute("SELECT question From QuestionsProcessed LIMIT 1
0")
        print("Questions after preprocessed")
        print('='*100)
        reader.fetchone()
        for row in reader:
            print(row)
            print('-'*100)
conn_r.commit()
conn_r.close()
```

```
Questions after preprocessed
=======================================================================
=============================
('dynam datagrid bind silverlight dynam datagrid bind silverlight dynam
datagrid bind silverlight bind datagrid dynam code wrote code debug cod
e block seem bind correct grid come column form come grid column althou
gh necessari bind nthank repli advance..',)
-----------------------------------------------------------------------
-----------------------------
('java.lang.noclassdeffounderror javax servlet jsp tagext taglibraryval
id java.lang.noclassdeffounderror javax servlet jsp tagext taglibraryva
lid java.lang.noclassdeffounderror javax servlet jsp tagext taglibraryv
```

```
tld java.lang.noclassdeffounderror javax servlet jsp tagext taglibraryv
alid follow guid link instal jstl got follow error tri launch jsp page 
java.lang.noclassdeffounderror javax servlet jsp tagext taglibraryvalid
taglib declar instal jstl 1.1 tomcat webapp tri project work also tri v
ersion 1.2 jstl still messag caus solv',)
------------------------------------------------------------------------
-----------------------------
('java.sql.sqlexcept microsoft odbc driver manag invalid descriptor ind
ex java.sql.sqlexcept microsoft odbc driver manag invalid descriptor in
dex java.sql.sqlexcept microsoft odbc driver manag invalid descriptor i
ndex use follow code display caus solv',)
------------------------------------------------------------------------
-----------------------------
('better way updat feed fb php sdk better way updat feed fb php sdk bet
ter way updat feed fb php sdk novic facebook api read mani tutori still
confused.i find post feed api method like correct second way use curl s
ometh like way better',)
------------------------------------------------------------------------
-----------------------------
('btnadd click event open two window record ad btnadd click event open 
two window record ad btnadd click event open two window record ad open 
window search.aspx use code hav add button search.aspx nwhen insert rec
ord btnadd click event open anoth window nafter insert record close win
dow',)
------------------------------------------------------------------------
-----------------------------
('sql inject issu prevent correct form submiss php sql inject issu prev
ent correct form submiss php sql inject issu prevent correct form submi
ss php check everyth think make sure input field safe type sql inject g
ood news safe bad news one tag mess form submiss place even touch life 
figur exact html use templat file forgiv okay entir php script get exec
ut see data post none forum field post problem use someth titl field no
ne data get post current use print post see submit noth work flawless s
tatement though also mention script work flawless local machin use host
come across problem state list input test mess',)
------------------------------------------------------------------------
-----------------------------
('countabl subaddit lebesgu measur countabl subaddit lebesgu measur cou
ntabl subaddit lebesgu measur let lbrace rbrace sequenc set sigma -alge
bra mathcal want show left bigcup right leq sum left right countabl add
```

```
it measur defin set sigma algebra mathcal think use monoton properti so
mewher proof start appreci littl help nthank ad han answer make follow
addit construct given han answer clear bigcup bigcup cap emptyset neq l
eft bigcup right left bigcup right sum left right also construct subset
monoton left right leq left right final would sum leq sum result follo
w',)
-----------------------------------------------------------------------
-----------------------------
('hql equival sql queri hql equival sql queri hql equival sql queri hql
queri replac name class properti name error occur hql error',)
-----------------------------------------------------------------------
-----------------------------
('undefin symbol architectur i386 objc class skpsmtpmessag referenc err
or undefin symbol architectur i386 objc class skpsmtpmessag referenc er
ror undefin symbol architectur i386 objc class skpsmtpmessag referenc e
rror import framework send email applic background import framework i.e
skpsmtpmessag somebodi suggest get error collect2 ld return exit status
import framework correct sorc taken framework follow mfmailcomposeviewc
ontrol question lock field updat answer drag drop folder project click
copi nthat',)
-----------------------------------------------------------------------
-----------------------------
```

Saving Preprocessed data to a Database

In [59]:

```
#Taking 0.5 Million entries to a dataframe.
write_db = 'Titlemoreweight.db'
if os.path.isfile(write_db):
    conn_r = create_connection(write_db)
    if conn_r is not None:
        preprocessed_data = pd.read_sql_query("""SELECT question, Tags
 FROM QuestionsProcessed""", conn_r)
conn_r.commit()
conn_r.close()
```

In [60]:

```
preprocessed_data.head()
```

Out[60]:

	question	tags
0	dynam datagrid bind silverlight dynam datagrid bind silverlight dynam datagrid bind silverlight ...	c# silverlight data-binding
1	dynam datagrid bind silverlight dynam datagrid bind silverlight dynam datagrid bind silverlight ...	c# silverlight data-binding columns
2	java.lang.noclassdeffounderror javax servlet jsp tagext taglibraryvalid java.lang.noclassdeffoun...	jsp jstl
3	java.sql.sqlexcept microsoft odbc driver manag invalid descriptor index java.sql.sqlexcept micro...	java jdbc
4	better way updat feed fb php sdk better way updat feed fb php sdk better way updat feed fb php s...	facebook api facebook-php-sdk

In [61]:

```
print("number of data points in sample :", preprocessed_data.shape[0])
print("number of dimensions :", preprocessed_data.shape[1])
```

```
number of data points in sample : 150000
number of dimensions : 2
```

Converting string Tags to multilable output variables

In [62]:

```
vectorizer = CountVectorizer(tokenizer = lambda x: x.split(), binary='t
rue')
multilabel_y = vectorizer.fit_transform(preprocessed_data['tags'])
```

Selecting 500 Tags

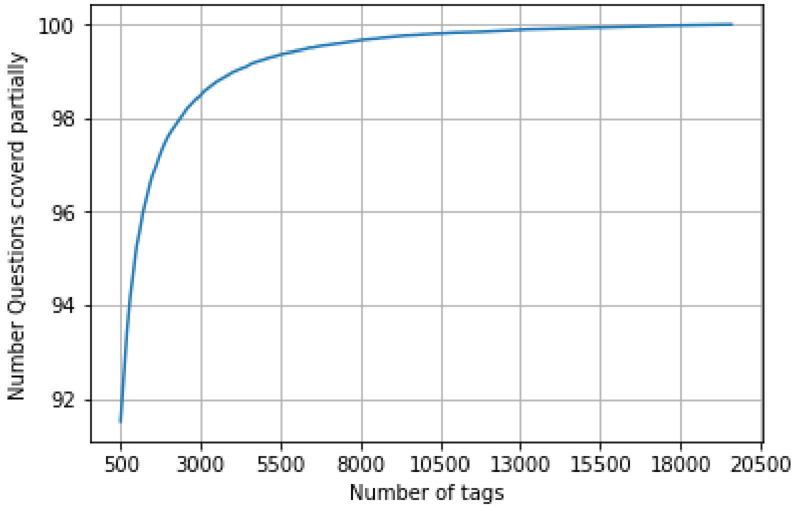
In [63]:

```
questions_explained = []
total_tags=multilabel_y.shape[1]
total_qs=preprocessed_data.shape[0]
for i in range(500, total_tags, 100):
    questions_explained.append(np.round(((total_qs-questions_explained_
fn(i))/total_qs)*100,3))
```

In [64]:

```
fig, ax = plt.subplots()
```

```
ax.plot(questions_explained)
xlabel = list(500+np.array(range(-50,450,50))*50)
ax.set_xticklabels(xlabel)
plt.xlabel("Number of tags")
plt.ylabel("Number Questions coverd partially")
plt.grid()
plt.show()
# you can choose any number of tags based on your computing power, mini
mun is 500(it covers 90% of the tags)
print("with ",5500,"tags we are covering ",questions_explained[50],"% o
f questions")
print("with ",500,"tags we are covering ",questions_explained[0],"% of
 questions")
```



```
with  5500 tags we are covering  99.363 % of questions
with  500 tags we are covering  91.515 % of questions
```

In [65]:
```
# we will be taking 500 tags
multilabel_yx = tags_to_choose(500)
print("number of questions that are not covered :", questions_explained
_fn(500),"out of ", total_qs)
```

```
number of questions that are not covered : 12728 out of  150000
```

In [66]:
```
preprocessed_data.shape
```

Out[66]:
```
(150000, 2)
```

In [67]:
```
train_datasize=120000
x_train=preprocessed_data.head(train_datasize)
x_test=preprocessed_data.tail(preprocessed_data.shape[0] - train_datasi
ze)

y_train = multilabel_yx[0:train_datasize,:]
y_test = multilabel_yx[train_datasize:preprocessed_data.shape[0],:]
```

In [68]:
```
print("Number of data points in y train data :", y_train.shape)
print("Number of data points in y test data :", y_test.shape)
print("Number of data points in x train data :", x_train.shape)
print("Number of data points in x test data :", x_test.shape)
```

```
Number of data points in y train data : (120000, 500)
Number of data points in y test data : (30000, 500)
Number of data points in x train data : (120000, 2)
Number of data points in x test data : (30000, 2)
```

4.5.2 Featurizing data with TfIdf vectorizer

In [69]:
```
start = datetime.now()
vectorizer = TfidfVectorizer(min_df=0.00009, max_features=200000, smoot
h_idf=True, norm="l2", \
                             tokenizer = lambda x: x.split(), sublinear
_tf=False, ngram_range=(1,3))
x_train_multilabel = vectorizer.fit_transform(x_train['question'])
x_test_multilabel = vectorizer.transform(x_test['question'])
print("Time taken to run this cell :", datetime.now() - start)
```

```
Time taken to run this cell : 0:01:49.784085
```

In [70]:
```
print("Dimensions of train data X:",x_train_multilabel.shape, "Y :",y_t
rain.shape)
print("Dimensions of test data X:",x_test_multilabel.shape,"Y:",y_test.
shape)
```

```
Dimensions of train data X: (120000, 101362) Y : (120000, 500)
Dimensions of test data X: (30000, 101362) Y: (30000, 500)
```

4.5.3 Applying Logistic Regression with OneVsRest Classifier

In [71]:

```
start = datetime.now()
classifier = OneVsRestClassifier(SGDClassifier(loss='log', alpha=0.0000
1, penalty='l1'), n_jobs=-1)
classifier.fit(x_train_multilabel, y_train)
predictions = classifier.predict (x_test_multilabel)


print("Accuracy :",metrics.accuracy_score(y_test, predictions))
print("Hamming loss ",metrics.hamming_loss(y_test,predictions))


precision = precision_score(y_test, predictions, average='micro')
recall = recall_score(y_test, predictions, average='micro')
f1 = f1_score(y_test, predictions, average='micro')

global_report = global_report.append({
                        'Vectorizer': 'Tf-IDF',
                        'Model': 'Logistic Regression (SGD with log los
s)',
                        'NGram': '(1,3)',
                        'Parameter': 0.00001,
                        'Precision': precision,
                        'Recall': recall,
                        'F1_Score_Micro':f1
                    },
                    ignore_index=True)

print("Micro-average quality numbers")
print("Precision: {:.4f}, Recall: {:.4f}, F1-measure: {:.4f}".format(pr
ecision, recall, f1))

precision = precision_score(y_test, predictions, average='macro')
recall = recall_score(y_test, predictions, average='macro')
f1 = f1_score(y_test, predictions, average='macro')

print("Macro-average quality numbers")
print("Precision: {:.4f}, Recall: {:.4f}, F1-measure: {:.4f}".format(pr
ecision, recall, f1))



print (metrics.classification_report(y_test, predictions))
print("Time taken to run this cell :", datetime.now() - start)
```

```
Accuracy : 0.2045
Hamming loss  0.0030292
Micro-average quality numbers
Precision: 0.7131, Recall: 0.3144, F1-measure: 0.4364
Macro-average quality numbers
Precision: 0.5388, Recall: 0.2350, F1-measure: 0.3075
              precision    recall  f1-score   support

           0       0.80      0.32      0.45      1111
           1       0.73      0.17      0.28      2052
           2       0.67      0.36      0.47      2388
           3       0.75      0.50      0.60      2226
           4       0.82      0.43      0.57      2014
           5       0.57      0.11      0.18       642
           6       0.83      0.35      0.49      1756
           7       0.93      0.62      0.74      1690
           8       0.65      0.19      0.29       341
           9       0.78      0.77      0.78      2344
          10       0.68      0.35      0.46       821
          11       0.57      0.21      0.30      1143
          12       0.85      0.32      0.46       768
          13       0.64      0.26      0.37       745
          14       0.80      0.55      0.65       952
          15       0.56      0.25      0.34       314
          16       0.59      0.19      0.29       624
```

```
17       0.80      0.53      0.64       535
18       0.88      0.44      0.59       631
19       0.94      0.46      0.61       101
20       0.74      0.20      0.32       245
21       0.83      0.49      0.62       694
22       0.60      0.24      0.34       568
23       0.73      0.26      0.38       423
24       0.75      0.24      0.36       406
25       0.69      0.29      0.41      1373
26       0.60      0.27      0.37       253
27       0.28      0.06      0.09       357
28       0.84      0.23      0.37       222
29       0.69      0.25      0.36       273
30       0.60      0.22      0.33       308
31       0.63      0.25      0.35       256
32       0.76      0.38      0.51       295
33       0.36      0.06      0.10       263
34       0.88      0.45      0.59       256
35       0.51      0.28      0.36       280
36       0.46      0.20      0.28       290
37       0.21      0.05      0.08       200
38       0.38      0.19      0.26       109
39       0.70      0.41      0.52       209
40       0.63      0.40      0.49       113
41       0.73      0.21      0.32       197
42       0.81      0.42      0.56        52
43       0.50      0.01      0.02       179
44       0.83      0.42      0.56       431
45       0.40      0.09      0.14        47
46       0.79      0.30      0.43        37
47       0.68      0.28      0.40       155
48       0.69      0.48      0.56       254
49       0.61      0.25      0.35       201
50       0.69      0.30      0.41        61
51       0.96      0.70      0.81       246
52       0.72      0.60      0.66       146
53       0.94      0.89      0.91       516
54       0.86      0.60      0.71       170
55       0.35      0.07      0.12       234
```

```
56       0.27      0.04      0.07       357
57       0.74      0.22      0.34        78
58       0.92      0.68      0.78       102
59       0.55      0.19      0.28       122
60       0.86      0.54      0.67       138
61       0.64      0.25      0.36        36
62       0.57      0.09      0.16       172
63       0.09      0.02      0.03        60
64       0.70      0.47      0.56       106
65       0.33      0.12      0.17        34
66       0.32      0.12      0.17       101
67       0.57      0.21      0.31        38
68       0.64      0.33      0.43       104
69       0.53      0.13      0.21       144
70       0.51      0.21      0.30       135
71       0.26      0.09      0.14       190
72       0.78      0.32      0.46       139
73       1.00      0.01      0.03        69
74       0.14      0.01      0.01       133
75       0.81      0.35      0.49       181
76       0.53      0.37      0.44       113
77       0.70      0.20      0.31       158
78       0.41      0.11      0.17       142
79       0.54      0.22      0.31        96
80       0.50      0.15      0.23       101
81       0.75      0.21      0.33        56
82       0.14      0.02      0.03        62
83       0.66      0.45      0.54        77
84       0.90      0.09      0.16       100
85       0.49      0.33      0.40        54
86       0.44      0.09      0.15        79
87       0.61      0.22      0.32        92
88       0.72      0.23      0.34       124
89       0.67      0.40      0.50       101
90       0.33      0.03      0.05        40
91       0.58      0.27      0.37        66
92       0.44      0.26      0.33        58
93       0.83      0.12      0.21       161
94       0.61      0.18      0.27       130
```

```
 95    0.55    0.13    0.21     47
 96    0.75    0.51    0.61    107
 97    0.56    0.23    0.33     39
 98    0.20    0.03    0.05    111
 99    0.71    0.11    0.18     95
100    0.34    0.16    0.22    129
101    0.91    0.34    0.50     91
102    0.50    0.19    0.27     27
103    0.93    0.78    0.85     90
104    0.40    0.02    0.03    124
105    0.41    0.16    0.23     76
106    0.25    0.04    0.06    371
107    0.72    0.33    0.46    114
108    0.59    0.31    0.40     98
109    0.83    0.30    0.44     63
110    0.62    0.33    0.43     24
111    0.74    0.32    0.45     53
112    0.43    0.09    0.15     65
113    0.53    0.26    0.35     70
114    0.75    0.44    0.56     27
115    0.33    0.01    0.03     72
116    0.57    0.30    0.39     27
117    0.79    0.17    0.28     90
118    0.68    0.34    0.45     95
119    0.30    0.15    0.20     92
120    0.39    0.21    0.27     87
121    0.58    0.31    0.41     45
122    0.33    0.01    0.01    182
123    0.41    0.12    0.18     94
124    0.77    0.32    0.45     62
125    0.86    0.42    0.56     91
126    0.89    0.35    0.50     69
127    0.63    0.42    0.51     73
128    0.88    0.28    0.42     25
129    0.00    0.00    0.00     68
130    0.33    0.12    0.18    123
131    0.18    0.05    0.08     84
132    0.00    0.00    0.00     67
133    0.64    0.11    0.19    127
```

```
134       0.90      0.40      0.55         45
135       0.49      0.31      0.38         88
136       0.00      0.00      0.00         63
137       0.93      0.72      0.81         96
138       0.62      0.07      0.13         71
139       0.93      0.54      0.68         92
140       1.00      0.04      0.08         23
141       0.62      0.14      0.23         90
142       0.33      0.10      0.15         10
143       0.50      0.16      0.24         44
144       0.61      0.21      0.31         67
145       0.72      0.40      0.52        131
146       0.28      0.10      0.14         83
147       0.00      0.00      0.00         32
148       0.54      0.06      0.11        115
149       0.48      0.21      0.29         63
150       0.76      0.30      0.43         83
151       0.70      0.39      0.50        101
152       0.20      0.03      0.06         29
153       0.94      0.84      0.89        191
154       0.97      0.56      0.71         54
155       0.68      0.30      0.41         84
156       0.54      0.19      0.28         37
157       0.38      0.28      0.32         65
158       0.53      0.17      0.26         46
159       0.87      0.41      0.56         80
160       0.00      0.00      0.00         66
161       0.00      0.00      0.00         56
162       0.50      0.17      0.25        127
163       0.80      0.46      0.58        111
164       0.27      0.09      0.14         32
165       0.44      0.14      0.22         28
166       0.33      0.02      0.04         98
167       0.76      0.53      0.63         88
168       0.87      0.46      0.60         59
169       0.00      0.00      0.00         42
170       0.67      0.50      0.57          4
171       0.75      0.43      0.55         95
172       0.67      0.04      0.07         54
```

```
173       0.84      0.49      0.62        65
174       0.58      0.35      0.44        31
175       0.62      0.25      0.36        32
176       0.64      0.31      0.42        58
177       0.79      0.14      0.24        76
178       0.25      0.02      0.03        55
179       0.89      0.76      0.82        74
180       0.95      0.55      0.69        64
181       1.00      0.09      0.16        57
182       0.31      0.11      0.16        36
183       0.64      0.27      0.38        52
184       0.75      0.38      0.50        48
185       0.67      0.12      0.21        16
186       0.00      0.00      0.00        28
187       0.33      0.08      0.13        36
188       0.67      0.38      0.49        26
189       0.33      0.14      0.19        44
190       0.57      0.17      0.27        46
191       0.29      0.08      0.12        75
192       0.50      0.08      0.14        50
193       0.75      0.45      0.56        20
194       0.50      0.07      0.13        27
195       0.60      0.50      0.55         6
196       0.25      0.04      0.07        68
197       0.59      0.59      0.59        29
198       0.82      0.09      0.16       104
199       0.57      0.33      0.42        36
200       1.00      0.75      0.86         4
201       1.00      0.25      0.40         4
202       0.38      0.03      0.06        96
203       0.94      0.54      0.69        61
204       0.40      0.07      0.12        82
205       0.47      0.26      0.34        34
206       0.79      0.35      0.48        66
207       0.27      0.07      0.11        97
208       0.22      0.04      0.07        89
209       0.76      0.53      0.62        55
210       0.82      0.36      0.50        78
211       0.75      0.04      0.07        78
```

```
212        0.68       0.17       0.27        158
213        0.29       0.05       0.08         44
214        0.65       0.49       0.56         35
215        0.90       0.58       0.71         48
216        0.69       0.58       0.63         62
217        1.00       0.18       0.31         11
218        0.96       0.38       0.55         68
219        0.35       0.12       0.17         60
220        0.50       0.08       0.14         25
221        0.43       0.18       0.25         57
222        0.94       0.42       0.58         36
223        0.46       0.07       0.12         88
224        0.44       0.09       0.15         46
225        0.57       0.07       0.12         60
226        0.52       0.17       0.26         65
227        1.00       0.43       0.60          7
228        0.14       0.08       0.11         12
229        0.27       0.04       0.08         68
230        0.67       0.15       0.24         40
231        0.22       0.08       0.11         26
232        0.81       0.57       0.67         30
233        0.80       0.10       0.17         41
234        0.27       0.06       0.09         53
235        0.65       0.49       0.56         35
236        0.40       0.11       0.17         18
237        0.00       0.00       0.00         22
238        0.73       0.54       0.62         59
239        0.78       0.42       0.55         43
240        0.43       0.13       0.20         45
241        0.60       0.07       0.12         46
242        0.33       0.05       0.09         38
243        0.86       0.34       0.49         56
244        0.27       0.09       0.13         35
245        0.50       0.02       0.05         42
246        0.25       0.06       0.10         33
247        0.39       0.15       0.22         47
248        0.64       0.28       0.39         25
249        0.77       0.51       0.62         39
250        1.00       0.06       0.12         77
```

```
251      0.79     0.41     0.54      56
252      0.11     0.03     0.04      72
253      1.00     1.00     1.00       4
254      0.71     0.41     0.52      29
255      0.35     0.06     0.11     113
256      0.82     0.61     0.70      59
257      0.00     0.00     0.00      59
258      0.93     0.64     0.76      39
259      0.82     0.75     0.78      12
260      0.75     0.67     0.71       9
261      0.96     0.55     0.70      44
262      0.78     0.56     0.65      32
263      0.25     0.01     0.01     156
264      1.00     0.40     0.57       5
265      0.09     0.02     0.03     198
266      0.00     0.00     0.00      40
267      0.33     0.03     0.06      29
268      0.00     0.00     0.00      39
269      0.00     0.00     0.00       6
270      0.75     0.60     0.67       5
271      0.00     0.00     0.00      17
272      0.25     0.04     0.06      54
273      0.38     0.13     0.19      23
274      0.00     0.00     0.00     126
275      0.77     0.31     0.44      32
276      0.67     0.60     0.63      10
277      0.83     0.52     0.64      67
278      0.48     0.19     0.27      53
279      0.67     0.12     0.21      16
280      0.77     0.53     0.62      19
281      0.67     0.07     0.12      61
282      0.50     0.02     0.05      81
283      0.66     0.20     0.31      94
284      0.70     0.23     0.34      31
285      0.40     0.05     0.08      43
286      0.60     0.11     0.19      79
287      0.30     0.40     0.34      20
288      0.78     0.82     0.80      17
289      0.82     0.48     0.61      56
```

```
290       0.00      0.00      0.00        63
291       0.44      0.17      0.25        46
292       0.65      0.30      0.41        50
293       0.50      0.12      0.19        17
294       0.83      0.19      0.30        27
295       0.50      0.03      0.06        63
296       0.40      0.16      0.23        25
297       0.00      0.00      0.00        38
298       0.00      0.00      0.00        62
299       0.73      0.45      0.56        49
300       0.25      0.03      0.05        39
301       0.50      0.12      0.20         8
302       0.00      0.00      0.00        18
303       0.43      0.16      0.23        19
304       0.78      0.39      0.52        18
305       0.44      0.31      0.36        26
306       0.67      0.20      0.31        10
307       0.17      0.08      0.11        12
308       1.00      0.38      0.55         8
309       0.50      0.05      0.10        57
310       0.43      0.08      0.14        37
311       0.91      0.58      0.71        50
312       0.62      0.14      0.23        36
313       0.45      0.26      0.33        19
314       0.92      0.59      0.72        59
315       0.87      0.56      0.68        48
316       0.33      0.09      0.14        45
317       0.33      0.07      0.11        29
318       0.50      0.38      0.43        37
319       0.50      0.08      0.14        38
320       0.82      0.52      0.64        44
321       0.72      0.32      0.44        41
322       0.00      0.00      0.00        25
323       0.00      0.00      0.00         4
324       1.00      0.09      0.17        11
325       1.00      1.00      1.00         3
326       0.62      0.16      0.25        32
327       0.60      0.24      0.34        50
328       0.14      0.02      0.04        46
```

```
329         0.96        0.55        0.70         47
330         1.00        0.58        0.73         31
331         0.00        0.00        0.00         11
332         0.00        0.00        0.00         31
333         1.00        0.08        0.14         26
334         0.00        0.00        0.00         41
335         0.00        0.00        0.00         39
336         0.79        0.38        0.51         40
337         0.27        0.08        0.12         37
338         1.00        0.30        0.46         10
339         0.00        0.00        0.00         38
340         1.00        0.78        0.88         23
341         0.17        0.06        0.09         33
342         0.31        0.10        0.15         40
343         0.60        0.17        0.27         35
344         0.56        0.33        0.42         30
345         0.00        0.00        0.00         33
346         0.86        0.24        0.38         25
347         0.50        0.22        0.31          9
348         0.50        0.50        0.50          2
349         0.81        0.38        0.52         34
350         0.33        0.13        0.19         38
351         0.50        0.16        0.25         49
352         0.00        0.00        0.00         22
353         0.47        0.21        0.29         39
354         0.25        0.06        0.09         18
355         0.43        0.10        0.16         31
356         0.50        0.06        0.11         17
357         0.92        0.34        0.50         35
358         0.25        0.02        0.04         43
359         0.68        0.45        0.54         47
360         0.50        0.03        0.06         29
361         0.74        0.61        0.67         38
362         0.47        0.22        0.30         36
363         0.36        0.22        0.27         55
364         0.73        0.24        0.36         34
365         0.46        0.25        0.32         24
366         0.00        0.00        0.00         19
367         0.30        0.60        0.40          5
```

```
368    0.50    0.09    0.15    33
369    0.78    0.53    0.63    34
370    0.43    0.15    0.22    20
371    0.00    0.00    0.00    17
372    0.47    0.23    0.30    31
373    0.00    0.00    0.00    34
374    0.25    0.04    0.07    23
375    0.50    0.26    0.34    31
376    0.62    0.18    0.28    28
377    0.83    0.68    0.75    22
378    1.00    0.33    0.50     9
379    1.00    0.07    0.13    29
380    0.47    0.27    0.34    30
381    0.85    0.63    0.72    35
382    0.69    0.25    0.37    36
383    1.00    0.25    0.40     4
384    0.94    0.67    0.78    24
385    0.58    0.28    0.38    25
386    0.00    0.00    0.00    27
387    0.55    0.17    0.26    36
388    0.75    0.39    0.51    31
389    0.87    0.35    0.50    37
390    0.46    0.22    0.30    27
391    0.59    0.35    0.44    46
392    0.33    0.25    0.29     4
393    0.17    0.11    0.13    19
394    0.00    0.00    0.00    12
395    0.44    0.27    0.33    26
396    0.24    0.06    0.09    69
397    1.00    0.16    0.28    25
398    0.25    0.03    0.06    32
399    0.40    0.12    0.19    33
400    0.00    0.00    0.00    38
401    0.50    0.18    0.26    17
402    0.00    0.00    0.00    24
403    1.00    0.31    0.48    16
404    1.00    0.27    0.42    15
405    0.50    0.05    0.09    20
406    0.33    0.07    0.11    15
```

```
407        0.60       0.12       0.20         25
408        0.33       0.26       0.29         19
409        0.60       0.26       0.36         46
410        0.00       0.00       0.00         45
411        0.67       0.10       0.17         21
412        0.00       0.00       0.00          8
413        0.58       0.40       0.47         35
414        0.00       0.00       0.00         34
415        1.00       0.50       0.67         14
416        0.10       0.03       0.05         29
417        0.44       0.14       0.22         28
418        0.40       0.10       0.15         21
419        1.00       0.04       0.07         26
420        0.58       0.29       0.39         38
421        0.00       0.00       0.00        131
422        0.50       0.12       0.19         26
423        0.58       0.28       0.38         25
424        0.78       0.38       0.51         48
425        0.00       0.00       0.00         24
426        0.33       0.05       0.08         42
427        0.40       0.31       0.35         26
428        0.00       0.00       0.00         10
429        0.56       0.41       0.47         54
430        0.70       0.22       0.33         32
431        0.40       0.17       0.24         48
432        0.27       0.11       0.16         35
433        1.00       0.05       0.09         22
434        0.50       0.12       0.20         24
435        1.00       0.46       0.63         59
436        0.00       0.00       0.00         35
437        0.00       0.00       0.00         12
438        0.61       0.28       0.38         50
439        0.25       0.06       0.09         36
440        0.00       0.00       0.00         35
441        0.00       0.00       0.00          8
442        0.00       0.00       0.00         48
443        1.00       0.44       0.62         18
444        0.77       0.19       0.31         52
445        0.67       0.20       0.31         20
```

```
446       0.00      0.00      0.00        18
447       0.00      0.00      0.00         6
448       0.33      0.04      0.06        28
449       0.29      0.05      0.09        38
450       0.00      0.00      0.00       129
451       0.94      0.55      0.70        29
452       0.17      0.02      0.03        58
453       0.20      0.03      0.05        32
454       0.75      0.38      0.50        16
455       0.95      0.53      0.68        34
456       0.29      0.44      0.35         9
457       0.96      0.73      0.83        30
458       0.60      0.15      0.24        40
459       0.57      0.11      0.18        37
460       0.90      0.60      0.72        30
461       1.00      0.22      0.36        27
462       0.29      0.17      0.21        12
463       0.50      0.35      0.41        17
464       0.88      0.25      0.39        56
465       0.80      0.44      0.57         9
466       0.00      0.00      0.00        22
467       0.17      0.11      0.13         9
468       0.00      0.00      0.00        15
469       0.00      0.00      0.00        14
470       0.67      0.25      0.36        16
471       0.86      0.33      0.48        18
472       0.25      0.12      0.17        16
473       0.73      0.50      0.59        22
474       0.12      0.08      0.10        12
475       1.00      0.47      0.64       110
476       0.50      0.45      0.47        20
477       0.70      0.45      0.55        31
478       0.71      0.12      0.20        42
479       0.00      0.00      0.00         4
480       0.45      0.11      0.17        47
481       1.00      0.33      0.50        30
482       0.00      0.00      0.00        35
483       0.11      0.03      0.05        30
484       0.33      0.10      0.15        20
```

```
         485       0.71      0.33      0.45        15
         486       1.00      0.12      0.21        17
         487       1.00      0.09      0.17        11
         488       0.50      0.14      0.22        36
         489       0.12      0.03      0.05        32
         490       0.80      0.29      0.42        14
         491       0.73      0.32      0.44        25
         492       0.42      0.57      0.48        23
         493       0.00      0.00      0.00         9
         494       0.50      0.11      0.18        37
         495       0.00      0.00      0.00        24
         496       0.82      0.26      0.40        34
         497       0.17      0.04      0.06        25
         498       0.00      0.00      0.00         9
         499       0.38      0.18      0.24        17

   micro avg       0.71      0.31      0.44     55953
   macro avg       0.54      0.24      0.31     55953
weighted avg       0.65      0.31      0.40     55953
 samples avg       0.41      0.30      0.32     55953

Time taken to run this cell : 0:01:56.473712
```

In [72]:

```
start = datetime.now()
classifier_2 = OneVsRestClassifier(LogisticRegression(penalty='l1'), n_
jobs=-1)
classifier_2.fit(x_train_multilabel, y_train)
predictions_2 = classifier_2.predict(x_test_multilabel)
print("Accuracy :",metrics.accuracy_score(y_test, predictions_2))
print("Hamming loss ",metrics.hamming_loss(y_test,predictions_2))


precision = precision_score(y_test, predictions_2, average='micro')
recall = recall_score(y_test, predictions_2, average='micro')
f1 = f1_score(y_test, predictions_2, average='micro')

global_report = global_report.append({
                        'Vectorizer': 'Tf-IDF',
                        'Model': 'Logistic Regression',
                        'NGram': '(1,3)',
                        'Parameter': 1,
                        'Precision': precision,
                        'Recall': recall,
                        'F1_Score_Micro':f1
                     },
                    ignore_index=True)

print("Micro-average quality numbers")
print("Precision: {:.4f}, Recall: {:.4f}, F1-measure: {:.4f}".format(pr
ecision, recall, f1))

precision = precision_score(y_test, predictions_2, average='macro')
recall = recall_score(y_test, predictions_2, average='macro')
f1 = f1_score(y_test, predictions_2, average='macro')

print("Macro-average quality numbers")
print("Precision: {:.4f}, Recall: {:.4f}, F1-measure: {:.4f}".format(pr
ecision, recall, f1))

print (metrics.classification_report(y_test, predictions_2))
print("Time taken to run this cell :", datetime.now() - start)
```

```
Accuracy : 0.21033333333333334
Hamming loss  0.003007
Micro-average quality numbers
Precision: 0.7086, Recall: 0.3293, F1-measure: 0.4496
Macro-average quality numbers
Precision: 0.5460, Recall: 0.2565, F1-measure: 0.3290
              precision    recall  f1-score   support

           0       0.80      0.33      0.47      1111
           1       0.71      0.19      0.30      2052
           2       0.68      0.36      0.47      2388
           3       0.75      0.50      0.60      2226
           4       0.82      0.44      0.57      2014
           5       0.57      0.12      0.19       642
           6       0.83      0.35      0.49      1756
           7       0.93      0.65      0.77      1690
```

```
 8       0.64      0.19      0.30       341

 9       0.78      0.78      0.78      2344
10       0.68      0.36      0.47       821
11       0.58      0.22      0.31      1143
12       0.84      0.33      0.47       768
13       0.62      0.27      0.38       745
14       0.79      0.55      0.65       952
15       0.56      0.25      0.34       314
16       0.58      0.18      0.28       624
17       0.80      0.53      0.64       535
18       0.87      0.47      0.61       631
19       0.94      0.50      0.66       101
20       0.76      0.20      0.32       245
21       0.83      0.49      0.62       694
22       0.59      0.25      0.35       568
23       0.71      0.26      0.39       423
24       0.73      0.25      0.37       406
25       0.70      0.29      0.41      1373
26       0.58      0.25      0.35       253
27       0.29      0.08      0.12       357
28       0.83      0.24      0.38       222
29       0.65      0.25      0.36       273
30       0.62      0.25      0.36       308
31       0.62      0.25      0.35       256
32       0.75      0.40      0.52       295
33       0.39      0.07      0.12       263
34       0.88      0.47      0.61       256
35       0.52      0.29      0.37       280
36       0.47      0.22      0.30       290
37       0.18      0.04      0.07       200
38       0.42      0.21      0.28       109
39       0.69      0.43      0.53       209
40       0.67      0.38      0.49       113
41       0.73      0.22      0.34       197
42       0.79      0.44      0.57        52
43       0.43      0.02      0.03       179
44       0.82      0.41      0.55       431
45       0.45      0.11      0.17        47
46       0.78      0.38      0.51        37
```

```
47       0.68      0.29      0.41       155

48       0.70      0.48      0.57       254
49       0.61      0.27      0.37       201
50       0.75      0.34      0.47        61
51       0.96      0.71      0.82       246
52       0.71      0.60      0.65       146
53       0.94      0.89      0.91       516
54       0.84      0.58      0.69       170
55       0.32      0.06      0.10       234
56       0.27      0.04      0.08       357
57       0.70      0.21      0.32        78
58       0.92      0.70      0.79       102
59       0.54      0.18      0.27       122
60       0.86      0.54      0.66       138
61       0.60      0.25      0.35        36
62       0.49      0.10      0.16       172
63       0.11      0.02      0.03        60
64       0.69      0.51      0.59       106
65       0.38      0.15      0.21        34
66       0.31      0.13      0.18       101
67       0.62      0.21      0.31        38
68       0.65      0.34      0.44       104
69       0.45      0.13      0.20       144
70       0.53      0.23      0.32       135
71       0.32      0.12      0.18       190
72       0.76      0.32      0.45       139
73       0.50      0.01      0.03        69
74       0.14      0.01      0.01       133
75       0.78      0.38      0.51       181
76       0.52      0.35      0.41       113
77       0.67      0.20      0.30       158
78       0.39      0.13      0.20       142
79       0.56      0.20      0.29        96
80       0.59      0.17      0.26       101
81       0.67      0.21      0.32        56
82       0.17      0.02      0.03        62
83       0.67      0.44      0.53        77
84       0.92      0.12      0.21       100
85       0.51      0.35      0.42        54
```

```
 86     0.47     0.09     0.15      79

 87     0.70     0.21     0.32      92
 88     0.67     0.21     0.32     124
 89     0.71     0.40     0.51     101
 90     0.33     0.05     0.09      40
 91     0.61     0.30     0.40      66
 92     0.42     0.24     0.31      58
 93     0.95     0.24     0.38     161
 94     0.66     0.18     0.28     130
 95     0.62     0.17     0.27      47
 96     0.77     0.50     0.61     107
 97     0.62     0.26     0.36      39
 98     0.33     0.05     0.08     111
 99     0.67     0.15     0.24      95
100     0.33     0.12     0.18     129
101     0.84     0.40     0.54      91
102     0.44     0.15     0.22      27
103     0.92     0.79     0.85      90
104     0.30     0.02     0.04     124
105     0.39     0.16     0.22      76
106     0.29     0.09     0.14     371
107     0.70     0.35     0.47     114
108     0.58     0.33     0.42      98
109     0.87     0.32     0.47      63
110     0.69     0.38     0.49      24
111     0.75     0.34     0.47      53
112     0.38     0.08     0.13      65
113     0.58     0.30     0.40      70
114     0.81     0.48     0.60      27
115     0.25     0.01     0.03      72
116     0.60     0.33     0.43      27
117     0.80     0.22     0.35      90
118     0.67     0.35     0.46      95
119     0.29     0.14     0.19      92
120     0.43     0.28     0.34      87
121     0.57     0.29     0.38      45
122     0.25     0.01     0.02     182
123     0.43     0.14     0.21      94
124     0.80     0.32     0.46      62
```

```
125        0.83       0.43       0.57        91

126        0.81       0.38       0.51        69
127        0.64       0.44       0.52        73
128        0.89       0.32       0.47        25
129        1.00       0.01       0.03        68
130        0.36       0.13       0.19       123
131        0.24       0.06       0.10        84
132        0.00       0.00       0.00        67
133        0.52       0.12       0.19       127
134        0.80       0.44       0.57        45
135        0.52       0.34       0.41        88
136        0.00       0.00       0.00        63
137        0.92       0.72       0.81        96
138        0.62       0.07       0.13        71
139        0.93       0.58       0.71        92
140        1.00       0.04       0.08        23
141        0.64       0.16       0.25        90
142        0.50       0.10       0.17        10
143        0.47       0.16       0.24        44
144        0.64       0.24       0.35        67
145        0.76       0.42       0.54       131
146        0.26       0.08       0.13        83
147        0.17       0.03       0.05        32
148        0.48       0.11       0.18       115
149        0.63       0.27       0.38        63
150        0.77       0.33       0.46        83
151        0.69       0.35       0.46       101
152        0.17       0.03       0.06        29
153        0.93       0.86       0.90       191
154        0.94       0.59       0.73        54
155        0.63       0.26       0.37        84
156        0.63       0.32       0.43        37
157        0.33       0.25       0.28        65
158        0.50       0.17       0.26        46
159        0.83       0.44       0.57        80
160        0.11       0.02       0.03        66
161        0.00       0.00       0.00        56
162        0.50       0.19       0.27       127
163        0.73       0.57       0.64       111
```

```
164         0.22       0.06       0.10         32

165         0.50       0.11       0.18         28
166         0.29       0.02       0.04         98
167         0.76       0.55       0.64         88
168         0.88       0.47       0.62         59
169         0.00       0.00       0.00         42
170         0.50       0.50       0.50          4
171         0.75       0.42       0.54         95
172         0.50       0.06       0.10         54
173         0.85       0.52       0.65         65
174         0.60       0.39       0.47         31
175         0.64       0.28       0.39         32
176         0.65       0.34       0.45         58
177         0.65       0.14       0.24         76
178         0.50       0.04       0.07         55
179         0.88       0.78       0.83         74
180         0.90       0.56       0.69         64
181         1.00       0.09       0.16         57
182         0.36       0.11       0.17         36
183         0.56       0.27       0.36         52
184         0.70       0.40       0.51         48
185         0.57       0.25       0.35         16
186         0.00       0.00       0.00         28
187         0.31       0.11       0.16         36
188         0.56       0.38       0.45         26
189         0.38       0.14       0.20         44
190         0.47       0.15       0.23         46
191         0.29       0.09       0.14         75
192         0.58       0.14       0.23         50
193         0.82       0.45       0.58         20
194         0.50       0.07       0.13         27
195         0.50       0.50       0.50          6
196         0.29       0.07       0.12         68
197         0.54       0.52       0.53         29
198         0.79       0.11       0.19        104
199         0.57       0.33       0.42         36
200         1.00       0.75       0.86          4
201         1.00       0.25       0.40          4
202         0.43       0.03       0.06         96
203         0.83       0.50       0.73         61
```

```
203        0.92       0.59       0.72         61

204        0.36       0.06       0.10         82
205        0.48       0.29       0.36         34
206        0.81       0.39       0.53         66
207        0.21       0.06       0.10         97
208        0.26       0.07       0.11         89
209        0.76       0.53       0.62         55
210        0.81       0.44       0.57         78
211        0.50       0.05       0.09         78
212        0.74       0.34       0.46        158
213        0.38       0.07       0.12         44
214        0.69       0.51       0.59         35
215        0.91       0.67       0.77         48
216        0.74       0.65       0.69         62
217        1.00       0.09       0.17         11
218        1.00       0.43       0.60         68
219        0.33       0.10       0.15         60
220        0.40       0.08       0.13         25
221        0.44       0.19       0.27         57
222        0.83       0.42       0.56         36
223        0.67       0.09       0.16         88
224        0.33       0.07       0.11         46
225        0.56       0.08       0.14         60
226        0.52       0.17       0.26         65
227        0.80       0.57       0.67          7
228        0.25       0.17       0.20         12
229        0.36       0.06       0.10         68
230        0.88       0.17       0.29         40
231        0.22       0.08       0.11         26
232        0.81       0.57       0.67         30
233        0.86       0.15       0.25         41
234        0.27       0.08       0.12         53
235        0.61       0.49       0.54         35
236        0.50       0.11       0.18         18
237        0.14       0.05       0.07         22
238        0.70       0.54       0.61         59
239        0.78       0.42       0.55         43
240        0.41       0.16       0.23         45
241        0.67       0.09       0.15         46
```

```
242      0.33      0.05      0.09       38

243      0.87      0.36      0.51       56
244      0.27      0.11      0.16       35
245      0.25      0.02      0.04       42
246      0.33      0.06      0.10       33
247      0.42      0.17      0.24       47
248      0.70      0.28      0.40       25
249      0.76      0.49      0.59       39
250      1.00      0.10      0.19       77
251      0.79      0.48      0.60       56
252      0.26      0.07      0.11       72
253      1.00      1.00      1.00        4
254      0.67      0.41      0.51       29
255      0.53      0.16      0.24      113
256      0.80      0.66      0.72       59
257      0.00      0.00      0.00       59
258      0.93      0.67      0.78       39
259      0.82      0.75      0.78       12
260      0.86      0.67      0.75        9
261      0.97      0.64      0.77       44
262      0.78      0.56      0.65       32
263      0.11      0.01      0.01      156
264      1.00      0.40      0.57        5
265      0.12      0.06      0.08      198
266      0.00      0.00      0.00       40
267      0.33      0.03      0.06       29
268      0.00      0.00      0.00       39
269      0.00      0.00      0.00        6
270      0.75      0.60      0.67        5
271      0.00      0.00      0.00       17
272      0.30      0.06      0.09       54
273      0.40      0.17      0.24       23
274      0.00      0.00      0.00      126
275      0.71      0.31      0.43       32
276      0.60      0.60      0.60       10
277      0.85      0.61      0.71       67
278      0.50      0.21      0.29       53
279      0.67      0.12      0.21       16
280      0.80      0.63      0.71       19
```

```
281       0.67      0.07      0.12        61

282       0.54      0.09      0.15        81
283       0.56      0.20      0.30        94
284       0.70      0.23      0.34        31
285       0.40      0.05      0.08        43
286       0.58      0.18      0.27        79
287       0.29      0.35      0.32        20
288       0.78      0.82      0.80        17
289       0.82      0.59      0.69        56
290       0.10      0.02      0.03        63
291       0.44      0.17      0.25        46
292       0.62      0.30      0.41        50
293       0.60      0.18      0.27        17
294       0.83      0.19      0.30        27
295       0.50      0.03      0.06        63
296       0.33      0.12      0.18        25
297       0.00      0.00      0.00        38
298       0.14      0.02      0.03        62
299       0.77      0.49      0.60        49
300       0.38      0.08      0.13        39
301       0.67      0.25      0.36         8
302       0.00      0.00      0.00        18
303       0.33      0.21      0.26        19
304       0.67      0.44      0.53        18
305       0.56      0.35      0.43        26
306       0.67      0.20      0.31        10
307       0.14      0.08      0.11        12
308       0.75      0.38      0.50         8
309       0.44      0.07      0.12        57
310       0.57      0.11      0.18        37
311       0.85      0.66      0.74        50
312       0.50      0.17      0.25        36
313       0.42      0.26      0.32        19
314       0.91      0.69      0.79        59
315       0.84      0.65      0.73        48
316       0.36      0.11      0.17        45
317       0.56      0.17      0.26        29
318       0.50      0.38      0.43        37
319       0.50      0.08      0.14        38
320       0.81      0.57      0.67        44
```

```
320        0.81       0.57       0.67         44

321        0.79       0.37       0.50         41
322        0.00       0.00       0.00         25
323        0.00       0.00       0.00          4
324        0.33       0.09       0.14         11
325        0.75       1.00       0.86          3
326        0.62       0.16       0.25         32
327        0.55       0.22       0.31         50
328        0.14       0.02       0.04         46
329        0.97       0.64       0.77         47
330        0.95       0.61       0.75         31
331        0.00       0.00       0.00         11
332        0.00       0.00       0.00         31
333        0.80       0.15       0.26         26
334        0.00       0.00       0.00         41
335        0.00       0.00       0.00         39
336        0.76       0.40       0.52         40
337        0.23       0.08       0.12         37
338        1.00       0.50       0.67         10
339        0.00       0.00       0.00         38
340        1.00       0.74       0.85         23
341        0.22       0.06       0.10         33
342        0.31       0.10       0.15         40
343        0.67       0.17       0.27         35
344        0.48       0.33       0.39         30
345        0.00       0.00       0.00         33
346        0.83       0.40       0.54         25
347        0.50       0.22       0.31          9
348        0.50       0.50       0.50          2
349        0.86       0.56       0.68         34
350        0.42       0.21       0.28         38
351        0.47       0.14       0.22         49
352        0.00       0.00       0.00         22
353        0.48       0.26       0.33         39
354        0.50       0.22       0.31         18
355        0.38       0.10       0.15         31
356        0.00       0.00       0.00         17
357        0.93       0.40       0.56         35
358        0.25       0.02       0.04         43
359        0.56       0.63       0.58         47
```

```
359     0.56    0.62    0.59    47

360     0.67    0.07    0.12    29
361     0.77    0.61    0.68    38
362     0.45    0.25    0.32    36
363     0.31    0.27    0.29    55
364     0.64    0.26    0.37    34
365     0.50    0.29    0.37    24
366     0.00    0.00    0.00    19
367     0.38    0.60    0.46     5
368     0.57    0.12    0.20    33
369     0.79    0.56    0.66    34
370     0.75    0.15    0.25    20
371     0.00    0.00    0.00    17
372     0.46    0.19    0.27    31
373     0.25    0.03    0.05    34
374     0.20    0.04    0.07    23
375     0.50    0.29    0.37    31
376     0.50    0.18    0.26    28
377     0.88    0.64    0.74    22
378     1.00    0.33    0.50     9
379     1.00    0.10    0.19    29
380     0.33    0.23    0.27    30
381     0.85    0.63    0.72    35
382     0.56    0.28    0.37    36
383     0.00    0.00    0.00     4
384     0.94    0.67    0.78    24
385     0.62    0.32    0.42    25
386     0.25    0.04    0.06    27
387     0.58    0.19    0.29    36
388     0.72    0.42    0.53    31
389     0.88    0.38    0.53    37
390     0.50    0.30    0.37    27
391     0.68    0.41    0.51    46
392     0.33    0.25    0.29     4
393     0.14    0.11    0.12    19
394     0.00    0.00    0.00    12
395     0.50    0.31    0.38    26
396     0.29    0.07    0.12    69
397     0.87    0.52    0.65    25
398     0.33    0.06    0.11    33
```

```
398         0.33        0.06        0.11          32

399         0.42        0.15        0.22          33
400         0.00        0.00        0.00          38
401         0.50        0.18        0.26          17
402         0.00        0.00        0.00          24
403         0.80        0.50        0.62          16
404         1.00        0.33        0.50          15
405         0.25        0.05        0.08          20
406         0.25        0.07        0.11          15
407         0.43        0.12        0.19          25
408         0.38        0.26        0.31          19
409         0.58        0.24        0.34          46
410         0.00        0.00        0.00          45
411         0.40        0.10        0.15          21
412         0.00        0.00        0.00           8
413         0.64        0.40        0.49          35
414         0.00        0.00        0.00          34
415         1.00        0.50        0.67          14
416         0.18        0.07        0.10          29
417         0.33        0.11        0.16          28
418         0.25        0.05        0.08          21
419         1.00        0.08        0.14          26
420         0.58        0.29        0.39          38
421         0.00        0.00        0.00         131
422         0.25        0.08        0.12          26
423         0.54        0.28        0.37          25
424         0.84        0.33        0.48          48
425         0.33        0.04        0.07          24
426         0.45        0.12        0.19          42
427         0.44        0.31        0.36          26
428         1.00        0.10        0.18          10
429         0.58        0.41        0.48          54
430         0.69        0.28        0.40          32
431         0.43        0.19        0.26          48
432         0.28        0.14        0.19          35
433         1.00        0.05        0.09          22
434         0.57        0.17        0.26          24
435         0.94        0.54        0.69          59
436         0.11        0.03        0.05          35
437         0.00        0.00        0.00          13
```

```
437         0.00        0.00        0.00          12

438         0.59        0.26        0.36          50
439         0.22        0.06        0.09          36
440         0.00        0.00        0.00          35
441         0.00        0.00        0.00           8
442         0.00        0.00        0.00          48
443         0.90        0.50        0.64          18
444         0.80        0.23        0.36          52
445         0.80        0.20        0.32          20
446         0.00        0.00        0.00          18
447         0.00        0.00        0.00           6
448         0.40        0.07        0.12          28
449         0.43        0.08        0.13          38
450         0.97        0.53        0.68         129
451         0.84        0.72        0.78          29
452         0.27        0.05        0.09          58
453         0.20        0.03        0.05          32
454         0.67        0.38        0.48          16
455         0.95        0.59        0.73          34
456         0.33        0.44        0.38           9
457         0.96        0.77        0.85          30
458         0.67        0.20        0.31          40
459         0.43        0.08        0.14          37
460         0.83        0.67        0.74          30
461         1.00        0.26        0.41          27
462         0.33        0.17        0.22          12
463         0.42        0.29        0.34          17
464         0.90        0.34        0.49          56
465         1.00        0.67        0.80           9
466         1.00        0.05        0.09          22
467         0.17        0.11        0.13           9
468         0.00        0.00        0.00          15
469         0.50        0.07        0.12          14
470         0.67        0.25        0.36          16
471         0.91        0.56        0.69          18
472         0.33        0.25        0.29          16
473         0.69        0.50        0.58          22
474         0.12        0.08        0.10          12
475         1.00        0.75        0.86         110
476         0.58        0.55        0.56          20
```

```
         476       0.58      0.55      0.56        20

         477       0.68      0.48      0.57        31
         478       0.67      0.14      0.24        42
         479       0.00      0.00      0.00         4
         480       0.50      0.17      0.25        47
         481       1.00      0.37      0.54        30
         482       0.00      0.00      0.00        35
         483       0.22      0.07      0.10        30
         484       0.33      0.15      0.21        20
         485       0.80      0.53      0.64        15
         486       1.00      0.06      0.11        17
         487       0.50      0.09      0.15        11
         488       0.50      0.14      0.22        36
         489       0.12      0.03      0.05        32
         490       0.80      0.29      0.42        14
         491       0.73      0.32      0.44        25
         492       0.37      0.57      0.45        23
         493       0.00      0.00      0.00         9
         494       0.46      0.16      0.24        37
         495       0.00      0.00      0.00        24
         496       0.82      0.26      0.40        34
         497       0.12      0.04      0.06        25
         498       1.00      0.11      0.20         9
         499       0.67      0.24      0.35        17

   micro avg       0.71      0.33      0.45     55953
   macro avg       0.55      0.26      0.33     55953
weighted avg       0.65      0.33      0.42     55953
 samples avg       0.42      0.31      0.34     55953

Time taken to run this cell : 0:05:16.940751
```

4.5.4 Featurizing data with BoW vectorizer

In [73]:

```
start = datetime.now()
vectorizer_bow = CountVectorizer(min_df=0.00009, max_features=200000, \
                             tokenizer = lambda x: x.split(), ngram_ran
ge=(1,4))
x_train_multilabel_bow = vectorizer_bow.fit_transform(x_train['questio
n'])
x_test_multilabel_bow = vectorizer_bow.transform(x_test['question'])
print("Time taken to run this cell :", datetime.now() - start)
```

```
Time taken to run this cell : 0:03:25.682941
```

In [74]:

```
print("Dimensions of train data X:",x_train_multilabel_bow.shape, "Y :"
,y_train.shape)
print("Dimensions of test data X:",x_test_multilabel_bow.shape,"Y:",y_t
est.shape)
```

```
Dimensions of train data X: (120000, 102634) Y : (120000, 500)
Dimensions of test data X: (30000, 102634) Y: (30000, 500)
```

4.5.5 Applying Logistic Regression with OneVsRest Classifier

In [75]:

```
start = datetime.now()
classifier = OneVsRestClassifier(SGDClassifier(loss='log', alpha=0.0000
1, penalty='l1'))
classifier.fit(x_train_multilabel_bow, y_train)
predictions = classifier.predict (x_test_multilabel_bow)


print("Accuracy :",metrics.accuracy_score(y_test, predictions))
print("Hamming loss ",metrics.hamming_loss(y_test,predictions))


precision = precision_score(y_test, predictions, average='micro')
recall = recall_score(y_test, predictions, average='micro')
f1 = f1_score(y_test, predictions, average='micro')

global_report = global_report.append({
                        'Vectorizer': 'BoW',
                        'Model': 'Logistic Regression (SGD with log los
s)',
                        'NGram': '(1,4)',
                                 'Parameter': 0.00001,
                                 'Precision': precision,
                                 'Recall': recall,
                                 'F1_Score_Micro':f1
                             },
                            ignore_index=True)

print("Micro-average quality numbers")
print("Precision: {:.4f}, Recall: {:.4f}, F1-measure: {:.4f}".format(pr
ecision, recall, f1))

precision = precision_score(y_test, predictions, average='macro')
recall = recall_score(y_test, predictions, average='macro')
f1 = f1_score(y_test, predictions, average='macro')

print("Macro-average quality numbers")
print("Precision: {:.4f}, Recall: {:.4f}, F1-measure: {:.4f}".format(pr
ecision, recall, f1))

print (metrics.classification_report(y_test, predictions))
print("Time taken to run this cell :", datetime.now() - start)
```

```
Accuracy : 0.0765
Hamming loss  0.007617333333333334
Micro-average quality numbers
Precision: 0.2297, Recall: 0.4427, F1-measure: 0.3024
Macro-average quality numbers
Precision: 0.1552, Recall: 0.3670, F1-measure: 0.2071
              precision    recall  f1-score   support

           0       0.35      0.45      0.39      1111
           1       0.32      0.37      0.34      2052
           2       0.43      0.48      0.45      2388
           3       0.52      0.60      0.56      2226
           4       0.46      0.55      0.50      2014
           5       0.12      0.22      0.16       642
           6       0.41      0.49      0.44      1756
           7       0.68      0.73      0.71      1690
           8       0.22      0.31      0.26       341
           9       0.66      0.74      0.70      2344
```

```
10        0.38      0.45      0.41       821
11        0.27      0.33      0.30      1143
12        0.39      0.47      0.43       768
13        0.29      0.42      0.35       745
14        0.48      0.62      0.54       952
15        0.21      0.37      0.27       314
16        0.26      0.39      0.31       624
17        0.38      0.58      0.46       535
18        0.44      0.60      0.51       631
19        0.35      0.66      0.46       101
20        0.21      0.35      0.26       245
21        0.47      0.59      0.52       694
22        0.29      0.43      0.35       568
23        0.26      0.39      0.31       423
24        0.26      0.43      0.33       406
25        0.55      0.48      0.51      1373
26        0.18      0.42      0.26       253
27        0.14      0.26      0.18       357
28        0.16      0.36      0.22       222
29        0.28      0.40      0.33       273
30        0.22      0.39      0.28       308
31        0.19      0.36      0.25       256
32        0.28      0.42      0.34       295
33        0.09      0.19      0.12       263
34        0.34      0.57      0.43       256
35        0.21      0.37      0.27       280
36        0.31      0.45      0.37       290
37        0.19      0.28      0.23       200
38        0.11      0.39      0.18       109
39        0.22      0.38      0.28       209
40        0.16      0.43      0.23       113
41        0.21      0.37      0.27       197
42        0.14      0.63      0.23        52
43        0.06      0.15      0.08       179
44        0.53      0.53      0.53       431
45        0.05      0.17      0.07        47
46        0.10      0.51      0.17        37
47        0.19      0.35      0.25       155
48        0.35      0.48      0.41       254
49        0.20      0.37      0.26       201
```

```
50    0.11   0.43   0.17    61

51    0.52   0.79   0.63   246
52    0.34   0.68   0.46   146
53    0.74   0.91   0.81   516
54    0.33   0.58   0.42   170
55    0.10   0.17   0.12   234
56    0.12   0.19   0.14   357
57    0.07   0.29   0.11    78
58    0.40   0.71   0.51   102
59    0.12   0.31   0.18   122
60    0.38   0.62   0.48   138
61    0.04   0.22   0.07    36
62    0.11   0.27   0.16   172
63    0.01   0.05   0.02    60
64    0.19   0.49   0.27   106
65    0.19   0.44   0.27    34
66    0.08   0.28   0.13   101
67    0.07   0.26   0.11    38
68    0.17   0.43   0.25   104
69    0.09   0.17   0.12   144
70    0.17   0.30   0.22   135
71    0.11   0.20   0.14   190
72    0.25   0.44   0.32   139
73    0.04   0.16   0.07    69
74    0.08   0.16   0.11   133
75    0.38   0.47   0.42   181
76    0.19   0.47   0.27   113
77    0.15   0.37   0.21   158
78    0.11   0.23   0.15   142
79    0.13   0.23   0.17    96
80    0.26   0.50   0.34   101
81    0.10   0.27   0.14    56
82    0.05   0.21   0.08    62
83    0.17   0.43   0.25    77
84    0.10   0.28   0.15   100
85    0.14   0.52   0.22    54
86    0.07   0.27   0.11    79
87    0.18   0.37   0.24    92
88    0.16   0.34   0.21   124
```

```
 89     0.23    0.48    0.31    101

 90     0.03    0.12    0.04     40
 91     0.20    0.36    0.26     66
 92     0.08    0.43    0.13     58
 93     0.34    0.52    0.41    161
 94     0.12    0.28    0.17    130
 95     0.14    0.23    0.18     47
 96     0.24    0.52    0.33    107
 97     0.09    0.44    0.15     39
 98     0.07    0.20    0.11    111
 99     0.12    0.25    0.16     95
100     0.11    0.22    0.14    129
101     0.26    0.51    0.35     91
102     0.03    0.19    0.05     27
103     0.43    0.86    0.57     90
104     0.09    0.21    0.12    124
105     0.06    0.18    0.09     76
106     0.26    0.32    0.29    371
107     0.25    0.41    0.31    114
108     0.17    0.44    0.25     98
109     0.11    0.35    0.16     63
110     0.11    0.46    0.18     24
111     0.15    0.43    0.22     53
112     0.07    0.18    0.10     65
113     0.11    0.36    0.17     70
114     0.12    0.59    0.20     27
115     0.11    0.29    0.16     72
116     0.16    0.33    0.22     27
117     0.19    0.39    0.25     90
118     0.16    0.39    0.22     95
119     0.16    0.40    0.23     92
120     0.15    0.34    0.21     87
121     0.17    0.38    0.24     45
122     0.08    0.12    0.09    182
123     0.12    0.31    0.17     94
124     0.22    0.48    0.30     62
125     0.18    0.48    0.27     91
126     0.13    0.45    0.21     69
127     0.22    0.47    0.30     73
```

```
128       0.12      0.44      0.19        25

129       0.02      0.09      0.04        68
130       0.12      0.26      0.16       123
131       0.06      0.21      0.10        84
132       0.04      0.10      0.05        67
133       0.11      0.33      0.16       127
134       0.10      0.42      0.16        45
135       0.18      0.43      0.25        88
136       0.04      0.16      0.07        63
137       0.60      0.79      0.68        96
138       0.06      0.24      0.10        71
139       0.34      0.71      0.46        92
140       0.03      0.13      0.05        23
141       0.12      0.34      0.17        90
142       0.02      0.30      0.04        10
143       0.07      0.25      0.11        44
144       0.20      0.51      0.28        67
145       0.42      0.44      0.43       131
146       0.08      0.18      0.11        83
147       0.08      0.25      0.12        32
148       0.13      0.24      0.17       115
149       0.19      0.44      0.26        63
150       0.24      0.48      0.32        83
151       0.33      0.43      0.37       101
152       0.06      0.21      0.09        29
153       0.76      0.79      0.78       191
154       0.34      0.67      0.45        54
155       0.20      0.39      0.27        84
156       0.16      0.57      0.25        37
157       0.10      0.34      0.15        65
158       0.10      0.37      0.15        46
159       0.21      0.53      0.30        80
160       0.05      0.20      0.08        66
161       0.05      0.14      0.07        56
162       0.16      0.20      0.18       127
163       0.44      0.65      0.52       111
164       0.05      0.19      0.07        32
165       0.02      0.11      0.03        28
166       0.05      0.12      0.07        98
```

```
167       0.30      0.62      0.41        88

168       0.27      0.53      0.36        59
169       0.05      0.21      0.08        42
170       0.05      0.50      0.09         4
171       0.19      0.45      0.27        95
172       0.03      0.15      0.05        54
173       0.29      0.58      0.39        65
174       0.16      0.52      0.24        31
175       0.09      0.47      0.16        32
176       0.14      0.38      0.21        58
177       0.10      0.24      0.14        76
178       0.03      0.13      0.04        55
179       0.55      0.81      0.66        74
180       0.31      0.70      0.43        64
181       0.07      0.16      0.09        57
182       0.07      0.39      0.12        36
183       0.13      0.44      0.20        52
184       0.20      0.46      0.28        48
185       0.20      0.56      0.29        16
186       0.03      0.18      0.06        28
187       0.12      0.28      0.17        36
188       0.10      0.35      0.16        26
189       0.09      0.30      0.13        44
190       0.15      0.33      0.21        46
191       0.09      0.27      0.14        75
192       0.05      0.28      0.09        50
193       0.17      0.40      0.24        20
194       0.03      0.15      0.05        27
195       0.06      0.50      0.10         6
196       0.14      0.24      0.18        68
197       0.14      0.48      0.21        29
198       0.12      0.28      0.17       104
199       0.08      0.39      0.14        36
200       0.17      1.00      0.29         4
201       0.07      0.50      0.12         4
202       0.05      0.16      0.08        96
203       0.41      0.64      0.50        61
204       0.11      0.28      0.15        82
205       0.15      0.53      0.24        34
```

```
206        0.30       0.52       0.38        66

207        0.09       0.38       0.15        97
208        0.05       0.13       0.08        89
209        0.34       0.62       0.44        55
210        0.28       0.63       0.38        78
211        0.06       0.18       0.09        78
212        0.38       0.52       0.44       158
213        0.04       0.18       0.07        44
214        0.17       0.54       0.26        35
215        0.37       0.73       0.49        48
216        0.31       0.63       0.42        62
217        0.00       0.00       0.00        11
218        0.28       0.54       0.37        68
219        0.13       0.38       0.20        60
220        0.02       0.16       0.04        25
221        0.13       0.37       0.19        57
222        0.08       0.50       0.14        36
223        0.08       0.23       0.12        88
224        0.05       0.22       0.08        46
225        0.04       0.18       0.06        60
226        0.10       0.28       0.15        65
227        0.05       0.57       0.09         7
228        0.14       0.67       0.24        12
229        0.02       0.09       0.04        68
230        0.06       0.23       0.10        40
231        0.04       0.15       0.07        26
232        0.17       0.70       0.28        30
233        0.11       0.27       0.15        41
234        0.08       0.28       0.13        53
235        0.12       0.54       0.20        35
236        0.04       0.17       0.07        18
237        0.02       0.09       0.03        22
238        0.24       0.68       0.36        59
239        0.24       0.51       0.32        43
240        0.08       0.24       0.12        45
241        0.04       0.15       0.06        46
242        0.04       0.21       0.07        38
243        0.23       0.38       0.28        56
244        0.05       0.29       0.09        35
```

```
245        0.03       0.14       0.05         42

246        0.03       0.12       0.05         33
247        0.14       0.34       0.20         47
248        0.05       0.36       0.08         25
249        0.15       0.62       0.25         39
250        0.21       0.45       0.29         77
251        0.34       0.61       0.44         56
252        0.09       0.22       0.13         72
253        0.18       1.00       0.31          4
254        0.13       0.59       0.21         29
255        0.18       0.23       0.20        113
256        0.34       0.71       0.46         59
257        0.04       0.10       0.06         59
258        0.32       0.62       0.42         39
259        0.11       0.75       0.19         12
260        0.06       0.56       0.12          9
261        0.38       0.80       0.51         44
262        0.14       0.44       0.21         32
263        0.08       0.27       0.13        156
264        0.10       0.60       0.18          5
265        0.14       0.27       0.18        198
266        0.04       0.12       0.05         40
267        0.08       0.38       0.13         29
268        0.00       0.00       0.00         39
269        0.01       0.17       0.02          6
270        0.06       0.60       0.11          5
271        0.05       0.41       0.09         17
272        0.10       0.33       0.16         54
273        0.05       0.30       0.09         23
274        0.02       0.03       0.02        126
275        0.12       0.41       0.19         32
276        0.08       0.60       0.15         10
277        0.29       0.51       0.37         67
278        0.21       0.38       0.27         53
279        0.01       0.06       0.02         16
280        0.18       0.63       0.28         19
281        0.05       0.13       0.07         61
282        0.11       0.25       0.15         81
283        0.19       0.39       0.26         94
```

```
284        0.15       0.42       0.22        31

285        0.06       0.26       0.09        43
286        0.13       0.29       0.18        79
287        0.16       0.50       0.24        20
288        0.19       0.88       0.32        17
289        0.35       0.61       0.45        56
290        0.04       0.14       0.06        63
291        0.10       0.30       0.15        46
292        0.08       0.28       0.12        50
293        0.05       0.35       0.08        17
294        0.07       0.33       0.11        27
295        0.02       0.08       0.04        63
296        0.08       0.28       0.12        25
297        0.01       0.03       0.01        38
298        0.04       0.13       0.06        62
299        0.35       0.63       0.45        49
300        0.04       0.23       0.07        39
301        0.01       0.12       0.02         8
302        0.03       0.22       0.05        18
303        0.09       0.32       0.14        19
304        0.12       0.44       0.19        18
305        0.14       0.69       0.23        26
306        0.04       0.30       0.07        10
307        0.06       0.17       0.09        12
308        0.09       0.50       0.15         8
309        0.12       0.33       0.18        57
310        0.08       0.22       0.12        37
311        0.42       0.68       0.52        50
312        0.12       0.33       0.17        36
313        0.02       0.16       0.04        19
314        0.42       0.76       0.54        59
315        0.38       0.62       0.47        48
316        0.09       0.22       0.12        45
317        0.09       0.31       0.14        29
318        0.14       0.43       0.21        37
319        0.07       0.21       0.10        38
320        0.36       0.59       0.45        44
321        0.29       0.51       0.37        41
322        0.01       0.04       0.01        25
```

```
323        0.00       0.00       0.00          4

324        0.02       0.09       0.03         11
325        0.23       1.00       0.38          3
326        0.09       0.31       0.14         32
327        0.16       0.36       0.22         50
328        0.07       0.26       0.12         46
329        0.56       0.74       0.64         47
330        0.42       0.81       0.56         31
331        0.00       0.00       0.00         11
332        0.02       0.10       0.04         31
333        0.03       0.12       0.04         26
334        0.00       0.00       0.00         41
335        0.06       0.21       0.09         39
336        0.14       0.33       0.20         40
337        0.09       0.27       0.14         37
338        0.14       0.50       0.22         10
339        0.00       0.00       0.00         38
340        0.57       0.87       0.69         23
341        0.05       0.24       0.09         33
342        0.03       0.17       0.06         40
343        0.05       0.14       0.07         35
344        0.10       0.30       0.15         30
345        0.02       0.09       0.04         33
346        0.12       0.52       0.19         25
347        0.07       0.44       0.11          9
348        0.03       0.50       0.05          2
349        0.26       0.65       0.37         34
350        0.09       0.37       0.14         38
351        0.19       0.41       0.26         49
352        0.01       0.09       0.02         22
353        0.07       0.18       0.10         39
354        0.06       0.28       0.10         18
355        0.05       0.19       0.08         31
356        0.02       0.12       0.03         17
357        0.25       0.54       0.34         35
358        0.04       0.09       0.05         43
359        0.21       0.66       0.32         47
360        0.04       0.17       0.07         29
361        0.37       0.66       0.48         38
```

362	0.16	0.39	0.23	36
363	0.22	0.44	0.29	55
364	0.09	0.24	0.13	34
365	0.08	0.25	0.13	24
366	0.03	0.21	0.05	19
367	0.04	0.60	0.08	5
368	0.07	0.27	0.11	33
369	0.21	0.62	0.31	34
370	0.16	0.50	0.24	20
371	0.02	0.18	0.04	17
372	0.07	0.26	0.11	31
373	0.04	0.12	0.06	34
374	0.03	0.13	0.05	23
375	0.15	0.39	0.22	31
376	0.08	0.32	0.13	28
377	0.26	0.77	0.39	22
378	0.01	0.11	0.02	9
379	0.09	0.24	0.14	29
380	0.11	0.33	0.16	30
381	0.19	0.74	0.30	35
382	0.18	0.42	0.25	36
383	0.01	0.25	0.03	4
384	0.37	0.88	0.52	24
385	0.10	0.36	0.15	25
386	0.06	0.22	0.10	27
387	0.14	0.39	0.21	36
388	0.19	0.45	0.27	31
389	0.21	0.49	0.30	37
390	0.17	0.41	0.24	27
391	0.19	0.46	0.26	46
392	0.03	0.50	0.06	4
393	0.11	0.42	0.17	19
394	0.03	0.33	0.06	12
395	0.10	0.38	0.16	26
396	0.15	0.29	0.20	69
397	0.15	0.44	0.23	25
398	0.06	0.25	0.10	32
399	0.06	0.21	0.10	33
400	0.05	0.16	0.08	38

```
401     0.09    0.29    0.14     17

402     0.03    0.21    0.05     24
403     0.10    0.62    0.17     16
404     0.16    0.53    0.25     15
405     0.06    0.25    0.10     20
406     0.02    0.20    0.04     15
407     0.07    0.32    0.11     25
408     0.04    0.16    0.07     19
409     0.09    0.33    0.14     46
410     0.00    0.00    0.00     45
411     0.05    0.29    0.08     21
412     0.00    0.00    0.00      8
413     0.25    0.57    0.35     35
414     0.02    0.09    0.04     34
415     0.24    0.79    0.37     14
416     0.07    0.21    0.11     29
417     0.05    0.14    0.07     28
418     0.14    0.19    0.16     21
419     0.05    0.15    0.07     26
420     0.21    0.32    0.25     38
421     0.19    0.31    0.24    131
422     0.03    0.19    0.05     26
423     0.11    0.32    0.17     25
424     0.59    0.56    0.57     48
425     0.03    0.17    0.05     24
426     0.08    0.31    0.13     42
427     0.13    0.35    0.19     26
428     0.03    0.20    0.05     10
429     0.27    0.54    0.36     54
430     0.11    0.31    0.16     32
431     0.22    0.46    0.30     48
432     0.12    0.31    0.17     35
433     0.02    0.14    0.04     22
434     0.05    0.21    0.08     24
435     0.40    0.61    0.49     59
436     0.03    0.11    0.05     35
437     0.00    0.00    0.00     12
438     0.26    0.32    0.29     50
439     0.05    0.17    0.07     36
```

```
440       0.06      0.14      0.09        35

441       0.00      0.00      0.00         8
442       0.10      0.23      0.14        48
443       0.14      0.50      0.22        18
444       0.25      0.37      0.30        52
445       0.08      0.30      0.12        20
446       0.07      0.22      0.10        18
447       0.02      0.17      0.04         6
448       0.10      0.21      0.13        28
449       0.06      0.16      0.09        38
450       0.98      0.63      0.76       129
451       0.33      0.72      0.45        29
452       0.09      0.14      0.11        58
453       0.00      0.00      0.00        32
454       0.22      0.62      0.33        16
455       0.37      0.65      0.47        34
456       0.08      0.44      0.13         9
457       0.45      0.87      0.59        30
458       0.12      0.33      0.18        40
459       0.13      0.30      0.18        37
460       0.35      0.77      0.48        30
461       0.08      0.33      0.13        27
462       0.05      0.25      0.09        12
463       0.08      0.41      0.13        17
464       0.31      0.59      0.41        56
465       0.21      0.44      0.29         9
466       0.16      0.45      0.24        22
467       0.04      0.33      0.08         9
468       0.04      0.33      0.07        15
469       0.01      0.07      0.02        14
470       0.07      0.50      0.13        16
471       0.17      0.67      0.28        18
472       0.05      0.25      0.08        16
473       0.24      0.50      0.32        22
474       0.06      0.25      0.10        12
475       0.92      0.79      0.85       110
476       0.24      0.65      0.35        20
477       0.22      0.55      0.32        31
478       0.11      0.36      0.17        42
```

```
         479      0.01      0.25      0.02         4

         480      0.16      0.43      0.24        47
         481      0.25      0.53      0.34        30
         482      0.03      0.11      0.05        35
         483      0.11      0.23      0.15        30
         484      0.08      0.40      0.14        20
         485      0.19      0.53      0.28        15
         486      0.04      0.12      0.05        17
         487      0.01      0.09      0.02        11
         488      0.09      0.25      0.14        36
         489      0.08      0.22      0.12        32
         490      0.11      0.43      0.17        14
         491      0.14      0.40      0.21        25
         492      0.20      0.48      0.28        23
         493      0.00      0.00      0.00         9
         494      0.14      0.22      0.17        37
         495      0.03      0.12      0.04        24
         496      0.18      0.44      0.25        34
         497      0.06      0.20      0.10        25
         498      0.01      0.11      0.02         9
         499      0.10      0.35      0.15        17

   micro avg      0.23      0.44      0.30     55953
   macro avg      0.16      0.37      0.21     55953
weighted avg      0.31      0.44      0.35     55953
 samples avg      0.32      0.42      0.32     55953

Time taken to run this cell : 1:15:27.156165
```

In [76]:

```
start = datetime.now()
classifier_2 = OneVsRestClassifier(LogisticRegression(penalty='l1'))
classifier_2.fit(x_train_multilabel_bow, y_train)
predictions_2 = classifier_2.predict(x_test_multilabel_bow)
print("Accuracy :",metrics.accuracy_score(y_test, predictions_2))
print("Hamming loss ",metrics.hamming_loss(y_test,predictions_2))


precision = precision_score(y_test, predictions_2, average='micro')
```

```
recall = recall_score(y_test, predictions_2, average='micro')
f1 = f1_score(y_test, predictions_2, average='micro')

global_report = global_report.append({
                        'Vectorizer': 'BoW',
                        'Model': 'Logistic Regression',
                        'NGram': '(1,4)',
                        'Parameter': 1,
                        'Precision': precision,
                        'Recall': recall,
                        'F1_Score_Micro':f1
                     },
                    ignore_index=True)

print("Micro-average quality numbers")
print("Precision: {:.4f}, Recall: {:.4f}, F1-measure: {:.4f}".format(pr
ecision, recall, f1))

precision = precision_score(y_test, predictions_2, average='macro')
recall = recall_score(y_test, predictions_2, average='macro')
f1 = f1_score(y_test, predictions_2, average='macro')

print("Macro-average quality numbers")
print("Precision: {:.4f}, Recall: {:.4f}, F1-measure: {:.4f}".format(pr
ecision, recall, f1))

print (metrics.classification_report(y_test, predictions_2))
print("Time taken to run this cell :", datetime.now() - start)
```

```
Accuracy : 0.18633333333333332
Hamming loss  0.003399333333333333
Micro-average quality numbers
Precision: 0.5644, Recall: 0.3887, F1-measure: 0.4604
Macro-average quality numbers
Precision: 0.4320, Recall: 0.3112, F1-measure: 0.3527
              precision    recall  f1-score   support

           0       0.61      0.40      0.48      1111
           1       0.47      0.30      0.36      2052
           2       0.54      0.42      0.47      2388
```

```
 3     0.66     0.56     0.60     2226
 4     0.66     0.50     0.57     2014
 5     0.30     0.14     0.19      642
 6     0.64     0.41     0.50     1756
 7     0.89     0.72     0.79     1690
 8     0.55     0.24     0.34      341
 9     0.77     0.78     0.78     2344
10     0.57     0.40     0.47      821
11     0.43     0.29     0.34     1143
12     0.64     0.40     0.49      768
13     0.55     0.37     0.44      745
14     0.69     0.56     0.62      952
15     0.46     0.27     0.34      314
16     0.41     0.26     0.32      624
17     0.72     0.56     0.63      535
18     0.73     0.54     0.62      631
19     0.88     0.57     0.69      101
20     0.53     0.26     0.35      245
21     0.73     0.54     0.62      694
22     0.48     0.32     0.38      568
23     0.52     0.32     0.40      423
24     0.53     0.34     0.42      406
25     0.70     0.49     0.58     1373
26     0.52     0.34     0.41      253
27     0.24     0.16     0.19      357
28     0.50     0.28     0.36      222
29     0.58     0.30     0.40      273
30     0.48     0.32     0.38      308
31     0.45     0.30     0.36      256
32     0.69     0.41     0.52      295
33     0.22     0.11     0.15      263
34     0.74     0.53     0.62      256
35     0.42     0.34     0.37      280
36     0.47     0.32     0.38      290
37     0.36     0.17     0.23      200
38     0.34     0.29     0.32      109
39     0.62     0.39     0.48      209
40     0.52     0.39     0.45      113
41     0.54     0.28     0.37      197
```

```
42       0.51      0.56      0.53        52
43       0.17      0.10      0.13       179
44       0.77      0.50      0.61       431
45       0.29      0.11      0.16        47
46       0.44      0.38      0.41        37
47       0.54      0.34      0.41       155
48       0.65      0.50      0.57       254
49       0.50      0.34      0.41       201
50       0.40      0.28      0.33        61
51       0.92      0.73      0.82       246
52       0.67      0.62      0.64       146
53       0.92      0.91      0.92       516
54       0.75      0.55      0.64       170
55       0.21      0.10      0.13       234
56       0.19      0.11      0.14       357
57       0.55      0.27      0.36        78
58       0.87      0.68      0.76       102
59       0.35      0.24      0.28       122
60       0.72      0.57      0.64       138
61       0.22      0.17      0.19        36
62       0.35      0.19      0.25       172
63       0.08      0.03      0.05        60
64       0.59      0.51      0.55       106
65       0.35      0.32      0.34        34
66       0.29      0.25      0.27       101
67       0.43      0.34      0.38        38
68       0.56      0.39      0.46       104
69       0.32      0.17      0.22       144
70       0.38      0.24      0.29       135
71       0.24      0.14      0.17       190
72       0.56      0.36      0.44       139
73       0.21      0.04      0.07        69
74       0.07      0.04      0.05       133
75       0.73      0.46      0.57       181
76       0.43      0.43      0.43       113
77       0.42      0.22      0.29       158
78       0.28      0.15      0.19       142
79       0.45      0.18      0.25        96
80       0.52      0.25      0.34       101
```

```
 81    0.42    0.18    0.25     56
 82    0.13    0.05    0.07     62
 83    0.58    0.44    0.50     77
 84    0.45    0.15    0.23    100
 85    0.55    0.43    0.48     54
 86    0.30    0.16    0.21     79
 87    0.44    0.24    0.31     92
 88    0.45    0.31    0.37    124
 89    0.60    0.39    0.47    101
 90    0.27    0.10    0.15     40
 91    0.56    0.44    0.49     66
 92    0.42    0.31    0.36     58
 93    0.67    0.42    0.51    161
 94    0.40    0.25    0.30    130
 95    0.44    0.26    0.32     47
 96    0.64    0.50    0.57    107
 97    0.50    0.31    0.38     39
 98    0.22    0.12    0.15    111
 99    0.39    0.22    0.28     95
100    0.29    0.22    0.25    129
101    0.74    0.46    0.57     91
102    0.19    0.19    0.19     27
103    0.85    0.84    0.85     90
104    0.20    0.07    0.11    124
105    0.30    0.17    0.22     76
106    0.35    0.18    0.24    371
107    0.67    0.40    0.50    114
108    0.49    0.37    0.42     98
109    0.60    0.40    0.48     63
110    0.63    0.50    0.56     24
111    0.54    0.28    0.37     53
112    0.16    0.09    0.12     65
113    0.53    0.36    0.43     70
114    0.50    0.56    0.53     27
115    0.31    0.12    0.18     72
116    0.33    0.19    0.24     27
117    0.54    0.32    0.40     90
118    0.50    0.40    0.44     95
119    0.37    0.27    0.31     92
```

```
120        0.28      0.21      0.24       87
121        0.57      0.36      0.44       45
122        0.18      0.08      0.11      182
123        0.31      0.12      0.17       94
124        0.72      0.42      0.53       62
125        0.73      0.44      0.55       91
126        0.69      0.51      0.58       69
127        0.49      0.48      0.48       73
128        1.00      0.52      0.68       25
129        0.20      0.03      0.05       68
130        0.31      0.19      0.23      123
131        0.25      0.15      0.19       84
132        0.00      0.00      0.00       67
133        0.30      0.20      0.24      127
134        0.47      0.60      0.52       45
135        0.49      0.42      0.45       88
136        0.00      0.00      0.00       63
137        0.89      0.77      0.83       96
138        0.19      0.08      0.12       71
139        0.84      0.64      0.73       92
140        0.23      0.13      0.17       23
141        0.42      0.23      0.30       90
142        0.15      0.20      0.17       10
143        0.27      0.18      0.22       44
144        0.59      0.43      0.50       67
145        0.65      0.46      0.54      131
146        0.17      0.07      0.10       83
147        0.27      0.12      0.17       32
148        0.30      0.15      0.20      115
149        0.47      0.29      0.36       63
150        0.62      0.35      0.45       83
151        0.73      0.47      0.57      101
152        0.27      0.10      0.15       29
153        0.92      0.85      0.89      191
154        0.86      0.69      0.76       54
155        0.58      0.35      0.43       84
156        0.50      0.38      0.43       37
157        0.32      0.38      0.35       65
158        0.33      0.20      0.25       46
```

```
159     0.79     0.53     0.63      80
160     0.19     0.11     0.14      66
161     0.17     0.07     0.10      56
162     0.41     0.23     0.29     127
163     0.67     0.59     0.63     111
164     0.17     0.06     0.09      32
165     0.25     0.14     0.18      28
166     0.03     0.01     0.02      98
167     0.68     0.58     0.63      88
168     0.76     0.54     0.63      59
169     0.05     0.02     0.03      42
170     0.29     0.50     0.36       4
171     0.58     0.40     0.47      95
172     0.27     0.17     0.21      54
173     0.74     0.54     0.62      65
174     0.60     0.48     0.54      31
175     0.62     0.47     0.54      32
176     0.54     0.34     0.42      58
177     0.42     0.21     0.28      76
178     0.05     0.02     0.03      55
179     0.80     0.85     0.82      74
180     0.78     0.62     0.70      64
181     0.35     0.12     0.18      57
182     0.47     0.25     0.33      36
183     0.35     0.42     0.38      52
184     0.49     0.44     0.46      48
185     0.25     0.31     0.28      16
186     0.12     0.07     0.09      28
187     0.16     0.17     0.16      36
188     0.28     0.31     0.29      26
189     0.19     0.09     0.12      44
190     0.55     0.35     0.43      46
191     0.32     0.20     0.25      75
192     0.21     0.24     0.22      50
193     0.50     0.45     0.47      20
194     0.31     0.19     0.23      27
195     0.38     0.50     0.43       6
196     0.17     0.13     0.15      68
197     0.33     0.41     0.37      29
```

```
198         0.33        0.17        0.23         104
199         0.45        0.25        0.32          36
200         1.00        1.00        1.00           4
201         0.75        0.75        0.75           4
202         0.25        0.07        0.11          96
203         0.82        0.61        0.70          61
204         0.45        0.17        0.25          82
205         0.46        0.35        0.40          34
206         0.66        0.41        0.50          66
207         0.17        0.09        0.12          97
208         0.14        0.06        0.08          89
209         0.78        0.53        0.63          55
210         0.72        0.54        0.62          78
211         0.22        0.08        0.11          78
212         0.68        0.47        0.56         158
213         0.24        0.09        0.13          44
214         0.62        0.60        0.61          35
215         0.87        0.71        0.78          48
216         0.66        0.66        0.66          62
217         0.50        0.36        0.42          11
218         0.81        0.44        0.57          68
219         0.31        0.22        0.25          60
220         0.33        0.08        0.13          25
221         0.31        0.14        0.19          57
222         0.59        0.53        0.56          36
223         0.37        0.17        0.23          88
224         0.26        0.15        0.19          46
225         0.32        0.10        0.15          60
226         0.26        0.14        0.18          65
227         0.50        0.57        0.53           7
228         0.41        0.58        0.48          12
229         0.18        0.13        0.15          68
230         0.44        0.28        0.34          40
231         0.08        0.04        0.05          26
232         0.72        0.60        0.65          30
233         0.33        0.20        0.25          41
234         0.18        0.08        0.11          53
235         0.51        0.54        0.53          35
236         0.62        0.28        0.38          18
```

```
237       0.17      0.09      0.12        22
238       0.67      0.56      0.61        59
239       0.67      0.56      0.61        43
240       0.34      0.27      0.30        45
241       0.33      0.13      0.19        46
242       0.13      0.05      0.08        38
243       0.94      0.29      0.44        56
244       0.24      0.17      0.20        35
245       0.10      0.05      0.06        42
246       0.07      0.03      0.04        33
247       0.38      0.28      0.32        47
248       0.62      0.20      0.30        25
249       0.56      0.59      0.57        39
250       0.56      0.19      0.29        77
251       0.74      0.55      0.63        56
252       0.20      0.06      0.09        72
253       0.67      1.00      0.80         4
254       0.50      0.38      0.43        29
255       0.41      0.27      0.32       113
256       0.75      0.71      0.73        59
257       0.18      0.05      0.08        59
258       0.83      0.62      0.71        39
259       0.64      0.58      0.61        12
260       0.56      0.56      0.56         9
261       0.94      0.66      0.77        44
262       0.76      0.50      0.60        32
263       0.18      0.17      0.17       156
264       1.00      0.60      0.75         5
265       0.18      0.07      0.10       198
266       0.07      0.03      0.04        40
267       0.20      0.14      0.16        29
268       0.00      0.00      0.00        39
269       0.11      0.17      0.13         6
270       0.75      0.60      0.67         5
271       0.40      0.12      0.18        17
272       0.30      0.13      0.18        54
273       0.38      0.26      0.31        23
274       0.00      0.00      0.00       126
275       0.44      0.34      0.39        32
```

```
276     0.40    0.60    0.48    10
277     0.82    0.61    0.70    67
278     0.47    0.36    0.41    53
279     0.33    0.12    0.18    16
280     0.72    0.68    0.70    19
281     0.30    0.13    0.18    61
282     0.38    0.21    0.27    81
283     0.41    0.34    0.37    94
284     0.46    0.35    0.40    31
285     0.31    0.21    0.25    43
286     0.35    0.22    0.27    79
287     0.29    0.40    0.33    20
288     0.83    0.88    0.86    17
289     0.73    0.64    0.69    56
290     0.15    0.08    0.10    63
291     0.48    0.33    0.39    46
292     0.43    0.36    0.39    50
293     0.62    0.29    0.40    17
294     0.36    0.15    0.21    27
295     0.04    0.02    0.02    63
296     0.23    0.12    0.16    25
297     0.04    0.03    0.03    38
298     0.25    0.10    0.14    62
299     0.67    0.59    0.63    49
300     0.19    0.15    0.17    39
301     0.00    0.00    0.00     8
302     0.17    0.11    0.13    18
303     0.15    0.11    0.12    19
304     0.60    0.67    0.63    18
305     0.44    0.46    0.45    26
306     0.23    0.30    0.26    10
307     0.17    0.17    0.17    12
308     0.60    0.38    0.46     8
309     0.37    0.12    0.18    57
310     0.31    0.22    0.25    37
311     0.80    0.72    0.76    50
312     0.38    0.31    0.34    36
313     0.24    0.26    0.25    19
314     0.88    0.71    0.79    59
```

```
315        0.79      0.71      0.75        48
316        0.22      0.13      0.17        45
317        0.62      0.34      0.44        29
318        0.33      0.32      0.33        37
319        0.24      0.11      0.15        38
320        0.81      0.68      0.74        44
321        0.64      0.39      0.48        41
322        0.15      0.08      0.11        25
323        0.25      0.25      0.25         4
324        0.12      0.09      0.11        11
325        0.75      1.00      0.86         3
326        0.45      0.28      0.35        32
327        0.49      0.34      0.40        50
328        0.26      0.11      0.15        46
329        0.90      0.74      0.81        47
330        0.91      0.68      0.78        31
331        0.00      0.00      0.00        11
332        0.00      0.00      0.00        31
333        0.08      0.04      0.05        26
334        0.07      0.02      0.04        41
335        0.08      0.03      0.04        39
336        0.69      0.45      0.55        40
337        0.28      0.22      0.24        37
338        0.83      0.50      0.62        10
339        0.00      0.00      0.00        38
340        0.94      0.70      0.80        23
341        0.26      0.15      0.19        33
342        0.16      0.07      0.10        40
343        0.38      0.17      0.24        35
344        0.52      0.40      0.45        30
345        0.08      0.03      0.04        33
346        0.65      0.44      0.52        25
347        0.75      0.33      0.46         9
348        0.50      0.50      0.50         2
349        0.79      0.68      0.73        34
350        0.36      0.47      0.41        38
351        0.57      0.27      0.36        49
352        0.00      0.00      0.00        22
353        0.35      0.23      0.28        39
```

```
354     0.50    0.17    0.25    18
355     0.21    0.13    0.16    31
356     0.14    0.06    0.08    17
357     0.73    0.69    0.71    35
358     0.44    0.09    0.15    43
359     0.56    0.64    0.59    47
360     0.20    0.10    0.14    29
361     0.65    0.58    0.61    38
362     0.41    0.31    0.35    36
363     0.36    0.36    0.36    55
364     0.29    0.21    0.24    34
365     0.27    0.17    0.21    24
366     0.12    0.11    0.11    19
367     0.22    0.40    0.29     5
368     0.20    0.12    0.15    33
369     0.69    0.65    0.67    34
370     0.42    0.40    0.41    20
371     0.07    0.06    0.06    17
372     0.31    0.13    0.18    31
373     0.11    0.03    0.05    34
374     0.15    0.13    0.14    23
375     0.41    0.39    0.40    31
376     0.62    0.29    0.39    28
377     0.78    0.64    0.70    22
378     0.36    0.56    0.43     9
379     0.33    0.21    0.26    29
380     0.26    0.27    0.26    30
381     0.72    0.60    0.66    35
382     0.41    0.31    0.35    36
383     0.17    0.25    0.20     4
384     0.90    0.79    0.84    24
385     0.60    0.36    0.45    25
386     0.08    0.04    0.05    27
387     0.75    0.33    0.46    36
388     0.61    0.45    0.52    31
389     0.65    0.46    0.54    37
390     0.39    0.33    0.36    27
391     0.58    0.41    0.48    46
392     0.25    0.50    0.33     4
```

```
393    0.26    0.32    0.29     19
394    0.00    0.00    0.00     12
395    0.50    0.42    0.46     26
396    0.19    0.04    0.07     69
397    0.67    0.72    0.69     25
398    0.14    0.06    0.09     32
399    0.32    0.18    0.23     33
400    0.32    0.18    0.23     38
401    0.50    0.29    0.37     17
402    0.12    0.04    0.06     24
403    0.73    0.50    0.59     16
404    0.83    0.33    0.48     15
405    0.00    0.00    0.00     20
406    0.00    0.00    0.00     15
407    0.21    0.12    0.15     25
408    0.29    0.26    0.28     19
409    0.53    0.37    0.44     46
410    0.00    0.00    0.00     45
411    0.17    0.10    0.12     21
412    0.00    0.00    0.00      8
413    0.50    0.51    0.51     35
414    0.00    0.00    0.00     34
415    1.00    0.57    0.73     14
416    0.23    0.10    0.14     29
417    0.33    0.14    0.20     28
418    0.43    0.14    0.21     21
419    0.25    0.15    0.19     26
420    0.52    0.29    0.37     38
421    0.17    0.02    0.04    131
422    0.29    0.23    0.26     26
423    0.40    0.40    0.40     25
424    0.81    0.62    0.71     48
425    0.17    0.08    0.11     24
426    0.29    0.14    0.19     42
427    0.39    0.35    0.37     26
428    0.11    0.10    0.11     10
429    0.54    0.39    0.45     54
430    0.55    0.34    0.42     32
431    0.39    0.27    0.32     48
```

```
432     0.44    0.20    0.27     35
433     0.38    0.14    0.20     22
434     0.18    0.12    0.15     24
435     0.78    0.59    0.67     59
436     0.22    0.14    0.17     35
437     0.00    0.00    0.00     12
438     0.61    0.38    0.47     50
439     0.09    0.03    0.04     36
440     0.21    0.14    0.17     35
441     0.43    0.38    0.40      8
442     0.21    0.19    0.20     48
443     0.75    0.50    0.60     18
444     0.64    0.35    0.45     52
445     0.60    0.15    0.24     20
446     0.14    0.06    0.08     18
447     0.00    0.00    0.00      6
448     0.27    0.14    0.19     28
449     0.25    0.13    0.17     38
450     0.98    0.84    0.91    129
451     0.72    0.72    0.72     29
452     0.26    0.12    0.16     58
453     0.10    0.03    0.05     32
454     0.43    0.38    0.40     16
455     0.92    0.68    0.78     34
456     0.40    0.44    0.42      9
457     0.86    0.83    0.85     30
458     0.36    0.23    0.28     40
459     0.47    0.22    0.30     37
460     0.81    0.73    0.77     30
461     0.55    0.41    0.47     27
462     0.14    0.08    0.11     12
463     0.33    0.29    0.31     17
464     0.80    0.50    0.62     56
465     0.44    0.44    0.44      9
466     0.38    0.23    0.29     22
467     0.40    0.22    0.29      9
468     0.00    0.00    0.00     15
469     0.25    0.07    0.11     14
470     0.55    0.38    0.44     16
```

```
         471       0.85      0.61      0.71        18
         472       0.29      0.31      0.30        16
         473       0.46      0.55      0.50        22
         474       0.67      0.17      0.27        12
         475       1.00      0.97      0.99       110
         476       0.48      0.70      0.57        20
         477       0.62      0.52      0.56        31
         478       0.44      0.17      0.24        42
         479       0.00      0.00      0.00         4
         480       0.52      0.26      0.34        47
         481       0.75      0.40      0.52        30
         482       0.00      0.00      0.00        35
         483       0.00      0.00      0.00        30
         484       0.38      0.15      0.21        20
         485       0.80      0.53      0.64        15
         486       0.33      0.18      0.23        17
         487       0.33      0.09      0.14        11
         488       0.43      0.17      0.24        36
         489       0.18      0.09      0.12        32
         490       0.36      0.36      0.36        14
         491       0.50      0.24      0.32        25
         492       0.37      0.57      0.45        23
         493       0.00      0.00      0.00         9
         494       0.26      0.14      0.18        37
         495       0.12      0.08      0.10        24
         496       0.80      0.35      0.49        34
         497       0.20      0.12      0.15        25
         498       0.12      0.11      0.12         9
         499       0.47      0.41      0.44        17

   micro avg       0.56      0.39      0.46     55953
   macro avg       0.43      0.31      0.35     55953
weighted avg       0.54      0.39      0.45     55953
 samples avg       0.43      0.37      0.37     55953

Time taken to run this cell : 0:50:37.329590
```

4.5.6 Applying Hyperparameter tuning using GridSearch Logistic

Regression with OneVsRest Classifier

In [77]:

```
start = datetime.now()
parameters = {'estimator__alpha': [10**i for i in range(-6, 4, 1)]}
classifier = OneVsRestClassifier(SGDClassifier(loss='log', penalty='l1'
), n_jobs=-1)
g_clf = GridSearchCV(classifier, parameters, n_jobs=-1, verbose=50, sco
ring='f1_micro', cv=5)
g_clf.fit(x_train_multilabel_bow, y_train)
predictions = g_clf.predict (x_test_multilabel_bow)

print("Optimal Parameters: ", g_clf.best_params_)

print("Accuracy :",metrics.accuracy_score(y_test, predictions))
print("Hamming loss ",metrics.hamming_loss(y_test,predictions))


precision = precision_score(y_test, predictions, average='micro')
recall = recall_score(y_test, predictions, average='micro')
f1 = f1_score(y_test, predictions, average='micro')

global_report = global_report.append({
                        'Vectorizer': 'BoW',
                        'Model': 'Logistic Regression (SGD with log los
s) - Hypertuned',
                        'NGram': '(1,4)',
                        'Parameter': g_clf.best_params_['estimator__alp
ha'],
                        'Precision': precision,
                        'Recall': recall,
                        'F1_Score_Micro':f1
                     },
                    ignore_index=True)

print("Micro-average quality numbers")
print("Precision: {:.4f}, Recall: {:.4f}, F1-measure: {:.4f}".format(pr
ecision, recall, f1))

precision = precision_score(y_test, predictions, average='macro')
```

```
recall = recall_score(y_test, predictions, average='macro')
f1 = f1_score(y_test, predictions, average='macro')

print("Macro-average quality numbers")
print("Precision: {:.4f}, Recall: {:.4f}, F1-measure: {:.4f}".format(pr
ecision, recall, f1))

print (metrics.classification_report(y_test, predictions))
print("Time taken to run this cell :", datetime.now() - start)
```

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent work
ers.
[Parallel(n_jobs=-1)]: Done   1 tasks      | elapsed: 96.6min
[Parallel(n_jobs=-1)]: Done   2 tasks      | elapsed: 107.5min
[Parallel(n_jobs=-1)]: Done   3 tasks      | elapsed: 112.2min
[Parallel(n_jobs=-1)]: Done   4 tasks      | elapsed: 118.8min
[Parallel(n_jobs=-1)]: Done   5 tasks      | elapsed: 121.6min
[Parallel(n_jobs=-1)]: Done   6 tasks      | elapsed: 127.4min
[Parallel(n_jobs=-1)]: Done   7 tasks      | elapsed: 128.7min
[Parallel(n_jobs=-1)]: Done   8 tasks      | elapsed: 141.1min
[Parallel(n_jobs=-1)]: Done   9 tasks      | elapsed: 170.0min
[Parallel(n_jobs=-1)]: Done  10 tasks      | elapsed: 180.5min
[Parallel(n_jobs=-1)]: Done  11 tasks      | elapsed: 186.8min
[Parallel(n_jobs=-1)]: Done  12 tasks      | elapsed: 188.8min
[Parallel(n_jobs=-1)]: Done  13 tasks      | elapsed: 201.5min
[Parallel(n_jobs=-1)]: Done  14 tasks      | elapsed: 203.3min
[Parallel(n_jobs=-1)]: Done  15 tasks      | elapsed: 204.8min
[Parallel(n_jobs=-1)]: Done  16 tasks      | elapsed: 207.3min
[Parallel(n_jobs=-1)]: Done  17 tasks      | elapsed: 217.4min
[Parallel(n_jobs=-1)]: Done  18 tasks      | elapsed: 220.6min
[Parallel(n_jobs=-1)]: Done  19 tasks      | elapsed: 220.9min
[Parallel(n_jobs=-1)]: Done  20 tasks      | elapsed: 221.1min
[Parallel(n_jobs=-1)]: Done  21 tasks      | elapsed: 222.2min
[Parallel(n_jobs=-1)]: Done  22 tasks      | elapsed: 223.8min
[Parallel(n_jobs=-1)]: Done  23 tasks      | elapsed: 226.7min
[Parallel(n_jobs=-1)]: Done  24 tasks      | elapsed: 231.8min
[Parallel(n_jobs=-1)]: Done  25 tasks      | elapsed: 237.1min
[Parallel(n_jobs=-1)]: Done  26 tasks      | elapsed: 242.1min
[Parallel(n_jobs=-1)]: Done  27 tasks      | elapsed: 242.7min
```

```
[Parallel(n_jobs=-1)]: Done  28 tasks      | elapsed: 243.0min
[Parallel(n_jobs=-1)]: Done  29 tasks      | elapsed: 245.6min

[Parallel(n_jobs=-1)]: Done  30 tasks      | elapsed: 245.8min
[Parallel(n_jobs=-1)]: Done  31 tasks      | elapsed: 246.3min
[Parallel(n_jobs=-1)]: Done  32 tasks      | elapsed: 250.9min
[Parallel(n_jobs=-1)]: Done  33 tasks      | elapsed: 255.6min
[Parallel(n_jobs=-1)]: Done  34 tasks      | elapsed: 258.1min
[Parallel(n_jobs=-1)]: Done  35 tasks      | elapsed: 260.4min
[Parallel(n_jobs=-1)]: Done  37 out of  50 | elapsed: 260.7min remainin
g: 91.6min
[Parallel(n_jobs=-1)]: Done  39 out of  50 | elapsed: 261.4min remainin
g: 73.7min
[Parallel(n_jobs=-1)]: Done  41 out of  50 | elapsed: 270.4min remainin
g: 59.4min
[Parallel(n_jobs=-1)]: Done  43 out of  50 | elapsed: 274.8min remainin
g: 44.7min
[Parallel(n_jobs=-1)]: Done  45 out of  50 | elapsed: 275.3min remainin
g: 30.6min
[Parallel(n_jobs=-1)]: Done  47 out of  50 | elapsed: 275.6min remainin
g: 17.6min
[Parallel(n_jobs=-1)]: Done  50 out of  50 | elapsed: 278.1min finished
Optimal Parameters:  {'estimator__alpha': 0.001}
Accuracy : 0.1647
Hamming loss  0.0034135333333333334
Micro-average quality numbers
Precision: 0.5805, Recall: 0.3062, F1-measure: 0.4009
Macro-average quality numbers
Precision: 0.3810, Recall: 0.2136, F1-measure: 0.2544
              precision    recall  f1-score   support

           0       0.64      0.28      0.39      1111
           1       0.56      0.10      0.17      2052
           2       0.65      0.31      0.42      2388
           3       0.73      0.49      0.59      2226
           4       0.78      0.44      0.57      2014
           5       0.46      0.11      0.18       642
           6       0.68      0.37      0.48      1756
           7       0.83      0.59      0.69      1690
           8       0.56      0.23      0.33       341
```

```
 9        0.77      0.77      0.77      2344
10        0.61      0.32      0.42       821

11        0.53      0.18      0.27      1143
12        0.69      0.38      0.49       768
13        0.73      0.23      0.35       745
14        0.77      0.46      0.57       952
15        0.41      0.27      0.32       314
16        0.55      0.15      0.24       624
17        0.73      0.59      0.65       535
18        0.80      0.51      0.63       631
19        0.65      0.49      0.56       101
20        0.41      0.32      0.36       245
21        0.73      0.51      0.60       694
22        0.51      0.23      0.31       568
23        0.66      0.20      0.31       423
24        0.78      0.06      0.11       406
25        0.66      0.40      0.50      1373
26        0.46      0.40      0.42       253
27        0.21      0.14      0.17       357
28        0.53      0.36      0.43       222
29        0.58      0.46      0.51       273
30        0.41      0.21      0.28       308
31        0.59      0.22      0.32       256
32        0.71      0.38      0.50       295
33        0.23      0.09      0.13       263
34        0.85      0.48      0.61       256
35        0.40      0.29      0.33       280
36        0.22      0.20      0.21       290
37        0.31      0.12      0.17       200
38        0.29      0.21      0.25       109
39        0.63      0.35      0.45       209
40        0.44      0.40      0.42       113
41        0.50      0.28      0.36       197
42        0.32      0.19      0.24        52
43        0.00      0.00      0.00       179
44        0.83      0.44      0.57       431
45        0.29      0.15      0.20        47
46        0.47      0.22      0.30        37
47        0.49      0.16      0.24       155
```

```
48        0.59      0.51      0.55       254
49        0.55      0.18      0.27       201

50        0.61      0.28      0.38        61
51        0.93      0.79      0.86       246
52        0.67      0.54      0.60       146
53        0.94      0.83      0.88       516
54        0.85      0.55      0.67       170
55        0.34      0.18      0.24       234
56        0.12      0.07      0.09       357
57        0.48      0.27      0.34        78
58        0.70      0.70      0.70       102
59        0.45      0.17      0.25       122
60        0.81      0.62      0.70       138
61        0.26      0.22      0.24        36
62        0.26      0.11      0.16       172
63        0.25      0.03      0.06        60
64        0.67      0.38      0.48       106
65        0.20      0.03      0.05        34
66        0.31      0.15      0.20       101
67        0.20      0.21      0.20        38
68        0.42      0.36      0.38       104
69        0.40      0.12      0.18       144
70        0.12      0.19      0.15       135
71        0.22      0.18      0.20       190
72        0.72      0.32      0.44       139
73        0.23      0.10      0.14        69
74        0.00      0.00      0.00       133
75        0.88      0.13      0.22       181
76        0.46      0.47      0.46       113
77        0.37      0.30      0.33       158
78        0.18      0.08      0.11       142
79        0.21      0.18      0.19        96
80        0.19      0.22      0.20       101
81        0.55      0.20      0.29        56
82        0.06      0.02      0.03        62
83        0.42      0.64      0.50        77
84        0.50      0.01      0.02       100
85        0.53      0.46      0.50        54
86        0.37      0.18      0.24        79
```

```
 87       0.33      0.16      0.22        92
 88       0.63      0.23      0.34       124

 89       0.64      0.50      0.56       101
 90       0.20      0.03      0.04        40
 91       0.41      0.27      0.33        66
 92       0.39      0.28      0.32        58
 93       0.25      0.01      0.01       161
 94       0.43      0.10      0.16       130
 95       0.00      0.00      0.00        47
 96       0.75      0.64      0.69       107
 97       0.38      0.38      0.38        39
 98       0.14      0.04      0.06       111
 99       0.67      0.11      0.18        95
100       0.16      0.13      0.14       129
101       0.74      0.43      0.54        91
102       0.20      0.07      0.11        27
103       0.91      0.71      0.80        90
104       0.00      0.00      0.00       124
105       0.10      0.20      0.13        76
106       0.00      0.00      0.00       371
107       0.71      0.36      0.48       114
108       0.54      0.40      0.46        98
109       0.84      0.33      0.48        63
110       0.75      0.25      0.38        24
111       0.43      0.43      0.43        53
112       0.15      0.11      0.12        65
113       0.36      0.24      0.29        70
114       0.34      0.37      0.36        27
115       0.50      0.01      0.03        72
116       0.19      0.30      0.23        27
117       0.39      0.23      0.29        90
118       0.64      0.22      0.33        95
119       0.30      0.30      0.30        92
120       0.22      0.34      0.27        87
121       0.42      0.44      0.43        45
122       0.00      0.00      0.00       182
123       0.23      0.12      0.15        94
124       0.81      0.27      0.41        62
125       0.80      0.45      0.58        91
```

```
126        0.76       0.41       0.53        69
127        0.53       0.44       0.48        73

128        0.33       0.32       0.33        25
129        0.00       0.00       0.00        68
130        0.44       0.20       0.27       123
131        0.13       0.11       0.12        84
132        0.00       0.00       0.00        67
133        0.16       0.06       0.08       127
134        0.28       0.22       0.25        45
135        0.51       0.41       0.45        88
136        0.03       0.05       0.04        63
137        0.90       0.78       0.84        96
138        0.00       0.00       0.00        71
139        0.85       0.75       0.80        92
140        0.20       0.04       0.07        23
141        0.39       0.10       0.16        90
142        0.22       0.20       0.21        10
143        0.14       0.02       0.04        44
144        0.62       0.31       0.42        67
145        0.66       0.40       0.50       131
146        0.16       0.11       0.13        83
147        0.12       0.06       0.08        32
148        0.42       0.04       0.08       115
149        0.44       0.30       0.36        63
150        0.57       0.41       0.48        83
151        0.70       0.30       0.42       101
152        0.12       0.03       0.05        29
153        0.88       0.71       0.78       191
154        0.82       0.59       0.69        54
155        0.43       0.33       0.38        84
156        0.23       0.08       0.12        37
157        0.20       0.22       0.21        65
158        0.30       0.17       0.22        46
159        0.78       0.47       0.59        80
160        0.10       0.03       0.05        66
161        0.00       0.00       0.00        56
162        0.33       0.20       0.25       127
163        0.77       0.40       0.52       111
164        0.11       0.03       0.05        32
```

```
165         0.20        0.07        0.11          28
166         0.35        0.06        0.10          98

167         0.69        0.57        0.62          88
168         0.59        0.49        0.54          59
169         0.20        0.05        0.08          42
170         0.60        0.75        0.67           4
171         0.80        0.37        0.50          95
172         0.00        0.00        0.00          54
173         0.65        0.31        0.42          65
174         0.50        0.35        0.42          31
175         0.58        0.22        0.32          32
176         0.22        0.29        0.25          58
177         0.33        0.16        0.21          76
178         0.40        0.04        0.07          55
179         0.92        0.59        0.72          74
180         0.74        0.62        0.68          64
181         1.00        0.02        0.03          57
182         0.17        0.25        0.20          36
183         0.50        0.21        0.30          52
184         0.58        0.40        0.47          48
185         0.00        0.00        0.00          16
186         0.00        0.00        0.00          28
187         0.00        0.00        0.00          36
188         0.22        0.08        0.11          26
189         0.15        0.09        0.11          44
190         0.16        0.09        0.11          46
191         0.22        0.11        0.14          75
192         0.00        0.00        0.00          50
193         0.45        0.25        0.32          20
194         0.33        0.04        0.07          27
195         0.40        0.33        0.36           6
196         0.20        0.03        0.05          68
197         0.50        0.52        0.51          29
198         0.00        0.00        0.00         104
199         0.59        0.36        0.45          36
200         1.00        0.75        0.86           4
201         1.00        0.50        0.67           4
202         0.50        0.04        0.08          96
203         0.59        0.66        0.62          61
```

```
204        0.41      0.11      0.17       82
205        0.50      0.29      0.37       34

206        0.73      0.50      0.59       66
207        0.42      0.16      0.24       97
208        0.15      0.03      0.06       89
209        0.78      0.51      0.62       55
210        0.71      0.47      0.57       78
211        0.08      0.03      0.04       78
212        1.00      0.01      0.01      158
213        0.00      0.00      0.00       44
214        0.59      0.49      0.53       35
215        0.84      0.44      0.58       48
216        0.63      0.53      0.58       62
217        0.00      0.00      0.00       11
218        0.88      0.43      0.57       68
219        0.34      0.20      0.25       60
220        0.00      0.00      0.00       25
221        0.46      0.21      0.29       57
222        0.50      0.31      0.38       36
223        0.33      0.05      0.08       88
224        0.24      0.24      0.24       46
225        0.38      0.08      0.14       60
226        0.39      0.14      0.20       65
227        1.00      0.43      0.60        7
228        0.07      0.08      0.08       12
229        0.27      0.04      0.08       68
230        0.70      0.17      0.28       40
231        0.14      0.12      0.12       26
232        0.94      0.57      0.71       30
233        0.00      0.00      0.00       41
234        0.28      0.17      0.21       53
235        0.30      0.17      0.22       35
236        0.31      0.22      0.26       18
237        0.00      0.00      0.00       22
238        0.73      0.51      0.60       59
239        0.48      0.51      0.49       43
240        0.21      0.11      0.14       45
241        0.00      0.00      0.00       46
242        0.20      0.08      0.11       38
```

```
243         0.72        0.32        0.44          56
244         0.13        0.20        0.16          35

245         0.01        0.02        0.01          42
246         0.20        0.03        0.05          33
247         0.28        0.26        0.27          47
248         0.23        0.12        0.16          25
249         0.20        0.41        0.27          39
250         0.00        0.00        0.00          77
251         0.72        0.55        0.63          56
252         0.00        0.00        0.00          72
253         0.67        1.00        0.80           4
254         0.60        0.31        0.41          29
255         0.35        0.08        0.13         113
256         0.80        0.56        0.66          59
257         0.09        0.05        0.07          59
258         0.97        0.72        0.82          39
259         0.54        0.58        0.56          12
260         0.57        0.44        0.50           9
261         0.96        0.61        0.75          44
262         0.65        0.41        0.50          32
263         0.00        0.00        0.00         156
264         0.60        0.60        0.60           5
265         0.00        0.00        0.00         198
266         0.00        0.00        0.00          40
267         0.00        0.00        0.00          29
268         0.00        0.00        0.00          39
269         0.00        0.00        0.00           6
270         0.75        0.60        0.67           5
271         0.50        0.12        0.19          17
272         0.00        0.00        0.00          54
273         0.17        0.09        0.11          23
274         0.00        0.00        0.00         126
275         0.55        0.38        0.44          32
276         0.50        0.30        0.37          10
277         0.82        0.60        0.69          67
278         0.44        0.08        0.13          53
279         0.60        0.19        0.29          16
280         0.64        0.84        0.73          19
281         0.32        0.11        0.17          61
```

```
282         0.00        0.00        0.00          81
283         0.51        0.23        0.32          94

284         0.67        0.13        0.22          31
285         0.25        0.09        0.14          43
286         0.07        0.03        0.04          79
287         0.38        0.40        0.39          20
288         0.81        0.76        0.79          17
289         0.78        0.52        0.62          56
290         0.00        0.00        0.00          63
291         0.38        0.20        0.26          46
292         0.36        0.20        0.26          50
293         0.40        0.24        0.30          17
294         0.29        0.07        0.12          27
295         0.14        0.03        0.05          63
296         0.29        0.36        0.32          25
297         0.00        0.00        0.00          38
298         0.00        0.00        0.00          62
299         0.73        0.39        0.51          49
300         0.00        0.00        0.00          39
301         0.17        0.25        0.20           8
302         0.09        0.17        0.12          18
303         0.00        0.00        0.00          19
304         1.00        0.22        0.36          18
305         0.47        0.35        0.40          26
306         0.67        0.20        0.31          10
307         1.00        0.08        0.15          12
308         1.00        0.12        0.22           8
309         0.33        0.07        0.12          57
310         0.33        0.08        0.13          37
311         0.86        0.60        0.71          50
312         0.50        0.08        0.14          36
313         0.08        0.05        0.06          19
314         0.84        0.36        0.50          59
315         0.90        0.38        0.53          48
316         0.35        0.13        0.19          45
317         0.09        0.07        0.08          29
318         0.35        0.19        0.25          37
319         0.57        0.11        0.18          38
320         0.81        0.68        0.74          44
```

```
321        0.61       0.27       0.37         41
322        0.00       0.00       0.00         25

323        0.00       0.00       0.00          4
324        0.20       0.09       0.13         11
325        0.25       0.67       0.36          3
326        0.35       0.19       0.24         32
327        0.56       0.20       0.29         50
328        0.11       0.02       0.04         46
329        0.96       0.47       0.63         47
330        0.85       0.74       0.79         31
331        0.00       0.00       0.00         11
332        0.00       0.00       0.00         31
333        0.43       0.12       0.18         26
334        0.00       0.00       0.00         41
335        0.08       0.05       0.06         39
336        0.78       0.17       0.29         40
337        0.11       0.05       0.07         37
338        0.50       0.20       0.29         10
339        0.00       0.00       0.00         38
340        0.70       0.70       0.70         23
341        0.16       0.09       0.12         33
342        0.10       0.05       0.07         40
343        0.36       0.11       0.17         35
344        0.50       0.30       0.37         30
345        0.00       0.00       0.00         33
346        0.00       0.00       0.00         25
347        0.11       0.44       0.17          9
348        0.50       0.50       0.50          2
349        0.85       0.65       0.73         34
350        0.20       0.05       0.08         38
351        0.42       0.16       0.24         49
352        0.00       0.00       0.00         22
353        0.29       0.13       0.18         39
354        0.00       0.00       0.00         18
355        0.11       0.03       0.05         31
356        0.00       0.00       0.00         17
357        0.82       0.26       0.39         35
358        0.00       0.00       0.00         43
359        0.00       0.00       0.00         47
```

```
360       0.00      0.00      0.00        29
361       0.75      0.63      0.69        38

362       0.25      0.19      0.22        36
363       0.48      0.22      0.30        55
364       0.23      0.24      0.23        34
365       0.31      0.17      0.22        24
366       0.33      0.11      0.16        19
367       0.20      0.20      0.20         5
368       0.12      0.09      0.10        33
369       0.75      0.35      0.48        34
370       0.17      0.05      0.08        20
371       0.00      0.00      0.00        17
372       0.14      0.32      0.20        31
373       0.00      0.00      0.00        34
374       0.25      0.04      0.07        23
375       0.25      0.16      0.20        31
376       0.26      0.18      0.21        28
377       0.74      0.77      0.76        22
378       1.00      0.11      0.20         9
379       0.00      0.00      0.00        29
380       0.19      0.27      0.22        30
381       0.71      0.71      0.71        35
382       0.55      0.17      0.26        36
383       0.00      0.00      0.00         4
384       0.95      0.75      0.84        24
385       0.43      0.12      0.19        25
386       0.00      0.00      0.00        27
387       0.28      0.31      0.29        36
388       0.55      0.35      0.43        31
389       0.57      0.11      0.18        37
390       0.33      0.04      0.07        27
391       0.00      0.00      0.00        46
392       1.00      0.50      0.67         4
393       0.33      0.16      0.21        19
394       0.00      0.00      0.00        12
395       0.27      0.15      0.20        26
396       0.00      0.00      0.00        69
397       0.00      0.00      0.00        25
398       0.36      0.12      0.19        32
```

```
399      0.30      0.09      0.14       33
400      0.17      0.03      0.05       38

401      0.50      0.12      0.19       17
402      0.00      0.00      0.00       24
403      1.00      0.12      0.22       16
404      0.00      0.00      0.00       15
405      0.00      0.00      0.00       20
406      0.00      0.00      0.00       15
407      0.38      0.12      0.18       25
408      0.23      0.16      0.19       19
409      0.50      0.20      0.28       46
410      0.00      0.00      0.00       45
411      0.00      0.00      0.00       21
412      0.33      0.12      0.18        8
413      0.57      0.46      0.51       35
414      0.20      0.06      0.09       34
415      1.00      0.43      0.60       14
416      0.29      0.24      0.26       29
417      0.00      0.00      0.00       28
418      0.00      0.00      0.00       21
419      0.00      0.00      0.00       26
420      0.62      0.34      0.44       38
421      0.00      0.00      0.00      131
422      0.00      0.00      0.00       26
423      0.00      0.00      0.00       25
424      0.79      0.31      0.45       48
425      0.00      0.00      0.00       24
426      0.00      0.00      0.00       42
427      0.31      0.15      0.21       26
428      0.00      0.00      0.00       10
429      0.48      0.28      0.35       54
430      0.00      0.00      0.00       32
431      0.25      0.10      0.15       48
432      0.16      0.09      0.11       35
433      0.00      0.00      0.00       22
434      0.00      0.00      0.00       24
435      1.00      0.10      0.18       59
436      0.00      0.00      0.00       35
437      0.05      0.17      0.07       12
```

```
438       0.69      0.18      0.29      50
439       0.25      0.11      0.15      36

440       0.00      0.00      0.00      35
441       0.00      0.00      0.00       8
442       0.26      0.15      0.19      48
443       1.00      0.22      0.36      18
444       0.78      0.13      0.23      52
445       1.00      0.10      0.18      20
446       0.33      0.06      0.10      18
447       0.00      0.00      0.00       6
448       0.00      0.00      0.00      28
449       0.08      0.05      0.06      38
450       0.00      0.00      0.00     129
451       0.00      0.00      0.00      29
452       0.00      0.00      0.00      58
453       0.00      0.00      0.00      32
454       0.00      0.00      0.00      16
455       1.00      0.44      0.61      34
456       0.13      0.33      0.19       9
457       0.93      0.47      0.62      30
458       0.00      0.00      0.00      40
459       0.14      0.03      0.05      37
460       0.74      0.57      0.64      30
461       0.90      0.33      0.49      27
462       0.38      0.25      0.30      12
463       0.57      0.24      0.33      17
464       0.00      0.00      0.00      56
465       1.00      0.11      0.20       9
466       0.00      0.00      0.00      22
467       0.00      0.00      0.00       9
468       0.17      0.07      0.10      15
469       0.13      0.14      0.14      14
470       0.75      0.19      0.30      16
471       0.83      0.28      0.42      18
472       0.20      0.19      0.19      16
473       0.40      0.45      0.43      22
474       0.25      0.17      0.20      12
475       0.00      0.00      0.00     110
476       0.45      0.45      0.45      20
```

```
         477       0.75      0.10      0.17        31
         478       0.29      0.05      0.08        42
         479       0.00      0.00      0.00         4
         480       0.36      0.17      0.23        47
         481       0.00      0.00      0.00        30
         482       0.00      0.00      0.00        35
         483       0.10      0.03      0.05        30
         484       0.20      0.10      0.13        20
         485       1.00      0.33      0.50        15
         486       0.50      0.12      0.19        17
         487       0.33      0.09      0.14        11
         488       0.50      0.11      0.18        36
         489       0.14      0.06      0.09        32
         490       0.57      0.29      0.38        14
         491       0.80      0.16      0.27        25
         492       0.28      0.30      0.29        23
         493       0.00      0.00      0.00         9
         494       0.00      0.00      0.00        37
         495       0.00      0.00      0.00        24
         496       0.88      0.21      0.33        34
         497       0.00      0.00      0.00        25
         498       0.00      0.00      0.00         9
         499       0.00      0.00      0.00        17

   micro avg       0.58      0.31      0.40     55953
   macro avg       0.38      0.21      0.25     55953
weighted avg       0.54      0.31      0.37     55953
 samples avg       0.37      0.29      0.30     55953

Time taken to run this cell : 4:41:47.328133
```

4.5.7 Applying Linear SVM with OneVsRest Classifier - TFIDF Vectorizer

In [78]:

```
start = datetime.now()
classifier = OneVsRestClassifier(SGDClassifier(loss='hinge', alpha=0.00001, penalty='l1'), n_jobs=-1)
```

```
classifier.fit(x_train_multilabel, y_train)
predictions = classifier.predict (x_test_multilabel)


print("Accuracy :",metrics.accuracy_score(y_test, predictions))
print("Hamming loss ",metrics.hamming_loss(y_test,predictions))


precision = precision_score(y_test, predictions, average='micro')
recall = recall_score(y_test, predictions, average='micro')
f1 = f1_score(y_test, predictions, average='micro')

global_report = global_report.append({
                        'Vectorizer': 'Tf-IDF',
                        'Model': 'Linear SVM (SGD with hinge loss)',
                        'NGram': '(1,3)',
                        'Parameter': 0.00001,
                        'Precision': precision,
                        'Recall': recall,
                        'F1_Score_Micro':f1
                     },
                    ignore_index=True)

print("Micro-average quality numbers")
print("Precision: {:.4f}, Recall: {:.4f}, F1-measure: {:.4f}".format(pr
ecision, recall, f1))

precision = precision_score(y_test, predictions, average='macro')
recall = recall_score(y_test, predictions, average='macro')
f1 = f1_score(y_test, predictions, average='macro')

print("Macro-average quality numbers")
print("Precision: {:.4f}, Recall: {:.4f}, F1-measure: {:.4f}".format(pr
ecision, recall, f1))

print (metrics.classification_report(y_test, predictions))
print("Time taken to run this cell :", datetime.now() - start)
```

```
Accuracy : 0.21893333333333334
Hamming loss  0.0029227333333333334
```

```
Micro-average quality numbers
Precision: 0.7923, Recall: 0.2934, F1-measure: 0.4282
Macro-average quality numbers
Precision: 0.4256, Recall: 0.2180, F1-measure: 0.2638
              precision    recall  f1-score   support

           0       0.80      0.35      0.49      1111
           1       0.73      0.14      0.24      2052
           2       0.78      0.28      0.42      2388
           3       0.82      0.45      0.58      2226
           4       0.84      0.42      0.56      2014
           5       0.61      0.14      0.23       642
           6       0.81      0.35      0.49      1756
           7       0.93      0.71      0.80      1690
           8       0.71      0.20      0.32       341
           9       0.77      0.87      0.82      2344
          10       0.70      0.32      0.44       821
          11       0.73      0.01      0.03      1143
          12       0.85      0.33      0.48       768
          13       0.76      0.21      0.33       745
          14       0.83      0.51      0.63       952
          15       0.74      0.21      0.32       314
          16       0.68      0.16      0.26       624
          17       0.80      0.60      0.68       535
          18       0.85      0.51      0.63       631
          19       0.88      0.56      0.69       101
          20       0.72      0.24      0.36       245
          21       0.86      0.47      0.61       694
          22       0.76      0.06      0.10       568
          23       0.75      0.23      0.35       423
          24       0.82      0.14      0.24       406
          25       1.00      0.00      0.00      1373
          26       0.58      0.30      0.40       253
          27       0.00      0.00      0.00       357
          28       0.76      0.31      0.44       222
          29       0.81      0.25      0.38       273
          30       0.65      0.13      0.21       308
          31       0.67      0.27      0.38       256

          32       0.75      0.44      0.56       295
```

```
33      0.00     0.00     0.00      263
34      0.86     0.54     0.66      256
35      0.52     0.05     0.10      280
36      0.20     0.00     0.01      290
37      1.00     0.01     0.01      200
38      1.00     0.01     0.02      109
39      0.66     0.41     0.50      209
40      0.58     0.41     0.48      113
41      0.82     0.21     0.33      197
42      0.69     0.46     0.55       52
43      0.00     0.00     0.00      179
44      0.84     0.52     0.64      431
45      0.38     0.11     0.17       47
46      0.82     0.38     0.52       37
47      0.80     0.13     0.22      155
48      0.71     0.50     0.59      254
49      0.72     0.15     0.25      201
50      0.88     0.23     0.36       61
51      0.93     0.78     0.85      246
52      0.69     0.69     0.69      146
53      0.94     0.91     0.93      516
54      0.85     0.59     0.70      170
55      0.00     0.00     0.00      234
56      0.00     0.00     0.00      357
57      0.00     0.00     0.00       78
58      0.92     0.69     0.79      102
59      0.00     0.00     0.00      122
60      0.88     0.57     0.69      138
61      0.67     0.06     0.10       36
62      0.00     0.00     0.00      172
63      0.00     0.00     0.00       60
64      0.66     0.46     0.54      106
65      1.00     0.06     0.11       34
66      1.00     0.01     0.02      101
67      0.60     0.32     0.41       38
68      0.66     0.34     0.45      104
69      0.00     0.00     0.00      144
70      0.00     0.00     0.00      135

71      0.00     0.00     0.00      190
```

```
 72     0.82    0.30    0.44    139
 73     0.00    0.00    0.00     69
 74     0.00    0.00    0.00    133
 75     0.84    0.51    0.63    181
 76     0.59    0.35    0.44    113
 77     0.70    0.22    0.34    158
 78     0.00    0.00    0.00    142
 79     0.00    0.00    0.00     96
 80     0.00    0.00    0.00    101
 81     0.69    0.20    0.31     56
 82     0.00    0.00    0.00     62
 83     0.64    0.55    0.59     77
 84     0.92    0.11    0.20    100
 85     0.53    0.48    0.50     54
 86     0.00    0.00    0.00     79
 87     0.67    0.02    0.04     92
 88     0.71    0.24    0.36    124
 89     0.77    0.46    0.57    101
 90     0.56    0.12    0.20     40
 91     0.74    0.26    0.38     66
 92     0.50    0.03    0.06     58
 93     0.96    0.27    0.43    161
 94     0.83    0.15    0.25    130
 95     1.00    0.02    0.04     47
 96     0.74    0.55    0.63    107
 97     0.79    0.28    0.42     39
 98     0.00    0.00    0.00    111
 99     0.71    0.16    0.26     95
100     0.00    0.00    0.00    129
101     0.94    0.37    0.54     91
102     0.35    0.22    0.27     27
103     0.94    0.81    0.87     90
104     0.00    0.00    0.00    124
105     0.00    0.00    0.00     76
106     0.00    0.00    0.00    371
107     0.71    0.36    0.48    114
108     0.61    0.32    0.42     98
109     0.80    0.38    0.52     63

110     0.80    0.33    0.47     24
```

```
111        0.75       0.17       0.28         53
112        0.00       0.00       0.00         65
113        0.50       0.07       0.12         70
114        0.90       0.33       0.49         27
115        0.00       0.00       0.00         72
116        0.55       0.22       0.32         27
117        0.00       0.00       0.00         90
118        0.69       0.37       0.48         95
119        1.00       0.01       0.02         92
120        0.00       0.00       0.00         87
121        0.56       0.20       0.30         45
122        0.00       0.00       0.00        182
123        0.00       0.00       0.00         94
124        0.82       0.45       0.58         62
125        0.84       0.46       0.60         91
126        0.73       0.51       0.60         69
127        0.64       0.41       0.50         73
128        1.00       0.40       0.57         25
129        0.00       0.00       0.00         68
130        0.00       0.00       0.00        123
131        0.00       0.00       0.00         84
132        0.00       0.00       0.00         67
133        0.00       0.00       0.00        127
134        0.78       0.47       0.58         45
135        0.50       0.40       0.44         88
136        0.00       0.00       0.00         63
137        0.92       0.79       0.85         96
138        0.70       0.10       0.17         71
139        0.84       0.71       0.77         92
140        0.00       0.00       0.00         23
141        0.67       0.07       0.12         90
142        0.50       0.20       0.29         10
143        0.00       0.00       0.00         44
144        0.61       0.28       0.39         67
145        0.81       0.32       0.46        131
146        0.00       0.00       0.00         83
147        0.00       0.00       0.00         32
148        0.00       0.00       0.00        115

149        0.43       0.05       0.09         63
```

```
150        0.74       0.34       0.46         83
151        0.84       0.38       0.52        101
152        0.00       0.00       0.00         29
153        0.91       0.94       0.93        191
154        0.91       0.59       0.72         54
155        0.00       0.00       0.00         84
156        0.62       0.27       0.38         37
157        0.50       0.03       0.06         65
158        0.62       0.11       0.19         46
159        0.84       0.54       0.66         80
160        0.00       0.00       0.00         66
161        0.00       0.00       0.00         56
162        0.00       0.00       0.00        127
163        0.69       0.82       0.75        111
164        0.00       0.00       0.00         32
165        1.00       0.04       0.07         28
166        0.00       0.00       0.00         98
167        0.77       0.65       0.70         88
168        0.85       0.47       0.61         59
169        0.00       0.00       0.00         42
170        0.67       0.50       0.57          4
171        0.75       0.47       0.58         95
172        0.00       0.00       0.00         54
173        0.77       0.57       0.65         65
174        0.63       0.39       0.48         31
175        1.00       0.09       0.17         32
176        0.78       0.12       0.21         58
177        0.90       0.12       0.21         76
178        0.00       0.00       0.00         55
179        0.84       0.80       0.82         74
180        0.89       0.61       0.72         64
181        0.00       0.00       0.00         57
182        0.00       0.00       0.00         36
183        0.67       0.23       0.34         52
184        0.83       0.31       0.45         48
185        0.33       0.06       0.11         16
186        0.00       0.00       0.00         28
187        0.40       0.06       0.10         36

188        0.75       0.12       0.20         26
```

```
189        0.00       0.00       0.00         44
190        0.80       0.09       0.16         46
191        0.00       0.00       0.00         75
192        0.56       0.10       0.17         50
193        0.80       0.40       0.53         20
194        0.00       0.00       0.00         27
195        0.57       0.67       0.62          6
196        0.67       0.03       0.06         68
197        0.50       0.55       0.52         29
198        0.84       0.15       0.26        104
199        0.67       0.33       0.44         36
200        1.00       0.75       0.86          4
201        1.00       0.75       0.86          4
202        0.00       0.00       0.00         96
203        0.86       0.62       0.72         61
204        0.00       0.00       0.00         82
205        0.00       0.00       0.00         34
206        0.76       0.42       0.54         66
207        0.00       0.00       0.00         97
208        0.00       0.00       0.00         89
209        0.78       0.65       0.71         55
210        0.79       0.49       0.60         78
211        0.00       0.00       0.00         78
212        0.88       0.39       0.54        158
213        0.50       0.09       0.15         44
214        0.66       0.66       0.66         35
215        0.76       0.79       0.78         48
216        0.71       0.74       0.72         62
217        0.00       0.00       0.00         11
218        0.97       0.49       0.65         68
219        0.00       0.00       0.00         60
220        0.00       0.00       0.00         25
221        1.00       0.02       0.03         57
222        0.90       0.50       0.64         36
223        0.00       0.00       0.00         88
224        0.00       0.00       0.00         46
225        0.00       0.00       0.00         60
226        0.00       0.00       0.00         65

227        0.50       0.43       0.46          7
```

```
         228       0.50      0.25      0.33        12
         229       0.00      0.00      0.00        68
         230       1.00      0.10      0.18        40
         231       0.00      0.00      0.00        26
         232       0.81      0.57      0.67        30
         233       0.80      0.10      0.17        41
         234       0.00      0.00      0.00        53
         235       0.64      0.46      0.53        35
         236       0.57      0.22      0.32        18
         237       0.00      0.00      0.00        22
         238       0.73      0.59      0.65        59
         239       0.76      0.51      0.61        43
         240       0.00      0.00      0.00        45
         241       0.50      0.02      0.04        46
         242       0.00      0.00      0.00        38
         243       0.84      0.38      0.52        56
         244       0.00      0.00      0.00        35
         245       0.00      0.00      0.00        42
         246       0.00      0.00      0.00        33
         247       0.00      0.00      0.00        47
         248       0.75      0.24      0.36        25
         249       0.70      0.49      0.58        39
         250       0.00      0.00      0.00        77
         251       0.79      0.61      0.69        56
         252       0.00      0.00      0.00        72
         253       1.00      1.00      1.00         4
         254       0.69      0.38      0.49        29
         255       0.00      0.00      0.00       113
         256       0.71      0.69      0.70        59
         257       0.00      0.00      0.00        59
         258       0.93      0.69      0.79        39
         259       0.75      0.75      0.75        12
         260       0.60      0.67      0.63         9
         261       0.92      0.75      0.83        44
         262       0.79      0.59      0.68        32
         263       0.00      0.00      0.00       156
         264       1.00      0.40      0.57         5
         265       0.00      0.00      0.00       198

         266       0.00      0.00      0.00        40
         267       0.00      0.00      0.00        29
         268       0.00      0.00      0.00        39
         269       0.00      0.00      0.00         6
         270       0.75      0.60      0.67         5
         271       0.00      0.00      0.00        17
         272       0.33      0.04      0.07        54
         273       0.00      0.00      0.00        23
         274       0.00      0.00      0.00       126
         275       0.79      0.34      0.48        32
         276       0.60      0.30      0.40        10
         277       0.83      0.58      0.68        67
         278       0.00      0.00      0.00        53
         279       0.00      0.00      0.00        16
         280       0.75      0.63      0.69        19
         281       0.00      0.00      0.00        61
         282       1.00      0.01      0.02        81
         283       0.50      0.01      0.02        94
         284       0.75      0.19      0.31        31
         285       0.00      0.00      0.00        43
         286       0.00      0.00      0.00        79
         287       0.00      0.00      0.00        20
         288       0.79      0.88      0.83        17
         289       0.82      0.59      0.69        56
         290       0.00      0.00      0.00        63
         291       0.00      0.00      0.00        46
         292       0.00      0.00      0.00        50
         293       0.67      0.24      0.35        17
         294       0.00      0.00      0.00        27
         295       0.00      0.00      0.00        63
         296       0.00      0.00      0.00        25
         297       0.00      0.00      0.00        38
         298       0.25      0.02      0.03        62
         299       0.72      0.59      0.65        49
         300       0.00      0.00      0.00        39
         301       0.00      0.00      0.00         8
         302       0.00      0.00      0.00        18
         303       0.50      0.16      0.24        19
         304       0.85      0.61      0.71        18

         305       0.48      0.46      0.47        26
```

306	0.00	0.00	0.00	10
307	0.33	0.08	0.13	12
308	1.00	0.38	0.55	8
309	0.40	0.04	0.06	57
310	0.00	0.00	0.00	37
311	0.80	0.74	0.77	50
312	0.00	0.00	0.00	36
313	0.00	0.00	0.00	19
314	0.86	0.73	0.79	59
315	0.79	0.65	0.71	48
316	0.00	0.00	0.00	45
317	0.00	0.00	0.00	29
318	0.67	0.05	0.10	37
319	0.00	0.00	0.00	38
320	0.85	0.75	0.80	44
321	0.81	0.32	0.46	41
322	0.00	0.00	0.00	25
323	0.00	0.00	0.00	4
324	0.00	0.00	0.00	11
325	0.75	1.00	0.86	3
326	0.00	0.00	0.00	32
327	0.50	0.08	0.14	50
328	0.00	0.00	0.00	46
329	0.95	0.74	0.83	47
330	0.95	0.61	0.75	31
331	0.00	0.00	0.00	11
332	0.00	0.00	0.00	31
333	0.00	0.00	0.00	26
334	0.00	0.00	0.00	41
335	0.00	0.00	0.00	39
336	0.80	0.40	0.53	40
337	0.00	0.00	0.00	37
338	0.75	0.60	0.67	10
339	0.00	0.00	0.00	38
340	0.86	0.78	0.82	23
341	0.00	0.00	0.00	33
342	0.00	0.00	0.00	40
343	0.75	0.09	0.15	35
344	0.00	0.00	0.00	30

```
         345       0.00      0.00      0.00        33
         346       0.86      0.24      0.38        25
         347       0.57      0.44      0.50         9
         348       0.50      0.50      0.50         2
         349       0.83      0.71      0.76        34
         350       0.00      0.00      0.00        38
         351       0.00      0.00      0.00        49
         352       0.00      0.00      0.00        22
         353       0.00      0.00      0.00        39
         354       0.00      0.00      0.00        18
         355       0.00      0.00      0.00        31
         356       0.00      0.00      0.00        17
         357       0.78      0.71      0.75        35
         358       0.00      0.00      0.00        43
         359       0.51      0.53      0.52        47
         360       0.00      0.00      0.00        29
         361       0.69      0.63      0.66        38
         362       0.57      0.11      0.19        36
         363       0.31      0.07      0.12        55
         364       0.00      0.00      0.00        34
         365       0.00      0.00      0.00        24
         366       0.00      0.00      0.00        19
         367       0.43      0.60      0.50         5
         368       0.00      0.00      0.00        33
         369       0.76      0.65      0.70        34
         370       0.00      0.00      0.00        20
         371       0.00      0.00      0.00        17
         372       0.00      0.00      0.00        31
         373       1.00      0.03      0.06        34
         374       0.00      0.00      0.00        23
         375       0.00      0.00      0.00        31
         376       0.00      0.00      0.00        28
         377       0.80      0.73      0.76        22
         378       0.00      0.00      0.00         9
         379       0.00      0.00      0.00        29
         380       0.00      0.00      0.00        30
         381       0.83      0.71      0.77        35
         382       0.67      0.17      0.27        36

         383       0.00      0.00      0.00         4
         384       0.95      0.75      0.84        24
         385       0.62      0.32      0.42        25
         386       0.00      0.00      0.00        27
         387       0.59      0.28      0.38        36
         388       0.72      0.42      0.53        31
         389       0.73      0.43      0.54        37
         390       0.61      0.41      0.49        27
         391       0.56      0.39      0.46        46
         392       0.50      0.50      0.50         4
         393       0.00      0.00      0.00        19
         394       0.00      0.00      0.00        12
         395       0.75      0.12      0.20        26
         396       0.00      0.00      0.00        69
         397       0.79      0.44      0.56        25
         398       0.00      0.00      0.00        32
         399       0.00      0.00      0.00        33
         400       0.00      0.00      0.00        38
         401       0.71      0.29      0.42        17
         402       0.00      0.00      0.00        24
         403       0.69      0.56      0.62        16
         404       1.00      0.33      0.50        15
         405       0.00      0.00      0.00        20
         406       0.00      0.00      0.00        15
         407       0.00      0.00      0.00        25
         408       1.00      0.05      0.10        19
         409       0.67      0.17      0.28        46
         410       0.00      0.00      0.00        45
         411       0.00      0.00      0.00        21
         412       0.00      0.00      0.00         8
         413       0.60      0.09      0.15        35
         414       0.00      0.00      0.00        34
         415       0.89      0.57      0.70        14
         416       0.00      0.00      0.00        29
         417       0.00      0.00      0.00        28
         418       0.00      0.00      0.00        21
         419       0.67      0.08      0.14        26
         420       0.00      0.00      0.00        38
         421       0.00      0.00      0.00       131

         422       0.00      0.00      0.00        26
```

```
423        0.79        0.44        0.56         25
424        0.77        0.50        0.61         48
425        0.50        0.08        0.14         24
426        0.20        0.02        0.04         42
427        0.00        0.00        0.00         26
428        0.00        0.00        0.00         10
429        0.75        0.06        0.10         54
430        0.00        0.00        0.00         32
431        0.00        0.00        0.00         48
432        0.00        0.00        0.00         35
433        0.00        0.00        0.00         22
434        1.00        0.04        0.08         24
435        0.90        0.63        0.74         59
436        0.00        0.00        0.00         35
437        0.00        0.00        0.00         12
438        0.61        0.40        0.48         50
439        0.00        0.00        0.00         36
440        0.00        0.00        0.00         35
441        1.00        0.25        0.40          8
442        0.00        0.00        0.00         48
443        0.90        0.50        0.64         18
444        0.61        0.27        0.37         52
445        0.00        0.00        0.00         20
446        0.00        0.00        0.00         18
447        0.00        0.00        0.00          6
448        0.00        0.00        0.00         28
449        0.00        0.00        0.00         38
450        0.98        0.93        0.96        129
451        0.81        0.72        0.76         29
452        0.00        0.00        0.00         58
453        0.00        0.00        0.00         32
454        0.86        0.38        0.52         16
455        0.88        0.65        0.75         34
456        0.00        0.00        0.00          9
457        0.90        0.87        0.88         30
458        0.70        0.17        0.28         40
459        0.50        0.05        0.10         37
460        0.91        0.70        0.79         30

461        1.00        0.26        0.41         27
```

```
462     0.38     0.25     0.30       12
463     0.53     0.47     0.50       17
464     0.85     0.52     0.64       56
465     0.60     0.67     0.63        9
466     0.00     0.00     0.00       22
467     0.50     0.22     0.31        9
468     0.00     0.00     0.00       15
469     0.00     0.00     0.00       14
470     0.78     0.44     0.56       16
471     0.91     0.56     0.69       18
472     0.33     0.19     0.24       16
473     0.62     0.59     0.60       22
474     0.18     0.17     0.17       12
475     1.00     0.96     0.98      110
476     0.67     0.20     0.31       20
477     0.71     0.65     0.68       31
478     0.75     0.14     0.24       42
479     0.00     0.00     0.00        4
480     0.00     0.00     0.00       47
481     1.00     0.40     0.57       30
482     0.00     0.00     0.00       35
483     0.00     0.00     0.00       30
484     0.29     0.10     0.15       20
485     1.00     0.47     0.64       15
486     0.00     0.00     0.00       17
487     0.00     0.00     0.00       11
488     0.50     0.11     0.18       36
489     0.00     0.00     0.00       32
490     0.75     0.43     0.55       14
491     0.90     0.36     0.51       25
492     0.45     0.57     0.50       23
493     0.00     0.00     0.00        9
494     0.00     0.00     0.00       37
495     0.00     0.00     0.00       24
496     0.82     0.26     0.40       34
497     0.00     0.00     0.00       25
498     0.00     0.00     0.00        9
499     0.00     0.00     0.00       17
```

```
   micro avg       0.79      0.29      0.43     55953
   macro avg       0.43      0.22      0.26     55953
weighted avg       0.62      0.29      0.36     55953
 samples avg       0.42      0.28      0.32     55953

Time taken to run this cell : 0:01:41.415308
```

4.5.8 Applying Linear SVM with OneVsRest Classifier BOW Vectorizer

In [80]:

```
start = datetime.now()
classifier = OneVsRestClassifier(SGDClassifier(loss='hinge', alpha=0.00001, penalty='l1'), n_jobs=-1)
classifier.fit(x_train_multilabel_bow, y_train)
predictions = classifier.predict (x_test_multilabel_bow)


print("Accuracy :",metrics.accuracy_score(y_test, predictions))
print("Hamming loss ",metrics.hamming_loss(y_test,predictions))


precision = precision_score(y_test, predictions, average='micro')
recall = recall_score(y_test, predictions, average='micro')
f1 = f1_score(y_test, predictions, average='micro')

global_report = global_report.append({
                        'Vectorizer': 'BoW',
                        'Model': 'Linear SVM (SGD with hinge loss)',
                        'NGram': '(1,4)',
                        'Parameter': 0.00001,
                        'Precision': precision,
                        'Recall': recall,
                        'F1_Score_Micro':f1
                     },
                    ignore_index=True)

print("Micro-average quality numbers")
print("Precision: {:.4f}, Recall: {:.4f}, F1-measure: {:.4f}".format(precision, recall, f1))
```

```
precision = precision_score(y_test, predictions, average='macro')
recall = recall_score(y_test, predictions, average='macro')
f1 = f1_score(y_test, predictions, average='macro')

print("Macro-average quality numbers")
print("Precision: {:.4f}, Recall: {:.4f}, F1-measure: {:.4f}".format(pr
ecision, recall, f1))

print (metrics.classification_report(y_test, predictions))
print("Time taken to run this cell :", datetime.now() - start)
```

```
Accuracy : 0.07953333333333333
Hamming loss  0.007597533333333333
Micro-average quality numbers
Precision: 0.2304, Recall: 0.4430, F1-measure: 0.3032
Macro-average quality numbers
Precision: 0.1532, Recall: 0.3646, F1-measure: 0.2044
              precision    recall  f1-score   support

           0       0.33      0.48      0.39      1111
           1       0.32      0.38      0.35      2052
           2       0.43      0.49      0.46      2388
           3       0.52      0.59      0.55      2226
           4       0.49      0.55      0.52      2014
           5       0.12      0.19      0.15       642
           6       0.44      0.47      0.45      1756
           7       0.71      0.79      0.74      1690
           8       0.16      0.30      0.21       341
           9       0.68      0.74      0.71      2344
          10       0.40      0.45      0.42       821
          11       0.26      0.35      0.30      1143
          12       0.37      0.48      0.42       768
          13       0.32      0.41      0.36       745
          14       0.52      0.60      0.56       952
          15       0.24      0.39      0.30       314
          16       0.29      0.36      0.32       624
          17       0.36      0.59      0.45       535
          18       0.42      0.58      0.49       631
          19       0.25      0.71      0.37       101
```

```
19        0.25      0.71      0.57       101
20        0.18      0.35      0.24       245
21        0.46      0.58      0.52       694
22        0.29      0.40      0.33       568
23        0.30      0.37      0.33       423
24        0.30      0.41      0.35       406
25        0.54      0.52      0.53      1373
26        0.22      0.48      0.30       253
27        0.14      0.25      0.18       357
28        0.13      0.33      0.19       222
29        0.23      0.39      0.29       273
30        0.22      0.40      0.29       308
31        0.20      0.36      0.26       256
32        0.29      0.45      0.35       295
33        0.09      0.20      0.12       263
34        0.43      0.58      0.49       256
35        0.25      0.42      0.31       280
36        0.26      0.42      0.32       290
37        0.15      0.25      0.18       200
38        0.12      0.36      0.19       109
39        0.21      0.39      0.27       209
40        0.18      0.39      0.24       113
41        0.22      0.38      0.28       197
42        0.10      0.60      0.17        52
43        0.07      0.16      0.09       179
44        0.44      0.57      0.49       431
45        0.04      0.19      0.07        47
46        0.11      0.51      0.18        37
47        0.15      0.32      0.21       155
48        0.33      0.51      0.40       254
49        0.23      0.40      0.29       201
50        0.11      0.41      0.17        61
51        0.56      0.80      0.66       246
52        0.32      0.66      0.43       146
53        0.64      0.91      0.75       516
54        0.32      0.58      0.41       170
55        0.11      0.23      0.15       234
56        0.13      0.21      0.16       357
57        0.11      0.40      0.18        78

58        0.33      0.65      0.44       102
```

```
58      0.33    0.65    0.44    102
59      0.10    0.35    0.16    122
60      0.46    0.59    0.52    138
61      0.05    0.28    0.08     36
62      0.12    0.27    0.16    172
63      0.03    0.08    0.04     60
64      0.22    0.50    0.31    106
65      0.17    0.35    0.23     34
66      0.09    0.26    0.13    101
67      0.05    0.24    0.09     38
68      0.13    0.45    0.20    104
69      0.09    0.19    0.12    144
70      0.13    0.26    0.18    135
71      0.13    0.21    0.16    190
72      0.25    0.47    0.32    139
73      0.03    0.13    0.05     69
74      0.06    0.17    0.08    133
75      0.29    0.51    0.37    181
76      0.19    0.50    0.28    113
77      0.13    0.33    0.19    158
78      0.16    0.30    0.21    142
79      0.10    0.24    0.14     96
80      0.20    0.38    0.26    101
81      0.10    0.29    0.15     56
82      0.04    0.16    0.07     62
83      0.17    0.48    0.25     77
84      0.10    0.23    0.14    100
85      0.16    0.54    0.25     54
86      0.07    0.22    0.11     79
87      0.22    0.43    0.30     92
88      0.17    0.40    0.24    124
89      0.22    0.44    0.29    101
90      0.03    0.17    0.06     40
91      0.27    0.50    0.35     66
92      0.16    0.48    0.24     58
93      0.31    0.53    0.39    161
94      0.12    0.31    0.18    130
95      0.13    0.28    0.18     47
96      0.24    0.53    0.33    107

97      0.09    0.33    0.14     39
```

```
 97      0.09    0.33    0.14     59
 98      0.07    0.20    0.11    111
 99      0.10    0.29    0.15     95
100      0.13    0.24    0.17    129
101      0.23    0.51    0.32     91
102      0.03    0.19    0.06     27
103      0.42    0.90    0.57     90
104      0.07    0.18    0.10    124
105      0.05    0.12    0.07     76
106      0.25    0.29    0.27    371
107      0.23    0.40    0.29    114
108      0.17    0.35    0.23     98
109      0.14    0.33    0.20     63
110      0.12    0.54    0.20     24
111      0.12    0.45    0.20     53
112      0.10    0.25    0.15     65
113      0.14    0.37    0.21     70
114      0.17    0.56    0.26     27
115      0.08    0.24    0.12     72
116      0.18    0.44    0.25     27
117      0.13    0.36    0.19     90
118      0.16    0.38    0.23     95
119      0.16    0.35    0.22     92
120      0.11    0.36    0.17     87
121      0.11    0.47    0.18     45
122      0.09    0.13    0.10    182
123      0.12    0.24    0.16     94
124      0.21    0.52    0.29     62
125      0.29    0.51    0.37     91
126      0.20    0.48    0.28     69
127      0.20    0.41    0.27     73
128      0.17    0.56    0.26     25
129      0.02    0.06    0.03     68
130      0.09    0.27    0.13    123
131      0.10    0.27    0.14     84
132      0.04    0.10    0.06     67
133      0.09    0.20    0.13    127
134      0.19    0.40    0.25     45
135      0.26    0.48    0.34     88

136      0.03    0.13    0.05     63
```

```
136         0.05        0.15        0.05         65
137         0.62        0.80        0.70         96
138         0.07        0.35        0.11         71
139         0.31        0.65        0.42         92
140         0.02        0.09        0.03         23
141         0.14        0.28        0.19         90
142         0.03        0.40        0.05         10
143         0.07        0.18        0.10         44
144         0.13        0.43        0.21         67
145         0.37        0.45        0.40        131
146         0.09        0.16        0.11         83
147         0.07        0.22        0.10         32
148         0.15        0.23        0.18        115
149         0.14        0.46        0.22         63
150         0.19        0.45        0.27         83
151         0.34        0.48        0.40        101
152         0.07        0.31        0.12         29
153         0.74        0.82        0.78        191
154         0.39        0.65        0.49         54
155         0.18        0.46        0.26         84
156         0.15        0.46        0.23         37
157         0.14        0.37        0.20         65
158         0.07        0.22        0.11         46
159         0.23        0.53        0.32         80
160         0.04        0.12        0.05         66
161         0.03        0.07        0.05         56
162         0.15        0.24        0.18        127
163         0.45        0.68        0.54        111
164         0.05        0.22        0.08         32
165         0.04        0.25        0.06         28
166         0.03        0.10        0.05         98
167         0.28        0.60        0.38         88
168         0.26        0.58        0.36         59
169         0.04        0.17        0.06         42
170         0.03        0.50        0.05          4
171         0.22        0.38        0.28         95
172         0.04        0.15        0.06         54
173         0.24        0.55        0.33         65
174         0.17        0.55        0.25         31

175         0.12        0.47        0.19         32
```

```
175        0.12       0.47       0.19         52
176        0.14       0.34       0.20         58
177        0.11       0.21       0.15         76
178        0.03       0.15       0.05         55
179        0.48       0.84       0.61         74
180        0.27       0.62       0.37         64
181        0.09       0.19       0.13         57
182        0.06       0.33       0.10         36
183        0.07       0.46       0.13         52
184        0.18       0.44       0.25         48
185        0.19       0.56       0.28         16
186        0.04       0.18       0.06         28
187        0.12       0.31       0.17         36
188        0.14       0.38       0.20         26
189        0.07       0.32       0.12         44
190        0.16       0.35       0.22         46
191        0.10       0.19       0.13         75
192        0.08       0.26       0.12         50
193        0.12       0.45       0.20         20
194        0.02       0.11       0.03         27
195        0.03       0.33       0.06          6
196        0.15       0.25       0.19         68
197        0.12       0.48       0.19         29
198        0.14       0.29       0.19        104
199        0.15       0.42       0.22         36
200        0.07       1.00       0.14          4
201        0.10       0.75       0.17          4
202        0.06       0.18       0.09         96
203        0.35       0.59       0.44         61
204        0.07       0.29       0.11         82
205        0.15       0.50       0.23         34
206        0.25       0.53       0.34         66
207        0.15       0.37       0.21         97
208        0.05       0.11       0.07         89
209        0.37       0.60       0.46         55
210        0.34       0.64       0.44         78
211        0.08       0.23       0.12         78
212        0.35       0.51       0.42        158
213        0.05       0.16       0.07         44

214        0.18       0.46       0.26         35
```

```
214     0.18     0.46     0.26     55
215     0.41     0.75     0.53     48
216     0.35     0.65     0.45     62
217     0.02     0.18     0.03     11
218     0.39     0.57     0.47     68
219     0.15     0.33     0.21     60
220     0.03     0.16     0.05     25
221     0.15     0.30     0.20     57
222     0.12     0.44     0.19     36
223     0.06     0.17     0.09     88
224     0.04     0.22     0.07     46
225     0.06     0.22     0.10     60
226     0.11     0.26     0.16     65
227     0.03     0.43     0.06      7
228     0.12     0.58     0.19     12
229     0.03     0.15     0.05     68
230     0.10     0.30     0.15     40
231     0.07     0.23     0.11     26
232     0.20     0.63     0.30     30
233     0.06     0.20     0.09     41
234     0.06     0.26     0.10     53
235     0.15     0.51     0.24     35
236     0.06     0.22     0.09     18
237     0.02     0.18     0.04     22
238     0.29     0.68     0.40     59
239     0.22     0.53     0.31     43
240     0.13     0.31     0.18     45
241     0.04     0.20     0.07     46
242     0.05     0.18     0.07     38
243     0.21     0.50     0.29     56
244     0.07     0.23     0.11     35
245     0.03     0.14     0.05     42
246     0.01     0.06     0.02     33
247     0.11     0.30     0.16     47
248     0.06     0.28     0.10     25
249     0.17     0.54     0.26     39
250     0.30     0.47     0.37     77
251     0.34     0.59     0.43     56
252     0.08     0.18     0.11     72

253     0.12     0.75     0.21      4
```

```
253          0.12         0.75         0.21          4
254          0.11         0.45         0.18         29
255          0.13         0.20         0.16        113
256          0.37         0.69         0.48         59
257          0.06         0.14         0.08         59
258          0.31         0.64         0.42         39
259          0.14         0.83         0.23         12
260          0.10         0.67         0.17          9
261          0.31         0.66         0.42         44
262          0.18         0.56         0.27         32
263          0.07         0.18         0.10        156
264          0.06         0.40         0.11          5
265          0.16         0.38         0.23        198
266          0.02         0.07         0.03         40
267          0.04         0.28         0.08         29
268          0.00         0.00         0.00         39
269          0.02         0.33         0.05          6
270          0.11         0.60         0.18          5
271          0.03         0.29         0.06         17
272          0.09         0.22         0.12         54
273          0.06         0.35         0.10         23
274          0.03         0.06         0.04        126
275          0.12         0.47         0.20         32
276          0.07         0.50         0.12         10
277          0.31         0.58         0.40         67
278          0.16         0.40         0.23         53
279          0.01         0.06         0.02         16
280          0.35         0.79         0.48         19
281          0.09         0.18         0.12         61
282          0.09         0.19         0.12         81
283          0.17         0.39         0.24         94
284          0.11         0.39         0.17         31
285          0.08         0.30         0.12         43
286          0.05         0.20         0.09         79
287          0.11         0.40         0.18         20
288          0.33         0.76         0.46         17
289          0.35         0.62         0.45         56
290          0.03         0.11         0.05         63
291          0.08         0.28         0.12         46

292          0.08         0.36         0.14         50
```

```
292          0.08       0.30       0.14       30
293          0.06       0.35       0.10       17
294          0.06       0.22       0.09       27
295          0.03       0.06       0.04       63
296          0.05       0.28       0.08       25
297          0.01       0.08       0.02       38
298          0.05       0.13       0.07       62
299          0.31       0.61       0.41       49
300          0.04       0.15       0.06       39
301          0.01       0.12       0.01        8
302          0.03       0.22       0.06       18
303          0.10       0.32       0.15       19
304          0.10       0.44       0.17       18
305          0.17       0.65       0.26       26
306          0.02       0.20       0.04       10
307          0.02       0.08       0.03       12
308          0.06       0.50       0.11        8
309          0.11       0.25       0.15       57
310          0.10       0.27       0.15       37
311          0.42       0.72       0.53       50
312          0.10       0.31       0.16       36
313          0.06       0.37       0.10       19
314          0.41       0.69       0.52       59
315          0.20       0.62       0.31       48
316          0.07       0.24       0.11       45
317          0.09       0.41       0.14       29
318          0.09       0.32       0.14       37
319          0.06       0.18       0.09       38
320          0.43       0.57       0.49       44
321          0.28       0.54       0.37       41
322          0.05       0.24       0.09       25
323          0.01       0.25       0.01        4
324          0.02       0.18       0.04       11
325          0.09       1.00       0.17        3
326          0.10       0.41       0.16       32
327          0.13       0.38       0.19       50
328          0.07       0.26       0.12       46
329          0.46       0.74       0.57       47
330          0.37       0.81       0.51       31

331          0.00       0.00       0.00       11
```

```
331    0.00    0.00    0.00     11
332    0.02    0.06    0.03     31
333    0.07    0.27    0.12     26
334    0.00    0.00    0.00     41
335    0.03    0.08    0.04     39
336    0.16    0.38    0.23     40
337    0.05    0.14    0.08     37
338    0.09    0.40    0.14     10
339    0.01    0.03    0.01     38
340    0.44    0.83    0.58     23
341    0.07    0.24    0.11     33
342    0.03    0.12    0.05     40
343    0.03    0.09    0.04     35
344    0.10    0.33    0.15     30
345    0.06    0.27    0.10     33
346    0.09    0.44    0.15     25
347    0.05    0.22    0.08      9
348    0.03    0.50    0.05      2
349    0.23    0.62    0.33     34
350    0.09    0.34    0.14     38
351    0.23    0.39    0.29     49
352    0.01    0.09    0.02     22
353    0.11    0.31    0.16     39
354    0.02    0.11    0.04     18
355    0.04    0.13    0.06     31
356    0.01    0.06    0.02     17
357    0.20    0.49    0.28     35
358    0.03    0.07    0.04     43
359    0.33    0.62    0.43     47
360    0.06    0.21    0.09     29
361    0.37    0.66    0.48     38
362    0.12    0.33    0.18     36
363    0.21    0.47    0.29     55
364    0.08    0.29    0.13     34
365    0.08    0.29    0.13     24
366    0.04    0.21    0.06     19
367    0.09    0.80    0.16      5
368    0.09    0.33    0.14     33
369    0.23    0.62    0.33     34

370    0.11    0.35    0.17     20
```

```
370        0.11       0.35       0.17        20
371        0.02       0.12       0.04        17
372        0.10       0.29       0.15        31
373        0.04       0.24       0.08        34
374        0.02       0.13       0.04        23
375        0.14       0.42       0.21        31
376        0.08       0.21       0.12        28
377        0.16       0.73       0.26        22
378        0.01       0.11       0.01         9
379        0.06       0.21       0.09        29
380        0.10       0.40       0.16        30
381        0.22       0.74       0.34        35
382        0.17       0.36       0.23        36
383        0.01       0.25       0.02         4
384        0.45       0.83       0.59        24
385        0.08       0.24       0.12        25
386        0.04       0.15       0.06        27
387        0.12       0.36       0.17        36
388        0.15       0.55       0.23        31
389        0.26       0.49       0.34        37
390        0.14       0.56       0.22        27
391        0.17       0.43       0.24        46
392        0.04       0.50       0.07         4
393        0.12       0.53       0.20        19
394        0.01       0.08       0.01        12
395        0.12       0.46       0.19        26
396        0.07       0.16       0.10        69
397        0.11       0.36       0.17        25
398        0.04       0.16       0.06        32
399        0.06       0.18       0.09        33
400        0.01       0.03       0.02        38
401        0.08       0.29       0.13        17
402        0.04       0.25       0.07        24
403        0.11       0.69       0.19        16
404        0.08       0.47       0.14        15
405        0.06       0.30       0.11        20
406        0.03       0.20       0.06        15
407        0.05       0.20       0.08        25
408        0.05       0.26       0.09        19

409        0.16       0.35       0.22        46
```

```
409          0.10         0.35         0.22          40
410          0.01         0.02         0.01          45
411          0.06         0.38         0.11          21
412          0.00         0.00         0.00           8
413          0.26         0.63         0.37          35
414          0.02         0.06         0.03          34
415          0.26         0.64         0.37          14
416          0.05         0.21         0.08          29
417          0.04         0.14         0.07          28
418          0.07         0.19         0.11          21
419          0.04         0.12         0.05          26
420          0.26         0.45         0.33          38
421          0.23         0.26         0.24         131
422          0.07         0.27         0.11          26
423          0.12         0.36         0.18          25
424          0.52         0.67         0.58          48
425          0.03         0.17         0.05          24
426          0.08         0.21         0.11          42
427          0.08         0.31         0.13          26
428          0.00         0.00         0.00          10
429          0.30         0.56         0.39          54
430          0.11         0.28         0.15          32
431          0.17         0.38         0.24          48
432          0.09         0.23         0.13          35
433          0.02         0.18         0.04          22
434          0.05         0.29         0.08          24
435          0.49         0.58         0.53          59
436          0.06         0.23         0.09          35
437          0.01         0.08         0.02          12
438          0.27         0.44         0.33          50
439          0.06         0.19         0.10          36
440          0.05         0.14         0.07          35
441          0.02         0.12         0.04           8
442          0.11         0.21         0.15          48
443          0.17         0.56         0.26          18
444          0.24         0.35         0.29          52
445          0.10         0.40         0.16          20
446          0.03         0.11         0.05          18
447          0.00         0.00         0.00           6

448          0.07         0.18         0.10          28
```

```
448      0.07      0.18      0.10       28
449      0.05      0.26      0.08       38
450      0.97      0.65      0.78      129
451      0.22      0.66      0.33       29
452      0.07      0.10      0.08       58
453      0.03      0.09      0.05       32
454      0.15      0.56      0.24       16
455      0.48      0.65      0.55       34
456      0.06      0.33      0.10        9
457      0.42      0.77      0.54       30
458      0.09      0.23      0.12       40
459      0.11      0.22      0.15       37
460      0.29      0.77      0.43       30
461      0.10      0.41      0.16       27
462      0.08      0.33      0.12       12
463      0.10      0.47      0.16       17
464      0.29      0.48      0.36       56
465      0.14      0.56      0.22        9
466      0.22      0.36      0.27       22
467      0.02      0.22      0.04        9
468      0.02      0.13      0.03       15
469      0.01      0.07      0.02       14
470      0.08      0.44      0.13       16
471      0.14      0.67      0.23       18
472      0.02      0.19      0.04       16
473      0.16      0.68      0.27       22
474      0.05      0.17      0.08       12
475      0.94      0.55      0.69      110
476      0.20      0.75      0.32       20
477      0.21      0.55      0.31       31
478      0.13      0.26      0.18       42
479      0.01      0.25      0.02        4
480      0.18      0.36      0.24       47
481      0.32      0.60      0.42       30
482      0.03      0.14      0.05       35
483      0.09      0.30      0.14       30
484      0.09      0.25      0.13       20
485      0.24      0.53      0.33       15
486      0.05      0.24      0.09       17

487      0.01      0.18      0.03       11
```

```
         488       0.08      0.25      0.12        36
         489       0.03      0.09      0.04        32
         490       0.12      0.50      0.19        14
         491       0.16      0.44      0.23        25
         492       0.20      0.57      0.30        23
         493       0.01      0.11      0.01         9
         494       0.21      0.32      0.25        37
         495       0.04      0.21      0.06        24
         496       0.20      0.41      0.27        34
         497       0.05      0.16      0.07        25
         498       0.01      0.11      0.02         9
         499       0.17      0.53      0.26        17

   micro avg       0.23      0.44      0.30     55953
   macro avg       0.15      0.36      0.20     55953
weighted avg       0.31      0.44      0.35     55953
 samples avg       0.32      0.42      0.32     55953

Time taken to run this cell : 0:12:51.098087
```

4.5.9 Applying Hyperparameter tuning using GridSearch Logistic Regression(LR) with OneVsRest Classifier

In [82]:

```
start = datetime.now()
parameters = {'estimator__C': [10**i for i in range(-6, 4, 1)]}
classifier = OneVsRestClassifier(LogisticRegression(penalty='l1'))
g_clf = GridSearchCV(classifier, parameters, n_jobs=-1, verbose=50, sco
ring='f1_micro', cv=5)
g_clf.fit(x_train_multilabel_bow, y_train)
predictions = g_clf.predict (x_test_multilabel_bow)

print("Optimal Parameters: ", g_clf.best_params_)

print("Accuracy :",metrics.accuracy_score(y_test, predictions))
print("Hamming loss ",metrics.hamming_loss(y_test,predictions))
```

```
precision = precision_score(y_test, predictions, average='micro')
recall = recall_score(y_test, predictions, average='micro')
f1 = f1_score(y_test, predictions, average='micro')

global_report = global_report.append({
                            'Vectorizer': 'BoW',
                            'Model': 'Logistic Regression - Hypertuned',
                            'NGram': '(1,4)',
                            'Parameter': g_clf.best_params_['estimator__C'
],
                            'Precision': precision,
                            'Recall': recall,
                            'F1_Score_Micro':f1
                         },
                        ignore_index=True)

print("Micro-average quality numbers")
print("Precision: {:.4f}, Recall: {:.4f}, F1-measure: {:.4f}".format(pr
ecision, recall, f1))

precision = precision_score(y_test, predictions, average='macro')
recall = recall_score(y_test, predictions, average='macro')
f1 = f1_score(y_test, predictions, average='macro')

print("Macro-average quality numbers")
print("Precision: {:.4f}, Recall: {:.4f}, F1-measure: {:.4f}".format(pr
ecision, recall, f1))

print (metrics.classification_report(y_test, predictions))
print("Time taken to run this cell :", datetime.now() - start)
```

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent work
ers.
[Parallel(n_jobs=-1)]: Done   1 tasks      | elapsed:  6.1min
[Parallel(n_jobs=-1)]: Done   2 tasks      | elapsed:  6.1min
[Parallel(n_jobs=-1)]: Done   3 tasks      | elapsed:  6.1min
[Parallel(n_jobs=-1)]: Done   4 tasks      | elapsed:  6.1min
[Parallel(n_jobs=-1)]: Done   5 tasks      | elapsed:  6.2min
[Parallel(n_jobs=-1)]: Done   6 tasks      | elapsed:  6.2min
```

```
[Parallel(n_jobs=-1)]: Done   7 tasks      | elapsed:  6.2min
[Parallel(n_jobs=-1)]: Done   8 tasks      | elapsed:  6.3min
[Parallel(n_jobs=-1)]: Done   9 tasks      | elapsed: 12.0min
[Parallel(n_jobs=-1)]: Done  10 tasks      | elapsed: 12.8min
[Parallel(n_jobs=-1)]: Done  11 tasks      | elapsed: 20.1min
[Parallel(n_jobs=-1)]: Done  12 tasks      | elapsed: 20.6min
[Parallel(n_jobs=-1)]: Done  13 tasks      | elapsed: 21.5min
[Parallel(n_jobs=-1)]: Done  14 tasks      | elapsed: 25.7min
[Parallel(n_jobs=-1)]: Done  15 tasks      | elapsed: 26.8min
[Parallel(n_jobs=-1)]: Done  16 tasks      | elapsed: 27.8min
[Parallel(n_jobs=-1)]: Done  17 tasks      | elapsed: 31.5min
[Parallel(n_jobs=-1)]: Done  18 tasks      | elapsed: 33.1min
[Parallel(n_jobs=-1)]: Done  19 tasks      | elapsed: 38.3min
[Parallel(n_jobs=-1)]: Done  20 tasks      | elapsed: 41.3min
[Parallel(n_jobs=-1)]: Done  21 tasks      | elapsed: 47.3min
[Parallel(n_jobs=-1)]: Done  22 tasks      | elapsed: 52.0min
[Parallel(n_jobs=-1)]: Done  23 tasks      | elapsed: 52.1min
[Parallel(n_jobs=-1)]: Done  24 tasks      | elapsed: 52.8min
[Parallel(n_jobs=-1)]: Done  25 tasks      | elapsed: 56.5min
[Parallel(n_jobs=-1)]: Done  26 tasks      | elapsed: 66.4min
[Parallel(n_jobs=-1)]: Done  27 tasks      | elapsed: 74.7min
[Parallel(n_jobs=-1)]: Done  28 tasks      | elapsed: 76.3min
[Parallel(n_jobs=-1)]: Done  29 tasks      | elapsed: 83.6min
[Parallel(n_jobs=-1)]: Done  30 tasks      | elapsed: 89.3min
[Parallel(n_jobs=-1)]: Done  31 tasks      | elapsed: 119.8min
[Parallel(n_jobs=-1)]: Done  32 tasks      | elapsed: 121.2min
[Parallel(n_jobs=-1)]: Done  33 tasks      | elapsed: 122.8min
[Parallel(n_jobs=-1)]: Done  34 tasks      | elapsed: 130.9min
[Parallel(n_jobs=-1)]: Done  35 tasks      | elapsed: 145.6min
[Parallel(n_jobs=-1)]: Done  37 out of  50 | elapsed: 162.8min remainin
g: 57.2min
[Parallel(n_jobs=-1)]: Done  39 out of  50 | elapsed: 198.9min remainin
g: 56.1min
[Parallel(n_jobs=-1)]: Done  41 out of  50 | elapsed: 208.7min remainin
g: 45.8min
[Parallel(n_jobs=-1)]: Done  43 out of  50 | elapsed: 222.6min remainin
g: 36.2min
[Parallel(n_jobs=-1)]: Done  45 out of  50 | elapsed: 236.9min remainin
g: 26.3min
[Parallel(n_jobs=-1)]: Done  47 out of  50 | elapsed: 257.1min remainin
```

```
[Parallel(n_jobs=-1)]: Done  47 out of  50 | elapsed: [illegible]min remainin

g: 16.4min
[Parallel(n_jobs=-1)]: Done  50 out of  50 | elapsed: 266.2min finished
Optimal Parameters:  {'estimator__C': 1}
Accuracy : 0.18636666666666668
Hamming loss  0.0033992666666666665
Micro-average quality numbers
Precision: 0.5644, Recall: 0.3888, F1-measure: 0.4604
Macro-average quality numbers
Precision: 0.4321, Recall: 0.3113, F1-measure: 0.3528
              precision    recall  f1-score   support

           0       0.61      0.40      0.48      1111
           1       0.47      0.30      0.36      2052
           2       0.54      0.42      0.48      2388
           3       0.66      0.56      0.60      2226
           4       0.66      0.50      0.57      2014
           5       0.30      0.14      0.19       642
           6       0.64      0.41      0.50      1756
           7       0.89      0.72      0.79      1690
           8       0.55      0.24      0.34       341
           9       0.77      0.78      0.77      2344
          10       0.57      0.39      0.46       821
          11       0.43      0.29      0.34      1143
          12       0.64      0.40      0.49       768
          13       0.55      0.37      0.44       745
          14       0.69      0.56      0.62       952
          15       0.46      0.27      0.34       314
          16       0.42      0.27      0.32       624
          17       0.72      0.56      0.63       535
          18       0.73      0.54      0.62       631
          19       0.88      0.57      0.69       101
          20       0.53      0.26      0.35       245
          21       0.73      0.54      0.62       694
          22       0.48      0.32      0.38       568
          23       0.51      0.32      0.39       423
          24       0.52      0.34      0.41       406
          25       0.70      0.49      0.58      1373
          26       0.52      0.34      0.41       253
          27       0.24      0.16      0.19       357
```

```
27          0.24        0.18        0.19         337

28          0.50        0.28        0.36         222
29          0.58        0.30        0.40         273
30          0.48        0.32        0.39         308
31          0.45        0.30        0.36         256
32          0.69        0.41        0.52         295
33          0.22        0.11        0.15         263
34          0.74        0.53        0.62         256
35          0.42        0.34        0.37         280
36          0.47        0.32        0.38         290
37          0.36        0.17        0.23         200
38          0.34        0.29        0.32         109
39          0.62        0.39        0.48         209
40          0.53        0.40        0.45         113
41          0.54        0.28        0.37         197
42          0.51        0.56        0.53          52
43          0.17        0.10        0.13         179
44          0.77        0.50        0.61         431
45          0.29        0.11        0.16          47
46          0.44        0.38        0.41          37
47          0.54        0.34        0.41         155
48          0.65        0.50        0.57         254
49          0.50        0.34        0.41         201
50          0.40        0.28        0.33          61
51          0.92        0.73        0.82         246
52          0.67        0.62        0.64         146
53          0.92        0.91        0.92         516
54          0.75        0.55        0.64         170
55          0.21        0.10        0.13         234
56          0.19        0.11        0.14         357
57          0.57        0.27        0.37          78
58          0.87        0.68        0.76         102
59          0.35        0.24        0.28         122
60          0.72        0.57        0.64         138
61          0.22        0.17        0.19          36
62          0.35        0.19        0.25         172
63          0.08        0.03        0.05          60
64          0.59        0.51        0.55         106
65          0.35        0.32        0.34          34
66          0.29        0.25        0.27         101
```

```
 67     0.43     0.34     0.38      38
 68     0.56     0.39     0.46     104
 69     0.32     0.17     0.22     144
 70     0.38     0.24     0.29     135
 71     0.24     0.14     0.17     190
 72     0.56     0.36     0.44     139
 73     0.21     0.04     0.07      69
 74     0.07     0.04     0.05     133
 75     0.73     0.46     0.57     181
 76     0.43     0.43     0.43     113
 77     0.42     0.22     0.29     158
 78     0.28     0.15     0.19     142
 79     0.45     0.18     0.25      96
 80     0.52     0.25     0.34     101
 81     0.44     0.20     0.27      56
 82     0.12     0.05     0.07      62
 83     0.58     0.44     0.50      77
 84     0.45     0.15     0.23     100
 85     0.55     0.43     0.48      54
 86     0.30     0.16     0.21      79
 87     0.44     0.24     0.31      92
 88     0.45     0.31     0.37     124
 89     0.60     0.39     0.47     101
 90     0.27     0.10     0.15      40
 91     0.56     0.44     0.49      66
 92     0.42     0.31     0.36      58
 93     0.67     0.42     0.51     161
 94     0.40     0.25     0.30     130
 95     0.44     0.26     0.32      47
 96     0.64     0.50     0.57     107
 97     0.50     0.31     0.38      39
 98     0.22     0.12     0.15     111
 99     0.39     0.22     0.28      95
100     0.29     0.22     0.25     129
101     0.74     0.46     0.57      91
102     0.19     0.19     0.19      27
103     0.86     0.86     0.86      90
104     0.20     0.07     0.11     124
105     0.30     0.17     0.22      76
```

```
105         0.30        0.17        0.22          76

106         0.35        0.18        0.24         371
107         0.68        0.40        0.51         114
108         0.49        0.37        0.42          98
109         0.60        0.40        0.48          63
110         0.63        0.50        0.56          24
111         0.54        0.28        0.37          53
112         0.16        0.09        0.12          65
113         0.53        0.36        0.43          70
114         0.50        0.56        0.53          27
115         0.31        0.12        0.18          72
116         0.33        0.19        0.24          27
117         0.54        0.32        0.40          90
118         0.50        0.40        0.44          95
119         0.37        0.27        0.31          92
120         0.28        0.21        0.24          87
121         0.57        0.36        0.44          45
122         0.18        0.08        0.11         182
123         0.31        0.12        0.17          94
124         0.72        0.42        0.53          62
125         0.73        0.44        0.55          91
126         0.69        0.51        0.58          69
127         0.49        0.48        0.48          73
128         1.00        0.52        0.68          25
129         0.20        0.03        0.05          68
130         0.31        0.19        0.23         123
131         0.25        0.15        0.19          84
132         0.00        0.00        0.00          67
133         0.30        0.20        0.24         127
134         0.47        0.60        0.52          45
135         0.49        0.41        0.44          88
136         0.00        0.00        0.00          63
137         0.89        0.77        0.83          96
138         0.19        0.08        0.12          71
139         0.84        0.64        0.73          92
140         0.21        0.13        0.16          23
141         0.42        0.23        0.30          90
142         0.15        0.20        0.17          10
143         0.27        0.18        0.22          44
144         0.59        0.43        0.50          67
```

```
145       0.65      0.46      0.54         131
146       0.17      0.07      0.10          83
147       0.27      0.12      0.17          32
148       0.30      0.15      0.20         115
149       0.47      0.29      0.36          63
150       0.62      0.35      0.45          83
151       0.73      0.47      0.57         101
152       0.27      0.10      0.15          29
153       0.92      0.85      0.89         191
154       0.86      0.69      0.76          54
155       0.58      0.35      0.43          84
156       0.50      0.38      0.43          37
157       0.32      0.38      0.35          65
158       0.33      0.20      0.25          46
159       0.79      0.53      0.63          80
160       0.19      0.11      0.14          66
161       0.17      0.07      0.10          56
162       0.41      0.23      0.29         127
163       0.67      0.59      0.63         111
164       0.17      0.06      0.09          32
165       0.25      0.14      0.18          28
166       0.03      0.01      0.02          98
167       0.68      0.58      0.63          88
168       0.76      0.54      0.63          59
169       0.05      0.02      0.03          42
170       0.29      0.50      0.36           4
171       0.58      0.40      0.47          95
172       0.27      0.17      0.21          54
173       0.74      0.54      0.62          65
174       0.60      0.48      0.54          31
175       0.62      0.47      0.54          32
176       0.54      0.34      0.42          58
177       0.42      0.21      0.28          76
178       0.05      0.02      0.03          55
179       0.80      0.85      0.82          74
180       0.78      0.62      0.70          64
181       0.35      0.12      0.18          57
182       0.50      0.25      0.33          36
183       0.35      0.42      0.38          52
```

```
184       0.49      0.44      0.46         48
185       0.25      0.31      0.28         16
186       0.12      0.07      0.09         28
187       0.16      0.17      0.16         36
188       0.28      0.31      0.29         26
189       0.19      0.09      0.12         44
190       0.55      0.35      0.43         46
191       0.32      0.20      0.25         75
192       0.21      0.24      0.22         50
193       0.50      0.45      0.47         20
194       0.31      0.19      0.23         27
195       0.38      0.50      0.43          6
196       0.17      0.13      0.15         68
197       0.33      0.41      0.37         29
198       0.33      0.17      0.23        104
199       0.45      0.25      0.32         36
200       1.00      1.00      1.00          4
201       0.75      0.75      0.75          4
202       0.25      0.07      0.11         96
203       0.82      0.61      0.70         61
204       0.45      0.17      0.25         82
205       0.46      0.35      0.40         34
206       0.66      0.41      0.50         66
207       0.17      0.09      0.12         97
208       0.14      0.06      0.08         89
209       0.78      0.53      0.63         55
210       0.72      0.54      0.62         78
211       0.22      0.08      0.11         78
212       0.68      0.48      0.57        158
213       0.24      0.09      0.13         44
214       0.62      0.60      0.61         35
215       0.89      0.71      0.79         48
216       0.66      0.66      0.66         62
217       0.50      0.36      0.42         11
218       0.81      0.44      0.57         68
219       0.31      0.22      0.25         60
220       0.33      0.08      0.13         25
221       0.31      0.14      0.19         57
222       0.59      0.53      0.56         36
```

```
223         0.37        0.17        0.23         88
224         0.26        0.15        0.19         46
225         0.32        0.10        0.15         60
226         0.26        0.14        0.18         65
227         0.50        0.57        0.53          7
228         0.41        0.58        0.48         12
229         0.18        0.13        0.15         68
230         0.44        0.28        0.34         40
231         0.08        0.04        0.05         26
232         0.72        0.60        0.65         30
233         0.33        0.20        0.25         41
234         0.19        0.08        0.11         53
235         0.51        0.54        0.53         35
236         0.62        0.28        0.38         18
237         0.17        0.09        0.12         22
238         0.67        0.56        0.61         59
239         0.67        0.56        0.61         43
240         0.34        0.27        0.30         45
241         0.33        0.13        0.19         46
242         0.13        0.05        0.08         38
243         0.94        0.29        0.44         56
244         0.24        0.17        0.20         35
245         0.10        0.05        0.06         42
246         0.07        0.03        0.04         33
247         0.38        0.28        0.32         47
248         0.62        0.20        0.30         25
249         0.56        0.59        0.57         39
250         0.56        0.19        0.29         77
251         0.74        0.55        0.63         56
252         0.20        0.06        0.09         72
253         0.67        1.00        0.80          4
254         0.48        0.34        0.40         29
255         0.41        0.27        0.32        113
256         0.75        0.71        0.73         59
257         0.18        0.05        0.08         59
258         0.83        0.62        0.71         39
259         0.64        0.58        0.61         12
260         0.56        0.56        0.56          9
261         0.94        0.66        0.77         44
```

```
262         0.76        0.50        0.60         32
263         0.18        0.17        0.17        156
264         1.00        0.60        0.75          5
265         0.18        0.07        0.10        198
266         0.07        0.03        0.04         40
267         0.20        0.14        0.16         29
268         0.00        0.00        0.00         39
269         0.11        0.17        0.13          6
270         0.75        0.60        0.67          5
271         0.40        0.12        0.18         17
272         0.30        0.13        0.18         54
273         0.38        0.26        0.31         23
274         0.00        0.00        0.00        126
275         0.44        0.34        0.39         32
276         0.40        0.60        0.48         10
277         0.82        0.61        0.70         67
278         0.47        0.36        0.41         53
279         0.33        0.12        0.18         16
280         0.72        0.68        0.70         19
281         0.30        0.13        0.18         61
282         0.38        0.21        0.27         81
283         0.41        0.34        0.37         94
284         0.46        0.35        0.40         31
285         0.31        0.21        0.25         43
286         0.35        0.22        0.27         79
287         0.29        0.40        0.33         20
288         0.83        0.88        0.86         17
289         0.73        0.64        0.69         56
290         0.15        0.08        0.10         63
291         0.48        0.33        0.39         46
292         0.43        0.36        0.39         50
293         0.62        0.29        0.40         17
294         0.36        0.15        0.21         27
295         0.04        0.02        0.02         63
296         0.23        0.12        0.16         25
297         0.04        0.03        0.03         38
298         0.25        0.10        0.14         62
299         0.67        0.59        0.63         49
300         0.19        0.15        0.17         39
```

```
301         0.00        0.00        0.00           8
302         0.17        0.11        0.13          18
303         0.15        0.11        0.12          19
304         0.60        0.67        0.63          18
305         0.44        0.46        0.45          26
306         0.23        0.30        0.26          10
307         0.17        0.17        0.17          12
308         0.60        0.38        0.46           8
309         0.37        0.12        0.18          57
310         0.31        0.22        0.25          37
311         0.80        0.72        0.76          50
312         0.38        0.31        0.34          36
313         0.24        0.26        0.25          19
314         0.88        0.71        0.79          59
315         0.79        0.71        0.75          48
316         0.22        0.13        0.17          45
317         0.62        0.34        0.44          29
318         0.33        0.32        0.33          37
319         0.24        0.11        0.15          38
320         0.81        0.68        0.74          44
321         0.64        0.39        0.48          41
322         0.15        0.08        0.11          25
323         0.25        0.25        0.25           4
324         0.12        0.09        0.11          11
325         0.75        1.00        0.86           3
326         0.45        0.28        0.35          32
327         0.49        0.34        0.40          50
328         0.26        0.11        0.15          46
329         0.90        0.74        0.81          47
330         0.91        0.68        0.78          31
331         0.00        0.00        0.00          11
332         0.00        0.00        0.00          31
333         0.08        0.04        0.05          26
334         0.07        0.02        0.04          41
335         0.08        0.03        0.04          39
336         0.69        0.45        0.55          40
337         0.28        0.22        0.24          37
338         0.83        0.50        0.62          10
339         0.00        0.00        0.00          38
```

```
340       0.94      0.70      0.80        23
341       0.26      0.15      0.19        33
342       0.16      0.07      0.10        40
343       0.38      0.17      0.24        35
344       0.52      0.40      0.45        30
345       0.08      0.03      0.04        33
346       0.65      0.44      0.52        25
347       0.75      0.33      0.46         9
348       0.50      0.50      0.50         2
349       0.79      0.68      0.73        34
350       0.36      0.47      0.41        38
351       0.57      0.27      0.36        49
352       0.00      0.00      0.00        22
353       0.35      0.23      0.28        39
354       0.50      0.17      0.25        18
355       0.21      0.13      0.16        31
356       0.14      0.06      0.08        17
357       0.73      0.69      0.71        35
358       0.44      0.09      0.15        43
359       0.56      0.64      0.59        47
360       0.20      0.10      0.14        29
361       0.65      0.58      0.61        38
362       0.41      0.31      0.35        36
363       0.36      0.36      0.36        55
364       0.29      0.21      0.24        34
365       0.27      0.17      0.21        24
366       0.12      0.11      0.11        19
367       0.22      0.40      0.29         5
368       0.20      0.12      0.15        33
369       0.69      0.65      0.67        34
370       0.42      0.40      0.41        20
371       0.07      0.06      0.06        17
372       0.31      0.13      0.18        31
373       0.11      0.03      0.05        34
374       0.15      0.13      0.14        23
375       0.41      0.39      0.40        31
376       0.62      0.29      0.39        28
377       0.78      0.64      0.70        22
378       0.36      0.56      0.43         9
```

```
378        0.30       0.30       0.43          9

379        0.33       0.21       0.26         29
380        0.26       0.27       0.26         30
381        0.72       0.60       0.66         35
382        0.41       0.31       0.35         36
383        0.17       0.25       0.20          4
384        0.90       0.79       0.84         24
385        0.60       0.36       0.45         25
386        0.08       0.04       0.05         27
387        0.75       0.33       0.46         36
388        0.61       0.45       0.52         31
389        0.65       0.46       0.54         37
390        0.39       0.33       0.36         27
391        0.58       0.41       0.48         46
392        0.25       0.50       0.33          4
393        0.26       0.32       0.29         19
394        0.00       0.00       0.00         12
395        0.50       0.42       0.46         26
396        0.19       0.04       0.07         69
397        0.67       0.72       0.69         25
398        0.14       0.06       0.09         32
399        0.32       0.18       0.23         33
400        0.32       0.18       0.23         38
401        0.50       0.29       0.37         17
402        0.12       0.04       0.06         24
403        0.73       0.50       0.59         16
404        0.83       0.33       0.48         15
405        0.00       0.00       0.00         20
406        0.00       0.00       0.00         15
407        0.21       0.12       0.15         25
408        0.29       0.26       0.28         19
409        0.53       0.37       0.44         46
410        0.00       0.00       0.00         45
411        0.17       0.10       0.12         21
412        0.00       0.00       0.00          8
413        0.50       0.51       0.51         35
414        0.00       0.00       0.00         34
415        1.00       0.57       0.73         14
416        0.23       0.10       0.14         29
417        0.33       0.14       0.20         28
```

```
418          0.43        0.14        0.21          21
419          0.25        0.15        0.19          26
420          0.52        0.29        0.37          38
421          0.17        0.02        0.04         131
422          0.29        0.23        0.26          26
423          0.40        0.40        0.40          25
424          0.81        0.62        0.71          48
425          0.17        0.08        0.11          24
426          0.29        0.14        0.19          42
427          0.39        0.35        0.37          26
428          0.11        0.10        0.11          10
429          0.54        0.39        0.45          54
430          0.55        0.34        0.42          32
431          0.39        0.27        0.32          48
432          0.44        0.20        0.27          35
433          0.38        0.14        0.20          22
434          0.18        0.12        0.15          24
435          0.78        0.59        0.67          59
436          0.22        0.14        0.17          35
437          0.00        0.00        0.00          12
438          0.61        0.38        0.47          50
439          0.09        0.03        0.04          36
440          0.21        0.14        0.17          35
441          0.43        0.38        0.40           8
442          0.21        0.19        0.20          48
443          0.75        0.50        0.60          18
444          0.64        0.35        0.45          52
445          0.60        0.15        0.24          20
446          0.14        0.06        0.08          18
447          0.00        0.00        0.00           6
448          0.27        0.14        0.19          28
449          0.25        0.13        0.17          38
450          0.98        0.84        0.91         129
451          0.72        0.72        0.72          29
452          0.26        0.12        0.16          58
453          0.10        0.03        0.05          32
454          0.43        0.38        0.40          16
455          0.92        0.68        0.78          34
456          0.40        0.44        0.42           9
```

```
457         0.86        0.83        0.85         30
458         0.36        0.23        0.28         40
459         0.47        0.22        0.30         37
460         0.81        0.73        0.77         30
461         0.55        0.41        0.47         27
462         0.14        0.08        0.11         12
463         0.33        0.29        0.31         17
464         0.80        0.50        0.62         56
465         0.40        0.44        0.42          9
466         0.38        0.23        0.29         22
467         0.40        0.22        0.29          9
468         0.00        0.00        0.00         15
469         0.25        0.07        0.11         14
470         0.55        0.38        0.44         16
471         0.85        0.61        0.71         18
472         0.29        0.31        0.30         16
473         0.48        0.59        0.53         22
474         0.67        0.17        0.27         12
475         1.00        0.97        0.99        110
476         0.48        0.70        0.57         20
477         0.62        0.52        0.56         31
478         0.41        0.17        0.24         42
479         0.00        0.00        0.00          4
480         0.52        0.26        0.34         47
481         0.75        0.40        0.52         30
482         0.00        0.00        0.00         35
483         0.00        0.00        0.00         30
484         0.38        0.15        0.21         20
485         0.80        0.53        0.64         15
486         0.33        0.18        0.23         17
487         0.33        0.09        0.14         11
488         0.43        0.17        0.24         36
489         0.18        0.09        0.12         32
490         0.36        0.36        0.36         14
491         0.50        0.24        0.32         25
492         0.37        0.57        0.45         23
493         0.00        0.00        0.00          9
494         0.26        0.14        0.18         37
495         0.12        0.08        0.10         24
```

```
         496       0.80      0.35      0.49        34
         497       0.20      0.12      0.15        25
         498       0.12      0.11      0.12         9
         499       0.47      0.41      0.44        17

   micro avg       0.56      0.39      0.46     55953
   macro avg       0.43      0.31      0.35     55953
weighted avg       0.54      0.39      0.45     55953
 samples avg       0.43      0.37      0.37     55953

Time taken to run this cell : 5:20:04.315891
```

Conclusions

In [83]:

```
global_report
```

Out[83]:

	Vectorizer	Model	NGram	Parameter	Precision	Recall	F1_Score_Micro
0	Tf-IDF	Logistic Regression (SGD with log loss)	(1,3)	0.00001	0.713139	0.314389	0.436393
1	Tf-IDF	Logistic Regression	(1,3)	1.00000	0.708631	0.329258	0.449610
2	BoW	Logistic Regression (SGD with log loss)	(1,4)	0.00001	0.229672	0.442675	0.302433
3	BoW	Logistic Regression	(1,4)	1.00000	0.564393	0.388719	0.460366
4	BoW	Logistic Regression (SGD with log loss) - Hypertuned	(1,4)	0.00100	0.580476	0.306168	0.400889
5	Tf-IDF	Linear SVM (SGD with hinge loss)	(1,3)	0.00001	0.792334	0.293353	0.428178
6	BoW	Linear SVM (SGD with hinge loss)	(1,4)	0.00001	0.230407	0.443033	0.303153
7	BoW	Logistic Regression - Hypertuned	(1,4)	1.00000	0.564387	0.388826	0.460440

Summarized Conclusion -

This is the StackOverflow Tag Prediction problem which is a multi-label classification based problem.

For multilabel classification, we used OneVsRest Classifier technique to solve.

Dataset size - ~6M However, due to memory constraints & performance issues, we tried to limit the dataset size to 0.15M that is, 150K points only.

Performance metric we used - Micro F1 Score as the dataset is quite imbalanced.

We tried 2 different Vectorizer - Term Frequency-Inverse Document Frequency(TF-IDF) and Bag Of Words.

The Feature Engineering which we used was to use 3-gram and 4-gram vectorizer.

Since this involves high features in the dataset, we tried to limit ourselves with Logistic Regression and SVM only. Since these algorithms tend to perform well with high feature dataset.

Result -

Logistic Regression with Lasso regression penalty tuned with 'C' hyperparameter and cross validation set to 5, we got the highest micro f1 score of *0.46* with best parameter to 1.

However, this F1Score can be increased by two ways.

1. Increasing the dataset size from 0.15M to atleast 5M or 10M or 15M.
2. Increasing the output tag limit from 500 to atleast 3k.