In [1]:

```
from IPython.core.display import display, HTML
display(HTML("<style>.container { width:100% !important; }</style>"))
```

In [2]:

```
import warnings
warnings.filterwarnings("ignore")
import pandas as pd
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
from sqlalchemy import create_engine # database connection
import datetime as dt
from nltk.corpus import stopwords
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 tqdm import tqdm
from sklearn.calibration import CalibratedClassifierCV
%autosave 180
```

```
Autosaving every 180 seconds
```

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 (https://www.kaggle.com/c/facebook-recruiting-iii-keyword-extraction/data)
Youtube : https://youtu.be/nNDqbUhtIRg (https://youtu.be/nNDqbUhtIRg)
Research paper : https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tagging-1.pdf (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 (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 (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 contai
n 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 (https://www.kaggle.com/wiki/MeanFScore)
http://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html (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 (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 [8]:

```
#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 [9]:

```
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 : 0:01:34.650409
```

3.1.3 Checking for duplicates

In [10]:

```
#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(*) as 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 first to genarate train.db file")
```

```
Time taken to run this cell : 0:16:10.687961
```

In [11]:

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

Out[11]:

	Title	Body	Tags	cnt_dup
0	Implementing Boundary Value Analysis of S...	<pre> <code>#include<iostream>\n#include&...	c++ c	1
1	Dynamic Datagrid Binding in Silverlight?	<p>I should do binding for datagrid dynamicall...	c# silverlight data-binding	1
2	Dynamic Datagrid Binding in Silverlight?	<p>I should do binding for datagrid dynamicall...	c# silverlight data-binding columns	1
3	java.lang.NoClassDefFoundError: javax/serv...	<p>I followed the guide in <a href="http://sta...	jsp jstl	1
4	java.sql.SQLException:[Microsoft][ODBC Dri...	I use the following code \n\n<pre> <code>...	java jdbc	2

In [12]:

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

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

3.1.4 Checking for the number of times each question has appeared in the database

In [0]:

```
#Number of times each question appeared in the database
#2656284 question occured only 1 time. 1272336 occurs 2 times. 277575 questions occurs 3 times and so on.
df_no_dup.cnt_dup.value_counts()
```

Out[0]:

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

In [ ]:

```
start = datetime.now()
df_no_dup["tag_count"] = df_no_dup["Tags"].apply(lambda text: len(str(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()
```

3.1.5 Checking for the distribution of number of tags per question.

In [0]:

```
#Distribution of number of tags per question.
#1206157 questions has 3 tags, 1111706 has 2 tags, 814996 questions has 4 tags, 568298 questions has one tag & 50
5158 questions has 5 tags.
df_no_dup.tag_count.value_counts()
```

Out[0]:

```
3    1206157
2    1111706
4     814996
1     568298
5     505158
Name: tag_count, dtype: int64
```

In [13]:

```
#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 [3]:

```
#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""", con)
    #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 above cells to generate train.db file")
```

Time taken to run this cell : 0:00:49.192848

Observations from the above analysis.

1. There were almost 30% questions which were duplicates. So the first thing we did, is remove the duplicate questions from the actual dataset and save it in a new dataset.
2. 2656284 questions have occured only 1 time. 1272336 occurs 2 times. 277575 questions occurs 3 times and so on.
3. There are 1206157 questions which have 3 tags, 1111706 have 2 tags, 814996 questions have 4 tags, 568298 questions have one tag & 505158 questions have 5 tags.

3.1.6 Analysis of Title texts and Body texts

In [3]:

```
#Load the de-duplicated dataset
start = datetime.now()
con = sqlite3.connect('train_no_dup.db')
dataframe = pd.read_sql_query("""SELECT * FROM no_dup_train""", con)
#Always remember to close the database
con.close()
```

In [40]:

```
#Utiliy functions for feature extraction
#Counting the number of code segments present in a given body text
def count_code(string):
    if string.__contains__("<code>"):
        return string.count("<code>")
    else:
        return int(0)

#Returns the count of 'http' elements present in a string. Return 0 otherwise.
def count_http(string):
    if string.__contains__("http"):
        return string.count("http")
    else:
        return int(0)

#Returns the number of times a reference link is present in a string
def count_href(string):
    if string.__contains__("a href"):
        return string.count("a href")
    else:
        return int(0)

#Number of times a greater than sign appears in a string
def count_greater(string):
    if string.__contains__(">"):
        return string.count(">")
    else:
        return int(0)
```

In [43]:

```
#Simple feature engineering
basic_feats = pd.DataFrame()
basic_feats["Length_Title"] = dataframe['Title'].apply(lambda x: len(str(x))) #Length of RAW Title text
basic_feats["Length_Body"] = dataframe['Body'].apply(lambda x: len(str(x))) #Length of RAW body text
basic_feats['count_Body_code'] = dataframe['Body'].apply(lambda x: count_code(str(x))) #Check how many code segme
nts are present in a give body text
basic_feats['count_Body_http'] = dataframe['Body'].apply(lambda x: count_http(str(x))) #Lazy way to count the num
ber of URLs present in a body text. Not 100% accurate, but close enough
basic_feats['count_Body_href'] = dataframe['Body'].apply(lambda x: count_href(str(x))) #Lazy way to count the ref
erence to an externel site. Not 100% accurate, but close enough
basic_feats['count_Body_grtsign'] = dataframe['Body'].apply(lambda x: count_greater(str(x))) #Very lazy way to co
unt html tags present in a string. Not 100% accurate, but close enough

#Save the dataset containing basic features
basic_feats.to_csv('basic_feats.csv', columns=basic_feats.columns)
basic_feats.head()
```

Out[43]:

	Length_Title	Length_Body	count_Body_code	count_Body_http	count_Body_href	count_Body_grtsign
0	79	2037	2	0	0	14
1	44	860	1	0	0	8
2	44	860	1	0	0	8
3	80	665	2	2	1	18
4	83	973	2	0	0	14

3.1.7 High level statistics of the dataset containing simple features

In [48]:

```
#Get a high level stats of the given dataset
basic_feats.describe()
```

Out[48]:

	Length_Title	Length_Body	count_Body_code	count_Body_http	count_Body_href	count_Body_grtsign
count	4.206315e+06	4.206315e+06	4.206315e+06	4.206315e+06	4.206315e+06	4.206315e+06
mean	5.079118e+01	1.120116e+03	1.574385e+00	5.555927e-01	2.690229e-01	1.538712e+01
std	2.006422e+01	1.487497e+03	2.478619e+00	1.792181e+00	9.447945e-01	1.337127e+01
min	5.000000e+00	4.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	3.600000e+01	4.160000e+02	0.000000e+00	0.000000e+00	0.000000e+00	6.000000e+00
50%	4.800000e+01	7.250000e+02	1.000000e+00	0.000000e+00	0.000000e+00	1.200000e+01
75%	6.200000e+01	1.285000e+03	2.000000e+00	0.000000e+00	0.000000e+00	2.000000e+01
max	2.190000e+02	4.825800e+04	5.000000e+02	5.000000e+02	3.420000e+02	1.886000e+03

In [80]:

```
#Get the percentage of questions which does not have a code snippet included in their body
zero = basic_feats[basic_feats['count_Body_code'] == 0].shape[0]
per = (zero/basic_feats.shape[0]) * 100
print("Percentage of users who have not included any code snippet in the body text: {:.2f}%".format(per))

#Get the percentage of questions which does not have a http referrence included in their body
zero = basic_feats[basic_feats['count_Body_http'] == 0].shape[0]
per = (zero/basic_feats.shape[0]) * 100
print("Percentage of users who have not included any http referrence URL in the body text: {:.2f}%".format(per))

#Get the percentage of questions which are provided with external reference links in their body text
zero = basic_feats[basic_feats['count_Body_href'] == 0].shape[0]
per = (1-zero/basic_feats.shape[0]) * 100
print("Percentage of users who have used an external referrence in their body text: {:.2f}%".format(per))
```

```
Percentage of users who have not included any code snippet in the body text: 42.30%
Percentage of users who have not included any http referrence URL in the body text: 76.14%
Percentage of users who have used an external referrence in their body text: 17.39%
```

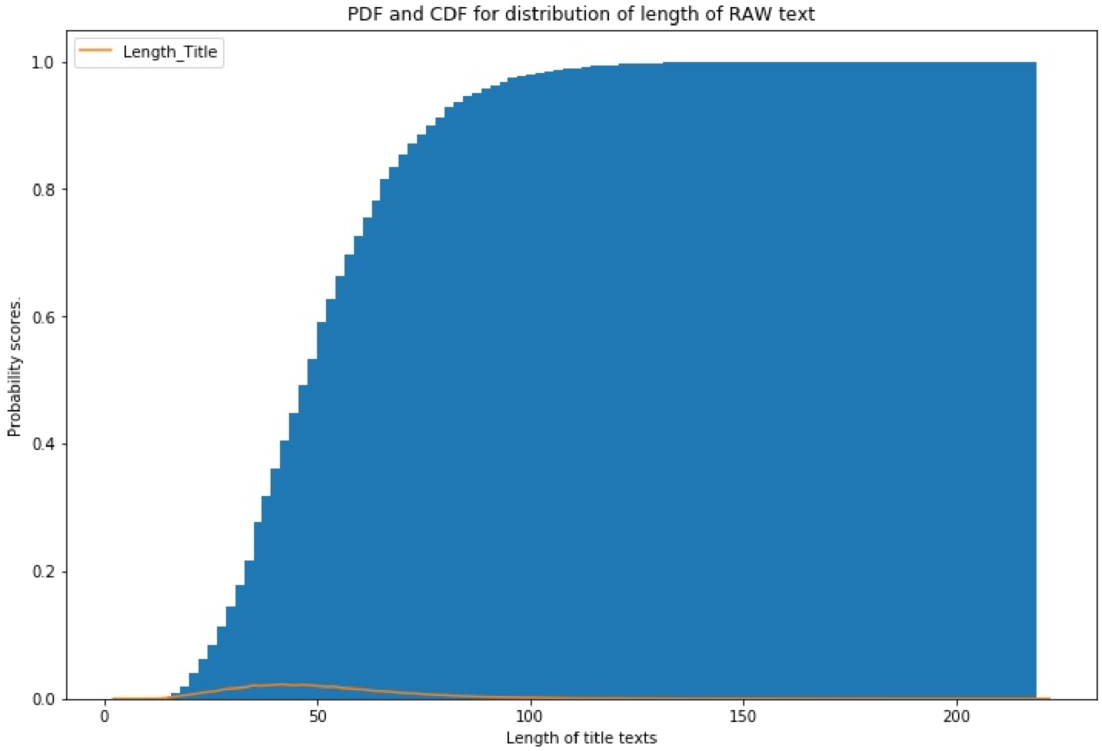
Observations from the above analysis:

1. A quick high level statistic revealed that the average length of reviews are somewhere around 50.
2. Most users (50%) writes title text between 36 and 62 characters.
3. The average length of the body text given by all the users is somewhere around 1120 characters.
4. Almost 25% users has used an external reference in their question ('href' tag).
5. More than 50% users writes 725 words on an average to describe the problem.
6. On an average, each user has included less than 1 URLs in their body text.
7. There are almost 42% questions which does not have any code snippet included in the body text.
8. More than 75% questions have 2 or more code snippets included.
9. The maximum length of body text seen for any user is as long as 48K characters!
10. As many as 76% users did not include any http URL in their body text.
11. As many as 17% users used an external link in their body text

3.1.8 Histograms of some of the extracted features

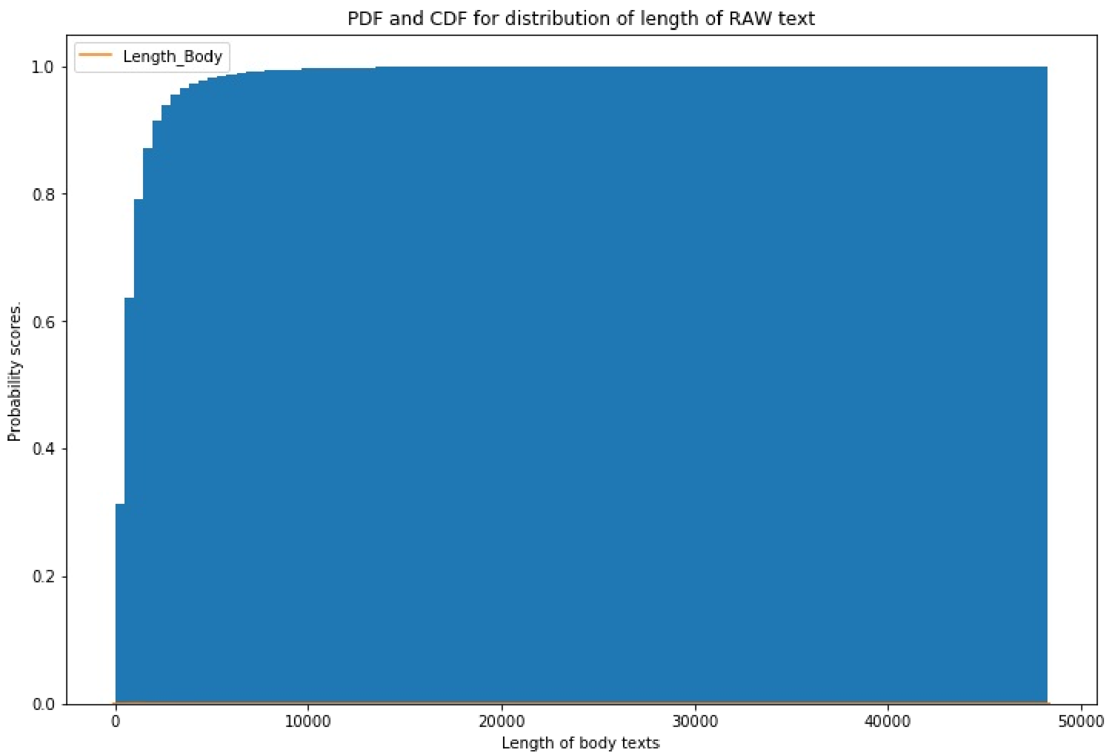
In [104]:

```
import scipy.stats as ss
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns  # for nicer graphics

plt.figure(figsize=(12, 8))
myHist = plt.hist(basic_feats['Length_Title'].values, 100, density=True, cumulative=True)
plt.title('PDF and CDF for distribution of length of RAW text')
plt.xlabel('Length of title texts')
plt.ylabel('Probability scores.')
sns.kdeplot(basic_feats['Length_Title']);
plt.show()
```



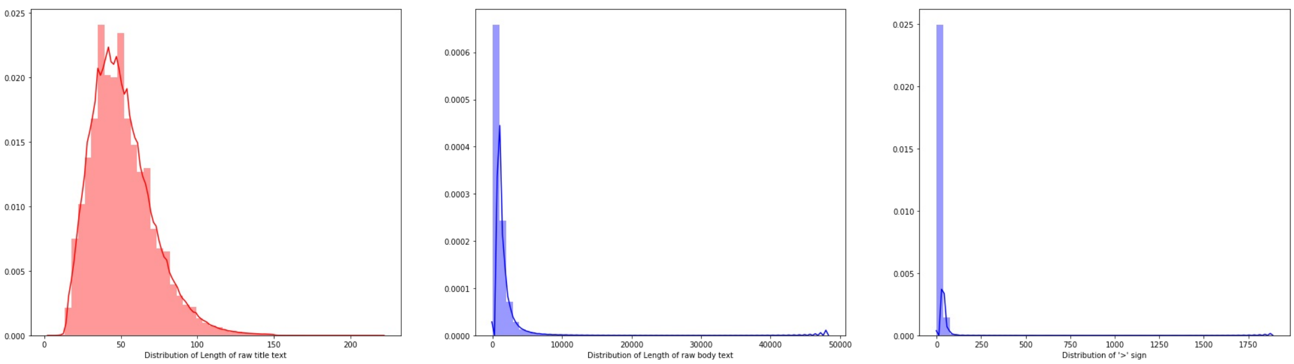
In [120]:

```
plt.figure(figsize=(12, 8))
myHist = plt.hist(basic_feats['Length_Body'].values, 100, density=True, cumulative=True)
plt.title('PDF and CDF for distribution of length of RAW text')
plt.xlabel('Length of body texts')
plt.ylabel('Probability scores.')
sns.kdeplot(basic_feats['Length_Body']);
plt.show()
```



In [119]:

```
#Draw only PDF
plt.figure(figsize=(30, 8))
plt.subplot(1,3,1)
sns.distplot([basic_feats['Length_Title']], color = 'red', axlabel="Distribution of Length of raw title text")
plt.subplot(1,3,2)
sns.distplot([basic_feats['Length_Body']], color = 'blue', axlabel="Distribution of Length of raw body text")
plt.subplot(1,3,3)
sns.distplot([basic_feats['count_Body_grtsign']], color = 'blue', axlabel="Distribution of '>' sign")
plt.show()
```



Observations:

1. We can see most of the title texts has median length around 50.
2. The median length of the body texts are around 750.
3. Each user, on an average gives 12 HTML tags.
4. All the distributions are mostly left skewed.
5. From the CDF, we can tell that almost 99% of the questions have title length less than 100, and almost 99% of the questions has body length less than 2000 words.

3.2 Analysis of Tags

3.2.1 Total number of unique tags

In [15]:

```
#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 o
ur 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 [16]:

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

```
Number of data points : 4206314
Number of unique tags : 42048
```

In [17]:

```
#'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-store']
```

3.2.3 Number of times a tag appeared

In [18]:

```
#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 #axis=0 for columns. Column contain the number of times the tags have occured
result = dict(zip(tags, freqs))
```

In [3]:

```
#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[3]:

	Tags	Counts
0	.a	18
1	.app	37
2	.asp.net-mvc	1
3	.aspxauth	21
4	.bash-profile	138

In [7]:

```
#Sort the tags according to their number of occurences.
#We see that c#, java, php, javascript and android are the 5 most frequently occuring tags.
tag_df_sorted = tag_df.sort_values(['Counts'], ascending=False)
tag_counts = tag_df_sorted['Counts'].values
tag_df_sorted.head()
```

Out[7]:

	Tags	Counts
4337	c#	331505
18069	java	299414
27249	php	284103
18157	javascript	265423
1234	android	235436

In [21]:

```
#tag_counts contains how many times each tags appeared.
tag_counts
```

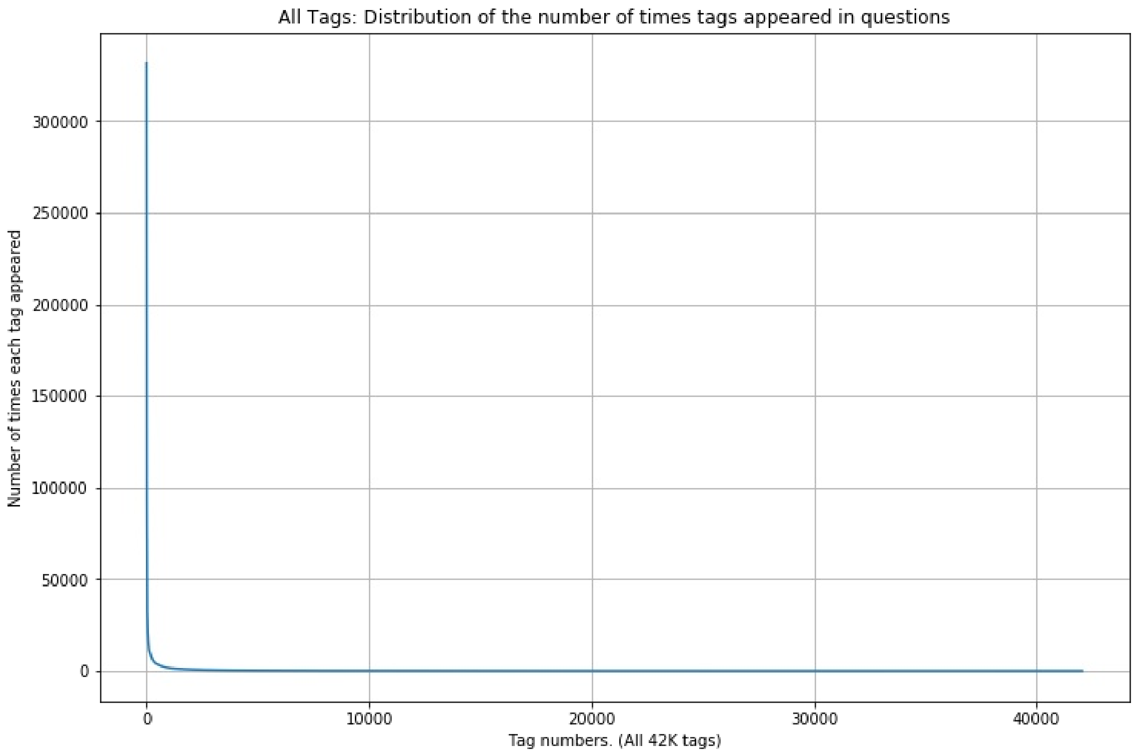
Out[21]:

```
array([331505, 299414, 284103, ...,      1,      1,      1])
```

Analysis of Tags : Distribution of all 42K tags, i.e the number of times each tag appeared in questions.

In [27]:

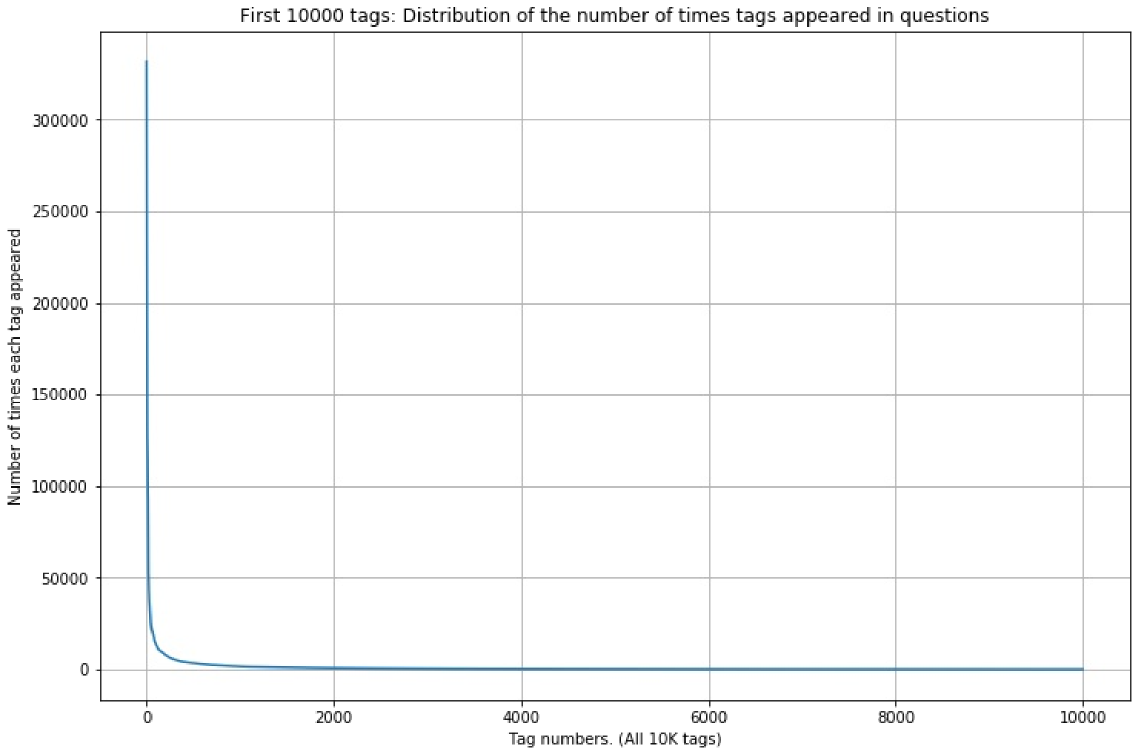
```
plt.figure(figsize=(12, 8))
plt.plot(tag_counts)
plt.title("All Tags: Distribution of the number of times tags appeared in questions")
plt.grid()
plt.xlabel("Tag numbers. (All 42K tags)")
plt.ylabel("Number of times each tag appeared")
plt.show()
```



Analysis of Tags : Zooming in. Distribution of all first 10K tags, i.e the number of times each tag appeared in questions.

In [29]:

```
plt.figure(figsize=(12, 8))
plt.plot(tag_counts[0:10000])
plt.title('First 10000 tags: Distribution of the number of times tags appeared in questions')
plt.grid()
plt.xlabel("Tag numbers. (All 10K tags)")
plt.ylabel("Number of times each tag appeared")
plt.show()
print("Frequency of occurence of top 400 tags\n")
print(len(tag_counts[0:10000:25]), tag_counts[0:10000:25])
```

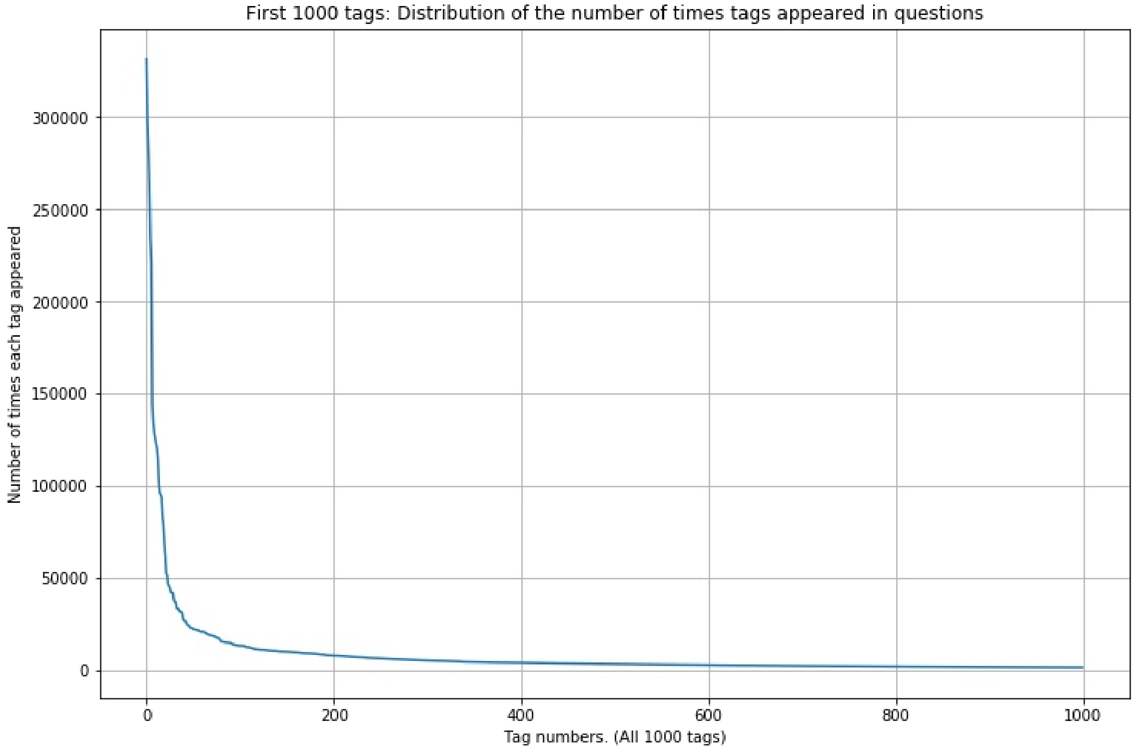



```
Frequency of occurence of top 400 tags

400 [331505  44829  22429  17728  13364  11162  10029   9148   8054   7151
   6466   5865   5370   4983   4526   4281   4144   3929   3750   3593
   3453   3299   3123   2989   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]
```

Analysis of Tags : Zooming in. Distribution of all first 1000 tags, i.e the number of times each tag appeared in questions.

In [30]:

```
plt.figure(figsize=(12, 8))
plt.plot(tag_counts[0:1000])
plt.title('First 1000 tags: Distribution of the number of times tags appeared in questions')
plt.grid()
plt.xlabel("Tag numbers. (All 1000 tags)")
plt.ylabel("Number of times each tag appeared")
plt.show()
print("Frequency of occurence of top 200 tags\n")
print(len(tag_counts[0:1000:5]), tag_counts[0:1000:5])
```

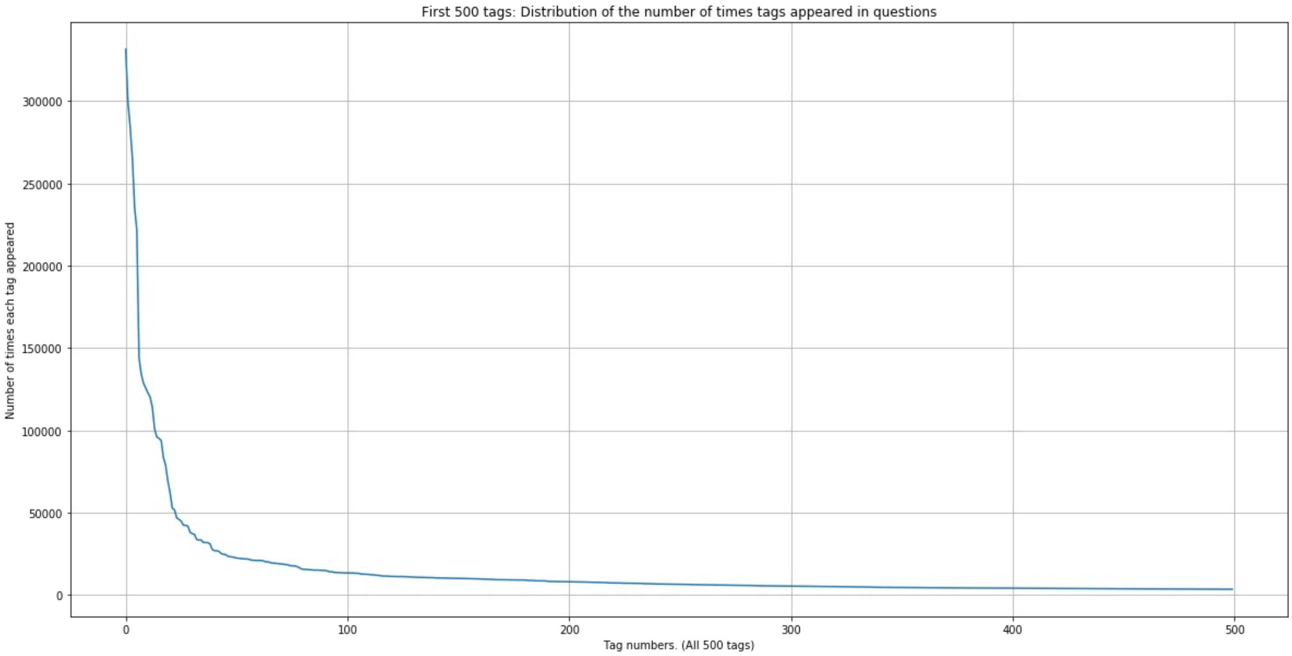



```
Frequency of occurence of top 200 tags

200 [331505 221533 122769  95160  62023  44829  37170  31897  26925  24537
  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   2989   2984   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]
```

Analysis of Tags : Zooming in. Distribution of all first 500 tags, i.e the number of times each tag appeared in questions.

In [31]:

```
plt.figure(figsize=(12, 8))
plt.plot(tag_counts[0:500])
plt.title('First 500 tags: Distribution of the number of times tags appeared in questions')
plt.grid()
plt.xlabel("Tag numbers. (All 500 tags)")
plt.ylabel("Number of times each tag appeared")
plt.show()
print("Frequency of occurence of 100 tags\n")
print(len(tag_counts[0:500:5]), tag_counts[0:500:5])
```

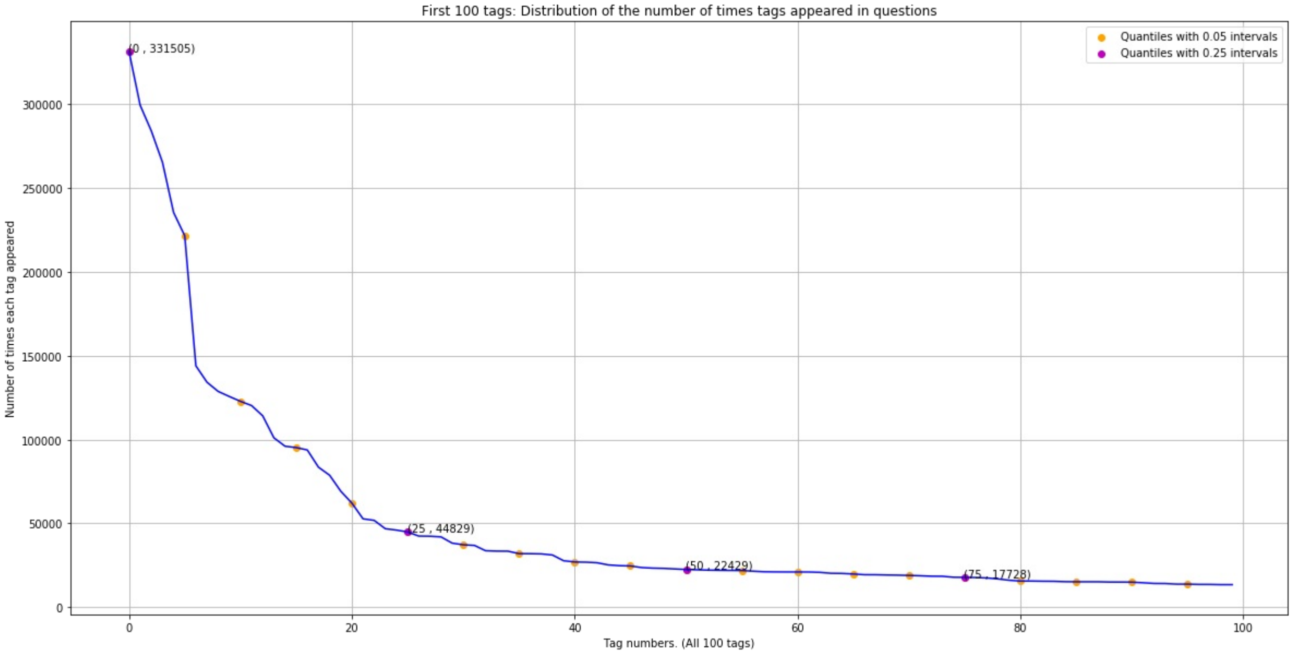



```
Frequency of occurence of 100 tags

100 [331505 221533 122769  95160  62023  44829  37170  31897  26925  24537
  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]
```

Analysis of Tags : Distribution of all first 100 tags, i.e the number of times each tag appeared in questions.

In [32]:

```
plt.figure(figsize=(12, 8))
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") #qu
antiles with 0.25 difference
plt.scatter(x=list(range(0,100,25)), y=tag_counts[0:100:25], c='m', label = "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 the number of times tags appeared in questions')
plt.grid()
plt.xlabel("Tag numbers. (All 100 tags)")
plt.ylabel("Number of times each 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  24537
  22429  21820  20957  19758  18905  17728  15533  15097  14884  13703]
```

In [36]:

```
#Store tags greater than 10K in one list
list_tags_grt_thn_10k = tag_df_sorted[tag_df_sorted.Counts>10000].Tags
#Print the length of the list
print ('{} Tags are used more than 10000 times'.format(len(list_tags_grt_thn_10k)))

# Store tags greater than 100K in one list
list_tags_grt_thn_100k = tag_df_sorted[tag_df_sorted.Counts>100000].Tags
#Print the length of the list.
print ('{} Tags are used more than 100000 times'.format(len(list_tags_grt_thn_100k)))

#Tags with the most frequency
print("Most frequently occuring tag: {}".format(tag_df_sorted.iloc[0][0]))
print("Number of times {} occurs: {}".format(tag_df_sorted.iloc[0][0],tag_counts[0]))
```

```
153 Tags are used more than 10000 times
14 Tags are used more than 100000 times
Most frequently occuring tag: c#
Number of times c# occurs: 331505
```

In [37]:

```
# 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 that 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 that are used more than 100000 times'.format(len(lst_tags_gt_100k)))
```

```
153 Tags that are used more than 10000 times
14 Tags that 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 [38]:

```
#Storing the count of tag in each question in list 'tag_count'
tag_quest_count = tag_dtm.sum(axis=1).tolist()
#Converting each value in the 'tag_quest_count' to integer.
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 4206314 datapoints.
[3, 4, 2, 2, 3]
```

In [39]:

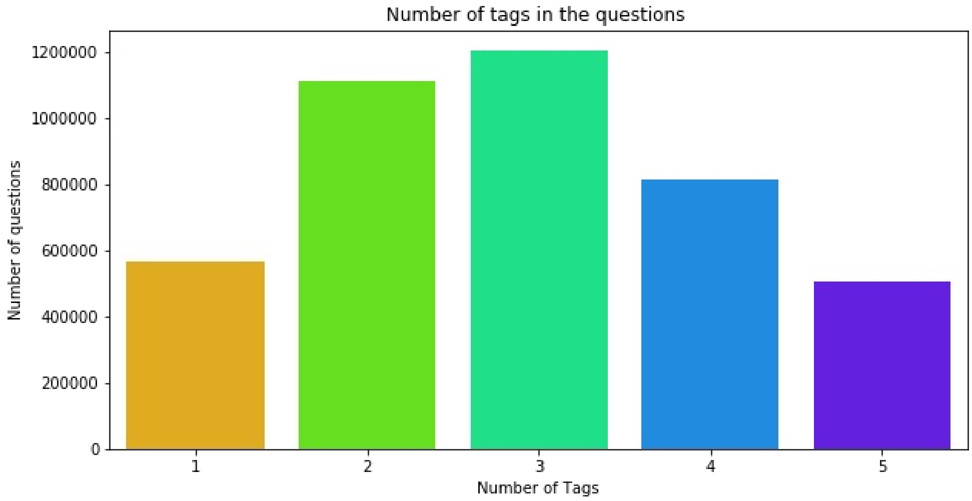
```
print( "Maximum number of tags per question: %d"%max(tag_quest_count))
print( "Minimum number of tags per question: %d"%min(tag_quest_count))
print( "Avg. number of tags per question: %f"% ((sum(tag_quest_count)*1.0)/len(tag_quest_count)))
```

```
Maximum number of tags per question: 5
Minimum number of tags per question: 1
Avg. number of tags per question: 2.899440
```

Histogram for distribution of tags.

In [40]:

```
plt.figure(figsize=(10,5))
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()
```



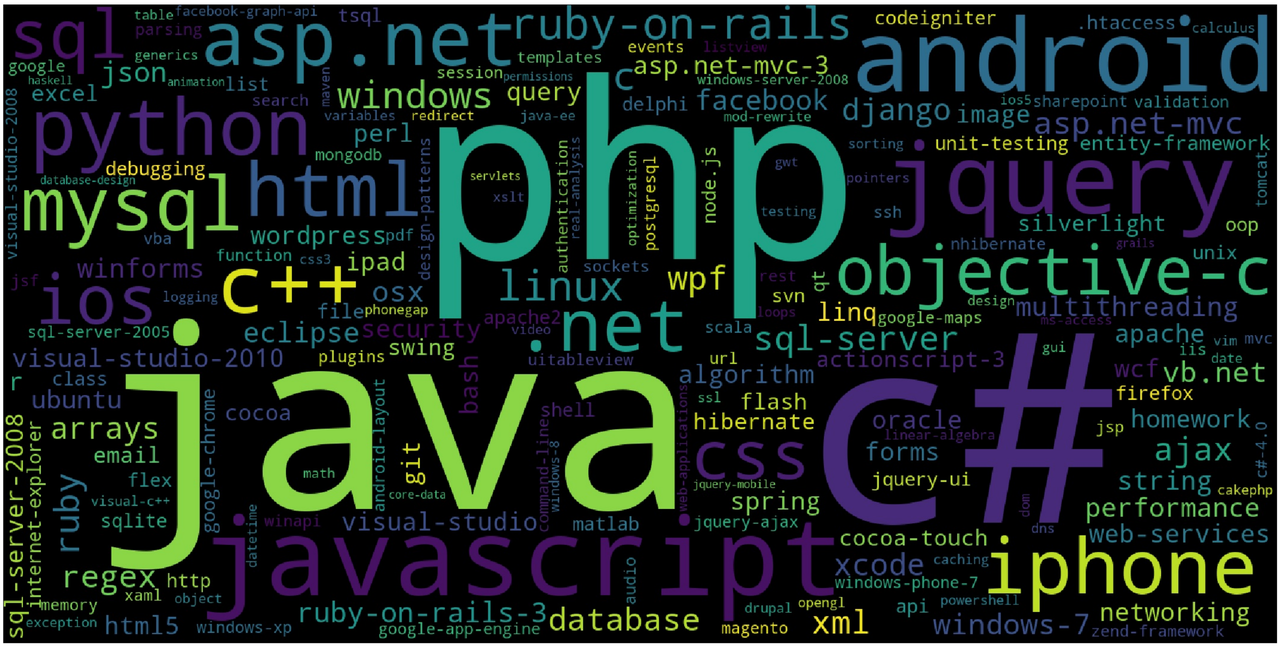
Observations from the above analysis.

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, and a vast majority of questions also has 4 tags. 5 and 1 tags are there in comparitively lower number of questions.

3.2.5 Most Frequent Tags

In [41]:

```
# 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 : 0:00:05.515045

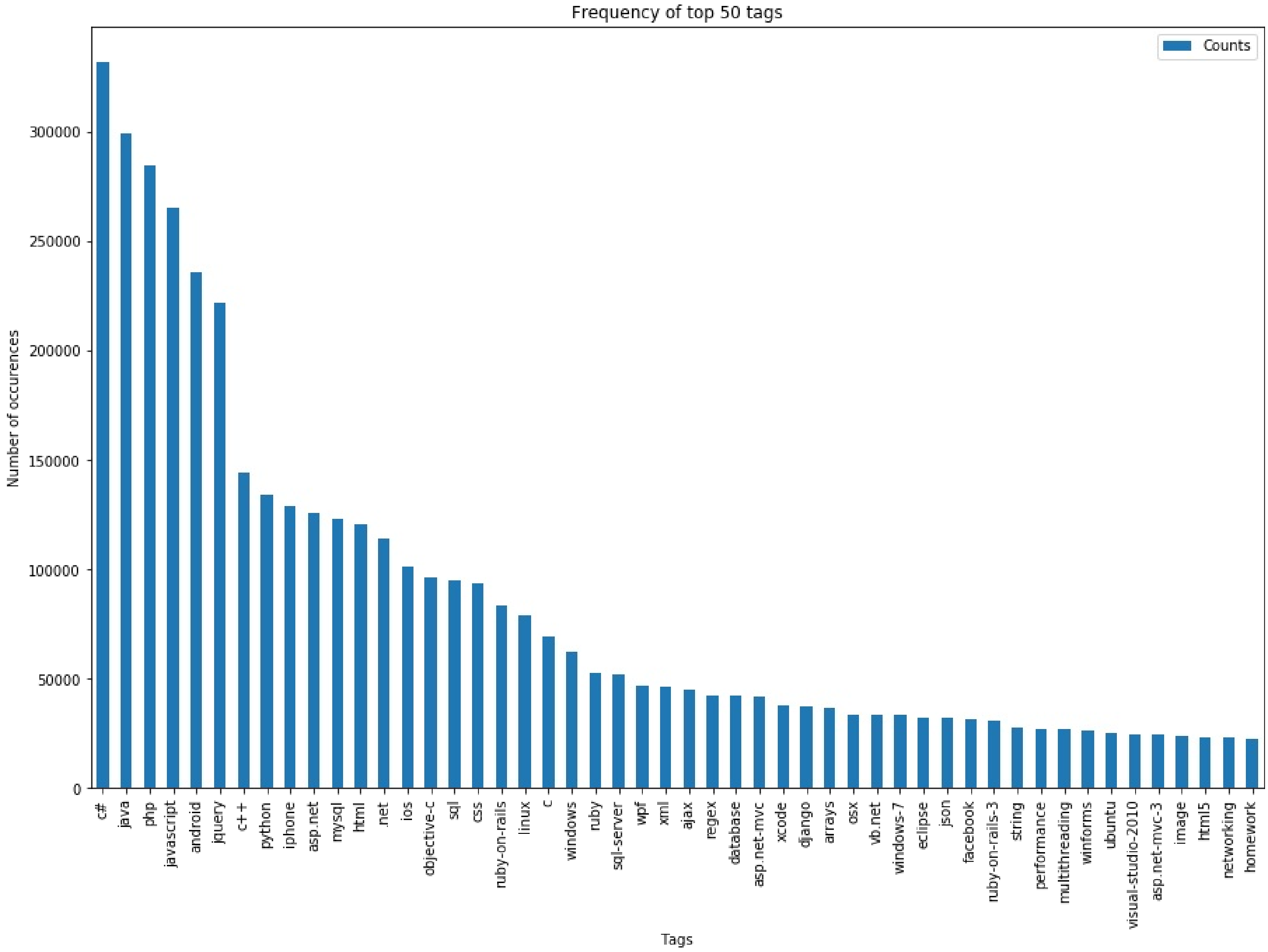
Observations from the above word cloud.

A look at the word cloud shows that "c#", "java", "php", "asp.net", "javascript", "c++" are some of the most frequent tags, closely followed by "python", "jquery", "html". This suggests c# is the most talked about programming language in StackOverflow. In terms of mobile operating systems, we see that "android" is the most popular platform closely followed by "iphone" and "ios". People have asked more questions about "linux" than about "windows".

3.2.6 The top 50 tags

In [8]:

```
i=np.arange(50)
tag_df_sorted.head(50).plot(kind='bar', figsize=(15,10), rot=90)
plt.title('Frequency of top 50 tags')
plt.xticks(i, tag_df_sorted['Tags'])
plt.xlabel('Tags')
plt.ylabel('Number of occurences')
plt.show()
```



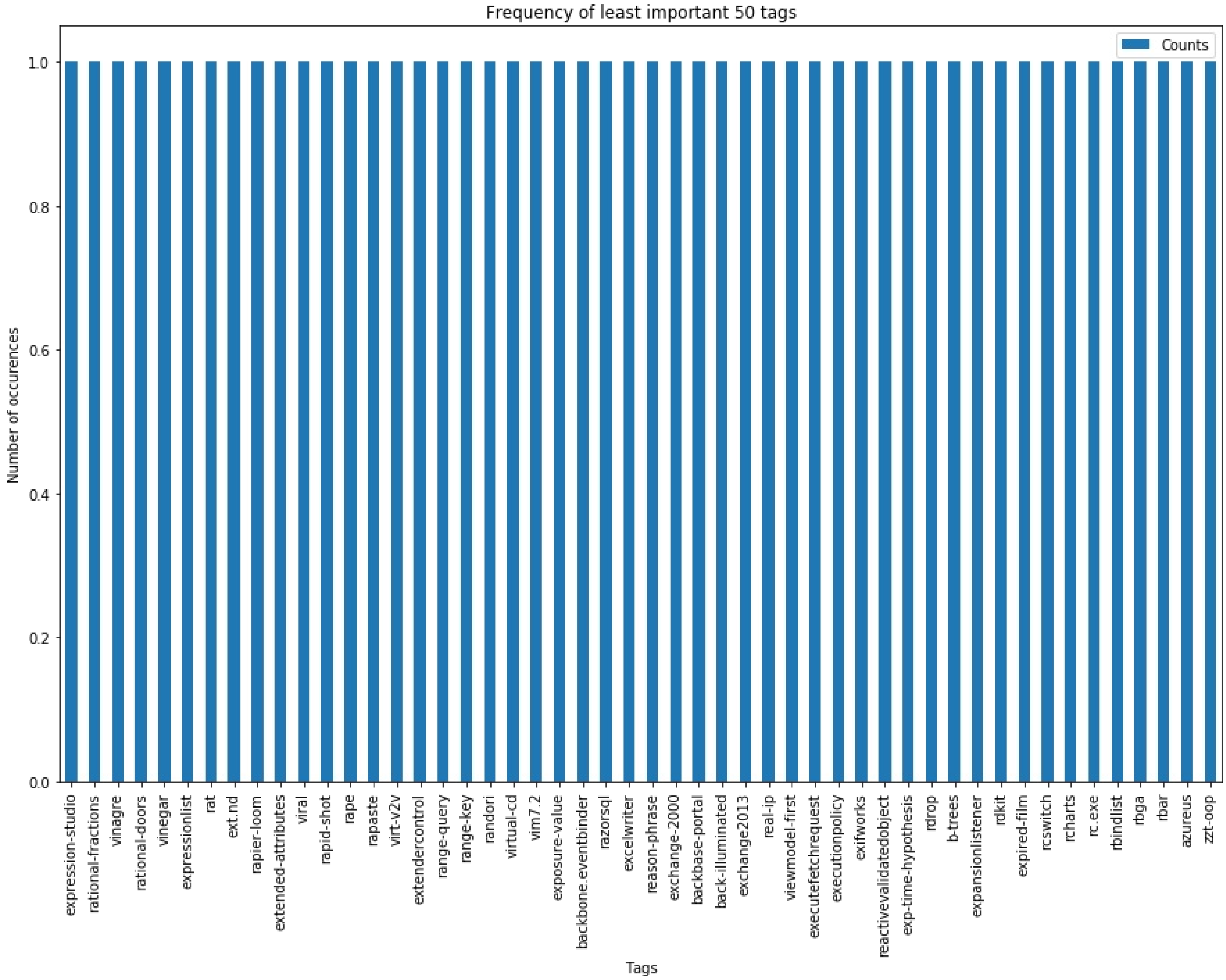
Observations from the above plot.

1. Majority of the most frequent tags are programming language.
2. C# is the top most frequent programming language, followed by java, php, javascript.
3. Android, IOS, Linux and Windows are among the top most frequent operating systems.
4. MySQL and SQL-Server are the most popular databases.

3.2.7 The least fruquent 50 tags

In [8]:

```
i=np.arange(50)
tag_df_sorted.tail(50).plot(kind='bar', figsize=(15,10), rot=90)
plt.title('Frequency of least important 50 tags')
plt.xticks(i, tag_df_sorted['Tags'][-50:])
plt.xlabel('Tags')
plt.ylabel('Number of occurences')
plt.show()
```



In [18]:

```
#These are the least 50 important tags
print(list(tag_df_sorted['Tags'][-50:]))
```

```
['expression-studio', 'rational-fractions', 'vinagre', 'rational-doors', 'vinegar', 'expressionlist'
, 'rat', 'ext.nd', 'rapier-loom', 'extended-attributes', 'viral', 'rapid-shot', 'rape', 'rapaste', '
virt-v2v', 'extendercontrol', 'range-query', 'range-key', 'randori', 'virtual-cd', 'vim7.2', 'exposu
re-value', 'backbone.eventbinder', 'razorsql', 'excelwriter', 'reason-phrase', 'exchange-2000', 'bac
kbase-portal', 'back-illuminated', 'exchange2013', 'real-ip', 'viewmodel-first', 'executefetchreques
t', 'executionpolicy', 'exifworks', 'reactivevalidatedobject', 'exp-time-hypothesis', 'rdrop', 'b-tr
ees', 'expansionlistener', 'rdkit', 'expired-film', 'rcswitch', 'rcharts', 'rc.exe', 'rbindlist', 'r
bga', 'rbar', 'azureus', 'zzt-oop']
```

3.2.8 EDA using K-Means Clustering on BOW representations of tags

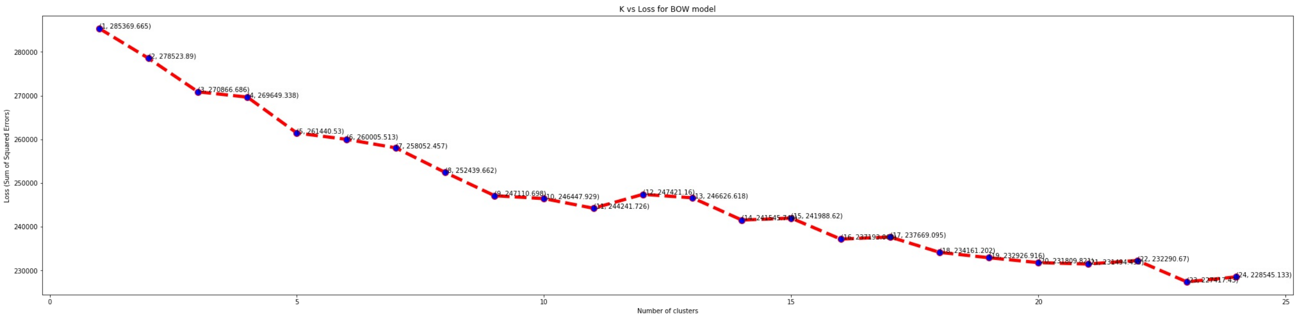
In [38]:

```
from sklearn.cluster import KMeans

#Elbow method to determine the best value of K in K-Means clustering.
def plot_elbow(sumOfSquaredErrors, n_clusters, vectorizationType):
    '''This function is used to plot the elbow curve for sum of squared errors vs cluster values and obtain the o
ptimal
    value of the hyperparameter K.'''

    k_values = n_clusters
    loss = sumOfSquaredErrors

    #Plot K_Values vs Loss Values
    plt.figure(figsize=(35,8))
    plt.plot(k_values,loss,color='red',linestyle='dashed',linewidth=5,marker='o',markerfacecolor='blue',markersiz
```

```
    plt.plot(k_values,loss,color='red',linestyle='dashed',linewidth=3,marker='o',markerfacecolor='blue',markersiz
e=10)
    for xy in zip(k_values, np.round(loss,3)):
        plt.annotate('(%s, %s)' % xy, xy=xy, textcoords='data')
    plt.title('K vs Loss for {} model'.format(vectorizationType))
    plt.xlabel('Number of clusters')
    plt.ylabel('Loss (Sum of Squared Errors)')
    plt.show()

    optimal_k = input("Please select the optimal numberof clusters from the above elbow plot and press enter : ")
    print("The optimal number of clusters selected from the elbow method is {}".format(optimal_k))

    return optimal_k

#Function to perform KMeans Clustering.
def KMeansPlusPlus(tags_vectors):
    '''This function is used for multiple method calls which would determine the optimal value of k. The loss is
calculated for each clusters and the value of the optimal
    number of clusters is obtained by visualy examining the elbow plot. At the end the k-means algorithm will be
run with the best value of K selected from the elbow plot'''
    t_start = datetime.now()
    sumOfSquaredErrors = []
    n_clusters = range(1,25)
    k_means = [KMeans(n_clusters=i, n_init=5, init='k-means++', n_jobs=8, random_state=0) for i in n_clusters] #
algorithm = elkan for dense data data, default: algorithm = auto
    k_means_centroids = [k_mean.fit(tags_vectors) for k_mean in k_means]
    sumOfSquaredErrors = [k_mean.inertia_ for k_mean in k_means_centroids] # Inertia: Sum of distances of samples
to their closest cluster center
    optimal_k = int(plot_elbow(sumOfSquaredErrors, n_clusters, "BOW"))

    #Run k-medoids with the optimal number of clusters obtained from the elbow method
    kmeans = KMeans(n_clusters=optimal_k, init='k-means++', algorithm='auto', n_jobs=8, random_state=0).fit(tags_
vectors)
    print("Time taken to perform K-Means clustering on Tags data: ",datetime.now() - t_start)

    return kmeans, optimal_k

#Function to draw word clouds for each clusters.
from wordcloud import WordCloud
def word_clouds(kmeans_object, tags_corpus):
    #Labels of each data point
    labels=kmeans_object.labels_
    clusters_dict = {i: np.where(labels == i)[0] for i in range(optimal_k)}
    # Transform this dictionary into list (if you need a list as result)
    clusters_list = []
    print("The number of datapoints in each cluster are as follows : ")
    for key, value in clusters_dict.items():
        temp = [key,value]
        clusters_list.append(temp)
        print("Cluster = {}, Number of data points = {}".format(key+1,len(value)))

    from wordcloud import WordCloud
    for cluster_number in range(optimal_k):
        cluster = [clusters_dict[cluster_number][i] for i in range(clusters_dict[cluster_number].size)]

        reviews_cluster = []
        for i in cluster:
            reviews_cluster.append(tags_corpus[i])

        review_corpus = ""
        for review in reviews_cluster:
            review_corpus = review_corpus + " " + review

        # lower max_font_size
        wordcloud = WordCloud(width=800, height=450, margin=2, prefer_horizontal=0.9, scale=1, max_words=75,
                              min_font_size=4, random_state=42, background_color='black',
                              contour_color='black', repeat=False).generate(str(review_corpus))
        plt.figure(figsize=(16,9))
        plt.title("Word Cloud for Cluster {}".format(cluster_number+1))
        plt.imshow(wordcloud, interpolation="bilinear")
        plt.axis("off")
        plt.show()
```

In [39]:

```python
#Taking 100000K data points sampled randomly from tags_data. Not taking all the datapoints
data = tag_data.sample(n=100000, random_state=0).reset_index().drop(columns='index')
tags_corpus=data['Tags'].apply(lambda x: str(x)) #Avoid encoding problems
cv_object = CountVectorizer(dtype='float',tokenizer = lambda x: x.split()).fit(tags_corpus) #Initializing the BOW constructor
tags_vectors = cv_object.transform(tags_corpus) #Creating BOW vectors of all the tags
#tags_vectors = standardize(tags_bow, False) #Column Standardization of the Bag of Words vector
kmeans_object, optimal_k = KMeansPlusPlus(tags_vectors) #KMeans++ Algorithm function call to get the best kmeans object and optimal number of clusters
```



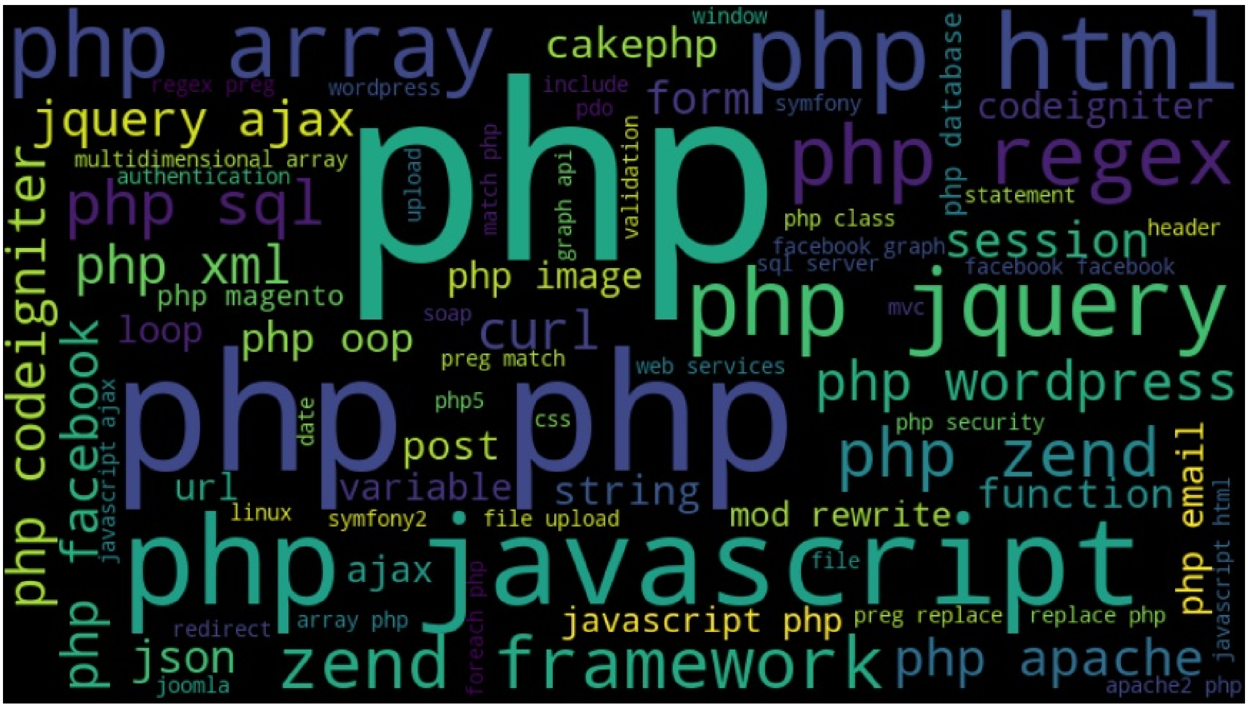

```
Please select the optimal numberof clusters from the above elbow plot and press enter : 10
The optimal number of clusters selected from the elbow method is 10
Time taken to perform K-Means clustering on Tags data:  0:33:44.131575
```

In [40]:

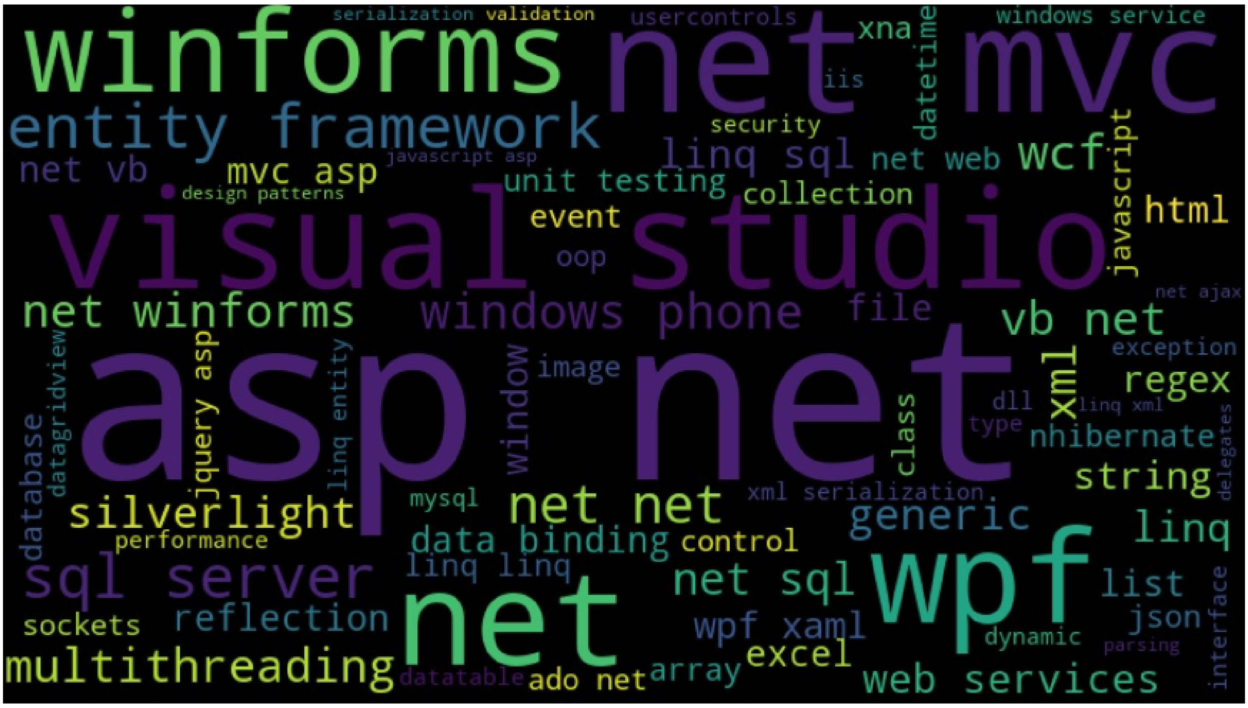
```python
#Plot word clouds of similar tags
word_clouds(kmeans_object, tags_corpus)
```

```
The number of datapoints in each cluster are as follows :
Cluster = 1, Number of data points = 5470
Cluster = 2, Number of data points = 7699
Cluster = 3, Number of data points = 6130
Cluster = 4, Number of data points = 56729
Cluster = 5, Number of data points = 4620
Cluster = 6, Number of data points = 5563
Cluster = 7, Number of data points = 2794
Cluster = 8, Number of data points = 458
Cluster = 9, Number of data points = 8502
Cluster = 10, Number of data points = 2035
```

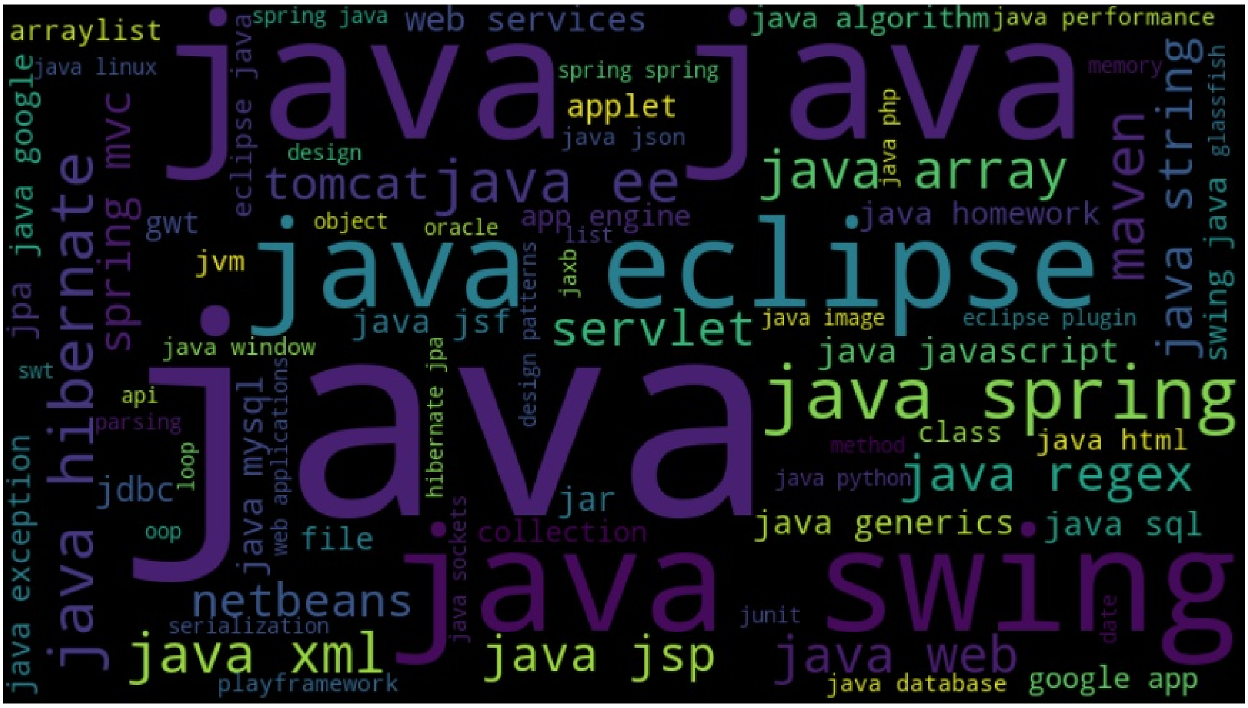
Word Cloud for Cluster 1



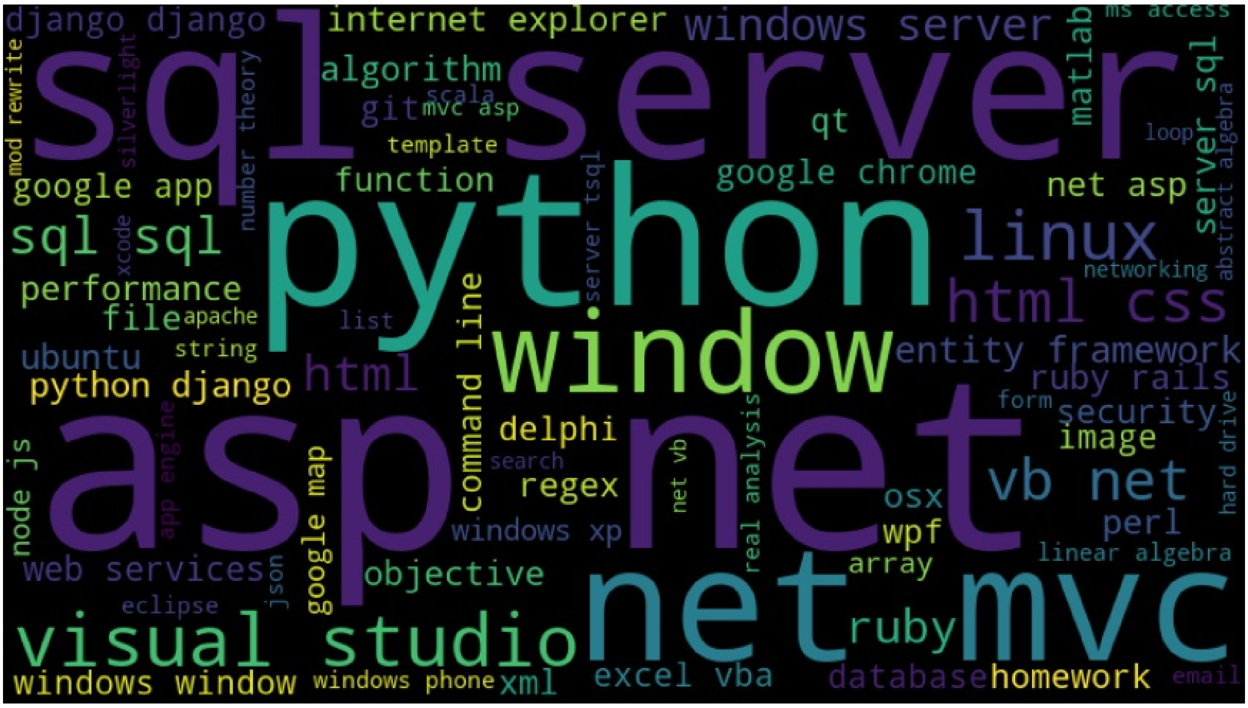
Word Cloud for Cluster 2



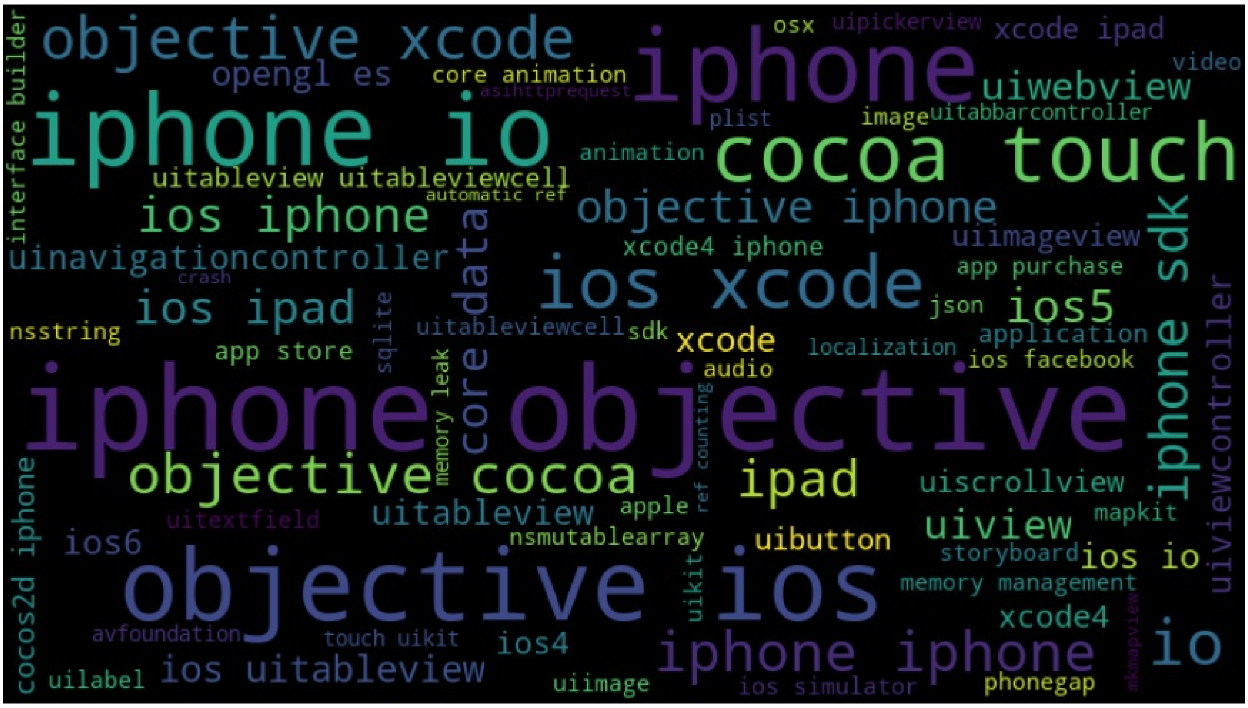
Word Cloud for Cluster 3



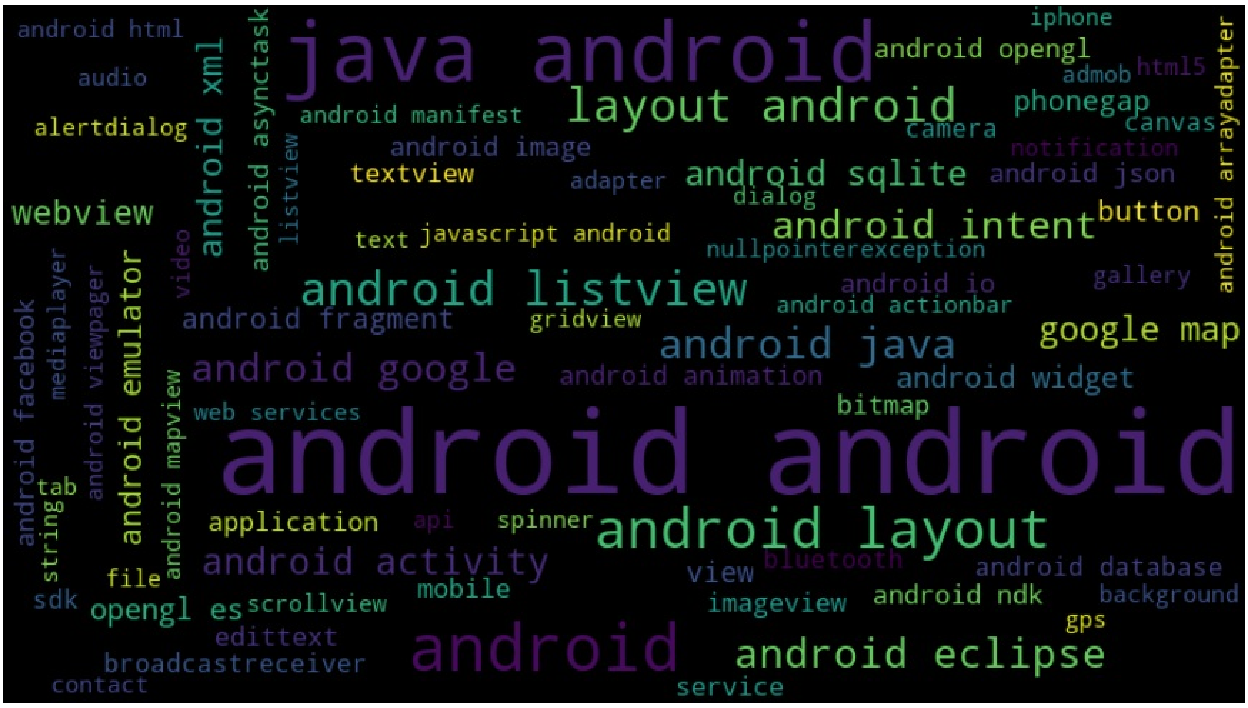
Word Cloud for Cluster 4



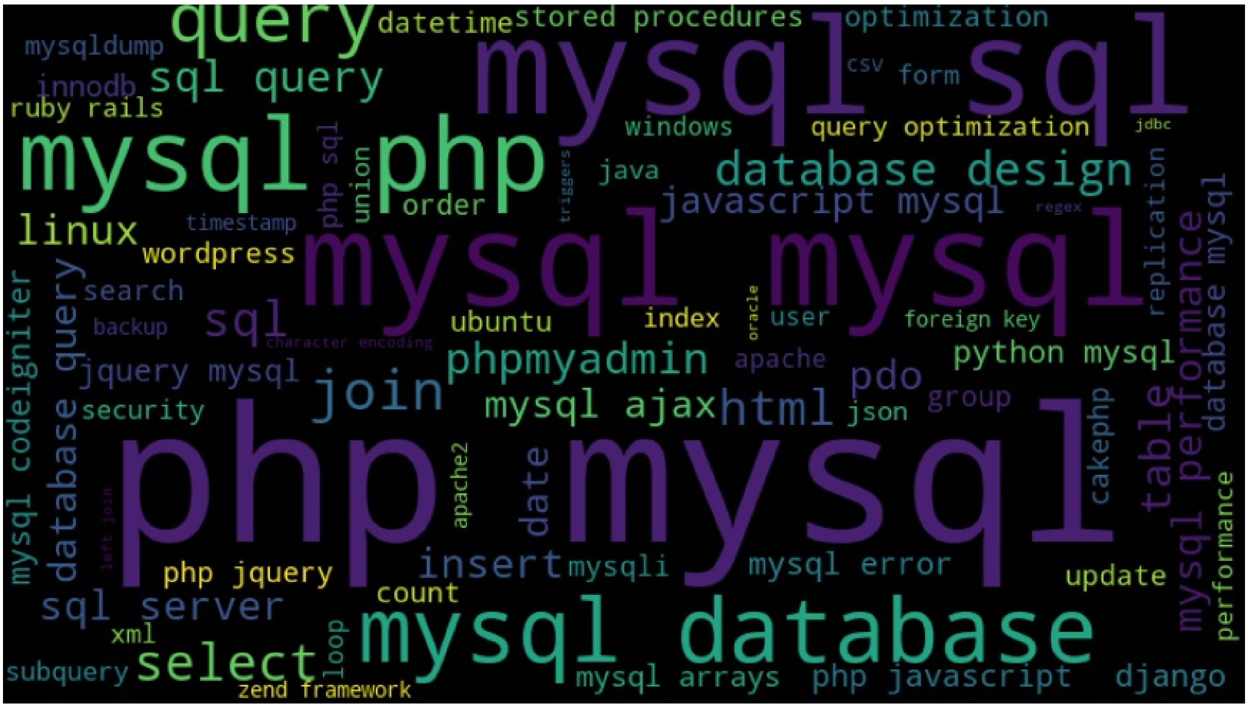
Word Cloud for Cluster 5



Word Cloud for Cluster 6



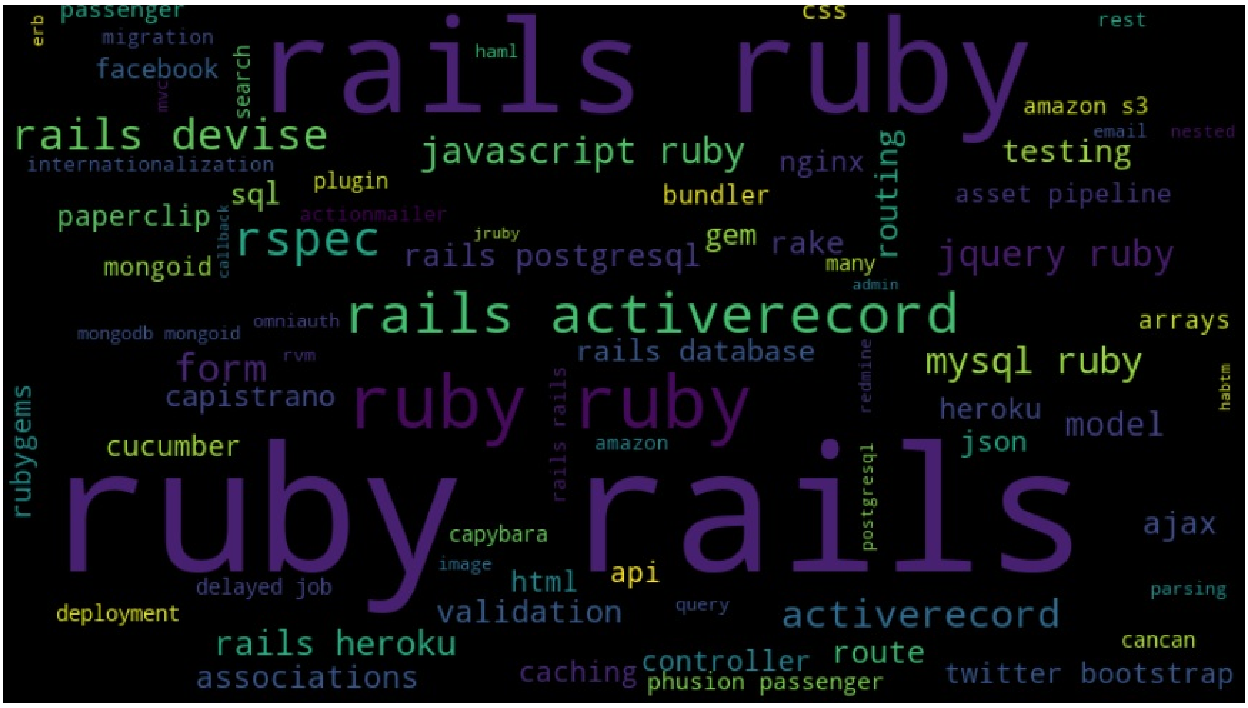
Word Cloud for Cluster 7



Word Cloud for Cluster 8

Word Cloud for Cluster 9

Word Cloud for Cluster 10



Analysis from the tags clusters.

In Cluster 1, we see a lot of tags related to the PHP language. We can also see the word facebook, which is logical considering the fact that facebook is built using PHP. So this has clustered a lot of questions on PHP.

In Cluster 2, we see most of the tags belonging to visual studio and it's related frameworks like asp .net, mvc framworks etc.

In Cluster 3, we see a lot of questions about Java and it's various frameworks - like spring, hybernate, swing etc. So this cluster mostly contains questions on Java.

In Cluster 4, This has grouped questions mostly on sql server. How to use SQL server from various programming languages. We see the tags asp and net occuring together alamost always. The tags python and django occurs together most of the times.

In Cluster 5, we see a lot of tags related to IOS development. This cluster has essentially grouped questions on developing IOS application, problems related to Iphone. We can also see this cluster contaisn objective C as a frequent tag. This seems logical as ios apps are mostly written in objective c.

In Cluster 6, it mostly has tags related to android application development using java and java ide's like eclipse etc.

In Cluster 7, PHP occurs a lot with mysql. This is logical, since PHP related questins are azsked ery often in StackOverflow (as we have seen above), and most of the questions has queries related to using sql database in PHP.

In Cluster 8, multithreading occurs a lot of times with Java. Cluster 8 has grouped tags related to parallelization using Java. I have inferred this by seeing the following words - multithreading, semaphore, asynchronous, multiprocessing etc. These tags have a hi=gh tendency to occur together.

In Cluster 9, most of the tags belongs to frontend languages. We can see tags like html, jquery, javascript, chrome, node, ajax, various frontent plugins etc. All these tags has a very high tendency to occur together.

In Cluster 10, we see most of the tags belonging to ruby on rails. There are a few tags which occurs together with ruby on rails. Few of them are - rails, activeboard, hereku, rails devise, ruby, mongodb etc.

In general, we see that there are some tags which has a higher tendency to occur with some specific sets of tags. For example, if someone has tagged a question as android, there is a higher likelihood that it might also contain the tag java. Similarly, if one has tagged a question as IOS, there is a higher chance that the tags objective c might be present as well.

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 [47]:

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

In [3]:

```
#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 (question text NOT NULL, code text, tags text
, words_pre integer, words_post integer, is_code integer);"""
create_database_table("Processed.db", sql_create_table)
```

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

In [ ]:

```
# 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 ORDER BY RANDOM() LIMIT 1000000;")

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)
```

We will create a new data base to store the sampled and preprocessed questions __

In [0]:

```
#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], row[2] #question=body

    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')

    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 stop_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,words_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( "\nAverage length of questions(Title+Body) before processing: %d"%no_dup_avg_len_pre)
print( "Average length of questions(Title+Body) after processing: %d"%no_dup_avg_len_post)
print("Percentage 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
number of questions completed= 200000
number of questions completed= 300000
number of questions completed= 400000
number of questions completed= 500000
number of questions completed= 600000
number of questions completed= 700000
number of questions completed= 800000
number of questions completed= 900000
Avg. length of questions(Title+Body) before processing: 1169
Avg. length of questions(Title+Body) after processing: 327
Percent of questions containing code: 57
Time taken to run this cell : 0:47:05.946582
```

In [0]:

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

In [0]:

```
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 10")
        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
====================================================================================================
('ef code first defin one mani relationship differ key troubl defin one zero mani relationship entit
i ef object model look like use fluent api object composit pk defin batch id batch detail id use flu
ent api object composit pk defin batch detail id compani id map exist databas tpt basic idea submitt
edtransact zero mani submittedsplittransact associ navig realli need one way submittedtransact submi
ttedsplittransact need dbcontext class onmodelcr overrid map class lazi load occur submittedtransact
submittedsplittransact help would much appreci edit taken advic made follow chang dbcontext class ad
follow onmodelcr overrid must miss someth get follow except thrown submittedtransact key batch id ba
tch detail id zero one mani submittedsplittransact key batch detail id compani id rather assum conve
nt creat relationship two object configur requir sinc obvious wrong',)
----------------------------------------------------------------------------------------------------
('explan new statement review section c code came accross statement block come accross new oper use
way someon explain new call way',)
----------------------------------------------------------------------------------------------------
('error function notat function solv logic riddl iloczyni list structur list possibl candid solut li
st possibl coordin matrix wan na choos one candid compar possibl candid element equal wan na delet c
oordin call function skasuj look like ni knowledg haskel cant see what wrong',)
----------------------------------------------------------------------------------------------------
('step plan move one isp anoth one work busi plan switch isp realli soon need chang lot inform dns w
an wan wifi question guy help mayb peopl plan correct chang current isp new one first dns know recei
v new ip isp major chang need take consider exchang server owa vpn two site link wireless connect km
away citrix server vmware exchang domain control link place import server crucial step inform need k
now avoid downtim busi regard ndavid',)
----------------------------------------------------------------------------------------------------
('use ef migrat creat databas googl migrat tutori af first run applic creat databas ef enabl migrat
way creat databas migrat rune applic tri',)
----------------------------------------------------------------------------------------------------
('magento unit test problem magento site recent look way check integr magento site given point unit
test jump one method would assum would big job write whole lot test check everyth site work anyon in
volv unit test magento advis follow possibl test whole site custom modul nis exampl test would amaz
given site heavili link databas would nbe possibl fulli test site without disturb databas better way
automaticlli check integr magento site say integr realli mean fault site ship payment etc work corre
ct',)
----------------------------------------------------------------------------------------------------
('find network devic without bonjour write mac applic need discov mac pcs iphon ipad connect wifi ne
twork bonjour seem reason choic turn problem mani type router mine exampl work block bonjour servic
need find ip devic tri connect applic specif port determin process run best approach accomplish task
without violat app store sandbox',)
----------------------------------------------------------------------------------------------------
('send multipl row mysql databas want send user mysql databas column user skill time nnow want abl a
dd one row user differ time etc would code send databas nthen use help schema',)
----------------------------------------------------------------------------------------------------
('insert data mysql php powerpoint event powerpoint present run continu way updat slide present auto
mat data mysql databas websit',)
----------------------------------------------------------------------------------------------------
```

In [5]:

```
#Taking 1 Million entries to a dataframe.
read_db = 'Processed.db'
if os.path.isfile(read_db):
    conn_r = create_connection(read_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 [6]:

```
preprocessed_data.head()
```

Out[6]:

	question	tags
0	chang cpu soni vaio pcg grx tri everywher find...	cpu motherboard sony-vaio replacement disassembly
1	display size grayscal qimag qt abl display ima...	c++ qt qt4
2	datagrid selecteditem set back null eventtocom...	mvvm silverlight-4.0
3	filter string collect base listview item resol...	c# winforms string listview collections
4	disabl home button without use type keyguard c...	android android-layout android-manifest androi...

In [7]:

```
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 : 999999
Number of dimensions : 2
```

4. Machine Learning Models

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 [10]:

```
# 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 [14]:

```
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))
```

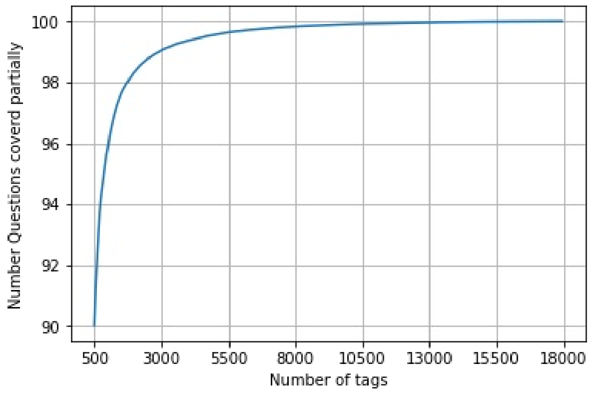
In [15]:

```
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))
```

A variance plot which shows the percentage of variance retained with different number of tags.

In [14]:

```
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, minimun is 50(it covers 90% of the tags)
print("With ",5500,"tags we are covering ",questions_explained[50],"% of questions")
```




```
With  5500 tags we are covering  99.035 % of questions
```

In [15]:

```
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 : 9645 out of  999999
```

In [16]:

```
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 : 35422
Number of tags taken : 5500 ( 15.527073570097679 %)
```

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

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

In [17]:

```
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 [18]:

```
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 : (799999, 5500)
Number of data points in test data : (200000, 5500)
```

4.3 Featurizing data

In [0]:

```
start = datetime.now()
vectorizer = TfidfVectorizer(min_df=0.00009, max_features=200000, smooth_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 : 0:09:50.460431
```

In [0]:

```
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)
```

```
Diamensions of train data X: (799999, 88244) Y : (799999, 5500)
Diamensions of test data X: (200000, 88244) Y: (200000, 5500)
```

In [0]:

```
# 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[0]:

```
"\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(accur
acy_score(y_test,predictions))\nprint(metrics.f1_score(y_test, predictions, average = 'macro'))\npri
nt(metrics.f1_score(y_test, predictions, average = 'micro'))\nprint(metrics.hamming_loss(y_test,pred
ictions))\n\n"
```

4.4 Applying Logistic Regression with OneVsRest Classifier

In [0]:

```
# this will be taking so much time try not to run it, download the lr_with_equal_weight.pkl file and use to predi
ct
# This takes about 6-7 hours to run.
classifier1 = OneVsRestclassifier1(SGDclassifier1(loss='log', alpha=0.00001, penalty='l1'), n_jobs=-1)
classifier1.fit(x_train_multilabel, y_train)
predictions = classifier1.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))
```

```
accuracy : 0.081965
macro f1 score : 0.0963020140154
micro f1 scoore : 0.374270748817
hamming loss : 0.00041225090909090907
Precision recall report :
              precision    recall  f1-score   support

          0       0.62      0.23      0.33     15760
          1       0.79      0.43      0.56     14039
          2       0.82      0.55      0.66     13446
          3       0.76      0.42      0.54     12730
```

4	0.94	0.76	0.84	11229
5	0.85	0.64	0.73	10561
6	0.70	0.30	0.42	6958
7	0.87	0.61	0.72	6309
8	0.70	0.40	0.50	6032
9	0.78	0.43	0.55	6020
10	0.86	0.62	0.72	5707
11	0.52	0.17	0.25	5723
12	0.55	0.10	0.16	5521
13	0.59	0.25	0.35	4722
14	0.61	0.22	0.32	4468
15	0.79	0.52	0.63	4536
16	0.58	0.27	0.37	4545
17	0.80	0.53	0.64	4069
18	0.61	0.24	0.35	3638
19	0.57	0.18	0.27	3218
20	0.33	0.06	0.10	3000
21	0.73	0.34	0.46	2585
22	0.59	0.29	0.38	2439
23	0.88	0.61	0.72	2199
24	0.64	0.39	0.48	2157
25	0.67	0.39	0.49	2123
26	0.86	0.65	0.74	1948
27	0.35	0.07	0.12	2027
28	0.59	0.29	0.39	2013
29	0.61	0.20	0.30	1801
30	0.48	0.24	0.32	1728
31	0.94	0.75	0.84	1725
32	0.60	0.26	0.36	1581
33	0.49	0.14	0.22	1533
34	0.81	0.33	0.47	1565
35	0.75	0.62	0.68	1568
36	0.76	0.50	0.60	1542
37	0.74	0.50	0.59	1536
38	0.37	0.12	0.19	1524
39	0.40	0.12	0.19	1345
40	0.65	0.38	0.48	1292
41	0.41	0.11	0.17	1264
42	0.69	0.25	0.37	1265
43	0.59	0.29	0.38	1171
44	0.41	0.15	0.22	1173
45	0.38	0.10	0.16	1137
46	0.62	0.12	0.20	1125
47	0.26	0.07	0.11	1116
48	0.44	0.15	0.22	1042
49	0.40	0.02	0.03	1096
50	0.63	0.38	0.48	1031
51	0.47	0.14	0.22	1033
52	0.87	0.68	0.76	1042
53	0.32	0.09	0.14	1027
54	0.53	0.14	0.22	1063
55	0.63	0.34	0.44	1048
56	0.78	0.42	0.54	1054
57	0.91	0.77	0.83	1058
58	0.37	0.10	0.16	1000
59	0.26	0.03	0.05	973
60	0.76	0.42	0.54	978
61	0.74	0.43	0.54	977
62	0.27	0.06	0.10	957
63	0.81	0.22	0.34	958
64	0.88	0.63	0.73	944
65	0.76	0.49	0.60	923
66	0.67	0.36	0.47	959
67	0.55	0.15	0.24	951
68	0.38	0.13	0.20	924
69	0.71	0.25	0.37	897
70	0.78	0.47	0.59	900
71	0.82	0.40	0.54	893
72	0.21	0.01	0.01	836
73	0.74	0.16	0.26	850
74	0.58	0.37	0.45	838
75	0.88	0.64	0.74	855
76	0.47	0.28	0.35	837
77	0.68	0.41	0.52	824
78	0.14	0.01	0.01	793
79	0.34	0.09	0.14	751
80	0.31	0.08	0.13	793
81	0.71	0.33	0.45	758
82	0.60	0.28	0.38	764
83	0.82	0.59	0.69	710
84	0.82	0.48	0.61	734
85	0.79	0.42	0.55	723
86	0.44	0.23	0.30	708

87	0.93	0.58	0.72	714
88	0.91	0.53	0.67	683
89	0.58	0.20	0.30	711
90	0.71	0.42	0.53	699
91	0.44	0.03	0.06	725
92	0.71	0.47	0.57	676
93	0.47	0.10	0.16	672
94	0.66	0.40	0.50	645
95	0.86	0.66	0.75	691
96	0.57	0.09	0.15	664
97	0.91	0.59	0.72	633
98	0.64	0.38	0.48	615
99	0.53	0.19	0.29	667
100	0.89	0.71	0.79	656
101	0.22	0.03	0.05	648
102	0.64	0.13	0.22	654
103	0.92	0.63	0.75	653
104	0.87	0.52	0.65	656
105	0.20	0.02	0.04	607
106	0.68	0.34	0.45	635
107	0.23	0.03	0.05	594
108	0.40	0.18	0.25	592
109	0.32	0.07	0.12	604
110	0.46	0.21	0.29	606
111	0.70	0.39	0.50	567
112	0.68	0.27	0.38	571
113	0.61	0.36	0.45	578
114	0.47	0.18	0.26	564
115	0.35	0.13	0.19	537
116	0.93	0.66	0.77	583
117	0.59	0.09	0.15	534
118	0.66	0.35	0.46	566
119	0.20	0.04	0.07	567
120	0.48	0.16	0.24	497
121	0.55	0.19	0.29	536
122	0.24	0.05	0.08	528
123	0.81	0.53	0.64	550
124	0.50	0.21	0.29	563
125	0.35	0.06	0.10	545
126	0.49	0.18	0.27	544
127	0.95	0.76	0.84	549
128	0.63	0.34	0.44	495
129	0.94	0.59	0.73	509
130	0.34	0.11	0.16	501
131	0.28	0.04	0.07	524
132	0.48	0.26	0.34	485
133	0.55	0.37	0.45	515
134	0.32	0.04	0.08	536
135	0.77	0.38	0.51	526
136	0.67	0.34	0.45	493
137	0.40	0.08	0.14	501
138	0.31	0.05	0.09	501
139	0.29	0.02	0.04	523
140	0.88	0.64	0.74	508
141	0.33	0.11	0.16	490
142	0.77	0.50	0.60	482
143	0.49	0.25	0.33	461
144	0.74	0.48	0.58	496
145	0.62	0.17	0.26	521
146	0.39	0.13	0.19	481
147	0.00	0.00	0.00	486
148	0.37	0.09	0.14	497
149	0.54	0.09	0.16	470
150	0.37	0.11	0.17	459
151	0.74	0.45	0.56	464
152	0.50	0.24	0.32	482
153	0.46	0.09	0.15	507
154	0.29	0.04	0.07	503
155	0.90	0.59	0.71	456
156	0.50	0.27	0.35	480
157	0.54	0.26	0.35	443
158	0.92	0.70	0.80	457
159	0.57	0.08	0.13	478
160	0.16	0.03	0.05	470
161	0.37	0.18	0.24	468
162	0.24	0.05	0.09	428
163	0.40	0.08	0.13	462
164	0.73	0.32	0.45	493
165	0.93	0.68	0.79	437
166	0.40	0.20	0.26	435
167	0.30	0.02	0.03	448
168	0.53	0.16	0.25	436
169	0.36	0.10	0.15	437

170	0.38	0.09	0.15	410
171	0.59	0.32	0.41	450
172	0.69	0.39	0.50	435
173	0.91	0.67	0.77	427
174	0.45	0.16	0.24	427
175	0.43	0.17	0.24	424
176	0.64	0.43	0.52	410
177	0.67	0.29	0.40	426
178	0.74	0.49	0.59	459
179	0.52	0.13	0.20	433
180	0.71	0.36	0.48	452
181	0.91	0.62	0.74	427
182	0.46	0.13	0.20	410
183	0.28	0.02	0.04	404
184	0.69	0.42	0.52	406
185	0.68	0.41	0.52	411
186	0.22	0.02	0.03	394
187	0.90	0.65	0.75	414
188	0.64	0.10	0.18	430
189	0.16	0.04	0.06	389
190	0.28	0.03	0.05	418
191	0.36	0.16	0.22	371
192	0.83	0.57	0.68	363
193	0.91	0.55	0.69	389
194	0.44	0.04	0.07	411
195	0.49	0.22	0.31	383
196	0.95	0.74	0.83	423
197	0.91	0.54	0.68	378
198	0.69	0.38	0.49	382
199	0.12	0.01	0.02	344
200	0.71	0.31	0.44	383
201	0.77	0.34	0.47	390
202	0.18	0.02	0.04	405
203	0.43	0.07	0.11	365
204	0.42	0.14	0.21	346
205	0.21	0.05	0.08	378
206	0.67	0.27	0.39	390
207	0.33	0.07	0.11	379
208	0.39	0.11	0.17	386
209	0.42	0.15	0.22	339
210	0.27	0.07	0.12	382
211	0.37	0.05	0.08	374
212	0.62	0.38	0.47	364
213	0.94	0.76	0.84	372
214	0.96	0.63	0.76	350
215	0.76	0.38	0.50	352
216	0.00	0.00	0.00	351
217	0.64	0.29	0.40	329
218	0.72	0.31	0.44	341
219	0.94	0.71	0.81	331
220	0.49	0.27	0.35	342
221	0.76	0.39	0.52	339
222	0.29	0.04	0.06	332
223	0.43	0.12	0.18	327
224	0.31	0.06	0.11	324
225	0.51	0.21	0.30	352
226	0.65	0.30	0.41	317
227	0.54	0.12	0.20	355
228	0.57	0.19	0.29	341
229	0.58	0.37	0.46	334
230	0.64	0.49	0.56	304
231	0.43	0.04	0.07	321
232	0.77	0.50	0.61	311
233	0.32	0.10	0.15	312
234	0.09	0.01	0.02	306
235	0.03	0.00	0.01	305
236	0.16	0.02	0.04	340
237	0.58	0.30	0.40	316
238	0.65	0.23	0.34	297
239	0.35	0.13	0.19	305
240	0.73	0.44	0.55	310
241	0.67	0.36	0.47	307
242	0.58	0.16	0.25	316
243	0.26	0.07	0.11	314
244	0.51	0.12	0.19	316
245	0.67	0.46	0.55	313
246	0.79	0.46	0.58	325
247	0.60	0.36	0.45	291
248	0.33	0.01	0.02	311
249	0.57	0.24	0.33	314
250	0.38	0.05	0.09	309
251	0.30	0.08	0.13	300
252	0.55	0.27	0.36	325

253	0.76	0.51	0.61	316
254	0.43	0.09	0.15	306
255	0.54	0.19	0.28	289
256	0.49	0.11	0.18	304
257	0.16	0.02	0.04	268
258	0.85	0.58	0.69	266
259	0.06	0.00	0.01	298
260	0.55	0.36	0.43	292
261	0.25	0.05	0.08	289
262	0.50	0.01	0.01	305
263	0.00	0.00	0.00	281
264	0.59	0.25	0.35	295
265	0.16	0.02	0.04	281
266	0.83	0.52	0.64	269
267	0.45	0.12	0.19	312
268	0.75	0.40	0.52	294
269	0.34	0.05	0.09	285
270	0.56	0.33	0.42	279
271	0.50	0.28	0.36	269
272	0.59	0.38	0.46	277
273	0.69	0.31	0.43	272
274	0.36	0.01	0.03	285
275	0.94	0.69	0.80	295
276	0.46	0.19	0.27	283
277	0.65	0.29	0.40	250
278	0.57	0.20	0.30	281
279	0.86	0.58	0.69	270
280	0.62	0.35	0.44	272
281	0.32	0.07	0.11	278
282	0.00	0.00	0.00	264
283	0.85	0.59	0.70	281
284	0.78	0.53	0.63	261
285	0.33	0.09	0.14	283
286	0.00	0.00	0.00	275
287	0.29	0.03	0.05	274
288	0.37	0.04	0.06	284
289	0.00	0.00	0.00	260
290	0.54	0.24	0.34	245
291	0.07	0.00	0.01	267
292	0.33	0.07	0.11	263
293	0.30	0.09	0.14	268
294	0.33	0.11	0.16	270
295	0.48	0.06	0.10	261
296	0.84	0.59	0.69	240
297	0.43	0.22	0.29	250
298	0.81	0.51	0.63	245
299	0.11	0.01	0.01	283
300	0.51	0.21	0.30	236
301	0.78	0.51	0.62	267
302	0.19	0.02	0.04	243
303	0.26	0.04	0.06	276
304	0.89	0.71	0.79	280
305	0.37	0.14	0.20	249
306	0.24	0.02	0.04	258
307	0.00	0.00	0.00	262
308	0.53	0.20	0.29	248
309	0.58	0.25	0.35	244
310	0.33	0.06	0.09	254
311	0.41	0.10	0.16	263
312	0.52	0.25	0.33	232
313	0.75	0.55	0.63	235
314	0.61	0.11	0.19	248
315	0.49	0.16	0.25	263
316	0.33	0.08	0.12	264
317	0.61	0.06	0.12	216
318	0.05	0.00	0.01	230
319	0.53	0.27	0.36	230
320	0.00	0.00	0.00	239
321	0.45	0.08	0.13	265
322	0.69	0.32	0.44	253
323	0.23	0.04	0.06	238
324	0.72	0.37	0.49	232
325	0.22	0.05	0.08	239
326	0.49	0.18	0.26	261
327	0.64	0.14	0.23	261
328	0.67	0.47	0.55	231
329	0.46	0.13	0.20	264
330	0.18	0.02	0.03	242
331	0.80	0.37	0.50	231
332	0.63	0.28	0.39	234
333	0.50	0.32	0.39	212
334	0.26	0.05	0.09	221
335	0.15	0.03	0.05	242

336	0.57	0.30	0.40	211
337	0.20	0.01	0.03	212
338	0.00	0.00	0.00	222
339	0.22	0.02	0.04	227
340	0.66	0.30	0.41	216
341	0.57	0.26	0.36	231
342	0.45	0.22	0.29	233
343	0.17	0.03	0.04	232
344	0.28	0.02	0.04	209
345	0.37	0.11	0.17	216
346	0.27	0.09	0.13	222
347	0.48	0.19	0.28	243
348	0.51	0.26	0.35	222
349	0.57	0.12	0.20	228
350	0.44	0.12	0.18	205
351	0.58	0.30	0.39	177
352	0.77	0.39	0.52	234
353	0.96	0.57	0.71	230
354	0.47	0.21	0.29	195
355	0.90	0.42	0.57	209
356	0.06	0.00	0.01	205
357	0.50	0.11	0.18	211
358	0.43	0.16	0.23	230
359	0.27	0.08	0.12	211
360	0.39	0.09	0.14	221
361	0.24	0.04	0.08	200
362	0.82	0.15	0.25	219
363	0.36	0.07	0.12	222
364	0.62	0.27	0.38	213
365	0.94	0.36	0.52	199
366	0.80	0.37	0.51	200
367	0.76	0.29	0.42	199
368	0.57	0.26	0.36	212
369	0.93	0.71	0.80	214
370	0.10	0.02	0.03	197
371	0.20	0.03	0.05	212
372	0.41	0.14	0.21	210
373	0.43	0.03	0.05	211
374	0.41	0.15	0.22	213
375	0.00	0.00	0.00	216
376	0.87	0.53	0.66	195
377	0.95	0.67	0.79	187
378	0.15	0.03	0.04	191
379	0.17	0.02	0.04	178
380	0.79	0.48	0.60	193
381	0.13	0.02	0.04	187
382	0.67	0.03	0.06	193
383	0.17	0.04	0.06	204
384	0.28	0.15	0.19	193
385	0.12	0.02	0.04	207
386	0.84	0.45	0.59	211
387	0.06	0.00	0.01	210
388	0.31	0.04	0.06	223
389	0.24	0.09	0.13	203
390	0.72	0.24	0.36	199
391	0.40	0.08	0.13	200
392	0.22	0.05	0.09	183
393	0.62	0.31	0.41	189
394	0.96	0.66	0.78	194
395	0.53	0.18	0.27	183
396	0.43	0.21	0.28	189
397	0.71	0.34	0.46	191
398	0.34	0.06	0.11	206
399	0.33	0.01	0.03	221
400	0.28	0.04	0.07	196
401	0.28	0.09	0.14	179
402	0.28	0.08	0.12	187
403	0.51	0.22	0.31	203
404	0.46	0.12	0.19	205
405	0.35	0.08	0.13	218
406	0.19	0.04	0.06	196
407	0.72	0.35	0.47	206
408	0.31	0.06	0.10	203
409	0.70	0.43	0.53	187
410	0.85	0.54	0.66	208
411	0.83	0.45	0.58	193
412	0.33	0.02	0.03	192
413	0.66	0.36	0.46	182
414	0.45	0.19	0.27	175
415	0.64	0.49	0.55	181
416	0.00	0.00	0.00	202
417	0.92	0.44	0.60	202
418	0.17	0.01	0.02	195

419	0.78	0.25	0.38	177
420	0.26	0.07	0.11	168
421	0.80	0.45	0.58	187
422	0.92	0.46	0.62	209
423	0.66	0.16	0.26	177
424	0.35	0.06	0.10	182
425	0.52	0.14	0.23	187
426	0.22	0.04	0.07	185
427	0.43	0.13	0.20	185
428	0.42	0.18	0.25	185
429	0.92	0.46	0.61	175
430	0.90	0.49	0.64	190
431	0.31	0.03	0.05	185
432	0.71	0.03	0.05	189
433	0.60	0.20	0.30	184
434	0.79	0.36	0.49	200
435	0.20	0.01	0.01	167
436	0.21	0.01	0.03	209
437	0.50	0.07	0.12	200
438	0.29	0.09	0.14	169
439	0.44	0.15	0.23	170
440	0.25	0.04	0.07	182
441	0.62	0.34	0.44	156
442	0.20	0.02	0.03	170
443	0.00	0.00	0.00	189
444	0.00	0.00	0.00	172
445	0.33	0.11	0.16	180
446	0.21	0.06	0.10	175
447	0.48	0.12	0.19	187
448	0.00	0.00	0.00	170
449	0.41	0.24	0.30	170
450	0.35	0.10	0.16	176
451	0.62	0.15	0.24	194
452	0.61	0.31	0.41	175
453	0.19	0.04	0.07	187
454	0.11	0.01	0.01	181
455	0.62	0.14	0.23	177
456	0.50	0.18	0.26	170
457	0.24	0.03	0.05	182
458	0.68	0.37	0.48	172
459	0.00	0.00	0.00	190
460	0.43	0.16	0.23	183
461	0.94	0.63	0.75	182
462	0.35	0.16	0.22	173
463	0.91	0.69	0.79	171
464	0.58	0.27	0.37	173
465	0.77	0.41	0.53	184
466	0.72	0.22	0.34	175
467	0.43	0.19	0.26	162
468	0.12	0.01	0.02	176
469	0.91	0.46	0.61	177
470	0.52	0.07	0.13	167
471	0.27	0.06	0.10	192
472	0.50	0.32	0.39	168
473	0.32	0.05	0.09	188
474	0.31	0.05	0.08	163
475	0.44	0.17	0.24	160
476	0.89	0.56	0.69	180
477	0.92	0.46	0.61	182
478	0.49	0.27	0.35	171
479	0.57	0.18	0.27	174
480	0.96	0.52	0.68	162
481	0.21	0.04	0.06	169
482	0.33	0.03	0.06	157
483	0.77	0.48	0.59	200
484	0.58	0.21	0.31	177
485	0.51	0.26	0.34	175
486	0.64	0.51	0.57	185
487	0.96	0.52	0.67	167
488	0.00	0.00	0.00	192
489	0.30	0.09	0.14	176
490	0.00	0.00	0.00	167
491	0.33	0.01	0.01	177
492	0.47	0.26	0.33	160
493	0.46	0.22	0.30	159
494	0.15	0.03	0.04	159
495	0.31	0.10	0.15	162
496	0.82	0.46	0.59	167
497	0.17	0.02	0.03	168
498	0.40	0.12	0.19	154
499	0.00	0.00	0.00	184
500	0.14	0.03	0.05	167
501	0.41	0.20	0.27	153

502	0.78	0.55	0.65	143
503	0.22	0.07	0.10	177
504	0.69	0.32	0.44	177
505	0.90	0.50	0.64	152
506	0.80	0.40	0.54	179
507	0.60	0.12	0.20	171
508	0.61	0.28	0.39	151
509	0.51	0.23	0.32	162
510	0.63	0.24	0.35	158
511	0.18	0.03	0.05	164
512	0.00	0.00	0.00	149
513	0.78	0.60	0.68	174
514	0.51	0.15	0.23	172
515	0.34	0.14	0.20	144
516	0.57	0.15	0.23	164
517	0.88	0.67	0.76	152
518	0.60	0.02	0.03	175
519	0.29	0.04	0.06	168
520	0.52	0.11	0.18	145
521	0.89	0.38	0.53	165
522	0.91	0.55	0.69	151
523	0.93	0.57	0.71	171
524	0.89	0.53	0.66	160
525	0.59	0.41	0.49	139
526	0.57	0.19	0.29	165
527	0.57	0.22	0.31	148
528	0.64	0.21	0.32	178
529	0.31	0.06	0.10	152
530	0.11	0.01	0.01	143
531	0.57	0.20	0.30	174
532	0.63	0.20	0.30	135
533	0.35	0.05	0.09	179
534	0.26	0.04	0.08	135
535	0.29	0.09	0.14	157
536	0.88	0.53	0.66	163
537	0.79	0.39	0.53	127
538	0.34	0.13	0.19	130
539	0.55	0.20	0.29	155
540	0.43	0.18	0.25	165
541	0.35	0.11	0.16	139
542	0.38	0.05	0.09	159
543	0.44	0.18	0.25	140
544	0.76	0.17	0.28	143
545	0.44	0.12	0.19	147
546	0.47	0.18	0.26	153
547	0.76	0.28	0.41	165
548	0.35	0.10	0.16	149
549	0.62	0.26	0.37	123
550	0.82	0.06	0.11	148
551	0.68	0.41	0.51	145
552	0.50	0.04	0.07	157
553	0.46	0.23	0.31	151
554	0.50	0.01	0.01	152
555	0.43	0.17	0.24	147
556	0.72	0.35	0.47	143
557	0.47	0.20	0.28	139
558	0.92	0.54	0.68	165
559	0.37	0.10	0.16	147
560	0.27	0.13	0.17	139
561	0.29	0.08	0.12	152
562	0.45	0.26	0.33	132
563	0.41	0.17	0.24	150
564	0.30	0.08	0.13	165
565	0.73	0.38	0.50	147
566	0.27	0.05	0.08	151
567	0.52	0.24	0.33	153
568	0.48	0.19	0.27	148
569	0.17	0.04	0.06	142
570	0.11	0.02	0.04	140
571	0.07	0.01	0.01	149
572	1.00	0.02	0.04	146
573	0.51	0.29	0.37	135
574	0.73	0.24	0.36	137
575	0.50	0.11	0.18	142
576	0.24	0.10	0.14	145
577	0.82	0.25	0.38	145
578	0.72	0.33	0.45	131
579	0.40	0.15	0.22	142
580	0.00	0.00	0.00	143
581	0.38	0.09	0.15	139
582	0.57	0.15	0.24	150
583	0.00	0.00	0.00	121
584	0.57	0.28	0.38	148

585	0.61	0.41	0.49	134
586	0.64	0.37	0.47	151
587	0.74	0.11	0.20	150
588	0.48	0.11	0.18	141
589	0.20	0.03	0.05	137
590	0.79	0.36	0.50	154
591	0.52	0.22	0.31	126
592	0.85	0.49	0.62	144
593	0.29	0.06	0.10	130
594	0.46	0.15	0.22	148
595	0.13	0.02	0.03	115
596	0.64	0.46	0.53	142
597	0.95	0.46	0.62	123
598	0.63	0.21	0.32	150
599	0.00	0.00	0.00	134
600	0.24	0.04	0.07	154
601	0.36	0.08	0.14	165
602	0.50	0.02	0.04	150
603	0.49	0.15	0.23	137
604	0.89	0.53	0.67	133
605	0.38	0.14	0.21	146
606	0.88	0.12	0.21	129
607	0.17	0.03	0.05	151
608	0.86	0.55	0.67	138
609	0.36	0.13	0.19	124
610	0.40	0.01	0.03	144
611	0.00	0.00	0.00	150
612	0.00	0.00	0.00	130
613	0.21	0.05	0.08	127
614	0.41	0.17	0.24	141
615	0.10	0.02	0.03	133
616	0.54	0.29	0.38	132
617	0.67	0.02	0.03	131
618	0.21	0.03	0.06	125
619	0.63	0.37	0.46	123
620	0.00	0.00	0.00	148
621	0.12	0.01	0.02	117
622	0.72	0.47	0.57	129
623	0.36	0.04	0.06	113
624	0.88	0.51	0.64	110
625	0.92	0.63	0.75	121
626	0.22	0.08	0.12	125
627	0.95	0.59	0.73	132
628	0.67	0.30	0.42	116
629	0.81	0.38	0.52	126
630	0.29	0.04	0.07	126
631	0.28	0.06	0.10	148
632	0.91	0.61	0.74	140
633	0.50	0.02	0.03	128
634	0.40	0.16	0.22	128
635	0.00	0.00	0.00	140
636	0.95	0.41	0.57	130
637	0.62	0.23	0.34	126
638	0.75	0.08	0.15	143
639	0.67	0.31	0.42	121
640	0.16	0.04	0.07	117
641	0.36	0.12	0.19	112
642	0.46	0.14	0.21	137
643	0.96	0.61	0.74	141
644	0.71	0.37	0.49	127
645	0.28	0.06	0.10	128
646	0.10	0.01	0.01	124
647	0.11	0.03	0.05	138
648	0.13	0.03	0.04	119
649	0.00	0.00	0.00	137
650	0.33	0.01	0.02	121
651	0.07	0.02	0.03	108
652	0.72	0.41	0.52	122
653	0.61	0.26	0.36	139
654	0.40	0.02	0.03	112
655	0.53	0.14	0.22	125
656	0.64	0.19	0.29	124
657	0.30	0.08	0.12	117
658	0.50	0.20	0.28	116
659	0.37	0.08	0.14	130
660	0.15	0.02	0.03	121
661	0.75	0.35	0.48	124
662	0.48	0.12	0.19	121
663	0.84	0.63	0.72	126
664	0.00	0.00	0.00	118
665	0.18	0.06	0.09	113
666	0.00	0.00	0.00	128
667	0.53	0.12	0.20	139

668	0.29	0.04	0.07	131
669	0.26	0.05	0.08	127
670	0.47	0.07	0.12	125
671	0.33	0.02	0.03	111
672	0.55	0.37	0.44	127
673	0.72	0.48	0.57	130
674	0.19	0.02	0.04	130
675	0.60	0.20	0.30	126
676	0.15	0.02	0.03	104
677	0.53	0.14	0.22	127
678	0.57	0.15	0.24	130
679	0.26	0.10	0.14	112
680	0.43	0.09	0.15	131
681	0.00	0.00	0.00	140
682	0.53	0.35	0.42	114
683	0.78	0.12	0.22	112
684	0.35	0.06	0.10	115
685	0.66	0.15	0.24	128
686	0.57	0.10	0.17	122
687	0.25	0.03	0.05	109
688	0.29	0.02	0.03	108
689	0.00	0.00	0.00	125
690	0.50	0.01	0.02	117
691	0.36	0.09	0.15	127
692	0.80	0.35	0.49	129
693	0.42	0.16	0.23	118
694	0.72	0.37	0.49	151
695	0.67	0.29	0.41	112
696	0.81	0.22	0.34	119
697	0.19	0.05	0.07	109
698	0.58	0.33	0.42	122
699	0.96	0.49	0.65	102
700	0.29	0.07	0.11	102
701	0.46	0.26	0.33	107
702	0.25	0.03	0.05	105
703	0.25	0.01	0.02	113
704	0.62	0.27	0.37	98
705	0.21	0.05	0.08	100
706	0.72	0.33	0.45	131
707	0.45	0.21	0.29	112
708	0.44	0.03	0.06	119
709	0.28	0.07	0.11	105
710	0.18	0.03	0.04	117
711	0.39	0.14	0.21	115
712	0.41	0.10	0.16	129
713	0.68	0.27	0.38	101
714	0.57	0.10	0.17	122
715	0.00	0.00	0.00	97
716	0.38	0.16	0.23	116
717	0.43	0.08	0.14	110
718	0.38	0.04	0.08	113
719	0.75	0.49	0.59	110
720	0.78	0.05	0.10	130
721	0.00	0.00	0.00	104
722	0.89	0.66	0.75	119
723	0.00	0.00	0.00	108
724	0.43	0.22	0.29	112
725	0.32	0.05	0.08	126
726	0.93	0.67	0.78	120
727	0.30	0.05	0.09	130
728	0.67	0.02	0.04	103
729	0.70	0.17	0.28	111
730	0.33	0.03	0.05	110
731	0.00	0.00	0.00	96
732	0.55	0.05	0.10	112
733	0.39	0.08	0.13	90
734	0.28	0.11	0.15	95
735	0.80	0.39	0.52	116
736	0.40	0.02	0.03	128
737	0.25	0.09	0.13	93
738	0.89	0.15	0.26	107
739	0.58	0.29	0.39	99
740	0.40	0.04	0.07	105
741	0.46	0.05	0.09	116
742	0.68	0.43	0.53	105
743	0.40	0.19	0.26	84
744	0.44	0.14	0.21	102
745	0.69	0.23	0.34	111
746	0.36	0.10	0.15	104
747	0.44	0.14	0.21	110
748	0.58	0.21	0.30	92
749	0.87	0.57	0.69	106
750	0.00	0.00	0.00	116

751	0.28	0.09	0.14	109
752	0.85	0.54	0.66	104
753	1.00	0.01	0.02	119
754	0.27	0.06	0.10	96
755	0.17	0.04	0.06	104
756	0.00	0.00	0.00	101
757	0.50	0.19	0.28	114
758	0.00	0.00	0.00	112
759	0.67	0.04	0.08	95
760	0.00	0.00	0.00	102
761	0.31	0.11	0.17	105
762	0.57	0.25	0.35	109
763	0.09	0.01	0.02	112
764	0.94	0.40	0.56	116
765	0.60	0.31	0.41	109
766	0.00	0.00	0.00	96
767	0.50	0.09	0.15	114
768	0.00	0.00	0.00	99
769	0.65	0.15	0.25	98
770	0.48	0.21	0.30	107
771	0.00	0.00	0.00	103
772	0.00	0.00	0.00	96
773	0.00	0.00	0.00	106
774	0.76	0.33	0.46	97
775	0.27	0.03	0.06	91
776	0.00	0.00	0.00	101
777	0.76	0.38	0.50	109
778	0.00	0.00	0.00	104
779	0.33	0.08	0.13	116
780	0.00	0.00	0.00	102
781	0.85	0.26	0.40	106
782	0.64	0.15	0.24	108
783	0.80	0.08	0.15	95
784	0.91	0.36	0.52	108
785	0.94	0.43	0.59	113
786	0.40	0.06	0.10	109
787	0.78	0.41	0.54	112
788	0.00	0.00	0.00	104
789	0.43	0.17	0.25	92
790	0.44	0.06	0.11	116
791	0.29	0.04	0.07	96
792	0.58	0.15	0.24	118
793	0.64	0.27	0.38	106
794	0.26	0.06	0.10	93
795	0.80	0.31	0.45	103
796	0.39	0.12	0.18	104
797	0.57	0.09	0.16	89
798	0.55	0.06	0.11	97
799	0.00	0.00	0.00	92
800	0.55	0.14	0.22	85
801	1.00	0.04	0.08	93
802	0.79	0.28	0.41	93
803	0.36	0.13	0.19	102
804	0.65	0.12	0.20	108
805	0.87	0.37	0.52	111
806	0.61	0.14	0.23	98
807	0.20	0.03	0.06	94
808	0.15	0.02	0.04	84
809	0.84	0.32	0.46	100
810	0.22	0.02	0.04	92
811	0.37	0.11	0.17	88
812	0.39	0.13	0.20	104
813	0.50	0.04	0.08	90
814	0.38	0.07	0.12	109
815	0.23	0.04	0.06	81
816	0.70	0.22	0.33	96
817	0.98	0.53	0.69	88
818	0.56	0.24	0.33	101
819	0.94	0.45	0.61	103
820	0.00	0.00	0.00	94
821	0.72	0.17	0.27	108
822	0.29	0.06	0.09	90
823	0.81	0.44	0.57	97
824	0.50	0.02	0.04	90
825	0.52	0.23	0.32	102
826	0.12	0.01	0.02	85
827	0.20	0.02	0.03	109
828	0.30	0.03	0.05	103
829	0.98	0.40	0.56	106
830	0.88	0.26	0.40	108
831	0.50	0.04	0.07	84
832	0.00	0.00	0.00	98
833	0.77	0.26	0.39	92

834	0.50	0.10	0.17	91
835	0.87	0.28	0.43	92
836	0.28	0.07	0.11	104
837	0.63	0.24	0.34	102
838	0.22	0.07	0.11	111
839	0.00	0.00	0.00	96
840	0.41	0.15	0.22	86
841	0.34	0.10	0.16	105
842	0.20	0.01	0.02	92
843	0.39	0.16	0.23	86
844	0.00	0.00	0.00	108
845	0.45	0.06	0.11	82
846	0.22	0.04	0.07	101
847	0.97	0.60	0.74	94
848	1.00	0.41	0.58	101
849	0.39	0.14	0.20	88
850	0.88	0.36	0.51	81
851	0.79	0.10	0.18	109
852	0.45	0.13	0.20	101
853	0.25	0.03	0.06	91
854	0.29	0.06	0.10	95
855	0.20	0.01	0.02	99
856	0.14	0.01	0.02	79
857	0.67	0.32	0.43	91
858	0.00	0.00	0.00	89
859	0.42	0.09	0.15	91
860	0.49	0.19	0.28	88
861	0.32	0.07	0.11	101
862	0.51	0.30	0.37	81
863	0.69	0.20	0.31	101
864	0.28	0.11	0.16	80
865	0.00	0.00	0.00	97
866	0.88	0.46	0.60	94
867	0.00	0.00	0.00	97
868	0.29	0.07	0.11	91
869	0.35	0.09	0.14	88
870	0.53	0.25	0.34	112
871	0.93	0.57	0.71	94
872	0.00	0.00	0.00	84
873	0.89	0.53	0.66	74
874	0.91	0.53	0.67	80
875	0.46	0.23	0.31	79
876	0.56	0.07	0.12	71
877	0.77	0.26	0.39	92
878	1.00	0.08	0.15	99
879	0.56	0.14	0.23	98
880	0.37	0.18	0.24	82
881	0.70	0.35	0.47	80
882	0.91	0.55	0.69	94
883	0.07	0.01	0.02	102
884	0.88	0.22	0.35	95
885	0.91	0.57	0.70	87
886	0.20	0.01	0.02	88
887	0.41	0.08	0.13	90
888	0.84	0.46	0.60	104
889	0.20	0.01	0.02	93
890	0.14	0.02	0.04	83
891	0.00	0.00	0.00	92
892	0.58	0.17	0.26	88
893	0.00	0.00	0.00	74
894	1.00	0.40	0.57	98
895	0.47	0.22	0.30	73
896	0.00	0.00	0.00	87
897	0.29	0.03	0.05	73
898	0.58	0.22	0.32	86
899	0.24	0.08	0.12	100
900	0.43	0.14	0.21	93
901	0.82	0.36	0.50	86
902	0.38	0.07	0.12	107
903	0.43	0.03	0.06	97
904	0.52	0.17	0.26	88
905	0.00	0.00	0.00	94
906	0.14	0.02	0.04	83
907	0.00	0.00	0.00	85
908	0.00	0.00	0.00	90
909	0.14	0.01	0.02	83
910	0.60	0.07	0.13	83
911	0.19	0.03	0.06	87
912	0.94	0.38	0.54	87
913	0.56	0.10	0.18	86
914	0.52	0.16	0.25	91
915	0.25	0.02	0.04	87
916	0.00	0.00	0.00	92

917	0.00	0.00	0.00	92
918	0.81	0.37	0.51	78
919	0.44	0.10	0.16	81
920	0.00	0.00	0.00	87
921	0.00	0.00	0.00	95
922	0.85	0.27	0.41	82
923	0.33	0.02	0.04	89
924	0.00	0.00	0.00	73
925	0.41	0.09	0.14	82
926	0.43	0.03	0.06	91
927	0.38	0.10	0.15	83
928	0.33	0.03	0.05	79
929	0.55	0.07	0.12	89
930	0.29	0.07	0.11	85
931	0.00	0.00	0.00	95
932	0.25	0.01	0.02	80
933	0.50	0.07	0.12	72
934	0.64	0.29	0.40	79
935	0.52	0.15	0.23	75
936	0.70	0.22	0.34	85
937	0.47	0.09	0.16	75
938	0.23	0.09	0.13	69
939	0.00	0.00	0.00	85
940	0.11	0.01	0.02	72
941	0.00	0.00	0.00	69
942	0.44	0.09	0.14	94
943	0.00	0.00	0.00	85
944	0.94	0.36	0.52	89
945	0.19	0.04	0.06	77
946	0.78	0.15	0.25	93
947	0.00	0.00	0.00	81
948	0.95	0.50	0.66	78
949	0.00	0.00	0.00	75
950	0.00	0.00	0.00	80
951	0.12	0.01	0.02	88
952	0.29	0.03	0.05	80
953	1.00	0.71	0.83	85
954	0.83	0.55	0.66	71
955	0.00	0.00	0.00	80
956	0.81	0.37	0.51	68
957	0.87	0.52	0.65	75
958	0.43	0.13	0.20	90
959	0.81	0.15	0.25	87
960	0.89	0.38	0.53	87
961	0.74	0.29	0.42	68
962	0.65	0.26	0.37	86
963	0.57	0.19	0.28	85
964	0.43	0.15	0.23	78
965	0.76	0.44	0.56	88
966	0.93	0.46	0.61	85
967	0.52	0.23	0.32	70
968	0.33	0.04	0.07	82
969	0.88	0.47	0.61	92
970	0.31	0.05	0.09	73
971	0.00	0.00	0.00	77
972	0.46	0.16	0.24	82
973	0.80	0.10	0.18	80
974	0.12	0.01	0.02	83
975	0.98	0.58	0.73	76
976	0.00	0.00	0.00	85
977	0.00	0.00	0.00	65
978	0.57	0.11	0.19	72
979	0.33	0.02	0.04	85
980	0.23	0.05	0.08	64
981	0.25	0.03	0.05	76
982	0.58	0.07	0.13	96
983	0.94	0.31	0.46	94
984	0.29	0.02	0.04	87
985	0.33	0.01	0.03	75
986	0.00	0.00	0.00	79
987	0.00	0.00	0.00	86
988	0.50	0.01	0.02	88
989	0.00	0.00	0.00	84
990	0.52	0.14	0.22	95
991	0.37	0.15	0.22	71
992	0.57	0.38	0.46	68
993	0.00	0.00	0.00	75
994	0.00	0.00	0.00	90
995	0.95	0.43	0.60	83
996	0.89	0.43	0.58	79
997	0.71	0.08	0.14	64
998	0.27	0.04	0.07	74
999	0.81	0.36	0.50	81

1000	0.00	0.00	0.00	74
1001	0.14	0.02	0.03	62
1002	0.67	0.25	0.37	71
1003	0.00	0.00	0.00	72
1004	0.50	0.08	0.14	75
1005	0.93	0.53	0.67	72
1006	0.52	0.15	0.23	81
1007	0.00	0.00	0.00	74
1008	0.17	0.01	0.03	72
1009	0.00	0.00	0.00	75
1010	0.47	0.16	0.24	91
1011	0.59	0.18	0.27	90
1012	0.62	0.25	0.36	80
1013	0.00	0.00	0.00	88
1014	0.80	0.06	0.11	71
1015	0.57	0.11	0.18	74
1016	0.88	0.22	0.35	68
1017	0.70	0.39	0.50	71
1018	0.65	0.21	0.32	80
1019	0.00	0.00	0.00	83
1020	0.46	0.08	0.14	74
1021	0.93	0.49	0.64	78
1022	0.86	0.32	0.47	77
1023	0.12	0.01	0.02	78
1024	0.68	0.31	0.43	67
1025	0.50	0.01	0.02	80
1026	0.69	0.23	0.35	77
1027	0.80	0.32	0.46	88
1028	0.24	0.06	0.09	70
1029	0.00	0.00	0.00	79
1030	0.33	0.07	0.12	67
1031	0.88	0.47	0.61	75
1032	0.56	0.28	0.38	64
1033	0.88	0.21	0.34	70
1034	0.17	0.06	0.09	69
1035	0.44	0.10	0.16	72
1036	0.30	0.04	0.07	79
1037	0.24	0.05	0.08	84
1038	0.00	0.00	0.00	87
1039	0.68	0.35	0.46	65
1040	0.72	0.36	0.48	73
1041	0.00	0.00	0.00	77
1042	0.27	0.05	0.09	77
1043	0.16	0.07	0.09	60
1044	0.00	0.00	0.00	73
1045	0.00	0.00	0.00	67
1046	0.43	0.04	0.07	83
1047	1.00	0.40	0.57	70
1048	1.00	0.02	0.03	65
1049	0.62	0.14	0.22	74
1050	0.50	0.02	0.03	62
1051	0.58	0.16	0.25	70
1052	0.00	0.00	0.00	69
1053	0.25	0.08	0.12	72
1054	0.44	0.15	0.23	72
1055	0.90	0.52	0.66	73
1056	0.74	0.34	0.46	92
1057	0.67	0.05	0.10	73
1058	0.31	0.12	0.17	68
1059	0.00	0.00	0.00	71
1060	0.33	0.10	0.16	69
1061	0.85	0.24	0.37	72
1062	0.44	0.29	0.35	66
1063	0.14	0.01	0.02	84
1064	0.00	0.00	0.00	78
1065	0.81	0.45	0.58	66
1066	0.21	0.04	0.07	69
1067	0.11	0.01	0.02	80
1068	1.00	0.01	0.03	71
1069	0.52	0.18	0.27	60
1070	0.20	0.01	0.02	77
1071	0.88	0.29	0.43	80
1072	0.25	0.06	0.10	80
1073	0.00	0.00	0.00	74
1074	0.21	0.04	0.07	69
1075	0.44	0.07	0.12	56
1076	0.32	0.13	0.18	63
1077	0.58	0.19	0.29	58
1078	0.00	0.00	0.00	63
1079	0.83	0.24	0.37	85
1080	0.52	0.15	0.24	78
1081	0.00	0.00	0.00	84
1082	0.74	0.42	0.54	73

1083	0.09	0.02	0.03	55
1084	0.51	0.26	0.34	70
1085	0.69	0.26	0.38	85
1086	0.00	0.00	0.00	68
1087	0.40	0.02	0.05	82
1088	0.00	0.00	0.00	67
1089	0.81	0.44	0.57	78
1090	0.70	0.11	0.19	64
1091	0.35	0.09	0.15	75
1092	0.38	0.16	0.23	61
1093	0.65	0.17	0.28	63
1094	0.00	0.00	0.00	77
1095	0.36	0.13	0.19	70
1096	0.86	0.34	0.48	71
1097	0.44	0.12	0.18	69
1098	0.58	0.22	0.32	63
1099	0.80	0.49	0.61	67
1100	0.57	0.06	0.11	68
1101	0.00	0.00	0.00	57
1102	0.90	0.54	0.67	69
1103	0.14	0.01	0.03	70
1104	0.40	0.05	0.09	75
1105	0.21	0.05	0.08	62
1106	0.25	0.01	0.03	72
1107	0.00	0.00	0.00	76
1108	0.00	0.00	0.00	72
1109	0.00	0.00	0.00	86
1110	0.85	0.43	0.57	82
1111	0.00	0.00	0.00	70
1112	0.50	0.01	0.03	72
1113	0.65	0.24	0.35	70
1114	0.20	0.02	0.03	57
1115	0.25	0.04	0.07	68
1116	0.00	0.00	0.00	64
1117	0.29	0.03	0.05	66
1118	0.50	0.11	0.18	81
1119	0.68	0.24	0.35	63
1120	0.15	0.06	0.09	62
1121	0.00	0.00	0.00	79
1122	0.80	0.21	0.34	56
1123	0.24	0.06	0.09	71
1124	0.00	0.00	0.00	78
1125	0.80	0.06	0.11	66
1126	0.00	0.00	0.00	62
1127	0.75	0.18	0.29	66
1128	0.00	0.00	0.00	70
1129	0.94	0.46	0.62	65
1130	0.85	0.37	0.51	63
1131	0.89	0.52	0.66	79
1132	0.38	0.07	0.12	67
1133	0.00	0.00	0.00	64
1134	0.20	0.03	0.05	67
1135	0.73	0.21	0.32	78
1136	0.44	0.07	0.13	54
1137	0.00	0.00	0.00	64
1138	0.39	0.09	0.15	76
1139	0.00	0.00	0.00	64
1140	0.00	0.00	0.00	67
1141	0.06	0.01	0.02	70
1142	0.44	0.06	0.11	66
1143	0.74	0.40	0.52	62
1144	0.00	0.00	0.00	67
1145	0.43	0.06	0.11	47
1146	0.35	0.09	0.14	69
1147	0.71	0.40	0.51	63
1148	0.37	0.10	0.16	70
1149	0.41	0.13	0.19	55
1150	0.57	0.33	0.42	49
1151	0.57	0.07	0.12	58
1152	0.00	0.00	0.00	65
1153	0.00	0.00	0.00	67
1154	0.00	0.00	0.00	66
1155	0.94	0.52	0.67	62
1156	0.62	0.07	0.12	72
1157	0.90	0.42	0.57	62
1158	0.00	0.00	0.00	60
1159	0.43	0.16	0.23	64
1160	0.30	0.05	0.09	59
1161	0.10	0.02	0.03	55
1162	0.51	0.29	0.37	63
1163	0.77	0.36	0.49	64
1164	0.00	0.00	0.00	54
1165	0.32	0.10	0.15	62

1166	0.00	0.00	0.00	73
1167	0.46	0.21	0.29	56
1168	0.33	0.03	0.06	60
1169	0.35	0.11	0.17	63
1170	0.80	0.05	0.10	73
1171	0.60	0.31	0.41	58
1172	0.29	0.03	0.06	59
1173	0.23	0.04	0.07	68
1174	0.45	0.14	0.22	63
1175	0.98	0.60	0.74	70
1176	0.87	0.42	0.57	62
1177	0.00	0.00	0.00	62
1178	0.00	0.00	0.00	45
1179	0.97	0.37	0.53	79
1180	0.70	0.12	0.21	58
1181	0.88	0.30	0.44	71
1182	0.12	0.02	0.03	56
1183	0.00	0.00	0.00	63
1184	0.00	0.00	0.00	72
1185	0.33	0.04	0.06	56
1186	0.82	0.19	0.30	75
1187	0.17	0.02	0.03	57
1188	0.45	0.08	0.14	60
1189	0.25	0.02	0.03	65
1190	0.50	0.01	0.03	68
1191	0.59	0.16	0.25	62
1192	0.00	0.00	0.00	68
1193	0.00	0.00	0.00	66
1194	0.40	0.04	0.06	57
1195	0.11	0.01	0.03	67
1196	0.88	0.10	0.18	69
1197	0.36	0.06	0.10	66
1198	0.40	0.03	0.06	62
1199	0.33	0.08	0.14	59
1200	0.92	0.21	0.34	57
1201	1.00	0.31	0.47	62
1202	0.87	0.47	0.61	58
1203	0.00	0.00	0.00	67
1204	0.63	0.35	0.45	74
1205	0.50	0.02	0.04	55
1206	0.55	0.09	0.16	65
1207	0.47	0.11	0.17	75
1208	0.63	0.20	0.30	61
1209	0.69	0.39	0.49	62
1210	0.14	0.02	0.03	59
1211	0.50	0.19	0.28	47
1212	0.00	0.00	0.00	59
1213	0.95	0.36	0.52	59
1214	1.00	0.03	0.05	74
1215	0.25	0.02	0.03	65
1216	0.00	0.00	0.00	60
1217	0.53	0.19	0.27	54
1218	0.00	0.00	0.00	62
1219	0.93	0.68	0.79	78
1220	0.85	0.57	0.68	72
1221	0.75	0.35	0.48	60
1222	0.43	0.14	0.21	63
1223	0.00	0.00	0.00	66
1224	0.56	0.14	0.23	69
1225	0.00	0.00	0.00	69
1226	0.80	0.18	0.29	68
1227	0.53	0.17	0.26	58
1228	0.00	0.00	0.00	51
1229	0.00	0.00	0.00	59
1230	0.00	0.00	0.00	75
1231	0.50	0.11	0.18	64
1232	0.00	0.00	0.00	66
1233	0.29	0.03	0.06	58
1234	0.00	0.00	0.00	63
1235	0.06	0.02	0.03	62
1236	0.00	0.00	0.00	57
1237	1.00	0.01	0.03	77
1238	0.81	0.40	0.54	52
1239	0.86	0.30	0.45	63
1240	0.90	0.40	0.55	48
1241	0.00	0.00	0.00	71
1242	0.79	0.18	0.29	62
1243	0.43	0.10	0.16	61
1244	0.00	0.00	0.00	53
1245	0.09	0.01	0.02	75
1246	0.38	0.05	0.10	55
1247	0.50	0.02	0.04	55
1248	0.00	0.00	0.00	49

1249	0.33	0.05	0.09	74
1250	0.97	0.47	0.64	59
1251	0.38	0.14	0.21	56
1252	0.33	0.10	0.15	63
1253	0.59	0.21	0.31	48
1254	0.95	0.60	0.73	62
1255	0.00	0.00	0.00	69
1256	0.30	0.05	0.08	65
1257	0.00	0.00	0.00	62
1258	0.39	0.14	0.20	51
1259	0.62	0.12	0.21	64
1260	0.00	0.00	0.00	64
1261	0.00	0.00	0.00	63
1262	0.93	0.22	0.36	58
1263	0.36	0.07	0.12	54
1264	0.00	0.00	0.00	62
1265	0.00	0.00	0.00	59
1266	0.90	0.46	0.60	57
1267	0.14	0.02	0.03	51
1268	0.25	0.04	0.07	46
1269	0.97	0.53	0.68	55
1270	0.88	0.10	0.18	69
1271	0.60	0.14	0.22	65
1272	0.38	0.08	0.14	60
1273	0.35	0.10	0.16	59
1274	0.25	0.05	0.08	62
1275	0.00	0.00	0.00	52
1276	0.40	0.07	0.12	57
1277	0.29	0.03	0.06	61
1278	0.70	0.11	0.19	62
1279	0.93	0.57	0.71	47
1280	0.25	0.03	0.06	63
1281	0.58	0.11	0.19	61
1282	0.60	0.18	0.28	50
1283	0.27	0.08	0.12	52
1284	0.68	0.23	0.35	56
1285	0.67	0.04	0.07	57
1286	0.71	0.10	0.18	49
1287	0.57	0.14	0.23	56
1288	0.57	0.27	0.36	49
1289	0.00	0.00	0.00	55
1290	0.00	0.00	0.00	68
1291	0.90	0.50	0.64	52
1292	0.29	0.03	0.05	73
1293	0.88	0.43	0.58	67
1294	0.00	0.00	0.00	54
1295	0.25	0.06	0.10	34
1296	1.00	0.34	0.51	56
1297	0.00	0.00	0.00	66
1298	1.00	0.03	0.06	68
1299	0.57	0.06	0.11	64
1300	0.91	0.50	0.65	64
1301	0.00	0.00	0.00	48
1302	0.00	0.00	0.00	63
1303	0.00	0.00	0.00	62
1304	0.50	0.02	0.04	54
1305	0.23	0.10	0.14	51
1306	0.22	0.07	0.11	55
1307	0.00	0.00	0.00	53
1308	0.61	0.31	0.41	54
1309	0.67	0.16	0.26	61
1310	0.00	0.00	0.00	42
1311	0.25	0.02	0.03	55
1312	0.00	0.00	0.00	64
1313	0.00	0.00	0.00	58
1314	0.90	0.36	0.51	50
1315	0.00	0.00	0.00	57
1316	0.59	0.22	0.32	46
1317	1.00	0.05	0.09	42
1318	0.50	0.22	0.30	74
1319	0.00	0.00	0.00	55
1320	0.00	0.00	0.00	59
1321	1.00	0.02	0.04	56
1322	0.00	0.00	0.00	61
1323	0.00	0.00	0.00	43
1324	0.47	0.18	0.26	45
1325	0.62	0.09	0.16	56
1326	0.72	0.35	0.47	52
1327	0.52	0.20	0.29	56
1328	0.00	0.00	0.00	56
1329	0.56	0.10	0.17	51
1330	0.00	0.00	0.00	54
1331	0.50	0.12	0.19	51

1332	0.00	0.00	0.00	48
1333	0.00	0.00	0.00	51
1334	0.00	0.00	0.00	38
1335	0.91	0.42	0.58	50
1336	0.00	0.00	0.00	48
1337	0.38	0.10	0.15	52
1338	0.58	0.21	0.31	52
1339	0.25	0.04	0.06	56
1340	0.50	0.04	0.07	52
1341	1.00	0.02	0.03	58
1342	0.00	0.00	0.00	56
1343	0.33	0.03	0.06	62
1344	0.93	0.32	0.47	44
1345	0.38	0.06	0.10	53
1346	0.20	0.02	0.03	53
1347	0.00	0.00	0.00	52
1348	0.50	0.10	0.17	58
1349	0.64	0.36	0.46	50
1350	0.00	0.00	0.00	62
1351	0.96	0.39	0.55	59
1352	0.00	0.00	0.00	57
1353	0.63	0.24	0.35	50
1354	0.67	0.11	0.19	55
1355	0.00	0.00	0.00	55
1356	0.17	0.02	0.03	56
1357	0.16	0.08	0.11	38
1358	0.20	0.04	0.06	53
1359	1.00	0.23	0.37	44
1360	1.00	0.23	0.38	56
1361	0.25	0.04	0.06	56
1362	1.00	0.33	0.49	46
1363	0.73	0.22	0.34	49
1364	0.00	0.00	0.00	66
1365	0.33	0.05	0.09	60
1366	0.86	0.11	0.19	56
1367	0.00	0.00	0.00	63
1368	0.53	0.15	0.23	67
1369	1.00	0.44	0.61	59
1370	0.94	0.33	0.48	49
1371	0.76	0.25	0.38	51
1372	0.20	0.02	0.04	50
1373	0.93	0.40	0.56	63
1374	0.20	0.02	0.03	55
1375	0.00	0.00	0.00	60
1376	0.52	0.18	0.27	60
1377	0.00	0.00	0.00	42
1378	0.94	0.30	0.45	54
1379	0.00	0.00	0.00	50
1380	0.00	0.00	0.00	45
1381	0.60	0.06	0.12	47
1382	0.11	0.02	0.03	54
1383	0.33	0.04	0.08	45
1384	0.00	0.00	0.00	52
1385	0.73	0.23	0.35	48
1386	0.60	0.06	0.11	50
1387	0.17	0.02	0.04	47
1388	0.75	0.16	0.26	57
1389	0.00	0.00	0.00	49
1390	0.55	0.27	0.36	44
1391	0.00	0.00	0.00	58
1392	0.77	0.19	0.30	54
1393	0.38	0.12	0.18	51
1394	0.50	0.02	0.04	51
1395	0.83	0.21	0.33	48
1396	0.67	0.13	0.22	61
1397	1.00	0.02	0.03	61
1398	0.62	0.15	0.24	55
1399	0.74	0.25	0.37	57
1400	0.50	0.06	0.11	49
1401	0.50	0.04	0.07	56
1402	0.54	0.13	0.22	52
1403	0.75	0.12	0.21	49
1404	0.92	0.80	0.86	41
1405	0.75	0.32	0.44	57
1406	0.33	0.02	0.04	54
1407	0.70	0.55	0.62	47
1408	0.38	0.07	0.12	41
1409	1.00	0.39	0.56	49
1410	1.00	0.44	0.61	48
1411	0.17	0.02	0.03	55
1412	0.73	0.13	0.23	60
1413	1.00	0.01	0.03	67
1414	0.00	0.00	0.00	50

1415	0.00	0.00	0.00	53
1416	0.40	0.10	0.16	59
1417	0.53	0.14	0.22	66
1418	0.67	0.04	0.08	50
1419	0.80	0.11	0.20	36
1420	0.30	0.06	0.11	47
1421	0.00	0.00	0.00	46
1422	0.38	0.10	0.16	51
1423	0.82	0.18	0.30	49
1424	0.50	0.07	0.12	56
1425	0.00	0.00	0.00	51
1426	0.67	0.04	0.07	53
1427	0.30	0.06	0.11	47
1428	0.00	0.00	0.00	39
1429	0.97	0.56	0.71	50
1430	0.86	0.20	0.33	59
1431	0.00	0.00	0.00	67
1432	0.00	0.00	0.00	53
1433	0.38	0.08	0.14	72
1434	0.62	0.10	0.17	51
1435	0.54	0.12	0.20	56
1436	0.67	0.11	0.18	56
1437	0.57	0.16	0.25	51
1438	0.00	0.00	0.00	46
1439	0.67	0.04	0.07	52
1440	0.00	0.00	0.00	41
1441	1.00	0.04	0.08	47
1442	1.00	0.02	0.04	45
1443	0.10	0.02	0.03	54
1444	0.15	0.04	0.06	52
1445	0.00	0.00	0.00	52
1446	0.61	0.25	0.35	44
1447	1.00	0.17	0.29	47
1448	0.00	0.00	0.00	48
1449	0.33	0.02	0.03	56
1450	0.00	0.00	0.00	54
1451	0.12	0.02	0.03	65
1452	0.50	0.07	0.13	55
1453	0.29	0.07	0.11	61
1454	0.00	0.00	0.00	62
1455	0.65	0.22	0.33	49
1456	0.20	0.02	0.03	53
1457	0.62	0.31	0.41	42
1458	0.75	0.05	0.10	59
1459	0.00	0.00	0.00	49
1460	0.71	0.10	0.18	50
1461	0.00	0.00	0.00	45
1462	0.42	0.11	0.17	47
1463	0.71	0.33	0.45	45
1464	1.00	0.04	0.08	50
1465	0.33	0.05	0.08	62
1466	0.00	0.00	0.00	51
1467	0.33	0.02	0.03	62
1468	0.93	0.48	0.63	54
1469	0.50	0.11	0.17	38
1470	0.81	0.26	0.40	65
1471	1.00	0.29	0.45	52
1472	0.50	0.09	0.15	44
1473	0.17	0.04	0.06	50
1474	0.00	0.00	0.00	56
1475	0.00	0.00	0.00	58
1476	0.12	0.02	0.03	58
1477	0.00	0.00	0.00	39
1478	0.96	0.48	0.64	50
1479	0.00	0.00	0.00	49
1480	0.00	0.00	0.00	41
1481	0.83	0.33	0.47	57
1482	0.00	0.00	0.00	49
1483	0.00	0.00	0.00	49
1484	1.00	0.10	0.18	59
1485	0.93	0.28	0.43	47
1486	0.50	0.02	0.04	53
1487	0.00	0.00	0.00	42
1488	0.00	0.00	0.00	47
1489	0.33	0.02	0.04	52
1490	0.72	0.30	0.42	44
1491	0.00	0.00	0.00	47
1492	0.81	0.25	0.39	51
1493	0.00	0.00	0.00	39
1494	0.00	0.00	0.00	38
1495	0.40	0.12	0.19	49
1496	0.62	0.16	0.26	49
1497	0.00	0.00	0.00	51

1498	1.00	0.04	0.07	52
1499	0.50	0.06	0.11	48
1500	0.00	0.00	0.00	51
1501	0.25	0.02	0.03	56
1502	0.00	0.00	0.00	48
1503	0.82	0.48	0.61	58
1504	0.50	0.02	0.04	44
1505	0.00	0.00	0.00	45
1506	0.20	0.02	0.04	44
1507	0.00	0.00	0.00	55
1508	0.33	0.04	0.08	45
1509	0.62	0.17	0.27	46
1510	0.00	0.00	0.00	46
1511	0.00	0.00	0.00	43
1512	0.89	0.19	0.31	42
1513	0.00	0.00	0.00	44
1514	0.58	0.33	0.42	45
1515	1.00	0.48	0.65	42
1516	1.00	0.36	0.53	42
1517	0.22	0.10	0.14	49
1518	1.00	0.18	0.30	51
1519	0.50	0.02	0.04	47
1520	0.00	0.00	0.00	48
1521	0.00	0.00	0.00	54
1522	0.22	0.05	0.09	38
1523	0.00	0.00	0.00	44
1524	0.67	0.04	0.07	55
1525	0.00	0.00	0.00	47
1526	0.00	0.00	0.00	55
1527	0.00	0.00	0.00	48
1528	0.67	0.04	0.07	54
1529	0.67	0.06	0.12	63
1530	0.77	0.25	0.38	40
1531	0.00	0.00	0.00	40
1532	0.22	0.04	0.07	48
1533	0.00	0.00	0.00	49
1534	0.00	0.00	0.00	45
1535	1.00	0.19	0.32	42
1536	1.00	0.06	0.11	54
1537	0.64	0.12	0.21	56
1538	0.50	0.03	0.05	38
1539	0.00	0.00	0.00	47
1540	0.44	0.10	0.16	40
1541	0.82	0.20	0.32	46
1542	1.00	0.15	0.26	46
1543	0.25	0.02	0.04	42
1544	0.70	0.33	0.45	48
1545	1.00	0.02	0.05	41
1546	0.00	0.00	0.00	35
1547	0.00	0.00	0.00	45
1548	0.20	0.04	0.06	55
1549	0.88	0.30	0.44	47
1550	1.00	0.12	0.22	48
1551	0.84	0.68	0.75	40
1552	0.67	0.04	0.07	51
1553	0.75	0.07	0.12	44
1554	0.91	0.20	0.32	51
1555	0.00	0.00	0.00	59
1556	0.50	0.18	0.27	60
1557	1.00	0.07	0.12	46
1558	0.67	0.05	0.09	43
1559	0.00	0.00	0.00	52
1560	0.67	0.09	0.16	44
1561	0.95	0.50	0.66	38
1562	0.40	0.10	0.15	42
1563	0.30	0.06	0.10	49
1564	1.00	0.15	0.25	48
1565	1.00	0.38	0.56	52
1566	0.97	0.63	0.76	46
1567	0.00	0.00	0.00	46
1568	0.81	0.44	0.57	39
1569	0.57	0.09	0.15	47
1570	0.60	0.12	0.21	48
1571	0.00	0.00	0.00	47
1572	0.00	0.00	0.00	52
1573	0.00	0.00	0.00	31
1574	0.95	0.38	0.55	55
1575	0.14	0.02	0.04	49
1576	1.00	0.43	0.61	46
1577	0.25	0.02	0.03	55
1578	0.00	0.00	0.00	42
1579	0.89	0.20	0.32	41
1580	0.00	0.00	0.00	47

1581	0.40	0.08	0.13	50
1582	0.00	0.00	0.00	47
1583	0.50	0.11	0.18	54
1584	0.50	0.04	0.08	49
1585	0.25	0.06	0.09	35
1586	0.00	0.00	0.00	43
1587	0.64	0.13	0.22	53
1588	0.00	0.00	0.00	49
1589	0.00	0.00	0.00	44
1590	0.50	0.05	0.09	39
1591	0.00	0.00	0.00	36
1592	0.00	0.00	0.00	46
1593	0.75	0.22	0.34	55
1594	0.91	0.21	0.34	47
1595	1.00	0.22	0.35	51
1596	0.00	0.00	0.00	42
1597	0.00	0.00	0.00	50
1598	0.53	0.20	0.29	40
1599	0.00	0.00	0.00	38
1600	0.00	0.00	0.00	47
1601	0.88	0.38	0.53	37
1602	0.25	0.02	0.03	62
1603	0.00	0.00	0.00	43
1604	0.00	0.00	0.00	66
1605	0.33	0.03	0.06	33
1606	0.00	0.00	0.00	35
1607	1.00	0.29	0.44	42
1608	0.96	0.57	0.71	44
1609	0.67	0.05	0.09	40
1610	0.91	0.46	0.61	46
1611	0.33	0.04	0.07	55
1612	0.88	0.35	0.50	43
1613	0.00	0.00	0.00	51
1614	0.69	0.24	0.35	38
1615	0.00	0.00	0.00	47
1616	0.45	0.10	0.16	51
1617	0.00	0.00	0.00	52
1618	0.25	0.02	0.04	43
1619	1.00	0.03	0.05	37
1620	0.00	0.00	0.00	50
1621	0.00	0.00	0.00	44
1622	0.56	0.12	0.20	41
1623	0.50	0.13	0.21	46
1624	1.00	0.05	0.09	42
1625	0.94	0.33	0.49	48
1626	0.20	0.02	0.04	51
1627	0.00	0.00	0.00	37
1628	0.20	0.04	0.07	48
1629	0.00	0.00	0.00	43
1630	0.00	0.00	0.00	50
1631	0.00	0.00	0.00	41
1632	0.29	0.04	0.08	45
1633	0.90	0.40	0.55	45
1634	0.43	0.11	0.17	56
1635	0.71	0.27	0.39	44
1636	1.00	0.33	0.50	39
1637	0.74	0.27	0.40	51
1638	0.00	0.00	0.00	31
1639	0.00	0.00	0.00	53
1640	1.00	0.19	0.31	59
1641	0.20	0.03	0.05	35
1642	0.38	0.10	0.15	52
1643	0.00	0.00	0.00	32
1644	0.00	0.00	0.00	45
1645	0.00	0.00	0.00	50
1646	0.36	0.08	0.13	52
1647	0.53	0.26	0.34	39
1648	0.25	0.02	0.03	56
1649	0.75	0.32	0.45	37
1650	0.30	0.07	0.12	42
1651	0.62	0.09	0.16	55
1652	0.89	0.47	0.62	34
1653	0.83	0.12	0.22	40
1654	0.00	0.00	0.00	45
1655	0.00	0.00	0.00	56
1656	0.00	0.00	0.00	50
1657	0.00	0.00	0.00	46
1658	0.84	0.37	0.52	43
1659	0.88	0.45	0.59	49
1660	0.80	0.23	0.36	52
1661	1.00	0.02	0.04	54
1662	0.00	0.00	0.00	43
1663	0.00	0.00	0.00	59

1664	0.00	0.00	0.00	45
1665	0.00	0.00	0.00	51
1666	0.00	0.00	0.00	47
1667	0.17	0.02	0.04	50
1668	0.86	0.30	0.44	40
1669	0.25	0.03	0.05	38
1670	1.00	0.14	0.24	37
1671	0.50	0.02	0.04	51
1672	0.86	0.51	0.64	47
1673	0.86	0.12	0.21	49
1674	0.25	0.02	0.04	45
1675	0.00	0.00	0.00	46
1676	0.00	0.00	0.00	45
1677	0.38	0.07	0.11	45
1678	0.00	0.00	0.00	43
1679	1.00	0.02	0.04	52
1680	0.60	0.07	0.13	41
1681	0.00	0.00	0.00	41
1682	0.00	0.00	0.00	35
1683	0.67	0.05	0.09	41
1684	0.50	0.11	0.19	35
1685	1.00	0.02	0.04	53
1686	0.00	0.00	0.00	43
1687	0.00	0.00	0.00	39
1688	0.00	0.00	0.00	38
1689	0.50	0.18	0.26	51
1690	0.50	0.06	0.11	47
1691	0.00	0.00	0.00	30
1692	0.64	0.23	0.34	30
1693	0.00	0.00	0.00	47
1694	0.00	0.00	0.00	51
1695	0.00	0.00	0.00	43
1696	0.86	0.30	0.44	40
1697	0.00	0.00	0.00	33
1698	0.00	0.00	0.00	45
1699	0.00	0.00	0.00	42
1700	1.00	0.42	0.59	45
1701	0.83	0.38	0.53	39
1702	0.00	0.00	0.00	56
1703	1.00	0.36	0.53	44
1704	0.83	0.34	0.48	44
1705	1.00	0.40	0.57	40
1706	1.00	0.23	0.37	35
1707	0.00	0.00	0.00	32
1708	1.00	0.27	0.42	45
1709	0.00	0.00	0.00	37
1710	0.00	0.00	0.00	47
1711	0.25	0.07	0.11	30
1712	0.00	0.00	0.00	38
1713	0.00	0.00	0.00	39
1714	0.73	0.31	0.43	36
1715	0.00	0.00	0.00	38
1716	0.20	0.02	0.03	55
1717	0.60	0.07	0.13	42
1718	0.55	0.24	0.33	46
1719	0.54	0.14	0.22	51
1720	0.27	0.11	0.16	35
1721	0.85	0.47	0.61	36
1722	0.89	0.42	0.57	38
1723	0.92	0.30	0.45	40
1724	0.67	0.04	0.07	53
1725	0.00	0.00	0.00	27
1726	0.20	0.02	0.04	48
1727	0.83	0.50	0.62	38
1728	0.18	0.05	0.08	38
1729	0.86	0.11	0.19	57
1730	0.85	0.47	0.60	47
1731	0.00	0.00	0.00	48
1732	0.00	0.00	0.00	41
1733	0.15	0.06	0.09	33
1734	0.33	0.05	0.09	37
1735	0.50	0.04	0.08	45
1736	0.95	0.41	0.57	44
1737	0.80	0.26	0.39	47
1738	1.00	0.38	0.55	48
1739	0.25	0.02	0.04	48
1740	0.00	0.00	0.00	51
1741	0.91	0.24	0.38	42
1742	0.93	0.29	0.44	45
1743	1.00	0.14	0.24	43
1744	0.00	0.00	0.00	50
1745	1.00	0.25	0.40	40
1746	0.67	0.16	0.26	49

1747	0.00	0.00	0.00	37
1748	0.83	0.42	0.56	36
1749	0.40	0.05	0.09	41
1750	0.00	0.00	0.00	41
1751	0.91	0.29	0.44	34
1752	0.00	0.00	0.00	37
1753	0.80	0.20	0.31	41
1754	0.00	0.00	0.00	46
1755	0.00	0.00	0.00	35
1756	0.59	0.22	0.32	46
1757	0.00	0.00	0.00	44
1758	0.50	0.05	0.09	43
1759	0.17	0.03	0.06	30
1760	0.00	0.00	0.00	46
1761	0.00	0.00	0.00	39
1762	0.00	0.00	0.00	41
1763	0.00	0.00	0.00	47
1764	0.86	0.18	0.29	34
1765	0.00	0.00	0.00	32
1766	0.71	0.29	0.41	42
1767	0.90	0.24	0.38	38
1768	0.00	0.00	0.00	35
1769	0.57	0.12	0.20	33
1770	0.67	0.05	0.10	39
1771	0.00	0.00	0.00	37
1772	0.54	0.15	0.23	48
1773	1.00	0.33	0.49	46
1774	0.67	0.14	0.23	44
1775	0.50	0.02	0.03	63
1776	0.80	0.10	0.18	40
1777	1.00	0.03	0.05	39
1778	0.50	0.08	0.14	38
1779	0.00	0.00	0.00	44
1780	0.92	0.55	0.69	44
1781	0.67	0.05	0.09	40
1782	0.33	0.05	0.08	43
1783	0.00	0.00	0.00	39
1784	0.44	0.09	0.15	44
1785	0.71	0.13	0.22	38
1786	0.00	0.00	0.00	39
1787	1.00	0.05	0.09	44
1788	0.00	0.00	0.00	46
1789	0.70	0.17	0.28	40
1790	0.75	0.27	0.39	45
1791	0.00	0.00	0.00	39
1792	0.20	0.05	0.08	41
1793	0.71	0.21	0.33	47
1794	0.38	0.07	0.12	43
1795	0.76	0.38	0.51	34
1796	0.72	0.40	0.51	45
1797	1.00	0.19	0.32	31
1798	0.25	0.06	0.09	36
1799	0.68	0.27	0.39	55
1800	0.00	0.00	0.00	30
1801	0.00	0.00	0.00	35
1802	1.00	0.23	0.37	48
1803	0.12	0.03	0.04	38
1804	0.00	0.00	0.00	35
1805	0.00	0.00	0.00	32
1806	0.71	0.27	0.39	37
1807	1.00	0.19	0.32	37
1808	0.00	0.00	0.00	36
1809	0.00	0.00	0.00	42
1810	0.00	0.00	0.00	42
1811	0.00	0.00	0.00	35
1812	0.57	0.10	0.17	39
1813	0.71	0.28	0.40	36
1814	0.43	0.06	0.11	48
1815	1.00	0.44	0.62	45
1816	0.75	0.26	0.39	34
1817	0.67	0.19	0.29	32
1818	1.00	0.27	0.43	44
1819	0.00	0.00	0.00	46
1820	0.00	0.00	0.00	40
1821	0.00	0.00	0.00	37
1822	0.00	0.00	0.00	35
1823	0.00	0.00	0.00	33
1824	0.00	0.00	0.00	38
1825	1.00	0.05	0.10	38
1826	0.73	0.18	0.29	45
1827	0.00	0.00	0.00	36
1828	0.00	0.00	0.00	45
1829	0.96	0.68	0.80	38

1830	0.17	0.03	0.05	35
1831	0.75	0.26	0.39	34
1832	0.50	0.03	0.06	33
1833	0.60	0.13	0.21	23
1834	0.50	0.02	0.04	44
1835	0.00	0.00	0.00	50
1836	1.00	0.05	0.09	44
1837	0.86	0.26	0.40	46
1838	0.00	0.00	0.00	33
1839	0.60	0.20	0.30	45
1840	0.00	0.00	0.00	37
1841	1.00	0.03	0.05	39
1842	0.00	0.00	0.00	40
1843	0.00	0.00	0.00	41
1844	0.33	0.05	0.08	43
1845	0.00	0.00	0.00	36
1846	0.00	0.00	0.00	38
1847	0.00	0.00	0.00	33
1848	0.00	0.00	0.00	37
1849	1.00	0.12	0.21	34
1850	0.00	0.00	0.00	42
1851	0.60	0.41	0.48	37
1852	0.80	0.11	0.19	37
1853	0.91	0.24	0.38	41
1854	1.00	0.45	0.62	40
1855	0.00	0.00	0.00	40
1856	0.00	0.00	0.00	39
1857	0.00	0.00	0.00	30
1858	0.33	0.02	0.04	49
1859	0.67	0.28	0.39	29
1860	0.00	0.00	0.00	45
1861	0.25	0.05	0.08	40
1862	0.90	0.23	0.37	39
1863	0.00	0.00	0.00	37
1864	0.81	0.35	0.49	37
1865	0.91	0.28	0.43	36
1866	0.00	0.00	0.00	39
1867	0.38	0.07	0.12	42
1868	0.73	0.25	0.37	44
1869	0.00	0.00	0.00	39
1870	0.00	0.00	0.00	46
1871	0.00	0.00	0.00	43
1872	0.14	0.03	0.05	34
1873	0.40	0.04	0.08	47
1874	0.57	0.10	0.17	39
1875	0.33	0.03	0.05	36
1876	0.56	0.14	0.22	37
1877	0.00	0.00	0.00	47
1878	0.50	0.06	0.11	48
1879	0.67	0.19	0.29	32
1880	0.87	0.28	0.43	46
1881	0.17	0.03	0.05	38
1882	0.00	0.00	0.00	36
1883	0.00	0.00	0.00	40
1884	0.38	0.09	0.14	34
1885	0.00	0.00	0.00	41
1886	0.00	0.00	0.00	42
1887	0.00	0.00	0.00	38
1888	1.00	0.02	0.04	49
1889	1.00	0.42	0.59	36
1890	0.70	0.19	0.30	36
1891	0.67	0.23	0.34	44
1892	0.33	0.04	0.07	24
1893	0.00	0.00	0.00	36
1894	1.00	0.39	0.56	46
1895	0.00	0.00	0.00	33
1896	1.00	0.12	0.21	42
1897	0.00	0.00	0.00	35
1898	0.00	0.00	0.00	31
1899	0.71	0.33	0.45	36
1900	0.00	0.00	0.00	30
1901	0.62	0.10	0.18	49
1902	0.67	0.12	0.20	34
1903	1.00	0.07	0.14	40
1904	0.00	0.00	0.00	42
1905	0.00	0.00	0.00	44
1906	0.84	0.34	0.48	47
1907	0.00	0.00	0.00	46
1908	0.57	0.33	0.42	36
1909	1.00	0.06	0.11	35
1910	0.00	0.00	0.00	46
1911	0.00	0.00	0.00	39
1912	0.85	0.29	0.43	38

1913	0.00	0.00	0.00	38
1914	0.73	0.19	0.30	43
1915	0.84	0.52	0.64	31
1916	0.33	0.08	0.12	39
1917	0.00	0.00	0.00	38
1918	0.75	0.20	0.32	45
1919	0.58	0.19	0.29	37
1920	0.00	0.00	0.00	29
1921	0.00	0.00	0.00	31
1922	0.61	0.34	0.44	41
1923	0.17	0.02	0.03	54
1924	0.80	0.12	0.22	32
1925	0.00	0.00	0.00	32
1926	0.00	0.00	0.00	38
1927	0.94	0.38	0.54	42
1928	0.00	0.00	0.00	41
1929	0.00	0.00	0.00	47
1930	1.00	0.40	0.57	30
1931	1.00	0.05	0.09	41
1932	0.00	0.00	0.00	40
1933	0.62	0.19	0.29	43
1934	0.00	0.00	0.00	42
1935	0.33	0.06	0.10	36
1936	0.57	0.29	0.38	42
1937	1.00	0.03	0.05	36
1938	0.94	0.50	0.65	32
1939	1.00	0.12	0.21	50
1940	0.33	0.03	0.05	35
1941	0.00	0.00	0.00	41
1942	0.80	0.20	0.32	40
1943	0.00	0.00	0.00	38
1944	0.84	0.47	0.60	34
1945	0.00	0.00	0.00	42
1946	0.90	0.32	0.47	28
1947	0.00	0.00	0.00	37
1948	0.00	0.00	0.00	32
1949	0.00	0.00	0.00	32
1950	0.69	0.35	0.46	26
1951	0.00	0.00	0.00	49
1952	0.00	0.00	0.00	32
1953	0.50	0.03	0.06	31
1954	0.71	0.12	0.21	40
1955	0.00	0.00	0.00	47
1956	1.00	0.07	0.13	43
1957	0.00	0.00	0.00	38
1958	0.77	0.26	0.39	38
1959	0.00	0.00	0.00	34
1960	0.32	0.21	0.25	39
1961	1.00	0.03	0.06	34
1962	0.20	0.02	0.04	42
1963	0.60	0.09	0.16	32
1964	0.00	0.00	0.00	41
1965	0.33	0.02	0.04	42
1966	0.00	0.00	0.00	37
1967	0.00	0.00	0.00	41
1968	0.86	0.60	0.71	30
1969	0.50	0.24	0.32	25
1970	0.50	0.15	0.23	40
1971	0.00	0.00	0.00	43
1972	0.00	0.00	0.00	42
1973	0.00	0.00	0.00	32
1974	0.00	0.00	0.00	33
1975	1.00	0.21	0.35	28
1976	0.00	0.00	0.00	35
1977	0.92	0.22	0.36	49
1978	1.00	0.33	0.49	49
1979	0.00	0.00	0.00	34
1980	0.00	0.00	0.00	28
1981	1.00	0.24	0.38	34
1982	0.00	0.00	0.00	30
1983	0.50	0.03	0.05	40
1984	0.00	0.00	0.00	38
1985	0.00	0.00	0.00	42
1986	0.00	0.00	0.00	32
1987	0.00	0.00	0.00	37
1988	0.25	0.03	0.05	34
1989	0.75	0.15	0.24	41
1990	0.00	0.00	0.00	34
1991	0.00	0.00	0.00	34
1992	0.00	0.00	0.00	30
1993	0.67	0.17	0.27	36
1994	0.83	0.16	0.26	32
1995	0.00	0.00	0.00	38

1996	0.00	0.00	0.00	32
1997	0.00	0.00	0.00	39
1998	0.00	0.00	0.00	32
1999	0.73	0.18	0.29	44
2000	0.50	0.02	0.05	41
2001	1.00	0.24	0.39	37
2002	0.30	0.08	0.12	38
2003	0.00	0.00	0.00	31
2004	0.00	0.00	0.00	35
2005	0.80	0.24	0.36	34
2006	0.80	0.24	0.36	34
2007	1.00	0.06	0.12	31
2008	0.00	0.00	0.00	40
2009	1.00	0.25	0.40	40
2010	0.40	0.05	0.09	39
2011	0.62	0.14	0.22	37
2012	0.00	0.00	0.00	35
2013	0.00	0.00	0.00	27
2014	0.00	0.00	0.00	38
2015	0.00	0.00	0.00	34
2016	0.00	0.00	0.00	33
2017	0.00	0.00	0.00	31
2018	1.00	0.06	0.11	34
2019	0.00	0.00	0.00	40
2020	0.00	0.00	0.00	29
2021	0.00	0.00	0.00	34
2022	0.00	0.00	0.00	37
2023	0.54	0.23	0.33	30
2024	0.00	0.00	0.00	34
2025	0.00	0.00	0.00	36
2026	0.92	0.22	0.36	49
2027	0.00	0.00	0.00	22
2028	0.94	0.38	0.55	39
2029	0.00	0.00	0.00	36
2030	1.00	0.49	0.65	37
2031	0.90	0.28	0.43	32
2032	1.00	0.17	0.29	41
2033	0.00	0.00	0.00	28
2034	0.30	0.08	0.12	38
2035	0.00	0.00	0.00	26
2036	0.00	0.00	0.00	33
2037	0.00	0.00	0.00	32
2038	0.80	0.22	0.34	37
2039	0.00	0.00	0.00	32
2040	0.55	0.15	0.24	40
2041	0.40	0.07	0.12	29
2042	0.00	0.00	0.00	30
2043	0.00	0.00	0.00	33
2044	0.00	0.00	0.00	35
2045	0.50	0.18	0.26	34
2046	0.50	0.03	0.06	31
2047	0.50	0.06	0.11	32
2048	0.00	0.00	0.00	36
2049	1.00	0.02	0.05	43
2050	0.00	0.00	0.00	27
2051	0.50	0.10	0.16	31
2052	0.00	0.00	0.00	34
2053	0.00	0.00	0.00	32
2054	0.71	0.11	0.19	45
2055	0.00	0.00	0.00	39
2056	0.95	0.58	0.72	33
2057	0.40	0.05	0.09	38
2058	0.25	0.03	0.05	33
2059	0.00	0.00	0.00	44
2060	1.00	0.46	0.63	35
2061	0.40	0.10	0.16	40
2062	0.00	0.00	0.00	31
2063	1.00	0.44	0.61	32
2064	0.00	0.00	0.00	45
2065	0.93	0.40	0.56	35
2066	0.00	0.00	0.00	37
2067	0.40	0.06	0.10	35
2068	0.00	0.00	0.00	43
2069	0.00	0.00	0.00	26
2070	0.00	0.00	0.00	40
2071	1.00	0.46	0.63	37
2072	0.00	0.00	0.00	31
2073	0.40	0.11	0.18	35
2074	0.00	0.00	0.00	35
2075	0.00	0.00	0.00	31
2076	0.00	0.00	0.00	30
2077	0.83	0.18	0.29	28
2078	0.00	0.00	0.00	37

2079	0.00	0.00	0.00	38
2080	0.00	0.00	0.00	28
2081	0.00	0.00	0.00	28
2082	0.00	0.00	0.00	33
2083	1.00	0.11	0.19	28
2084	1.00	0.26	0.41	23
2085	0.84	0.46	0.59	35
2086	0.60	0.08	0.14	39
2087	0.00	0.00	0.00	31
2088	0.00	0.00	0.00	25
2089	0.77	0.46	0.58	37
2090	0.00	0.00	0.00	34
2091	0.00	0.00	0.00	34
2092	0.00	0.00	0.00	38
2093	0.00	0.00	0.00	36
2094	0.29	0.06	0.10	33
2095	0.40	0.05	0.09	40
2096	0.67	0.11	0.18	38
2097	0.33	0.04	0.07	25
2098	0.00	0.00	0.00	33
2099	1.00	0.19	0.32	42
2100	0.00	0.00	0.00	29
2101	0.00	0.00	0.00	29
2102	0.50	0.06	0.10	35
2103	0.67	0.10	0.17	40
2104	0.00	0.00	0.00	42
2105	0.00	0.00	0.00	36
2106	0.00	0.00	0.00	33
2107	0.00	0.00	0.00	33
2108	0.00	0.00	0.00	34
2109	0.00	0.00	0.00	42
2110	0.00	0.00	0.00	28
2111	0.40	0.05	0.09	40
2112	1.00	0.04	0.08	24
2113	0.00	0.00	0.00	36
2114	0.43	0.09	0.15	33
2115	0.00	0.00	0.00	32
2116	0.67	0.15	0.24	27
2117	0.00	0.00	0.00	30
2118	0.79	0.38	0.51	29
2119	0.50	0.07	0.12	28
2120	0.94	0.46	0.62	35
2121	0.00	0.00	0.00	35
2122	0.00	0.00	0.00	37
2123	0.00	0.00	0.00	35
2124	0.40	0.06	0.10	35
2125	0.00	0.00	0.00	37
2126	0.00	0.00	0.00	35
2127	0.40	0.06	0.11	32
2128	0.36	0.13	0.20	30
2129	0.00	0.00	0.00	32
2130	0.00	0.00	0.00	41
2131	1.00	0.04	0.07	26
2132	0.00	0.00	0.00	34
2133	0.00	0.00	0.00	29
2134	0.00	0.00	0.00	36
2135	0.00	0.00	0.00	29
2136	0.00	0.00	0.00	35
2137	0.83	0.37	0.51	27
2138	0.00	0.00	0.00	35
2139	0.85	0.37	0.51	30
2140	0.00	0.00	0.00	33
2141	0.67	0.05	0.10	38
2142	0.00	0.00	0.00	37
2143	1.00	0.10	0.18	31
2144	0.71	0.14	0.24	35
2145	1.00	0.37	0.54	38
2146	1.00	0.17	0.29	35
2147	0.38	0.15	0.22	33
2148	0.00	0.00	0.00	32
2149	0.67	0.05	0.10	37
2150	0.00	0.00	0.00	41
2151	0.00	0.00	0.00	39
2152	0.00	0.00	0.00	36
2153	0.00	0.00	0.00	31
2154	0.00	0.00	0.00	30
2155	1.00	0.42	0.59	26
2156	0.00	0.00	0.00	32
2157	0.00	0.00	0.00	38
2158	0.00	0.00	0.00	33
2159	0.00	0.00	0.00	32
2160	0.33	0.03	0.06	32
2161	0.00	0.00	0.00	34

2162	0.50	0.22	0.31	27
2163	0.00	0.00	0.00	37
2164	1.00	0.03	0.06	30
2165	0.00	0.00	0.00	35
2166	0.56	0.21	0.30	24
2167	0.00	0.00	0.00	37
2168	0.87	0.50	0.63	26
2169	0.00	0.00	0.00	27
2170	0.00	0.00	0.00	39
2171	0.00	0.00	0.00	25
2172	0.00	0.00	0.00	33
2173	0.00	0.00	0.00	39
2174	0.94	0.43	0.59	35
2175	1.00	0.33	0.50	30
2176	0.00	0.00	0.00	36
2177	0.33	0.04	0.06	28
2178	0.00	0.00	0.00	34
2179	0.00	0.00	0.00	35
2180	0.00	0.00	0.00	23
2181	0.00	0.00	0.00	34
2182	0.00	0.00	0.00	27
2183	1.00	0.08	0.15	25
2184	0.00	0.00	0.00	33
2185	1.00	0.15	0.26	33
2186	0.33	0.16	0.21	19
2187	0.00	0.00	0.00	38
2188	0.00	0.00	0.00	20
2189	0.00	0.00	0.00	32
2190	0.33	0.06	0.11	31
2191	0.67	0.12	0.21	33
2192	0.00	0.00	0.00	28
2193	1.00	0.06	0.11	36
2194	0.00	0.00	0.00	35
2195	0.00	0.00	0.00	26
2196	0.00	0.00	0.00	32
2197	0.00	0.00	0.00	34
2198	1.00	0.03	0.06	33
2199	0.00	0.00	0.00	27
2200	0.60	0.10	0.17	31
2201	0.00	0.00	0.00	22
2202	0.00	0.00	0.00	28
2203	0.75	0.19	0.30	32
2204	0.00	0.00	0.00	34
2205	0.00	0.00	0.00	27
2206	1.00	0.11	0.21	35
2207	0.00	0.00	0.00	32
2208	1.00	0.03	0.06	31
2209	0.00	0.00	0.00	34
2210	0.00	0.00	0.00	31
2211	0.00	0.00	0.00	38
2212	1.00	0.03	0.07	29
2213	1.00	0.08	0.15	24
2214	0.00	0.00	0.00	26
2215	0.60	0.08	0.14	39
2216	0.50	0.11	0.18	28
2217	0.00	0.00	0.00	29
2218	0.00	0.00	0.00	39
2219	0.00	0.00	0.00	26
2220	0.00	0.00	0.00	29
2221	1.00	0.41	0.58	22
2222	0.00	0.00	0.00	28
2223	1.00	0.08	0.15	37
2224	0.00	0.00	0.00	31
2225	0.20	0.03	0.04	40
2226	1.00	0.18	0.31	33
2227	0.00	0.00	0.00	41
2228	0.00	0.00	0.00	33
2229	0.00	0.00	0.00	29
2230	0.00	0.00	0.00	34
2231	0.00	0.00	0.00	28
2232	0.86	0.23	0.36	26
2233	0.00	0.00	0.00	27
2234	1.00	0.23	0.38	26
2235	1.00	0.39	0.57	33
2236	0.00	0.00	0.00	33
2237	0.64	0.19	0.30	36
2238	1.00	0.16	0.27	38
2239	0.00	0.00	0.00	27
2240	0.93	0.37	0.53	35
2241	0.00	0.00	0.00	41
2242	0.50	0.03	0.06	30
2243	0.00	0.00	0.00	29
2244	0.00	0.00	0.00	37

2245	0.50	0.15	0.24	39
2246	0.00	0.00	0.00	29
2247	0.00	0.00	0.00	30
2248	0.00	0.00	0.00	37
2249	0.00	0.00	0.00	33
2250	0.50	0.04	0.07	27
2251	0.00	0.00	0.00	31
2252	0.00	0.00	0.00	27
2253	0.00	0.00	0.00	32
2254	0.73	0.23	0.35	35
2255	0.00	0.00	0.00	37
2256	0.00	0.00	0.00	33
2257	0.82	0.45	0.58	20
2258	0.00	0.00	0.00	28
2259	0.43	0.13	0.20	23
2260	0.00	0.00	0.00	31
2261	1.00	0.10	0.19	29
2262	0.60	0.12	0.19	26
2263	0.00	0.00	0.00	32
2264	0.00	0.00	0.00	35
2265	0.00	0.00	0.00	33
2266	0.67	0.23	0.34	35
2267	0.00	0.00	0.00	30
2268	0.50	0.05	0.08	22
2269	0.00	0.00	0.00	31
2270	0.00	0.00	0.00	32
2271	0.00	0.00	0.00	28
2272	0.83	0.19	0.31	26
2273	0.00	0.00	0.00	27
2274	0.00	0.00	0.00	33
2275	0.00	0.00	0.00	33
2276	0.50	0.09	0.15	22
2277	0.00	0.00	0.00	33
2278	0.00	0.00	0.00	36
2279	1.00	0.32	0.49	34
2280	0.00	0.00	0.00	24
2281	0.00	0.00	0.00	26
2282	0.40	0.09	0.15	22
2283	0.20	0.04	0.06	28
2284	0.00	0.00	0.00	43
2285	0.00	0.00	0.00	31
2286	0.00	0.00	0.00	30
2287	0.00	0.00	0.00	32
2288	0.00	0.00	0.00	28
2289	0.88	0.19	0.31	37
2290	0.00	0.00	0.00	23
2291	0.00	0.00	0.00	33
2292	0.50	0.03	0.06	33
2293	0.00	0.00	0.00	29
2294	0.00	0.00	0.00	28
2295	0.00	0.00	0.00	29
2296	0.00	0.00	0.00	24
2297	0.00	0.00	0.00	28
2298	1.00	0.15	0.27	26
2299	0.00	0.00	0.00	28
2300	1.00	0.10	0.18	31
2301	0.00	0.00	0.00	28
2302	0.00	0.00	0.00	34
2303	0.50	0.04	0.07	27
2304	0.00	0.00	0.00	31
2305	0.00	0.00	0.00	38
2306	0.00	0.00	0.00	37
2307	0.83	0.36	0.50	28
2308	1.00	0.04	0.07	28
2309	0.00	0.00	0.00	26
2310	1.00	0.21	0.35	28
2311	0.00	0.00	0.00	29
2312	1.00	0.11	0.19	38
2313	0.50	0.04	0.07	25
2314	1.00	0.05	0.09	22
2315	0.00	0.00	0.00	33
2316	0.00	0.00	0.00	30
2317	0.00	0.00	0.00	37
2318	0.00	0.00	0.00	26
2319	0.20	0.05	0.08	21
2320	0.00	0.00	0.00	29
2321	0.00	0.00	0.00	23
2322	0.00	0.00	0.00	33
2323	0.00	0.00	0.00	29
2324	0.00	0.00	0.00	29
2325	0.40	0.10	0.15	21
2326	0.00	0.00	0.00	36
2327	0.00	0.00	0.00	34

2328	0.00	0.00	0.00	25
2329	1.00	0.07	0.13	28
2330	0.00	0.00	0.00	30
2331	0.79	0.38	0.51	29
2332	0.00	0.00	0.00	32
2333	0.00	0.00	0.00	34
2334	0.50	0.03	0.06	30
2335	0.00	0.00	0.00	29
2336	1.00	0.03	0.06	30
2337	0.00	0.00	0.00	26
2338	0.92	0.40	0.56	30
2339	0.00	0.00	0.00	35
2340	0.00	0.00	0.00	26
2341	0.00	0.00	0.00	33
2342	1.00	0.15	0.27	39
2343	0.80	0.15	0.26	26
2344	0.00	0.00	0.00	39
2345	0.00	0.00	0.00	36
2346	0.00	0.00	0.00	37
2347	0.00	0.00	0.00	18
2348	0.60	0.10	0.17	31
2349	0.50	0.05	0.09	20
2350	0.00	0.00	0.00	32
2351	0.00	0.00	0.00	32
2352	0.00	0.00	0.00	28
2353	0.00	0.00	0.00	22
2354	0.92	0.33	0.49	36
2355	0.67	0.06	0.11	33
2356	0.00	0.00	0.00	31
2357	0.60	0.09	0.16	32
2358	0.12	0.05	0.07	19
2359	0.00	0.00	0.00	29
2360	0.00	0.00	0.00	27
2361	0.00	0.00	0.00	25
2362	1.00	0.04	0.08	24
2363	0.00	0.00	0.00	35
2364	0.00	0.00	0.00	32
2365	0.00	0.00	0.00	39
2366	0.00	0.00	0.00	32
2367	0.00	0.00	0.00	31
2368	0.00	0.00	0.00	32
2369	0.00	0.00	0.00	29
2370	0.00	0.00	0.00	32
2371	0.00	0.00	0.00	31
2372	0.00	0.00	0.00	32
2373	0.67	0.06	0.12	31
2374	0.00	0.00	0.00	30
2375	0.00	0.00	0.00	20
2376	0.83	0.18	0.29	28
2377	0.00	0.00	0.00	35
2378	0.00	0.00	0.00	24
2379	1.00	0.04	0.08	23
2380	0.00	0.00	0.00	31
2381	0.67	0.05	0.10	38
2382	0.00	0.00	0.00	26
2383	0.00	0.00	0.00	33
2384	0.00	0.00	0.00	36
2385	0.00	0.00	0.00	24
2386	0.54	0.33	0.41	21
2387	0.00	0.00	0.00	28
2388	0.00	0.00	0.00	22
2389	1.00	0.18	0.30	28
2390	0.88	0.20	0.33	35
2391	0.00	0.00	0.00	23
2392	0.00	0.00	0.00	27
2393	0.00	0.00	0.00	24
2394	1.00	0.43	0.61	23
2395	0.00	0.00	0.00	24
2396	1.00	0.03	0.06	31
2397	0.00	0.00	0.00	28
2398	0.00	0.00	0.00	35
2399	0.40	0.08	0.13	25
2400	0.00	0.00	0.00	33
2401	0.00	0.00	0.00	22
2402	0.25	0.03	0.05	36
2403	0.00	0.00	0.00	29
2404	0.50	0.08	0.13	26
2405	0.00	0.00	0.00	26
2406	0.58	0.42	0.49	26
2407	1.00	0.04	0.07	26
2408	1.00	0.03	0.06	32
2409	0.00	0.00	0.00	29
2410	0.00	0.00	0.00	26

2411	0.00	0.00	0.00	30
2412	0.00	0.00	0.00	30
2413	0.00	0.00	0.00	29
2414	0.00	0.00	0.00	33
2415	0.00	0.00	0.00	22
2416	0.00	0.00	0.00	27
2417	0.50	0.09	0.15	22
2418	0.00	0.00	0.00	33
2419	1.00	0.03	0.07	29
2420	0.00	0.00	0.00	38
2421	0.00	0.00	0.00	28
2422	0.00	0.00	0.00	25
2423	0.78	0.32	0.45	22
2424	0.50	0.03	0.05	35
2425	1.00	0.11	0.19	28
2426	0.50	0.03	0.06	34
2427	0.00	0.00	0.00	23
2428	0.00	0.00	0.00	30
2429	0.00	0.00	0.00	21
2430	0.00	0.00	0.00	26
2431	0.50	0.04	0.08	23
2432	0.00	0.00	0.00	33
2433	0.00	0.00	0.00	26
2434	0.78	0.48	0.60	29
2435	0.00	0.00	0.00	29
2436	0.00	0.00	0.00	29
2437	0.00	0.00	0.00	27
2438	0.00	0.00	0.00	26
2439	0.00	0.00	0.00	27
2440	0.00	0.00	0.00	28
2441	1.00	0.33	0.50	30
2442	0.00	0.00	0.00	26
2443	0.00	0.00	0.00	27
2444	0.00	0.00	0.00	30
2445	1.00	0.42	0.59	24
2446	0.00	0.00	0.00	21
2447	0.80	0.13	0.22	31
2448	1.00	0.04	0.08	23
2449	0.00	0.00	0.00	34
2450	0.00	0.00	0.00	33
2451	0.00	0.00	0.00	27
2452	1.00	0.07	0.13	29
2453	0.75	0.10	0.18	29
2454	0.00	0.00	0.00	28
2455	0.17	0.04	0.06	27
2456	0.00	0.00	0.00	25
2457	0.00	0.00	0.00	26
2458	0.71	0.16	0.26	31
2459	0.00	0.00	0.00	31
2460	0.00	0.00	0.00	30
2461	1.00	0.18	0.30	28
2462	0.67	0.07	0.12	30
2463	0.00	0.00	0.00	33
2464	0.00	0.00	0.00	29
2465	0.00	0.00	0.00	19
2466	0.00	0.00	0.00	25
2467	0.00	0.00	0.00	32
2468	0.00	0.00	0.00	29
2469	0.00	0.00	0.00	23
2470	0.92	0.41	0.56	27
2471	0.00	0.00	0.00	19
2472	0.00	0.00	0.00	25
2473	0.00	0.00	0.00	31
2474	0.00	0.00	0.00	27
2475	0.00	0.00	0.00	25
2476	0.92	0.37	0.52	30
2477	0.00	0.00	0.00	32
2478	0.67	0.07	0.13	28
2479	0.00	0.00	0.00	32
2480	0.00	0.00	0.00	36
2481	0.00	0.00	0.00	30
2482	0.00	0.00	0.00	23
2483	0.00	0.00	0.00	29
2484	0.62	0.22	0.32	23
2485	0.00	0.00	0.00	20
2486	0.00	0.00	0.00	24
2487	0.00	0.00	0.00	26
2488	0.00	0.00	0.00	27
2489	1.00	0.03	0.06	32
2490	0.00	0.00	0.00	32
2491	0.00	0.00	0.00	24
2492	0.50	0.19	0.27	27
2493	0.00	0.00	0.00	26

2494	0.00	0.00	0.00	24
2495	0.00	0.00	0.00	28
2496	0.00	0.00	0.00	20
2497	0.50	0.03	0.06	29
2498	1.00	0.18	0.30	34
2499	0.92	0.44	0.59	25
2500	0.00	0.00	0.00	30
2501	0.00	0.00	0.00	27
2502	0.50	0.14	0.22	28
2503	0.00	0.00	0.00	22
2504	0.00	0.00	0.00	26
2505	0.00	0.00	0.00	28
2506	0.33	0.04	0.08	23
2507	0.00	0.00	0.00	17
2508	0.00	0.00	0.00	25
2509	0.00	0.00	0.00	34
2510	0.00	0.00	0.00	24
2511	0.40	0.11	0.17	19
2512	0.00	0.00	0.00	27
2513	0.00	0.00	0.00	30
2514	0.75	0.12	0.21	24
2515	0.00	0.00	0.00	26
2516	0.00	0.00	0.00	18
2517	0.00	0.00	0.00	36
2518	1.00	0.03	0.06	30
2519	0.00	0.00	0.00	31
2520	0.00	0.00	0.00	33
2521	1.00	0.33	0.50	21
2522	0.00	0.00	0.00	12
2523	0.00	0.00	0.00	27
2524	0.89	0.35	0.50	23
2525	0.00	0.00	0.00	31
2526	0.00	0.00	0.00	35
2527	0.00	0.00	0.00	30
2528	0.00	0.00	0.00	24
2529	0.87	0.33	0.47	40
2530	0.25	0.03	0.05	33
2531	0.00	0.00	0.00	17
2532	0.00	0.00	0.00	29
2533	0.00	0.00	0.00	24
2534	1.00	0.07	0.13	28
2535	0.00	0.00	0.00	26
2536	0.00	0.00	0.00	26
2537	0.00	0.00	0.00	31
2538	0.00	0.00	0.00	28
2539	0.00	0.00	0.00	18
2540	0.67	0.20	0.31	30
2541	1.00	0.07	0.13	29
2542	0.00	0.00	0.00	23
2543	0.75	0.09	0.17	32
2544	1.00	0.19	0.31	27
2545	1.00	0.08	0.15	38
2546	1.00	0.04	0.07	26
2547	0.00	0.00	0.00	31
2548	0.00	0.00	0.00	27
2549	0.00	0.00	0.00	31
2550	0.67	0.08	0.14	26
2551	0.45	0.24	0.31	21
2552	0.00	0.00	0.00	28
2553	0.00	0.00	0.00	31
2554	0.67	0.11	0.18	19
2555	1.00	0.17	0.30	23
2556	0.60	0.39	0.47	23
2557	0.00	0.00	0.00	19
2558	0.00	0.00	0.00	23
2559	0.00	0.00	0.00	26
2560	0.00	0.00	0.00	20
2561	0.14	0.06	0.08	17
2562	1.00	0.10	0.18	20
2563	0.80	0.16	0.27	25
2564	0.00	0.00	0.00	21
2565	0.00	0.00	0.00	28
2566	0.00	0.00	0.00	26
2567	0.00	0.00	0.00	30
2568	0.00	0.00	0.00	37
2569	0.75	0.27	0.40	22
2570	1.00	0.12	0.22	24
2571	0.00	0.00	0.00	20
2572	0.00	0.00	0.00	26
2573	1.00	0.07	0.12	30
2574	0.00	0.00	0.00	29
2575	0.00	0.00	0.00	28
2576	0.00	0.00	0.00	22

2577	0.00	0.00	0.00	25
2578	0.00	0.00	0.00	24
2579	0.00	0.00	0.00	29
2580	0.00	0.00	0.00	27
2581	0.00	0.00	0.00	29
2582	0.00	0.00	0.00	21
2583	1.00	0.13	0.23	23
2584	0.00	0.00	0.00	27
2585	0.86	0.70	0.78	27
2586	0.00	0.00	0.00	25
2587	1.00	0.21	0.34	29
2588	0.00	0.00	0.00	20
2589	0.00	0.00	0.00	28
2590	0.00	0.00	0.00	28
2591	0.00	0.00	0.00	29
2592	1.00	0.05	0.10	20
2593	0.00	0.00	0.00	31
2594	0.00	0.00	0.00	19
2595	0.00	0.00	0.00	31
2596	0.00	0.00	0.00	28
2597	0.67	0.06	0.11	32
2598	0.60	0.10	0.18	29
2599	0.00	0.00	0.00	20
2600	0.00	0.00	0.00	18
2601	0.00	0.00	0.00	14
2602	0.00	0.00	0.00	29
2603	0.25	0.04	0.07	26
2604	0.00	0.00	0.00	25
2605	0.00	0.00	0.00	23
2606	1.00	0.05	0.09	22
2607	0.00	0.00	0.00	25
2608	1.00	0.04	0.08	25
2609	0.00	0.00	0.00	30
2610	0.00	0.00	0.00	26
2611	0.00	0.00	0.00	26
2612	0.00	0.00	0.00	30
2613	0.00	0.00	0.00	28
2614	0.00	0.00	0.00	28
2615	0.00	0.00	0.00	32
2616	0.00	0.00	0.00	23
2617	0.00	0.00	0.00	21
2618	0.00	0.00	0.00	26
2619	0.00	0.00	0.00	29
2620	0.86	0.32	0.46	19
2621	0.00	0.00	0.00	28
2622	0.00	0.00	0.00	23
2623	0.00	0.00	0.00	26
2624	0.00	0.00	0.00	24
2625	0.00	0.00	0.00	24
2626	0.00	0.00	0.00	30
2627	0.00	0.00	0.00	28
2628	0.83	0.29	0.43	17
2629	0.00	0.00	0.00	31
2630	0.00	0.00	0.00	30
2631	0.00	0.00	0.00	33
2632	0.00	0.00	0.00	31
2633	0.86	0.16	0.27	37
2634	0.00	0.00	0.00	21
2635	0.00	0.00	0.00	30
2636	0.00	0.00	0.00	22
2637	0.00	0.00	0.00	24
2638	0.00	0.00	0.00	29
2639	0.00	0.00	0.00	29
2640	0.00	0.00	0.00	20
2641	0.00	0.00	0.00	27
2642	0.00	0.00	0.00	28
2643	0.00	0.00	0.00	29
2644	0.89	0.31	0.46	26
2645	0.00	0.00	0.00	22
2646	0.00	0.00	0.00	20
2647	0.67	0.07	0.13	27
2648	0.00	0.00	0.00	30
2649	0.00	0.00	0.00	19
2650	0.00	0.00	0.00	15
2651	0.00	0.00	0.00	32
2652	0.00	0.00	0.00	19
2653	0.00	0.00	0.00	28
2654	1.00	0.35	0.52	23
2655	0.00	0.00	0.00	27
2656	0.00	0.00	0.00	26
2657	0.00	0.00	0.00	31
2658	0.00	0.00	0.00	21
2659	0.50	0.04	0.07	28

2660	0.00	0.00	0.00	24
2661	0.00	0.00	0.00	18
2662	0.83	0.19	0.31	26
2663	0.00	0.00	0.00	26
2664	0.00	0.00	0.00	28
2665	0.00	0.00	0.00	22
2666	0.67	0.07	0.13	28
2667	0.00	0.00	0.00	31
2668	0.00	0.00	0.00	18
2669	0.00	0.00	0.00	32
2670	0.00	0.00	0.00	24
2671	0.00	0.00	0.00	22
2672	0.00	0.00	0.00	23
2673	0.93	0.56	0.70	25
2674	0.50	0.04	0.07	26
2675	1.00	0.13	0.23	23
2676	0.00	0.00	0.00	23
2677	0.00	0.00	0.00	24
2678	0.00	0.00	0.00	26
2679	0.00	0.00	0.00	19
2680	0.00	0.00	0.00	19
2681	0.00	0.00	0.00	21
2682	0.89	0.27	0.41	30
2683	0.00	0.00	0.00	28
2684	0.00	0.00	0.00	26
2685	0.00	0.00	0.00	23
2686	0.50	0.11	0.18	28
2687	0.00	0.00	0.00	21
2688	0.00	0.00	0.00	32
2689	0.00	0.00	0.00	27
2690	1.00	0.17	0.30	23
2691	0.00	0.00	0.00	23
2692	0.00	0.00	0.00	24
2693	0.00	0.00	0.00	24
2694	0.00	0.00	0.00	20
2695	0.00	0.00	0.00	29
2696	0.00	0.00	0.00	20
2697	0.80	0.15	0.26	26
2698	0.00	0.00	0.00	30
2699	0.00	0.00	0.00	20
2700	0.00	0.00	0.00	25
2701	1.00	0.04	0.08	23
2702	0.00	0.00	0.00	24
2703	0.40	0.08	0.14	24
2704	0.00	0.00	0.00	29
2705	0.00	0.00	0.00	36
2706	0.20	0.03	0.06	29
2707	0.00	0.00	0.00	25
2708	0.00	0.00	0.00	21
2709	0.67	0.07	0.13	28
2710	0.00	0.00	0.00	14
2711	0.00	0.00	0.00	28
2712	0.00	0.00	0.00	21
2713	0.00	0.00	0.00	33
2714	0.00	0.00	0.00	21
2715	0.50	0.04	0.08	23
2716	0.00	0.00	0.00	26
2717	0.00	0.00	0.00	22
2718	0.50	0.07	0.12	30
2719	0.00	0.00	0.00	25
2720	0.00	0.00	0.00	25
2721	0.00	0.00	0.00	23
2722	0.00	0.00	0.00	20
2723	0.00	0.00	0.00	29
2724	0.00	0.00	0.00	20
2725	0.78	0.33	0.47	21
2726	0.00	0.00	0.00	25
2727	0.00	0.00	0.00	27
2728	0.00	0.00	0.00	24
2729	1.00	0.33	0.50	15
2730	0.00	0.00	0.00	26
2731	0.00	0.00	0.00	28
2732	0.00	0.00	0.00	30
2733	0.00	0.00	0.00	35
2734	0.80	0.17	0.28	24
2735	0.00	0.00	0.00	17
2736	0.50	0.19	0.28	26
2737	0.00	0.00	0.00	22
2738	0.00	0.00	0.00	33
2739	0.00	0.00	0.00	29
2740	0.00	0.00	0.00	28
2741	1.00	0.33	0.50	27
2742	1.00	0.52	0.69	23

2743	0.00	0.00	0.00	23
2744	0.00	0.00	0.00	20
2745	0.00	0.00	0.00	28
2746	0.00	0.00	0.00	25
2747	0.00	0.00	0.00	22
2748	0.00	0.00	0.00	24
2749	0.00	0.00	0.00	28
2750	1.00	0.10	0.19	29
2751	0.00	0.00	0.00	25
2752	0.00	0.00	0.00	23
2753	0.00	0.00	0.00	30
2754	0.00	0.00	0.00	20
2755	0.00	0.00	0.00	23
2756	0.00	0.00	0.00	26
2757	1.00	0.06	0.11	18
2758	0.80	0.22	0.35	18
2759	0.00	0.00	0.00	23
2760	0.00	0.00	0.00	30
2761	0.00	0.00	0.00	18
2762	0.00	0.00	0.00	21
2763	0.00	0.00	0.00	20
2764	0.00	0.00	0.00	17
2765	0.00	0.00	0.00	28
2766	1.00	0.06	0.11	18
2767	0.00	0.00	0.00	24
2768	1.00	0.25	0.40	24
2769	0.00	0.00	0.00	23
2770	0.00	0.00	0.00	19
2771	0.00	0.00	0.00	23
2772	1.00	0.11	0.19	19
2773	0.00	0.00	0.00	19
2774	1.00	0.24	0.38	21
2775	0.00	0.00	0.00	19
2776	0.00	0.00	0.00	23
2777	0.00	0.00	0.00	29
2778	0.00	0.00	0.00	21
2779	0.00	0.00	0.00	20
2780	0.00	0.00	0.00	23
2781	0.00	0.00	0.00	26
2782	0.00	0.00	0.00	31
2783	0.00	0.00	0.00	24
2784	0.00	0.00	0.00	23
2785	0.00	0.00	0.00	17
2786	0.00	0.00	0.00	26
2787	0.00	0.00	0.00	27
2788	0.71	0.20	0.31	25
2789	0.00	0.00	0.00	21
2790	0.00	0.00	0.00	23
2791	0.00	0.00	0.00	29
2792	0.00	0.00	0.00	35
2793	0.00	0.00	0.00	18
2794	0.00	0.00	0.00	17
2795	0.00	0.00	0.00	21
2796	0.00	0.00	0.00	19
2797	1.00	0.05	0.09	21
2798	0.00	0.00	0.00	17
2799	0.00	0.00	0.00	22
2800	1.00	0.04	0.08	24
2801	0.50	0.11	0.17	19
2802	0.00	0.00	0.00	23
2803	0.00	0.00	0.00	17
2804	0.00	0.00	0.00	23
2805	0.00	0.00	0.00	22
2806	0.00	0.00	0.00	24
2807	0.00	0.00	0.00	18
2808	1.00	0.04	0.08	24
2809	1.00	0.04	0.08	24
2810	0.00	0.00	0.00	20
2811	0.00	0.00	0.00	20
2812	0.00	0.00	0.00	23
2813	0.00	0.00	0.00	24
2814	0.00	0.00	0.00	17
2815	0.00	0.00	0.00	26
2816	0.00	0.00	0.00	16
2817	0.00	0.00	0.00	23
2818	0.00	0.00	0.00	26
2819	0.25	0.07	0.11	14
2820	0.00	0.00	0.00	22
2821	1.00	0.10	0.17	21
2822	0.00	0.00	0.00	24
2823	0.00	0.00	0.00	18
2824	0.00	0.00	0.00	26
2825	0.00	0.00	0.00	18

2826	0.75	0.15	0.25	20
2827	0.00	0.00	0.00	17
2828	0.00	0.00	0.00	25
2829	1.00	0.04	0.07	28
2830	0.00	0.00	0.00	19
2831	0.00	0.00	0.00	25
2832	0.00	0.00	0.00	20
2833	0.00	0.00	0.00	21
2834	0.00	0.00	0.00	25
2835	1.00	0.17	0.29	18
2836	0.00	0.00	0.00	26
2837	0.00	0.00	0.00	31
2838	1.00	0.08	0.15	24
2839	0.00	0.00	0.00	21
2840	0.00	0.00	0.00	20
2841	0.00	0.00	0.00	28
2842	1.00	0.23	0.37	35
2843	1.00	0.16	0.27	19
2844	0.00	0.00	0.00	24
2845	0.00	0.00	0.00	21
2846	1.00	0.08	0.15	25
2847	0.00	0.00	0.00	23
2848	0.00	0.00	0.00	26
2849	0.00	0.00	0.00	30
2850	0.00	0.00	0.00	31
2851	1.00	0.16	0.27	19
2852	0.00	0.00	0.00	29
2853	0.00	0.00	0.00	27
2854	0.00	0.00	0.00	22
2855	0.00	0.00	0.00	27
2856	0.00	0.00	0.00	18
2857	0.00	0.00	0.00	18
2858	0.00	0.00	0.00	22
2859	0.00	0.00	0.00	19
2860	0.00	0.00	0.00	22
2861	0.00	0.00	0.00	21
2862	0.00	0.00	0.00	23
2863	0.00	0.00	0.00	24
2864	0.00	0.00	0.00	28
2865	0.00	0.00	0.00	18
2866	0.67	0.27	0.39	22
2867	0.00	0.00	0.00	28
2868	0.00	0.00	0.00	27
2869	0.00	0.00	0.00	24
2870	0.00	0.00	0.00	21
2871	0.00	0.00	0.00	22
2872	0.00	0.00	0.00	21
2873	0.00	0.00	0.00	26
2874	0.00	0.00	0.00	25
2875	1.00	0.05	0.09	21
2876	0.00	0.00	0.00	25
2877	0.00	0.00	0.00	22
2878	0.80	0.19	0.31	21
2879	1.00	0.11	0.20	27
2880	1.00	0.04	0.08	24
2881	0.00	0.00	0.00	26
2882	0.00	0.00	0.00	29
2883	0.00	0.00	0.00	26
2884	0.00	0.00	0.00	25
2885	0.33	0.05	0.09	19
2886	0.83	0.26	0.40	19
2887	0.00	0.00	0.00	18
2888	0.00	0.00	0.00	22
2889	0.00	0.00	0.00	20
2890	0.00	0.00	0.00	28
2891	0.00	0.00	0.00	34
2892	0.00	0.00	0.00	18
2893	0.00	0.00	0.00	26
2894	0.00	0.00	0.00	19
2895	0.00	0.00	0.00	26
2896	0.00	0.00	0.00	17
2897	0.00	0.00	0.00	25
2898	0.00	0.00	0.00	19
2899	0.00	0.00	0.00	19
2900	0.00	0.00	0.00	28
2901	0.00	0.00	0.00	27
2902	0.00	0.00	0.00	19
2903	0.00	0.00	0.00	26
2904	0.00	0.00	0.00	21
2905	1.00	0.16	0.27	19
2906	0.00	0.00	0.00	19
2907	1.00	0.20	0.33	20
2908	0.00	0.00	0.00	19

2909	0.00	0.00	0.00	23
2910	0.00	0.00	0.00	20
2911	0.00	0.00	0.00	24
2912	1.00	0.05	0.09	22
2913	0.00	0.00	0.00	21
2914	0.00	0.00	0.00	28
2915	0.00	0.00	0.00	20
2916	0.00	0.00	0.00	24
2917	0.00	0.00	0.00	23
2918	1.00	0.04	0.08	25
2919	0.00	0.00	0.00	18
2920	1.00	0.14	0.25	21
2921	0.00	0.00	0.00	28
2922	0.00	0.00	0.00	17
2923	0.00	0.00	0.00	17
2924	0.00	0.00	0.00	25
2925	0.00	0.00	0.00	18
2926	0.00	0.00	0.00	20
2927	0.00	0.00	0.00	22
2928	1.00	0.05	0.09	21
2929	0.00	0.00	0.00	15
2930	0.00	0.00	0.00	21
2931	0.00	0.00	0.00	25
2932	0.00	0.00	0.00	21
2933	0.00	0.00	0.00	12
2934	0.00	0.00	0.00	29
2935	0.00	0.00	0.00	29
2936	0.00	0.00	0.00	20
2937	0.67	0.09	0.16	22
2938	0.00	0.00	0.00	24
2939	1.00	0.16	0.28	31
2940	0.00	0.00	0.00	23
2941	0.00	0.00	0.00	24
2942	0.00	0.00	0.00	23
2943	0.00	0.00	0.00	22
2944	0.00	0.00	0.00	17
2945	0.00	0.00	0.00	22
2946	0.00	0.00	0.00	17
2947	0.00	0.00	0.00	27
2948	0.00	0.00	0.00	18
2949	0.00	0.00	0.00	23
2950	0.00	0.00	0.00	22
2951	0.80	0.21	0.33	19
2952	0.00	0.00	0.00	15
2953	1.00	0.16	0.27	19
2954	0.00	0.00	0.00	19
2955	0.00	0.00	0.00	17
2956	0.00	0.00	0.00	20
2957	1.00	0.06	0.12	16
2958	0.00	0.00	0.00	17
2959	0.00	0.00	0.00	24
2960	0.00	0.00	0.00	23
2961	0.00	0.00	0.00	28
2962	0.50	0.05	0.10	19
2963	0.00	0.00	0.00	17
2964	0.00	0.00	0.00	25
2965	0.00	0.00	0.00	24
2966	0.00	0.00	0.00	18
2967	0.00	0.00	0.00	22
2968	0.00	0.00	0.00	17
2969	0.00	0.00	0.00	16
2970	0.00	0.00	0.00	24
2971	0.00	0.00	0.00	25
2972	0.00	0.00	0.00	18
2973	0.00	0.00	0.00	24
2974	0.00	0.00	0.00	19
2975	0.00	0.00	0.00	27
2976	0.00	0.00	0.00	21
2977	0.67	0.09	0.15	23
2978	0.00	0.00	0.00	26
2979	0.00	0.00	0.00	22
2980	0.00	0.00	0.00	24
2981	0.00	0.00	0.00	19
2982	1.00	0.05	0.09	21
2983	0.00	0.00	0.00	23
2984	0.00	0.00	0.00	24
2985	1.00	0.09	0.16	23
2986	1.00	0.09	0.16	23
2987	0.00	0.00	0.00	25
2988	1.00	0.17	0.29	24
2989	0.00	0.00	0.00	17
2990	0.00	0.00	0.00	23
2991	0.00	0.00	0.00	27

2992	0.00	0.00	0.00	18
2993	1.00	0.21	0.35	19
2994	0.00	0.00	0.00	27
2995	0.40	0.08	0.13	25
2996	0.00	0.00	0.00	21
2997	0.00	0.00	0.00	16
2998	0.00	0.00	0.00	28
2999	0.00	0.00	0.00	25
3000	0.00	0.00	0.00	16
3001	0.00	0.00	0.00	23
3002	0.00	0.00	0.00	20
3003	0.00	0.00	0.00	28
3004	0.00	0.00	0.00	14
3005	1.00	0.05	0.09	21
3006	0.00	0.00	0.00	19
3007	0.00	0.00	0.00	26
3008	0.00	0.00	0.00	27
3009	0.50	0.04	0.07	26
3010	0.00	0.00	0.00	20
3011	0.00	0.00	0.00	21
3012	0.00	0.00	0.00	21
3013	0.00	0.00	0.00	15
3014	0.00	0.00	0.00	27
3015	0.67	0.11	0.18	19
3016	1.00	0.05	0.10	19
3017	0.00	0.00	0.00	20
3018	0.00	0.00	0.00	19
3019	1.00	0.06	0.12	16
3020	0.00	0.00	0.00	15
3021	0.50	0.06	0.10	18
3022	0.00	0.00	0.00	18
3023	0.00	0.00	0.00	21
3024	1.00	0.27	0.42	26
3025	0.00	0.00	0.00	18
3026	0.50	0.04	0.08	23
3027	0.00	0.00	0.00	28
3028	0.83	0.24	0.37	21
3029	0.75	0.14	0.23	22
3030	0.00	0.00	0.00	21
3031	0.00	0.00	0.00	19
3032	0.00	0.00	0.00	23
3033	0.00	0.00	0.00	21
3034	0.00	0.00	0.00	17
3035	0.00	0.00	0.00	20
3036	0.67	0.10	0.17	21
3037	0.00	0.00	0.00	26
3038	0.00	0.00	0.00	27
3039	0.00	0.00	0.00	21
3040	0.00	0.00	0.00	19
3041	0.00	0.00	0.00	20
3042	0.00	0.00	0.00	24
3043	0.00	0.00	0.00	28
3044	0.00	0.00	0.00	18
3045	0.00	0.00	0.00	26
3046	0.00	0.00	0.00	26
3047	0.00	0.00	0.00	23
3048	0.00	0.00	0.00	18
3049	0.00	0.00	0.00	23
3050	1.00	0.18	0.30	17
3051	0.50	0.04	0.07	26
3052	0.00	0.00	0.00	32
3053	0.00	0.00	0.00	24
3054	0.00	0.00	0.00	16
3055	0.00	0.00	0.00	21
3056	0.00	0.00	0.00	23
3057	0.00	0.00	0.00	28
3058	0.00	0.00	0.00	13
3059	0.00	0.00	0.00	17
3060	0.00	0.00	0.00	15
3061	0.00	0.00	0.00	19
3062	0.00	0.00	0.00	18
3063	0.00	0.00	0.00	18
3064	0.00	0.00	0.00	22
3065	0.00	0.00	0.00	16
3066	0.00	0.00	0.00	18
3067	0.00	0.00	0.00	18
3068	0.00	0.00	0.00	22
3069	0.00	0.00	0.00	27
3070	0.00	0.00	0.00	23
3071	0.00	0.00	0.00	16
3072	0.00	0.00	0.00	24
3073	1.00	0.50	0.67	20
3074	0.00	0.00	0.00	22

3075	1.00	0.04	0.08	25
3076	0.00	0.00	0.00	18
3077	0.00	0.00	0.00	21
3078	0.00	0.00	0.00	18
3079	0.00	0.00	0.00	15
3080	1.00	0.07	0.12	15
3081	0.00	0.00	0.00	20
3082	0.00	0.00	0.00	23
3083	0.00	0.00	0.00	17
3084	0.00	0.00	0.00	16
3085	0.00	0.00	0.00	25
3086	0.00	0.00	0.00	13
3087	0.00	0.00	0.00	24
3088	0.00	0.00	0.00	22
3089	0.00	0.00	0.00	25
3090	0.00	0.00	0.00	21
3091	0.00	0.00	0.00	15
3092	0.00	0.00	0.00	19
3093	0.00	0.00	0.00	21
3094	0.00	0.00	0.00	22
3095	0.00	0.00	0.00	22
3096	0.00	0.00	0.00	26
3097	0.00	0.00	0.00	23
3098	0.00	0.00	0.00	22
3099	0.00	0.00	0.00	17
3100	1.00	0.22	0.36	18
3101	0.00	0.00	0.00	19
3102	0.00	0.00	0.00	15
3103	0.00	0.00	0.00	17
3104	0.00	0.00	0.00	20
3105	0.00	0.00	0.00	16
3106	0.00	0.00	0.00	14
3107	0.00	0.00	0.00	22
3108	0.00	0.00	0.00	24
3109	0.00	0.00	0.00	20
3110	0.00	0.00	0.00	19
3111	0.00	0.00	0.00	23
3112	0.00	0.00	0.00	21
3113	0.00	0.00	0.00	19
3114	0.00	0.00	0.00	18
3115	0.00	0.00	0.00	22
3116	0.00	0.00	0.00	19
3117	0.00	0.00	0.00	20
3118	0.00	0.00	0.00	18
3119	0.00	0.00	0.00	23
3120	0.00	0.00	0.00	18
3121	0.00	0.00	0.00	19
3122	1.00	0.19	0.32	16
3123	0.00	0.00	0.00	20
3124	0.50	0.05	0.08	22
3125	0.17	0.07	0.10	14
3126	0.00	0.00	0.00	16
3127	0.00	0.00	0.00	18
3128	0.00	0.00	0.00	33
3129	0.00	0.00	0.00	19
3130	0.00	0.00	0.00	28
3131	0.00	0.00	0.00	22
3132	0.00	0.00	0.00	20
3133	0.25	0.06	0.10	17
3134	0.00	0.00	0.00	19
3135	0.00	0.00	0.00	20
3136	0.00	0.00	0.00	20
3137	0.00	0.00	0.00	21
3138	0.00	0.00	0.00	21
3139	0.00	0.00	0.00	22
3140	0.00	0.00	0.00	18
3141	0.00	0.00	0.00	15
3142	0.00	0.00	0.00	20
3143	0.00	0.00	0.00	17
3144	0.00	0.00	0.00	23
3145	0.00	0.00	0.00	19
3146	0.00	0.00	0.00	17
3147	1.00	0.31	0.48	16
3148	0.80	0.50	0.62	16
3149	0.00	0.00	0.00	23
3150	0.00	0.00	0.00	25
3151	0.00	0.00	0.00	25
3152	0.00	0.00	0.00	26
3153	0.00	0.00	0.00	27
3154	0.00	0.00	0.00	20
3155	1.00	0.33	0.50	18
3156	0.00	0.00	0.00	17
3157	0.75	0.21	0.33	14

3158	0.00	0.00	0.00	23
3159	0.00	0.00	0.00	19
3160	0.50	0.05	0.09	20
3161	0.00	0.00	0.00	18
3162	0.00	0.00	0.00	19
3163	0.00	0.00	0.00	21
3164	0.00	0.00	0.00	16
3165	0.00	0.00	0.00	22
3166	0.00	0.00	0.00	19
3167	0.00	0.00	0.00	21
3168	0.00	0.00	0.00	27
3169	0.00	0.00	0.00	21
3170	0.00	0.00	0.00	23
3171	0.00	0.00	0.00	15
3172	0.00	0.00	0.00	24
3173	0.00	0.00	0.00	18
3174	0.00	0.00	0.00	21
3175	0.00	0.00	0.00	14
3176	0.00	0.00	0.00	19
3177	0.00	0.00	0.00	22
3178	0.00	0.00	0.00	20
3179	0.00	0.00	0.00	18
3180	0.00	0.00	0.00	20
3181	0.00	0.00	0.00	27
3182	0.00	0.00	0.00	23
3183	0.00	0.00	0.00	13
3184	0.00	0.00	0.00	22
3185	0.00	0.00	0.00	20
3186	0.00	0.00	0.00	28
3187	0.00	0.00	0.00	19
3188	0.00	0.00	0.00	23
3189	0.00	0.00	0.00	25
3190	0.00	0.00	0.00	21
3191	0.00	0.00	0.00	20
3192	0.00	0.00	0.00	22
3193	0.00	0.00	0.00	21
3194	0.00	0.00	0.00	16
3195	0.00	0.00	0.00	21
3196	0.00	0.00	0.00	21
3197	1.00	0.05	0.10	20
3198	0.00	0.00	0.00	18
3199	0.00	0.00	0.00	23
3200	0.33	0.05	0.09	19
3201	1.00	0.06	0.11	18
3202	0.00	0.00	0.00	25
3203	0.00	0.00	0.00	21
3204	1.00	0.07	0.12	15
3205	0.00	0.00	0.00	18
3206	0.00	0.00	0.00	23
3207	0.00	0.00	0.00	15
3208	0.00	0.00	0.00	20
3209	0.00	0.00	0.00	21
3210	0.00	0.00	0.00	20
3211	0.00	0.00	0.00	22
3212	0.00	0.00	0.00	21
3213	0.00	0.00	0.00	22
3214	0.00	0.00	0.00	25
3215	0.00	0.00	0.00	16
3216	0.00	0.00	0.00	7
3217	1.00	0.18	0.30	17
3218	0.00	0.00	0.00	26
3219	0.00	0.00	0.00	19
3220	0.00	0.00	0.00	29
3221	0.00	0.00	0.00	25
3222	0.00	0.00	0.00	14
3223	1.00	0.12	0.21	17
3224	0.00	0.00	0.00	23
3225	0.00	0.00	0.00	22
3226	0.00	0.00	0.00	20
3227	0.00	0.00	0.00	24
3228	0.00	0.00	0.00	17
3229	0.00	0.00	0.00	31
3230	0.00	0.00	0.00	21
3231	0.00	0.00	0.00	22
3232	0.00	0.00	0.00	15
3233	0.00	0.00	0.00	21
3234	0.00	0.00	0.00	23
3235	0.00	0.00	0.00	21
3236	0.00	0.00	0.00	14
3237	0.00	0.00	0.00	21
3238	0.00	0.00	0.00	17
3239	0.00	0.00	0.00	22
3240	0.00	0.00	0.00	22

3241	0.00	0.00	0.00	15
3242	0.00	0.00	0.00	21
3243	0.00	0.00	0.00	15
3244	0.00	0.00	0.00	29
3245	0.00	0.00	0.00	17
3246	0.00	0.00	0.00	22
3247	0.00	0.00	0.00	25
3248	0.00	0.00	0.00	20
3249	0.00	0.00	0.00	22
3250	0.00	0.00	0.00	24
3251	0.00	0.00	0.00	19
3252	0.00	0.00	0.00	17
3253	0.00	0.00	0.00	16
3254	0.00	0.00	0.00	25
3255	0.00	0.00	0.00	15
3256	0.00	0.00	0.00	17
3257	0.00	0.00	0.00	15
3258	0.00	0.00	0.00	21
3259	0.00	0.00	0.00	14
3260	0.00	0.00	0.00	18
3261	0.00	0.00	0.00	24
3262	0.00	0.00	0.00	20
3263	0.00	0.00	0.00	16
3264	1.00	0.05	0.10	19
3265	0.00	0.00	0.00	21
3266	0.00	0.00	0.00	20
3267	0.00	0.00	0.00	22
3268	0.00	0.00	0.00	13
3269	0.00	0.00	0.00	18
3270	0.00	0.00	0.00	15
3271	0.00	0.00	0.00	19
3272	0.00	0.00	0.00	25
3273	0.00	0.00	0.00	18
3274	0.00	0.00	0.00	22
3275	0.00	0.00	0.00	23
3276	0.00	0.00	0.00	17
3277	0.00	0.00	0.00	20
3278	0.00	0.00	0.00	22
3279	0.00	0.00	0.00	21
3280	0.00	0.00	0.00	19
3281	0.00	0.00	0.00	18
3282	0.00	0.00	0.00	20
3283	0.00	0.00	0.00	15
3284	0.00	0.00	0.00	17
3285	0.00	0.00	0.00	20
3286	0.00	0.00	0.00	11
3287	0.00	0.00	0.00	16
3288	0.00	0.00	0.00	14
3289	0.00	0.00	0.00	27
3290	0.00	0.00	0.00	26
3291	0.00	0.00	0.00	24
3292	0.00	0.00	0.00	19
3293	0.00	0.00	0.00	15
3294	1.00	0.05	0.09	22
3295	0.00	0.00	0.00	19
3296	0.00	0.00	0.00	26
3297	0.00	0.00	0.00	22
3298	0.00	0.00	0.00	16
3299	0.00	0.00	0.00	19
3300	0.00	0.00	0.00	16
3301	1.00	0.05	0.10	19
3302	1.00	0.06	0.11	17
3303	0.00	0.00	0.00	17
3304	0.00	0.00	0.00	16
3305	0.00	0.00	0.00	26
3306	0.00	0.00	0.00	16
3307	0.00	0.00	0.00	21
3308	0.00	0.00	0.00	15
3309	0.00	0.00	0.00	14
3310	0.00	0.00	0.00	16
3311	0.00	0.00	0.00	26
3312	0.00	0.00	0.00	21
3313	0.00	0.00	0.00	17
3314	0.00	0.00	0.00	20
3315	0.00	0.00	0.00	18
3316	0.00	0.00	0.00	20
3317	0.00	0.00	0.00	20
3318	0.00	0.00	0.00	19
3319	0.00	0.00	0.00	11
3320	0.00	0.00	0.00	17
3321	0.00	0.00	0.00	21
3322	0.00	0.00	0.00	20
3323	0.00	0.00	0.00	19

3324	1.00	0.12	0.21	17
3325	0.00	0.00	0.00	13
3326	0.00	0.00	0.00	18
3327	0.00	0.00	0.00	15
3328	1.00	0.04	0.08	24
3329	0.00	0.00	0.00	23
3330	1.00	0.25	0.40	12
3331	0.33	0.06	0.11	16
3332	0.00	0.00	0.00	19
3333	0.00	0.00	0.00	23
3334	0.00	0.00	0.00	21
3335	0.00	0.00	0.00	12
3336	0.00	0.00	0.00	16
3337	0.00	0.00	0.00	8
3338	0.00	0.00	0.00	21
3339	0.00	0.00	0.00	22
3340	0.00	0.00	0.00	23
3341	0.00	0.00	0.00	14
3342	0.00	0.00	0.00	26
3343	0.00	0.00	0.00	19
3344	0.00	0.00	0.00	10
3345	0.00	0.00	0.00	22
3346	0.00	0.00	0.00	19
3347	0.00	0.00	0.00	21
3348	0.00	0.00	0.00	17
3349	0.00	0.00	0.00	20
3350	0.00	0.00	0.00	21
3351	0.00	0.00	0.00	21
3352	0.00	0.00	0.00	16
3353	0.00	0.00	0.00	19
3354	0.00	0.00	0.00	15
3355	0.00	0.00	0.00	19
3356	0.00	0.00	0.00	14
3357	0.00	0.00	0.00	17
3358	0.00	0.00	0.00	19
3359	0.00	0.00	0.00	17
3360	0.00	0.00	0.00	11
3361	0.00	0.00	0.00	20
3362	0.00	0.00	0.00	18
3363	0.00	0.00	0.00	23
3364	0.00	0.00	0.00	19
3365	0.00	0.00	0.00	15
3366	0.00	0.00	0.00	28
3367	1.00	0.06	0.12	16
3368	0.00	0.00	0.00	12
3369	0.00	0.00	0.00	16
3370	0.00	0.00	0.00	18
3371	0.00	0.00	0.00	24
3372	0.00	0.00	0.00	22
3373	0.00	0.00	0.00	12
3374	0.00	0.00	0.00	23
3375	0.00	0.00	0.00	23
3376	0.00	0.00	0.00	22
3377	0.00	0.00	0.00	16
3378	0.00	0.00	0.00	16
3379	0.00	0.00	0.00	14
3380	0.00	0.00	0.00	21
3381	0.00	0.00	0.00	17
3382	0.00	0.00	0.00	19
3383	0.00	0.00	0.00	16
3384	0.00	0.00	0.00	18
3385	0.00	0.00	0.00	10
3386	0.00	0.00	0.00	28
3387	0.00	0.00	0.00	18
3388	0.00	0.00	0.00	16
3389	1.00	0.06	0.12	16
3390	0.00	0.00	0.00	8
3391	0.00	0.00	0.00	24
3392	0.00	0.00	0.00	17
3393	0.00	0.00	0.00	15
3394	1.00	0.25	0.40	20
3395	0.00	0.00	0.00	23
3396	0.00	0.00	0.00	14
3397	0.00	0.00	0.00	13
3398	0.00	0.00	0.00	19
3399	0.00	0.00	0.00	21
3400	0.00	0.00	0.00	18
3401	0.00	0.00	0.00	22
3402	0.00	0.00	0.00	15
3403	0.00	0.00	0.00	15
3404	0.33	0.10	0.15	10
3405	0.00	0.00	0.00	19
3406	0.00	0.00	0.00	25

3407	0.00	0.00	0.00	19
3408	0.00	0.00	0.00	16
3409	0.00	0.00	0.00	19
3410	0.00	0.00	0.00	21
3411	0.00	0.00	0.00	16
3412	0.00	0.00	0.00	16
3413	0.00	0.00	0.00	12
3414	0.00	0.00	0.00	16
3415	0.00	0.00	0.00	19
3416	0.00	0.00	0.00	19
3417	0.00	0.00	0.00	19
3418	0.00	0.00	0.00	8
3419	0.00	0.00	0.00	20
3420	0.00	0.00	0.00	23
3421	0.00	0.00	0.00	12
3422	0.00	0.00	0.00	22
3423	0.00	0.00	0.00	20
3424	0.00	0.00	0.00	21
3425	0.00	0.00	0.00	16
3426	0.00	0.00	0.00	21
3427	0.00	0.00	0.00	17
3428	0.00	0.00	0.00	12
3429	0.00	0.00	0.00	15
3430	0.00	0.00	0.00	22
3431	0.00	0.00	0.00	16
3432	0.00	0.00	0.00	15
3433	0.00	0.00	0.00	16
3434	0.00	0.00	0.00	16
3435	0.00	0.00	0.00	21
3436	0.00	0.00	0.00	16
3437	0.00	0.00	0.00	14
3438	0.00	0.00	0.00	19
3439	0.00	0.00	0.00	12
3440	0.00	0.00	0.00	17
3441	0.00	0.00	0.00	16
3442	0.00	0.00	0.00	16
3443	0.00	0.00	0.00	15
3444	0.00	0.00	0.00	14
3445	0.00	0.00	0.00	21
3446	0.00	0.00	0.00	20
3447	0.00	0.00	0.00	23
3448	0.00	0.00	0.00	13
3449	0.00	0.00	0.00	19
3450	0.00	0.00	0.00	20
3451	0.00	0.00	0.00	11
3452	0.00	0.00	0.00	13
3453	0.00	0.00	0.00	21
3454	0.00	0.00	0.00	20
3455	0.00	0.00	0.00	11
3456	0.00	0.00	0.00	20
3457	0.00	0.00	0.00	16
3458	0.00	0.00	0.00	19
3459	0.00	0.00	0.00	14
3460	0.00	0.00	0.00	20
3461	0.00	0.00	0.00	19
3462	0.00	0.00	0.00	21
3463	0.00	0.00	0.00	20
3464	0.00	0.00	0.00	14
3465	0.00	0.00	0.00	13
3466	0.00	0.00	0.00	20
3467	0.00	0.00	0.00	22
3468	0.00	0.00	0.00	18
3469	0.00	0.00	0.00	14
3470	0.00	0.00	0.00	18
3471	0.00	0.00	0.00	17
3472	0.00	0.00	0.00	18
3473	0.00	0.00	0.00	15
3474	0.00	0.00	0.00	20
3475	1.00	0.16	0.27	19
3476	0.00	0.00	0.00	15
3477	0.00	0.00	0.00	11
3478	0.00	0.00	0.00	19
3479	0.00	0.00	0.00	16
3480	0.00	0.00	0.00	18
3481	0.00	0.00	0.00	14
3482	0.00	0.00	0.00	14
3483	0.00	0.00	0.00	20
3484	0.67	0.12	0.20	17
3485	0.00	0.00	0.00	16
3486	0.00	0.00	0.00	15
3487	0.00	0.00	0.00	21
3488	0.00	0.00	0.00	15
3489	0.00	0.00	0.00	21

3490	0.00	0.00	0.00	21
3491	0.00	0.00	0.00	19
3492	0.00	0.00	0.00	23
3493	1.00	0.12	0.21	17
3494	0.00	0.00	0.00	21
3495	0.00	0.00	0.00	11
3496	0.00	0.00	0.00	14
3497	0.00	0.00	0.00	15
3498	0.00	0.00	0.00	17
3499	0.00	0.00	0.00	19
3500	0.00	0.00	0.00	15
3501	0.00	0.00	0.00	20
3502	0.00	0.00	0.00	15
3503	0.00	0.00	0.00	19
3504	0.00	0.00	0.00	23
3505	0.50	0.06	0.11	16
3506	0.00	0.00	0.00	17
3507	0.00	0.00	0.00	20
3508	0.00	0.00	0.00	11
3509	0.00	0.00	0.00	20
3510	0.00	0.00	0.00	15
3511	0.00	0.00	0.00	14
3512	0.00	0.00	0.00	14
3513	0.00	0.00	0.00	17
3514	0.00	0.00	0.00	20
3515	0.00	0.00	0.00	19
3516	0.00	0.00	0.00	18
3517	0.00	0.00	0.00	16
3518	0.00	0.00	0.00	15
3519	0.00	0.00	0.00	19
3520	0.00	0.00	0.00	17
3521	0.00	0.00	0.00	15
3522	0.00	0.00	0.00	23
3523	0.00	0.00	0.00	17
3524	0.00	0.00	0.00	21
3525	0.00	0.00	0.00	17
3526	0.00	0.00	0.00	12
3527	0.00	0.00	0.00	20
3528	0.00	0.00	0.00	25
3529	0.00	0.00	0.00	19
3530	0.00	0.00	0.00	9
3531	0.00	0.00	0.00	18
3532	0.00	0.00	0.00	17
3533	0.00	0.00	0.00	13
3534	0.00	0.00	0.00	19
3535	0.00	0.00	0.00	12
3536	0.00	0.00	0.00	20
3537	0.00	0.00	0.00	22
3538	0.00	0.00	0.00	12
3539	1.00	0.06	0.12	16
3540	0.00	0.00	0.00	14
3541	0.60	0.20	0.30	15
3542	0.00	0.00	0.00	17
3543	0.00	0.00	0.00	17
3544	0.00	0.00	0.00	17
3545	0.00	0.00	0.00	14
3546	0.00	0.00	0.00	14
3547	0.00	0.00	0.00	18
3548	0.00	0.00	0.00	21
3549	0.00	0.00	0.00	11
3550	0.00	0.00	0.00	13
3551	0.00	0.00	0.00	17
3552	0.00	0.00	0.00	12
3553	0.00	0.00	0.00	13
3554	0.00	0.00	0.00	16
3555	0.00	0.00	0.00	24
3556	0.00	0.00	0.00	8
3557	0.00	0.00	0.00	15
3558	0.00	0.00	0.00	13
3559	0.00	0.00	0.00	22
3560	0.00	0.00	0.00	15
3561	0.00	0.00	0.00	19
3562	0.00	0.00	0.00	16
3563	0.00	0.00	0.00	21
3564	0.00	0.00	0.00	19
3565	0.00	0.00	0.00	19
3566	0.00	0.00	0.00	16
3567	0.00	0.00	0.00	13
3568	0.00	0.00	0.00	20
3569	0.00	0.00	0.00	13
3570	0.00	0.00	0.00	16
3571	1.00	0.04	0.08	25
3572	0.00	0.00	0.00	18

3573	0.00	0.00	0.00	11
3574	0.00	0.00	0.00	19
3575	0.00	0.00	0.00	23
3576	0.00	0.00	0.00	12
3577	0.00	0.00	0.00	21
3578	0.00	0.00	0.00	16
3579	0.00	0.00	0.00	21
3580	0.00	0.00	0.00	17
3581	0.00	0.00	0.00	21
3582	0.00	0.00	0.00	13
3583	0.00	0.00	0.00	24
3584	0.00	0.00	0.00	18
3585	0.00	0.00	0.00	13
3586	0.00	0.00	0.00	14
3587	0.00	0.00	0.00	22
3588	0.00	0.00	0.00	14
3589	0.00	0.00	0.00	18
3590	0.00	0.00	0.00	23
3591	0.00	0.00	0.00	18
3592	0.00	0.00	0.00	11
3593	0.00	0.00	0.00	16
3594	1.00	0.25	0.40	12
3595	0.00	0.00	0.00	21
3596	0.00	0.00	0.00	17
3597	0.00	0.00	0.00	19
3598	0.00	0.00	0.00	13
3599	0.00	0.00	0.00	18
3600	0.00	0.00	0.00	17
3601	0.00	0.00	0.00	18
3602	1.00	0.08	0.14	13
3603	0.00	0.00	0.00	12
3604	0.00	0.00	0.00	18
3605	0.00	0.00	0.00	16
3606	0.00	0.00	0.00	15
3607	0.00	0.00	0.00	22
3608	0.00	0.00	0.00	21
3609	0.00	0.00	0.00	20
3610	0.00	0.00	0.00	17
3611	0.00	0.00	0.00	19
3612	0.00	0.00	0.00	13
3613	0.00	0.00	0.00	12
3614	0.00	0.00	0.00	18
3615	0.00	0.00	0.00	7
3616	0.00	0.00	0.00	23
3617	0.00	0.00	0.00	14
3618	0.00	0.00	0.00	21
3619	0.00	0.00	0.00	18
3620	0.00	0.00	0.00	20
3621	0.00	0.00	0.00	15
3622	0.00	0.00	0.00	17
3623	0.00	0.00	0.00	16
3624	0.00	0.00	0.00	18
3625	0.00	0.00	0.00	21
3626	1.00	0.25	0.40	12
3627	0.00	0.00	0.00	18
3628	0.50	0.07	0.12	14
3629	0.00	0.00	0.00	13
3630	0.00	0.00	0.00	10
3631	0.00	0.00	0.00	17
3632	0.00	0.00	0.00	8
3633	0.00	0.00	0.00	16
3634	0.00	0.00	0.00	19
3635	0.00	0.00	0.00	14
3636	0.00	0.00	0.00	13
3637	0.00	0.00	0.00	18
3638	0.00	0.00	0.00	23
3639	0.00	0.00	0.00	20
3640	0.00	0.00	0.00	17
3641	0.00	0.00	0.00	20
3642	0.50	0.09	0.15	11
3643	0.00	0.00	0.00	13
3644	0.00	0.00	0.00	19
3645	0.00	0.00	0.00	11
3646	0.33	0.08	0.12	13
3647	0.00	0.00	0.00	13
3648	0.00	0.00	0.00	19
3649	0.00	0.00	0.00	19
3650	0.00	0.00	0.00	12
3651	0.00	0.00	0.00	18
3652	0.00	0.00	0.00	18
3653	0.00	0.00	0.00	12
3654	0.00	0.00	0.00	20
3655	0.00	0.00	0.00	22

3656	0.00	0.00	0.00	19
3657	0.00	0.00	0.00	10
3658	0.00	0.00	0.00	15
3659	0.00	0.00	0.00	11
3660	0.00	0.00	0.00	15
3661	0.00	0.00	0.00	18
3662	0.00	0.00	0.00	18
3663	0.00	0.00	0.00	19
3664	0.00	0.00	0.00	12
3665	1.00	0.04	0.08	24
3666	0.00	0.00	0.00	18
3667	0.00	0.00	0.00	16
3668	0.00	0.00	0.00	12
3669	0.00	0.00	0.00	22
3670	0.00	0.00	0.00	19
3671	0.00	0.00	0.00	19
3672	0.00	0.00	0.00	19
3673	0.00	0.00	0.00	14
3674	0.00	0.00	0.00	18
3675	0.00	0.00	0.00	16
3676	0.00	0.00	0.00	12
3677	0.00	0.00	0.00	17
3678	0.00	0.00	0.00	20
3679	0.00	0.00	0.00	21
3680	0.00	0.00	0.00	22
3681	0.00	0.00	0.00	15
3682	0.00	0.00	0.00	17
3683	0.00	0.00	0.00	19
3684	0.00	0.00	0.00	13
3685	0.00	0.00	0.00	17
3686	0.00	0.00	0.00	18
3687	0.00	0.00	0.00	26
3688	0.00	0.00	0.00	20
3689	1.00	0.10	0.18	20
3690	0.00	0.00	0.00	22
3691	0.00	0.00	0.00	18
3692	0.00	0.00	0.00	15
3693	0.00	0.00	0.00	15
3694	0.40	0.14	0.21	14
3695	0.00	0.00	0.00	19
3696	0.00	0.00	0.00	13
3697	0.00	0.00	0.00	13
3698	0.00	0.00	0.00	16
3699	0.00	0.00	0.00	17
3700	0.00	0.00	0.00	19
3701	0.00	0.00	0.00	15
3702	0.00	0.00	0.00	23
3703	0.00	0.00	0.00	19
3704	0.00	0.00	0.00	12
3705	0.00	0.00	0.00	21
3706	0.00	0.00	0.00	17
3707	0.00	0.00	0.00	19
3708	0.00	0.00	0.00	19
3709	0.00	0.00	0.00	13
3710	0.00	0.00	0.00	13
3711	0.00	0.00	0.00	11
3712	0.00	0.00	0.00	18
3713	0.00	0.00	0.00	17
3714	0.00	0.00	0.00	18
3715	0.00	0.00	0.00	13
3716	0.00	0.00	0.00	21
3717	0.00	0.00	0.00	17
3718	0.00	0.00	0.00	13
3719	0.00	0.00	0.00	18
3720	0.00	0.00	0.00	11
3721	0.00	0.00	0.00	15
3722	0.00	0.00	0.00	12
3723	0.00	0.00	0.00	19
3724	0.00	0.00	0.00	12
3725	0.00	0.00	0.00	14
3726	0.00	0.00	0.00	16
3727	0.00	0.00	0.00	14
3728	0.00	0.00	0.00	19
3729	0.00	0.00	0.00	15
3730	0.00	0.00	0.00	12
3731	0.00	0.00	0.00	16
3732	0.00	0.00	0.00	17
3733	0.00	0.00	0.00	17
3734	0.00	0.00	0.00	16
3735	0.00	0.00	0.00	18
3736	0.00	0.00	0.00	15
3737	0.00	0.00	0.00	15
3738	0.00	0.00	0.00	15

3739	0.00	0.00	0.00	19
3740	0.00	0.00	0.00	16
3741	0.00	0.00	0.00	20
3742	0.00	0.00	0.00	15
3743	0.00	0.00	0.00	13
3744	1.00	0.15	0.27	13
3745	0.00	0.00	0.00	15
3746	0.00	0.00	0.00	16
3747	0.00	0.00	0.00	19
3748	0.00	0.00	0.00	11
3749	0.00	0.00	0.00	20
3750	0.00	0.00	0.00	17
3751	0.00	0.00	0.00	11
3752	0.00	0.00	0.00	13
3753	0.00	0.00	0.00	18
3754	0.00	0.00	0.00	17
3755	0.00	0.00	0.00	20
3756	0.00	0.00	0.00	16
3757	0.00	0.00	0.00	14
3758	0.00	0.00	0.00	14
3759	0.00	0.00	0.00	22
3760	0.00	0.00	0.00	15
3761	0.00	0.00	0.00	17
3762	0.00	0.00	0.00	17
3763	0.00	0.00	0.00	15
3764	1.00	0.21	0.35	19
3765	0.00	0.00	0.00	17
3766	0.00	0.00	0.00	7
3767	0.00	0.00	0.00	15
3768	0.00	0.00	0.00	12
3769	0.00	0.00	0.00	14
3770	0.00	0.00	0.00	15
3771	0.00	0.00	0.00	16
3772	0.00	0.00	0.00	15
3773	0.00	0.00	0.00	16
3774	0.00	0.00	0.00	17
3775	0.00	0.00	0.00	16
3776	0.00	0.00	0.00	11
3777	0.00	0.00	0.00	19
3778	0.00	0.00	0.00	22
3779	0.00	0.00	0.00	9
3780	1.00	0.15	0.27	13
3781	0.00	0.00	0.00	12
3782	0.00	0.00	0.00	23
3783	0.00	0.00	0.00	13
3784	0.00	0.00	0.00	15
3785	0.00	0.00	0.00	19
3786	0.00	0.00	0.00	17
3787	0.00	0.00	0.00	13
3788	0.00	0.00	0.00	18
3789	1.00	0.06	0.11	17
3790	0.00	0.00	0.00	14
3791	0.00	0.00	0.00	13
3792	0.00	0.00	0.00	18
3793	0.00	0.00	0.00	12
3794	0.00	0.00	0.00	22
3795	0.00	0.00	0.00	14
3796	0.00	0.00	0.00	23
3797	0.00	0.00	0.00	8
3798	0.00	0.00	0.00	23
3799	0.00	0.00	0.00	9
3800	0.00	0.00	0.00	17
3801	0.00	0.00	0.00	17
3802	0.00	0.00	0.00	14
3803	0.00	0.00	0.00	21
3804	0.00	0.00	0.00	15
3805	0.00	0.00	0.00	13
3806	0.00	0.00	0.00	13
3807	0.00	0.00	0.00	10
3808	0.00	0.00	0.00	14
3809	0.00	0.00	0.00	17
3810	0.00	0.00	0.00	21
3811	0.00	0.00	0.00	14
3812	0.00	0.00	0.00	18
3813	0.00	0.00	0.00	19
3814	0.00	0.00	0.00	16
3815	0.00	0.00	0.00	14
3816	0.00	0.00	0.00	14
3817	0.00	0.00	0.00	14
3818	0.00	0.00	0.00	15
3819	0.00	0.00	0.00	18
3820	0.00	0.00	0.00	16
3821	0.00	0.00	0.00	19

3822	0.00	0.00	0.00	21
3823	0.00	0.00	0.00	16
3824	0.00	0.00	0.00	17
3825	0.00	0.00	0.00	16
3826	0.00	0.00	0.00	20
3827	0.00	0.00	0.00	17
3828	0.00	0.00	0.00	17
3829	0.00	0.00	0.00	16
3830	0.00	0.00	0.00	19
3831	0.00	0.00	0.00	15
3832	0.00	0.00	0.00	20
3833	0.00	0.00	0.00	16
3834	0.00	0.00	0.00	13
3835	0.00	0.00	0.00	14
3836	0.00	0.00	0.00	12
3837	0.00	0.00	0.00	14
3838	0.00	0.00	0.00	9
3839	0.00	0.00	0.00	13
3840	0.00	0.00	0.00	14
3841	0.00	0.00	0.00	19
3842	0.00	0.00	0.00	19
3843	0.00	0.00	0.00	16
3844	0.00	0.00	0.00	13
3845	0.00	0.00	0.00	21
3846	0.00	0.00	0.00	7
3847	0.00	0.00	0.00	16
3848	0.00	0.00	0.00	10
3849	0.00	0.00	0.00	19
3850	0.00	0.00	0.00	18
3851	0.00	0.00	0.00	11
3852	0.00	0.00	0.00	17
3853	0.00	0.00	0.00	13
3854	0.00	0.00	0.00	20
3855	0.00	0.00	0.00	20
3856	0.00	0.00	0.00	10
3857	0.00	0.00	0.00	20
3858	0.00	0.00	0.00	22
3859	0.00	0.00	0.00	13
3860	0.00	0.00	0.00	19
3861	0.00	0.00	0.00	16
3862	0.00	0.00	0.00	18
3863	0.00	0.00	0.00	10
3864	1.00	0.15	0.27	13
3865	0.00	0.00	0.00	15
3866	0.00	0.00	0.00	13
3867	0.00	0.00	0.00	18
3868	0.00	0.00	0.00	13
3869	0.00	0.00	0.00	17
3870	0.00	0.00	0.00	14
3871	0.00	0.00	0.00	11
3872	0.00	0.00	0.00	10
3873	0.00	0.00	0.00	17
3874	0.00	0.00	0.00	9
3875	0.00	0.00	0.00	13
3876	0.00	0.00	0.00	12
3877	0.00	0.00	0.00	13
3878	0.00	0.00	0.00	16
3879	0.00	0.00	0.00	17
3880	0.00	0.00	0.00	11
3881	0.00	0.00	0.00	17
3882	0.00	0.00	0.00	13
3883	0.00	0.00	0.00	11
3884	0.00	0.00	0.00	15
3885	0.00	0.00	0.00	17
3886	0.00	0.00	0.00	14
3887	1.00	0.20	0.33	10
3888	0.00	0.00	0.00	16
3889	0.00	0.00	0.00	13
3890	0.00	0.00	0.00	14
3891	0.00	0.00	0.00	15
3892	0.00	0.00	0.00	19
3893	0.00	0.00	0.00	9
3894	0.00	0.00	0.00	16
3895	0.00	0.00	0.00	18
3896	0.00	0.00	0.00	17
3897	0.00	0.00	0.00	18
3898	0.00	0.00	0.00	10
3899	0.00	0.00	0.00	14
3900	0.00	0.00	0.00	22
3901	0.00	0.00	0.00	23
3902	0.00	0.00	0.00	11
3903	0.00	0.00	0.00	10
3904	0.00	0.00	0.00	7

3905	0.00	0.00	0.00	19
3906	1.00	0.13	0.24	15
3907	0.00	0.00	0.00	9
3908	0.00	0.00	0.00	12
3909	0.00	0.00	0.00	17
3910	0.00	0.00	0.00	11
3911	0.00	0.00	0.00	14
3912	0.00	0.00	0.00	18
3913	0.00	0.00	0.00	12
3914	0.00	0.00	0.00	15
3915	0.00	0.00	0.00	12
3916	0.00	0.00	0.00	14
3917	0.00	0.00	0.00	12
3918	0.00	0.00	0.00	11
3919	0.00	0.00	0.00	12
3920	0.00	0.00	0.00	24
3921	0.00	0.00	0.00	13
3922	0.00	0.00	0.00	15
3923	1.00	0.07	0.12	15
3924	0.00	0.00	0.00	10
3925	0.00	0.00	0.00	20
3926	0.00	0.00	0.00	15
3927	0.00	0.00	0.00	20
3928	0.00	0.00	0.00	11
3929	0.00	0.00	0.00	15
3930	0.00	0.00	0.00	8
3931	0.00	0.00	0.00	16
3932	0.00	0.00	0.00	15
3933	0.00	0.00	0.00	15
3934	0.00	0.00	0.00	17
3935	0.00	0.00	0.00	10
3936	0.00	0.00	0.00	21
3937	0.00	0.00	0.00	14
3938	0.00	0.00	0.00	19
3939	0.00	0.00	0.00	17
3940	0.00	0.00	0.00	19
3941	0.00	0.00	0.00	13
3942	0.00	0.00	0.00	12
3943	0.00	0.00	0.00	18
3944	0.00	0.00	0.00	17
3945	0.00	0.00	0.00	17
3946	0.00	0.00	0.00	12
3947	0.00	0.00	0.00	15
3948	0.00	0.00	0.00	14
3949	0.00	0.00	0.00	17
3950	0.00	0.00	0.00	14
3951	0.00	0.00	0.00	15
3952	0.00	0.00	0.00	17
3953	0.00	0.00	0.00	11
3954	0.00	0.00	0.00	14
3955	0.00	0.00	0.00	15
3956	0.00	0.00	0.00	17
3957	0.00	0.00	0.00	9
3958	0.00	0.00	0.00	20
3959	1.00	0.33	0.50	9
3960	0.00	0.00	0.00	13
3961	0.00	0.00	0.00	18
3962	0.00	0.00	0.00	14
3963	0.00	0.00	0.00	15
3964	0.00	0.00	0.00	13
3965	0.00	0.00	0.00	16
3966	0.00	0.00	0.00	15
3967	0.00	0.00	0.00	15
3968	0.00	0.00	0.00	17
3969	0.00	0.00	0.00	20
3970	0.00	0.00	0.00	16
3971	0.00	0.00	0.00	19
3972	1.00	0.12	0.22	16
3973	0.00	0.00	0.00	15
3974	0.00	0.00	0.00	8
3975	0.00	0.00	0.00	16
3976	0.00	0.00	0.00	15
3977	0.00	0.00	0.00	14
3978	0.00	0.00	0.00	16
3979	0.00	0.00	0.00	13
3980	0.00	0.00	0.00	28
3981	0.00	0.00	0.00	16
3982	0.00	0.00	0.00	12
3983	0.00	0.00	0.00	13
3984	0.00	0.00	0.00	12
3985	0.00	0.00	0.00	15
3986	0.00	0.00	0.00	10
3987	0.00	0.00	0.00	20

3988	0.00	0.00	0.00	17
3989	0.00	0.00	0.00	14
3990	0.00	0.00	0.00	11
3991	0.00	0.00	0.00	14
3992	0.00	0.00	0.00	13
3993	1.00	0.23	0.38	13
3994	0.00	0.00	0.00	18
3995	0.00	0.00	0.00	13
3996	0.00	0.00	0.00	13
3997	0.00	0.00	0.00	19
3998	0.00	0.00	0.00	10
3999	1.00	0.13	0.24	15
4000	0.00	0.00	0.00	20
4001	0.00	0.00	0.00	16
4002	0.00	0.00	0.00	11
4003	0.00	0.00	0.00	14
4004	0.00	0.00	0.00	15
4005	0.00	0.00	0.00	21
4006	0.00	0.00	0.00	12
4007	0.00	0.00	0.00	15
4008	0.00	0.00	0.00	9
4009	0.50	0.06	0.11	16
4010	0.00	0.00	0.00	12
4011	0.00	0.00	0.00	16
4012	0.00	0.00	0.00	19
4013	0.00	0.00	0.00	13
4014	0.00	0.00	0.00	13
4015	0.00	0.00	0.00	13
4016	0.00	0.00	0.00	16
4017	0.00	0.00	0.00	17
4018	0.00	0.00	0.00	10
4019	0.00	0.00	0.00	12
4020	0.00	0.00	0.00	13
4021	0.00	0.00	0.00	17
4022	0.00	0.00	0.00	16
4023	0.00	0.00	0.00	14
4024	0.00	0.00	0.00	11
4025	0.00	0.00	0.00	8
4026	0.00	0.00	0.00	8
4027	0.00	0.00	0.00	18
4028	0.00	0.00	0.00	13
4029	0.00	0.00	0.00	11
4030	0.00	0.00	0.00	19
4031	0.00	0.00	0.00	9
4032	0.00	0.00	0.00	12
4033	0.00	0.00	0.00	14
4034	0.00	0.00	0.00	17
4035	0.00	0.00	0.00	10
4036	0.00	0.00	0.00	12
4037	0.00	0.00	0.00	13
4038	0.00	0.00	0.00	13
4039	0.00	0.00	0.00	13
4040	0.00	0.00	0.00	12
4041	0.00	0.00	0.00	17
4042	0.00	0.00	0.00	10
4043	0.00	0.00	0.00	15
4044	0.00	0.00	0.00	13
4045	0.00	0.00	0.00	20
4046	0.00	0.00	0.00	16
4047	0.00	0.00	0.00	12
4048	0.00	0.00	0.00	16
4049	0.00	0.00	0.00	14
4050	0.00	0.00	0.00	15
4051	0.00	0.00	0.00	20
4052	0.00	0.00	0.00	10
4053	0.00	0.00	0.00	14
4054	0.00	0.00	0.00	14
4055	0.00	0.00	0.00	5
4056	0.00	0.00	0.00	15
4057	1.00	0.07	0.12	15
4058	0.00	0.00	0.00	17
4059	0.00	0.00	0.00	13
4060	0.00	0.00	0.00	14
4061	0.00	0.00	0.00	10
4062	0.00	0.00	0.00	15
4063	0.00	0.00	0.00	15
4064	0.00	0.00	0.00	17
4065	0.00	0.00	0.00	17
4066	0.00	0.00	0.00	14
4067	0.00	0.00	0.00	15
4068	0.00	0.00	0.00	21
4069	0.00	0.00	0.00	9
4070	0.00	0.00	0.00	9

4071	0.00	0.00	0.00	21
4072	0.00	0.00	0.00	18
4073	0.00	0.00	0.00	9
4074	0.00	0.00	0.00	12
4075	0.00	0.00	0.00	20
4076	0.00	0.00	0.00	15
4077	0.00	0.00	0.00	15
4078	0.00	0.00	0.00	9
4079	0.00	0.00	0.00	15
4080	0.00	0.00	0.00	19
4081	0.00	0.00	0.00	10
4082	0.00	0.00	0.00	11
4083	0.00	0.00	0.00	12
4084	0.00	0.00	0.00	14
4085	0.00	0.00	0.00	9
4086	0.00	0.00	0.00	9
4087	0.00	0.00	0.00	9
4088	0.00	0.00	0.00	18
4089	0.00	0.00	0.00	14
4090	0.00	0.00	0.00	18
4091	0.00	0.00	0.00	14
4092	0.00	0.00	0.00	13
4093	0.00	0.00	0.00	16
4094	0.00	0.00	0.00	14
4095	0.00	0.00	0.00	19
4096	0.00	0.00	0.00	15
4097	0.00	0.00	0.00	14
4098	0.00	0.00	0.00	16
4099	0.00	0.00	0.00	21
4100	0.00	0.00	0.00	18
4101	0.00	0.00	0.00	15
4102	0.00	0.00	0.00	15
4103	0.00	0.00	0.00	17
4104	0.00	0.00	0.00	13
4105	0.00	0.00	0.00	15
4106	0.00	0.00	0.00	14
4107	0.00	0.00	0.00	13
4108	0.00	0.00	0.00	15
4109	0.00	0.00	0.00	15
4110	0.00	0.00	0.00	13
4111	0.00	0.00	0.00	16
4112	0.00	0.00	0.00	13
4113	0.00	0.00	0.00	12
4114	0.00	0.00	0.00	13
4115	0.00	0.00	0.00	11
4116	0.00	0.00	0.00	15
4117	0.00	0.00	0.00	12
4118	0.00	0.00	0.00	12
4119	0.00	0.00	0.00	18
4120	1.00	0.09	0.17	11
4121	0.00	0.00	0.00	9
4122	0.00	0.00	0.00	12
4123	0.00	0.00	0.00	11
4124	0.00	0.00	0.00	9
4125	0.00	0.00	0.00	9
4126	0.00	0.00	0.00	15
4127	0.00	0.00	0.00	16
4128	0.00	0.00	0.00	13
4129	0.00	0.00	0.00	11
4130	0.00	0.00	0.00	7
4131	0.00	0.00	0.00	12
4132	0.00	0.00	0.00	15
4133	1.00	0.08	0.15	12
4134	0.00	0.00	0.00	16
4135	0.00	0.00	0.00	16
4136	0.00	0.00	0.00	11
4137	0.00	0.00	0.00	12
4138	0.00	0.00	0.00	12
4139	0.00	0.00	0.00	21
4140	0.00	0.00	0.00	13
4141	0.00	0.00	0.00	7
4142	0.00	0.00	0.00	12
4143	0.00	0.00	0.00	19
4144	0.00	0.00	0.00	10
4145	0.00	0.00	0.00	13
4146	0.00	0.00	0.00	18
4147	0.00	0.00	0.00	14
4148	0.00	0.00	0.00	11
4149	0.00	0.00	0.00	7
4150	0.00	0.00	0.00	10
4151	0.00	0.00	0.00	18
4152	0.00	0.00	0.00	14
4153	0.00	0.00	0.00	16

4154	0.00	0.00	0.00	12
4155	0.00	0.00	0.00	10
4156	0.00	0.00	0.00	15
4157	0.00	0.00	0.00	16
4158	0.00	0.00	0.00	19
4159	0.00	0.00	0.00	10
4160	0.00	0.00	0.00	17
4161	0.00	0.00	0.00	18
4162	0.00	0.00	0.00	12
4163	0.00	0.00	0.00	11
4164	0.00	0.00	0.00	8
4165	0.00	0.00	0.00	17
4166	0.00	0.00	0.00	17
4167	0.00	0.00	0.00	8
4168	0.00	0.00	0.00	12
4169	0.00	0.00	0.00	19
4170	0.00	0.00	0.00	15
4171	0.00	0.00	0.00	10
4172	0.00	0.00	0.00	17
4173	0.00	0.00	0.00	12
4174	0.00	0.00	0.00	14
4175	0.00	0.00	0.00	18
4176	0.00	0.00	0.00	8
4177	0.00	0.00	0.00	20
4178	0.00	0.00	0.00	15
4179	0.00	0.00	0.00	16
4180	0.00	0.00	0.00	12
4181	0.00	0.00	0.00	18
4182	0.00	0.00	0.00	8
4183	0.00	0.00	0.00	18
4184	0.00	0.00	0.00	16
4185	0.00	0.00	0.00	12
4186	0.00	0.00	0.00	16
4187	0.00	0.00	0.00	14
4188	0.00	0.00	0.00	17
4189	0.00	0.00	0.00	13
4190	0.00	0.00	0.00	11
4191	0.00	0.00	0.00	14
4192	0.00	0.00	0.00	11
4193	0.00	0.00	0.00	11
4194	0.00	0.00	0.00	17
4195	0.00	0.00	0.00	6
4196	0.00	0.00	0.00	17
4197	0.00	0.00	0.00	13
4198	0.00	0.00	0.00	12
4199	0.00	0.00	0.00	9
4200	0.00	0.00	0.00	12
4201	0.00	0.00	0.00	13
4202	0.00	0.00	0.00	13
4203	0.00	0.00	0.00	15
4204	0.00	0.00	0.00	15
4205	0.00	0.00	0.00	11
4206	0.00	0.00	0.00	14
4207	0.00	0.00	0.00	9
4208	0.00	0.00	0.00	15
4209	0.00	0.00	0.00	14
4210	0.00	0.00	0.00	11
4211	0.00	0.00	0.00	12
4212	0.00	0.00	0.00	12
4213	0.00	0.00	0.00	14
4214	0.00	0.00	0.00	9
4215	0.00	0.00	0.00	7
4216	0.00	0.00	0.00	12
4217	0.00	0.00	0.00	11
4218	0.00	0.00	0.00	13
4219	1.00	0.09	0.17	11
4220	1.00	0.07	0.13	14
4221	0.00	0.00	0.00	11
4222	1.00	0.08	0.14	13
4223	0.00	0.00	0.00	4
4224	0.00	0.00	0.00	12
4225	0.00	0.00	0.00	13
4226	0.00	0.00	0.00	7
4227	0.00	0.00	0.00	14
4228	0.00	0.00	0.00	9
4229	0.00	0.00	0.00	14
4230	0.00	0.00	0.00	11
4231	0.00	0.00	0.00	13
4232	0.00	0.00	0.00	16
4233	0.00	0.00	0.00	20
4234	0.00	0.00	0.00	12
4235	0.00	0.00	0.00	12
4236	0.00	0.00	0.00	13

4237	0.00	0.00	0.00	11
4238	0.00	0.00	0.00	15
4239	0.00	0.00	0.00	10
4240	0.00	0.00	0.00	11
4241	0.00	0.00	0.00	17
4242	0.00	0.00	0.00	16
4243	0.00	0.00	0.00	17
4244	0.00	0.00	0.00	12
4245	0.00	0.00	0.00	16
4246	0.00	0.00	0.00	10
4247	0.00	0.00	0.00	19
4248	0.00	0.00	0.00	9
4249	0.00	0.00	0.00	15
4250	0.00	0.00	0.00	18
4251	0.00	0.00	0.00	11
4252	0.00	0.00	0.00	9
4253	0.00	0.00	0.00	16
4254	0.00	0.00	0.00	13
4255	0.00	0.00	0.00	7
4256	0.00	0.00	0.00	11
4257	0.00	0.00	0.00	17
4258	0.00	0.00	0.00	12
4259	0.00	0.00	0.00	12
4260	0.00	0.00	0.00	17
4261	0.00	0.00	0.00	12
4262	0.00	0.00	0.00	10
4263	0.00	0.00	0.00	21
4264	0.00	0.00	0.00	16
4265	0.00	0.00	0.00	13
4266	0.00	0.00	0.00	13
4267	0.00	0.00	0.00	12
4268	0.00	0.00	0.00	14
4269	0.00	0.00	0.00	16
4270	0.00	0.00	0.00	12
4271	0.00	0.00	0.00	10
4272	0.00	0.00	0.00	15
4273	0.00	0.00	0.00	9
4274	0.00	0.00	0.00	17
4275	0.00	0.00	0.00	16
4276	0.00	0.00	0.00	8
4277	0.00	0.00	0.00	14
4278	0.00	0.00	0.00	18
4279	0.00	0.00	0.00	17
4280	0.00	0.00	0.00	12
4281	0.00	0.00	0.00	4
4282	0.00	0.00	0.00	17
4283	0.00	0.00	0.00	14
4284	0.00	0.00	0.00	15
4285	0.00	0.00	0.00	22
4286	0.00	0.00	0.00	18
4287	0.00	0.00	0.00	9
4288	0.00	0.00	0.00	14
4289	0.00	0.00	0.00	9
4290	0.00	0.00	0.00	12
4291	0.00	0.00	0.00	11
4292	1.00	0.06	0.11	17
4293	0.00	0.00	0.00	8
4294	0.00	0.00	0.00	8
4295	0.00	0.00	0.00	9
4296	0.00	0.00	0.00	9
4297	0.00	0.00	0.00	19
4298	0.00	0.00	0.00	11
4299	0.00	0.00	0.00	6
4300	0.00	0.00	0.00	13
4301	0.00	0.00	0.00	14
4302	0.00	0.00	0.00	14
4303	0.00	0.00	0.00	15
4304	0.00	0.00	0.00	4
4305	0.00	0.00	0.00	13
4306	0.00	0.00	0.00	12
4307	0.00	0.00	0.00	7
4308	0.00	0.00	0.00	19
4309	0.00	0.00	0.00	12
4310	0.00	0.00	0.00	15
4311	0.00	0.00	0.00	13
4312	0.00	0.00	0.00	20
4313	0.00	0.00	0.00	10
4314	0.00	0.00	0.00	10
4315	0.00	0.00	0.00	12
4316	0.00	0.00	0.00	11
4317	0.00	0.00	0.00	11
4318	0.00	0.00	0.00	13
4319	0.00	0.00	0.00	11

4320	0.00	0.00	0.00	10
4321	0.00	0.00	0.00	13
4322	0.00	0.00	0.00	10
4323	0.00	0.00	0.00	14
4324	0.00	0.00	0.00	13
4325	0.00	0.00	0.00	8
4326	0.00	0.00	0.00	13
4327	0.00	0.00	0.00	15
4328	0.00	0.00	0.00	15
4329	0.00	0.00	0.00	15
4330	0.00	0.00	0.00	13
4331	0.00	0.00	0.00	9
4332	0.00	0.00	0.00	12
4333	0.00	0.00	0.00	13
4334	0.00	0.00	0.00	12
4335	0.00	0.00	0.00	16
4336	0.00	0.00	0.00	14
4337	0.00	0.00	0.00	11
4338	0.00	0.00	0.00	11
4339	0.00	0.00	0.00	18
4340	0.00	0.00	0.00	12
4341	0.00	0.00	0.00	13
4342	0.00	0.00	0.00	6
4343	0.00	0.00	0.00	16
4344	0.00	0.00	0.00	14
4345	0.00	0.00	0.00	15
4346	0.00	0.00	0.00	10
4347	0.00	0.00	0.00	14
4348	0.00	0.00	0.00	12
4349	0.00	0.00	0.00	14
4350	0.00	0.00	0.00	17
4351	0.00	0.00	0.00	16
4352	0.00	0.00	0.00	11
4353	0.00	0.00	0.00	9
4354	0.00	0.00	0.00	17
4355	0.00	0.00	0.00	23
4356	0.00	0.00	0.00	6
4357	0.00	0.00	0.00	10
4358	0.00	0.00	0.00	9
4359	0.00	0.00	0.00	10
4360	0.00	0.00	0.00	17
4361	0.00	0.00	0.00	5
4362	0.00	0.00	0.00	13
4363	0.00	0.00	0.00	11
4364	0.00	0.00	0.00	17
4365	0.00	0.00	0.00	14
4366	0.00	0.00	0.00	13
4367	0.00	0.00	0.00	10
4368	0.75	0.17	0.27	18
4369	0.00	0.00	0.00	7
4370	0.00	0.00	0.00	12
4371	0.00	0.00	0.00	14
4372	0.00	0.00	0.00	6
4373	0.00	0.00	0.00	8
4374	0.00	0.00	0.00	16
4375	0.00	0.00	0.00	11
4376	0.00	0.00	0.00	18
4377	0.00	0.00	0.00	9
4378	0.00	0.00	0.00	14
4379	0.00	0.00	0.00	8
4380	0.00	0.00	0.00	9
4381	0.00	0.00	0.00	10
4382	0.00	0.00	0.00	16
4383	0.00	0.00	0.00	13
4384	0.00	0.00	0.00	9
4385	0.00	0.00	0.00	12
4386	0.00	0.00	0.00	14
4387	0.00	0.00	0.00	11
4388	0.00	0.00	0.00	8
4389	0.00	0.00	0.00	12
4390	0.00	0.00	0.00	8
4391	0.00	0.00	0.00	16
4392	0.00	0.00	0.00	7
4393	0.00	0.00	0.00	8
4394	0.00	0.00	0.00	11
4395	0.00	0.00	0.00	9
4396	0.00	0.00	0.00	11
4397	0.00	0.00	0.00	13
4398	0.00	0.00	0.00	17
4399	0.00	0.00	0.00	10
4400	0.00	0.00	0.00	17
4401	0.00	0.00	0.00	8
4402	0.33	0.08	0.13	12

4403	0.00	0.00	0.00	14
4404	0.00	0.00	0.00	14
4405	0.00	0.00	0.00	10
4406	0.00	0.00	0.00	14
4407	0.00	0.00	0.00	13
4408	0.00	0.00	0.00	13
4409	0.00	0.00	0.00	11
4410	0.00	0.00	0.00	16
4411	0.00	0.00	0.00	12
4412	0.00	0.00	0.00	10
4413	0.00	0.00	0.00	16
4414	0.00	0.00	0.00	14
4415	0.00	0.00	0.00	11
4416	0.00	0.00	0.00	14
4417	0.00	0.00	0.00	13
4418	0.00	0.00	0.00	8
4419	0.00	0.00	0.00	12
4420	0.00	0.00	0.00	13
4421	0.00	0.00	0.00	15
4422	0.00	0.00	0.00	14
4423	0.00	0.00	0.00	15
4424	0.00	0.00	0.00	9
4425	0.00	0.00	0.00	10
4426	0.00	0.00	0.00	17
4427	0.00	0.00	0.00	12
4428	0.00	0.00	0.00	12
4429	0.00	0.00	0.00	13
4430	0.00	0.00	0.00	10
4431	0.00	0.00	0.00	10
4432	0.00	0.00	0.00	10
4433	0.00	0.00	0.00	15
4434	0.00	0.00	0.00	13
4435	0.00	0.00	0.00	21
4436	0.00	0.00	0.00	17
4437	0.00	0.00	0.00	9
4438	0.00	0.00	0.00	11
4439	0.00	0.00	0.00	17
4440	0.00	0.00	0.00	14
4441	0.00	0.00	0.00	15
4442	0.00	0.00	0.00	8
4443	0.00	0.00	0.00	13
4444	0.00	0.00	0.00	10
4445	0.00	0.00	0.00	13
4446	0.00	0.00	0.00	10
4447	0.00	0.00	0.00	10
4448	0.00	0.00	0.00	7
4449	0.00	0.00	0.00	12
4450	0.00	0.00	0.00	8
4451	0.00	0.00	0.00	13
4452	0.00	0.00	0.00	15
4453	0.00	0.00	0.00	8
4454	0.00	0.00	0.00	4
4455	0.00	0.00	0.00	15
4456	0.00	0.00	0.00	9
4457	0.00	0.00	0.00	10
4458	0.00	0.00	0.00	13
4459	0.00	0.00	0.00	14
4460	0.00	0.00	0.00	10
4461	0.00	0.00	0.00	12
4462	0.00	0.00	0.00	10
4463	0.00	0.00	0.00	12
4464	0.00	0.00	0.00	9
4465	0.00	0.00	0.00	9
4466	0.00	0.00	0.00	12
4467	0.00	0.00	0.00	10
4468	0.00	0.00	0.00	11
4469	0.00	0.00	0.00	13
4470	0.00	0.00	0.00	18
4471	0.00	0.00	0.00	11
4472	0.00	0.00	0.00	16
4473	0.00	0.00	0.00	12
4474	0.00	0.00	0.00	10
4475	0.00	0.00	0.00	11
4476	0.00	0.00	0.00	13
4477	0.00	0.00	0.00	12
4478	0.00	0.00	0.00	11
4479	0.00	0.00	0.00	14
4480	0.00	0.00	0.00	10
4481	0.00	0.00	0.00	11
4482	0.00	0.00	0.00	13
4483	0.00	0.00	0.00	13
4484	0.00	0.00	0.00	15
4485	0.00	0.00	0.00	13

4486	0.00	0.00	0.00	14
4487	0.00	0.00	0.00	15
4488	0.00	0.00	0.00	14
4489	0.00	0.00	0.00	13
4490	0.00	0.00	0.00	18
4491	0.00	0.00	0.00	10
4492	0.00	0.00	0.00	12
4493	0.00	0.00	0.00	16
4494	0.00	0.00	0.00	8
4495	0.00	0.00	0.00	9
4496	0.00	0.00	0.00	8
4497	0.00	0.00	0.00	13
4498	0.00	0.00	0.00	18
4499	0.00	0.00	0.00	11
4500	0.00	0.00	0.00	8
4501	0.00	0.00	0.00	17
4502	0.00	0.00	0.00	9
4503	0.00	0.00	0.00	12
4504	0.00	0.00	0.00	7
4505	0.00	0.00	0.00	13
4506	0.00	0.00	0.00	13
4507	0.00	0.00	0.00	12
4508	0.00	0.00	0.00	13
4509	0.00	0.00	0.00	19
4510	0.00	0.00	0.00	12
4511	0.00	0.00	0.00	12
4512	0.00	0.00	0.00	13
4513	0.00	0.00	0.00	11
4514	0.00	0.00	0.00	8
4515	0.00	0.00	0.00	9
4516	0.00	0.00	0.00	10
4517	0.00	0.00	0.00	13
4518	0.00	0.00	0.00	9
4519	0.00	0.00	0.00	12
4520	0.00	0.00	0.00	12
4521	0.00	0.00	0.00	14
4522	0.00	0.00	0.00	6
4523	0.00	0.00	0.00	14
4524	0.00	0.00	0.00	13
4525	0.00	0.00	0.00	11
4526	0.00	0.00	0.00	14
4527	0.00	0.00	0.00	12
4528	0.00	0.00	0.00	12
4529	0.00	0.00	0.00	10
4530	0.00	0.00	0.00	15
4531	0.00	0.00	0.00	16
4532	0.00	0.00	0.00	12
4533	0.00	0.00	0.00	14
4534	0.00	0.00	0.00	13
4535	0.00	0.00	0.00	12
4536	0.00	0.00	0.00	11
4537	0.00	0.00	0.00	18
4538	0.00	0.00	0.00	7
4539	0.00	0.00	0.00	11
4540	0.00	0.00	0.00	11
4541	0.00	0.00	0.00	12
4542	0.00	0.00	0.00	13
4543	0.00	0.00	0.00	9
4544	0.00	0.00	0.00	12
4545	0.00	0.00	0.00	12
4546	0.00	0.00	0.00	12
4547	0.00	0.00	0.00	8
4548	0.00	0.00	0.00	12
4549	0.00	0.00	0.00	9
4550	0.00	0.00	0.00	8
4551	0.00	0.00	0.00	13
4552	0.00	0.00	0.00	10
4553	0.00	0.00	0.00	8
4554	0.00	0.00	0.00	10
4555	0.00	0.00	0.00	8
4556	0.00	0.00	0.00	5
4557	0.00	0.00	0.00	10
4558	0.00	0.00	0.00	9
4559	0.00	0.00	0.00	14
4560	0.00	0.00	0.00	16
4561	0.00	0.00	0.00	15
4562	0.00	0.00	0.00	11
4563	0.00	0.00	0.00	9
4564	0.00	0.00	0.00	13
4565	0.00	0.00	0.00	12
4566	0.00	0.00	0.00	8
4567	0.00	0.00	0.00	5
4568	0.00	0.00	0.00	7

4569	0.00	0.00	0.00	7
4570	0.00	0.00	0.00	10
4571	0.00	0.00	0.00	12
4572	0.00	0.00	0.00	14
4573	0.00	0.00	0.00	12
4574	0.00	0.00	0.00	8
4575	0.00	0.00	0.00	11
4576	0.00	0.00	0.00	10
4577	0.00	0.00	0.00	9
4578	0.00	0.00	0.00	14
4579	0.00	0.00	0.00	13
4580	0.00	0.00	0.00	14
4581	0.00	0.00	0.00	9
4582	0.00	0.00	0.00	15
4583	0.00	0.00	0.00	13
4584	0.00	0.00	0.00	7
4585	0.00	0.00	0.00	9
4586	0.00	0.00	0.00	15
4587	0.00	0.00	0.00	13
4588	0.00	0.00	0.00	11
4589	0.00	0.00	0.00	6
4590	0.00	0.00	0.00	6
4591	0.00	0.00	0.00	11
4592	0.00	0.00	0.00	12
4593	0.00	0.00	0.00	12
4594	0.00	0.00	0.00	10
4595	0.00	0.00	0.00	14
4596	0.00	0.00	0.00	11
4597	0.00	0.00	0.00	11
4598	0.00	0.00	0.00	9
4599	0.00	0.00	0.00	7
4600	0.00	0.00	0.00	11
4601	0.00	0.00	0.00	12
4602	0.00	0.00	0.00	9
4603	0.00	0.00	0.00	13
4604	0.00	0.00	0.00	15
4605	0.00	0.00	0.00	11
4606	0.00	0.00	0.00	9
4607	0.00	0.00	0.00	10
4608	0.00	0.00	0.00	6
4609	0.00	0.00	0.00	6
4610	0.00	0.00	0.00	12
4611	0.00	0.00	0.00	9
4612	0.00	0.00	0.00	13
4613	0.00	0.00	0.00	14
4614	0.00	0.00	0.00	8
4615	0.00	0.00	0.00	12
4616	0.00	0.00	0.00	13
4617	0.00	0.00	0.00	7
4618	0.00	0.00	0.00	11
4619	0.00	0.00	0.00	14
4620	0.00	0.00	0.00	11
4621	0.00	0.00	0.00	9
4622	0.00	0.00	0.00	6
4623	0.00	0.00	0.00	12
4624	0.00	0.00	0.00	11
4625	0.00	0.00	0.00	10
4626	0.00	0.00	0.00	9
4627	0.00	0.00	0.00	8
4628	0.00	0.00	0.00	11
4629	0.00	0.00	0.00	11
4630	0.00	0.00	0.00	13
4631	0.00	0.00	0.00	15
4632	0.00	0.00	0.00	11
4633	0.00	0.00	0.00	7
4634	0.00	0.00	0.00	11
4635	0.00	0.00	0.00	8
4636	0.00	0.00	0.00	7
4637	0.00	0.00	0.00	8
4638	0.00	0.00	0.00	9
4639	0.00	0.00	0.00	13
4640	0.00	0.00	0.00	12
4641	0.00	0.00	0.00	11
4642	0.00	0.00	0.00	8
4643	0.00	0.00	0.00	12
4644	0.00	0.00	0.00	9
4645	0.00	0.00	0.00	12
4646	0.00	0.00	0.00	10
4647	0.00	0.00	0.00	17
4648	0.00	0.00	0.00	10
4649	0.00	0.00	0.00	12
4650	0.00	0.00	0.00	13
4651	0.00	0.00	0.00	12

4652	0.00	0.00	0.00	11
4653	0.00	0.00	0.00	10
4654	0.00	0.00	0.00	11
4655	0.00	0.00	0.00	14
4656	0.00	0.00	0.00	10
4657	0.00	0.00	0.00	9
4658	0.00	0.00	0.00	9
4659	0.00	0.00	0.00	9
4660	0.00	0.00	0.00	13
4661	0.00	0.00	0.00	8
4662	0.00	0.00	0.00	12
4663	0.00	0.00	0.00	12
4664	0.00	0.00	0.00	14
4665	0.00	0.00	0.00	11
4666	0.00	0.00	0.00	9
4667	0.00	0.00	0.00	7
4668	0.00	0.00	0.00	8
4669	0.00	0.00	0.00	6
4670	0.00	0.00	0.00	12
4671	0.00	0.00	0.00	6
4672	0.00	0.00	0.00	14
4673	0.00	0.00	0.00	14
4674	0.00	0.00	0.00	13
4675	0.00	0.00	0.00	12
4676	0.00	0.00	0.00	13
4677	0.00	0.00	0.00	12
4678	0.00	0.00	0.00	11
4679	0.00	0.00	0.00	14
4680	0.00	0.00	0.00	7
4681	0.00	0.00	0.00	9
4682	0.00	0.00	0.00	15
4683	0.00	0.00	0.00	10
4684	0.00	0.00	0.00	7
4685	0.00	0.00	0.00	12
4686	0.00	0.00	0.00	9
4687	0.00	0.00	0.00	11
4688	0.00	0.00	0.00	10
4689	0.00	0.00	0.00	17
4690	0.00	0.00	0.00	11
4691	0.00	0.00	0.00	16
4692	0.00	0.00	0.00	12
4693	0.00	0.00	0.00	9
4694	0.00	0.00	0.00	16
4695	0.00	0.00	0.00	10
4696	0.00	0.00	0.00	13
4697	0.00	0.00	0.00	10
4698	0.00	0.00	0.00	13
4699	0.00	0.00	0.00	12
4700	0.00	0.00	0.00	16
4701	0.00	0.00	0.00	5
4702	0.00	0.00	0.00	10
4703	0.00	0.00	0.00	8
4704	0.00	0.00	0.00	17
4705	0.00	0.00	0.00	12
4706	0.00	0.00	0.00	5
4707	0.00	0.00	0.00	11
4708	0.00	0.00	0.00	13
4709	0.00	0.00	0.00	11
4710	0.00	0.00	0.00	10
4711	0.00	0.00	0.00	12
4712	0.00	0.00	0.00	9
4713	0.00	0.00	0.00	14
4714	0.00	0.00	0.00	14
4715	0.00	0.00	0.00	11
4716	0.00	0.00	0.00	10
4717	0.00	0.00	0.00	16
4718	0.00	0.00	0.00	15
4719	0.00	0.00	0.00	14
4720	0.00	0.00	0.00	10
4721	0.00	0.00	0.00	18
4722	0.00	0.00	0.00	9
4723	0.00	0.00	0.00	15
4724	0.00	0.00	0.00	10
4725	0.00	0.00	0.00	6
4726	0.00	0.00	0.00	8
4727	0.00	0.00	0.00	9
4728	0.00	0.00	0.00	12
4729	0.00	0.00	0.00	10
4730	0.00	0.00	0.00	16
4731	0.00	0.00	0.00	9
4732	0.00	0.00	0.00	10
4733	0.00	0.00	0.00	13
4734	0.00	0.00	0.00	14

4735	0.00	0.00	0.00	20
4736	0.00	0.00	0.00	9
4737	0.00	0.00	0.00	8
4738	0.00	0.00	0.00	16
4739	0.00	0.00	0.00	6
4740	0.00	0.00	0.00	10
4741	0.00	0.00	0.00	10
4742	0.00	0.00	0.00	10
4743	0.00	0.00	0.00	8
4744	0.00	0.00	0.00	9
4745	0.00	0.00	0.00	12
4746	0.00	0.00	0.00	11
4747	0.00	0.00	0.00	18
4748	0.00	0.00	0.00	7
4749	0.00	0.00	0.00	10
4750	0.00	0.00	0.00	12
4751	0.00	0.00	0.00	13
4752	0.00	0.00	0.00	9
4753	0.00	0.00	0.00	8
4754	0.00	0.00	0.00	10
4755	0.00	0.00	0.00	14
4756	0.00	0.00	0.00	17
4757	0.00	0.00	0.00	15
4758	0.00	0.00	0.00	11
4759	0.00	0.00	0.00	10
4760	0.00	0.00	0.00	10
4761	0.00	0.00	0.00	14
4762	0.00	0.00	0.00	13
4763	0.00	0.00	0.00	13
4764	0.00	0.00	0.00	12
4765	0.00	0.00	0.00	8
4766	0.00	0.00	0.00	7
4767	0.00	0.00	0.00	14
4768	0.00	0.00	0.00	10
4769	0.00	0.00	0.00	11
4770	0.00	0.00	0.00	12
4771	0.00	0.00	0.00	11
4772	0.00	0.00	0.00	11
4773	0.00	0.00	0.00	17
4774	0.00	0.00	0.00	5
4775	0.00	0.00	0.00	5
4776	0.00	0.00	0.00	12
4777	0.00	0.00	0.00	12
4778	0.00	0.00	0.00	10
4779	0.00	0.00	0.00	16
4780	0.00	0.00	0.00	10
4781	0.00	0.00	0.00	5
4782	0.00	0.00	0.00	11
4783	0.00	0.00	0.00	7
4784	0.00	0.00	0.00	13
4785	0.00	0.00	0.00	8
4786	0.00	0.00	0.00	15
4787	0.00	0.00	0.00	8
4788	0.00	0.00	0.00	7
4789	0.00	0.00	0.00	10
4790	0.00	0.00	0.00	12
4791	0.00	0.00	0.00	11
4792	0.00	0.00	0.00	10
4793	0.00	0.00	0.00	13
4794	0.00	0.00	0.00	18
4795	0.00	0.00	0.00	6
4796	0.00	0.00	0.00	11
4797	0.00	0.00	0.00	9
4798	0.00	0.00	0.00	11
4799	0.00	0.00	0.00	10
4800	0.00	0.00	0.00	14
4801	0.00	0.00	0.00	9
4802	0.00	0.00	0.00	11
4803	0.00	0.00	0.00	12
4804	0.00	0.00	0.00	19
4805	0.00	0.00	0.00	10
4806	0.00	0.00	0.00	12
4807	0.00	0.00	0.00	12
4808	0.00	0.00	0.00	14
4809	0.00	0.00	0.00	12
4810	0.00	0.00	0.00	7
4811	0.00	0.00	0.00	16
4812	0.00	0.00	0.00	10
4813	0.00	0.00	0.00	14
4814	0.00	0.00	0.00	10
4815	0.00	0.00	0.00	10
4816	0.00	0.00	0.00	12
4817	0.00	0.00	0.00	14

4818	0.00	0.00	0.00	9
4819	0.00	0.00	0.00	13
4820	0.00	0.00	0.00	15
4821	0.00	0.00	0.00	5
4822	0.00	0.00	0.00	12
4823	0.00	0.00	0.00	11
4824	0.00	0.00	0.00	18
4825	0.00	0.00	0.00	8
4826	0.00	0.00	0.00	7
4827	0.00	0.00	0.00	13
4828	0.00	0.00	0.00	16
4829	0.00	0.00	0.00	5
4830	0.00	0.00	0.00	9
4831	0.00	0.00	0.00	12
4832	0.00	0.00	0.00	12
4833	0.00	0.00	0.00	12
4834	0.00	0.00	0.00	16
4835	0.00	0.00	0.00	9
4836	0.00	0.00	0.00	8
4837	0.00	0.00	0.00	10
4838	0.00	0.00	0.00	12
4839	0.00	0.00	0.00	10
4840	0.00	0.00	0.00	8
4841	0.00	0.00	0.00	13
4842	0.00	0.00	0.00	8
4843	0.00	0.00	0.00	10
4844	0.00	0.00	0.00	6
4845	0.00	0.00	0.00	13
4846	0.00	0.00	0.00	15
4847	0.00	0.00	0.00	16
4848	0.00	0.00	0.00	12
4849	0.00	0.00	0.00	13
4850	0.00	0.00	0.00	16
4851	0.00	0.00	0.00	13
4852	0.00	0.00	0.00	11
4853	0.00	0.00	0.00	10
4854	0.00	0.00	0.00	10
4855	0.00	0.00	0.00	7
4856	0.00	0.00	0.00	9
4857	0.00	0.00	0.00	12
4858	0.00	0.00	0.00	9
4859	0.00	0.00	0.00	11
4860	0.00	0.00	0.00	11
4861	0.00	0.00	0.00	15
4862	0.00	0.00	0.00	10
4863	0.00	0.00	0.00	9
4864	0.00	0.00	0.00	6
4865	0.00	0.00	0.00	14
4866	0.00	0.00	0.00	7
4867	0.00	0.00	0.00	8
4868	0.00	0.00	0.00	14
4869	0.00	0.00	0.00	10
4870	0.00	0.00	0.00	11
4871	0.00	0.00	0.00	11
4872	0.00	0.00	0.00	13
4873	0.00	0.00	0.00	9
4874	0.00	0.00	0.00	8
4875	0.00	0.00	0.00	10
4876	0.00	0.00	0.00	8
4877	0.00	0.00	0.00	8
4878	0.00	0.00	0.00	14
4879	0.00	0.00	0.00	11
4880	0.00	0.00	0.00	5
4881	0.00	0.00	0.00	10
4882	0.00	0.00	0.00	9
4883	0.00	0.00	0.00	10
4884	0.00	0.00	0.00	15
4885	0.00	0.00	0.00	11
4886	0.00	0.00	0.00	18
4887	0.00	0.00	0.00	12
4888	0.00	0.00	0.00	13
4889	0.00	0.00	0.00	8
4890	0.00	0.00	0.00	4
4891	0.00	0.00	0.00	10
4892	0.00	0.00	0.00	14
4893	0.00	0.00	0.00	12
4894	0.00	0.00	0.00	9
4895	1.00	0.12	0.22	8
4896	0.00	0.00	0.00	11
4897	0.00	0.00	0.00	14
4898	0.00	0.00	0.00	12
4899	0.00	0.00	0.00	11
4900	0.00	0.00	0.00	12

4901	0.00	0.00	0.00	13
4902	0.00	0.00	0.00	12
4903	0.00	0.00	0.00	11
4904	0.00	0.00	0.00	10
4905	0.00	0.00	0.00	11
4906	0.00	0.00	0.00	8
4907	0.00	0.00	0.00	9
4908	0.00	0.00	0.00	7
4909	0.00	0.00	0.00	13
4910	0.00	0.00	0.00	10
4911	0.00	0.00	0.00	10
4912	0.00	0.00	0.00	9
4913	0.00	0.00	0.00	13
4914	0.00	0.00	0.00	14
4915	0.00	0.00	0.00	12
4916	0.00	0.00	0.00	6
4917	0.00	0.00	0.00	8
4918	0.00	0.00	0.00	6
4919	0.00	0.00	0.00	6
4920	0.00	0.00	0.00	15
4921	0.00	0.00	0.00	10
4922	0.00	0.00	0.00	12
4923	0.00	0.00	0.00	7
4924	0.00	0.00	0.00	16
4925	0.00	0.00	0.00	13
4926	0.00	0.00	0.00	10
4927	0.00	0.00	0.00	8
4928	0.00	0.00	0.00	10
4929	0.00	0.00	0.00	10
4930	0.00	0.00	0.00	12
4931	0.00	0.00	0.00	11
4932	0.00	0.00	0.00	10
4933	0.00	0.00	0.00	11
4934	0.00	0.00	0.00	7
4935	0.00	0.00	0.00	13
4936	0.00	0.00	0.00	10
4937	0.00	0.00	0.00	13
4938	0.00	0.00	0.00	17
4939	0.00	0.00	0.00	13
4940	0.00	0.00	0.00	15
4941	0.00	0.00	0.00	13
4942	0.00	0.00	0.00	15
4943	0.00	0.00	0.00	13
4944	0.00	0.00	0.00	10
4945	0.00	0.00	0.00	9
4946	0.00	0.00	0.00	13
4947	0.00	0.00	0.00	7
4948	0.00	0.00	0.00	10
4949	0.00	0.00	0.00	9
4950	0.00	0.00	0.00	13
4951	0.00	0.00	0.00	12
4952	0.00	0.00	0.00	8
4953	0.00	0.00	0.00	14
4954	0.00	0.00	0.00	11
4955	0.00	0.00	0.00	11
4956	0.00	0.00	0.00	11
4957	0.00	0.00	0.00	8
4958	0.00	0.00	0.00	8
4959	0.00	0.00	0.00	13
4960	0.00	0.00	0.00	9
4961	0.00	0.00	0.00	12
4962	0.00	0.00	0.00	8
4963	0.00	0.00	0.00	3
4964	0.00	0.00	0.00	8
4965	0.00	0.00	0.00	14
4966	0.00	0.00	0.00	9
4967	0.00	0.00	0.00	12
4968	0.00	0.00	0.00	8
4969	0.00	0.00	0.00	7
4970	0.00	0.00	0.00	11
4971	0.00	0.00	0.00	8
4972	0.00	0.00	0.00	13
4973	0.00	0.00	0.00	12
4974	0.00	0.00	0.00	9
4975	0.00	0.00	0.00	14
4976	0.00	0.00	0.00	12
4977	0.00	0.00	0.00	8
4978	0.00	0.00	0.00	16
4979	0.00	0.00	0.00	12
4980	0.00	0.00	0.00	6
4981	0.00	0.00	0.00	15
4982	0.00	0.00	0.00	4
4983	0.00	0.00	0.00	8

4984	0.00	0.00	0.00	9
4985	0.00	0.00	0.00	13
4986	0.00	0.00	0.00	14
4987	0.00	0.00	0.00	7
4988	0.00	0.00	0.00	12
4989	0.00	0.00	0.00	15
4990	0.00	0.00	0.00	9
4991	0.00	0.00	0.00	13
4992	0.00	0.00	0.00	10
4993	0.00	0.00	0.00	8
4994	0.00	0.00	0.00	10
4995	0.00	0.00	0.00	11
4996	0.00	0.00	0.00	10
4997	0.00	0.00	0.00	4
4998	0.00	0.00	0.00	13
4999	0.00	0.00	0.00	8
5000	0.00	0.00	0.00	11
5001	0.00	0.00	0.00	5
5002	0.00	0.00	0.00	9
5003	0.00	0.00	0.00	6
5004	0.00	0.00	0.00	10
5005	0.00	0.00	0.00	8
5006	0.00	0.00	0.00	15
5007	0.00	0.00	0.00	14
5008	1.00	0.12	0.22	8
5009	0.00	0.00	0.00	10
5010	0.00	0.00	0.00	11
5011	0.00	0.00	0.00	10
5012	0.00	0.00	0.00	11
5013	0.00	0.00	0.00	14
5014	0.00	0.00	0.00	8
5015	0.00	0.00	0.00	14
5016	0.00	0.00	0.00	14
5017	0.00	0.00	0.00	11
5018	0.00	0.00	0.00	9
5019	0.00	0.00	0.00	14
5020	0.00	0.00	0.00	10
5021	0.00	0.00	0.00	15
5022	0.00	0.00	0.00	11
5023	0.00	0.00	0.00	6
5024	0.00	0.00	0.00	14
5025	0.00	0.00	0.00	8
5026	0.00	0.00	0.00	14
5027	0.00	0.00	0.00	6
5028	0.00	0.00	0.00	13
5029	0.00	0.00	0.00	5
5030	0.00	0.00	0.00	15
5031	0.00	0.00	0.00	8
5032	0.00	0.00	0.00	12
5033	0.00	0.00	0.00	13
5034	0.00	0.00	0.00	8
5035	0.00	0.00	0.00	11
5036	0.00	0.00	0.00	11
5037	0.00	0.00	0.00	12
5038	0.00	0.00	0.00	12
5039	0.00	0.00	0.00	17
5040	0.00	0.00	0.00	8
5041	0.00	0.00	0.00	9
5042	0.00	0.00	0.00	9
5043	0.00	0.00	0.00	14
5044	0.00	0.00	0.00	11
5045	0.00	0.00	0.00	9
5046	0.00	0.00	0.00	10
5047	0.00	0.00	0.00	10
5048	0.00	0.00	0.00	7
5049	0.00	0.00	0.00	9
5050	0.00	0.00	0.00	5
5051	0.00	0.00	0.00	10
5052	0.00	0.00	0.00	10
5053	0.00	0.00	0.00	14
5054	0.00	0.00	0.00	13
5055	0.00	0.00	0.00	7
5056	0.00	0.00	0.00	15
5057	0.00	0.00	0.00	8
5058	0.00	0.00	0.00	11
5059	0.00	0.00	0.00	9
5060	0.00	0.00	0.00	13
5061	0.00	0.00	0.00	13
5062	0.00	0.00	0.00	7
5063	0.00	0.00	0.00	14
5064	0.00	0.00	0.00	8
5065	0.00	0.00	0.00	6
5066	0.00	0.00	0.00	7

5067	0.00	0.00	0.00	10
5068	0.00	0.00	0.00	12
5069	0.00	0.00	0.00	9
5070	0.00	0.00	0.00	11
5071	0.00	0.00	0.00	8
5072	0.00	0.00	0.00	4
5073	0.00	0.00	0.00	14
5074	0.00	0.00	0.00	11
5075	0.00	0.00	0.00	14
5076	0.00	0.00	0.00	7
5077	0.00	0.00	0.00	10
5078	0.00	0.00	0.00	11
5079	0.00	0.00	0.00	10
5080	0.00	0.00	0.00	13
5081	0.00	0.00	0.00	12
5082	0.00	0.00	0.00	8
5083	0.00	0.00	0.00	15
5084	0.00	0.00	0.00	15
5085	0.00	0.00	0.00	11
5086	0.00	0.00	0.00	12
5087	0.00	0.00	0.00	9
5088	0.00	0.00	0.00	4
5089	0.00	0.00	0.00	8
5090	0.00	0.00	0.00	11
5091	0.00	0.00	0.00	6
5092	0.00	0.00	0.00	9
5093	0.00	0.00	0.00	10
5094	0.00	0.00	0.00	18
5095	0.00	0.00	0.00	6
5096	0.00	0.00	0.00	12
5097	0.00	0.00	0.00	9
5098	0.00	0.00	0.00	11
5099	0.00	0.00	0.00	7
5100	0.00	0.00	0.00	12
5101	0.00	0.00	0.00	7
5102	0.00	0.00	0.00	5
5103	0.00	0.00	0.00	11
5104	0.00	0.00	0.00	13
5105	0.00	0.00	0.00	10
5106	0.00	0.00	0.00	12
5107	0.00	0.00	0.00	7
5108	0.00	0.00	0.00	14
5109	0.00	0.00	0.00	11
5110	0.00	0.00	0.00	8
5111	0.00	0.00	0.00	10
5112	0.00	0.00	0.00	10
5113	0.00	0.00	0.00	9
5114	0.00	0.00	0.00	13
5115	0.00	0.00	0.00	8
5116	0.00	0.00	0.00	10
5117	0.00	0.00	0.00	8
5118	0.00	0.00	0.00	12
5119	0.00	0.00	0.00	8
5120	0.00	0.00	0.00	7
5121	0.00	0.00	0.00	12
5122	0.00	0.00	0.00	9
5123	0.00	0.00	0.00	9
5124	0.00	0.00	0.00	8
5125	0.00	0.00	0.00	8
5126	0.00	0.00	0.00	8
5127	0.00	0.00	0.00	13
5128	0.00	0.00	0.00	8
5129	0.00	0.00	0.00	9
5130	0.00	0.00	0.00	8
5131	0.00	0.00	0.00	10
5132	0.00	0.00	0.00	11
5133	0.00	0.00	0.00	11
5134	0.00	0.00	0.00	6
5135	0.00	0.00	0.00	11
5136	0.00	0.00	0.00	11
5137	0.00	0.00	0.00	12
5138	0.00	0.00	0.00	8
5139	0.00	0.00	0.00	10
5140	0.00	0.00	0.00	10
5141	0.00	0.00	0.00	10
5142	0.00	0.00	0.00	10
5143	0.00	0.00	0.00	5
5144	0.00	0.00	0.00	13
5145	0.00	0.00	0.00	11
5146	0.00	0.00	0.00	12
5147	0.00	0.00	0.00	9
5148	0.00	0.00	0.00	12
5149	0.00	0.00	0.00	8

5150	0.00	0.00	0.00	11
5151	0.00	0.00	0.00	10
5152	0.00	0.00	0.00	12
5153	0.00	0.00	0.00	12
5154	0.00	0.00	0.00	10
5155	0.00	0.00	0.00	10
5156	0.00	0.00	0.00	9
5157	0.00	0.00	0.00	13
5158	0.00	0.00	0.00	10
5159	0.00	0.00	0.00	6
5160	0.00	0.00	0.00	10
5161	0.00	0.00	0.00	12
5162	0.00	0.00	0.00	8
5163	0.00	0.00	0.00	10
5164	0.00	0.00	0.00	9
5165	0.00	0.00	0.00	11
5166	0.00	0.00	0.00	8
5167	0.00	0.00	0.00	9
5168	0.00	0.00	0.00	9
5169	0.00	0.00	0.00	8
5170	0.00	0.00	0.00	12
5171	0.00	0.00	0.00	6
5172	0.00	0.00	0.00	13
5173	0.00	0.00	0.00	11
5174	0.00	0.00	0.00	7
5175	0.00	0.00	0.00	7
5176	0.00	0.00	0.00	15
5177	0.00	0.00	0.00	10
5178	0.00	0.00	0.00	9
5179	0.00	0.00	0.00	7
5180	0.00	0.00	0.00	7
5181	0.00	0.00	0.00	11
5182	0.00	0.00	0.00	5
5183	0.00	0.00	0.00	17
5184	0.00	0.00	0.00	4
5185	0.00	0.00	0.00	7
5186	0.00	0.00	0.00	7
5187	0.00	0.00	0.00	10
5188	0.00	0.00	0.00	11
5189	0.00	0.00	0.00	13
5190	1.00	0.10	0.18	10
5191	0.00	0.00	0.00	8
5192	0.00	0.00	0.00	14
5193	0.00	0.00	0.00	12
5194	0.00	0.00	0.00	18
5195	0.00	0.00	0.00	10
5196	0.00	0.00	0.00	8
5197	0.00	0.00	0.00	8
5198	0.00	0.00	0.00	8
5199	0.00	0.00	0.00	11
5200	0.00	0.00	0.00	14
5201	0.00	0.00	0.00	12
5202	0.00	0.00	0.00	14
5203	0.00	0.00	0.00	13
5204	0.00	0.00	0.00	8
5205	0.00	0.00	0.00	10
5206	0.00	0.00	0.00	16
5207	0.00	0.00	0.00	9
5208	0.00	0.00	0.00	6
5209	0.00	0.00	0.00	8
5210	0.00	0.00	0.00	11
5211	0.00	0.00	0.00	11
5212	0.00	0.00	0.00	14
5213	0.00	0.00	0.00	6
5214	0.00	0.00	0.00	8
5215	0.00	0.00	0.00	11
5216	0.00	0.00	0.00	11
5217	0.00	0.00	0.00	9
5218	0.00	0.00	0.00	9
5219	0.00	0.00	0.00	10
5220	0.00	0.00	0.00	10
5221	0.00	0.00	0.00	10
5222	0.00	0.00	0.00	8
5223	0.00	0.00	0.00	8
5224	0.00	0.00	0.00	7
5225	0.00	0.00	0.00	7
5226	0.00	0.00	0.00	8
5227	0.00	0.00	0.00	13
5228	0.00	0.00	0.00	7
5229	0.00	0.00	0.00	6
5230	0.00	0.00	0.00	7
5231	0.00	0.00	0.00	10
5232	0.00	0.00	0.00	7

5233	0.00	0.00	0.00	9
5234	0.00	0.00	0.00	5
5235	0.00	0.00	0.00	1
5236	0.00	0.00	0.00	16
5237	0.00	0.00	0.00	7
5238	0.00	0.00	0.00	10
5239	0.00	0.00	0.00	14
5240	0.00	0.00	0.00	8
5241	0.00	0.00	0.00	8
5242	0.00	0.00	0.00	8
5243	0.00	0.00	0.00	5
5244	0.00	0.00	0.00	11
5245	0.00	0.00	0.00	8
5246	0.00	0.00	0.00	11
5247	0.00	0.00	0.00	11
5248	0.00	0.00	0.00	10
5249	0.00	0.00	0.00	13
5250	0.00	0.00	0.00	10
5251	0.00	0.00	0.00	12
5252	0.00	0.00	0.00	11
5253	0.00	0.00	0.00	12
5254	0.00	0.00	0.00	12
5255	0.00	0.00	0.00	10
5256	0.00	0.00	0.00	12
5257	0.00	0.00	0.00	11
5258	0.00	0.00	0.00	10
5259	0.00	0.00	0.00	8
5260	0.00	0.00	0.00	11
5261	0.00	0.00	0.00	10
5262	0.00	0.00	0.00	9
5263	0.00	0.00	0.00	10
5264	0.00	0.00	0.00	12
5265	1.00	0.09	0.17	11
5266	0.00	0.00	0.00	8
5267	0.00	0.00	0.00	12
5268	0.00	0.00	0.00	7
5269	0.00	0.00	0.00	9
5270	0.00	0.00	0.00	11
5271	0.00	0.00	0.00	9
5272	0.00	0.00	0.00	11
5273	0.00	0.00	0.00	7
5274	0.00	0.00	0.00	11
5275	0.00	0.00	0.00	11
5276	0.00	0.00	0.00	9
5277	0.00	0.00	0.00	7
5278	0.00	0.00	0.00	7
5279	0.00	0.00	0.00	8
5280	0.00	0.00	0.00	5
5281	0.00	0.00	0.00	8
5282	0.00	0.00	0.00	8
5283	0.00	0.00	0.00	13
5284	0.00	0.00	0.00	11
5285	0.00	0.00	0.00	6
5286	0.00	0.00	0.00	13
5287	0.00	0.00	0.00	15
5288	0.00	0.00	0.00	7
5289	0.00	0.00	0.00	8
5290	0.00	0.00	0.00	6
5291	0.00	0.00	0.00	9
5292	0.00	0.00	0.00	6
5293	0.00	0.00	0.00	9
5294	0.00	0.00	0.00	13
5295	0.00	0.00	0.00	11
5296	0.00	0.00	0.00	10
5297	0.00	0.00	0.00	13
5298	0.00	0.00	0.00	14
5299	0.00	0.00	0.00	10
5300	0.00	0.00	0.00	14
5301	0.00	0.00	0.00	11
5302	0.00	0.00	0.00	6
5303	0.00	0.00	0.00	6
5304	0.00	0.00	0.00	7
5305	0.00	0.00	0.00	9
5306	0.00	0.00	0.00	6
5307	0.00	0.00	0.00	10
5308	0.00	0.00	0.00	11
5309	0.00	0.00	0.00	11
5310	0.00	0.00	0.00	14
5311	0.00	0.00	0.00	10
5312	0.00	0.00	0.00	11
5313	0.00	0.00	0.00	11
5314	0.00	0.00	0.00	11
5315	0.00	0.00	0.00	11

5316	0.00	0.00	0.00	2
5317	0.00	0.00	0.00	5
5318	0.00	0.00	0.00	11
5319	0.00	0.00	0.00	12
5320	0.00	0.00	0.00	7
5321	0.00	0.00	0.00	7
5322	0.00	0.00	0.00	9
5323	0.00	0.00	0.00	9
5324	0.00	0.00	0.00	8
5325	0.00	0.00	0.00	10
5326	0.00	0.00	0.00	3
5327	0.00	0.00	0.00	13
5328	0.00	0.00	0.00	13
5329	0.00	0.00	0.00	7
5330	0.00	0.00	0.00	8
5331	0.00	0.00	0.00	9
5332	0.00	0.00	0.00	8
5333	0.00	0.00	0.00	11
5334	0.00	0.00	0.00	11
5335	0.00	0.00	0.00	6
5336	0.00	0.00	0.00	6
5337	0.00	0.00	0.00	6
5338	0.00	0.00	0.00	11
5339	0.00	0.00	0.00	12
5340	0.00	0.00	0.00	9
5341	0.00	0.00	0.00	8
5342	0.00	0.00	0.00	8
5343	0.00	0.00	0.00	7
5344	0.00	0.00	0.00	5
5345	0.00	0.00	0.00	11
5346	0.00	0.00	0.00	13
5347	0.00	0.00	0.00	10
5348	0.00	0.00	0.00	11
5349	0.00	0.00	0.00	7
5350	0.00	0.00	0.00	10
5351	0.00	0.00	0.00	7
5352	0.00	0.00	0.00	7
5353	0.00	0.00	0.00	11
5354	0.00	0.00	0.00	12
5355	0.00	0.00	0.00	12
5356	0.00	0.00	0.00	10
5357	0.00	0.00	0.00	9
5358	0.00	0.00	0.00	8
5359	0.00	0.00	0.00	7
5360	0.00	0.00	0.00	10
5361	0.00	0.00	0.00	6
5362	0.00	0.00	0.00	6
5363	0.00	0.00	0.00	9
5364	0.00	0.00	0.00	9
5365	0.00	0.00	0.00	17
5366	0.00	0.00	0.00	8
5367	0.00	0.00	0.00	9
5368	0.00	0.00	0.00	8
5369	0.00	0.00	0.00	8
5370	0.00	0.00	0.00	18
5371	0.00	0.00	0.00	14
5372	0.00	0.00	0.00	10
5373	0.00	0.00	0.00	7
5374	0.00	0.00	0.00	6
5375	0.00	0.00	0.00	12
5376	0.00	0.00	0.00	13
5377	0.00	0.00	0.00	9
5378	0.00	0.00	0.00	10
5379	0.00	0.00	0.00	10
5380	0.00	0.00	0.00	9
5381	0.00	0.00	0.00	7
5382	0.00	0.00	0.00	10
5383	0.00	0.00	0.00	9
5384	0.00	0.00	0.00	12
5385	0.00	0.00	0.00	15
5386	0.00	0.00	0.00	7
5387	0.00	0.00	0.00	8
5388	0.00	0.00	0.00	4
5389	0.00	0.00	0.00	7
5390	0.00	0.00	0.00	8
5391	0.00	0.00	0.00	4
5392	0.00	0.00	0.00	10
5393	0.00	0.00	0.00	7
5394	0.00	0.00	0.00	8
5395	0.00	0.00	0.00	16
5396	0.00	0.00	0.00	13
5397	0.00	0.00	0.00	11
5398	0.00	0.00	0.00	5

5399	0.00	0.00	0.00	5
5400	0.00	0.00	0.00	12
5401	0.00	0.00	0.00	7
5402	0.00	0.00	0.00	5
5403	0.00	0.00	0.00	12
5404	0.00	0.00	0.00	5
5405	0.00	0.00	0.00	10
5406	0.00	0.00	0.00	7
5407	0.00	0.00	0.00	12
5408	0.00	0.00	0.00	9
5409	0.00	0.00	0.00	9
5410	0.00	0.00	0.00	8
5411	0.00	0.00	0.00	6
5412	0.00	0.00	0.00	8
5413	0.00	0.00	0.00	6
5414	0.00	0.00	0.00	8
5415	0.00	0.00	0.00	16
5416	0.00	0.00	0.00	9
5417	0.00	0.00	0.00	11
5418	0.00	0.00	0.00	9
5419	0.00	0.00	0.00	14
5420	0.00	0.00	0.00	6
5421	0.00	0.00	0.00	11
5422	0.00	0.00	0.00	12
5423	0.00	0.00	0.00	8
5424	0.00	0.00	0.00	13
5425	0.00	0.00	0.00	4
5426	0.00	0.00	0.00	10
5427	0.00	0.00	0.00	9
5428	0.00	0.00	0.00	12
5429	0.00	0.00	0.00	11
5430	0.00	0.00	0.00	9
5431	0.00	0.00	0.00	15
5432	0.00	0.00	0.00	12
5433	0.00	0.00	0.00	8
5434	0.00	0.00	0.00	6
5435	0.00	0.00	0.00	12
5436	0.00	0.00	0.00	11
5437	0.00	0.00	0.00	10
5438	0.00	0.00	0.00	7
5439	0.00	0.00	0.00	9
5440	0.00	0.00	0.00	12
5441	0.00	0.00	0.00	10
5442	0.00	0.00	0.00	7
5443	0.00	0.00	0.00	12
5444	0.00	0.00	0.00	7
5445	0.00	0.00	0.00	9
5446	0.00	0.00	0.00	7
5447	0.00	0.00	0.00	6
5448	0.00	0.00	0.00	12
5449	0.00	0.00	0.00	9
5450	0.00	0.00	0.00	10
5451	0.00	0.00	0.00	6
5452	0.00	0.00	0.00	11
5453	0.00	0.00	0.00	7
5454	0.00	0.00	0.00	9
5455	0.00	0.00	0.00	11
5456	0.00	0.00	0.00	7
5457	0.00	0.00	0.00	9
5458	0.00	0.00	0.00	8
5459	0.00	0.00	0.00	11
5460	0.00	0.00	0.00	7
5461	0.00	0.00	0.00	11
5462	0.00	0.00	0.00	10
5463	0.00	0.00	0.00	9
5464	0.00	0.00	0.00	9
5465	0.00	0.00	0.00	7
5466	0.00	0.00	0.00	9
5467	0.00	0.00	0.00	14
5468	0.00	0.00	0.00	9
5469	0.00	0.00	0.00	12
5470	0.00	0.00	0.00	11
5471	0.00	0.00	0.00	8
5472	0.00	0.00	0.00	15
5473	0.00	0.00	0.00	4
5474	0.00	0.00	0.00	8
5475	0.00	0.00	0.00	9
5476	0.00	0.00	0.00	11
5477	0.00	0.00	0.00	8
5478	0.00	0.00	0.00	6
5479	0.00	0.00	0.00	7
5480	0.00	0.00	0.00	7
5481	0.00	0.00	0.00	10

```
       5482       0.00      0.00      0.00        12
       5483       0.00      0.00      0.00         6
       5484       0.00      0.00      0.00         9
       5485       0.00      0.00      0.00         8
       5486       0.00      0.00      0.00         8
       5487       0.00      0.00      0.00         9
       5488       0.00      0.00      0.00         7
       5489       0.00      0.00      0.00        10
       5490       0.00      0.00      0.00        12
       5491       0.00      0.00      0.00         6
       5492       0.00      0.00      0.00         8
       5493       0.00      0.00      0.00        13
       5494       0.00      0.00      0.00         6
       5495       0.00      0.00      0.00        10
       5496       0.00      0.00      0.00         7
       5497       0.00      0.00      0.00         9
       5498       0.00      0.00      0.00         6
       5499       0.00      0.00      0.00        13

avg / total       0.53      0.26      0.33    530065
```

In [0]:

```
from sklearn.externals import joblib
joblib.dump(classifier1, 'lr_with_equal_weight.pkl')
```

4.5 Modeling with less data points (0.5M data points) and more weight to title and 500 tags only.

In [5]:

```
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 [22]:

```
# 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 = 400000
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 500001;")
        # 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 Special 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 [23]:

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

In [24]:

```
#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)+" "+question+" "+str(tags)
#     else:
#         question=str(title)+" "+str(title)+" "+str(title)+" "+question

    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 stop_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,words_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("\nAverage length of questions(Title+Body) before processing: %d"%no_dup_avg_len_pre)
print("Average length of questions(Title+Body) after processing: %d"%no_dup_avg_len_post)
print("Percentage 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
Number of questions completed= 200000
Number of questions completed= 300000
Number of questions completed= 400000
Number of questions completed= 500000

Average length of questions(Title+Body) before processing: 1239
Average length of questions(Title+Body) after processing: 424
Percentage of questions containing code: 57
Time taken to run this cell : 0:25:16.112802
```

In [25]:

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

Sample quesitons after preprocessing of data __

In [26]:

```
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 10")
        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 bi
nd datagrid dynam code wrote code debug code block seem bind correct grid come column form come grid
column although necessari bind nthank repli advance..',)
----------------------------------------------------------------------------------------------------
('java.lang.noclassdeffounderror javax servlet jsp tagext taglibraryvalid java.lang.noclassdeffounde
rror javax servlet jsp tagext taglibraryvalid java.lang.noclassdeffounderror javax servlet jsp tagex
t taglibraryvalid follow guid link instal jstl got follow error tri launch jsp page java.lang.noclas
sdeffounderror javax servlet jsp tagext taglibraryvalid taglib declar instal jstl 1.1 tomcat webapp
tri project work also tri version 1.2 jstl still messag caus solv',)
----------------------------------------------------------------------------------------------------
('java.sql.sqlexcept microsoft odbc driver manag invalid descriptor index java.sql.sqlexcept microso
ft odbc driver manag invalid descriptor index java.sql.sqlexcept microsoft odbc driver manag invalid
descriptor index use follow code display caus solv',)
----------------------------------------------------------------------------------------------------
('better way updat feed fb php sdk better way updat feed fb php sdk better way updat feed fb php sdk
novic facebook api read mani tutori still confused.i find post feed api method like correct second w
ay use curl someth like way better',)
----------------------------------------------------------------------------------------------------
('btnadd click event open two window record ad btnadd click event open two window record ad btnadd c
lick event open two window record ad open window search.aspx use code hav add button search.aspx nwh
en insert record btnadd click event open anoth window nafter insert record close window',)
----------------------------------------------------------------------------------------------------
('sql inject issu prevent correct form submiss php sql inject issu prevent correct form submiss php
sql inject issu prevent correct form submiss php check everyth think make sure input field safe type
sql inject good 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 execut see data post none forum field post problem
use someth titl field none data get post current use print post see submit noth work flawless statem
ent though also mention script work flawless local machin use host come across problem state list in
put test mess',)
----------------------------------------------------------------------------------------------------
('countabl subaddit lebesgu measur countabl subaddit lebesgu measur countabl subaddit lebesgu measur
let lbrace rbrace sequenc set sigma -algebra mathcal want show left bigcup right leq sum left right
countabl addit measur defin set sigma algebra mathcal think use monoton properti somewher proof star
t appreci littl help nthank ad han answer make follow addit construct given han answer clear bigcup
bigcup cap emptyset neq left bigcup right left bigcup right sum left right also construct subset mon
oton left right leq left right final would sum leq sum result follow',)
----------------------------------------------------------------------------------------------------
('hql equival sql queri hql equival sql queri hql equival sql queri hql queri replac name class prop
erti name error occur hql error',)
----------------------------------------------------------------------------------------------------
('undefin symbol architectur i386 objc class skpsmtpmessag referenc error undefin symbol architectur
i386 objc class skpsmtpmessag referenc error undefin symbol architectur i386 objc class skpsmtpmessa
g referenc error import framework send email applic background import framework i.e skpsmtpmessag so
mebodi suggest get error collect2 ld return exit status import framework correct sorc taken framewor
k follow mfmailcomposeviewcontrol question lock field updat answer drag drop folder project click co
pi nthat',)
----------------------------------------------------------------------------------------------------
```

Loading Preprocessed data with 3 times more title weight __

In [9]:

```
#Taking 0.5 Million entries to a dataframe.
read_db = 'Titlemoreweight.db'
if os.path.isfile(read_db):
    conn_r = create_connection(read_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 [10]:

```
preprocessed_data.head()
```

Out[10]:

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

In [11]:

```
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 : 500000
Number of dimensions : 2
```

Converting string Tags to multilable output variables __

In [12]:

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

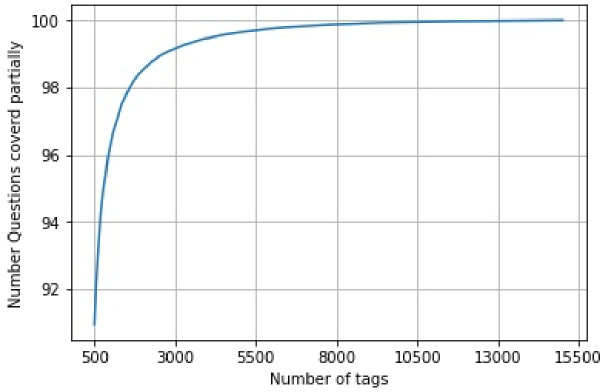
Selecting 500 Tags __

In [16]:

```
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 [17]:

```
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, minimun is 500(it covers 90% of the tags)
print("With ",5500,"tags we are covering ",questions_explained[50],"% of questions")
print("With ",500,"tags we are covering ",questions_explained[0],"% of questions")
```




```
With  5500 tags we are covering  99.157 % of questions
With  500 tags we are covering  90.956 % of questions
```

In [18]:

```
# 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:  45221 out of  500000
```

In [44]:

```
x_train=preprocessed_data.head(train_datasize)
x_test=preprocessed_data.tail(preprocessed_data.shape[0] - 400000)

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

In [46]:

```
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:  (400000, 500)
Number of data points in test data:  (100000, 500)
```

4.5.2 Featurizing data with TfIdf vectorizer

In [48]:

```
start = datetime.now()
vectorizer = TfidfVectorizer(min_df=0.00009, max_features=200000, smooth_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 : 0:04:14.145802
```

In [49]:

```
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)
```

```
Dimensions of train data X: (400000, 94927) Y : (400000, 500)
Dimensions of test data X: (100000, 94927) Y: (100000, 500)
```

4.5.3 Applying Logistic Regression with SGDClassifier and OneVsRest Classifier

In [54]:

```
start = datetime.now()
classifier2 = OneVsRestclassifier2(SGDclassifier2(loss='log', alpha=0.00001, penalty='l1'), n_jobs=-1)
classifier2.fit(x_train_multilabel, y_train)
predictions = classifier2.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')

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

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

```
Accuracy : 0.23682
Hamming loss  0.00277832
Micro-average quality numbers
Precision: 0.7222, Recall: 0.3263, F1-measure: 0.4495
Macro-average quality numbers
Precision: 0.5515, Recall: 0.2584, F1-measure: 0.3354

Classification Report
             precision    recall  f1-score   support

          0       0.94      0.64      0.76      5519
          1       0.68      0.26      0.38      8190
          2       0.82      0.38      0.51      6529
          3       0.81      0.43      0.57      3231
          4       0.81      0.41      0.54      6430
          5       0.81      0.34      0.48      2879
          6       0.87      0.50      0.63      5086
          7       0.88      0.54      0.67      4533
          8       0.60      0.13      0.22      3000
          9       0.81      0.52      0.64      2765
         10       0.59      0.17      0.27      3051
         11       0.69      0.33      0.45      3009
         12       0.65      0.24      0.35      2630
         13       0.71      0.23      0.35      1426
         14       0.90      0.53      0.67      2548
         15       0.68      0.18      0.29      2371
         16       0.65      0.23      0.34       873
         17       0.89      0.61      0.72      2151
         18       0.63      0.23      0.34      2204
         19       0.71      0.42      0.53       831
         20       0.77      0.41      0.53      1860
         21       0.28      0.08      0.12      2023
         22       0.49      0.21      0.30      1513
         23       0.91      0.49      0.64      1207
         24       0.57      0.28      0.38       506
         25       0.68      0.30      0.41       425
         26       0.65      0.40      0.50       793
         27       0.60      0.31      0.41      1291
         28       0.74      0.36      0.48      1208
         29       0.45      0.10      0.16       406
         30       0.73      0.18      0.29       504
         31       0.28      0.10      0.15       732
         32       0.57      0.24      0.34       441
         33       0.57      0.18      0.27      1645
         34       0.71      0.25      0.37      1058
         35       0.83      0.54      0.66       946
         36       0.68      0.19      0.30       644
         37       0.98      0.66      0.79       136
         38       0.63      0.36      0.46       570
         39       0.85      0.29      0.43       766
         40       0.62      0.28      0.39      1132
         41       0.45      0.19      0.27       174
         42       0.80      0.52      0.63       210
         43       0.81      0.41      0.54       433
         44       0.66      0.50      0.57       626
```

45	0.74	0.32	0.45	852
46	0.75	0.42	0.54	534
47	0.32	0.13	0.18	350
48	0.75	0.50	0.60	496
49	0.80	0.62	0.70	785
50	0.16	0.03	0.05	475
51	0.34	0.10	0.16	305
52	0.42	0.03	0.06	251
53	0.68	0.40	0.50	914
54	0.46	0.16	0.24	728
55	0.18	0.01	0.01	258
56	0.46	0.19	0.27	821
57	0.47	0.09	0.15	541
58	0.78	0.28	0.41	748
59	0.94	0.62	0.75	724
60	0.35	0.07	0.11	660
61	0.82	0.17	0.29	235
62	0.91	0.70	0.79	718
63	0.83	0.63	0.72	468
64	0.52	0.29	0.37	191
65	0.36	0.12	0.18	429
66	0.27	0.05	0.09	415
67	0.75	0.48	0.58	274
68	0.82	0.52	0.64	510
69	0.67	0.44	0.54	466
70	0.30	0.07	0.11	305
71	0.51	0.16	0.25	247
72	0.79	0.48	0.60	401
73	0.98	0.73	0.84	86
74	0.74	0.38	0.51	120
75	0.89	0.68	0.77	129
76	0.67	0.01	0.02	473
77	0.38	0.25	0.30	143
78	0.80	0.45	0.57	347
79	0.73	0.23	0.35	479
80	0.55	0.32	0.41	279
81	0.77	0.17	0.28	461
82	0.24	0.01	0.03	298
83	0.77	0.45	0.57	396
84	0.55	0.33	0.41	184
85	0.69	0.20	0.31	573
86	0.50	0.05	0.09	325
87	0.50	0.28	0.36	273
88	0.41	0.21	0.28	135
89	0.31	0.07	0.12	232
90	0.57	0.31	0.40	409
91	0.63	0.25	0.36	420
92	0.76	0.53	0.62	408
93	0.69	0.49	0.57	241
94	0.33	0.04	0.08	211
95	0.35	0.09	0.14	277
96	0.25	0.03	0.06	410
97	0.90	0.32	0.47	501
98	0.75	0.59	0.66	136
99	0.53	0.28	0.37	239
100	0.54	0.13	0.21	324
101	0.92	0.60	0.73	277
102	0.92	0.70	0.80	613
103	0.50	0.16	0.24	157
104	0.21	0.06	0.09	295
105	0.84	0.34	0.49	334
106	0.80	0.13	0.22	335
107	0.76	0.48	0.59	389
108	0.56	0.23	0.32	251
109	0.53	0.42	0.47	317
110	0.79	0.08	0.15	187
111	0.57	0.09	0.16	140
112	0.60	0.27	0.37	154
113	0.65	0.19	0.29	332
114	0.45	0.26	0.33	323
115	0.49	0.22	0.31	344
116	0.76	0.49	0.60	370
117	0.59	0.23	0.33	313
118	0.78	0.67	0.72	874
119	0.45	0.20	0.28	293
120	0.00	0.00	0.00	200
121	0.77	0.47	0.58	463
122	0.37	0.08	0.14	119
123	0.75	0.01	0.02	256
124	0.91	0.70	0.79	195
125	0.41	0.12	0.18	138
126	0.80	0.48	0.60	376
127	0.18	0.04	0.07	122

128	0.16	0.04	0.06	252
129	0.51	0.14	0.22	144
130	0.39	0.09	0.14	150
131	0.23	0.01	0.03	210
132	0.66	0.26	0.37	361
133	0.94	0.54	0.69	453
134	0.89	0.73	0.81	124
135	0.30	0.03	0.06	91
136	0.70	0.26	0.38	128
137	0.58	0.33	0.43	218
138	0.77	0.15	0.25	243
139	0.38	0.19	0.25	149
140	0.75	0.43	0.55	318
141	0.31	0.13	0.19	159
142	0.65	0.35	0.46	274
143	0.86	0.72	0.78	362
144	0.58	0.16	0.25	118
145	0.65	0.37	0.47	164
146	0.59	0.28	0.38	461
147	0.66	0.40	0.50	159
148	0.34	0.14	0.20	166
149	0.98	0.46	0.62	346
150	0.61	0.07	0.13	350
151	0.90	0.64	0.74	55
152	0.79	0.45	0.58	387
153	0.48	0.09	0.16	150
154	0.59	0.12	0.20	281
155	0.27	0.05	0.09	202
156	0.76	0.61	0.68	130
157	0.27	0.07	0.11	245
158	0.89	0.58	0.70	177
159	0.48	0.24	0.32	130
160	0.51	0.13	0.21	336
161	0.93	0.59	0.72	220
162	0.16	0.03	0.05	229
163	0.89	0.41	0.56	316
164	0.75	0.34	0.47	283
165	0.63	0.31	0.42	197
166	0.51	0.24	0.32	101
167	0.47	0.19	0.27	231
168	0.59	0.23	0.33	370
169	0.41	0.18	0.25	258
170	0.32	0.06	0.10	101
171	0.40	0.22	0.29	89
172	0.51	0.34	0.41	193
173	0.41	0.21	0.28	309
174	0.51	0.13	0.21	172
175	0.94	0.76	0.84	95
176	0.94	0.59	0.72	346
177	0.93	0.43	0.58	322
178	0.63	0.46	0.53	232
179	0.30	0.06	0.09	125
180	0.53	0.27	0.36	145
181	0.37	0.09	0.15	77
182	0.17	0.02	0.04	182
183	0.62	0.32	0.42	257
184	0.04	0.00	0.01	216
185	0.35	0.06	0.11	242
186	0.40	0.16	0.23	165
187	0.76	0.57	0.65	263
188	0.35	0.10	0.15	174
189	0.70	0.28	0.40	136
190	0.88	0.49	0.63	202
191	0.39	0.11	0.17	134
192	0.72	0.40	0.52	230
193	0.42	0.18	0.25	90
194	0.58	0.47	0.52	185
195	0.18	0.04	0.06	156
196	0.42	0.07	0.12	160
197	0.63	0.06	0.12	266
198	0.40	0.06	0.10	284
199	0.40	0.06	0.10	145
200	0.94	0.69	0.80	212
201	0.68	0.21	0.33	317
202	0.76	0.54	0.63	427
203	0.30	0.08	0.12	232
204	0.52	0.23	0.31	217
205	0.48	0.43	0.46	527
206	0.14	0.02	0.03	124
207	0.47	0.09	0.15	103
208	0.90	0.48	0.63	287
209	0.33	0.08	0.13	193
210	0.69	0.31	0.43	220

211	0.85	0.20	0.32	140
212	0.15	0.02	0.03	161
213	0.52	0.24	0.32	72
214	0.61	0.46	0.52	396
215	0.86	0.33	0.48	134
216	0.52	0.06	0.11	400
217	0.49	0.23	0.31	75
218	0.97	0.75	0.85	219
219	0.77	0.34	0.47	210
220	0.90	0.59	0.71	298
221	0.97	0.59	0.74	266
222	0.77	0.41	0.54	290
223	0.09	0.01	0.01	128
224	0.78	0.38	0.51	159
225	0.58	0.30	0.39	164
226	0.61	0.35	0.44	144
227	0.58	0.32	0.41	276
228	0.17	0.02	0.03	235
229	0.33	0.02	0.04	216
230	0.35	0.18	0.23	228
231	0.71	0.47	0.57	64
232	0.35	0.06	0.10	103
233	0.73	0.31	0.43	216
234	0.69	0.08	0.14	116
235	0.54	0.36	0.43	77
236	0.96	0.64	0.77	67
237	0.58	0.07	0.12	218
238	0.38	0.08	0.13	139
239	0.17	0.01	0.02	94
240	0.53	0.27	0.36	77
241	0.52	0.09	0.15	167
242	0.83	0.29	0.43	86
243	0.43	0.17	0.25	58
244	0.62	0.17	0.27	269
245	0.17	0.05	0.08	112
246	0.95	0.74	0.83	255
247	0.44	0.21	0.28	58
248	0.25	0.02	0.04	81
249	0.00	0.00	0.00	131
250	0.40	0.20	0.27	93
251	0.66	0.29	0.40	154
252	0.40	0.05	0.08	129
253	0.60	0.29	0.39	83
254	0.40	0.09	0.15	191
255	0.18	0.03	0.05	219
256	0.32	0.05	0.08	130
257	0.46	0.29	0.36	93
258	0.69	0.43	0.53	217
259	0.32	0.09	0.14	141
260	0.95	0.13	0.23	143
261	0.52	0.11	0.17	219
262	0.54	0.28	0.37	107
263	0.40	0.23	0.29	236
264	0.28	0.17	0.21	119
265	0.33	0.14	0.20	72
266	0.00	0.00	0.00	70
267	0.34	0.15	0.21	107
268	0.67	0.44	0.53	169
269	0.27	0.09	0.14	129
270	0.73	0.53	0.61	159
271	0.82	0.37	0.51	190
272	0.61	0.22	0.32	248
273	0.91	0.71	0.80	264
274	0.89	0.65	0.75	105
275	0.50	0.07	0.12	104
276	0.14	0.02	0.03	115
277	0.83	0.59	0.69	170
278	0.65	0.23	0.34	145
279	0.92	0.62	0.74	230
280	0.55	0.39	0.46	80
281	0.68	0.55	0.61	217
282	0.74	0.46	0.57	175
283	0.32	0.06	0.10	269
284	0.63	0.26	0.37	74
285	0.86	0.50	0.63	206
286	0.90	0.59	0.71	227
287	0.89	0.31	0.46	130
288	0.35	0.06	0.11	129
289	0.40	0.03	0.05	80
290	0.15	0.07	0.10	99
291	0.77	0.31	0.45	208
292	0.29	0.03	0.05	67
293	0.82	0.43	0.57	109

294	0.41	0.26	0.32	140
295	0.24	0.08	0.12	241
296	0.22	0.08	0.12	72
297	0.18	0.03	0.05	107
298	0.80	0.39	0.53	61
299	0.93	0.36	0.52	77
300	0.19	0.07	0.10	111
301	0.00	0.00	0.00	126
302	0.00	0.00	0.00	73
303	0.56	0.34	0.42	176
304	0.96	0.73	0.83	230
305	0.97	0.58	0.73	156
306	0.49	0.34	0.40	146
307	0.29	0.08	0.13	98
308	0.00	0.00	0.00	78
309	0.78	0.07	0.14	94
310	0.77	0.35	0.48	162
311	0.82	0.51	0.63	116
312	0.48	0.26	0.34	57
313	0.75	0.05	0.09	65
314	0.50	0.36	0.42	138
315	0.53	0.20	0.29	195
316	0.43	0.23	0.30	69
317	0.32	0.10	0.15	134
318	0.50	0.35	0.41	148
319	0.83	0.43	0.57	161
320	0.20	0.14	0.17	104
321	0.85	0.52	0.65	156
322	0.57	0.31	0.40	134
323	0.57	0.37	0.45	232
324	0.41	0.16	0.23	92
325	0.45	0.30	0.36	197
326	0.13	0.02	0.04	126
327	0.44	0.03	0.06	115
328	0.98	0.65	0.78	198
329	0.62	0.30	0.41	125
330	0.74	0.17	0.28	81
331	0.53	0.09	0.15	94
332	1.00	0.02	0.04	56
333	0.13	0.03	0.05	260
334	0.20	0.03	0.06	60
335	0.28	0.09	0.14	110
336	0.63	0.41	0.50	71
337	0.13	0.03	0.05	66
338	0.46	0.34	0.39	150
339	0.00	0.00	0.00	54
340	0.85	0.54	0.66	195
341	0.94	0.20	0.33	79
342	0.50	0.21	0.30	38
343	0.68	0.40	0.50	43
344	0.48	0.21	0.29	68
345	0.67	0.38	0.49	73
346	0.30	0.03	0.05	116
347	0.89	0.35	0.50	111
348	0.29	0.10	0.14	63
349	0.83	0.60	0.69	104
350	0.62	0.45	0.53	44
351	0.70	0.17	0.28	40
352	0.98	0.39	0.56	136
353	0.44	0.22	0.30	54
354	0.36	0.03	0.06	134
355	0.50	0.28	0.35	120
356	0.54	0.24	0.33	228
357	0.67	0.27	0.38	269
358	0.69	0.36	0.48	80
359	0.86	0.45	0.59	140
360	0.39	0.13	0.19	125
361	0.90	0.62	0.73	169
362	0.11	0.04	0.05	56
363	0.94	0.66	0.77	154
364	0.40	0.07	0.12	58
365	0.24	0.11	0.15	71
366	1.00	0.65	0.79	54
367	0.33	0.04	0.08	116
368	0.00	0.00	0.00	54
369	0.00	0.00	0.00	71
370	0.20	0.03	0.06	61
371	0.50	0.08	0.14	71
372	0.65	0.46	0.54	52
373	0.79	0.37	0.50	150
374	0.34	0.13	0.19	93
375	0.15	0.03	0.05	67
376	0.00	0.00	0.00	76

377	0.74	0.16	0.26	106
378	0.30	0.03	0.06	86
379	0.33	0.07	0.12	14
380	1.00	0.40	0.57	122
381	0.19	0.03	0.05	104
382	0.32	0.09	0.14	66
383	0.48	0.26	0.34	110
384	0.00	0.00	0.00	155
385	0.36	0.08	0.13	50
386	0.22	0.09	0.13	64
387	0.41	0.08	0.13	93
388	0.62	0.29	0.40	102
389	0.06	0.01	0.02	108
390	0.96	0.65	0.78	178
391	0.61	0.17	0.27	115
392	0.77	0.40	0.53	42
393	0.00	0.00	0.00	134
394	0.20	0.02	0.03	112
395	0.43	0.12	0.19	176
396	0.50	0.09	0.15	125
397	0.68	0.22	0.33	224
398	0.88	0.60	0.72	63
399	0.00	0.00	0.00	59
400	0.49	0.33	0.40	63
401	0.49	0.19	0.28	98
402	0.57	0.16	0.25	162
403	0.43	0.16	0.23	83
404	0.73	0.84	0.78	19
405	0.29	0.07	0.11	92
406	0.88	0.17	0.29	41
407	0.64	0.33	0.43	43
408	0.81	0.33	0.46	160
409	0.17	0.10	0.13	50
410	0.00	0.00	0.00	19
411	0.39	0.11	0.17	175
412	0.29	0.06	0.09	72
413	0.50	0.05	0.10	95
414	0.18	0.03	0.05	97
415	0.32	0.17	0.22	48
416	0.45	0.30	0.36	83
417	0.50	0.07	0.13	40
418	0.40	0.09	0.14	91
419	0.51	0.30	0.38	90
420	0.31	0.24	0.27	37
421	0.00	0.00	0.00	66
422	0.61	0.34	0.44	73
423	0.48	0.25	0.33	56
424	0.93	0.82	0.87	33
425	0.00	0.00	0.00	76
426	0.25	0.05	0.08	81
427	0.99	0.67	0.80	150
428	0.95	0.66	0.78	29
429	0.99	0.73	0.84	389
430	0.62	0.36	0.46	167
431	0.50	0.08	0.14	123
432	0.47	0.36	0.41	39
433	0.29	0.16	0.20	82
434	1.00	0.64	0.78	66
435	0.66	0.46	0.54	93
436	0.50	0.25	0.34	87
437	0.25	0.06	0.09	86
438	0.75	0.48	0.58	104
439	0.62	0.13	0.21	100
440	0.20	0.01	0.01	141
441	0.43	0.24	0.31	110
442	0.38	0.12	0.19	123
443	0.47	0.11	0.18	71
444	0.41	0.06	0.11	109
445	0.40	0.21	0.27	48
446	0.44	0.26	0.33	76
447	0.28	0.13	0.18	38
448	0.68	0.53	0.60	81
449	0.54	0.14	0.23	132
450	0.46	0.26	0.33	81
451	0.88	0.29	0.44	76
452	0.00	0.00	0.00	44
453	0.00	0.00	0.00	44
454	0.94	0.44	0.60	70
455	0.55	0.08	0.14	155
456	0.44	0.16	0.24	43
457	0.45	0.18	0.26	72
458	0.31	0.08	0.13	62
459	0.75	0.13	0.22	69

```
        460       0.07      0.01      0.02       119
        461       0.77      0.13      0.22        79
        462       0.69      0.23      0.35        47
        463       0.32      0.06      0.10       104
        464       0.65      0.33      0.44       106
        465       0.54      0.11      0.18        64
        466       0.58      0.29      0.39       173
        467       0.81      0.39      0.53       107
        468       0.85      0.13      0.23       126
        469       0.00      0.00      0.00       114
        470       0.94      0.79      0.86       140
        471       0.95      0.23      0.37        79
        472       0.40      0.29      0.34       143
        473       0.70      0.31      0.43       158
        474       0.35      0.06      0.10       138
        475       0.00      0.00      0.00        59
        476       0.58      0.33      0.42        88
        477       0.86      0.57      0.69       176
        478       0.94      0.71      0.81        24
        479       0.08      0.01      0.02        92
        480       0.83      0.48      0.61       100
        481       0.53      0.17      0.26       103
        482       0.47      0.23      0.31        74
        483       0.85      0.59      0.70       105
        484       0.25      0.02      0.04        83
        485       0.22      0.02      0.04        82
        486       0.39      0.13      0.19        71
        487       0.44      0.19      0.27       120
        488       0.33      0.02      0.04       105
        489       0.74      0.32      0.45        87
        490       1.00      0.81      0.90        32
        491       1.00      0.01      0.03        69
        492       0.00      0.00      0.00        49
        493       0.00      0.00      0.00       117
        494       0.52      0.18      0.27        61
        495       0.98      0.65      0.78       344
        496       0.36      0.19      0.25        52
        497       0.61      0.20      0.30       137
        498       0.36      0.04      0.07        98
        499       0.72      0.16      0.27        79

avg / total       0.67      0.33      0.43    173812

Time taken to run this cell : 0:07:13.482763

/home/saugata/anaconda3/lib/python3.6/site-packages/sklearn/metrics/classification.py:1135: Undefine
dMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples.
  'precision', 'predicted', average, warn_for)
/home/saugata/anaconda3/lib/python3.6/site-packages/sklearn/metrics/classification.py:1135: Undefine
dMetricWarning: F-score is ill-defined and being set to 0.0 in labels with no predicted samples.
  'precision', 'predicted', average, warn_for)
/home/saugata/anaconda3/lib/python3.6/site-packages/sklearn/metrics/classification.py:1135: Undefine
dMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predict
ed samples.
  'precision', 'predicted', average, warn_for)
```

In [ ]:

```
joblib.dump(classifier2, 'lr_with_more_title_weight.pkl')
```

4.5.4 Applying Logistic Regression with LogisticRegression and OneVsRest Classifier

In [0]:

```
start = datetime.now()
classifier3 = OneVsRestClassifier(LogisticRegression(penalty='l1'), n_jobs=-1)
classifier3.fit(x_train_multilabel, y_train)
predictions_2 = classifier3.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')

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

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

```
Accuracy : 0.25108
Hamming loss  0.00270302
Micro-average quality numbers
Precision: 0.7172, Recall: 0.3672, F1-measure: 0.4858
Macro-average quality numbers
Precision: 0.5570, Recall: 0.2950, F1-measure: 0.3710
             precision    recall  f1-score   support

          0       0.94      0.72      0.82      5519
          1       0.70      0.34      0.45      8190
          2       0.80      0.42      0.55      6529
          3       0.82      0.49      0.61      3231
          4       0.80      0.44      0.57      6430
          5       0.82      0.38      0.52      2879
          6       0.86      0.53      0.66      5086
          7       0.87      0.58      0.70      4533
          8       0.60      0.13      0.22      3000
          9       0.82      0.57      0.67      2765
         10       0.60      0.20      0.30      3051
         11       0.68      0.38      0.49      3009
         12       0.62      0.29      0.40      2630
         13       0.73      0.30      0.43      1426
         14       0.89      0.57      0.70      2548
         15       0.65      0.23      0.34      2371
         16       0.65      0.25      0.37       873
         17       0.89      0.63      0.74      2151
         18       0.60      0.25      0.35      2204
         19       0.71      0.41      0.52       831
         20       0.76      0.47      0.58      1860
         21       0.29      0.09      0.14      2023
         22       0.52      0.24      0.33      1513
         23       0.89      0.55      0.68      1207
         24       0.56      0.28      0.38       506
         25       0.69      0.34      0.45       425
         26       0.65      0.43      0.52       793
         27       0.62      0.38      0.47      1291
         28       0.74      0.39      0.51      1208
         29       0.46      0.10      0.17       406
         30       0.76      0.21      0.33       504
         31       0.26      0.08      0.12       732
         32       0.60      0.29      0.39       441
         33       0.60      0.27      0.38      1645
         34       0.69      0.26      0.38      1058
         35       0.83      0.58      0.68       946
         36       0.65      0.24      0.35       644
         37       0.98      0.65      0.78       136
         38       0.62      0.38      0.47       570
         39       0.84      0.31      0.45       766
         40       0.59      0.35      0.44      1132
         41       0.47      0.18      0.26       174
         42       0.76      0.49      0.59       210
         43       0.75      0.42      0.54       433
         44       0.66      0.52      0.58       626
         45       0.71      0.36      0.47       852
         46       0.77      0.45      0.57       534
         47       0.37      0.15      0.22       350
```

48	0.75	0.52	0.62	496
49	0.78	0.64	0.71	785
50	0.21	0.06	0.09	475
51	0.37	0.13	0.19	305
52	0.42	0.03	0.06	251
53	0.66	0.40	0.50	914
54	0.49	0.17	0.26	728
55	0.47	0.03	0.05	258
56	0.45	0.24	0.31	821
57	0.46	0.10	0.17	541
58	0.76	0.31	0.45	748
59	0.94	0.66	0.77	724
60	0.35	0.10	0.15	660
61	0.78	0.20	0.31	235
62	0.92	0.74	0.82	718
63	0.83	0.69	0.75	468
64	0.55	0.36	0.43	191
65	0.33	0.11	0.17	429
66	0.29	0.06	0.10	415
67	0.74	0.50	0.59	274
68	0.82	0.53	0.64	510
69	0.67	0.45	0.54	466
70	0.30	0.09	0.13	305
71	0.49	0.17	0.25	247
72	0.78	0.53	0.64	401
73	0.99	0.77	0.86	86
74	0.72	0.42	0.53	120
75	0.92	0.67	0.78	129
76	0.47	0.02	0.04	473
77	0.40	0.29	0.33	143
78	0.79	0.49	0.60	347
79	0.69	0.25	0.36	479
80	0.56	0.34	0.43	279
81	0.70	0.23	0.34	461
82	0.34	0.04	0.07	298
83	0.78	0.50	0.61	396
84	0.55	0.29	0.38	184
85	0.61	0.24	0.35	573
86	0.50	0.07	0.12	325
87	0.51	0.29	0.37	273
88	0.49	0.21	0.30	135
89	0.36	0.11	0.17	232
90	0.56	0.34	0.43	409
91	0.61	0.27	0.37	420
92	0.78	0.57	0.66	408
93	0.66	0.44	0.53	241
94	0.30	0.04	0.07	211
95	0.37	0.10	0.15	277
96	0.28	0.04	0.07	410
97	0.86	0.43	0.57	501
98	0.75	0.63	0.69	136
99	0.54	0.34	0.42	239
100	0.57	0.15	0.24	324
101	0.91	0.68	0.78	277
102	0.91	0.75	0.82	613
103	0.47	0.17	0.25	157
104	0.22	0.06	0.10	295
105	0.75	0.43	0.55	334
106	0.88	0.28	0.43	335
107	0.75	0.54	0.63	389
108	0.58	0.27	0.37	251
109	0.58	0.45	0.51	317
110	0.68	0.10	0.18	187
111	0.73	0.11	0.20	140
112	0.67	0.43	0.52	154
113	0.58	0.20	0.29	332
114	0.46	0.27	0.34	323
115	0.47	0.26	0.33	344
116	0.75	0.55	0.63	370
117	0.58	0.24	0.34	313
118	0.78	0.73	0.75	874
119	0.45	0.21	0.29	293
120	0.11	0.01	0.01	200
121	0.77	0.51	0.61	463
122	0.32	0.10	0.15	119
123	0.67	0.02	0.03	256
124	0.91	0.70	0.79	195
125	0.44	0.14	0.21	138
126	0.81	0.53	0.64	376
127	0.27	0.03	0.06	122
128	0.20	0.04	0.07	252
129	0.48	0.22	0.30	144
130	0.42	0.11	0.18	150

131	0.33	0.03	0.06	210
132	0.65	0.28	0.39	361
133	0.92	0.59	0.72	453
134	0.89	0.77	0.82	124
135	0.31	0.05	0.09	91
136	0.69	0.28	0.40	128
137	0.55	0.38	0.45	218
138	0.67	0.18	0.28	243
139	0.45	0.18	0.26	149
140	0.77	0.46	0.58	318
141	0.32	0.10	0.15	159
142	0.63	0.38	0.47	274
143	0.85	0.79	0.82	362
144	0.54	0.21	0.30	118
145	0.63	0.39	0.48	164
146	0.54	0.31	0.39	461
147	0.68	0.45	0.54	159
148	0.30	0.12	0.17	166
149	0.97	0.55	0.70	346
150	0.64	0.13	0.21	350
151	0.93	0.67	0.78	55
152	0.78	0.52	0.63	387
153	0.51	0.17	0.25	150
154	0.58	0.12	0.21	281
155	0.25	0.06	0.10	202
156	0.81	0.67	0.73	130
157	0.28	0.06	0.10	245
158	0.93	0.63	0.75	177
159	0.53	0.34	0.41	130
160	0.48	0.18	0.26	336
161	0.90	0.65	0.75	220
162	0.28	0.06	0.09	229
163	0.87	0.44	0.58	316
164	0.78	0.44	0.56	283
165	0.60	0.34	0.44	197
166	0.65	0.43	0.51	101
167	0.45	0.18	0.26	231
168	0.56	0.27	0.36	370
169	0.40	0.21	0.27	258
170	0.36	0.08	0.13	101
171	0.38	0.24	0.29	89
172	0.53	0.36	0.43	193
173	0.47	0.26	0.33	309
174	0.62	0.14	0.23	172
175	0.92	0.73	0.81	95
176	0.93	0.62	0.74	346
177	0.86	0.57	0.69	322
178	0.65	0.51	0.57	232
179	0.20	0.04	0.07	125
180	0.65	0.33	0.44	145
181	0.44	0.10	0.17	77
182	0.26	0.06	0.10	182
183	0.60	0.32	0.41	257
184	0.21	0.03	0.05	216
185	0.35	0.09	0.14	242
186	0.43	0.18	0.25	165
187	0.75	0.59	0.66	263
188	0.39	0.12	0.18	174
189	0.75	0.40	0.53	136
190	0.89	0.55	0.68	202
191	0.44	0.16	0.24	134
192	0.68	0.40	0.51	230
193	0.44	0.18	0.25	90
194	0.57	0.48	0.52	185
195	0.26	0.05	0.09	156
196	0.33	0.07	0.11	160
197	0.49	0.10	0.16	266
198	0.47	0.13	0.20	284
199	0.32	0.04	0.07	145
200	0.93	0.74	0.82	212
201	0.65	0.26	0.37	317
202	0.78	0.59	0.67	427
203	0.36	0.11	0.17	232
204	0.51	0.29	0.37	217
205	0.50	0.46	0.48	527
206	0.24	0.03	0.06	124
207	0.50	0.17	0.26	103
208	0.85	0.53	0.65	287
209	0.33	0.11	0.16	193
210	0.75	0.38	0.50	220
211	0.72	0.21	0.32	140
212	0.12	0.02	0.03	161
213	0.63	0.43	0.51	72

214	0.64	0.45	0.53	396
215	0.87	0.34	0.49	134
216	0.61	0.17	0.27	400
217	0.51	0.24	0.33	75
218	0.96	0.76	0.85	219
219	0.77	0.42	0.54	210
220	0.88	0.64	0.74	298
221	0.96	0.70	0.81	266
222	0.76	0.45	0.57	290
223	0.11	0.01	0.01	128
224	0.78	0.45	0.57	159
225	0.55	0.29	0.38	164
226	0.58	0.31	0.41	144
227	0.56	0.29	0.38	276
228	0.19	0.03	0.05	235
229	0.33	0.03	0.06	216
230	0.40	0.17	0.23	228
231	0.70	0.48	0.57	64
232	0.48	0.10	0.16	103
233	0.72	0.35	0.47	216
234	0.72	0.11	0.19	116
235	0.54	0.36	0.43	77
236	0.90	0.67	0.77	67
237	0.57	0.12	0.20	218
238	0.40	0.14	0.20	139
239	0.00	0.00	0.00	94
240	0.54	0.34	0.42	77
241	0.47	0.08	0.14	167
242	0.78	0.37	0.50	86
243	0.40	0.10	0.16	58
244	0.62	0.27	0.38	269
245	0.16	0.04	0.07	112
246	0.95	0.76	0.84	255
247	0.44	0.24	0.31	58
248	0.44	0.05	0.09	81
249	0.23	0.02	0.04	131
250	0.43	0.24	0.31	93
251	0.61	0.29	0.39	154
252	0.36	0.04	0.07	129
253	0.69	0.40	0.50	83
254	0.34	0.08	0.13	191
255	0.15	0.03	0.05	219
256	0.32	0.05	0.09	130
257	0.48	0.26	0.34	93
258	0.65	0.48	0.55	217
259	0.41	0.13	0.20	141
260	0.86	0.17	0.29	143
261	0.62	0.17	0.27	219
262	0.55	0.27	0.36	107
263	0.41	0.27	0.32	236
264	0.33	0.22	0.26	119
265	0.57	0.24	0.33	72
266	0.00	0.00	0.00	70
267	0.36	0.14	0.20	107
268	0.67	0.44	0.53	169
269	0.32	0.14	0.19	129
270	0.74	0.53	0.62	159
271	0.88	0.48	0.62	190
272	0.61	0.27	0.37	248
273	0.90	0.75	0.82	264
274	0.90	0.68	0.77	105
275	0.52	0.12	0.20	104
276	0.08	0.01	0.02	115
277	0.83	0.63	0.72	170
278	0.74	0.41	0.52	145
279	0.90	0.70	0.78	230
280	0.58	0.42	0.49	80
281	0.66	0.54	0.59	217
282	0.75	0.50	0.60	175
283	0.33	0.13	0.18	269
284	0.65	0.32	0.43	74
285	0.82	0.49	0.61	206
286	0.89	0.66	0.75	227
287	0.84	0.41	0.55	130
288	0.32	0.07	0.11	129
289	0.57	0.05	0.09	80
290	0.21	0.09	0.13	99
291	0.76	0.35	0.48	208
292	0.42	0.07	0.13	67
293	0.84	0.48	0.61	109
294	0.46	0.26	0.34	140
295	0.24	0.12	0.16	241
296	0.31	0.12	0.18	72

297	0.44	0.11	0.18	107
298	0.77	0.49	0.60	61
299	0.89	0.51	0.64	77
300	0.21	0.08	0.12	111
301	0.00	0.00	0.00	126
302	0.25	0.01	0.03	73
303	0.57	0.43	0.49	176
304	0.91	0.79	0.85	230
305	0.92	0.72	0.81	156
306	0.50	0.37	0.43	146
307	0.34	0.11	0.17	98
308	0.00	0.00	0.00	78
309	0.80	0.13	0.22	94
310	0.74	0.41	0.53	162
311	0.79	0.51	0.62	116
312	0.52	0.28	0.36	57
313	0.83	0.08	0.14	65
314	0.52	0.36	0.42	138
315	0.54	0.22	0.31	195
316	0.56	0.35	0.43	69
317	0.29	0.13	0.18	134
318	0.56	0.39	0.46	148
319	0.84	0.50	0.63	161
320	0.24	0.19	0.21	104
321	0.82	0.61	0.70	156
322	0.60	0.37	0.46	134
323	0.58	0.44	0.50	232
324	0.34	0.15	0.21	92
325	0.41	0.24	0.31	197
326	0.14	0.03	0.05	126
327	0.20	0.03	0.05	115
328	0.99	0.70	0.82	198
329	0.59	0.32	0.41	125
330	0.73	0.20	0.31	81
331	0.45	0.10	0.16	94
332	0.54	0.12	0.20	56
333	0.19	0.05	0.08	260
334	0.42	0.13	0.20	60
335	0.35	0.08	0.13	110
336	0.62	0.49	0.55	71
337	0.18	0.05	0.07	66
338	0.47	0.36	0.41	150
339	0.00	0.00	0.00	54
340	0.84	0.57	0.68	195
341	0.91	0.52	0.66	79
342	0.38	0.26	0.31	38
343	0.62	0.42	0.50	43
344	0.56	0.29	0.38	68
345	0.62	0.33	0.43	73
346	0.14	0.03	0.04	116
347	0.86	0.43	0.57	111
348	0.33	0.11	0.17	63
349	0.84	0.65	0.74	104
350	0.62	0.48	0.54	44
351	0.57	0.30	0.39	40
352	0.93	0.57	0.70	136
353	0.38	0.15	0.21	54
354	0.39	0.09	0.15	134
355	0.64	0.35	0.45	120
356	0.54	0.29	0.38	228
357	0.66	0.36	0.47	269
358	0.62	0.38	0.47	80
359	0.84	0.59	0.69	140
360	0.39	0.18	0.24	125
361	0.90	0.71	0.79	169
362	0.14	0.05	0.08	56
363	0.92	0.73	0.82	154
364	0.46	0.10	0.17	58
365	0.22	0.08	0.12	71
366	1.00	0.69	0.81	54
367	0.30	0.07	0.11	116
368	0.38	0.06	0.10	54
369	0.33	0.03	0.05	71
370	0.00	0.00	0.00	61
371	0.40	0.08	0.14	71
372	0.72	0.44	0.55	52
373	0.78	0.41	0.54	150
374	0.41	0.14	0.21	93
375	0.20	0.04	0.07	67
376	0.00	0.00	0.00	76
377	0.58	0.28	0.38	106
378	0.25	0.02	0.04	86
379	0.50	0.14	0.22	14

380	0.93	0.52	0.67	122
381	0.23	0.07	0.10	104
382	0.46	0.20	0.28	66
383	0.54	0.35	0.42	110
384	0.14	0.01	0.01	155
385	0.69	0.22	0.33	50
386	0.20	0.06	0.10	64
387	0.32	0.08	0.12	93
388	0.53	0.24	0.33	102
389	0.07	0.01	0.02	108
390	0.96	0.68	0.80	178
391	0.49	0.17	0.26	115
392	0.81	0.40	0.54	42
393	0.00	0.00	0.00	134
394	0.22	0.04	0.06	112
395	0.54	0.27	0.36	176
396	0.47	0.13	0.20	125
397	0.74	0.37	0.49	224
398	0.84	0.67	0.74	63
399	0.30	0.05	0.09	59
400	0.51	0.32	0.39	63
401	0.49	0.23	0.32	98
402	0.51	0.19	0.27	162
403	0.38	0.14	0.21	83
404	0.76	0.84	0.80	19
405	0.34	0.11	0.17	92
406	0.69	0.22	0.33	41
407	0.64	0.37	0.47	43
408	0.80	0.46	0.58	160
409	0.20	0.12	0.15	50
410	0.00	0.00	0.00	19
411	0.35	0.11	0.17	175
412	0.28	0.07	0.11	72
413	0.38	0.05	0.09	95
414	0.12	0.02	0.04	97
415	0.33	0.10	0.16	48
416	0.53	0.35	0.42	83
417	0.43	0.07	0.13	40
418	0.48	0.16	0.25	91
419	0.53	0.37	0.43	90
420	0.38	0.27	0.32	37
421	0.04	0.02	0.02	66
422	0.69	0.45	0.55	73
423	0.48	0.25	0.33	56
424	0.94	0.88	0.91	33
425	0.00	0.00	0.00	76
426	0.27	0.05	0.08	81
427	0.98	0.73	0.84	150
428	0.95	0.69	0.80	29
429	0.99	0.93	0.96	389
430	0.63	0.40	0.49	167
431	0.57	0.11	0.18	123
432	0.52	0.31	0.39	39
433	0.33	0.21	0.25	82
434	1.00	0.70	0.82	66
435	0.55	0.38	0.45	93
436	0.56	0.37	0.44	87
437	0.10	0.02	0.04	86
438	0.72	0.53	0.61	104
439	0.54	0.13	0.21	100
440	0.38	0.04	0.06	141
441	0.43	0.33	0.37	110
442	0.37	0.15	0.22	123
443	0.57	0.18	0.28	71
444	0.32	0.06	0.11	109
445	0.45	0.31	0.37	48
446	0.47	0.29	0.36	76
447	0.39	0.18	0.25	38
448	0.67	0.54	0.60	81
449	0.67	0.26	0.37	132
450	0.42	0.27	0.33	81
451	0.89	0.32	0.47	76
452	0.00	0.00	0.00	44
453	0.00	0.00	0.00	44
454	0.84	0.51	0.64	70
455	0.39	0.18	0.25	155
456	0.50	0.21	0.30	43
457	0.54	0.28	0.37	72
458	0.35	0.13	0.19	62
459	0.63	0.25	0.35	69
460	0.00	0.00	0.00	119
461	0.71	0.19	0.30	79
462	0.61	0.23	0.34	47

```
        463       0.39      0.14      0.21       104
        464       0.70      0.42      0.52       106
        465       0.64      0.22      0.33        64
        466       0.55      0.35      0.43       173
        467       0.78      0.42      0.55       107
        468       0.56      0.26      0.36       126
        469       0.20      0.01      0.02       114
        470       0.93      0.81      0.87       140
        471       0.85      0.42      0.56        79
        472       0.40      0.35      0.37       143
        473       0.67      0.37      0.47       158
        474       0.48      0.10      0.17       138
        475       0.00      0.00      0.00        59
        476       0.63      0.33      0.43        88
        477       0.83      0.65      0.73       176
        478       0.95      0.79      0.86        24
        479       0.22      0.04      0.07        92
        480       0.79      0.50      0.61       100
        481       0.51      0.28      0.36       103
        482       0.40      0.22      0.28        74
        483       0.78      0.63      0.69       105
        484       0.20      0.02      0.04        83
        485       0.20      0.02      0.04        82
        486       0.48      0.15      0.23        71
        487       0.45      0.21      0.29       120
        488       0.50      0.06      0.10       105
        489       0.73      0.37      0.49        87
        490       1.00      0.81      0.90        32
        491       0.33      0.03      0.05        69
        492       0.33      0.02      0.04        49
        493       0.11      0.02      0.03       117
        494       0.52      0.23      0.32        61
        495       0.95      0.79      0.87       344
        496       0.32      0.13      0.19        52
        497       0.59      0.28      0.38       137
        498       0.31      0.10      0.15        98
        499       0.48      0.20      0.29        79

avg / total       0.67      0.37      0.46    173812

Time taken to run this cell : 1:09:41.236859
```

In [ ]:

```
joblib.dump(classifier_2, 'lr_with_more_title_weight2.pkl')
```

5. Assignments

1. Use bag of words upto 4 grams and compute the micro f1 score with Logistic regression(OvR)
2. Perform hyperparameter tuning on alpha (or lambda) for Logistic regression to improve the performance using GridSearch
3. Try OneVsRestClassifier with Linear-SVM (SGDClassifier with loss-hinge)

Loading Preprocessed data with 3 times more title weight

In [4]:

```
#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:
        sampled_data = pd.read_sql_query("""SELECT question, Tags FROM QuestionsProcessed""", conn_r)
conn_r.commit()
conn_r.close()

#Display 10 questions.
sampled_data.head(10)
```

Out[4]:

	question	tags
0	dynam datagrid bind silverlight dynam datagrid...	c# silverlight data-binding
1	dynam datagrid bind silverlight dynam datagrid...	c# silverlight data-binding columns
2	java.lang.noclassdeffounderror javax servlet j...	jsp jstl
3	java.sql.sqlexcept microsoft odbc driver manag...	java jdbc
4	better way updat feed fb php sdk better way up...	facebook api facebook-php-sdk
5	btnadd click event open two window record ad b...	javascript asp.net web
6	sql inject issu prevent correct form submiss p...	php forms
7	countabl subaddit lebesgu measur countabl suba...	real-analysis measure-theory
8	hql equival sql queri hql equival sql queri hq...	hibernate hql
9	undefin symbol architectur i386 objc class skp...	iphone email-integration

In [5]:

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

```
Number of data points in sample : 500000
Number of dimensions : 2
```

Converting string Tags to multilable output variables __

In [6]:

```
vectorizer = CountVectorizer(tokenizer = lambda x: x.split(), binary='true')
multilabel_y = vectorizer.fit_transform(sampled_data['tags'])
```

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

In [7]:

```
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))
```

Selecting top 500 Tags __

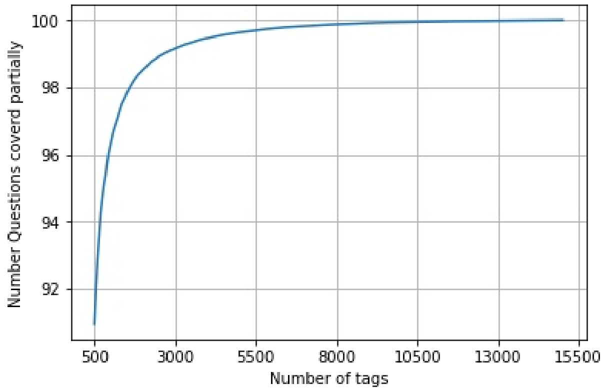
In [8]:

```
questions_explained = []
total_tags=multilabel_y.shape[1]
total_qs=sampled_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))
```

A variance plot showing the numbe rof partial covergae of questions with various tag numbers.

In [9]:

```
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 chose any number of tags based on your computing power, minimun is 500(it covers 90% of the tags)
print("With ",5500,"tags we are covering ",questions_explained[50],"% of questions")
print("With ",500,"tags we are covering ",questions_explained[0],"% of questions")
```




```
With  5500 tags we are covering  99.157 % of questions
With  500 tags we are covering  90.956 % of questions
```

In [10]:

```
# 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:  45221 out of  500000
```

In [11]:

```
train_datasize = 400000

x_train=sampled_data.head(train_datasize)
x_test=sampled_data.tail(sampled_data.shape[0] - train_datasize)

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

In [12]:

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

del(multilabel_yx, multilabel_y)
```

```
Number of data points in train data:  (400000, 500)
Number of data points in test data:  (100000, 500)
```

5.1 Featurizing the questions with BOW vectorizer - 1,2,3,4 - Grams

In [14]:

```
start = datetime.now()
vectorizer = CountVectorizer(min_df=0.00009, max_features=100000, analyzer='word', tokenizer = lambda x: x.split(
), ngram_range=(1,4))
x_train_multilabel = vectorizer.fit_transform(x_train['question'])
x_test_multilabel = vectorizer.transform(x_test['question'])
print("Time taken to featurize the class labels using BOW representation :", datetime.now() - start)

#Sorting indices to get rid of Value Error: WRITEBACKIFCOPY base is read-only
x_train_multilabel.sort_indices()
x_test_multilabel.sort_indices()

print("Dimensions of train and test data:")
print("x_train:",x_train_multilabel.shape, "y_train :",y_train.shape)
print("x_test:",x_test_multilabel.shape, "y_test :",y_test.shape)

#Save the data for later use.
import pickle
with open('x_train_multilabel.pkl', 'wb') as file:
    pickle.dump(x_train_multilabel, file)

with open('y_train.pkl', 'wb') as file:
    pickle.dump(y_train, file)

with open('x_test_multilabel.pkl', 'wb') as file:
    pickle.dump(x_test_multilabel, file)

with open('y_test.pkl', 'wb') as file:
    pickle.dump(y_test, file)
```

```
Time taken to featurize the class labels using BOW representation : 0:10:20.406049
Dimensions of train and test data:
x_train: (400000, 95585) y_train : (400000, 500)
x_test: (100000, 95585) y_test : (100000, 500)
```

In [4]:

```
import pickle

with open('x_train_multilabel.pkl', 'rb') as file:
    x_train_multilabel = pickle.load(file)

with open('y_train.pkl', 'rb') as file:
    y_train = pickle.load(file)

with open('x_test_multilabel.pkl', 'rb') as file:
    x_test_multilabel = pickle.load(file)

with open('y_test.pkl', 'rb') as file:
    y_test = pickle.load(file)

print("Dimensions of train and test data:")
print("x_train:",x_train_multilabel.shape, "y_train :",y_train.shape)
print("x_test:",x_test_multilabel.shape, "y_test :",y_test.shape)
```

```
Dimensions of train and test data:
x_train: (400000, 95585) y_train : (400000, 500)
x_test: (100000, 95585) y_test : (100000, 500)
```

5.2 Applying Logistic Regression with OneVsRest Classifier

In [6]:

```
import warnings
warnings.filterwarnings("ignore")

start = datetime.now()
classifier = OneVsRestClassifier(LogisticRegression(penalty='l1', C=1.0, random_state=0), 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')

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

print (metrics.classification_report(y_test, predictions))
print("Time taken to train the model :", datetime.now() - start)

import joblib
joblib.dump(classifier, 'lr_with_more_title_weight_lr_ovr.pkl')
```

```
Accuracy : 0.21224
Hamming loss  0.00313274
Micro-average quality numbers
Precision: 0.5686, Recall: 0.4097, Micro F1-measure: 0.4762
Macro-average quality numbers
Precision: 0.4511, Recall: 0.3346, Macro F1-measure: 0.3807
              precision    recall  f1-score   support

           0       0.90      0.73      0.81      5519
           1       0.52      0.41      0.46      8190
           2       0.64      0.47      0.54      6529
           3       0.68      0.53      0.59      3231
           4       0.66      0.49      0.56      6430
           5       0.62      0.42      0.50      2879
           6       0.74      0.57      0.64      5086
           7       0.75      0.61      0.68      4533
           8       0.34      0.18      0.24      3000
           9       0.70      0.59      0.64      2765
          10       0.42      0.29      0.35      3051
          11       0.59      0.45      0.51      3009
          12       0.48      0.36      0.41      2630
          13       0.54      0.38      0.44      1426
          14       0.80      0.61      0.69      2548
          15       0.48      0.30      0.37      2371
          16       0.54      0.29      0.38       873
          17       0.79      0.65      0.71      2151
          18       0.44      0.30      0.35      2204
          19       0.55      0.44      0.49       831
          20       0.70      0.49      0.57      1860
          21       0.26      0.18      0.21      2023
          22       0.40      0.30      0.34      1513
          23       0.76      0.58      0.66      1207
          24       0.45      0.34      0.39       506
          25       0.52      0.37      0.43       425
          26       0.57      0.44      0.50       793
          27       0.54      0.41      0.46      1291
          28       0.60      0.42      0.49      1208
          29       0.28      0.16      0.20       406
          30       0.46      0.24      0.31       504
          31       0.20      0.14      0.17       732
          32       0.47      0.32      0.38       441
          33       0.53      0.37      0.43      1645
          34       0.48      0.29      0.36      1058
          35       0.73      0.57      0.64       946
          36       0.48      0.29      0.36       644
          37       0.90      0.70      0.79       136
          38       0.51      0.38      0.43       570
          39       0.63      0.34      0.45       766
          40       0.53      0.43      0.47      1132
          41       0.35      0.31      0.33       174
```

42	0.66	0.54	0.60	210
43	0.64	0.45	0.53	433
44	0.59	0.48	0.53	626
45	0.56	0.38	0.46	852
46	0.63	0.48	0.55	534
47	0.30	0.24	0.26	350
48	0.63	0.54	0.58	496
49	0.76	0.64	0.69	785
50	0.19	0.12	0.14	475
51	0.26	0.21	0.23	305
52	0.24	0.11	0.15	251
53	0.54	0.41	0.47	914
54	0.39	0.25	0.30	728
55	0.16	0.08	0.11	258
56	0.35	0.29	0.32	821
57	0.35	0.19	0.25	541
58	0.60	0.35	0.44	748
59	0.89	0.70	0.79	724
60	0.35	0.19	0.24	660
61	0.43	0.24	0.31	235
62	0.88	0.72	0.79	718
63	0.78	0.69	0.73	468
64	0.44	0.31	0.36	191
65	0.29	0.18	0.22	429
66	0.22	0.12	0.16	415
67	0.67	0.55	0.61	274
68	0.72	0.54	0.62	510
69	0.60	0.50	0.54	466
70	0.25	0.16	0.20	305
71	0.34	0.22	0.27	247
72	0.71	0.53	0.61	401
73	0.85	0.79	0.82	86
74	0.57	0.43	0.49	120
75	0.82	0.71	0.76	129
76	0.12	0.05	0.07	473
77	0.37	0.30	0.33	143
78	0.67	0.47	0.55	347
79	0.49	0.27	0.35	479
80	0.43	0.37	0.40	279
81	0.51	0.28	0.36	461
82	0.14	0.06	0.09	298
83	0.72	0.52	0.60	396
84	0.39	0.38	0.38	184
85	0.43	0.30	0.35	573
86	0.27	0.12	0.17	325
87	0.50	0.41	0.45	273
88	0.46	0.31	0.37	135
89	0.25	0.17	0.20	232
90	0.49	0.41	0.45	409
91	0.51	0.33	0.40	420
92	0.68	0.57	0.62	408
93	0.55	0.49	0.52	241
94	0.20	0.09	0.13	211
95	0.31	0.17	0.22	277
96	0.21	0.12	0.15	410
97	0.76	0.47	0.58	501
98	0.67	0.62	0.64	136
99	0.45	0.37	0.41	239
100	0.33	0.20	0.25	324
101	0.85	0.73	0.79	277
102	0.89	0.76	0.82	613
103	0.37	0.23	0.28	157
104	0.21	0.12	0.15	295
105	0.65	0.46	0.54	334
106	0.66	0.36	0.47	335
107	0.69	0.58	0.63	389
108	0.51	0.33	0.40	251
109	0.56	0.47	0.51	317
110	0.30	0.11	0.16	187
111	0.45	0.19	0.26	140
112	0.57	0.47	0.52	154
113	0.49	0.28	0.36	332
114	0.43	0.28	0.34	323
115	0.44	0.33	0.38	344
116	0.67	0.54	0.60	370
117	0.44	0.30	0.36	313
118	0.76	0.75	0.75	874
119	0.36	0.26	0.30	293
120	0.13	0.09	0.10	200
121	0.66	0.50	0.57	463
122	0.23	0.12	0.16	119
123	0.15	0.04	0.06	256
124	0.86	0.72	0.78	195

125	0.29	0.17	0.22	138
126	0.72	0.53	0.61	376
127	0.15	0.07	0.09	122
128	0.14	0.06	0.08	252
129	0.43	0.38	0.40	144
130	0.31	0.18	0.23	150
131	0.19	0.09	0.12	210
132	0.51	0.33	0.40	361
133	0.84	0.64	0.73	453
134	0.81	0.77	0.79	124
135	0.15	0.12	0.13	91
136	0.51	0.38	0.43	128
137	0.45	0.40	0.42	218
138	0.34	0.21	0.26	243
139	0.30	0.19	0.23	149
140	0.69	0.54	0.61	318
141	0.19	0.12	0.15	159
142	0.57	0.42	0.49	274
143	0.81	0.83	0.82	362
144	0.43	0.28	0.34	118
145	0.50	0.43	0.46	164
146	0.51	0.39	0.44	461
147	0.69	0.43	0.53	159
148	0.32	0.20	0.25	166
149	0.90	0.59	0.72	346
150	0.49	0.22	0.30	350
151	0.90	0.67	0.77	55
152	0.71	0.52	0.60	387
153	0.37	0.33	0.35	150
154	0.32	0.15	0.20	281
155	0.25	0.19	0.22	202
156	0.74	0.66	0.70	130
157	0.21	0.10	0.14	245
158	0.90	0.69	0.79	177
159	0.46	0.40	0.43	130
160	0.38	0.24	0.29	336
161	0.79	0.65	0.71	220
162	0.20	0.11	0.14	229
163	0.78	0.46	0.58	316
164	0.63	0.42	0.50	283
165	0.53	0.37	0.44	197
166	0.54	0.53	0.54	101
167	0.37	0.23	0.28	231
168	0.43	0.35	0.38	370
169	0.39	0.24	0.29	258
170	0.27	0.17	0.21	101
171	0.34	0.28	0.31	89
172	0.48	0.38	0.42	193
173	0.46	0.33	0.39	309
174	0.30	0.14	0.19	172
175	0.79	0.74	0.76	95
176	0.85	0.63	0.73	346
177	0.81	0.60	0.69	322
178	0.53	0.47	0.50	232
179	0.21	0.10	0.14	125
180	0.46	0.40	0.43	145
181	0.30	0.21	0.25	77
182	0.18	0.12	0.14	182
183	0.52	0.37	0.43	257
184	0.23	0.13	0.17	216
185	0.31	0.20	0.24	242
186	0.34	0.22	0.27	165
187	0.68	0.57	0.62	263
188	0.19	0.10	0.13	174
189	0.64	0.48	0.55	136
190	0.82	0.59	0.69	202
191	0.29	0.21	0.24	134
192	0.58	0.46	0.51	230
193	0.26	0.19	0.22	90
194	0.55	0.55	0.55	185
195	0.16	0.08	0.10	156
196	0.14	0.09	0.11	160
197	0.28	0.17	0.21	266
198	0.28	0.15	0.20	284
199	0.19	0.08	0.11	145
200	0.86	0.77	0.82	212
201	0.48	0.26	0.33	317
202	0.69	0.63	0.66	427
203	0.21	0.15	0.17	232
204	0.38	0.29	0.33	217
205	0.49	0.49	0.49	527
206	0.15	0.06	0.09	124
207	0.40	0.34	0.37	103

208	0.77	0.55	0.64	287
209	0.20	0.11	0.14	193
210	0.54	0.39	0.45	220
211	0.44	0.20	0.28	140
212	0.15	0.09	0.11	161
213	0.45	0.53	0.49	72
214	0.60	0.42	0.50	396
215	0.67	0.42	0.51	134
216	0.46	0.26	0.34	400
217	0.32	0.25	0.28	75
218	0.93	0.77	0.84	219
219	0.58	0.42	0.49	210
220	0.84	0.67	0.75	298
221	0.89	0.70	0.79	266
222	0.66	0.45	0.53	290
223	0.12	0.05	0.07	128
224	0.69	0.48	0.57	159
225	0.40	0.35	0.37	164
226	0.47	0.34	0.39	144
227	0.54	0.40	0.46	276
228	0.08	0.04	0.06	235
229	0.14	0.06	0.09	216
230	0.32	0.21	0.25	228
231	0.63	0.53	0.58	64
232	0.23	0.16	0.18	103
233	0.61	0.39	0.48	216
234	0.50	0.23	0.32	116
235	0.45	0.32	0.38	77
236	0.88	0.69	0.77	67
237	0.29	0.18	0.22	218
238	0.26	0.18	0.21	139
239	0.22	0.06	0.10	94
240	0.39	0.31	0.35	77
241	0.31	0.13	0.19	167
242	0.63	0.42	0.50	86
243	0.31	0.24	0.27	58
244	0.52	0.40	0.45	269
245	0.12	0.08	0.10	112
246	0.92	0.82	0.86	255
247	0.22	0.22	0.22	58
248	0.14	0.07	0.10	81
249	0.05	0.02	0.03	131
250	0.40	0.26	0.31	93
251	0.57	0.34	0.43	154
252	0.10	0.05	0.06	129
253	0.47	0.34	0.39	83
254	0.24	0.13	0.17	191
255	0.11	0.06	0.08	219
256	0.13	0.08	0.10	130
257	0.38	0.31	0.34	93
258	0.63	0.53	0.57	217
259	0.27	0.18	0.21	141
260	0.65	0.24	0.35	143
261	0.40	0.18	0.25	219
262	0.48	0.36	0.41	107
263	0.37	0.24	0.29	236
264	0.27	0.21	0.23	119
265	0.43	0.28	0.34	72
266	0.10	0.06	0.07	70
267	0.35	0.24	0.29	107
268	0.53	0.47	0.50	169
269	0.29	0.17	0.22	129
270	0.69	0.54	0.61	159
271	0.77	0.53	0.63	190
272	0.46	0.34	0.39	248
273	0.84	0.75	0.79	264
274	0.84	0.67	0.74	105
275	0.20	0.12	0.15	104
276	0.07	0.03	0.05	115
277	0.77	0.61	0.68	170
278	0.71	0.48	0.57	145
279	0.88	0.75	0.81	230
280	0.58	0.40	0.47	80
281	0.65	0.55	0.59	217
282	0.69	0.52	0.59	175
283	0.26	0.17	0.21	269
284	0.53	0.39	0.45	74
285	0.69	0.51	0.59	206
286	0.83	0.71	0.76	227
287	0.65	0.42	0.51	130
288	0.16	0.08	0.11	129
289	0.16	0.11	0.13	80
290	0.19	0.14	0.16	99

291	0.59	0.39	0.47	208
292	0.28	0.15	0.19	67
293	0.76	0.54	0.63	109
294	0.34	0.26	0.30	140
295	0.24	0.17	0.20	241
296	0.23	0.15	0.18	72
297	0.22	0.15	0.18	107
298	0.62	0.57	0.60	61
299	0.73	0.56	0.63	77
300	0.15	0.12	0.13	111
301	0.03	0.01	0.01	126
302	0.16	0.11	0.13	73
303	0.55	0.44	0.49	176
304	0.89	0.81	0.85	230
305	0.82	0.71	0.76	156
306	0.43	0.37	0.40	146
307	0.22	0.11	0.15	98
308	0.04	0.01	0.02	78
309	0.46	0.17	0.25	94
310	0.59	0.38	0.46	162
311	0.71	0.50	0.59	116
312	0.45	0.35	0.40	57
313	0.35	0.11	0.16	65
314	0.41	0.36	0.38	138
315	0.50	0.29	0.36	195
316	0.39	0.32	0.35	69
317	0.26	0.21	0.23	134
318	0.54	0.41	0.47	148
319	0.81	0.56	0.66	161
320	0.18	0.18	0.18	104
321	0.69	0.62	0.65	156
322	0.55	0.46	0.50	134
323	0.53	0.45	0.49	232
324	0.21	0.16	0.18	92
325	0.37	0.23	0.29	197
326	0.10	0.07	0.08	126
327	0.17	0.08	0.11	115
328	0.94	0.71	0.81	198
329	0.43	0.30	0.35	125
330	0.53	0.26	0.35	81
331	0.33	0.15	0.21	94
332	0.29	0.20	0.23	56
333	0.15	0.08	0.10	260
334	0.16	0.12	0.13	60
335	0.25	0.12	0.16	110
336	0.59	0.54	0.56	71
337	0.12	0.09	0.10	66
338	0.39	0.45	0.42	150
339	0.05	0.04	0.04	54
340	0.79	0.57	0.66	195
341	0.68	0.51	0.58	79
342	0.37	0.50	0.43	38
343	0.55	0.40	0.46	43
344	0.33	0.28	0.30	68
345	0.64	0.34	0.45	73
346	0.10	0.07	0.08	116
347	0.61	0.49	0.54	111
348	0.24	0.19	0.21	63
349	0.83	0.71	0.77	104
350	0.57	0.57	0.57	44
351	0.25	0.28	0.26	40
352	0.76	0.62	0.68	136
353	0.40	0.22	0.29	54
354	0.24	0.12	0.16	134
355	0.51	0.42	0.46	120
356	0.43	0.31	0.36	228
357	0.55	0.42	0.48	269
358	0.56	0.36	0.44	80
359	0.75	0.63	0.68	140
360	0.30	0.19	0.23	125
361	0.87	0.73	0.80	169
362	0.17	0.12	0.14	56
363	0.83	0.77	0.80	154
364	0.19	0.19	0.19	58
365	0.22	0.15	0.18	71
366	0.90	0.67	0.77	54
367	0.14	0.10	0.12	116
368	0.26	0.19	0.22	54
369	0.09	0.06	0.07	71
370	0.23	0.11	0.15	61
371	0.29	0.10	0.15	71
372	0.52	0.44	0.48	52
373	0.59	0.47	0.52	150

374	0.26	0.22	0.23	93
375	0.12	0.09	0.10	67
376	0.07	0.03	0.04	76
377	0.45	0.37	0.41	106
378	0.07	0.02	0.04	86
379	0.23	0.21	0.22	14
380	0.76	0.56	0.64	122
381	0.10	0.06	0.07	104
382	0.25	0.15	0.19	66
383	0.49	0.41	0.45	110
384	0.16	0.07	0.10	155
385	0.40	0.36	0.38	50
386	0.21	0.12	0.16	64
387	0.26	0.12	0.16	93
388	0.47	0.33	0.39	102
389	0.13	0.06	0.08	108
390	0.90	0.68	0.78	178
391	0.36	0.21	0.26	115
392	0.66	0.45	0.54	42
393	0.04	0.01	0.01	134
394	0.29	0.16	0.21	112
395	0.37	0.30	0.33	176
396	0.30	0.17	0.22	125
397	0.64	0.45	0.52	224
398	0.75	0.67	0.71	63
399	0.11	0.07	0.09	59
400	0.44	0.41	0.43	63
401	0.42	0.32	0.36	98
402	0.45	0.24	0.31	162
403	0.24	0.18	0.21	83
404	0.64	0.84	0.73	19
405	0.21	0.15	0.18	92
406	0.47	0.39	0.43	41
407	0.50	0.33	0.39	43
408	0.68	0.49	0.57	160
409	0.12	0.08	0.10	50
410	0.00	0.00	0.00	19
411	0.30	0.22	0.25	175
412	0.22	0.15	0.18	72
413	0.22	0.09	0.13	95
414	0.24	0.15	0.19	97
415	0.14	0.10	0.12	48
416	0.43	0.34	0.38	83
417	0.23	0.15	0.18	40
418	0.24	0.13	0.17	91
419	0.52	0.42	0.47	90
420	0.28	0.24	0.26	37
421	0.07	0.05	0.06	66
422	0.46	0.40	0.43	73
423	0.37	0.29	0.32	56
424	0.85	0.88	0.87	33
425	0.10	0.04	0.06	76
426	0.06	0.02	0.04	81
427	0.92	0.73	0.81	150
428	1.00	0.76	0.86	29
429	0.98	0.95	0.97	389
430	0.56	0.44	0.49	167
431	0.45	0.15	0.22	123
432	0.26	0.18	0.21	39
433	0.31	0.28	0.29	82
434	0.90	0.71	0.80	66
435	0.56	0.47	0.51	93
436	0.49	0.38	0.43	87
437	0.16	0.09	0.12	86
438	0.64	0.49	0.55	104
439	0.46	0.21	0.29	100
440	0.17	0.07	0.10	141
441	0.40	0.43	0.41	110
442	0.24	0.20	0.22	123
443	0.29	0.21	0.25	71
444	0.22	0.12	0.15	109
445	0.42	0.35	0.39	48
446	0.41	0.28	0.33	76
447	0.23	0.26	0.24	38
448	0.60	0.56	0.58	81
449	0.44	0.28	0.34	132
450	0.41	0.33	0.37	81
451	0.67	0.38	0.49	76
452	0.11	0.07	0.08	44
453	0.00	0.00	0.00	44
454	0.75	0.54	0.63	70
455	0.29	0.25	0.27	155
456	0.31	0.26	0.28	43

```
         457       0.37      0.31      0.33        72
         458       0.20      0.18      0.19        62
         459       0.42      0.32      0.36        69
         460       0.14      0.08      0.11       119
         461       0.60      0.34      0.44        79
         462       0.30      0.26      0.28        47
         463       0.34      0.26      0.29       104
         464       0.56      0.42      0.48       106
         465       0.35      0.28      0.31        64
         466       0.44      0.33      0.38       173
         467       0.60      0.44      0.51       107
         468       0.42      0.29      0.35       126
         469       0.17      0.06      0.09       114
         470       0.93      0.81      0.87       140
         471       0.58      0.38      0.46        79
         472       0.41      0.41      0.41       143
         473       0.64      0.39      0.49       158
         474       0.28      0.12      0.16       138
         475       0.20      0.15      0.17        59
         476       0.63      0.45      0.53        88
         477       0.73      0.65      0.69       176
         478       0.90      0.79      0.84        24
         479       0.28      0.17      0.21        92
         480       0.68      0.58      0.63       100
         481       0.37      0.36      0.36       103
         482       0.26      0.15      0.19        74
         483       0.71      0.59      0.65       105
         484       0.18      0.07      0.10        83
         485       0.05      0.04      0.04        82
         486       0.30      0.18      0.23        71
         487       0.38      0.23      0.28       120
         488       0.23      0.10      0.13       105
         489       0.54      0.39      0.45        87
         490       0.90      0.84      0.87        32
         491       0.05      0.03      0.04        69
         492       0.14      0.06      0.09        49
         493       0.06      0.04      0.05       117
         494       0.49      0.38      0.43        61
         495       0.95      0.80      0.87       344
         496       0.19      0.12      0.14        52
         497       0.49      0.34      0.40       137
         498       0.33      0.15      0.21        98
         499       0.31      0.23      0.26        79

   micro avg       0.57      0.41      0.48    173812
   macro avg       0.45      0.33      0.38    173812
weighted avg       0.55      0.41      0.47    173812
 samples avg       0.44      0.39      0.38    173812

Time taken to train the model : 4:23:30.948387
```

Out[6]:

```
['lr_with_more_title_weight_lr_ovr.pkl']
```

5.3.1 Applying Logistic Regression (SGDClassifier with 'log' loss) with OneVsRest Classifier

In [4]:

```
import warnings
warnings.filterwarnings("ignore")

start = datetime.now()
classifier = OneVsRestClassifier(SGDClassifier(loss='log', alpha=0.001, penalty='l1', random_state=0), 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')

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

print (metrics.classification_report(y_test, predictions))
print("Time taken to train the model :", datetime.now() - start)

import joblib
joblib.dump(classifier, 'lr_with_more_title_weight_sgd_logloss_ovr.pkl')
```

```
Accuracy : 0.1868
Hamming loss  0.00326424
Micro-average quality numbers
Precision: 0.5507, Recall: 0.3310, Micro F1-measure: 0.4135
Macro-average quality numbers
Precision: 0.4102, Recall: 0.2470, Macro F1-measure: 0.2879
              precision    recall  f1-score   support

           0       0.72      0.68      0.70      5519
           1       0.54      0.21      0.30      8190
           2       0.70      0.36      0.48      6529
           3       0.54      0.47      0.50      3231
           4       0.76      0.40      0.52      6430
           5       0.71      0.35      0.47      2879
           6       0.67      0.57      0.62      5086
           7       0.75      0.63      0.68      4533
           8       0.48      0.15      0.23      3000
           9       0.55      0.60      0.58      2765
          10       0.44      0.22      0.29      3051
          11       0.77      0.28      0.41      3009
          12       0.54      0.27      0.36      2630
          13       0.48      0.19      0.27      1426
          14       0.79      0.63      0.70      2548
          15       0.61      0.15      0.24      2371
          16       0.51      0.27      0.35       873
          17       0.75      0.70      0.72      2151
          18       0.65      0.23      0.34      2204
          19       0.68      0.38      0.49       831
          20       0.73      0.47      0.57      1860
          21       0.25      0.09      0.13      2023
          22       0.37      0.21      0.27      1513
          23       0.82      0.55      0.66      1207
          24       0.43      0.36      0.39       506
          25       0.73      0.32      0.44       425
          26       0.43      0.36      0.39       793
          27       0.56      0.33      0.42      1291
          28       0.75      0.33      0.46      1208
          29       0.24      0.13      0.17       406
          30       0.25      0.27      0.26       504
          31       0.11      0.22      0.15       732
          32       0.50      0.33      0.40       441
          33       0.37      0.12      0.18      1645
          34       0.60      0.31      0.41      1058
          35       0.59      0.68      0.63       946
          36       0.58      0.27      0.37       644
          37       0.92      0.78      0.84       136
          38       0.49      0.42      0.45       570
          39       0.79      0.36      0.50       766
          40       0.49      0.29      0.36      1132
          41       0.19      0.25      0.22       174
```

42	0.56	0.58	0.57	210
43	0.60	0.52	0.56	433
44	0.63	0.46	0.53	626
45	0.32	0.28	0.30	852
46	0.66	0.41	0.50	534
47	0.20	0.23	0.22	350
48	0.68	0.49	0.57	496
49	0.70	0.62	0.66	785
50	0.19	0.10	0.13	475
51	0.09	0.12	0.11	305
52	0.21	0.07	0.11	251
53	0.52	0.57	0.55	914
54	0.35	0.18	0.23	728
55	0.07	0.01	0.01	258
56	0.37	0.13	0.19	821
57	0.37	0.10	0.16	541
58	0.69	0.32	0.43	748
59	0.85	0.74	0.79	724
60	0.27	0.07	0.11	660
61	0.84	0.23	0.36	235
62	0.90	0.69	0.78	718
63	0.75	0.61	0.68	468
64	0.43	0.55	0.49	191
65	0.25	0.09	0.14	429
66	0.15	0.17	0.16	415
67	0.35	0.52	0.41	274
68	0.65	0.60	0.62	510
69	0.61	0.43	0.50	466
70	0.27	0.15	0.20	305
71	0.23	0.19	0.21	247
72	0.71	0.41	0.52	401
73	0.78	0.83	0.80	86
74	0.61	0.31	0.41	120
75	0.84	0.72	0.77	129
76	0.00	0.00	0.00	473
77	0.30	0.27	0.28	143
78	0.69	0.50	0.58	347
79	0.70	0.22	0.33	479
80	0.19	0.40	0.26	279
81	0.56	0.15	0.24	461
82	0.08	0.02	0.03	298
83	0.72	0.47	0.57	396
84	0.37	0.38	0.38	184
85	0.45	0.16	0.24	573
86	0.17	0.03	0.06	325
87	0.38	0.23	0.29	273
88	0.32	0.28	0.30	135
89	0.20	0.16	0.18	232
90	0.42	0.36	0.38	409
91	0.58	0.30	0.40	420
92	0.68	0.57	0.62	408
93	0.59	0.48	0.53	241
94	0.26	0.09	0.13	211
95	0.18	0.24	0.20	277
96	0.26	0.03	0.05	410
97	0.90	0.16	0.28	501
98	0.62	0.65	0.63	136
99	0.51	0.29	0.37	239
100	0.38	0.06	0.11	324
101	0.92	0.56	0.70	277
102	0.68	0.73	0.70	613
103	0.48	0.20	0.29	157
104	0.17	0.13	0.15	295
105	0.67	0.42	0.51	334
106	0.46	0.04	0.07	335
107	0.73	0.50	0.59	389
108	0.45	0.25	0.32	251
109	0.43	0.50	0.46	317
110	0.36	0.07	0.12	187
111	0.38	0.19	0.25	140
112	0.13	0.10	0.12	154
113	0.67	0.17	0.27	332
114	0.36	0.24	0.29	323
115	0.31	0.20	0.25	344
116	0.51	0.58	0.54	370
117	0.50	0.18	0.27	313
118	0.77	0.61	0.68	874
119	0.39	0.20	0.26	293
120	0.05	0.02	0.03	200
121	0.75	0.42	0.54	463
122	0.26	0.13	0.18	119
123	0.00	0.00	0.00	256
124	0.45	0.78	0.57	195

125	0.34	0.24	0.28	138
126	0.50	0.62	0.56	376
127	0.16	0.06	0.08	122
128	0.17	0.06	0.09	252
129	0.00	0.00	0.00	144
130	0.09	0.03	0.05	150
131	0.04	0.01	0.02	210
132	0.37	0.07	0.12	361
133	0.86	0.58	0.69	453
134	0.78	0.73	0.76	124
135	0.00	0.00	0.00	91
136	0.34	0.29	0.31	128
137	0.41	0.28	0.33	218
138	0.33	0.01	0.02	243
139	0.33	0.21	0.25	149
140	0.63	0.42	0.51	318
141	0.09	0.15	0.11	159
142	0.65	0.37	0.47	274
143	0.65	0.70	0.68	362
144	0.49	0.21	0.30	118
145	0.40	0.40	0.40	164
146	0.59	0.25	0.35	461
147	0.62	0.45	0.52	159
148	0.24	0.13	0.17	166
149	0.78	0.54	0.64	346
150	1.00	0.00	0.01	350
151	0.90	0.47	0.62	55
152	0.35	0.56	0.43	387
153	0.25	0.03	0.06	150
154	0.31	0.06	0.10	281
155	0.27	0.12	0.16	202
156	0.58	0.68	0.63	130
157	0.25	0.11	0.16	245
158	0.84	0.47	0.60	177
159	0.38	0.39	0.38	130
160	0.44	0.20	0.27	336
161	0.62	0.65	0.64	220
162	0.11	0.10	0.11	229
163	0.85	0.41	0.56	316
164	0.62	0.31	0.41	283
165	0.41	0.24	0.30	197
166	0.13	0.23	0.17	101
167	0.40	0.21	0.28	231
168	0.30	0.11	0.17	370
169	0.39	0.24	0.30	258
170	0.14	0.07	0.09	101
171	0.46	0.29	0.36	89
172	0.34	0.27	0.30	193
173	0.42	0.32	0.36	309
174	0.26	0.27	0.26	172
175	0.89	0.75	0.81	95
176	0.86	0.60	0.71	346
177	0.94	0.31	0.47	322
178	0.49	0.47	0.48	232
179	0.50	0.02	0.05	125
180	0.44	0.20	0.27	145
181	0.37	0.14	0.21	77
182	0.14	0.04	0.07	182
183	0.50	0.34	0.40	257
184	0.12	0.04	0.06	216
185	0.22	0.10	0.13	242
186	0.26	0.14	0.18	165
187	0.56	0.62	0.59	263
188	0.30	0.13	0.18	174
189	0.22	0.08	0.12	136
190	0.57	0.31	0.40	202
191	0.27	0.10	0.14	134
192	0.78	0.34	0.48	230
193	0.31	0.21	0.25	90
194	0.57	0.49	0.53	185
195	0.10	0.04	0.06	156
196	0.31	0.03	0.05	160
197	0.00	0.00	0.00	266
198	0.32	0.05	0.08	284
199	0.20	0.06	0.09	145
200	0.83	0.83	0.83	212
201	0.27	0.04	0.07	317
202	0.60	0.70	0.64	427
203	0.22	0.10	0.14	232
204	0.29	0.18	0.23	217
205	0.40	0.60	0.48	527
206	0.03	0.02	0.02	124
207	0.07	0.08	0.08	103

208	0.87	0.41	0.56	287
209	0.17	0.07	0.10	193
210	0.47	0.22	0.30	220
211	0.70	0.05	0.09	140
212	0.07	0.06	0.06	161
213	0.09	0.31	0.14	72
214	0.60	0.41	0.48	396
215	0.86	0.27	0.41	134
216	0.00	0.00	0.00	400
217	0.48	0.31	0.37	75
218	0.88	0.79	0.83	219
219	0.77	0.31	0.45	210
220	0.92	0.32	0.48	298
221	0.96	0.54	0.69	266
222	0.83	0.24	0.37	290
223	0.26	0.05	0.08	128
224	0.77	0.33	0.46	159
225	0.41	0.23	0.29	164
226	0.52	0.36	0.43	144
227	0.43	0.38	0.41	276
228	0.08	0.01	0.02	235
229	0.00	0.00	0.00	216
230	0.31	0.19	0.23	228
231	0.64	0.50	0.56	64
232	0.06	0.03	0.04	103
233	0.67	0.39	0.50	216
234	0.00	0.00	0.00	116
235	0.59	0.52	0.55	77
236	0.90	0.70	0.79	67
237	0.00	0.00	0.00	218
238	0.05	0.02	0.03	139
239	0.25	0.02	0.04	94
240	0.37	0.13	0.19	77
241	0.43	0.02	0.03	167
242	0.82	0.27	0.40	86
243	0.58	0.12	0.20	58
244	0.22	0.07	0.11	269
245	0.13	0.06	0.08	112
246	0.95	0.78	0.86	255
247	0.39	0.22	0.29	58
248	0.08	0.05	0.06	81
249	0.00	0.00	0.00	131
250	0.14	0.05	0.08	93
251	0.25	0.30	0.27	154
252	0.17	0.03	0.05	129
253	0.35	0.40	0.38	83
254	0.24	0.09	0.13	191
255	0.11	0.02	0.03	219
256	0.09	0.02	0.03	130
257	0.40	0.39	0.39	93
258	0.71	0.37	0.48	217
259	0.07	0.07	0.07	141
260	0.86	0.13	0.23	143
261	0.49	0.09	0.15	219
262	0.39	0.22	0.29	107
263	0.33	0.30	0.31	236
264	0.15	0.26	0.19	119
265	0.20	0.25	0.22	72
266	0.25	0.11	0.16	70
267	0.21	0.07	0.10	107
268	0.66	0.35	0.46	169
269	0.19	0.11	0.14	129
270	0.69	0.55	0.61	159
271	0.38	0.11	0.17	190
272	0.45	0.10	0.16	248
273	0.88	0.64	0.74	264
274	0.60	0.74	0.67	105
275	0.00	0.00	0.00	104
276	0.06	0.04	0.05	115
277	0.83	0.51	0.63	170
278	0.48	0.14	0.22	145
279	0.92	0.40	0.56	230
280	0.54	0.41	0.47	80
281	0.67	0.54	0.60	217
282	0.75	0.43	0.55	175
283	0.42	0.04	0.07	269
284	0.64	0.34	0.44	74
285	0.88	0.43	0.58	206
286	0.89	0.56	0.69	227
287	0.61	0.39	0.48	130
288	0.27	0.10	0.15	129
289	0.20	0.01	0.02	80
290	0.14	0.16	0.15	99

291	0.59	0.50	0.54	208
292	0.31	0.15	0.20	67
293	0.52	0.22	0.31	109
294	0.25	0.27	0.26	140
295	0.16	0.16	0.16	241
296	0.16	0.12	0.14	72
297	0.15	0.14	0.14	107
298	0.79	0.18	0.29	61
299	0.17	0.36	0.23	77
300	0.13	0.08	0.10	111
301	0.00	0.00	0.00	126
302	0.00	0.00	0.00	73
303	0.52	0.43	0.47	176
304	0.95	0.63	0.76	230
305	0.95	0.61	0.74	156
306	0.40	0.43	0.42	146
307	0.17	0.05	0.08	98
308	0.25	0.01	0.02	78
309	0.27	0.13	0.17	94
310	0.62	0.28	0.38	162
311	0.68	0.44	0.53	116
312	0.56	0.33	0.42	57
313	0.00	0.00	0.00	65
314	0.45	0.33	0.38	138
315	0.41	0.21	0.27	195
316	0.46	0.39	0.42	69
317	0.12	0.09	0.10	134
318	0.33	0.26	0.29	148
319	0.55	0.36	0.43	161
320	0.16	0.25	0.20	104
321	0.50	0.75	0.60	156
322	0.48	0.25	0.33	134
323	0.51	0.30	0.38	232
324	0.13	0.21	0.16	92
325	0.25	0.09	0.13	197
326	0.07	0.05	0.06	126
327	0.33	0.01	0.02	115
328	0.88	0.55	0.67	198
329	0.51	0.28	0.36	125
330	0.62	0.06	0.11	81
331	0.32	0.13	0.18	94
332	0.00	0.00	0.00	56
333	0.03	0.00	0.01	260
334	0.00	0.00	0.00	60
335	0.15	0.12	0.13	110
336	0.53	0.45	0.49	71
337	0.15	0.18	0.16	66
338	0.46	0.37	0.41	150
339	0.00	0.00	0.00	54
340	0.84	0.47	0.61	195
341	0.16	0.11	0.13	79
342	0.31	0.39	0.35	38
343	0.11	0.30	0.16	43
344	0.18	0.13	0.15	68
345	0.50	0.37	0.43	73
346	0.11	0.14	0.12	116
347	0.89	0.29	0.44	111
348	0.12	0.03	0.05	63
349	0.80	0.58	0.67	104
350	0.67	0.32	0.43	44
351	0.04	0.05	0.04	40
352	0.93	0.29	0.45	136
353	0.37	0.31	0.34	54
354	0.00	0.00	0.00	134
355	0.26	0.11	0.15	120
356	0.24	0.07	0.10	228
357	0.55	0.11	0.18	269
358	0.50	0.26	0.34	80
359	0.79	0.24	0.37	140
360	0.30	0.14	0.19	125
361	0.88	0.31	0.46	169
362	0.09	0.05	0.07	56
363	0.84	0.73	0.78	154
364	0.05	0.07	0.06	58
365	0.18	0.27	0.21	71
366	0.90	0.70	0.79	54
367	0.18	0.16	0.17	116
368	0.00	0.00	0.00	54
369	0.00	0.00	0.00	71
370	0.00	0.00	0.00	61
371	0.40	0.08	0.14	71
372	0.59	0.42	0.49	52
373	0.75	0.20	0.32	150

374	0.40	0.18	0.25	93
375	0.33	0.01	0.03	67
376	0.00	0.00	0.00	76
377	0.91	0.09	0.17	106
378	0.00	0.00	0.00	86
379	0.11	0.07	0.09	14
380	0.97	0.28	0.43	122
381	0.13	0.06	0.08	104
382	0.16	0.14	0.15	66
383	0.43	0.26	0.33	110
384	0.00	0.00	0.00	155
385	0.33	0.04	0.07	50
386	0.17	0.14	0.15	64
387	0.00	0.00	0.00	93
388	0.50	0.19	0.27	102
389	0.00	0.00	0.00	108
390	0.97	0.40	0.57	178
391	0.52	0.12	0.20	115
392	0.93	0.31	0.46	42
393	0.00	0.00	0.00	134
394	0.00	0.00	0.00	112
395	0.33	0.02	0.03	176
396	0.00	0.00	0.00	125
397	0.65	0.12	0.21	224
398	0.83	0.30	0.44	63
399	0.00	0.00	0.00	59
400	0.26	0.44	0.33	63
401	0.11	0.02	0.03	98
402	0.41	0.06	0.10	162
403	0.37	0.16	0.22	83
404	0.74	0.74	0.74	19
405	0.17	0.23	0.20	92
406	0.33	0.15	0.20	41
407	0.69	0.21	0.32	43
408	0.00	0.00	0.00	160
409	0.25	0.22	0.23	50
410	0.00	0.00	0.00	19
411	0.34	0.13	0.18	175
412	0.08	0.01	0.02	72
413	1.00	0.02	0.04	95
414	0.13	0.08	0.10	97
415	0.28	0.10	0.15	48
416	0.40	0.25	0.31	83
417	0.02	0.03	0.02	40
418	0.19	0.13	0.16	91
419	0.33	0.32	0.32	90
420	0.20	0.16	0.18	37
421	0.10	0.03	0.05	66
422	0.52	0.37	0.43	73
423	0.34	0.18	0.24	56
424	0.90	0.82	0.86	33
425	0.00	0.00	0.00	76
426	0.33	0.02	0.05	81
427	0.89	0.75	0.81	150
428	0.80	0.69	0.74	29
429	0.29	0.02	0.04	389
430	0.57	0.24	0.34	167
431	0.00	0.00	0.00	123
432	0.37	0.36	0.36	39
433	0.34	0.24	0.28	82
434	1.00	0.55	0.71	66
435	0.39	0.63	0.48	93
436	0.00	0.00	0.00	87
437	0.50	0.08	0.14	86
438	0.58	0.37	0.45	104
439	0.67	0.06	0.11	100
440	0.33	0.01	0.01	141
441	0.29	0.25	0.27	110
442	0.19	0.10	0.13	123
443	0.00	0.00	0.00	71
444	0.31	0.14	0.19	109
445	0.22	0.15	0.18	48
446	0.42	0.26	0.32	76
447	0.00	0.00	0.00	38
448	0.66	0.51	0.57	81
449	0.38	0.06	0.10	132
450	0.45	0.25	0.32	81
451	0.56	0.39	0.46	76
452	0.00	0.00	0.00	44
453	0.00	0.00	0.00	44
454	0.70	0.57	0.63	70
455	0.00	0.00	0.00	155
456	0.20	0.16	0.18	43

```
         457       0.42      0.15      0.22        72
         458       0.19      0.10      0.13        62
         459       0.00      0.00      0.00        69
         460       0.05      0.02      0.03       119
         461       0.65      0.14      0.23        79
         462       0.06      0.06      0.06        47
         463       0.11      0.01      0.02       104
         464       0.57      0.31      0.40       106
         465       0.20      0.03      0.05        64
         466       0.45      0.23      0.30       173
         467       0.76      0.29      0.42       107
         468       0.00      0.00      0.00       126
         469       0.00      0.00      0.00       114
         470       0.96      0.64      0.76       140
         471       0.00      0.00      0.00        79
         472       0.36      0.39      0.38       143
         473       0.23      0.03      0.06       158
         474       1.00      0.02      0.04       138
         475       0.12      0.07      0.09        59
         476       0.58      0.42      0.49        88
         477       0.81      0.50      0.62       176
         478       0.93      0.58      0.72        24
         479       0.00      0.00      0.00        92
         480       0.79      0.34      0.48       100
         481       0.44      0.04      0.07       103
         482       0.25      0.22      0.23        74
         483       0.81      0.44      0.57       105
         484       0.02      0.01      0.02        83
         485       0.12      0.06      0.08        82
         486       0.25      0.04      0.07        71
         487       0.36      0.20      0.26       120
         488       0.00      0.00      0.00       105
         489       0.67      0.21      0.32        87
         490       1.00      0.72      0.84        32
         491       0.05      0.01      0.02        69
         492       0.00      0.00      0.00        49
         493       0.00      0.00      0.00       117
         494       0.48      0.33      0.39        61
         495       0.00      0.00      0.00       344
         496       0.13      0.12      0.12        52
         497       0.00      0.00      0.00       137
         498       0.05      0.03      0.04        98
         499       0.53      0.10      0.17        79

   micro avg       0.55      0.33      0.41    173812
   macro avg       0.41      0.25      0.29    173812
weighted avg       0.53      0.33      0.39    173812
 samples avg       0.38      0.31      0.32    173812

Time taken to train the model : 0:06:58.301673
```

Out[4]:

```
['lr_with_more_title_weight_sgd_logloss_ovr.pkl']
```

5.3.2 Hyperparameter 'alpha' tuning to increase performance on Logistic Regression model (SGDClassifier with 'log' loss) with OneVsRest Classifier

In [ ]:

```
'''from sklearn.model_selection import cross_val_score
models = []
cross_val_scores=[]

def get_best_estimate_sgd(x_train_multilabel, y_train): #9:47
    start = datetime.now()
    alpha_vals = np.logspace(-4,1,20)
    for alpha in alpha_vals:
        score = []
        classifier=OneVsRestClassifier(SGDClassifier(loss='log', alpha=alpha, penalty='l1'))
        f1_scores = cross_val_score(classifier, x_train_multilabel, y_train, cv=10, scoring='f1_micro', n_jobs=-1
) ##Perform 10-fold cross validation on the train set
        cross_val_scores.append(f1_scores.mean())
        models.append(classifier)
        print("For alpha = {}, the Cross Validation Micro F1 Score is = {}".format(alpha,f1_scores.mean()))

    max_score = max(cross_val_scores)
    best_estimator = models[cross_val_scores.index(max_score)]

    print("\n\nThe best estimator obtained using Hyperparameter tuning: ",best_estimator)
    print("The best Micro Average F1-Score obtained using Hyperparameter tuning: ",max_score)
    print("Time taken to perform 10 Fold Cross validation: ",datetime.now()-start)

    return(best_estimator)'''
```

In [6]:

```
#Function to determine the best estimator using Hyperparameter tuning by using simple for loops. Not using GridSe
arch for improving time complexity. Not using K-Fold cross validation. Very basic for loop to determine best para
ms
#Note: We can't use probability scores using SGDClassifiers. Hence, we need to Calibrate the model
from sklearn.model_selection import RandomizedSearchCV

st=datetime.now()

def get_RandomSearchCV(x_train_multilabel, y_train):
    '''This function will determine the best hyperparameters using KFold CV and GridSearchCV, using K fold cross
validation. '''
    from sklearn.model_selection import KFold
    n_folds = 5
    cv_kfold = KFold(n_splits=n_folds).split(x_train_multilabel)
    alpha = np.logspace(-4,4,30)
    params  = {"estimator__alpha":alpha}
    base_estimator=OneVsRestClassifier(SGDClassifier(loss='log', penalty='l1', random_state=0), n_jobs=-1)
    rsearch_cv = RandomizedSearchCV(estimator=base_estimator, param_distributions=params, n_iter=15, cv=cv_kfold,
scoring='f1_micro', n_jobs=-1, verbose=1)
    rsearch_cv.fit(x_train_multilabel, y_train)
    return  rsearch_cv

#Get the best estimator after performing Random Search Cross Validation.
rsearch_cv = get_RandomSearchCV(x_train_multilabel,y_train)

print("Time taken to perform hyperparameter tuning: ",datetime.now()-st)
print("Best estimator: ",rsearch_cv.best_estimator_)
print("Best Cross Validation Score: ",rsearch_cv.best_score_)
```

```
Fitting 5 folds for each of 15 candidates, totalling 75 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
/root/anaconda3/lib/python3.7/site-packages/sklearn/externals/joblib/externals/loky/process_executor
.py:706: UserWarning: A worker stopped while some jobs were given to the executor. This can be cause
d by a too short worker timeout or by a memory leak.
  "timeout or by a memory leak.", UserWarning
[Parallel(n_jobs=-1)]: Done  34 tasks      | elapsed: 212.1min
/root/anaconda3/lib/python3.7/site-packages/sklearn/externals/joblib/externals/loky/process_executor
.py:706: UserWarning: A worker stopped while some jobs were given to the executor. This can be cause
d by a too short worker timeout or by a memory leak.
  "timeout or by a memory leak.", UserWarning
[Parallel(n_jobs=-1)]: Done  75 out of  75 | elapsed: 402.7min finished

Time taken to perform hyperparameter tuning:  7:03:10.630894
Best estimator:  OneVsRestClassifier(estimator=SGDClassifier(alpha=0.00018873918221350977, average=F
alse, class_weight=None,
       early_stopping=False, epsilon=0.1, eta0=0.0, fit_intercept=True,
       l1_ratio=0.15, learning_rate='optimal', loss='log', max_iter=None,
       n_iter=None, n_iter_no_change=5, n_jobs=-1, penalty='l1',
       power_t=0.5, random_state=0, shuffle=True, tol=None,
       validation_fraction=0.1, verbose=0, warm_start=False),
          n_jobs=None)
Best Cross Validation Score:  0.46735172370575395
```

5.3.3 Applying Logistic Regression model (SGDClassifier with 'log' loss) with OneVsRest Classifier with the best estimator obtained using random search.

In [8]:

```
classifier=OneVsRestClassifier(estimator=SGDClassifier(alpha=0.00018873918221350977, average=False, class_weight=None,early_stopping=False, epsilon=0.1, eta0=0.0, fit_intercept=True,l1_ratio=0.15, learning_rate='optimal', loss='log', max_iter=None,n_iter=None, n_iter_no_change=5, n_jobs=-1, penalty='l1',power_t=0.5, random_state=0, shuffle=True, tol=None,validation_fraction=0.1, verbose=0, warm_start=False),n_jobs=-1)
```

In [9]:

```
#Train the model with the best estimator obtained from above
start_t = datetime.now()
#classifier = rsearch_cv.best_estimator_
classifier.fit(x_train_multilabel, y_train)
predictions = classifier.predict(x_test_multilabel)

print("\nAccuracy :",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')

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

print (metrics.classification_report(y_test, predictions))
print("Time taken to train the model :", datetime.now() - start_t)

import joblib
joblib.dump(classifier, 'lr_with_more_title_weight_sgd_log_ovr_hyp_tuned.pkl')
```

```
Accuracy : 0.14065
Hamming loss  0.00400758
Micro-average quality numbers
Precision: 0.4251, Recall: 0.4339, Micro F1-measure: 0.4295
Macro-average quality numbers
Precision: 0.3574, Recall: 0.3544, Macro F1-measure: 0.3430

/root/anaconda3/lib/python3.7/site-packages/sklearn/metrics/classification.py:1143: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in samples with no predicted labels.
  'precision', 'predicted', average, warn_for)
/root/anaconda3/lib/python3.7/site-packages/sklearn/metrics/classification.py:1145: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in samples with no true labels.
  'recall', 'true', average, warn_for)

              precision    recall  f1-score   support

           0       0.69      0.75      0.72      5519
           1       0.44      0.39      0.41      8190
           2       0.59      0.46      0.52      6529
           3       0.42      0.56      0.48      3231
           4       0.56      0.51      0.54      6430
           5       0.46      0.49      0.48      2879
           6       0.55      0.63      0.59      5086
           7       0.63      0.66      0.65      4533
           8       0.30      0.21      0.24      3000
           9       0.48      0.67      0.56      2765
          10       0.38      0.28      0.32      3051
          11       0.55      0.44      0.49      3009
          12       0.39      0.40      0.39      2630
          13       0.36      0.41      0.39      1426
          14       0.69      0.65      0.67      2548
          15       0.38      0.34      0.36      2371
          16       0.40      0.34      0.37       873
          17       0.60      0.69      0.64      2151
          18       0.37      0.31      0.34      2204
          19       0.44      0.49      0.46       831
          20       0.58      0.52      0.55      1860
          21       0.20      0.21      0.20      2023
          22       0.34      0.32      0.33      1513
          23       0.70      0.67      0.68      1207
```

24	0.36	0.41	0.38	506
25	0.41	0.48	0.44	425
26	0.41	0.48	0.44	793
27	0.46	0.40	0.43	1291
28	0.62	0.45	0.52	1208
29	0.11	0.24	0.15	406
30	0.25	0.37	0.30	504
31	0.12	0.24	0.16	732
32	0.26	0.42	0.32	441
33	0.39	0.32	0.35	1645
34	0.42	0.33	0.37	1058
35	0.49	0.63	0.55	946
36	0.32	0.40	0.35	644
37	0.49	0.75	0.59	136
38	0.45	0.51	0.48	570
39	0.49	0.45	0.47	766
40	0.44	0.43	0.43	1132
41	0.11	0.27	0.16	174
42	0.51	0.70	0.59	210
43	0.40	0.53	0.46	433
44	0.51	0.52	0.51	626
45	0.35	0.42	0.38	852
46	0.53	0.55	0.54	534
47	0.17	0.24	0.20	350
48	0.50	0.55	0.52	496
49	0.59	0.67	0.63	785
50	0.18	0.27	0.22	475
51	0.09	0.18	0.12	305
52	0.16	0.14	0.15	251
53	0.43	0.55	0.49	914
54	0.31	0.25	0.28	728
55	0.11	0.12	0.11	258
56	0.29	0.29	0.29	821
57	0.17	0.19	0.18	541
58	0.46	0.39	0.42	748
59	0.75	0.75	0.75	724
60	0.25	0.12	0.16	660
61	0.47	0.29	0.36	235
62	0.82	0.81	0.82	718
63	0.72	0.69	0.71	468
64	0.21	0.51	0.30	191
65	0.26	0.12	0.16	429
66	0.14	0.21	0.17	415
67	0.31	0.58	0.41	274
68	0.58	0.62	0.60	510
69	0.49	0.53	0.51	466
70	0.19	0.17	0.18	305
71	0.17	0.27	0.20	247
72	0.63	0.60	0.61	401
73	0.34	0.80	0.47	86
74	0.44	0.47	0.45	120
75	0.79	0.78	0.79	129
76	0.14	0.02	0.03	473
77	0.22	0.36	0.27	143
78	0.65	0.59	0.62	347
79	0.39	0.32	0.35	479
80	0.19	0.42	0.26	279
81	0.44	0.22	0.29	461
82	0.08	0.07	0.07	298
83	0.39	0.60	0.47	396
84	0.37	0.44	0.40	184
85	0.41	0.28	0.34	573
86	0.21	0.17	0.19	325
87	0.33	0.37	0.35	273
88	0.32	0.36	0.34	135
89	0.19	0.26	0.22	232
90	0.35	0.56	0.43	409
91	0.27	0.43	0.33	420
92	0.61	0.65	0.63	408
93	0.51	0.49	0.50	241
94	0.23	0.11	0.15	211
95	0.14	0.23	0.17	277
96	0.21	0.08	0.12	410
97	0.74	0.43	0.55	501
98	0.43	0.65	0.51	136
99	0.37	0.43	0.40	239
100	0.21	0.17	0.18	324
101	0.75	0.77	0.76	277
102	0.73	0.76	0.75	613
103	0.23	0.25	0.24	157
104	0.15	0.19	0.17	295
105	0.35	0.49	0.41	334
106	0.25	0.15	0.19	335

107	0.47	0.56	0.51	389
108	0.35	0.39	0.37	251
109	0.37	0.48	0.42	317
110	0.13	0.17	0.15	187
111	0.17	0.28	0.21	140
112	0.14	0.41	0.20	154
113	0.57	0.25	0.34	332
114	0.32	0.28	0.30	323
115	0.21	0.27	0.24	344
116	0.52	0.61	0.56	370
117	0.42	0.32	0.36	313
118	0.71	0.78	0.75	874
119	0.20	0.32	0.24	293
120	0.07	0.04	0.05	200
121	0.63	0.60	0.61	463
122	0.22	0.26	0.24	119
123	0.05	0.02	0.02	256
124	0.53	0.77	0.63	195
125	0.15	0.28	0.20	138
126	0.43	0.62	0.51	376
127	0.04	0.08	0.05	122
128	0.18	0.13	0.16	252
129	0.22	0.09	0.13	144
130	0.21	0.25	0.23	150
131	0.07	0.09	0.08	210
132	0.49	0.28	0.35	361
133	0.77	0.71	0.74	453
134	0.73	0.85	0.79	124
135	0.07	0.05	0.06	91
136	0.10	0.34	0.16	128
137	0.32	0.44	0.37	218
138	0.39	0.27	0.32	243
139	0.24	0.21	0.23	149
140	0.63	0.52	0.57	318
141	0.06	0.16	0.09	159
142	0.55	0.51	0.53	274
143	0.66	0.80	0.72	362
144	0.35	0.37	0.36	118
145	0.26	0.41	0.32	164
146	0.52	0.38	0.44	461
147	0.52	0.59	0.55	159
148	0.17	0.23	0.19	166
149	0.54	0.58	0.56	346
150	0.41	0.15	0.22	350
151	0.82	0.67	0.74	55
152	0.36	0.59	0.45	387
153	0.18	0.18	0.18	150
154	0.26	0.16	0.20	281
155	0.24	0.18	0.20	202
156	0.53	0.72	0.61	130
157	0.17	0.16	0.17	245
158	0.78	0.67	0.72	177
159	0.30	0.47	0.37	130
160	0.34	0.28	0.30	336
161	0.45	0.67	0.54	220
162	0.09	0.17	0.12	229
163	0.78	0.60	0.68	316
164	0.48	0.43	0.45	283
165	0.27	0.40	0.32	197
166	0.17	0.49	0.26	101
167	0.28	0.24	0.26	231
168	0.34	0.35	0.34	370
169	0.26	0.30	0.28	258
170	0.13	0.10	0.11	101
171	0.28	0.29	0.28	89
172	0.38	0.37	0.37	193
173	0.36	0.40	0.38	309
174	0.19	0.27	0.23	172
175	0.77	0.87	0.82	95
176	0.73	0.63	0.68	346
177	0.79	0.57	0.66	322
178	0.44	0.56	0.49	232
179	0.43	0.10	0.16	125
180	0.38	0.35	0.36	145
181	0.09	0.23	0.13	77
182	0.09	0.08	0.08	182
183	0.35	0.42	0.38	257
184	0.06	0.07	0.06	216
185	0.22	0.20	0.21	242
186	0.28	0.21	0.24	165
187	0.50	0.63	0.56	263
188	0.26	0.21	0.23	174
189	0.40	0.50	0.44	136

190	0.51	0.58	0.54	202
191	0.21	0.19	0.20	134
192	0.64	0.53	0.58	230
193	0.18	0.30	0.23	90
194	0.42	0.55	0.47	185
195	0.07	0.05	0.06	156
196	0.13	0.14	0.13	160
197	0.43	0.14	0.21	266
198	0.16	0.15	0.16	284
199	0.10	0.14	0.12	145
200	0.58	0.79	0.67	212
201	0.38	0.30	0.33	317
202	0.57	0.68	0.62	427
203	0.21	0.21	0.21	232
204	0.28	0.31	0.29	217
205	0.38	0.65	0.48	527
206	0.08	0.12	0.09	124
207	0.12	0.24	0.16	103
208	0.73	0.59	0.65	287
209	0.14	0.15	0.15	193
210	0.57	0.39	0.46	220
211	0.69	0.27	0.39	140
212	0.11	0.15	0.13	161
213	0.15	0.36	0.21	72
214	0.58	0.44	0.50	396
215	0.54	0.40	0.46	134
216	0.38	0.12	0.18	400
217	0.34	0.39	0.36	75
218	0.80	0.83	0.82	219
219	0.56	0.45	0.50	210
220	0.82	0.73	0.77	298
221	0.90	0.70	0.79	266
222	0.61	0.49	0.54	290
223	0.09	0.09	0.09	128
224	0.56	0.45	0.50	159
225	0.24	0.40	0.30	164
226	0.35	0.40	0.37	144
227	0.43	0.49	0.46	276
228	0.07	0.06	0.06	235
229	0.18	0.09	0.12	216
230	0.29	0.24	0.26	228
231	0.45	0.69	0.55	64
232	0.13	0.23	0.17	103
233	0.52	0.40	0.45	216
234	0.27	0.25	0.26	116
235	0.33	0.39	0.36	77
236	0.85	0.75	0.79	67
237	0.52	0.06	0.11	218
238	0.13	0.05	0.07	139
239	0.06	0.02	0.03	94
240	0.20	0.44	0.27	77
241	0.31	0.08	0.13	167
242	0.48	0.45	0.46	86
243	0.30	0.22	0.26	58
244	0.42	0.33	0.37	269
245	0.16	0.18	0.17	112
246	0.85	0.86	0.86	255
247	0.15	0.29	0.20	58
248	0.04	0.10	0.06	81
249	0.04	0.03	0.04	131
250	0.16	0.26	0.20	93
251	0.36	0.42	0.39	154
252	0.12	0.09	0.10	129
253	0.31	0.46	0.37	83
254	0.24	0.16	0.19	191
255	0.16	0.05	0.08	219
256	0.14	0.05	0.07	130
257	0.40	0.42	0.41	93
258	0.63	0.61	0.62	217
259	0.10	0.18	0.13	141
260	0.59	0.28	0.38	143
261	0.47	0.17	0.25	219
262	0.37	0.43	0.40	107
263	0.32	0.36	0.34	236
264	0.14	0.30	0.19	119
265	0.12	0.31	0.17	72
266	0.23	0.20	0.21	70
267	0.18	0.16	0.17	107
268	0.60	0.50	0.55	169
269	0.24	0.25	0.25	129
270	0.72	0.62	0.66	159
271	0.79	0.48	0.60	190
272	0.44	0.23	0.30	248

273	0.74	0.77	0.75	264
274	0.68	0.75	0.71	105
275	0.11	0.14	0.12	104
276	0.03	0.08	0.04	115
277	0.81	0.65	0.72	170
278	0.54	0.46	0.50	145
279	0.86	0.72	0.78	230
280	0.50	0.49	0.49	80
281	0.64	0.73	0.68	217
282	0.72	0.53	0.61	175
283	0.21	0.18	0.19	269
284	0.49	0.43	0.46	74
285	0.67	0.56	0.61	206
286	0.84	0.77	0.80	227
287	0.45	0.59	0.51	130
288	0.22	0.16	0.19	129
289	0.15	0.06	0.09	80
290	0.07	0.17	0.10	99
291	0.45	0.47	0.46	208
292	0.34	0.15	0.21	67
293	0.64	0.50	0.56	109
294	0.21	0.37	0.27	140
295	0.17	0.28	0.21	241
296	0.19	0.19	0.19	72
297	0.07	0.10	0.08	107
298	0.39	0.44	0.41	61
299	0.22	0.48	0.30	77
300	0.13	0.19	0.15	111
301	0.00	0.00	0.00	126
302	0.03	0.03	0.03	73
303	0.45	0.51	0.48	176
304	0.82	0.80	0.81	230
305	0.88	0.76	0.82	156
306	0.36	0.49	0.41	146
307	0.28	0.18	0.22	98
308	0.05	0.09	0.06	78
309	0.10	0.16	0.12	94
310	0.35	0.48	0.40	162
311	0.68	0.65	0.66	116
312	0.46	0.42	0.44	57
313	0.45	0.08	0.13	65
314	0.32	0.48	0.39	138
315	0.44	0.37	0.40	195
316	0.42	0.42	0.42	69
317	0.04	0.18	0.07	134
318	0.37	0.46	0.41	148
319	0.43	0.47	0.45	161
320	0.13	0.32	0.19	104
321	0.59	0.70	0.64	156
322	0.39	0.40	0.40	134
323	0.51	0.45	0.48	232
324	0.12	0.28	0.17	92
325	0.16	0.27	0.20	197
326	0.04	0.07	0.05	126
327	0.41	0.06	0.11	115
328	0.93	0.63	0.75	198
329	0.45	0.36	0.40	125
330	0.33	0.20	0.25	81
331	0.15	0.19	0.17	94
332	0.07	0.07	0.07	56
333	0.12	0.14	0.13	260
334	0.39	0.18	0.25	60
335	0.15	0.20	0.17	110
336	0.35	0.52	0.42	71
337	0.14	0.20	0.16	66
338	0.38	0.45	0.41	150
339	0.02	0.02	0.02	54
340	0.64	0.67	0.65	195
341	0.09	0.13	0.10	79
342	0.11	0.37	0.17	38
343	0.24	0.56	0.34	43
344	0.24	0.40	0.30	68
345	0.40	0.40	0.40	73
346	0.09	0.16	0.11	116
347	0.77	0.42	0.55	111
348	0.11	0.11	0.11	63
349	0.67	0.79	0.73	104
350	0.53	0.39	0.45	44
351	0.07	0.15	0.09	40
352	0.69	0.46	0.56	136
353	0.40	0.39	0.40	54
354	0.10	0.12	0.11	134
355	0.33	0.36	0.34	120

356	0.40	0.43	0.41	228
357	0.41	0.42	0.42	269
358	0.41	0.30	0.35	80
359	0.77	0.62	0.69	140
360	0.19	0.26	0.22	125
361	0.83	0.72	0.77	169
362	0.04	0.14	0.06	56
363	0.70	0.75	0.73	154
364	0.05	0.07	0.06	58
365	0.12	0.20	0.15	71
366	0.59	0.70	0.64	54
367	0.10	0.16	0.12	116
368	0.02	0.04	0.03	54
369	0.03	0.07	0.04	71
370	0.05	0.05	0.05	61
371	0.30	0.24	0.27	71
372	0.47	0.65	0.55	52
373	0.55	0.46	0.50	150
374	0.33	0.26	0.29	93
375	0.14	0.06	0.08	67
376	0.00	0.00	0.00	76
377	0.64	0.25	0.36	106
378	0.16	0.03	0.06	86
379	0.10	0.14	0.11	14
380	0.80	0.61	0.69	122
381	0.10	0.09	0.09	104
382	0.16	0.26	0.20	66
383	0.32	0.37	0.35	110
384	0.14	0.02	0.03	155
385	0.20	0.18	0.19	50
386	0.14	0.19	0.16	64
387	0.05	0.12	0.07	93
388	0.36	0.49	0.42	102
389	0.07	0.02	0.03	108
390	0.84	0.61	0.71	178
391	0.55	0.23	0.32	115
392	0.56	0.52	0.54	42
393	0.00	0.00	0.00	134
394	0.16	0.09	0.11	112
395	0.43	0.34	0.38	176
396	0.24	0.18	0.21	125
397	0.61	0.50	0.55	224
398	0.73	0.68	0.70	63
399	0.05	0.07	0.06	59
400	0.20	0.41	0.27	63
401	0.04	0.07	0.05	98
402	0.25	0.25	0.25	162
403	0.23	0.29	0.26	83
404	0.64	0.95	0.77	19
405	0.10	0.18	0.13	92
406	0.50	0.44	0.47	41
407	0.40	0.28	0.33	43
408	0.65	0.35	0.46	160
409	0.13	0.26	0.17	50
410	0.06	0.05	0.06	19
411	0.25	0.15	0.19	175
412	0.09	0.04	0.06	72
413	0.30	0.13	0.18	95
414	0.15	0.16	0.16	97
415	0.26	0.35	0.30	48
416	0.38	0.31	0.34	83
417	0.03	0.07	0.05	40
418	0.16	0.24	0.20	91
419	0.37	0.41	0.39	90
420	0.19	0.19	0.19	37
421	0.12	0.08	0.09	66
422	0.33	0.47	0.39	73
423	0.31	0.29	0.30	56
424	0.79	0.91	0.85	33
425	0.12	0.08	0.10	76
426	0.03	0.02	0.03	81
427	0.80	0.73	0.76	150
428	0.53	0.79	0.64	29
429	0.73	0.46	0.56	389
430	0.49	0.52	0.50	167
431	0.46	0.11	0.17	123
432	0.24	0.36	0.29	39
433	0.29	0.34	0.32	82
434	0.98	0.73	0.83	66
435	0.41	0.49	0.45	93
436	0.48	0.29	0.36	87
437	0.19	0.14	0.16	86
438	0.69	0.63	0.66	104

```
        439       0.20      0.23      0.21       100
        440       0.20      0.01      0.01       141
        441       0.31      0.34      0.32       110
        442       0.20      0.21      0.20       123
        443       0.44      0.20      0.27        71
        444       0.22      0.27      0.24       109
        445       0.25      0.27      0.26        48
        446       0.35      0.32      0.33        76
        447       0.12      0.29      0.17        38
        448       0.55      0.78      0.64        81
        449       0.42      0.24      0.31       132
        450       0.44      0.35      0.39        81
        451       0.46      0.47      0.47        76
        452       0.02      0.05      0.03        44
        453       0.07      0.09      0.08        44
        454       0.41      0.59      0.48        70
        455       0.23      0.15      0.18       155
        456       0.25      0.40      0.31        43
        457       0.12      0.15      0.13        72
        458       0.17      0.18      0.17        62
        459       0.09      0.20      0.13        69
        460       0.04      0.02      0.02       119
        461       0.58      0.35      0.44        79
        462       0.12      0.23      0.16        47
        463       0.26      0.09      0.13       104
        464       0.44      0.42      0.43       106
        465       0.12      0.14      0.13        64
        466       0.41      0.42      0.42       173
        467       0.58      0.42      0.49       107
        468       0.47      0.20      0.28       126
        469       0.06      0.02      0.03       114
        470       0.83      0.89      0.86       140
        471       0.74      0.25      0.38        79
        472       0.36      0.48      0.41       143
        473       0.29      0.30      0.29       158
        474       0.12      0.07      0.09       138
        475       0.18      0.20      0.19        59
        476       0.47      0.44      0.46        88
        477       0.77      0.69      0.73       176
        478       0.78      0.88      0.82        24
        479       0.14      0.14      0.14        92
        480       0.80      0.57      0.67       100
        481       0.35      0.33      0.34       103
        482       0.13      0.18      0.15        74
        483       0.69      0.56      0.62       105
        484       0.05      0.06      0.06        83
        485       0.12      0.13      0.12        82
        486       0.21      0.20      0.20        71
        487       0.41      0.33      0.36       120
        488       0.04      0.04      0.04       105
        489       0.56      0.43      0.48        87
        490       0.83      0.91      0.87        32
        491       0.03      0.04      0.04        69
        492       0.33      0.04      0.07        49
        493       0.06      0.03      0.04       117
        494       0.31      0.26      0.28        61
        495       0.92      0.38      0.53       344
        496       0.09      0.25      0.14        52
        497       0.53      0.22      0.31       137
        498       0.20      0.17      0.19        98
        499       0.30      0.30      0.30        79

   micro avg       0.43      0.43      0.43    173812
   macro avg       0.36      0.35      0.34    173812
weighted avg       0.44      0.43      0.43    173812
 samples avg       0.39      0.41      0.36    173812

Time taken to train the model : 0:06:53.827418
```

Out[9]:

```
['lr_with_more_title_weight_sgd_log_ovr_hyp_tuned.pkl']
```

5.4 Applying Linear-SVM (SGDClassifier with 'hinge' loss) with OneVsRest Classifier

In [8]:

```
start = datetime.now()

#Build the model
classifier = OneVsRestClassifier(SGDClassifier(loss='hinge', alpha=0.00001, penalty='l1', verbose=1), 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')

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

print (metrics.classification_report(y_test, predictions))
print("Time taken to train the model :", datetime.now() - start)

#Save the model for future use.
import joblib
joblib.dump(classifier, 'linear_SVM_with_more_title_weight_sgd_hinge_ovr.pkl')
```

```
Accuracy : 0.11029
Hamming loss  0.00594184
Micro-average quality numbers
Precision: 0.2872, Recall: 0.4785, Micro F1-measure: 0.3589
Macro-average quality numbers
Precision: 0.2076, Recall: 0.4065, Macro F1-measure: 0.2671
              precision    recall  f1-score   support

           0       0.70      0.80      0.75      5519
           1       0.44      0.45      0.45      8190
           2       0.53      0.51      0.52      6529
           3       0.51      0.63      0.56      3231
           4       0.53      0.54      0.54      6430
           5       0.41      0.49      0.45      2879
           6       0.58      0.62      0.60      5086
           7       0.57      0.68      0.62      4533
           8       0.21      0.23      0.22      3000
           9       0.56      0.66      0.60      2765
          10       0.30      0.37      0.33      3051
          11       0.48      0.50      0.49      3009
          12       0.36      0.44      0.39      2630
          13       0.34      0.44      0.39      1426
          14       0.57      0.67      0.62      2548
          15       0.36      0.39      0.38      2371
          16       0.29      0.38      0.33       873
          17       0.56      0.73      0.63      2151
          18       0.32      0.36      0.34      2204
          19       0.28      0.50      0.36       831
          20       0.51      0.59      0.54      1860
          21       0.18      0.24      0.20      2023
          22       0.28      0.36      0.31      1513
          23       0.47      0.70      0.56      1207
          24       0.25      0.42      0.31       506
          25       0.24      0.47      0.32       425
          26       0.31      0.53      0.39       793
          27       0.39      0.45      0.41      1291
          28       0.43      0.48      0.46      1208
          29       0.11      0.28      0.15       406
          30       0.17      0.38      0.24       504
          31       0.11      0.23      0.15       732
          32       0.19      0.42      0.26       441
          33       0.36      0.45      0.40      1645
          34       0.25      0.37      0.30      1058
          35       0.46      0.64      0.53       946
          36       0.24      0.43      0.31       644
          37       0.24      0.74      0.36       136
          38       0.33      0.48      0.39       570
          39       0.30      0.45      0.36       766
          40       0.34      0.48      0.39      1132
          41       0.08      0.29      0.13       174
```

42	0.33	0.59	0.42	210
43	0.36	0.52	0.43	433
44	0.30	0.56	0.39	626
45	0.28	0.46	0.35	852
46	0.32	0.59	0.41	534
47	0.13	0.35	0.19	350
48	0.30	0.57	0.39	496
49	0.61	0.71	0.65	785
50	0.13	0.26	0.17	475
51	0.09	0.30	0.14	305
52	0.08	0.16	0.11	251
53	0.30	0.49	0.38	914
54	0.18	0.28	0.22	728
55	0.06	0.14	0.09	258
56	0.22	0.38	0.28	821
57	0.15	0.28	0.19	541
58	0.30	0.43	0.35	748
59	0.63	0.75	0.68	724
60	0.18	0.26	0.21	660
61	0.17	0.31	0.22	235
62	0.52	0.79	0.63	718
63	0.48	0.78	0.59	468
64	0.17	0.47	0.25	191
65	0.11	0.25	0.16	429
66	0.14	0.29	0.18	415
67	0.26	0.59	0.36	274
68	0.42	0.62	0.50	510
69	0.28	0.52	0.37	466
70	0.10	0.21	0.14	305
71	0.13	0.29	0.18	247
72	0.37	0.58	0.45	401
73	0.28	0.84	0.42	86
74	0.15	0.47	0.22	120
75	0.31	0.69	0.43	129
76	0.09	0.14	0.11	473
77	0.10	0.38	0.15	143
78	0.34	0.59	0.43	347
79	0.23	0.33	0.27	479
80	0.18	0.51	0.26	279
81	0.18	0.37	0.25	461
82	0.08	0.23	0.12	298
83	0.29	0.56	0.38	396
84	0.16	0.49	0.25	184
85	0.22	0.37	0.27	573
86	0.09	0.20	0.13	325
87	0.21	0.45	0.29	273
88	0.11	0.35	0.16	135
89	0.12	0.28	0.17	232
90	0.30	0.48	0.37	409
91	0.28	0.39	0.33	420
92	0.39	0.62	0.48	408
93	0.22	0.56	0.32	241
94	0.07	0.19	0.10	211
95	0.12	0.29	0.17	277
96	0.11	0.17	0.13	410
97	0.39	0.58	0.46	501
98	0.15	0.67	0.25	136
99	0.22	0.46	0.29	239
100	0.14	0.25	0.18	324
101	0.49	0.77	0.59	277
102	0.64	0.80	0.71	613
103	0.12	0.32	0.18	157
104	0.11	0.24	0.15	295
105	0.35	0.53	0.42	334
106	0.23	0.41	0.29	335
107	0.38	0.60	0.47	389
108	0.21	0.39	0.27	251
109	0.29	0.49	0.36	317
110	0.06	0.21	0.09	187
111	0.07	0.29	0.11	140
112	0.17	0.64	0.27	154
113	0.20	0.39	0.26	332
114	0.19	0.35	0.25	323
115	0.17	0.38	0.24	344
116	0.36	0.57	0.44	370
117	0.19	0.34	0.24	313
118	0.62	0.81	0.70	874
119	0.16	0.34	0.21	293
120	0.05	0.12	0.07	200
121	0.40	0.56	0.47	463
122	0.11	0.33	0.16	119
123	0.03	0.05	0.03	256
124	0.40	0.77	0.53	195

125	0.11	0.36	0.16	138
126	0.40	0.62	0.49	376
127	0.03	0.12	0.05	122
128	0.07	0.15	0.10	252
129	0.20	0.35	0.25	144
130	0.10	0.29	0.15	150
131	0.06	0.15	0.09	210
132	0.22	0.37	0.28	361
133	0.53	0.66	0.59	453
134	0.38	0.81	0.52	124
135	0.03	0.14	0.06	91
136	0.11	0.45	0.18	128
137	0.20	0.47	0.28	218
138	0.12	0.32	0.17	243
139	0.12	0.32	0.17	149
140	0.34	0.55	0.42	318
141	0.08	0.24	0.11	159
142	0.35	0.52	0.42	274
143	0.61	0.86	0.71	362
144	0.12	0.36	0.18	118
145	0.16	0.52	0.25	164
146	0.26	0.48	0.33	461
147	0.23	0.57	0.33	159
148	0.09	0.27	0.14	166
149	0.41	0.63	0.50	346
150	0.13	0.26	0.17	350
151	0.24	0.75	0.36	55
152	0.40	0.59	0.48	387
153	0.19	0.32	0.23	150
154	0.09	0.20	0.12	281
155	0.11	0.31	0.16	202
156	0.33	0.69	0.45	130
157	0.14	0.26	0.18	245
158	0.47	0.70	0.56	177
159	0.14	0.46	0.22	130
160	0.16	0.30	0.21	336
161	0.39	0.70	0.50	220
162	0.09	0.21	0.12	229
163	0.43	0.57	0.49	316
164	0.25	0.54	0.35	283
165	0.16	0.34	0.22	197
166	0.21	0.56	0.30	101
167	0.15	0.35	0.21	231
168	0.19	0.37	0.25	370
169	0.20	0.30	0.24	258
170	0.06	0.23	0.10	101
171	0.09	0.37	0.15	89
172	0.18	0.42	0.25	193
173	0.26	0.44	0.33	309
174	0.09	0.23	0.13	172
175	0.31	0.83	0.45	95
176	0.54	0.69	0.60	346
177	0.41	0.67	0.51	322
178	0.27	0.58	0.37	232
179	0.05	0.16	0.08	125
180	0.20	0.50	0.28	145
181	0.04	0.22	0.07	77
182	0.08	0.23	0.12	182
183	0.27	0.56	0.36	257
184	0.09	0.19	0.12	216
185	0.14	0.30	0.19	242
186	0.11	0.30	0.16	165
187	0.33	0.62	0.43	263
188	0.10	0.25	0.14	174
189	0.38	0.53	0.44	136
190	0.40	0.66	0.50	202
191	0.09	0.27	0.14	134
192	0.26	0.51	0.35	230
193	0.09	0.28	0.14	90
194	0.29	0.58	0.39	185
195	0.04	0.18	0.07	156
196	0.06	0.15	0.08	160
197	0.15	0.25	0.18	266
198	0.17	0.31	0.22	284
199	0.04	0.08	0.05	145
200	0.47	0.81	0.60	212
201	0.23	0.45	0.30	317
202	0.47	0.66	0.55	427
203	0.14	0.20	0.16	232
204	0.16	0.39	0.23	217
205	0.35	0.59	0.44	527
206	0.04	0.11	0.06	124
207	0.20	0.43	0.27	103

208	0.36	0.57	0.44	287
209	0.06	0.15	0.09	193
210	0.21	0.43	0.28	220
211	0.11	0.32	0.17	140
212	0.06	0.17	0.09	161
213	0.23	0.58	0.33	72
214	0.41	0.60	0.49	396
215	0.20	0.49	0.28	134
216	0.20	0.30	0.24	400
217	0.09	0.41	0.15	75
218	0.61	0.83	0.70	219
219	0.25	0.45	0.32	210
220	0.53	0.73	0.61	298
221	0.62	0.78	0.69	266
222	0.35	0.57	0.44	290
223	0.04	0.12	0.06	128
224	0.21	0.52	0.30	159
225	0.14	0.46	0.21	164
226	0.21	0.47	0.29	144
227	0.33	0.55	0.41	276
228	0.03	0.09	0.04	235
229	0.05	0.11	0.06	216
230	0.09	0.26	0.13	228
231	0.13	0.58	0.21	64
232	0.05	0.17	0.07	103
233	0.25	0.46	0.32	216
234	0.13	0.27	0.17	116
235	0.14	0.47	0.22	77
236	0.28	0.78	0.41	67
237	0.13	0.27	0.17	218
238	0.08	0.27	0.13	139
239	0.03	0.07	0.04	94
240	0.15	0.39	0.22	77
241	0.06	0.17	0.09	167
242	0.22	0.51	0.30	86
243	0.04	0.21	0.07	58
244	0.27	0.48	0.34	269
245	0.07	0.21	0.11	112
246	0.61	0.82	0.70	255
247	0.08	0.33	0.12	58
248	0.01	0.06	0.02	81
249	0.03	0.14	0.05	131
250	0.11	0.31	0.16	93
251	0.19	0.52	0.28	154
252	0.03	0.09	0.05	129
253	0.13	0.47	0.20	83
254	0.08	0.17	0.11	191
255	0.06	0.12	0.08	219
256	0.03	0.12	0.05	130
257	0.15	0.43	0.22	93
258	0.35	0.57	0.43	217
259	0.10	0.29	0.15	141
260	0.21	0.36	0.26	143
261	0.11	0.23	0.15	219
262	0.13	0.40	0.20	107
263	0.23	0.42	0.30	236
264	0.09	0.29	0.14	119
265	0.11	0.42	0.17	72
266	0.06	0.24	0.10	70
267	0.11	0.34	0.17	107
268	0.26	0.54	0.35	169
269	0.15	0.36	0.21	129
270	0.30	0.63	0.40	159
271	0.28	0.58	0.38	190
272	0.19	0.40	0.26	248
273	0.57	0.79	0.67	264
274	0.46	0.72	0.57	105
275	0.06	0.21	0.10	104
276	0.04	0.14	0.06	115
277	0.35	0.66	0.46	170
278	0.30	0.59	0.40	145
279	0.43	0.78	0.55	230
280	0.17	0.49	0.25	80
281	0.41	0.60	0.49	217
282	0.35	0.61	0.44	175
283	0.16	0.29	0.20	269
284	0.14	0.46	0.21	74
285	0.29	0.63	0.39	206
286	0.46	0.75	0.57	227
287	0.18	0.56	0.28	130
288	0.06	0.12	0.08	129
289	0.05	0.25	0.08	80
290	0.05	0.18	0.07	99

291	0.28	0.53	0.37	208
292	0.04	0.19	0.06	67
293	0.22	0.60	0.32	109
294	0.14	0.44	0.22	140
295	0.13	0.29	0.18	241
296	0.09	0.35	0.15	72
297	0.08	0.24	0.12	107
298	0.23	0.56	0.32	61
299	0.32	0.65	0.43	77
300	0.06	0.23	0.10	111
301	0.01	0.02	0.01	126
302	0.05	0.12	0.07	73
303	0.27	0.47	0.34	176
304	0.71	0.83	0.77	230
305	0.50	0.75	0.60	156
306	0.28	0.54	0.37	146
307	0.10	0.28	0.14	98
308	0.01	0.08	0.02	78
309	0.06	0.22	0.09	94
310	0.23	0.51	0.32	162
311	0.31	0.56	0.40	116
312	0.09	0.33	0.14	57
313	0.03	0.09	0.04	65
314	0.17	0.38	0.23	138
315	0.24	0.32	0.28	195
316	0.16	0.46	0.24	69
317	0.09	0.25	0.13	134
318	0.23	0.59	0.33	148
319	0.38	0.57	0.45	161
320	0.11	0.32	0.16	104
321	0.38	0.67	0.48	156
322	0.16	0.47	0.24	134
323	0.32	0.49	0.38	232
324	0.07	0.22	0.11	92
325	0.17	0.35	0.23	197
326	0.05	0.17	0.08	126
327	0.02	0.07	0.03	115
328	0.52	0.73	0.61	198
329	0.19	0.40	0.25	125
330	0.12	0.35	0.18	81
331	0.06	0.18	0.09	94
332	0.07	0.16	0.09	56
333	0.07	0.17	0.10	260
334	0.06	0.22	0.10	60
335	0.08	0.23	0.12	110
336	0.20	0.58	0.30	71
337	0.03	0.14	0.05	66
338	0.17	0.41	0.24	150
339	0.01	0.06	0.02	54
340	0.50	0.67	0.58	195
341	0.30	0.57	0.40	79
342	0.13	0.53	0.21	38
343	0.15	0.49	0.23	43
344	0.13	0.32	0.19	68
345	0.18	0.44	0.26	73
346	0.05	0.17	0.08	116
347	0.17	0.59	0.27	111
348	0.04	0.19	0.07	63
349	0.35	0.73	0.47	104
350	0.19	0.57	0.28	44
351	0.14	0.38	0.20	40
352	0.31	0.62	0.42	136
353	0.07	0.19	0.10	54
354	0.06	0.19	0.09	134
355	0.17	0.50	0.26	120
356	0.29	0.45	0.35	228
357	0.30	0.40	0.34	269
358	0.18	0.46	0.26	80
359	0.32	0.73	0.45	140
360	0.11	0.33	0.16	125
361	0.58	0.80	0.68	169
362	0.03	0.09	0.04	56
363	0.53	0.76	0.63	154
364	0.10	0.38	0.16	58
365	0.08	0.27	0.12	71
366	0.35	0.72	0.47	54
367	0.05	0.22	0.08	116
368	0.04	0.11	0.06	54
369	0.03	0.11	0.04	71
370	0.02	0.08	0.03	61
371	0.05	0.23	0.09	71
372	0.21	0.48	0.30	52
373	0.31	0.58	0.40	150

374	0.14	0.47	0.22	93
375	0.04	0.16	0.07	67
376	0.01	0.04	0.02	76
377	0.16	0.31	0.21	106
378	0.03	0.09	0.05	86
379	0.02	0.29	0.04	14
380	0.21	0.56	0.30	122
381	0.05	0.21	0.09	104
382	0.06	0.29	0.10	66
383	0.15	0.40	0.22	110
384	0.04	0.11	0.06	155
385	0.09	0.44	0.16	50
386	0.09	0.23	0.13	64
387	0.06	0.20	0.10	93
388	0.18	0.43	0.25	102
389	0.03	0.09	0.05	108
390	0.57	0.73	0.64	178
391	0.11	0.31	0.17	115
392	0.20	0.57	0.29	42
393	0.02	0.04	0.03	134
394	0.05	0.13	0.07	112
395	0.15	0.39	0.22	176
396	0.09	0.22	0.13	125
397	0.41	0.49	0.45	224
398	0.29	0.71	0.41	63
399	0.02	0.12	0.04	59
400	0.13	0.46	0.20	63
401	0.10	0.37	0.15	98
402	0.10	0.21	0.13	162
403	0.10	0.31	0.15	83
404	0.31	0.89	0.46	19
405	0.06	0.24	0.10	92
406	0.08	0.49	0.14	41
407	0.26	0.49	0.34	43
408	0.27	0.47	0.35	160
409	0.06	0.22	0.10	50
410	0.01	0.11	0.02	19
411	0.14	0.27	0.18	175
412	0.06	0.15	0.08	72
413	0.07	0.14	0.09	95
414	0.07	0.25	0.11	97
415	0.05	0.19	0.08	48
416	0.18	0.41	0.25	83
417	0.04	0.15	0.06	40
418	0.06	0.16	0.09	91
419	0.18	0.47	0.26	90
420	0.07	0.27	0.11	37
421	0.04	0.14	0.06	66
422	0.15	0.49	0.23	73
423	0.12	0.32	0.17	56
424	0.42	0.91	0.57	33
425	0.04	0.12	0.06	76
426	0.08	0.15	0.10	81
427	0.47	0.79	0.59	150
428	0.36	0.72	0.48	29
429	0.87	0.87	0.87	389
430	0.28	0.54	0.37	167
431	0.07	0.16	0.10	123
432	0.13	0.38	0.19	39
433	0.13	0.39	0.20	82
434	0.47	0.77	0.59	66
435	0.27	0.46	0.34	93
436	0.18	0.48	0.27	87
437	0.05	0.15	0.08	86
438	0.32	0.67	0.43	104
439	0.06	0.21	0.09	100
440	0.05	0.13	0.07	141
441	0.19	0.48	0.27	110
442	0.10	0.25	0.14	123
443	0.14	0.32	0.20	71
444	0.12	0.33	0.18	109
445	0.08	0.33	0.14	48
446	0.21	0.47	0.29	76
447	0.05	0.29	0.08	38
448	0.23	0.69	0.35	81
449	0.23	0.42	0.30	132
450	0.15	0.46	0.22	81
451	0.17	0.49	0.26	76
452	0.05	0.14	0.07	44
453	0.03	0.16	0.05	44
454	0.24	0.63	0.35	70
455	0.13	0.39	0.20	155
456	0.09	0.33	0.14	43

```
        457       0.15      0.50      0.24        72
        458       0.04      0.15      0.07        62
        459       0.07      0.33      0.12        69
        460       0.01      0.03      0.02       119
        461       0.28      0.39      0.33        79
        462       0.07      0.15      0.09        47
        463       0.13      0.37      0.19       104
        464       0.25      0.48      0.32       106
        465       0.14      0.39      0.21        64
        466       0.28      0.40      0.33       173
        467       0.16      0.44      0.24       107
        468       0.17      0.36      0.23       126
        469       0.04      0.09      0.05       114
        470       0.64      0.86      0.73       140
        471       0.16      0.44      0.24        79
        472       0.25      0.45      0.32       143
        473       0.32      0.48      0.39       158
        474       0.10      0.20      0.14       138
        475       0.06      0.29      0.10        59
        476       0.21      0.51      0.30        88
        477       0.43      0.70      0.53       176
        478       0.40      0.83      0.54        24
        479       0.06      0.21      0.10        92
        480       0.36      0.62      0.45       100
        481       0.25      0.46      0.33       103
        482       0.08      0.32      0.13        74
        483       0.37      0.70      0.48       105
        484       0.08      0.23      0.12        83
        485       0.02      0.07      0.03        82
        486       0.09      0.32      0.14        71
        487       0.18      0.28      0.22       120
        488       0.07      0.18      0.10       105
        489       0.23      0.53      0.32        87
        490       0.56      0.84      0.68        32
        491       0.01      0.04      0.02        69
        492       0.02      0.10      0.03        49
        493       0.04      0.12      0.06       117
        494       0.10      0.33      0.15        61
        495       0.77      0.83      0.80       344
        496       0.10      0.21      0.14        52
        497       0.23      0.43      0.30       137
        498       0.10      0.31      0.15        98
        499       0.07      0.34      0.12        79

   micro avg       0.29      0.48      0.36    173812
   macro avg       0.21      0.41      0.27    173812
weighted avg       0.35      0.48      0.39    173812
 samples avg       0.36      0.45      0.36    173812

Time taken to train the model : 0:06:22.726302
```

Out[8]:

```
['linear_SVM_with_more_title_weight_sgd_hinge_ovr.pkl']
```

5.5.1 Hyperparameter tuning 'alpha' to increase the performance, for SGD Classifier with Hinge Loss using GridSearchCV

In [10]:

```
#Perform Grid Search.
from sklearn.model_selection import GridSearchCV
start = datetime.now()

def get_GridSearchCV(x_train_multilabel, y_train):
    '''This function will determine the best hyperparameters using KFold CV and GridSearchCV, using 10 fold cross
validation. '''
    from sklearn.model_selection import KFold
    n_folds = 10
    cv_kfold = KFold(n_splits=n_folds).split(x_train_multilabel)
    alpha = np.logspace(-4,4,12)
    params  = {"estimator__alpha":alpha}
    base_estimator=OneVsRestClassifier(SGDClassifier(loss='hinge', penalty='l1'))
    rsearch_cv = GridSearchCV(estimator=base_estimator, param_grid=params, cv=cv_kfold, scoring='f1_micro', n_job
s=-1, verbose=1)
    rsearch_cv.fit(x_train_multilabel, y_train)
    return  rsearch_cv

#Get the best estimator after performing Grid Search Cross Validation.
gsearch_cv = get_GridSearchCV(x_train_multilabel,y_train)

print("Time taken to perform hyperparameter tuning: ",datetime.now()-start)
print("Best estimator: ",gsearch_cv.best_estimator_)
print("Best Cross Validation Score: ",gsearch_cv.best_score_)
```

```
Fitting 10 folds for each of 12 candidates, totalling 120 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done  34 tasks      | elapsed: 202.7min
[Parallel(n_jobs=-1)]: Done 120 out of 120 | elapsed: 671.7min finished

Time taken to perform hyperparameter tuning:  11:27:57.905896
Best estimator:  OneVsRestClassifier(estimator=SGDClassifier(alpha=0.0005336699231206312, average=Fa
lse, class_weight=None,
       early_stopping=False, epsilon=0.1, eta0=0.0, fit_intercept=True,
       l1_ratio=0.15, learning_rate='optimal', loss='hinge', max_iter=None,
       n_iter=None, n_iter_no_change=5, n_jobs=None, penalty='l1',
       power_t=0.5, random_state=None, shuffle=True, tol=None,
       validation_fraction=0.1, verbose=0, warm_start=False),
          n_jobs=None)
Best Cross Validation Score:  0.45956534873702376
```

5.5.2 Applying SGDClassifier with Hinge Loss with OneVsRest Classifier with the best estimator obtained from Grid Search.

In [11]:

```
start = datetime.now()

#Fit the dataset with the best estimator.
classifier = gsearch_cv.best_estimator_
classifier.fit(x_train_multilabel, y_train)
y_pred = classifier.predict(x_test_multilabel)

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

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

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

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

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

print (metrics.classification_report(y_test, y_pred))
print("Time taken to train the model :", datetime.now() - start)

import joblib
joblib.dump(classifier, 'lr_with_more_title_weight_lr_sdg_hinge_ovr_hyp_tuned.pkl')
```

```
Accuracy : 0.1622
Hamming loss  0.00349386
Micro-average quality numbers
```

Precision: 0.4965, Recall: 0.3630, Micro F1-measure: 0.4194
Macro-average quality numbers
Precision: 0.3466, Recall: 0.2873, Macro F1-measure: 0.2946

	precision	recall	f1-score	support
0	0.71	0.72	0.71	5519
1	0.52	0.25	0.34	8190
2	0.59	0.43	0.50	6529
3	0.56	0.53	0.55	3231
4	0.62	0.45	0.52	6430
5	0.56	0.47	0.51	2879
6	0.70	0.57	0.63	5086
7	0.76	0.59	0.66	4533
8	0.31	0.19	0.24	3000
9	0.61	0.58	0.59	2765
10	0.45	0.14	0.21	3051
11	0.61	0.40	0.48	3009
12	0.58	0.30	0.40	2630
13	0.37	0.24	0.29	1426
14	0.72	0.62	0.67	2548
15	0.57	0.19	0.29	2371
16	0.54	0.38	0.45	873
17	0.75	0.68	0.71	2151
18	0.46	0.27	0.34	2204
19	0.44	0.49	0.46	831
20	0.69	0.55	0.61	1860
21	0.03	0.00	0.00	2023
22	0.34	0.27	0.30	1513
23	0.67	0.61	0.64	1207
24	0.33	0.36	0.35	506
25	0.58	0.39	0.47	425
26	0.37	0.41	0.39	793
27	0.51	0.32	0.39	1291
28	0.66	0.39	0.49	1208
29	0.21	0.16	0.18	406
30	0.33	0.27	0.30	504
31	0.12	0.12	0.12	732
32	0.32	0.35	0.34	441
33	0.00	0.00	0.00	1645
34	0.57	0.30	0.40	1058
35	0.64	0.59	0.61	946
36	0.61	0.29	0.39	644
37	0.85	0.81	0.83	136
38	0.39	0.44	0.41	570
39	0.70	0.39	0.50	766
40	0.47	0.30	0.37	1132
41	0.16	0.28	0.20	174
42	0.50	0.58	0.54	210
43	0.50	0.50	0.50	433
44	0.47	0.61	0.53	626
45	0.57	0.39	0.46	852
46	0.46	0.52	0.49	534
47	0.00	0.00	0.00	350
48	0.50	0.53	0.52	496
49	0.68	0.69	0.69	785
50	0.13	0.20	0.16	475
51	0.09	0.19	0.12	305
52	0.12	0.01	0.02	251
53	0.43	0.48	0.45	914
54	0.35	0.01	0.02	728
55	0.08	0.05	0.06	258
56	0.18	0.20	0.19	821
57	0.19	0.02	0.04	541
58	0.69	0.34	0.45	748
59	0.82	0.74	0.78	724
60	0.13	0.00	0.01	660
61	0.79	0.26	0.39	235
62	0.81	0.77	0.79	718
63	0.73	0.65	0.68	468
64	0.49	0.49	0.49	191
65	0.00	0.00	0.00	429
66	0.00	0.00	0.00	415
67	0.59	0.59	0.59	274
68	0.71	0.60	0.65	510
69	0.52	0.53	0.53	466
70	0.14	0.17	0.15	305
71	0.30	0.21	0.25	247
72	0.56	0.56	0.56	401
73	0.83	0.85	0.84	86
74	0.31	0.39	0.35	120
75	0.64	0.78	0.70	129
76	0.00	0.00	0.00	473
77	0.20	0.40	0.26	143

78	0.70	0.61	0.65	347
79	0.43	0.32	0.37	479
80	0.37	0.21	0.27	279
81	0.67	0.20	0.31	461
82	0.08	0.06	0.07	298
83	0.58	0.60	0.59	396
84	0.28	0.43	0.34	184
85	0.53	0.17	0.25	573
86	0.54	0.07	0.12	325
87	0.29	0.36	0.32	273
88	0.28	0.33	0.30	135
89	0.15	0.20	0.17	232
90	0.47	0.02	0.04	409
91	0.52	0.30	0.38	420
92	0.60	0.63	0.62	408
93	0.43	0.55	0.48	241
94	0.00	0.00	0.00	211
95	0.00	0.00	0.00	277
96	0.07	0.01	0.02	410
97	0.73	0.19	0.31	501
98	0.61	0.69	0.65	136
99	0.40	0.36	0.38	239
100	0.00	0.00	0.00	324
101	0.83	0.69	0.75	277
102	0.79	0.74	0.76	613
103	0.14	0.30	0.19	157
104	0.10	0.12	0.11	295
105	0.45	0.39	0.42	334
106	0.34	0.09	0.14	335
107	0.56	0.59	0.57	389
108	0.27	0.07	0.11	251
109	0.43	0.53	0.47	317
110	0.00	0.00	0.00	187
111	0.17	0.19	0.18	140
112	0.07	0.27	0.11	154
113	0.46	0.35	0.39	332
114	0.22	0.19	0.20	323
115	0.29	0.28	0.29	344
116	0.52	0.58	0.55	370
117	0.35	0.30	0.32	313
118	0.71	0.79	0.74	874
119	0.17	0.22	0.19	293
120	0.02	0.03	0.02	200
121	0.63	0.52	0.57	463
122	0.19	0.21	0.20	119
123	0.02	0.00	0.01	256
124	0.63	0.81	0.71	195
125	0.38	0.27	0.31	138
126	0.75	0.47	0.58	376
127	0.03	0.01	0.01	122
128	0.05	0.00	0.01	252
129	0.00	0.00	0.00	144
130	0.16	0.17	0.17	150
131	0.00	0.00	0.00	210
132	0.34	0.33	0.33	361
133	0.78	0.66	0.72	453
134	0.65	0.84	0.73	124
135	0.00	0.00	0.00	91
136	0.59	0.17	0.27	128
137	0.26	0.40	0.32	218
138	0.77	0.14	0.23	243
139	0.13	0.30	0.18	149
140	0.65	0.44	0.53	318
141	0.19	0.19	0.19	159
142	0.55	0.39	0.46	274
143	0.76	0.82	0.79	362
144	0.23	0.37	0.29	118
145	0.33	0.45	0.38	164
146	0.44	0.37	0.40	461
147	0.58	0.53	0.56	159
148	0.26	0.07	0.11	166
149	0.63	0.51	0.56	346
150	0.22	0.11	0.15	350
151	0.60	0.49	0.54	55
152	0.61	0.54	0.57	387
153	0.21	0.18	0.19	150
154	0.34	0.17	0.22	281
155	0.14	0.17	0.15	202
156	0.47	0.70	0.56	130
157	0.15	0.20	0.17	245
158	0.74	0.61	0.67	177
159	0.40	0.38	0.39	130
160	0.35	0.32	0.33	336

161	0.76	0.69	0.72	220
162	0.00	0.00	0.00	229
163	0.82	0.52	0.63	316
164	0.64	0.28	0.39	283
165	0.20	0.32	0.25	197
166	0.14	0.27	0.18	101
167	0.00	0.00	0.00	231
168	0.18	0.29	0.22	370
169	0.19	0.21	0.20	258
170	0.11	0.11	0.11	101
171	0.11	0.19	0.14	89
172	0.27	0.41	0.33	193
173	0.42	0.28	0.34	309
174	0.21	0.23	0.22	172
175	0.72	0.89	0.80	95
176	0.63	0.65	0.64	346
177	0.85	0.39	0.54	322
178	0.50	0.54	0.52	232
179	0.08	0.10	0.09	125
180	0.39	0.32	0.35	145
181	0.11	0.18	0.13	77
182	0.09	0.15	0.11	182
183	0.37	0.35	0.36	257
184	0.00	0.00	0.00	216
185	0.00	0.00	0.00	242
186	0.13	0.21	0.16	165
187	0.47	0.64	0.54	263
188	0.00	0.00	0.00	174
189	0.25	0.32	0.28	136
190	0.80	0.56	0.66	202
191	0.18	0.24	0.21	134
192	0.48	0.43	0.45	230
193	0.20	0.30	0.24	90
194	0.41	0.57	0.48	185
195	0.12	0.01	0.02	156
196	0.07	0.07	0.07	160
197	0.04	0.00	0.01	266
198	0.20	0.12	0.15	284
199	0.04	0.03	0.03	145
200	0.83	0.78	0.80	212
201	0.21	0.30	0.25	317
202	0.55	0.64	0.60	427
203	0.33	0.01	0.02	232
204	0.23	0.25	0.24	217
205	0.38	0.40	0.39	527
206	0.22	0.02	0.03	124
207	0.14	0.21	0.17	103
208	0.55	0.57	0.56	287
209	0.11	0.15	0.13	193
210	0.32	0.42	0.36	220
211	0.63	0.14	0.22	140
212	0.00	0.00	0.00	161
213	0.14	0.29	0.19	72
214	0.58	0.49	0.53	396
215	0.48	0.42	0.45	134
216	0.00	0.00	0.00	400
217	0.28	0.33	0.31	75
218	0.70	0.71	0.70	219
219	0.47	0.47	0.47	210
220	0.83	0.46	0.59	298
221	0.93	0.57	0.70	266
222	0.69	0.39	0.50	290
223	0.00	0.00	0.00	128
224	0.50	0.47	0.48	159
225	0.20	0.26	0.23	164
226	0.37	0.42	0.40	144
227	0.35	0.46	0.40	276
228	0.00	0.00	0.00	235
229	0.00	0.00	0.00	216
230	0.21	0.18	0.19	228
231	0.52	0.50	0.51	64
232	0.00	0.00	0.00	103
233	0.49	0.32	0.39	216
234	0.37	0.06	0.10	116
235	0.51	0.58	0.55	77
236	0.71	0.70	0.71	67
237	0.46	0.07	0.13	218
238	0.06	0.12	0.08	139
239	0.00	0.00	0.00	94
240	0.44	0.26	0.33	77
241	0.29	0.10	0.14	167
242	0.34	0.35	0.35	86
243	0.26	0.16	0.20	58

244	0.31	0.31	0.31	269
245	0.00	0.00	0.00	112
246	0.86	0.80	0.83	255
247	0.18	0.28	0.22	58
248	0.00	0.00	0.00	81
249	0.00	0.00	0.00	131
250	0.28	0.32	0.30	93
251	0.29	0.30	0.30	154
252	0.00	0.00	0.00	129
253	0.35	0.34	0.34	83
254	0.21	0.10	0.14	191
255	0.00	0.00	0.00	219
256	0.15	0.02	0.03	130
257	0.33	0.26	0.29	93
258	0.62	0.55	0.58	217
259	0.08	0.11	0.09	141
260	0.62	0.28	0.38	143
261	0.27	0.22	0.24	219
262	0.31	0.38	0.34	107
263	0.27	0.26	0.26	236
264	0.25	0.24	0.24	119
265	0.20	0.35	0.25	72
266	0.00	0.00	0.00	70
267	0.26	0.17	0.21	107
268	0.54	0.52	0.53	169
269	0.16	0.22	0.19	129
270	0.58	0.53	0.55	159
271	0.86	0.30	0.45	190
272	0.26	0.19	0.22	248
273	0.79	0.69	0.74	264
274	0.66	0.66	0.66	105
275	0.00	0.00	0.00	104
276	0.00	0.00	0.00	115
277	0.77	0.70	0.73	170
278	0.47	0.44	0.46	145
279	0.82	0.40	0.54	230
280	0.31	0.33	0.32	80
281	0.54	0.61	0.58	217
282	0.70	0.64	0.67	175
283	0.33	0.04	0.07	269
284	0.43	0.42	0.42	74
285	0.57	0.60	0.58	206
286	0.82	0.70	0.75	227
287	0.60	0.41	0.49	130
288	0.23	0.02	0.04	129
289	0.00	0.00	0.00	80
290	0.00	0.00	0.00	99
291	0.52	0.36	0.42	208
292	0.11	0.24	0.15	67
293	0.48	0.44	0.46	109
294	0.31	0.08	0.13	140
295	0.13	0.21	0.16	241
296	0.00	0.00	0.00	72
297	0.12	0.20	0.15	107
298	0.50	0.28	0.36	61
299	0.89	0.31	0.46	77
300	0.00	0.00	0.00	111
301	0.00	0.00	0.00	126
302	0.00	0.00	0.00	73
303	0.36	0.50	0.42	176
304	0.92	0.55	0.69	230
305	0.87	0.74	0.80	156
306	0.31	0.31	0.31	146
307	0.06	0.09	0.07	98
308	0.00	0.00	0.00	78
309	0.34	0.17	0.23	94
310	0.28	0.35	0.31	162
311	0.71	0.62	0.66	116
312	0.42	0.44	0.43	57
313	1.00	0.05	0.09	65
314	0.35	0.42	0.38	138
315	1.00	0.01	0.01	195
316	0.38	0.52	0.44	69
317	0.24	0.07	0.11	134
318	0.41	0.18	0.25	148
319	0.40	0.25	0.31	161
320	0.08	0.17	0.11	104
321	0.55	0.62	0.58	156
322	0.33	0.38	0.35	134
323	0.39	0.31	0.34	232
324	0.15	0.24	0.18	92
325	0.17	0.25	0.20	197
326	0.00	0.00	0.00	126

327	0.18	0.04	0.07	115
328	0.68	0.55	0.60	198
329	0.32	0.38	0.35	125
330	0.57	0.21	0.31	81
331	0.00	0.00	0.00	94
332	0.50	0.02	0.03	56
333	0.00	0.00	0.00	260
334	0.25	0.02	0.03	60
335	0.18	0.25	0.21	110
336	0.35	0.48	0.40	71
337	0.15	0.12	0.14	66
338	0.26	0.29	0.27	150
339	0.00	0.00	0.00	54
340	0.69	0.67	0.68	195
341	0.07	0.09	0.08	79
342	0.10	0.24	0.14	38
343	0.33	0.37	0.35	43
344	0.29	0.07	0.12	68
345	0.42	0.41	0.42	73
346	0.15	0.10	0.12	116
347	0.37	0.31	0.33	111
348	0.00	0.00	0.00	63
349	0.64	0.79	0.70	104
350	0.55	0.27	0.36	44
351	0.14	0.03	0.04	40
352	0.89	0.40	0.55	136
353	0.29	0.35	0.32	54
354	0.11	0.05	0.07	134
355	0.25	0.23	0.24	120
356	0.30	0.34	0.32	228
357	0.46	0.30	0.36	269
358	0.39	0.53	0.44	80
359	0.47	0.59	0.52	140
360	0.09	0.14	0.11	125
361	0.88	0.48	0.62	169
362	0.10	0.07	0.08	56
363	0.75	0.75	0.75	154
364	0.00	0.00	0.00	58
365	0.15	0.31	0.20	71
366	0.89	0.74	0.81	54
367	0.39	0.13	0.19	116
368	0.00	0.00	0.00	54
369	0.00	0.00	0.00	71
370	0.00	0.00	0.00	61
371	0.19	0.06	0.09	71
372	0.49	0.65	0.56	52
373	0.66	0.27	0.39	150
374	0.29	0.41	0.34	93
375	0.00	0.00	0.00	67
376	0.00	0.00	0.00	76
377	0.63	0.11	0.19	106
378	0.06	0.08	0.07	86
379	0.07	0.14	0.10	14
380	0.89	0.48	0.63	122
381	0.00	0.00	0.00	104
382	0.24	0.18	0.21	66
383	0.33	0.35	0.34	110
384	0.00	0.00	0.00	155
385	0.38	0.06	0.10	50
386	0.17	0.25	0.20	64
387	0.00	0.00	0.00	93
388	0.35	0.56	0.43	102
389	0.00	0.00	0.00	108
390	0.85	0.63	0.72	178
391	0.42	0.37	0.40	115
392	0.67	0.52	0.59	42
393	0.00	0.00	0.00	134
394	0.00	0.00	0.00	112
395	0.30	0.20	0.24	176
396	0.29	0.04	0.07	125
397	0.50	0.42	0.45	224
398	0.67	0.57	0.62	63
399	0.00	0.00	0.00	59
400	0.27	0.49	0.35	63
401	0.00	0.00	0.00	98
402	0.00	0.00	0.00	162
403	0.21	0.24	0.22	83
404	0.65	0.68	0.67	19
405	0.00	0.00	0.00	92
406	0.40	0.39	0.40	41
407	0.17	0.26	0.21	43
408	0.00	0.00	0.00	160
409	0.17	0.26	0.21	50

410	0.00	0.00	0.00	19
411	0.19	0.13	0.16	175
412	0.00	0.00	0.00	72
413	0.39	0.12	0.18	95
414	0.00	0.00	0.00	97
415	0.29	0.12	0.17	48
416	0.27	0.43	0.33	83
417	0.50	0.03	0.05	40
418	0.00	0.00	0.00	91
419	0.32	0.43	0.37	90
420	0.27	0.35	0.30	37
421	0.06	0.09	0.07	66
422	0.37	0.42	0.40	73
423	0.23	0.32	0.26	56
424	0.58	0.79	0.67	33
425	0.00	0.00	0.00	76
426	0.00	0.00	0.00	81
427	0.81	0.67	0.73	150
428	0.85	0.76	0.80	29
429	0.99	0.27	0.42	389
430	0.47	0.37	0.41	167
431	0.33	0.12	0.18	123
432	0.24	0.31	0.27	39
433	0.30	0.32	0.31	82
434	0.91	0.64	0.75	66
435	0.53	0.48	0.51	93
436	0.29	0.20	0.23	87
437	0.10	0.15	0.12	86
438	0.36	0.49	0.42	104
439	0.29	0.08	0.12	100
440	0.00	0.00	0.00	141
441	0.18	0.26	0.22	110
442	0.13	0.04	0.06	123
443	0.33	0.32	0.33	71
444	0.00	0.00	0.00	109
445	0.18	0.27	0.21	48
446	0.45	0.42	0.44	76
447	0.06	0.24	0.10	38
448	0.49	0.54	0.52	81
449	0.37	0.23	0.29	132
450	0.19	0.27	0.22	81
451	0.50	0.39	0.44	76
452	0.00	0.00	0.00	44
453	0.03	0.02	0.03	44
454	0.55	0.54	0.55	70
455	0.22	0.05	0.07	155
456	0.16	0.28	0.20	43
457	0.27	0.15	0.19	72
458	0.16	0.13	0.14	62
459	0.21	0.04	0.07	69
460	0.00	0.00	0.00	119
461	0.43	0.38	0.40	79
462	0.21	0.15	0.17	47
463	0.00	0.00	0.00	104
464	0.39	0.49	0.43	106
465	1.00	0.02	0.03	64
466	0.27	0.29	0.28	173
467	0.50	0.57	0.54	107
468	1.00	0.04	0.08	126
469	0.00	0.00	0.00	114
470	0.91	0.77	0.83	140
471	0.00	0.00	0.00	79
472	0.34	0.50	0.41	143
473	0.65	0.09	0.17	158
474	0.21	0.10	0.14	138
475	0.00	0.00	0.00	59
476	0.31	0.32	0.32	88
477	0.69	0.56	0.62	176
478	0.43	0.67	0.52	24
479	0.24	0.15	0.19	92
480	0.60	0.61	0.60	100
481	0.32	0.12	0.17	103
482	0.20	0.26	0.23	74
483	0.69	0.57	0.62	105
484	0.00	0.00	0.00	83
485	0.00	0.00	0.00	82
486	0.00	0.00	0.00	71
487	0.27	0.32	0.29	120
488	0.57	0.04	0.07	105
489	0.40	0.46	0.43	87
490	1.00	0.78	0.88	32
491	0.00	0.00	0.00	69
492	0.00	0.00	0.00	49

```
         493       0.00      0.00      0.00       117
         494       0.39      0.20      0.26        61
         495       0.00      0.00      0.00       344
         496       0.00      0.00      0.00        52
         497       0.26      0.25      0.25       137
         498       0.00      0.00      0.00        98
         499       0.44      0.20      0.28        79

   micro avg       0.50      0.36      0.42    173812
   macro avg       0.35      0.29      0.29    173812
weighted avg       0.47      0.36      0.40    173812
 samples avg       0.39      0.34      0.33    173812

Time taken to train the model : 0:16:18.931357
```

Out[11]:

```
['lr_with_more_title_weight_lr_sdg_hinge_ovr_hyp_tuned.pkl']
```

5.6 Applying Logistic Regression on BOW vectorizer - 1,2 - Grams + Hyperparameter tuning

In [15]:

```
from datetime import datetime

#Taking 25000 most important features and number of n_grams = (1,2)
start = datetime.now()
vectorizer = CountVectorizer(min_df=0.00009, max_features=25000, analyzer='word', tokenizer = lambda x: x.split()
, ngram_range=(1,2))
x_train_multilabel = vectorizer.fit_transform(x_train['question'])
x_test_multilabel = vectorizer.transform(x_test['question'])
print("Time taken to featurize the class labels using BOW representation :", datetime.now() - start)

#Sorting indices to get rid of Value Error: WRITEBACKIFCOPY base is read-only
x_train_multilabel.sort_indices()
x_test_multilabel.sort_indices()

print("Dimensions of train and test data:")
print("x_train:",x_train_multilabel.shape, "y_train :",y_train.shape)
print("x_test:",x_test_multilabel.shape, "y_test :",y_test.shape)

#Save the data for later use.
import pickle
with open('x_train_multilabel_bigrm.pkl', 'wb') as file:
    pickle.dump(x_train_multilabel, file)

with open('y_train_bigrm.pkl', 'wb') as file:
    pickle.dump(y_train, file)

with open('x_test_multilabel_bigrm.pkl', 'wb') as file:
    pickle.dump(x_test_multilabel, file)

with open('y_test_bigrm.pkl', 'wb') as file:
    pickle.dump(y_test, file)
```

```
Time taken to featurize the class labels using BOW representation : 0:01:31.044764
Dimensions of train and test data:
x_train: (400000, 25000) y_train : (400000, 500)
x_test: (100000, 25000) y_test : (100000, 500)
```

In [3]:

```python
import pickle

with open('x_train_multilabel_bigrm.pkl', 'rb') as file:
    x_train_multilabel = pickle.load(file)

with open('y_train_bigrm.pkl', 'rb') as file:
    y_train = pickle.load(file)

with open('x_test_multilabel_bigrm.pkl', 'rb') as file:
    x_test_multilabel = pickle.load(file)

with open('y_test_bigrm.pkl', 'rb') as file:
    y_test = pickle.load(file)

print("Dimensions of train and test data:")
print("x_train:",x_train_multilabel.shape, "y_train :",y_train.shape)
print("x_test:",x_test_multilabel.shape, "y_test :",y_test.shape)
```

```
Dimensions of train and test data:
x_train: (400000, 25000) y_train : (400000, 500)
x_test: (100000, 25000) y_test : (100000, 500)
```

In [5]:

```python
#Function to determine the best estimator using Hyperparameter tuning by using simple for loops. Not using GridSe
arch for improving time complexity. Not using K-Fold cross validation. Very basic for loop to determine best para
ms
#Note: We can't use probability scores using SGDClassifiers
import warnings
warnings.filterwarnings("ignore")

models = []
f1_scores=[]
def get_best_estimate_lr(x_train_multilabel, y_train, x_test_multilabel, y_test):
    c_vals = [1000,100,10,1,0.1,0.01]
    for C in tqdm(c_vals):
        score = []
        classifier=OneVsRestClassifier(LogisticRegression(penalty='l2', C=1.0, multi_class='ovr', solver='lbfgs',
random_state=0), n_jobs=-1)
        classifier.fit(x_train_multilabel, y_train)
        models.append(classifier)
        f1_scores.append(f1_score(y_test, classifier.predict(x_test_multilabel), average='micro')) #predictions:
f1_score(y_test, classifier.predict(x_test_multilabel), average='micro')

    max_score = max(f1_scores)
    best_estimator = models[f1_scores.index(max_score)]

    print("The best estimator obtained using Hyperparameter tuning: ",best_estimator)
    print("The best Micro Average F1-Score obtained using Hyperparameter tuning: ",max_score)

    return(best_estimator)

best_estimator=get_best_estimate_lr(x_train_multilabel, y_train, x_test_multilabel, y_test)
```

```
100%|██████████| 6/6 [4:10:02<00:00, 2621.96s/it]

The best estimator obtained using Hyperparameter tuning:  OneVsRestClassifier(estimator=LogisticRegr
ession(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr',
          n_jobs=None, penalty='l2', random_state=0, solver='lbfgs',
          tol=0.0001, verbose=0, warm_start=False),
          n_jobs=-1)
The best Micro Average F1-Score obtained using Hyperparameter tuning:  0.46704424197652467
```

In [6]:

```
#Train the model with best estimator
start = datetime.now()
classifier = best_estimator
classifier.fit(x_train_multilabel, y_train)
predictions = classifier.predict(x_test_multilabel)

print("\nAccuracy :",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')

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

print (metrics.classification_report(y_test, predictions))
print("Time taken to train the model :", datetime.now() - start)

import joblib
joblib.dump(classifier, 'lr_with_more_title_weight_lr_bow_2grm_hyp_tuned.pkl')
```

```
Accuracy : 0.21343
Hamming loss  0.00311664
Micro-average quality numbers
Precision: 0.5758, Recall: 0.3928, Micro F1-measure: 0.4670
Macro-average quality numbers
Precision: 0.4568, Recall: 0.3173, Macro F1-measure: 0.3706

/root/anaconda3/lib/python3.7/site-packages/sklearn/metrics/classification.py:1143: UndefinedMetricW
arning: Precision and F-score are ill-defined and being set to 0.0 in samples with no predicted labe
ls.
  'precision', 'predicted', average, warn_for)
/root/anaconda3/lib/python3.7/site-packages/sklearn/metrics/classification.py:1145: UndefinedMetricW
arning: Recall and F-score are ill-defined and being set to 0.0 in samples with no true labels.
  'recall', 'true', average, warn_for)

              precision    recall  f1-score   support

           0       0.89      0.74      0.80      5519
           1       0.58      0.38      0.46      8190
           2       0.73      0.44      0.55      6529
           3       0.68      0.52      0.59      3231
           4       0.68      0.48      0.56      6430
           5       0.67      0.40      0.50      2879
           6       0.79      0.56      0.65      5086
           7       0.77      0.61      0.68      4533
           8       0.42      0.17      0.24      3000
           9       0.68      0.61      0.64      2765
          10       0.43      0.27      0.34      3051
          11       0.57      0.44      0.49      3009
          12       0.48      0.34      0.40      2630
          13       0.56      0.37      0.44      1426
          14       0.78      0.58      0.66      2548
          15       0.47      0.31      0.37      2371
          16       0.60      0.29      0.39       873
          17       0.78      0.65      0.71      2151
          18       0.42      0.29      0.35      2204
          19       0.52      0.43      0.47       831
          20       0.67      0.44      0.54      1860
          21       0.25      0.16      0.20      2023
          22       0.41      0.29      0.34      1513
          23       0.75      0.57      0.65      1207
          24       0.45      0.31      0.37       506
          25       0.49      0.35      0.41       425
          26       0.58      0.41      0.48       793
          27       0.51      0.40      0.45      1291
          28       0.59      0.40      0.48      1208
          29       0.29      0.17      0.22       406
          30       0.41      0.23      0.29       504
          31       0.19      0.12      0.15       732
          32       0.38      0.32      0.34       441
          33       0.50      0.35      0.41      1645
          34       0.44      0.28      0.34      1058
          35       0.71      0.56      0.63       946
```

36	0.47	0.29	0.36	644
37	0.87	0.69	0.77	136
38	0.51	0.38	0.44	570
39	0.60	0.33	0.42	766
40	0.52	0.39	0.44	1132
41	0.28	0.26	0.27	174
42	0.64	0.51	0.57	210
43	0.62	0.44	0.52	433
44	0.59	0.45	0.51	626
45	0.52	0.37	0.43	852
46	0.60	0.44	0.51	534
47	0.29	0.21	0.24	350
48	0.64	0.48	0.55	496
49	0.76	0.62	0.68	785
50	0.23	0.13	0.17	475
51	0.25	0.21	0.23	305
52	0.21	0.09	0.13	251
53	0.52	0.40	0.45	914
54	0.39	0.24	0.29	728
55	0.16	0.08	0.11	258
56	0.33	0.27	0.29	821
57	0.31	0.18	0.22	541
58	0.55	0.35	0.43	748
59	0.90	0.66	0.76	724
60	0.33	0.17	0.23	660
61	0.41	0.24	0.30	235
62	0.87	0.69	0.77	718
63	0.77	0.67	0.72	468
64	0.44	0.28	0.34	191
65	0.30	0.19	0.23	429
66	0.18	0.12	0.14	415
67	0.67	0.47	0.56	274
68	0.74	0.52	0.61	510
69	0.59	0.45	0.51	466
70	0.23	0.16	0.19	305
71	0.30	0.20	0.24	247
72	0.72	0.50	0.59	401
73	0.82	0.79	0.80	86
74	0.56	0.40	0.47	120
75	0.88	0.71	0.79	129
76	0.14	0.06	0.09	473
77	0.39	0.31	0.35	143
78	0.68	0.45	0.54	347
79	0.45	0.24	0.31	479
80	0.44	0.36	0.40	279
81	0.46	0.24	0.32	461
82	0.14	0.08	0.10	298
83	0.72	0.51	0.59	396
84	0.41	0.35	0.38	184
85	0.42	0.27	0.33	573
86	0.23	0.12	0.16	325
87	0.49	0.32	0.39	273
88	0.44	0.36	0.40	135
89	0.24	0.16	0.19	232
90	0.48	0.38	0.42	409
91	0.48	0.31	0.38	420
92	0.68	0.53	0.59	408
93	0.50	0.47	0.49	241
94	0.26	0.10	0.14	211
95	0.29	0.17	0.22	277
96	0.21	0.12	0.15	410
97	0.75	0.42	0.54	501
98	0.65	0.54	0.59	136
99	0.46	0.35	0.40	239
100	0.31	0.19	0.23	324
101	0.89	0.67	0.76	277
102	0.88	0.74	0.80	613
103	0.33	0.17	0.23	157
104	0.20	0.14	0.16	295
105	0.68	0.46	0.55	334
106	0.65	0.31	0.42	335
107	0.67	0.54	0.60	389
108	0.50	0.30	0.38	251
109	0.55	0.44	0.49	317
110	0.33	0.11	0.16	187
111	0.35	0.14	0.20	140
112	0.53	0.38	0.44	154
113	0.49	0.26	0.34	332
114	0.43	0.30	0.36	323
115	0.43	0.31	0.36	344
116	0.69	0.53	0.60	370
117	0.39	0.24	0.29	313
118	0.76	0.71	0.73	874

119	0.34	0.24	0.28	293
120	0.16	0.12	0.13	200
121	0.66	0.50	0.57	463
122	0.34	0.17	0.22	119
123	0.14	0.04	0.06	256
124	0.85	0.68	0.76	195
125	0.30	0.14	0.20	138
126	0.71	0.50	0.58	376
127	0.12	0.07	0.08	122
128	0.13	0.07	0.09	252
129	0.40	0.30	0.34	144
130	0.32	0.21	0.25	150
131	0.18	0.08	0.11	210
132	0.53	0.30	0.39	361
133	0.87	0.61	0.72	453
134	0.82	0.73	0.77	124
135	0.15	0.12	0.14	91
136	0.46	0.38	0.42	128
137	0.48	0.40	0.43	218
138	0.38	0.21	0.27	243
139	0.34	0.20	0.25	149
140	0.72	0.47	0.57	318
141	0.20	0.10	0.13	159
142	0.58	0.36	0.44	274
143	0.81	0.77	0.79	362
144	0.41	0.24	0.30	118
145	0.44	0.38	0.41	164
146	0.52	0.37	0.43	461
147	0.70	0.41	0.52	159
148	0.31	0.19	0.23	166
149	0.88	0.50	0.64	346
150	0.44	0.19	0.26	350
151	0.90	0.67	0.77	55
152	0.70	0.49	0.57	387
153	0.36	0.33	0.35	150
154	0.29	0.14	0.19	281
155	0.25	0.18	0.21	202
156	0.75	0.65	0.70	130
157	0.24	0.10	0.14	245
158	0.89	0.65	0.75	177
159	0.43	0.41	0.42	130
160	0.43	0.24	0.31	336
161	0.79	0.62	0.69	220
162	0.19	0.11	0.14	229
163	0.82	0.42	0.56	316
164	0.66	0.42	0.52	283
165	0.48	0.31	0.38	197
166	0.57	0.56	0.57	101
167	0.32	0.20	0.25	231
168	0.42	0.33	0.37	370
169	0.35	0.24	0.28	258
170	0.30	0.18	0.22	101
171	0.33	0.27	0.30	89
172	0.45	0.33	0.38	193
173	0.46	0.32	0.38	309
174	0.29	0.15	0.19	172
175	0.78	0.75	0.76	95
176	0.89	0.59	0.71	346
177	0.81	0.54	0.65	322
178	0.53	0.46	0.49	232
179	0.21	0.11	0.15	125
180	0.52	0.40	0.45	145
181	0.33	0.22	0.27	77
182	0.16	0.09	0.12	182
183	0.51	0.34	0.41	257
184	0.19	0.11	0.14	216
185	0.28	0.14	0.19	242
186	0.34	0.22	0.27	165
187	0.68	0.56	0.62	263
188	0.20	0.11	0.14	174
189	0.64	0.43	0.51	136
190	0.81	0.52	0.64	202
191	0.32	0.20	0.25	134
192	0.57	0.39	0.46	230
193	0.29	0.22	0.25	90
194	0.60	0.51	0.55	185
195	0.20	0.08	0.12	156
196	0.15	0.10	0.12	160
197	0.27	0.14	0.18	266
198	0.36	0.17	0.23	284
199	0.17	0.08	0.11	145
200	0.88	0.72	0.79	212
201	0.45	0.26	0.33	317

202	0.70	0.60	0.64	427
203	0.25	0.15	0.18	232
204	0.44	0.31	0.37	217
205	0.50	0.46	0.48	527
206	0.19	0.09	0.12	124
207	0.41	0.34	0.37	103
208	0.73	0.48	0.58	287
209	0.22	0.15	0.18	193
210	0.57	0.41	0.48	220
211	0.47	0.20	0.28	140
212	0.14	0.08	0.10	161
213	0.47	0.54	0.50	72
214	0.63	0.40	0.49	396
215	0.62	0.37	0.47	134
216	0.43	0.25	0.31	400
217	0.29	0.23	0.25	75
218	0.91	0.71	0.79	219
219	0.61	0.41	0.49	210
220	0.88	0.63	0.73	298
221	0.91	0.68	0.78	266
222	0.68	0.40	0.50	290
223	0.12	0.05	0.07	128
224	0.69	0.48	0.57	159
225	0.42	0.32	0.36	164
226	0.46	0.31	0.37	144
227	0.53	0.42	0.47	276
228	0.11	0.05	0.07	235
229	0.18	0.09	0.12	216
230	0.33	0.21	0.26	228
231	0.59	0.56	0.58	64
232	0.21	0.14	0.17	103
233	0.62	0.37	0.46	216
234	0.40	0.20	0.26	116
235	0.51	0.32	0.40	77
236	0.90	0.70	0.79	67
237	0.35	0.17	0.22	218
238	0.24	0.18	0.20	139
239	0.23	0.07	0.11	94
240	0.49	0.30	0.37	77
241	0.25	0.12	0.16	167
242	0.69	0.38	0.49	86
243	0.24	0.22	0.23	58
244	0.50	0.42	0.46	269
245	0.15	0.09	0.11	112
246	0.90	0.76	0.82	255
247	0.26	0.24	0.25	58
248	0.10	0.05	0.07	81
249	0.08	0.03	0.04	131
250	0.38	0.27	0.32	93
251	0.59	0.31	0.41	154
252	0.15	0.08	0.10	129
253	0.46	0.37	0.41	83
254	0.24	0.13	0.17	191
255	0.12	0.07	0.09	219
256	0.11	0.05	0.07	130
257	0.40	0.25	0.31	93
258	0.60	0.46	0.52	217
259	0.29	0.19	0.23	141
260	0.79	0.24	0.37	143
261	0.41	0.18	0.25	219
262	0.53	0.30	0.38	107
263	0.43	0.26	0.33	236
264	0.30	0.24	0.26	119
265	0.46	0.24	0.31	72
266	0.17	0.09	0.11	70
267	0.34	0.21	0.26	107
268	0.59	0.41	0.49	169
269	0.33	0.21	0.26	129
270	0.72	0.56	0.63	159
271	0.75	0.47	0.58	190
272	0.42	0.29	0.34	248
273	0.85	0.73	0.78	264
274	0.84	0.69	0.75	105
275	0.25	0.12	0.17	104
276	0.06	0.03	0.04	115
277	0.76	0.56	0.65	170
278	0.65	0.41	0.50	145
279	0.87	0.70	0.77	230
280	0.58	0.35	0.44	80
281	0.66	0.52	0.58	217
282	0.68	0.47	0.56	175
283	0.26	0.17	0.21	269
284	0.57	0.36	0.45	74

285	0.73	0.47	0.57	206
286	0.83	0.69	0.75	227
287	0.62	0.38	0.48	130
288	0.24	0.09	0.13	129
289	0.14	0.10	0.12	80
290	0.23	0.16	0.19	99
291	0.54	0.37	0.44	208
292	0.28	0.15	0.19	67
293	0.80	0.55	0.65	109
294	0.35	0.28	0.31	140
295	0.22	0.12	0.16	241
296	0.35	0.15	0.21	72
297	0.25	0.17	0.20	107
298	0.62	0.52	0.57	61
299	0.75	0.53	0.62	77
300	0.19	0.15	0.17	111
301	0.07	0.02	0.03	126
302	0.17	0.11	0.13	73
303	0.53	0.35	0.42	176
304	0.89	0.79	0.84	230
305	0.88	0.67	0.76	156
306	0.45	0.31	0.37	146
307	0.30	0.11	0.16	98
308	0.06	0.03	0.04	78
309	0.48	0.14	0.21	94
310	0.62	0.36	0.45	162
311	0.74	0.48	0.58	116
312	0.50	0.32	0.39	57
313	0.24	0.09	0.13	65
314	0.40	0.30	0.34	138
315	0.54	0.30	0.38	195
316	0.40	0.28	0.32	69
317	0.28	0.22	0.25	134
318	0.48	0.36	0.41	148
319	0.74	0.43	0.55	161
320	0.23	0.22	0.22	104
321	0.73	0.56	0.64	156
322	0.55	0.49	0.52	134
323	0.51	0.42	0.46	232
324	0.24	0.16	0.19	92
325	0.38	0.26	0.31	197
326	0.14	0.09	0.11	126
327	0.14	0.05	0.08	115
328	0.94	0.68	0.79	198
329	0.45	0.30	0.36	125
330	0.55	0.28	0.37	81
331	0.33	0.13	0.18	94
332	0.29	0.21	0.25	56
333	0.21	0.10	0.13	260
334	0.17	0.15	0.16	60
335	0.34	0.13	0.19	110
336	0.60	0.49	0.54	71
337	0.17	0.09	0.12	66
338	0.45	0.40	0.42	150
339	0.14	0.07	0.10	54
340	0.78	0.55	0.65	195
341	0.72	0.58	0.64	79
342	0.37	0.47	0.41	38
343	0.52	0.28	0.36	43
344	0.38	0.28	0.32	68
345	0.70	0.36	0.47	73
346	0.09	0.05	0.07	116
347	0.69	0.46	0.55	111
348	0.34	0.22	0.27	63
349	0.86	0.69	0.77	104
350	0.59	0.52	0.55	44
351	0.26	0.30	0.28	40
352	0.80	0.50	0.62	136
353	0.41	0.17	0.24	54
354	0.20	0.11	0.14	134
355	0.52	0.42	0.46	120
356	0.47	0.30	0.37	228
357	0.58	0.41	0.48	269
358	0.60	0.31	0.41	80
359	0.78	0.59	0.67	140
360	0.30	0.23	0.26	125
361	0.87	0.64	0.74	169
362	0.09	0.07	0.08	56
363	0.87	0.71	0.78	154
364	0.33	0.26	0.29	58
365	0.32	0.17	0.22	71
366	0.90	0.65	0.75	54
367	0.18	0.13	0.15	116

368	0.20	0.17	0.18	54
369	0.12	0.07	0.09	71
370	0.17	0.07	0.10	61
371	0.23	0.08	0.12	71
372	0.54	0.40	0.46	52
373	0.59	0.39	0.47	150
374	0.29	0.19	0.23	93
375	0.21	0.12	0.15	67
376	0.04	0.01	0.02	76
377	0.46	0.35	0.40	106
378	0.16	0.05	0.07	86
379	0.14	0.14	0.14	14
380	0.80	0.45	0.58	122
381	0.13	0.07	0.09	104
382	0.18	0.11	0.13	66
383	0.49	0.36	0.42	110
384	0.14	0.05	0.07	155
385	0.46	0.36	0.40	50
386	0.26	0.14	0.18	64
387	0.27	0.13	0.17	93
388	0.48	0.29	0.36	102
389	0.14	0.07	0.10	108
390	0.92	0.65	0.76	178
391	0.31	0.17	0.22	115
392	0.67	0.43	0.52	42
393	0.04	0.01	0.01	134
394	0.19	0.11	0.14	112
395	0.36	0.30	0.33	176
396	0.31	0.18	0.23	125
397	0.66	0.44	0.53	224
398	0.75	0.60	0.67	63
399	0.12	0.07	0.09	59
400	0.40	0.35	0.37	63
401	0.34	0.26	0.29	98
402	0.40	0.25	0.31	162
403	0.22	0.13	0.17	83
404	0.58	0.74	0.65	19
405	0.20	0.14	0.16	92
406	0.60	0.37	0.45	41
407	0.45	0.33	0.38	43
408	0.67	0.42	0.52	160
409	0.25	0.16	0.20	50
410	0.00	0.00	0.00	19
411	0.27	0.19	0.22	175
412	0.25	0.14	0.18	72
413	0.28	0.13	0.17	95
414	0.31	0.16	0.21	97
415	0.17	0.12	0.14	48
416	0.48	0.35	0.41	83
417	0.18	0.07	0.11	40
418	0.33	0.15	0.21	91
419	0.53	0.38	0.44	90
420	0.32	0.27	0.29	37
421	0.11	0.06	0.08	66
422	0.50	0.36	0.42	73
423	0.42	0.25	0.31	56
424	0.90	0.79	0.84	33
425	0.17	0.07	0.10	76
426	0.08	0.02	0.04	81
427	0.95	0.69	0.80	150
428	1.00	0.69	0.82	29
429	0.99	0.86	0.92	389
430	0.57	0.41	0.48	167
431	0.42	0.19	0.26	123
432	0.24	0.13	0.17	39
433	0.34	0.30	0.32	82
434	0.96	0.67	0.79	66
435	0.53	0.43	0.47	93
436	0.49	0.31	0.38	87
437	0.14	0.07	0.09	86
438	0.68	0.45	0.54	104
439	0.40	0.19	0.26	100
440	0.16	0.05	0.08	141
441	0.38	0.30	0.34	110
442	0.24	0.18	0.21	123
443	0.27	0.15	0.20	71
444	0.26	0.09	0.14	109
445	0.41	0.29	0.34	48
446	0.36	0.26	0.31	76
447	0.22	0.21	0.22	38
448	0.61	0.52	0.56	81
449	0.45	0.25	0.32	132
450	0.42	0.32	0.36	81

```
         451       0.85      0.38      0.53        76
         452       0.11      0.07      0.08        44
         453       0.17      0.05      0.07        44
         454       0.75      0.56      0.64        70
         455       0.32      0.27      0.29       155
         456       0.40      0.23      0.29        43
         457       0.39      0.32      0.35        72
         458       0.26      0.18      0.21        62
         459       0.43      0.30      0.36        69
         460       0.12      0.08      0.09       119
         461       0.62      0.33      0.43        79
         462       0.31      0.21      0.25        47
         463       0.31      0.19      0.24       104
         464       0.62      0.37      0.46       106
         465       0.42      0.30      0.35        64
         466       0.42      0.29      0.35       173
         467       0.59      0.38      0.46       107
         468       0.39      0.29      0.33       126
         469       0.18      0.05      0.08       114
         470       0.92      0.79      0.85       140
         471       0.66      0.39      0.49        79
         472       0.38      0.38      0.38       143
         473       0.63      0.34      0.44       158
         474       0.27      0.09      0.14       138
         475       0.18      0.14      0.16        59
         476       0.64      0.41      0.50        88
         477       0.76      0.63      0.69       176
         478       0.79      0.79      0.79        24
         479       0.18      0.13      0.15        92
         480       0.72      0.52      0.60       100
         481       0.38      0.32      0.35       103
         482       0.25      0.18      0.21        74
         483       0.72      0.56      0.63       105
         484       0.16      0.06      0.09        83
         485       0.06      0.04      0.05        82
         486       0.32      0.17      0.22        71
         487       0.36      0.20      0.26       120
         488       0.24      0.09      0.13       105
         489       0.63      0.39      0.48        87
         490       0.93      0.81      0.87        32
         491       0.06      0.03      0.04        69
         492       0.09      0.04      0.06        49
         493       0.04      0.02      0.02       117
         494       0.48      0.33      0.39        61
         495       0.94      0.76      0.84       344
         496       0.23      0.17      0.20        52
         497       0.51      0.33      0.40       137
         498       0.29      0.14      0.19        98
         499       0.37      0.20      0.26        79

   micro avg       0.58      0.39      0.47    173812
   macro avg       0.46      0.32      0.37    173812
weighted avg       0.56      0.39      0.46    173812
 samples avg       0.43      0.37      0.37    173812

Time taken to train the model : 0:38:01.741741
```

Out[6]:

```
['lr_with_more_title_weight_lr_bow_2grm_hyp_tuned.pkl']
```

Conclusion:

What we did throughout this experiment:

The objective of this experiment was to suggest tags based on the questions that are posted in StackOverflow. StackOverflow, as we know, is a website which serves as a platform of millions to programmers around the globe to ask and answer questions. There are a wide range of questions that are asked in StackOverflow, from simple computer science questions to the most advance topics in programming. There's a multitude of different domains that are there in StackOverflow. In order to correctly classify a question to it's correct domain, StackOverflow uses this wonderful system of predicting tags based on the query questions. In this way, questions are given tags which results in the questions being answered by relevant people. Imagine, we have a question on Python - "What is the Pythonic way to scrap web pages?". Now, based on this questions StackOverflow may give it tags like 'Python', 'Web-Scrapping' etc. Now, had there been no system of tags, this questions would have been sent to anyone who is a member of StackOverflow. However, since we have these tags 'Python' and 'Web-Scrapping' as suggested by StackOverflow, the questions will only go to people who are interested in 'Python' and 'Web-Scrapping'. This increases productivity. It also reduces the hassle of a question being answered by an user who doesn't work with Python. Overall, this increases the user experience of all the users who come to StackOverflow.

Here, we are given a dataset which contains questions from almost 6 million users. Each question has corresponding tags associated to it. Based on the given data, we have to build a system which will predict tags based on new unseen questions. Each question is described by the following features - 'Title', 'Body', 'Tags'. The title and body text will be used to predict the tags. Remember, there may be more than one tags associated to a question. So it's not only a multiclass problem, but it's also a multilabel problem. In such a scenario the best metric that we would chose is the Micro Average F1 Score. We have chosen this metric to make use of the weighted average of the F1 score of each class. Micro averaged F1 score calculates metrics globally by counting the total true positives, false negatives and false positives. This is a better metric when we have class imbalance. We have to keep the following things in mind:

1. We have to predict as many tags as possible with high precision and recall.
2. Incorrect tags could impact customer experience on StackOverflow.
3. There is no strict latency constraints, which means given a query point, the program can take few minutes to correctly predict the tags.

The first thing we have done is perform some basic Exploratory Data Analysis on the given dataset. EDA is an important step to understand which features are important in the context of the problem and which features are not. In the EDA section we got information about the following:

1. Number of times each question appeared in the database
2. Distribution of number of tags per question. (Avg. number of tags per question: 2.899440)
3. Total number of unique tags present in the dataset (42048)
4. Number of times each tag has appeared across the entire dataset.
5. c#, java, php are the most frequent tags in StackOverflow. (c# occurs 331505 number of times)

From, what we have observed till now is each question can be associated to multiple tags. So we can treat this as a multilabel problem. Afer the EDA is done, we have performed some data cleaning tasks. Due to limitation in computing power, we will sample only 1 million data points. In the cleaning phase, we have seperated the code snippets from the body, we have removed special characters from the title as well as the body. We have removed all the necessary stopwords, we have used Snowball stemmer to stem the words.

After the data cleaning step has been done we will plot a variance plot which explains how much information across the entire dataset is retained with the number of tags. Rememeber there are 42000 unique tags that we have. We see that more than 99% variance is retained with number of tags = 5500 and more than 90% variance is retained when we use 500 tags. On building ML models with 5500 tags we see that it takes roughly 6-8 hours to train a Logistic Regression model. So, for the sake of time, we will use only 500 tags to train all our future models.

We will convert each of these 500 tags to binary outputs using the Count Vectorizer object of scikit learn. We will sample 0.5 million data points with 500 tags, and 3 times more weight added to the titles. We have featurized the data using TFIDF and BOW (1-4 grams) vectorizers and use several models with Hyperparameter tuning to get the highest value of micro f1 score. We will use a simple Logistic Regression model, we will use SGD Classifier with both 'log' loss and 'hinge' loss. At the end we have done a comparison between all the models that we have used to conclude which one actually performed better for our problem. It turns out that the simple Logistic Regression model works best for our problem set. Since, this is a multilabel problem we will use the OnvVsRest classifier along with our base estimators to train the model. Please find below the list of models that we have trained along with their class precision, recall values and micro f1 score.

Comparing performance of all the models.

In [10]:

```
from IPython.core.display import display, HTML
display(HTML("<style>.container { width:100% !important; }</style>"))

from prettytable import PrettyTable

#Table 1
print("TF-IDF with 1 million datapoints: ")
print("="*33)
table =PrettyTable()
table.field_names = ["Model", "Vectorizer", "Accuracy", "Hamming loss","Precision","Recall","Micro f1"]
table.add_row(["LogisticRegression", 'TF-IDF ', 0.081965,0.000412,0.53,0.26,0.096302])
print(table)
print("="*93+"\n"+"="*93+"\n\n")

#Table 2
print("TF-IDF with 0.5 million datapoints: ")
print("="*35)
table =PrettyTable()
table.field_names = ["Model", "Vectorizer", "Accuracy", "Hamming loss","Precision","Recall","Micro f1"]
table.add_row(["SGDClassifier(loss=log) ",'TF-IDF ',  0.23682, 0.002779, 0.7222, 0.3263, 0.4495])
table.add_row(["LogisticRegression", 'TF-IDF ', 0.25108,0.00270302,0.7172,0.3672,0.4858])
print(table)
print("="*99+"\n"+"="*99+"\n\n")

#Table 3
print("\nBOW with (1,4)-grams and 0.5M datapoints: ")
print("="*41)
table =PrettyTable()
table.field_names = ["Model", "Vectorizer", "Accuracy", "Hamming loss","Precision","Recall","Micro f1"]
table.add_row(["Logistic Regression",'BOW', 0.21224, 0.00313,0.5686,0.4097,0.4762])
table.add_row(["SGDClassifier(loss=log)", 'BOW', 0.1868, 0.00326,0.5507,0.3310,0.4135])
table.add_row(["SGDClassifier(loss=log) HypTuned", 'BOW', 0.14065, 0.00400758,0.4251,0.4339,0.4295])
table.add_row(["SGDClassifier(loss=hinge)", 'BOW', 0.11029, 0.005941,0.2872,0.4785,0.3589])
table.add_row(["SGDClassifier(loss=hinge) HypTuned", 'BOW', 0.1622, 0.003493,0.4965,0.3630,0.4194])
print(table)
print("="*109+"\n"+"="*109+"\n\n")

#Table 4
print("\nBOW with (1,2)-grams and 0.5M datapoints: ")
print("="*41)
table =PrettyTable()
table.field_names = ["Model", "Vectorizer", "Accuracy", "Hamming loss","Precision","Recall","Micro f1"]
table.add_row(["Logistic Regression (hyp tuned)",'BOW', 0.21343, 0.00311664,0.5758,0.3928,0.4670])
print(table)
print("="*94+"\n"+"="*94+"\n\n")
```

TF-IDF with 1 million datapoints:

Model	Vectorizer	Accuracy	Hamming loss	Precision	Recall	Micro f1
LogisticRegression	TF-IDF	0.081965	0.000412	0.53	0.26	0.096302

TF-IDF with 0.5 million datapoints:

Model	Vectorizer	Accuracy	Hamming loss	Precision	Recall	Micro f1
SGDClassifier(loss=log)	TF-IDF	0.23682	0.002779	0.7222	0.3263	0.4495
LogisticRegression	TF-IDF	0.25108	0.00270302	0.7172	0.3672	0.4858

BOW with (1,4)-grams and 0.5M datapoints:

Model	Vectorizer	Accuracy	Hamming loss	Precision	Recall	Micro f1
Logistic Regression	BOW	0.21224	0.00313	0.5686	0.4097	0.4762
SGDClassifier(loss=log)	BOW	0.1868	0.00326	0.5507	0.331	0.4135
SGDClassifier(loss=log) HypTuned	BOW	0.14065	0.00400758	0.4251	0.4339	0.4295
SGDClassifier(loss=hinge)	BOW	0.11029	0.005941	0.2872	0.4785	0.3589
SGDClassifier(loss=hinge) HypTuned	BOW	0.1622	0.003493	0.4965	0.363	0.4194

BOW with (1,2)-grams and 0.5M datapoints:

Model	Vectorizer	Accuracy	Hamming loss	Precision	Recall	Micro f1
Logistic Regression (hyp tuned)	BOW	0.21343	0.00311664	0.5758	0.3928	0.467

The following sections were added as part of the EDA tasks :

3.1.6, 3.1.7, 3.1.8, 3.2.7, 3.2.8