```
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
```

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 know
careers.

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

Problem Statemtent

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

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

1.2 Source / useful links

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

1.3 Real World / Business Objectives and Constraints

1. Predict as many tags as possible with high precision and recall.
2. Incorrect tags could impact customer experience on StackOverflow.
3. No strict latency constraints.

2. Machine Learning problem

2.1 Data

2.1.1 Data Overview

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

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

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

Size of Train.csv - 6.75GB

Size of Test.csv - 2GB

Number of rows in Train.csv = 6034195

The questions are randomized and contains a mix of verbose text sites as well as sites related to math and program
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,
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

       <p>The answer should come in the form of a table like</p>\n\n
       <pre><code>
       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
        </code></pre>\n\n
        <p>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)</p>\n\n
        <p>The output is not coming,can anyone correct the code or tell me what\'s wrong?<

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 though
of a data-point that are not mutually exclusive, such as topics that are relevant for a document. A question on Stack
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
score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 scor
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 m
imbalance.

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

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

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

3. Exploratory Data Analysis

3.1 Data Loading and Cleaning

3.1.1 Using Pandas with SQLite to Load the data

```
#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=chunk
        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

```
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 tra
```

```
Number of rows in the database :
 6034196
Time taken to count the number of rows : 0:01:15.750352
```

3.1.3 Checking for duplicates

```
#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 GR
    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
```

```
Time taken to run this cell : 0:04:33.560122
```

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

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

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

```
number of duplicate questions : 1827881 ( 30.2920389063 % )
```

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

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

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

```
# adding a new feature number of tags per question
print("Time taken to run this cell :", datetime.now() - start)
df_no_dup.head()
```

```
Time taken to run this cell : 0:00:03.169523
```

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

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

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

```
#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)
```

```
#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 genarate tr
```

```
Time taken to run this cell : 0:00:52.992676
```

3.2 Analysis of Tags

3.2.1 Total number of unique tags

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

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

```
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
```

```
#'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 tages we have : ['.a', '.app', '.asp.net-mvc', '.aspxauth', '.bash-profile',
```

3.2.3 Number of times a tag appeared

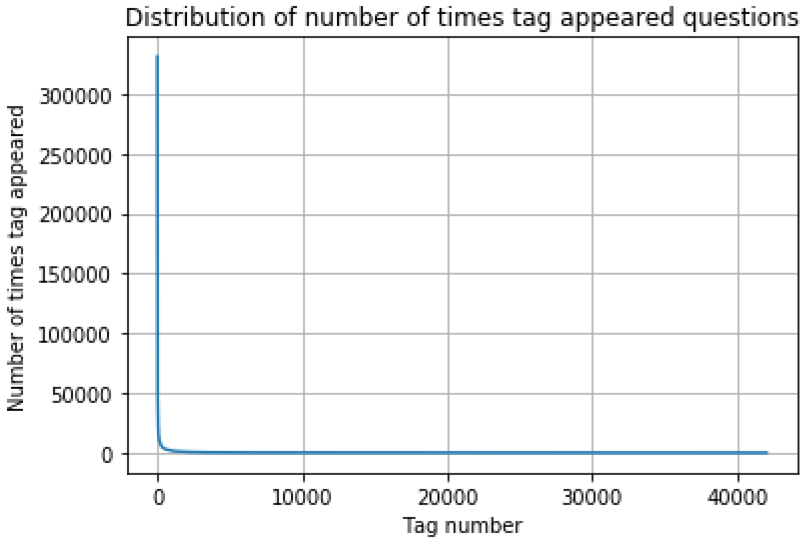
```
# https://stackoverflow.com/questions/15115765/how-to-access-sparse-matrix-elements
#Lets now store the document term matrix in a dictionary.
freqs = tag_dtm.sum(axis=0).A1
result = dict(zip(tags, freqs))
```

```
#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 = pd.read_csv("tag_counts_dict_dtm.csv", names=['Tags', 'Counts'])
tag_df.head()
```

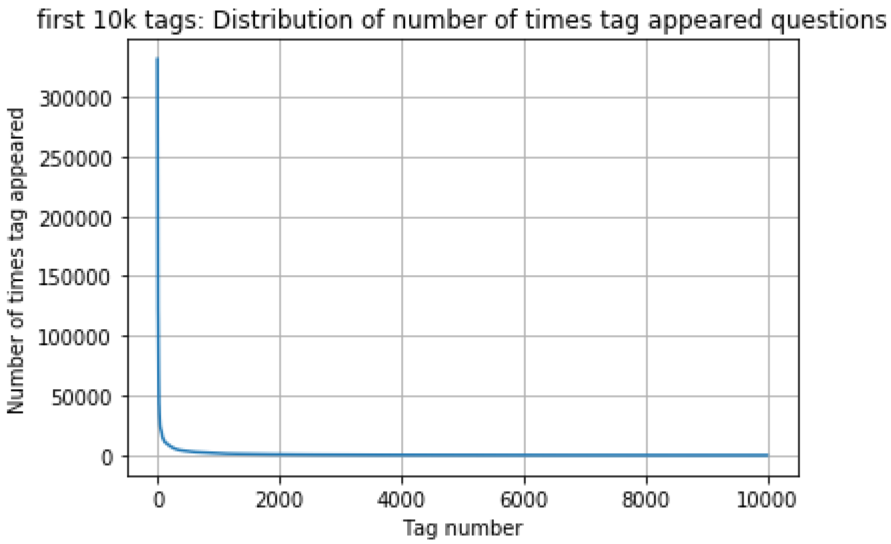
	Tags	Counts
0	.a	18
1	.app	37
2	.asp.net-mvc	1
3	.aspxauth	21
4	.bash-profile	138

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

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

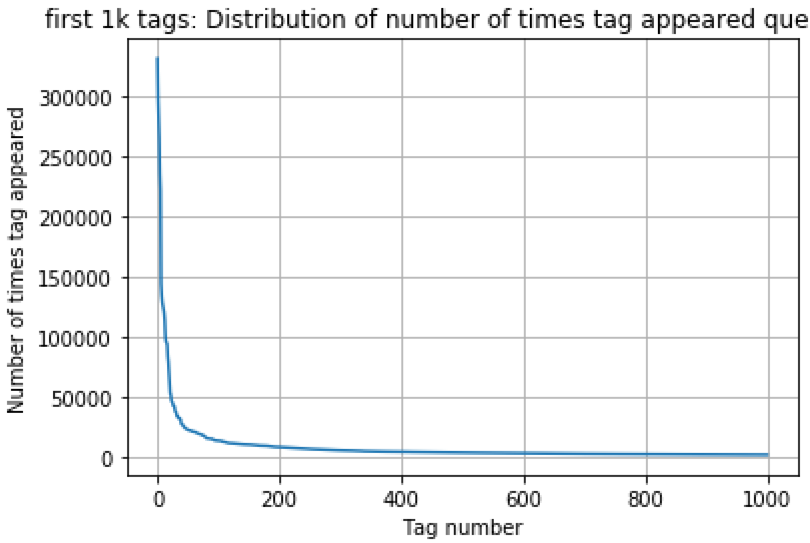


```
plt.plot(tag_counts[0:10000])
plt.title('first 10k tags: Distribution of number of times tag appeared questions')
plt.grid()
plt.xlabel("Tag number")
plt.ylabel("Number of times tag appeared")
plt.show()
print(len(tag_counts[0:10000:25]), tag_counts[0:10000:25])
```



```
400 [331505  44829  22429  17728  13364  11162  10029   9148   8054   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]
```

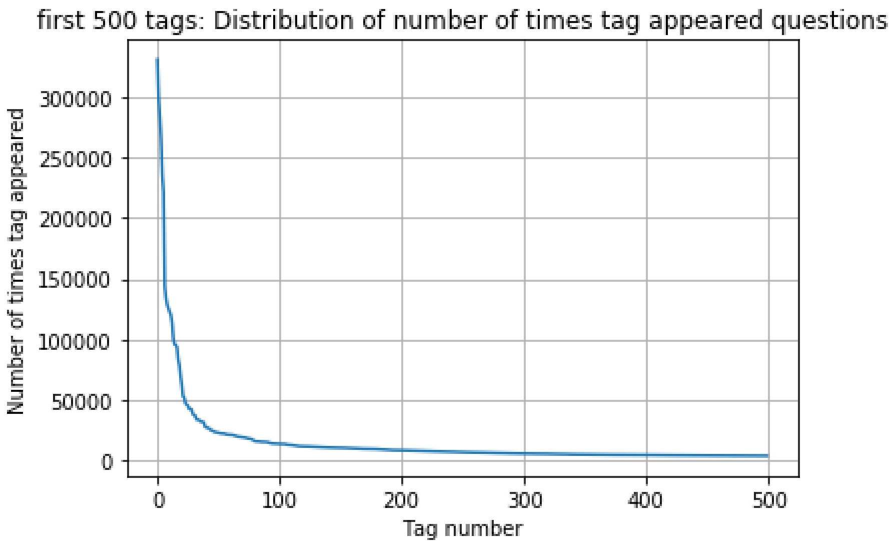
```
plt.plot(tag_counts[0:1000])
plt.title('first 1k tags: Distribution of number of times tag appeared questions')
plt.grid()
plt.xlabel("Tag number")
plt.ylabel("Number of times tag appeared")
plt.show()
print(len(tag_counts[0:1000:5]), tag_counts[0:1000:5])
```



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

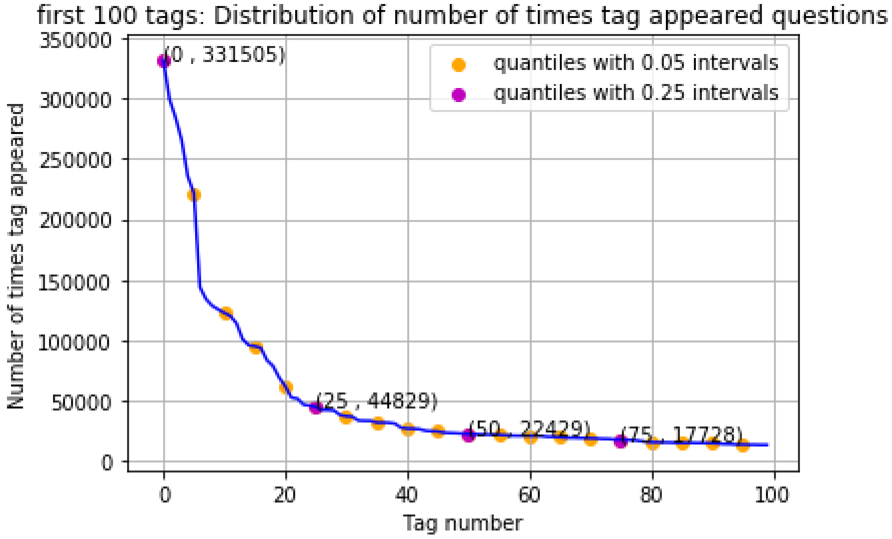
```
plt.plot(tag_counts[0:500])
plt.title('first 500 tags: Distribution of number of times tag appeared questions')
plt.grid()
plt.xlabel("Tag number")
plt.ylabel("Number of times tag appeared")
plt.show()
```

```
print(len(tag_counts[0:500:5]), tag_counts[0:500:5])
```



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

```
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
# quantiles with 0.25 difference
plt.scatter(x=list(range(0,100,25)), y=tag_counts[0:100:25], c='m', label = "quantiles with 0

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

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



```
20 [331505 221533 122769  95160  62023  44829  37170  31897  26925  24537
  22429  21820  20957  19758  18905  17728  15533  15097  14884  13703]
```

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

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

Observations:

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

3.2.4 Tags Per Question

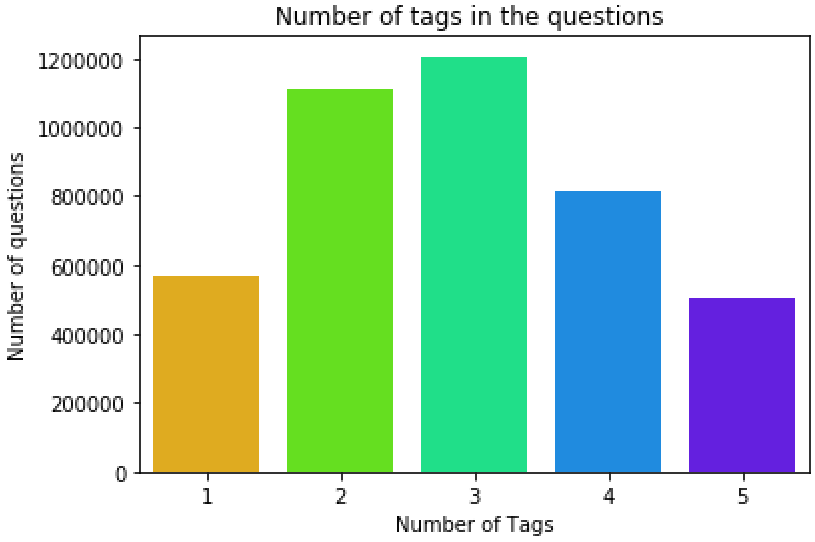
```
#Storing the count of tag in each question in list 'tag_count'
tag_quest_count = tag_dtm.sum(axis=1).tolist()
#Converting list of lists into single list, we will get [[3], [4], [2], [2], [3]] and we are
tag_quest_count=[int(j) for i in tag_quest_count for j in i]
print ('We have total {} datapoints.'.format(len(tag_quest_count)))

print(tag_quest_count[:5])
```

```
We have total 4206314 datapoints.
[3, 4, 2, 2, 3]
```

```
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_coun
```

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

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



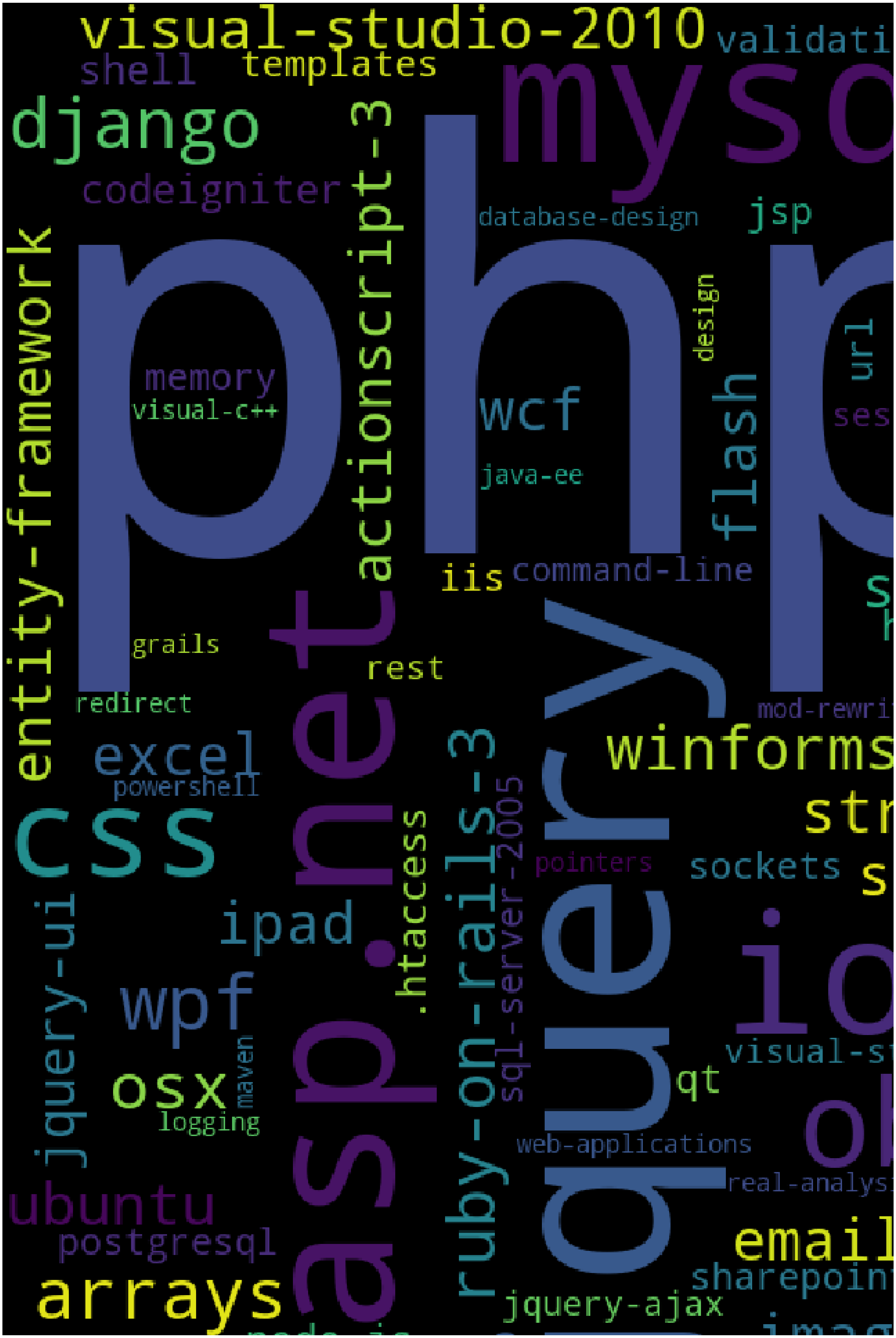
Observations:

1. Maximum number of tags per question: 5
2. Minimum number of tags per question: 1
3. Avg. number of tags per question: 2.899
4. Most of the questions are having 2 or 3 tags

3.2.5 Most Frequent Tags

```
# 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)
```

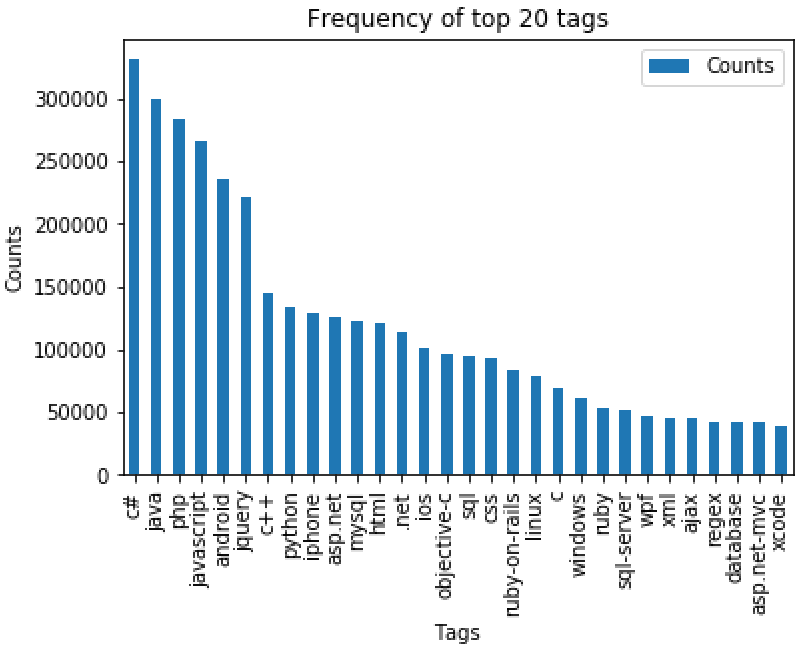

visual-studio-2010
validati
mysq
php
shell
templates
django
actionscript-3
codeigniter
database-design
jsp
entity-framework
design
url
memory
visual-c++
wcf
flash
ses
java-ee
iis
command-line
grails
rest
redirect
mod-rewri
jquery
asp.net
excel
winforms
powershell
css
str
.htaccess
ruby-on-rails-3
sql-server-2005
pointers
sockets
ipad
io
jquery-ui
wpf
visual-s
osx
maven
qt
logging
ob
web-applications
real-analysi
ubuntu
postgresql
email
arrays
sharepoin
jquery-ajax
imag

```
Time taken to run this cell : 0:00:05.470788
```

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

3.2.6 The top 20 tags

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



Observations:

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

3.3 Cleaning and preprocessing of Questions

3.3.1 Preprocessing

1. Sample 1M data points
2. Separate out code-snippets from Body
3. Remove Spcial characters from Question title and description (not in code)
4. Remove stop words (Except 'C')
5. Remove HTML Tags
6. Convert all the characters into small letters
7. Use SnowballStemmer to stem the words

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


#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()
    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,
create_database_table("Processed.db", sql_create_table)
```

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

```
# 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 10

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

```
Tables in the databse:
QuestionsProcessed
Cleared All the rows
Time taken to run this cell : 0:06:32.806567
```

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

```
#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]

    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)!

    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
    if (questions_proccesed%100000==0):
        print("number of questions completed=",questions_proccesed)

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

print( "Avg. length of questions(Title+Body) before processing: %d"%no_dup_avg_len_pre)
print( "Avg. length of questions(Title+Body) after processing: %d"%no_dup_avg_len_post)
print ("Percent of questions containing code: %d"%((questions_with_code*100.0)/questions_proc

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
```

```
# 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()


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 relati
----------------------------------------------------------------------------------------
('explan new statement review section c code came accross statement block come accross n
----------------------------------------------------------------------------------------
('error function notat function solv logic riddl iloczyni list structur list possibl can
----------------------------------------------------------------------------------------
('step plan move one isp anoth one work busi plan switch isp realli soon need chang lot
----------------------------------------------------------------------------------------
('use ef migrat creat databas googl migrat tutori af first run applic creat databas ef e
----------------------------------------------------------------------------------------
('magento unit test problem magento site recent look way check integr magento site given
----------------------------------------------------------------------------------------
('find network devic without bonjour write mac applic need discov mac pcs iphon ipad con
----------------------------------------------------------------------------------------
('send multipl row mysql databas want send user mysql databas column user skill time nno
----------------------------------------------------------------------------------------
('insert data mysql php powerpoint event powerpoint present run continu way updat slide
----------------------------------------------------------------------------------------
```

```
#Taking 1 Million entries to a dataframe.
write_db = 'Processed.db'
if os.path.isfile(write_db):
    conn_r = create_connection(write_db)
    if conn_r is not None:
        preprocessed_data = pd.read_sql_query("""SELECT question, Tags FROM QuestionsProcesse
conn_r.commit()
conn_r.close()
```

```
preprocessed_data.head()
```

	question	tags
0	resiz root window tkinter resiz root window re...	python tkinter
1	ef code first defin one mani relationship diff...	entity-framework-4.1
2	explan new statement review section c code cam...	c++
3	error function notat function solv logic riddl...	haskell logic
4	step plan move one isp anoth one work busi pla...	dns isp

```
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

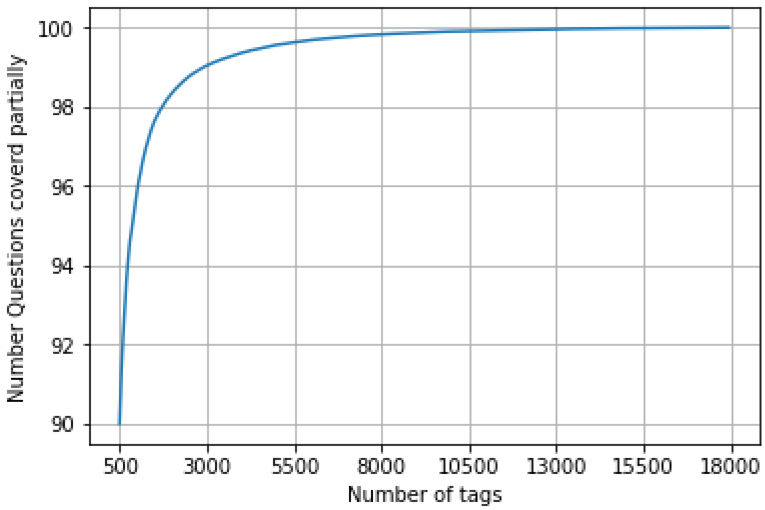
4.1 Converting tags for multilabel problems

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

```
# 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) __

```
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))


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


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.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
print("with ",5500,"tags we are covering ",questions_explained[50],"% of questions")
```



```
with  5500 tags we are covering  99.04 % of questions
```

```
multilabel_yx = tags_to_choose(5500)
print("number of questions that are not covered :", questions_explained_fn(5500),"out of ", t
```

```
number of questions that are not covered : 9599 out of  999999
```

```
print("Number of tags in sample :", multilabel_y.shape[1])
print("number of tags taken :", multilabel_yx.shape[1],"(",(multilabel_yx.shape[1]/multilabel
```

```
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)

```
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,:]
```

```
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

```
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
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
```

```
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)
```

```
# 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)
```

```
"\nfrom skmultilearn.adapt import MLkNN\nclassifier = MLkNN(k=21)\n\n# train\nclassifier
```

4.4 Applying Logistic Regression with OneVsRest Classifier

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

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

```
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
```

```
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
```

```
        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
         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
         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      0.30      0.42        91
         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
         972       0.48      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
        1145       0.45      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
```