In [3]:
```
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 sklearn.naive_bayes import GaussianNB
from datetime import datetime
import pickle
```

5. Assignments

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

Modeling with less data points (0.1M data points) and more weight to title and 500 tags only.

In [1]:
```
def tags_to_choose(n):
    t = multilabel_y.sum(axis=0).tolist()[0]
    sorted_tags_i = sorted(range(len(t)), key=lambda i: t[i], reverse=True)
    multilabel_yn=multilabel_y[:,sorted_tags_i[:n]]
    return multilabel_yn

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

In [ ]:
```
#Taking only 0.1 Million entries to a dataframe due to computational constrain.
con = sqlite3.connect('Titlemoreweight.db')
processed = pd.read_sql_query("""SELECT * FROM QuestionsProcessed""", con)
```

In [5]:
```
processed.head()
```

	question	code	tags	words_pre	words_post	is_code
0	dynam datagrid bind silverlight dynam datagrid...	['MyClass myInstance = new MyClass();\ndataGri...	c# silverlight data-binding	904	252	1
1	dynam datagrid bind silverlight dynam datagrid...	['MyClass myInstance = new MyClass();\ndataGri...	c# silverlight data-binding columns	904	252	1
2	java.lang.noclassdeffounderror javax servlet j...	['<%@taglib prefix="c" uri="http://java.sun...	jsp jstl	745	463	1
3	java.sql.sqlexcept microsoft odbc driver manag...	['try {\n Class.forName("sun.jdbc.odbc.Jdbc...	java jdbc	1056	249	1
4	better way updat feed fb php sdk better way up...	['$data = array("message" => "Hello World!"...	facebook api facebook-php-sdk	607	233	1

In [ ]:
```
preprocessed_data=pd.read_sql_query("""SELECT question, Tags FROM QuestionsProcessed LIMIT 100000""", con)
preprocessed_data.head()
```

In [7]:
```
print("number of data points in sample :", preprocessed_data.shape[0])
print("number of dimensions :", preprocessed_data.shape[1])
```

```
number of data points in sample : 100000
number of dimensions : 2
```

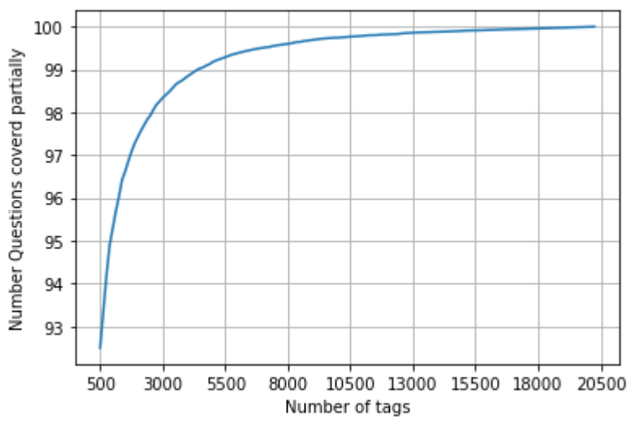
Converting string Tags to multilable output variables

In [8]:
```
vectorizer = CountVectorizer(tokenizer = lambda x: x.split(), binary='true')
multilabel_y = vectorizer.fit_transform(preprocessed_data['tags'])
```

Selecting 500 Tags

In [9]:
```
questions_explained = []
total_tags=multilabel_y.shape[1]
total_qs=preprocessed_data.shape[0]
for i in range(500, total_tags, 100):
    questions_explained.append(np.round(((total_qs-questions_explained_fn(i))/total_qs)*100,3))
```

In [10]:
```
fig, ax = plt.subplots()
ax.plot(questions_explained)
xlabel = list(500+np.array(range(-50,450,50))*50)
ax.set_xticklabels(xlabel)
plt.xlabel("Number of tags")
plt.ylabel("Number Questions coverd partially")
plt.grid()
plt.show()
# you can choose any number of tags based on your computing power, minimun is 500(it covers 90% of the tags)
print("with ",5500,"tags we are covering ",questions_explained[50],"% of questions")
print("with ",500,"tags we are covering ",questions_explained[0],"% of questions")
```




```
with  5500 tags we are covering  99.481 % of questions
with  500 tags we are covering  92.5 % of questions
```

In [11]:
```
# we will be taking 500 tags
multilabel_yx = tags_to_choose(500)
print("number of questions that are not covered :", questions_explained_fn(500),"out of ", total_qs)
```

```
number of questions that are not covered : 7500 out of  100000
```

In [12]:
```
# Train Test Split
train_datasize=80000
x_train=preprocessed_data.head(train_datasize)
x_test=preprocessed_data.tail(preprocessed_data.shape[0] - 80000)

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

In [13]:
```
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 : (80000, 500)
Number of data points in test data : (20000, 500)
```

1. Use bag of words upto 4 grams and compute the micro f1 score with Logistic regression(OvR)

Featurizing data with BOW vectorizer

NOTE: Limiting Maximum features to 100k because of coumputational strain

In [15]:
```
# Bag OF Word of n_gram_range=(1,4)
start = datetime.now()
vectorizer = CountVectorizer(tokenizer = lambda x: x.split(),  ngram_range=(1,4),max_features=100000)
x_train_multilabel = vectorizer.fit_transform(x_train['question'])
x_test_multilabel = vectorizer.transform(x_test['question'])
print("Time taken to run this cell :", datetime.now() - start)
```

```
Time taken to run this cell : 0:01:04.732305
```

In [16]:
```
print("Dimensions of train data X:",x_train_multilabel.shape, "Y :",y_train.shape)
print("Dimensions of test data X:",x_test_multilabel.shape,"Y:",y_test.shape)
```

```
Dimensions of train data X: (80000, 100000) Y : (80000, 500)
Dimensions of test data X: (20000, 100000) Y: (20000, 500)
```

Note : Taking only 100000 datapoint and 100k maximum feature because of computational constrain

In [26]:
```
start = datetime.now()
classifier_2 = OneVsRestClassifier(LogisticRegression(penalty='l1', n_jobs=-1),)
classifier_2.fit(x_train_multilabel, y_train)
predictions_2 = classifier_2.predict(x_test_multilabel)
print("Accuracy :",metrics.accuracy_score(y_test, predictions_2))
print("Hamming loss ",metrics.hamming_loss(y_test,predictions_2))


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

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

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

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

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

```
Accuracy : 0.17445
Hamming loss  0.003358
Micro-average quality numbers
Precision: 0.5830, Recall: 0.3649, F1-measure: 0.4489
Macro-average quality numbers
Precision: 0.4208, Recall: 0.2851, F1-measure: 0.3288
              precision    recall  f1-score   support

           0       0.69      0.39      0.50       820
           1       0.53      0.32      0.40      1931
           2       0.37      0.18      0.24       544
           3       0.51      0.21      0.29       222
           4       0.70      0.51      0.59      1311
           5       0.75      0.51      0.61      1014
           6       0.66      0.44      0.53      1374
           7       0.73      0.59      0.65       702
           8       0.90      0.63      0.74      1424
           9       0.75      0.64      0.69      1037
          10       0.72      0.57      0.64       797
          11       0.55      0.38      0.45       156
          12       0.57      0.36      0.44        36
          13       0.67      0.41      0.51       610
          14       0.41      0.24      0.30       405
          15       0.61      0.24      0.34       144
          16       0.48      0.25      0.33       425
          17       0.58      0.32      0.41       485
          18       0.73      0.65      0.69       269
          19       0.85      0.61      0.71       518
          20       0.49      0.29      0.36       529
          21       0.82      0.57      0.68       294
          22       0.80      0.41      0.54       520
          23       0.45      0.28      0.35       246
          24       0.54      0.37      0.44       312
          25       0.46      0.33      0.38       314
          26       0.56      0.30      0.39       190
          27       0.31      0.12      0.17       342
          28       0.36      0.26      0.30        96
          29       0.29      0.16      0.20        32
          30       0.64      0.47      0.54       747
          31       0.45      0.36      0.40        14
          32       0.62      0.61      0.62       166
          33       0.56      0.35      0.43       171
          34       0.54      0.31      0.39       256
          35       0.79      0.55      0.65       199
          36       0.15      0.08      0.11        60
          37       0.26      0.19      0.22       203
          38       0.68      0.49      0.57       201
          39       0.40      0.31      0.35       208
          40       0.25      0.15      0.19        13
          41       0.35      0.16      0.22       154
          42       0.34      0.30      0.32        69
          43       0.29      0.20      0.24       426
          44       0.44      0.27      0.34        77
          45       0.47      0.29      0.36       223
          46       0.51      0.33      0.40       144
          47       0.69      0.49      0.57       245
          48       0.42      0.20      0.27        91
          49       0.59      0.41      0.48       157
          50       0.90      0.67      0.77       132
          51       0.84      0.66      0.74        41
          52       0.56      0.42      0.48       124
          53       0.23      0.26      0.24        96
          54       0.18      0.12      0.15       128
          55       0.50      0.24      0.32        46
          56       0.66      0.64      0.65       151
          57       0.08      0.01      0.02        80
          58       0.32      0.17      0.22        65
          59       0.37      0.16      0.23       182
          60       0.90      0.69      0.78       148
          61       0.36      0.14      0.20       196
          62       0.27      0.21      0.24        58
          63       0.86      0.28      0.42        43
          64       0.61      0.34      0.44       197
          65       0.65      0.43      0.51        82
          66       0.69      0.54      0.61        50
          67       0.59      0.54      0.56       105
          68       0.16      0.05      0.08        98
          69       0.21      0.07      0.10       238
          70       0.25      0.09      0.13        35
          71       0.51      0.50      0.50        54
          72       0.00      0.00      0.00        25
          73       0.29      0.24      0.26        29
          74       0.20      0.10      0.14        29
          75       0.26      0.23      0.24        40
          76       0.77      0.55      0.64       105
          77       0.58      0.50      0.54        28
          78       0.18      0.09      0.12       202
          79       0.54      0.41      0.46        37
          80       0.50      0.33      0.40        15
          81       0.47      0.27      0.34        52
          82       0.38      0.28      0.32        50
          83       0.21      0.05      0.09        56
          84       0.73      0.59      0.65        54
          85       0.58      0.62      0.60        34
          86       0.23      0.20      0.21        30
          87       0.56      0.34      0.43        29
          88       0.71      0.83      0.77        24
          89       0.83      0.79      0.81       117
          90       0.11      0.06      0.08        66
          91       0.39      0.25      0.30        68
          92       0.67      0.30      0.41        67
          93       0.53      0.36      0.43        28
          94       0.42      0.29      0.34        17
          95       0.74      0.49      0.59        51
          96       0.57      0.43      0.49        53
          97       0.08      0.02      0.03        61
          98       0.08      0.03      0.04        79
```

99	0.47	0.44	0.46	18
100	0.50	0.09	0.15	11
101	0.62	0.58	0.60	207
102	0.00	0.00	0.00	6
103	0.25	0.03	0.06	30
104	0.29	0.09	0.14	54
105	0.70	0.41	0.52	39
106	0.29	0.13	0.18	70
107	0.40	0.14	0.21	14
108	0.58	0.17	0.26	66
109	0.53	0.34	0.41	50
110	0.56	0.26	0.36	87
111	0.40	0.43	0.42	51
112	0.86	0.06	0.12	291
113	0.97	0.76	0.85	49
114	0.32	0.10	0.15	110
115	0.17	0.04	0.06	28
116	0.00	0.00	0.00	5
117	0.25	0.09	0.13	56
118	0.74	0.45	0.56	125
119	0.70	0.43	0.54	44
120	0.75	0.36	0.48	42
121	0.38	0.20	0.26	55
122	0.67	0.43	0.52	68
123	0.16	0.10	0.12	82
124	0.00	0.00	0.00	0
125	0.71	0.71	0.71	7
126	0.12	0.06	0.08	18
127	0.40	0.13	0.20	31
128	0.75	0.46	0.57	13
129	0.64	0.54	0.59	50
130	0.19	0.10	0.13	91
131	0.64	0.60	0.62	35
132	0.21	0.12	0.15	26
133	0.25	0.06	0.10	32
134	0.61	0.40	0.48	35
135	0.76	0.68	0.71	37
136	0.00	0.00	0.00	55
137	0.24	0.39	0.30	41
138	0.38	0.40	0.39	15
139	0.28	0.14	0.19	99
140	0.87	0.70	0.77	86
141	0.45	0.25	0.32	53
142	0.75	0.08	0.15	36
143	0.52	0.52	0.52	66
144	0.53	0.47	0.50	64
145	0.27	0.12	0.17	25
146	0.15	0.10	0.12	125
147	0.24	0.27	0.25	15
148	0.64	0.52	0.57	48
149	0.34	0.26	0.30	65
150	0.00	0.00	0.00	11
151	0.33	0.33	0.33	15
152	0.21	0.13	0.16	52
153	0.44	0.39	0.41	18
154	0.43	0.19	0.26	16
155	0.27	0.20	0.23	20
156	0.47	0.22	0.30	121
157	0.51	0.31	0.38	107
158	0.00	0.00	0.00	15
159	0.67	0.50	0.57	105
160	0.47	0.39	0.43	69
161	0.54	0.36	0.43	56
162	0.25	0.09	0.13	47
163	0.04	0.01	0.01	121
164	0.39	0.29	0.33	41
165	0.00	0.00	0.00	229
166	0.71	0.30	0.42	98
167	0.39	0.21	0.27	33
168	0.53	0.18	0.27	44
169	0.70	0.47	0.56	45
170	0.84	0.41	0.55	51
171	0.00	0.00	0.00	18
172	0.49	0.46	0.47	48
173	0.33	0.42	0.37	12
174	0.29	0.16	0.21	62
175	0.72	0.52	0.61	44
176	0.92	0.77	0.84	30
177	0.48	0.43	0.46	30
178	0.00	0.00	0.00	0
179	1.00	1.00	1.00	1
180	0.52	0.30	0.38	40
181	0.17	0.07	0.10	44
182	0.50	0.50	0.50	2
183	0.53	0.37	0.44	75
184	0.25	0.25	0.25	4
185	0.50	0.23	0.32	64
186	0.23	0.25	0.24	12
187	0.95	0.64	0.76	55
188	0.78	0.66	0.71	64
189	0.41	0.15	0.22	96
190	0.12	0.05	0.07	22
191	0.81	0.17	0.28	76
192	0.68	0.42	0.52	45
193	0.86	0.43	0.57	14
194	0.55	0.46	0.50	50
195	0.82	0.45	0.58	20
196	0.81	0.63	0.71	35
197	0.50	0.33	0.40	94
198	0.00	0.00	0.00	14
199	0.12	0.04	0.06	25
200	0.50	0.17	0.25	54
201	0.25	0.09	0.13	22
202	0.30	0.26	0.28	43
203	0.14	0.02	0.04	43
204	0.98	0.65	0.78	62
205	0.00	0.00	0.00	3
206	0.14	0.07	0.09	43

207	0.33	0.14	0.20	7
208	0.20	0.12	0.15	8
209	0.56	0.12	0.20	42
210	0.22	0.40	0.29	10
211	0.35	0.20	0.25	40
212	0.73	0.48	0.58	23
213	0.00	0.00	0.00	6
214	0.63	0.47	0.54	47
215	0.38	0.13	0.19	62
216	0.62	0.42	0.50	77
217	0.18	0.09	0.12	22
218	0.33	0.33	0.33	3
219	0.06	0.04	0.05	28
220	0.67	0.07	0.13	81
221	0.24	0.13	0.17	31
222	0.17	0.06	0.09	34
223	0.96	0.45	0.61	60
224	0.33	0.30	0.32	10
225	0.75	0.60	0.67	10
226	0.75	0.76	0.76	92
227	0.89	0.62	0.73	13
228	0.40	0.15	0.22	13
229	0.87	0.77	0.81	43
230	0.29	0.17	0.21	35
231	0.00	0.00	0.00	4
232	0.25	0.10	0.14	20
233	0.48	0.32	0.38	145
234	0.84	0.56	0.67	55
235	0.00	0.00	0.00	2
236	0.27	0.08	0.12	37
237	0.69	0.42	0.52	90
238	0.50	0.19	0.27	58
239	0.50	0.25	0.33	20
240	0.89	0.69	0.78	61
241	0.78	0.74	0.76	42
242	0.56	0.63	0.59	30
243	0.80	0.53	0.64	66
244	0.56	0.24	0.33	42
245	0.10	0.06	0.08	31
246	1.00	0.50	0.67	6
247	0.25	0.17	0.20	18
248	0.83	0.57	0.67	51
249	0.57	0.47	0.52	17
250	0.62	0.59	0.60	22
251	0.76	0.42	0.54	52
252	0.62	0.17	0.27	29
253	0.11	0.07	0.09	28
254	0.00	0.00	0.00	10
255	0.20	0.20	0.20	5
256	0.25	0.33	0.29	3
257	0.64	0.34	0.44	41
258	0.23	0.10	0.14	30
259	1.00	0.67	0.80	3
260	0.00	0.00	0.00	38
261	0.00	0.00	0.00	1
262	0.46	0.32	0.37	19
263	0.00	0.00	0.00	14
264	0.07	0.05	0.06	37
265	0.11	0.11	0.11	9
266	0.20	0.24	0.22	45
267	0.64	0.55	0.59	33
268	0.81	0.81	0.81	16
269	0.47	0.46	0.46	35
270	0.60	0.27	0.37	11
271	0.00	0.00	0.00	30
272	0.27	0.38	0.32	8
273	0.12	0.14	0.13	21
274	0.46	0.31	0.37	123
275	0.38	0.25	0.30	67
276	0.89	0.80	0.84	20
277	0.00	0.00	0.00	14
278	0.33	0.16	0.21	19
279	0.73	0.67	0.70	12
280	0.00	0.00	0.00	15
281	0.92	0.65	0.76	17
282	0.94	0.71	0.81	41
283	0.64	0.47	0.54	15
284	0.51	0.34	0.41	74
285	0.38	0.13	0.20	38
286	0.13	0.12	0.13	16
287	0.29	0.07	0.11	30
288	0.90	0.64	0.75	28
289	0.00	0.00	0.00	21
290	0.80	0.59	0.68	41
291	0.10	0.08	0.09	12
292	0.39	0.29	0.33	24
293	0.47	0.35	0.40	20
294	0.21	0.22	0.21	23
295	0.00	0.00	0.00	29
296	0.25	0.11	0.15	28
297	0.36	0.29	0.32	42
298	0.19	0.06	0.09	53
299	0.00	0.00	0.00	36
300	0.21	0.12	0.15	41
301	0.53	0.46	0.49	37
302	0.88	0.58	0.70	26
303	0.20	0.09	0.13	11
304	0.33	0.23	0.27	31
305	0.33	0.29	0.31	17
306	0.40	0.22	0.29	9
307	0.50	0.33	0.40	6
308	0.00	0.00	0.00	34
309	0.61	0.40	0.48	43
310	0.06	0.03	0.04	30
311	0.33	0.14	0.20	50
312	0.20	0.04	0.07	24
313	0.38	0.14	0.21	42
314	0.42	0.23	0.29	22

315	0.00	0.00	0.00	58
316	0.50	0.10	0.17	10
317	0.38	0.26	0.31	57
318	0.57	0.40	0.47	10
319	0.17	0.09	0.12	11
320	0.14	0.18	0.16	11
321	0.25	0.12	0.17	8
322	0.56	0.41	0.47	22
323	0.79	0.79	0.79	28
324	0.63	0.58	0.60	50
325	0.29	0.11	0.16	18
326	0.18	0.09	0.12	33
327	0.16	0.18	0.17	17
328	0.36	0.17	0.23	29
329	0.33	0.14	0.20	7
330	0.42	0.50	0.45	10
331	0.19	0.20	0.19	25
332	1.00	1.00	1.00	2
333	0.71	0.45	0.56	11
334	0.00	0.00	0.00	24
335	0.33	0.20	0.25	5
336	0.17	0.03	0.05	33
337	0.45	0.17	0.24	30
338	0.91	0.71	0.80	42
339	0.22	0.08	0.11	26
340	0.48	0.36	0.41	36
341	1.00	0.54	0.70	13
342	0.60	0.55	0.57	11
343	0.43	0.30	0.35	10
344	0.14	0.05	0.07	21
345	0.00	0.00	0.00	0
346	0.00	0.00	0.00	6
347	0.00	0.00	0.00	12
348	0.17	0.08	0.11	13
349	0.88	0.29	0.44	24
350	0.75	0.44	0.56	27
351	0.42	0.12	0.18	43
352	0.25	0.03	0.06	30
353	0.45	0.41	0.43	22
354	0.19	0.10	0.13	31
355	0.62	0.80	0.70	10
356	0.00	0.00	0.00	20
357	0.65	0.65	0.65	20
358	0.61	0.39	0.48	28
359	0.59	0.48	0.53	21
360	0.22	0.08	0.12	25
361	0.52	0.49	0.50	35
362	0.78	0.69	0.74	36
363	0.36	0.24	0.29	17
364	1.00	0.46	0.63	13
365	0.20	0.05	0.08	21
366	0.43	0.17	0.24	18
367	0.44	0.07	0.12	97
368	0.57	0.45	0.50	29
369	0.60	0.75	0.67	12
370	0.22	0.15	0.18	13
371	0.20	0.11	0.14	18
372	0.67	0.33	0.44	6
373	0.40	0.33	0.36	6
374	0.62	0.17	0.26	30
375	0.19	0.19	0.19	27
376	0.20	0.04	0.06	28
377	0.00	0.00	0.00	2
378	0.20	0.25	0.22	4
379	0.25	0.05	0.09	19
380	0.20	0.20	0.20	5
381	0.58	0.39	0.47	18
382	0.50	0.41	0.45	22
383	0.20	0.06	0.10	16
384	0.57	0.31	0.40	13
385	0.50	0.17	0.25	18
386	0.77	0.91	0.83	11
387	0.46	0.45	0.46	88
388	0.10	0.08	0.09	13
389	0.00	0.00	0.00	6
390	0.00	0.00	0.00	6
391	0.95	0.71	0.81	51
392	0.00	0.00	0.00	13
393	0.50	0.35	0.41	37
394	0.00	0.00	0.00	6
395	0.00	0.00	0.00	9
396	0.00	0.00	0.00	13
397	0.75	0.50	0.60	6
398	0.52	0.41	0.46	29
399	0.89	0.73	0.80	33
400	0.33	0.10	0.15	31
401	0.56	0.10	0.17	50
402	0.85	0.61	0.71	18
403	0.17	0.14	0.15	7
404	0.64	0.62	0.63	26
405	0.84	0.82	0.83	56
406	0.60	0.75	0.67	4
407	0.08	0.06	0.07	17
408	0.75	0.55	0.63	11
409	0.10	0.06	0.07	18
410	0.33	0.30	0.32	10
411	0.40	0.13	0.20	45
412	0.82	0.45	0.58	20
413	0.40	0.16	0.23	25
414	0.00	0.00	0.00	20
415	0.00	0.00	0.00	6
416	0.20	0.12	0.15	26
417	0.60	0.30	0.40	10
418	0.00	0.00	0.00	18
419	0.75	0.50	0.60	6
420	0.56	0.53	0.55	17
421	0.00	0.00	0.00	1
422	0.00	0.00	0.00	6

```
423       0.00      0.00      0.00        12
424       1.00      0.50      0.67         4
425       0.60      0.27      0.37        11
426       0.17      0.09      0.12        11
427       0.86      0.75      0.80         8
428       0.64      0.27      0.38        26
429       0.52      0.62      0.57        40
430       0.00      0.00      0.00         2
431       0.08      0.03      0.04        35
432       0.71      0.33      0.45        15
433       0.00      0.00      0.00        18
434       0.00      0.00      0.00         0
435       0.00      0.00      0.00         0
436       0.38      0.18      0.24        28
437       0.33      0.15      0.21        33
438       0.79      0.55      0.65        20
439       0.27      0.08      0.13        36
440       0.18      0.11      0.14        18
441       0.42      0.56      0.48        18
442       0.69      0.69      0.69        16
443       0.00      0.00      0.00        22
444       0.00      0.00      0.00         6
445       0.72      0.62      0.67        21
446       0.79      0.57      0.66        46
447       0.19      0.06      0.09        69
448       0.00      0.00      0.00         7
449       0.00      0.00      0.00         3
450       0.12      0.04      0.06        52
451       0.14      0.06      0.09        16
452       0.89      0.94      0.91        17
453       0.00      0.00      0.00        13
454       0.33      0.18      0.24        11
455       0.00      0.00      0.00        12
456       0.12      0.17      0.14         6
457       0.22      0.11      0.15        18
458       0.00      0.00      0.00        15
459       0.95      0.64      0.77        28
460       0.14      0.06      0.08        18
461       0.57      0.40      0.47        10
462       0.40      0.08      0.14        24
463       0.00      0.00      0.00        18
464       0.96      0.62      0.75        39
465       0.21      0.27      0.24        11
466       0.18      0.06      0.09        35
467       0.07      0.05      0.06        21
468       0.25      0.03      0.05        37
469       0.00      0.00      0.00         5
470       0.00      0.00      0.00         8
471       0.53      0.27      0.36        37
472       0.00      0.00      0.00        47
473       0.46      0.43      0.44        14
474       0.88      0.65      0.75        23
475       0.51      0.50      0.50        66
476       0.00      0.00      0.00         3
477       0.70      0.37      0.48        19
478       0.00      0.00      0.00         1
479       0.17      0.13      0.15        23
480       0.54      0.12      0.19        60
481       0.33      0.15      0.21        26
482       0.50      0.50      0.50         4
483       0.00      0.00      0.00         8
484       0.91      0.43      0.59        23
485       0.64      0.39      0.48        18
486       0.62      0.42      0.50        12
487       0.79      0.38      0.51        29
488       1.00      1.00      1.00         1
489       0.00      0.00      0.00         6
490       0.33      0.14      0.20         7
491       0.00      0.00      0.00         3
492       0.12      0.20      0.15        10
493       0.38      0.32      0.34        19
494       0.20      0.14      0.17         7
495       0.43      0.38      0.40         8
496       0.39      0.39      0.39        18
497       0.38      0.12      0.19        72
498       0.17      0.12      0.14         8
499       0.52      0.41      0.46        32

micro avg       0.58      0.36      0.45     37472
macro avg       0.42      0.29      0.33     37472
weighted avg       0.55      0.36      0.43     37472
samples avg       0.43      0.36      0.36     37472

Time taken to run this cell : 0:33:34.684806
```

2. Perform hyperparam tuning on alpha (or lambda) for Logistic regression to improve the performance using GridSearch

In [ ]:
```
# Train
from sklearn.model_selection import GridSearchCV
start = datetime.now()

param={"estimator__C":[0.0001,0.001,0.01,0.1,1,10,100]} # When you use nested estimators with grid search you can scope the parameters with __
as a separator

lr_ovr = OneVsRestClassifier(LogisticRegression(C=1.0,penalty='l1', n_jobs=-1),)
classifier_2=GridSearchCV(estimator=lr_ovr, param_grid=param ,scoring="f1_micro", return_train_score=True)
classifier_2.fit(x_train_multilabel, y_train)

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

In [3]:
```
import pickle
"""file=open("gridsearch.pkl",'wb')
pickle.dump(classifier_2,file)
file.close()
"""
file=open("gridsearch.pkl",'rb')
gridsearch=pickle.load(file)
```

In [4]:
```
# CV Score
gridsearch.cv_results_
```

```
{'mean_fit_time': array([ 341.93791715,  363.59180037,  423.69569087,  678.68307408,
         935.50097712, 1013.79868793, 1081.59573984]),
 'std_fit_time': array([96.9851387 , 52.27677961, 19.59633154, 50.61290431, 42.16038204,
       37.69262534, 33.27912207]),
 'mean_score_time': array([3.76426148, 3.77001405, 3.71273176, 3.73366205, 3.7477115 ,
       3.70152036, 3.7257247 ]),
 'std_score_time': array([0.200987  , 0.23564235, 0.22734413, 0.21825075, 0.23177296,
       0.18633856, 0.21896373]),
 'param_estimator__C': masked_array(data=[0.0001, 0.001, 0.01, 0.1, 1, 10, 100],
              mask=[False, False, False, False, False, False, False],
       fill_value='?',
            dtype=object),
 'params': [{'estimator__C': 0.0001},
  {'estimator__C': 0.001},
  {'estimator__C': 0.01},
  {'estimator__C': 0.1},
  {'estimator__C': 1},
  {'estimator__C': 10},
  {'estimator__C': 100}],
 'split0_test_score': array([2.01804129e-04, 8.63213305e-02, 2.61891013e-01, 3.86039087e-01,
       4.24427405e-01, 4.22475698e-01, 4.17827891e-01]),
 'split1_test_score': array([2.49745052e-04, 7.40314386e-02, 2.35691296e-01, 3.93431914e-01,
       4.28738496e-01, 4.24752295e-01, 4.14620536e-01]),
 'split2_test_score': array([0.        , 0.05298808, 0.39069205, 0.50160225, 0.52290911,
       0.52000072, 0.51497723]),
 'mean_test_score': array([1.50518275e-04, 7.11138427e-02, 2.96090271e-01, 4.27023484e-01,
       4.58690868e-01, 4.55742102e-01, 4.49141063e-01]),
 'std_test_score': array([0.00010822, 0.0137637 , 0.06774206, 0.05282046, 0.0454424 ,
       0.04544636, 0.04657074]),
 'rank_test_score': array([7, 6, 5, 4, 1, 2, 3], dtype=int32),
 'split0_train_score': array([0.00404188, 0.27506444, 0.42659015, 0.54322329, 0.97229569,
       0.99788432, 0.99789373]),
 'split1_train_score': array([0.00373521, 0.26666264, 0.42668838, 0.54472801, 0.97188592,
       0.99821684, 0.99823488]),
 'split2_train_score': array([0.        , 0.15437084, 0.34208662, 0.49453785, 0.96709493,
       0.99865152, 0.99865154]),
 'mean_train_score': array([0.00259237, 0.23203264, 0.39845505, 0.52749638, 0.97042551,
       0.9982509 , 0.99826005]),
 'std_train_score': array([0.00183735, 0.0550222 , 0.03985852, 0.0233133 , 0.00236101,
       0.00031413, 0.00030988])}
```

In [7]:
```
# best score and hyperparameter
print("Best Parameter:",gridsearch.best_params_)
print("Best F1 micro Score:",gridsearch.best_score_)
```

```
Best Parameter: {'estimator__C': 1}
Best F1 micro Score: 0.45869086799788344
```

In [38]:
```
# Test
# Prediction using best score LOgistic regression
predictions_2 = gridsearch.predict(x_test_multilabel)

print("Accuracy :",metrics.accuracy_score(y_test, predictions_2))
print("Hamming loss ",metrics.hamming_loss(y_test,predictions_2))

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

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

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

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

print (metrics.classification_report(y_test, predictions_2))
```

```
Accuracy : 0.1743
Hamming loss  0.0033586
Micro-average quality numbers
Precision: 0.5828, Recall: 0.3650, F1-measure: 0.4488
Macro-average quality numbers
Precision: 0.4206, Recall: 0.2851, F1-measure: 0.3288
              precision    recall  f1-score   support

           0       0.69      0.39      0.50       820
           1       0.53      0.32      0.40      1931
           2       0.37      0.18      0.24       544
           3       0.51      0.21      0.29       222
           4       0.70      0.51      0.59      1311
           5       0.75      0.52      0.61      1014
           6       0.66      0.44      0.53      1374
           7       0.73      0.59      0.65       702
           8       0.90      0.63      0.74      1424
           9       0.75      0.64      0.69      1037
          10       0.72      0.57      0.64       797
          11       0.55      0.38      0.45       156
          12       0.57      0.36      0.44        36
          13       0.67      0.41      0.51       610
          14       0.41      0.24      0.30       405
          15       0.61      0.24      0.34       144
          16       0.49      0.25      0.33       425
          17       0.58      0.32      0.41       485
          18       0.73      0.65      0.69       269
          19       0.85      0.61      0.71       518
          20       0.49      0.29      0.36       529
          21       0.82      0.57      0.68       294
          22       0.80      0.41      0.54       520
          23       0.45      0.28      0.35       246
          24       0.54      0.37      0.44       312
          25       0.46      0.33      0.38       314
          26       0.56      0.30      0.39       190
          27       0.31      0.12      0.17       342
          28       0.36      0.26      0.30        96
          29       0.29      0.16      0.20        32
          30       0.64      0.47      0.54       747
          31       0.45      0.36      0.40        14
          32       0.62      0.61      0.62       166
          33       0.56      0.35      0.43       171
          34       0.54      0.31      0.39       256
          35       0.79      0.55      0.65       199
          36       0.15      0.08      0.11        60
          37       0.25      0.19      0.22       203
          38       0.68      0.49      0.57       201
          39       0.40      0.31      0.35       208
          40       0.25      0.15      0.19        13
          41       0.35      0.16      0.22       154
          42       0.35      0.30      0.33        69
          43       0.29      0.20      0.24       426
          44       0.44      0.27      0.34        77
          45       0.47      0.29      0.36       223
          46       0.51      0.33      0.40       144
          47       0.69      0.49      0.57       245
          48       0.42      0.20      0.27        91
          49       0.59      0.41      0.48       157
          50       0.90      0.67      0.77       132
          51       0.84      0.66      0.74        41
          52       0.56      0.42      0.48       124
          53       0.23      0.26      0.24        96
          54       0.18      0.12      0.15       128
          55       0.50      0.24      0.32        46
          56       0.66      0.64      0.65       151
          57       0.08      0.01      0.02        80
          58       0.32      0.17      0.22        65
          59       0.37      0.16      0.23       182
          60       0.90      0.69      0.78       148
          61       0.36      0.14      0.20       196
          62       0.27      0.21      0.24        58
          63       0.86      0.28      0.42        43
          64       0.61      0.34      0.44       197
          65       0.65      0.43      0.51        82
          66       0.69      0.54      0.61        50
          67       0.59      0.54      0.56       105
          68       0.16      0.05      0.08        98
          69       0.21      0.07      0.10       238
          70       0.25      0.09      0.13        35
          71       0.51      0.50      0.50        54
          72       0.00      0.00      0.00        25
          73       0.29      0.24      0.26        29
          74       0.20      0.10      0.14        29
          75       0.26      0.23      0.24        40
          76       0.77      0.55      0.64       105
          77       0.58      0.50      0.54        28
          78       0.18      0.09      0.12       202
          79       0.54      0.41      0.46        37
          80       0.50      0.33      0.40        15
          81       0.48      0.27      0.35        52
          82       0.38      0.28      0.32        50
          83       0.21      0.05      0.09        56
          84       0.73      0.59      0.65        54
          85       0.58      0.62      0.60        34
          86       0.23      0.20      0.21        30
          87       0.56      0.34      0.43        29
          88       0.71      0.83      0.77        24
          89       0.83      0.79      0.81       117
          90       0.11      0.06      0.08        66
          91       0.39      0.25      0.30        68
          92       0.67      0.30      0.41        67
          93       0.53      0.36      0.43        28
          94       0.42      0.29      0.34        17
          95       0.74      0.49      0.59        51
          96       0.57      0.43      0.49        53
          97       0.08      0.02      0.03        61
          98       0.08      0.03      0.04        79
```

99	0.47	0.44	0.46	18
100	0.50	0.09	0.15	11
101	0.62	0.58	0.60	207
102	0.00	0.00	0.00	6
103	0.25	0.03	0.06	30
104	0.29	0.09	0.14	54
105	0.70	0.41	0.52	39
106	0.29	0.13	0.18	70
107	0.40	0.14	0.21	14
108	0.58	0.17	0.26	66
109	0.53	0.34	0.41	50
110	0.56	0.26	0.36	87
111	0.40	0.43	0.42	51
112	0.86	0.06	0.12	291
113	0.97	0.76	0.85	49
114	0.32	0.10	0.15	110
115	0.17	0.04	0.06	28
116	0.00	0.00	0.00	5
117	0.25	0.09	0.13	56
118	0.74	0.45	0.56	125
119	0.70	0.43	0.54	44
120	0.75	0.36	0.48	42
121	0.40	0.22	0.28	55
122	0.67	0.43	0.52	68
123	0.16	0.10	0.12	82
124	0.00	0.00	0.00	0
125	0.71	0.71	0.71	7
126	0.12	0.06	0.08	18
127	0.40	0.13	0.20	31
128	0.75	0.46	0.57	13
129	0.64	0.54	0.59	50
130	0.19	0.10	0.13	91
131	0.64	0.60	0.62	35
132	0.21	0.12	0.15	26
133	0.25	0.06	0.10	32
134	0.61	0.40	0.48	35
135	0.76	0.68	0.71	37
136	0.00	0.00	0.00	55
137	0.24	0.39	0.30	41
138	0.38	0.40	0.39	15
139	0.28	0.14	0.19	99
140	0.87	0.70	0.77	86
141	0.45	0.25	0.32	53
142	0.75	0.08	0.15	36
143	0.52	0.52	0.52	66
144	0.53	0.47	0.50	64
145	0.27	0.12	0.17	25
146	0.15	0.10	0.12	125
147	0.24	0.27	0.25	15
148	0.64	0.52	0.57	48
149	0.34	0.26	0.30	65
150	0.00	0.00	0.00	11
151	0.33	0.33	0.33	15
152	0.21	0.13	0.16	52
153	0.44	0.39	0.41	18
154	0.43	0.19	0.26	16
155	0.27	0.20	0.23	20
156	0.47	0.22	0.30	121
157	0.51	0.31	0.38	107
158	0.00	0.00	0.00	15
159	0.67	0.50	0.57	105
160	0.46	0.39	0.42	69
161	0.54	0.36	0.43	56
162	0.25	0.09	0.13	47
163	0.04	0.01	0.01	121
164	0.39	0.29	0.33	41
165	0.00	0.00	0.00	229
166	0.71	0.30	0.42	98
167	0.39	0.21	0.27	33
168	0.53	0.18	0.27	44
169	0.70	0.47	0.56	45
170	0.84	0.41	0.55	51
171	0.00	0.00	0.00	18
172	0.49	0.46	0.47	48
173	0.33	0.42	0.37	12
174	0.29	0.16	0.21	62
175	0.72	0.52	0.61	44
176	0.92	0.77	0.84	30
177	0.48	0.43	0.46	30
178	0.00	0.00	0.00	0
179	1.00	1.00	1.00	1
180	0.52	0.30	0.38	40
181	0.17	0.07	0.10	44
182	0.50	0.50	0.50	2
183	0.53	0.37	0.44	75
184	0.25	0.25	0.25	4
185	0.50	0.23	0.32	64
186	0.23	0.25	0.24	12
187	0.95	0.64	0.76	55
188	0.78	0.66	0.71	64
189	0.41	0.15	0.22	96
190	0.11	0.05	0.06	22
191	0.81	0.17	0.28	76
192	0.68	0.42	0.52	45
193	0.86	0.43	0.57	14
194	0.55	0.46	0.50	50
195	0.82	0.45	0.58	20
196	0.81	0.63	0.71	35
197	0.50	0.33	0.40	94
198	0.00	0.00	0.00	14
199	0.12	0.04	0.06	25
200	0.50	0.17	0.25	54
201	0.25	0.09	0.13	22
202	0.30	0.26	0.28	43
203	0.14	0.02	0.04	43
204	0.98	0.65	0.78	62
205	0.00	0.00	0.00	3
206	0.14	0.07	0.09	43

207	0.33	0.14	0.20	7
208	0.20	0.12	0.15	8
209	0.56	0.12	0.20	42
210	0.22	0.40	0.29	10
211	0.35	0.20	0.25	40
212	0.73	0.48	0.58	23
213	0.00	0.00	0.00	6
214	0.63	0.47	0.54	47
215	0.38	0.13	0.19	62
216	0.62	0.42	0.50	77
217	0.18	0.09	0.12	22
218	0.33	0.33	0.33	3
219	0.06	0.04	0.05	28
220	0.67	0.07	0.13	81
221	0.24	0.13	0.17	31
222	0.17	0.06	0.09	34
223	0.96	0.45	0.61	60
224	0.33	0.30	0.32	10
225	0.75	0.60	0.67	10
226	0.75	0.76	0.76	92
227	0.89	0.62	0.73	13
228	0.40	0.15	0.22	13
229	0.85	0.77	0.80	43
230	0.29	0.17	0.21	35
231	0.00	0.00	0.00	4
232	0.25	0.10	0.14	20
233	0.47	0.32	0.38	145
234	0.84	0.56	0.67	55
235	0.00	0.00	0.00	2
236	0.27	0.08	0.12	37
237	0.69	0.42	0.52	90
238	0.50	0.19	0.27	58
239	0.50	0.25	0.33	20
240	0.89	0.69	0.78	61
241	0.78	0.74	0.76	42
242	0.56	0.63	0.59	30
243	0.80	0.53	0.64	66
244	0.56	0.24	0.33	42
245	0.10	0.06	0.08	31
246	1.00	0.50	0.67	6
247	0.25	0.17	0.20	18
248	0.83	0.57	0.67	51
249	0.57	0.47	0.52	17
250	0.62	0.59	0.60	22
251	0.76	0.42	0.54	52
252	0.62	0.17	0.27	29
253	0.11	0.07	0.09	28
254	0.00	0.00	0.00	10
255	0.20	0.20	0.20	5
256	0.25	0.33	0.29	3
257	0.64	0.34	0.44	41
258	0.23	0.10	0.14	30
259	1.00	0.67	0.80	3
260	0.00	0.00	0.00	38
261	0.00	0.00	0.00	1
262	0.46	0.32	0.37	19
263	0.00	0.00	0.00	14
264	0.07	0.05	0.06	37
265	0.11	0.11	0.11	9
266	0.20	0.24	0.22	45
267	0.64	0.55	0.59	33
268	0.81	0.81	0.81	16
269	0.47	0.46	0.46	35
270	0.60	0.27	0.37	11
271	0.00	0.00	0.00	30
272	0.30	0.38	0.33	8
273	0.12	0.14	0.13	21
274	0.46	0.31	0.37	123
275	0.38	0.25	0.30	67
276	0.89	0.80	0.84	20
277	0.00	0.00	0.00	14
278	0.33	0.16	0.21	19
279	0.73	0.67	0.70	12
280	0.00	0.00	0.00	15
281	0.92	0.65	0.76	17
282	0.94	0.71	0.81	41
283	0.64	0.47	0.54	15
284	0.51	0.34	0.41	74
285	0.38	0.13	0.20	38
286	0.13	0.12	0.13	16
287	0.29	0.07	0.11	30
288	0.90	0.64	0.75	28
289	0.00	0.00	0.00	21
290	0.80	0.59	0.68	41
291	0.10	0.08	0.09	12
292	0.41	0.29	0.34	24
293	0.47	0.35	0.40	20
294	0.21	0.22	0.21	23
295	0.00	0.00	0.00	29
296	0.25	0.11	0.15	28
297	0.36	0.29	0.32	42
298	0.19	0.06	0.09	53
299	0.00	0.00	0.00	36
300	0.21	0.12	0.15	41
301	0.53	0.46	0.49	37
302	0.88	0.58	0.70	26
303	0.20	0.09	0.13	11
304	0.33	0.23	0.27	31
305	0.33	0.29	0.31	17
306	0.40	0.22	0.29	9
307	0.50	0.33	0.40	6
308	0.00	0.00	0.00	34
309	0.61	0.40	0.48	43
310	0.06	0.03	0.04	30
311	0.33	0.14	0.20	50
312	0.20	0.04	0.07	24
313	0.38	0.14	0.21	42
314	0.42	0.23	0.29	22

315	0.00	0.00	0.00	58
316	0.50	0.10	0.17	10
317	0.38	0.26	0.31	57
318	0.57	0.40	0.47	10
319	0.17	0.09	0.12	11
320	0.14	0.18	0.16	11
321	0.25	0.12	0.17	8
322	0.56	0.41	0.47	22
323	0.79	0.79	0.79	28
324	0.63	0.58	0.60	50
325	0.29	0.11	0.16	18
326	0.18	0.09	0.12	33
327	0.16	0.18	0.17	17
328	0.36	0.17	0.23	29
329	0.33	0.14	0.20	7
330	0.42	0.50	0.45	10
331	0.19	0.20	0.19	25
332	1.00	1.00	1.00	2
333	0.71	0.45	0.56	11
334	0.00	0.00	0.00	24
335	0.33	0.20	0.25	5
336	0.17	0.03	0.05	33
337	0.45	0.17	0.24	30
338	0.91	0.71	0.80	42
339	0.22	0.08	0.11	26
340	0.48	0.36	0.41	36
341	1.00	0.54	0.70	13
342	0.60	0.55	0.57	11
343	0.43	0.30	0.35	10
344	0.14	0.05	0.07	21
345	0.00	0.00	0.00	0
346	0.00	0.00	0.00	6
347	0.00	0.00	0.00	12
348	0.17	0.08	0.11	13
349	0.88	0.29	0.44	24
350	0.75	0.44	0.56	27
351	0.36	0.09	0.15	43
352	0.25	0.03	0.06	30
353	0.45	0.41	0.43	22
354	0.19	0.10	0.13	31
355	0.62	0.80	0.70	10
356	0.00	0.00	0.00	20
357	0.65	0.65	0.65	20
358	0.61	0.39	0.48	28
359	0.59	0.48	0.53	21
360	0.22	0.08	0.12	25
361	0.52	0.49	0.50	35
362	0.78	0.69	0.74	36
363	0.36	0.24	0.29	17
364	1.00	0.46	0.63	13
365	0.20	0.05	0.08	21
366	0.43	0.17	0.24	18
367	0.43	0.06	0.11	97
368	0.57	0.45	0.50	29
369	0.60	0.75	0.67	12
370	0.22	0.15	0.18	13
371	0.20	0.11	0.14	18
372	0.67	0.33	0.44	6
373	0.40	0.33	0.36	6
374	0.62	0.17	0.26	30
375	0.19	0.19	0.19	27
376	0.20	0.04	0.06	28
377	0.00	0.00	0.00	2
378	0.20	0.25	0.22	4
379	0.25	0.05	0.09	19
380	0.20	0.20	0.20	5
381	0.58	0.39	0.47	18
382	0.50	0.41	0.45	22
383	0.20	0.06	0.10	16
384	0.57	0.31	0.40	13
385	0.50	0.17	0.25	18
386	0.77	0.91	0.83	11
387	0.46	0.45	0.46	88
388	0.10	0.08	0.09	13
389	0.00	0.00	0.00	6
390	0.00	0.00	0.00	6
391	0.95	0.71	0.81	51
392	0.00	0.00	0.00	13
393	0.50	0.35	0.41	37
394	0.00	0.00	0.00	6
395	0.00	0.00	0.00	9
396	0.00	0.00	0.00	13
397	0.75	0.50	0.60	6
398	0.52	0.41	0.46	29
399	0.89	0.73	0.80	33
400	0.33	0.10	0.15	31
401	0.56	0.10	0.17	50
402	0.85	0.61	0.71	18
403	0.17	0.14	0.15	7
404	0.64	0.62	0.63	26
405	0.84	0.82	0.83	56
406	0.60	0.75	0.67	4
407	0.08	0.06	0.07	17
408	0.75	0.55	0.63	11
409	0.10	0.06	0.07	18
410	0.33	0.30	0.32	10
411	0.40	0.13	0.20	45
412	0.82	0.45	0.58	20
413	0.40	0.16	0.23	25
414	0.00	0.00	0.00	20
415	0.00	0.00	0.00	6
416	0.20	0.12	0.15	26
417	0.60	0.30	0.40	10
418	0.00	0.00	0.00	18
419	0.75	0.50	0.60	6
420	0.56	0.53	0.55	17
421	0.00	0.00	0.00	1
422	0.00	0.00	0.00	6

423	0.00	0.00	0.00	12
424	1.00	0.50	0.67	4
425	0.60	0.27	0.37	11
426	0.17	0.09	0.12	11
427	0.86	0.75	0.80	8
428	0.64	0.27	0.38	26
429	0.52	0.62	0.57	40
430	0.00	0.00	0.00	2
431	0.08	0.03	0.04	35
432	0.71	0.33	0.45	15
433	0.00	0.00	0.00	18
434	0.00	0.00	0.00	0
435	0.00	0.00	0.00	0
436	0.38	0.18	0.24	28
437	0.33	0.15	0.21	33
438	0.79	0.55	0.65	20
439	0.27	0.08	0.13	36
440	0.17	0.11	0.13	18
441	0.42	0.56	0.48	18
442	0.69	0.69	0.69	16
443	0.00	0.00	0.00	22
444	0.00	0.00	0.00	6
445	0.72	0.62	0.67	21
446	0.79	0.57	0.66	46
447	0.19	0.06	0.09	69
448	0.00	0.00	0.00	7
449	0.00	0.00	0.00	3
450	0.12	0.04	0.06	52
451	0.14	0.06	0.09	16
452	0.89	0.94	0.91	17
453	0.00	0.00	0.00	13
454	0.33	0.18	0.24	11
455	0.00	0.00	0.00	12
456	0.12	0.17	0.14	6
457	0.22	0.11	0.15	18
458	0.00	0.00	0.00	15
459	0.95	0.64	0.77	28
460	0.14	0.06	0.08	18
461	0.57	0.40	0.47	10
462	0.40	0.08	0.14	24
463	0.00	0.00	0.00	18
464	0.96	0.62	0.75	39
465	0.21	0.27	0.24	11
466	0.18	0.06	0.09	35
467	0.07	0.05	0.06	21
468	0.25	0.03	0.05	37
469	0.00	0.00	0.00	5
470	0.00	0.00	0.00	8
471	0.53	0.27	0.36	37
472	0.00	0.00	0.00	47
473	0.46	0.43	0.44	14
474	0.88	0.65	0.75	23
475	0.51	0.50	0.50	66
476	0.00	0.00	0.00	3
477	0.70	0.37	0.48	19
478	0.00	0.00	0.00	1
479	0.17	0.13	0.15	23
480	0.54	0.12	0.19	60
481	0.33	0.15	0.21	26
482	0.50	0.50	0.50	4
483	0.00	0.00	0.00	8
484	0.91	0.43	0.59	23
485	0.64	0.39	0.48	18
486	0.62	0.42	0.50	12
487	0.79	0.38	0.51	29
488	1.00	1.00	1.00	1
489	0.00	0.00	0.00	6
490	0.33	0.14	0.20	7
491	0.00	0.00	0.00	3
492	0.12	0.20	0.15	10
493	0.38	0.32	0.34	19
494	0.20	0.14	0.17	7
495	0.43	0.38	0.40	8
496	0.39	0.39	0.39	18
497	0.38	0.12	0.19	72
498	0.17	0.12	0.14	8
499	0.50	0.41	0.45	32
micro avg	0.58	0.36	0.45	37472
macro avg	0.42	0.29	0.33	37472
weighted avg	0.55	0.36	0.43	37472
samples avg	0.43	0.36	0.36	37472

3. Try OneVsRestClassifier with Linear-SVM (SGDClassifier with loss-hinge) USING GridSearch

In [17]:
```
# Train
from sklearn.model_selection import GridSearchCV
start = datetime.now()

param={"estimator__alpha":[0.0001,0.001,0.01,0.1,1,10,100]} # When you use nested estimators with grid search you can scope the parameters wit
h __ as a separator

lr_ovr = OneVsRestClassifier(SGDClassifier(loss='hinge', penalty='l1',n_jobs=-1),)
classifier_2=GridSearchCV(estimator=lr_ovr, param_grid=param ,scoring="f1_micro", return_train_score=True)
classifier_2.fit(x_train_multilabel, y_train)

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

```
Time taken to run this cell : 2:08:38.593076
```

In [19]:
```
# Saving Result
import pickle
"""file=open("gridsearch_svm.pkl",'wb')
pickle.dump(classifier_2,file)
file.close()
"""
file=open("gridsearch_svm.pkl",'rb')
gridsearch_svm=pickle.load(file)
```

In [20]:
```
# Test
# Prediction using best score SVM
predictions_2 = gridsearch_svm.predict(x_test_multilabel)

print("Accuracy :",metrics.accuracy_score(y_test, predictions_2))
print("Hamming loss ",metrics.hamming_loss(y_test,predictions_2))

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

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

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

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

print (metrics.classification_report(y_test, predictions_2))
```

```
Accuracy : 0.13035
Hamming loss  0.0036607
Micro-average quality numbers
Precision: 0.5228, Recall: 0.2643, F1-measure: 0.3511
Macro-average quality numbers
Precision: 0.2781, Recall: 0.2035, F1-measure: 0.2115
              precision    recall  f1-score   support

           0       0.64      0.39      0.49       820
           1       0.66      0.03      0.06      1931
           2       0.38      0.16      0.23       544
           3       0.56      0.33      0.41       222
           4       0.70      0.40      0.51      1311
           5       0.79      0.53      0.63      1014
           6       0.73      0.40      0.52      1374
           7       0.74      0.54      0.62       702
           8       0.77      0.57      0.66      1424
           9       0.73      0.19      0.30      1037
          10       0.69      0.41      0.51       797
          11       0.46      0.43      0.44       156
          12       0.38      0.33      0.35        36
          13       0.74      0.40      0.52       610
          14       0.39      0.02      0.03       405
          15       0.41      0.22      0.28       144
          16       0.65      0.21      0.32       425
          17       0.62      0.18      0.27       485
          18       0.48      0.42      0.45       269
          19       0.84      0.62      0.71       518
          20       0.50      0.24      0.32       529
          21       0.64      0.68      0.66       294
          22       0.80      0.48      0.60       520
          23       0.55      0.26      0.35       246
          24       0.41      0.06      0.11       312
          25       0.71      0.02      0.03       314
          26       0.55      0.37      0.44       190
          27       0.00      0.00      0.00       342
          28       0.22      0.21      0.21        96
          29       0.38      0.16      0.22        32
          30       0.39      0.09      0.15       747
          31       0.32      0.57      0.41        14
          32       0.40      0.55      0.46       166
          33       0.59      0.30      0.40       171
          34       0.49      0.31      0.38       256
          35       0.62      0.65      0.64       199
          36       0.00      0.00      0.00        60
          37       0.00      0.00      0.00       203
          38       0.50      0.31      0.38       201
          39       0.00      0.00      0.00       208
          40       0.05      0.08      0.06        13
          41       0.46      0.15      0.23       154
          42       0.26      0.14      0.19        69
          43       0.00      0.00      0.00       426
          44       0.00      0.00      0.00        77
          45       0.33      0.06      0.11       223
          46       0.68      0.22      0.34       144
          47       0.76      0.50      0.60       245
          48       0.50      0.12      0.19        91
          49       0.38      0.31      0.34       157
          50       0.82      0.77      0.79       132
          51       0.72      0.68      0.70        41
          52       0.44      0.50      0.47       124
          53       0.00      0.00      0.00        96
          54       0.00      0.00      0.00       128
          55       0.62      0.50      0.55        46
          56       0.60      0.58      0.59       151
          57       0.00      0.00      0.00        80
          58       0.00      0.00      0.00        65
          59       0.00      0.00      0.00       182
          60       0.77      0.70      0.73       148
          61       0.22      0.17      0.19       196
          62       0.00      0.00      0.00        58
          63       0.75      0.35      0.48        43
          64       0.30      0.29      0.30       197
          65       0.47      0.52      0.50        82
          66       0.00      0.00      0.00        50
          67       0.49      0.68      0.57       105
          68       0.00      0.00      0.00        98
          69       0.00      0.00      0.00       238
          70       0.09      0.06      0.07        35
          71       0.31      0.26      0.28        54
          72       0.00      0.00      0.00        25
          73       0.44      0.14      0.21        29
          74       0.00      0.00      0.00        29
          75       0.53      0.20      0.29        40
          76       0.66      0.70      0.68       105
          77       0.27      0.25      0.26        28
          78       0.33      0.07      0.11       202
          79       0.40      0.51      0.45        37
          80       0.15      0.40      0.22        15
          81       0.30      0.37      0.33        52
          82       0.33      0.26      0.29        50
          83       0.00      0.00      0.00        56
          84       0.55      0.54      0.54        54
          85       0.53      0.26      0.35        34
          86       0.24      0.23      0.24        30
          87       0.23      0.10      0.14        29
          88       0.42      0.67      0.52        24
          89       0.73      0.72      0.72       117
          90       0.00      0.00      0.00        66
          91       0.34      0.21      0.26        68
          92       0.67      0.12      0.20        67
          93       0.32      0.36      0.34        28
          94       0.14      0.29      0.19        17
          95       0.33      0.55      0.41        51
          96       0.62      0.34      0.44        53
          97       0.00      0.00      0.00        61
          98       0.00      0.00      0.00        79
```

99	0.06	0.22	0.09	18
100	0.09	0.27	0.13	11
101	0.56	0.55	0.55	207
102	0.00	0.00	0.00	6
103	0.67	0.07	0.12	30
104	0.00	0.00	0.00	54
105	0.31	0.46	0.37	39
106	0.00	0.00	0.00	70
107	0.75	0.21	0.33	14
108	0.27	0.36	0.31	66
109	0.18	0.42	0.26	50
110	0.54	0.08	0.14	87
111	0.27	0.43	0.33	51
112	0.00	0.00	0.00	291
113	0.83	0.80	0.81	49
114	0.08	0.01	0.02	110
115	0.09	0.11	0.10	28
116	0.00	0.00	0.00	5
117	0.00	0.00	0.00	56
118	0.76	0.49	0.60	125
119	0.50	0.32	0.39	44
120	0.71	0.12	0.20	42
121	0.20	0.29	0.24	55
122	0.71	0.43	0.53	68
123	0.00	0.00	0.00	82
124	0.00	0.00	0.00	0
125	0.83	0.71	0.77	7
126	0.09	0.22	0.12	18
127	0.00	0.00	0.00	31
128	0.60	0.46	0.52	13
129	0.58	0.60	0.59	50
130	0.00	0.00	0.00	91
131	0.36	0.57	0.44	35
132	0.06	0.19	0.09	26
133	0.00	0.00	0.00	32
134	0.50	0.17	0.26	35
135	0.61	0.62	0.61	37
136	0.00	0.00	0.00	55
137	0.17	0.44	0.24	41
138	0.00	0.00	0.00	15
139	0.15	0.08	0.11	99
140	0.88	0.51	0.65	86
141	0.56	0.19	0.28	53
142	0.18	0.17	0.17	36
143	0.55	0.52	0.53	66
144	0.69	0.38	0.48	64
145	0.00	0.00	0.00	25
146	0.00	0.00	0.00	125
147	0.00	0.00	0.00	15
148	0.51	0.58	0.54	48
149	0.18	0.25	0.21	65
150	0.02	0.18	0.04	11
151	0.14	0.33	0.20	15
152	0.00	0.00	0.00	52
153	0.10	0.17	0.13	18
154	1.00	0.06	0.12	16
155	0.00	0.00	0.00	20
156	0.27	0.22	0.24	121
157	0.44	0.52	0.48	107
158	0.00	0.00	0.00	15
159	0.73	0.29	0.41	105
160	0.42	0.22	0.29	69
161	0.29	0.12	0.18	56
162	0.00	0.00	0.00	47
163	0.00	0.00	0.00	121
164	0.39	0.29	0.33	41
165	0.00	0.00	0.00	229
166	0.00	0.00	0.00	98
167	0.78	0.21	0.33	33
168	0.08	0.05	0.06	44
169	0.51	0.58	0.54	45
170	0.83	0.47	0.60	51
171	0.00	0.00	0.00	18
172	0.42	0.56	0.48	48
173	0.11	0.08	0.10	12
174	0.00	0.00	0.00	62
175	0.66	0.66	0.66	44
176	0.30	0.63	0.40	30
177	0.48	0.33	0.39	30
178	0.00	0.00	0.00	0
179	1.00	1.00	1.00	1
180	0.34	0.35	0.35	40
181	0.00	0.00	0.00	44
182	0.05	0.50	0.09	2
183	0.36	0.19	0.25	75
184	0.50	0.25	0.33	4
185	0.33	0.08	0.13	64
186	0.29	0.33	0.31	12
187	0.88	0.55	0.67	55
188	0.71	0.39	0.51	64
189	0.00	0.00	0.00	96
190	0.10	0.14	0.11	22
191	0.56	0.12	0.20	76
192	0.29	0.31	0.30	45
193	0.25	0.07	0.11	14
194	0.47	0.76	0.58	50
195	0.86	0.30	0.44	20
196	0.79	0.74	0.76	35
197	0.55	0.06	0.11	94
198	0.00	0.00	0.00	14
199	0.00	0.00	0.00	25
200	0.00	0.00	0.00	54
201	0.25	0.18	0.21	22
202	0.00	0.00	0.00	43
203	0.25	0.02	0.04	43
204	0.87	0.42	0.57	62
205	0.00	0.00	0.00	3
206	0.00	0.00	0.00	43

207	0.00	0.00	0.00	7
208	0.04	0.12	0.06	8
209	0.00	0.00	0.00	42
210	0.19	0.40	0.26	10
211	0.26	0.35	0.30	40
212	0.28	0.35	0.31	23
213	0.00	0.00	0.00	6
214	0.44	0.57	0.50	47
215	0.00	0.00	0.00	62
216	0.00	0.00	0.00	77
217	0.57	0.18	0.28	22
218	0.00	0.00	0.00	3
219	0.04	0.07	0.05	28
220	0.29	0.05	0.08	81
221	0.00	0.00	0.00	31
222	0.00	0.00	0.00	34
223	0.86	0.30	0.44	60
224	0.00	0.00	0.00	10
225	0.50	0.90	0.64	10
226	0.73	0.82	0.77	92
227	0.00	0.00	0.00	13
228	0.40	0.15	0.22	13
229	0.56	0.63	0.59	43
230	0.28	0.31	0.30	35
231	0.00	0.00	0.00	4
232	0.32	0.30	0.31	20
233	0.00	0.00	0.00	145
234	0.59	0.58	0.59	55
235	0.00	0.00	0.00	2
236	0.00	0.00	0.00	37
237	0.69	0.39	0.50	90
238	0.00	0.00	0.00	58
239	0.80	0.20	0.32	20
240	0.93	0.44	0.60	61
241	0.76	0.81	0.78	42
242	0.14	0.83	0.24	30
243	0.66	0.56	0.61	66
244	0.42	0.36	0.38	42
245	0.00	0.00	0.00	31
246	0.06	0.33	0.10	6
247	0.00	0.00	0.00	18
248	0.46	0.55	0.50	51
249	0.21	0.29	0.24	17
250	0.52	0.50	0.51	22
251	0.63	0.52	0.57	52
252	0.00	0.00	0.00	29
253	0.00	0.00	0.00	28
254	0.00	0.00	0.00	10
255	0.04	0.20	0.06	5
256	0.11	0.67	0.19	3
257	0.18	0.34	0.24	41
258	0.00	0.00	0.00	30
259	0.20	0.33	0.25	3
260	0.00	0.00	0.00	38
261	0.00	0.00	0.00	1
262	0.33	0.26	0.29	19
263	0.00	0.00	0.00	14
264	0.00	0.00	0.00	37
265	0.00	0.00	0.00	9
266	0.00	0.00	0.00	45
267	0.33	0.30	0.32	33
268	0.32	0.62	0.43	16
269	0.42	0.71	0.53	35
270	0.20	0.36	0.26	11
271	0.00	0.00	0.00	30
272	0.00	0.00	0.00	8
273	0.00	0.00	0.00	21
274	0.00	0.00	0.00	123
275	0.29	0.24	0.26	67
276	0.55	0.55	0.55	20
277	0.00	0.00	0.00	14
278	0.00	0.00	0.00	19
279	0.55	0.50	0.52	12
280	0.00	0.00	0.00	15
281	0.65	0.65	0.65	17
282	0.84	0.76	0.79	41
283	0.00	0.00	0.00	15
284	0.79	0.15	0.25	74
285	0.16	0.16	0.16	38
286	0.00	0.00	0.00	16
287	0.00	0.00	0.00	30
288	0.80	0.57	0.67	28
289	0.00	0.00	0.00	21
290	0.88	0.71	0.78	41
291	0.00	0.00	0.00	12
292	0.00	0.00	0.00	24
293	0.16	0.50	0.25	20
294	0.04	0.04	0.04	23
295	0.00	0.00	0.00	29
296	0.00	0.00	0.00	28
297	0.00	0.00	0.00	42
298	0.00	0.00	0.00	53
299	0.00	0.00	0.00	36
300	0.00	0.00	0.00	41
301	0.00	0.00	0.00	37
302	0.68	0.50	0.58	26
303	0.40	0.18	0.25	11
304	0.00	0.00	0.00	31
305	0.33	0.18	0.23	17
306	0.00	0.00	0.00	9
307	0.20	0.17	0.18	6
308	0.00	0.00	0.00	34
309	0.50	0.26	0.34	43
310	0.00	0.00	0.00	30
311	0.00	0.00	0.00	50
312	0.00	0.00	0.00	24
313	0.00	0.00	0.00	42
314	0.43	0.14	0.21	22

315	0.00	0.00	0.00	58
316	0.80	0.40	0.53	10
317	0.28	0.32	0.30	57
318	1.00	0.10	0.18	10
319	0.00	0.00	0.00	11
320	0.07	0.09	0.08	11
321	0.33	0.12	0.18	8
322	0.50	0.36	0.42	22
323	0.83	0.36	0.50	28
324	0.58	0.44	0.50	50
325	0.00	0.00	0.00	18
326	0.00	0.00	0.00	33
327	0.07	0.12	0.09	17
328	0.00	0.00	0.00	29
329	1.00	0.29	0.44	7
330	0.44	0.40	0.42	10
331	0.00	0.00	0.00	25
332	0.18	1.00	0.31	2
333	0.43	0.55	0.48	11
334	0.00	0.00	0.00	24
335	1.00	0.20	0.33	5
336	0.00	0.00	0.00	33
337	0.18	0.10	0.13	30
338	0.94	0.69	0.79	42
339	0.00	0.00	0.00	26
340	0.30	0.39	0.34	36
341	1.00	0.46	0.63	13
342	0.00	0.00	0.00	11
343	0.12	0.10	0.11	10
344	0.00	0.00	0.00	21
345	0.00	0.00	0.00	0
346	0.00	0.00	0.00	6
347	0.00	0.00	0.00	12
348	0.00	0.00	0.00	13
349	0.00	0.00	0.00	24
350	0.33	0.37	0.35	27
351	0.00	0.00	0.00	43
352	0.00	0.00	0.00	30
353	0.43	0.27	0.33	22
354	0.00	0.00	0.00	31
355	0.80	0.40	0.53	10
356	0.60	0.15	0.24	20
357	0.54	0.75	0.63	20
358	0.35	0.46	0.40	28
359	0.30	0.43	0.35	21
360	0.00	0.00	0.00	25
361	0.50	0.31	0.39	35
362	0.71	0.42	0.53	36
363	0.25	0.18	0.21	17
364	1.00	0.15	0.27	13
365	0.00	0.00	0.00	21
366	0.00	0.00	0.00	18
367	0.00	0.00	0.00	97
368	0.35	0.45	0.39	29
369	1.00	0.42	0.59	12
370	0.00	0.00	0.00	13
371	0.06	0.11	0.08	18
372	0.00	0.00	0.00	6
373	0.14	0.33	0.20	6
374	0.00	0.00	0.00	30
375	0.00	0.00	0.00	27
376	0.00	0.00	0.00	28
377	0.00	0.00	0.00	2
378	0.00	0.00	0.00	4
379	0.00	0.00	0.00	19
380	0.00	0.00	0.00	5
381	0.77	0.56	0.65	18
382	0.16	0.23	0.19	22
383	0.01	0.06	0.01	16
384	0.00	0.00	0.00	13
385	0.00	0.00	0.00	18
386	0.67	0.36	0.47	11
387	0.00	0.00	0.00	88
388	0.00	0.00	0.00	13
389	1.00	0.17	0.29	6
390	0.00	0.00	0.00	6
391	1.00	0.20	0.33	51
392	0.00	0.00	0.00	13
393	0.39	0.32	0.35	37
394	0.00	0.00	0.00	6
395	0.00	0.00	0.00	9
396	0.00	0.00	0.00	13
397	0.43	0.50	0.46	6
398	0.48	0.34	0.40	29
399	0.80	0.48	0.60	33
400	0.00	0.00	0.00	31
401	0.00	0.00	0.00	50
402	0.93	0.78	0.85	18
403	0.50	0.29	0.36	7
404	0.57	0.65	0.61	26
405	1.00	0.21	0.35	56
406	1.00	0.75	0.86	4
407	0.00	0.00	0.00	17
408	0.40	0.18	0.25	11
409	0.00	0.00	0.00	18
410	0.40	0.20	0.27	10
411	0.00	0.00	0.00	45
412	0.00	0.00	0.00	20
413	0.00	0.00	0.00	25
414	0.00	0.00	0.00	20
415	0.12	0.33	0.18	6
416	0.11	0.12	0.11	26
417	0.00	0.00	0.00	10
418	0.00	0.00	0.00	18
419	0.00	0.00	0.00	6
420	0.47	0.53	0.50	17
421	0.00	0.00	0.00	1
422	0.00	0.00	0.00	6

423	0.00	0.00	0.00	12
424	1.00	0.25	0.40	4
425	0.00	0.00	0.00	11
426	0.00	0.00	0.00	11
427	0.50	0.25	0.33	8
428	0.50	0.27	0.35	26
429	0.40	0.15	0.22	40
430	0.00	0.00	0.00	2
431	0.00	0.00	0.00	35
432	1.00	0.20	0.33	15
433	0.00	0.00	0.00	18
434	0.00	0.00	0.00	0
435	0.00	0.00	0.00	0
436	0.00	0.00	0.00	28
437	0.32	0.21	0.25	33
438	1.00	0.10	0.18	20
439	0.00	0.00	0.00	36
440	0.00	0.00	0.00	18
441	0.27	0.44	0.33	18
442	0.73	0.50	0.59	16
443	0.12	0.14	0.13	22
444	0.00	0.00	0.00	6
445	0.64	0.33	0.44	21
446	0.68	0.89	0.77	46
447	0.00	0.00	0.00	69
448	0.00	0.00	0.00	7
449	0.00	0.00	0.00	3
450	0.00	0.00	0.00	52
451	0.02	0.06	0.03	16
452	1.00	0.41	0.58	17
453	0.00	0.00	0.00	13
454	0.25	0.27	0.26	11
455	0.08	0.08	0.08	12
456	0.00	0.00	0.00	6
457	0.00	0.00	0.00	18
458	0.00	0.00	0.00	15
459	0.67	0.07	0.13	28
460	0.00	0.00	0.00	18
461	0.00	0.00	0.00	10
462	0.00	0.00	0.00	24
463	0.00	0.00	0.00	18
464	0.00	0.00	0.00	39
465	0.08	0.09	0.08	11
466	0.00	0.00	0.00	35
467	0.10	0.14	0.12	21
468	0.00	0.00	0.00	37
469	0.00	0.00	0.00	5
470	0.00	0.00	0.00	8
471	0.75	0.24	0.37	37
472	0.00	0.00	0.00	47
473	0.00	0.00	0.00	14
474	0.83	0.43	0.57	23
475	0.00	0.00	0.00	66
476	0.50	0.33	0.40	3
477	0.38	0.42	0.40	19
478	0.00	0.00	0.00	1
479	0.00	0.00	0.00	23
480	0.00	0.00	0.00	60
481	0.00	0.00	0.00	26
482	0.00	0.00	0.00	4
483	0.00	0.00	0.00	8
484	0.56	0.22	0.31	23
485	0.31	0.22	0.26	18
486	0.70	0.58	0.64	12
487	0.00	0.00	0.00	29
488	0.00	0.00	0.00	1
489	0.50	0.17	0.25	6
490	0.00	0.00	0.00	7
491	0.00	0.00	0.00	3
492	0.30	0.30	0.30	10
493	0.18	0.16	0.17	19
494	0.00	0.00	0.00	7
495	0.40	0.25	0.31	8
496	0.29	0.11	0.16	18
497	0.00	0.00	0.00	72
498	0.14	0.25	0.18	8
499	0.60	0.28	0.38	32
micro avg	0.52	0.26	0.35	37472
macro avg	0.28	0.20	0.21	37472
weighted avg	0.46	0.26	0.31	37472
samples avg	0.34	0.26	0.27	37472

Observations

In [22]:
```
# Pretty table
from prettytable import PrettyTable
x=PrettyTable(["model","vectoriser","micro_f1","accuracy","precision","recall","hamming_loss"])

x.add_row(["LR without hypertune","ngram=(1,4)","0.4489","0.174","0.583","0.3649","0.0033"])
x.add_row(["LR with hypertune","ngram=(1,4)","0.4488","0.1743","0.5828","0.3650","0.0033"])
x.add_row(["SVM without hypertune","ngram=(1,4)","0.3511","0.1303","0.5228","0.2035","0.0036"])

print("NOTE: These results are from only 100k datapints (80k train and 20k test) and ony 100k features(dimention)")
print("*"*100,"\n")
print("*"*30,"pretty table","*"*30)
print(x)
```

```
NOTE: These results are from only 100k datapints (80k train and 20k test) and ony 100k features(dim)
****************************************************************************************************

****************************** pretty table ******************************
+----------------------+-------------+----------+----------+-----------+--------+-------------+
|         model        |  vectoriser | micro_f1 | accuracy | precision | recall | hamming_loss |
+----------------------+-------------+----------+----------+-----------+--------+-------------+
|  LR without hypertune | ngram=(1,4) |  0.4489  |  0.174   |   0.583   | 0.3649 |    0.0033    |
|   LR with hypertune   | ngram=(1,4) |  0.4488  |  0.1743  |   0.5828  | 0.3650 |    0.0033    |
| SVM without hypertune | ngram=(1,4) |  0.3511  |  0.1303  |   0.5228  | 0.2035 |    0.0036    |
+----------------------+-------------+----------+----------+-----------+--------+-------------+
```