AKSHAT KAKKAD

91900133038

LHC REPORT

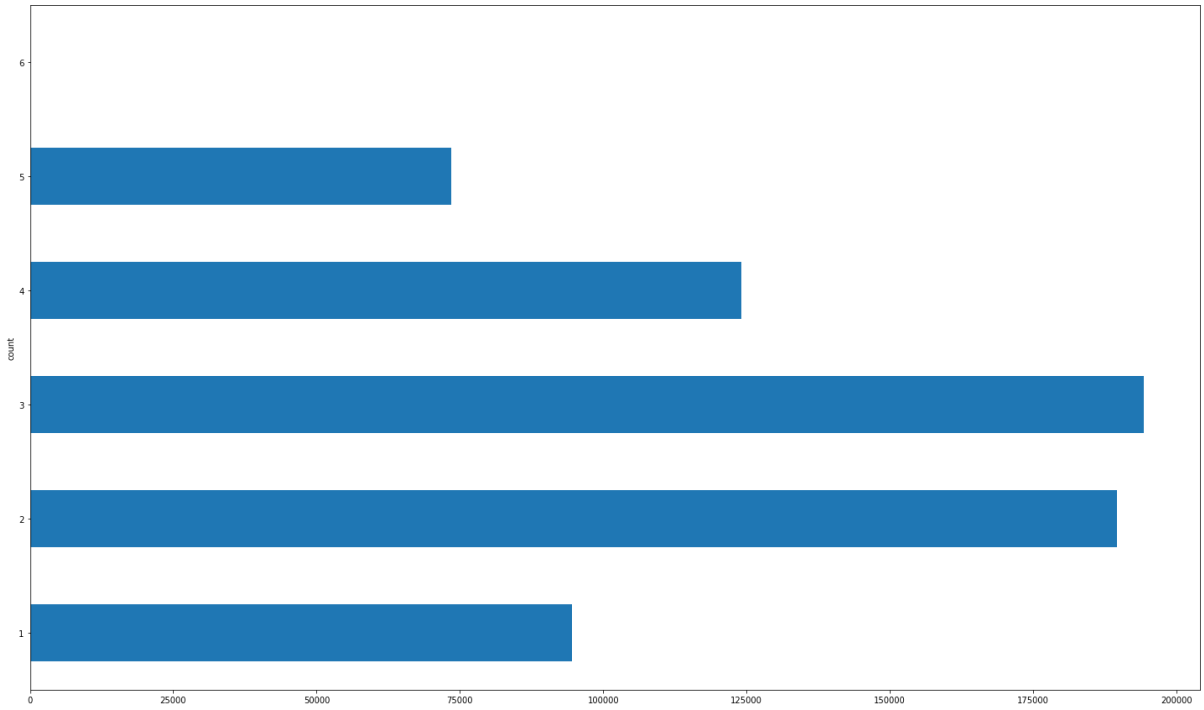
1. Determine the Number of Tags Per Question

   There are different number of tags for different questions we need to determine the distribution of number of tags in question. (Basically see what number of tags a users use for their questions).

```
[7]  set(dataset2['count'])

     {1, 2, 3, 4, 5, 6}
```

   Maximum tags in one Question => 6
   Minimum tags in one Question => 1



   Distribution of **No. of tags VS No. of Questions**

```
[9] freq
```

```
count
1     94523
2    189704
3    194334
4    124127
5     73490
6        26
dtype: int64
```

(This means that there are 94523 questions in dataset having tag count = 1)

2. Determine the Total Number of Unique Tags
   This question helps us to find that what are the total topics which stack overflow covers for its users.

```
total = len(unique)
print("Total Number of Unique Tags ", total)
```

```
Total Number of Unique Tags  25310
```

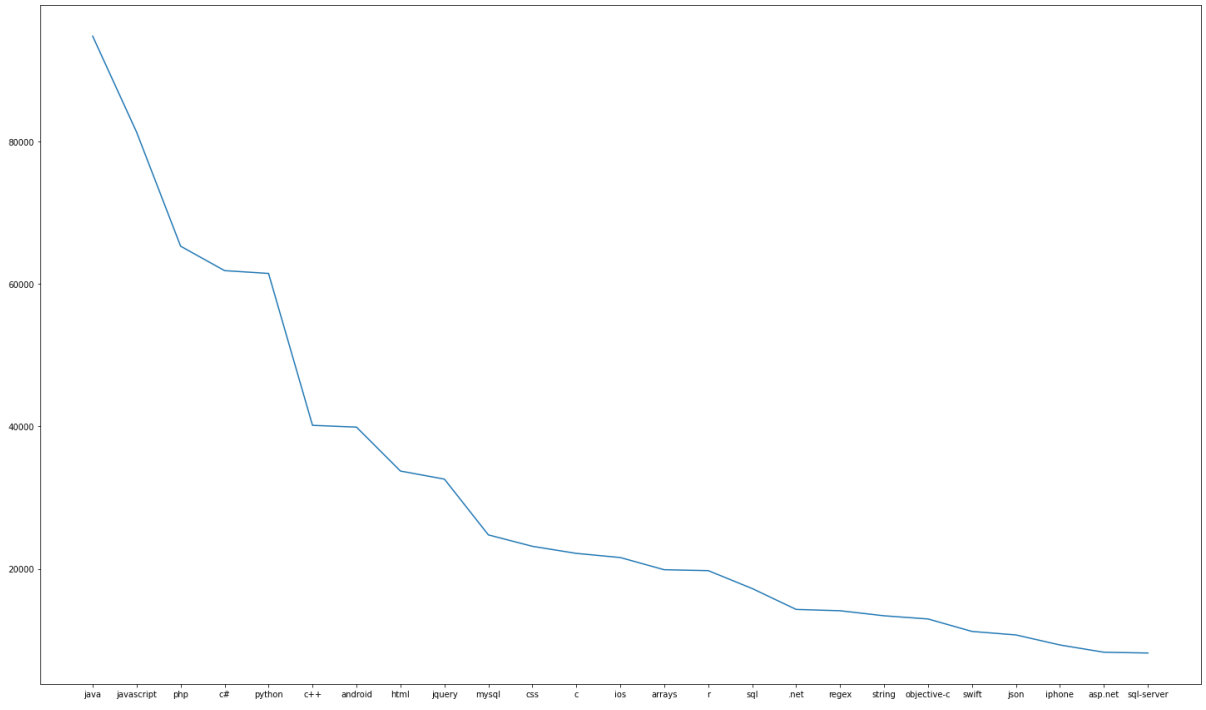
There is total 25310 different topics whose questions are covered by stack Overflow.

3. Determine the top-25 Tags appearing frequently
   Here we are determining most popular 25 topics that are discussed in stack Overflow

```
print("The top 25 Tags in Stack Overflow are : - ")
for i in sortunique:
  print(i)
  top25+=1
  if top25==25:
    break
```

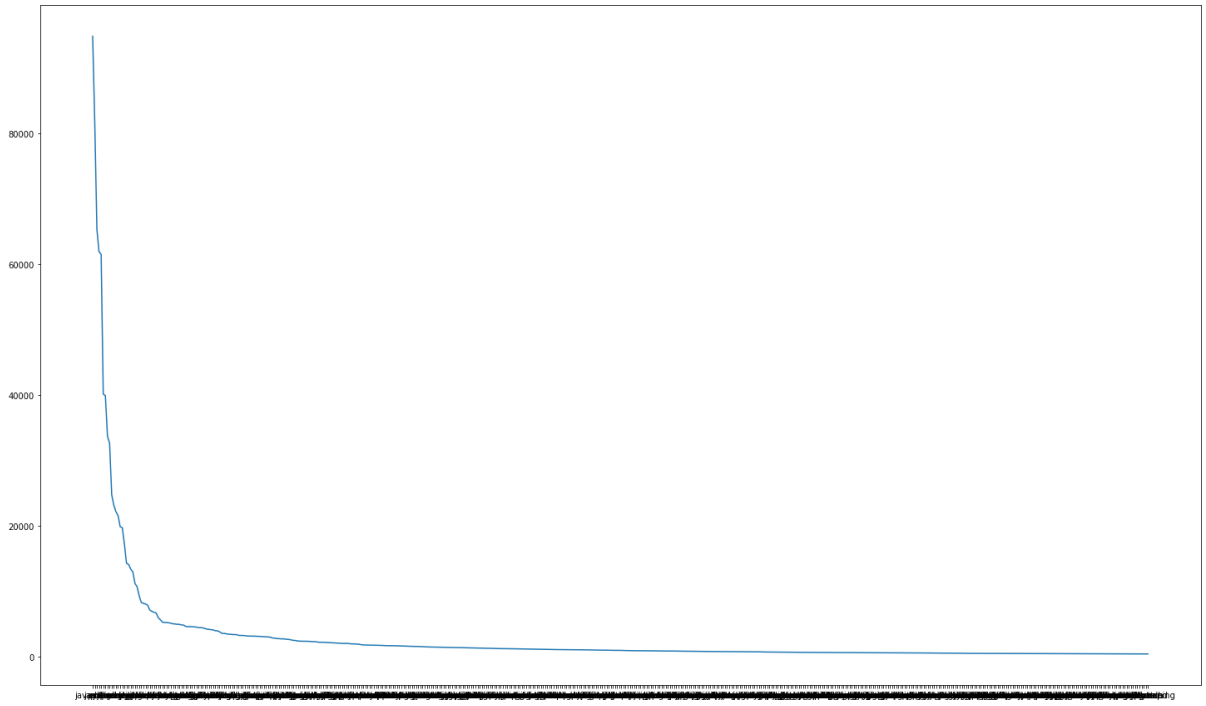
```
The top 25 Tags in Stack Overflow are : -
java
javascript
php
c#
python
c++
android
html
jquery
mysql
css
c
ios
arrays
r
sql
.net
regex
string
objective-c
swift
json
iphone
asp.net
sql-server
```

List of top 25 topics covered in stack overflow



This shows the graph of **TOP25 tags VS No. of Occurrences**

4. Determine the nature of the distribution of top-500 tags
   Here we are collecting information about top500 topics in stack Overflow



This shows the graph of **TOP500 tags VS No. of Occurrences**

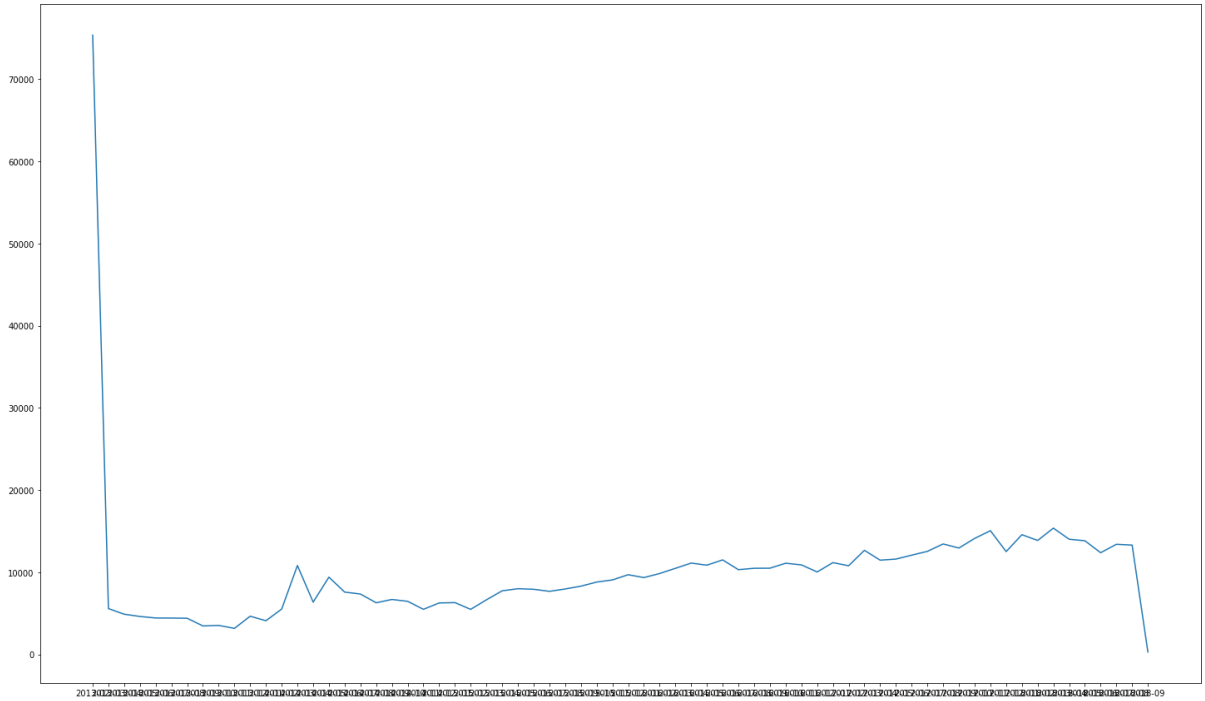
```
print(tags500)
```

```
gates', 'c-preprocessor', 'https', 'include', 'passwords', 'nullreferenceexception', 'datagridview', 'android-edittext', 'hex', 'filter', 'paral
```

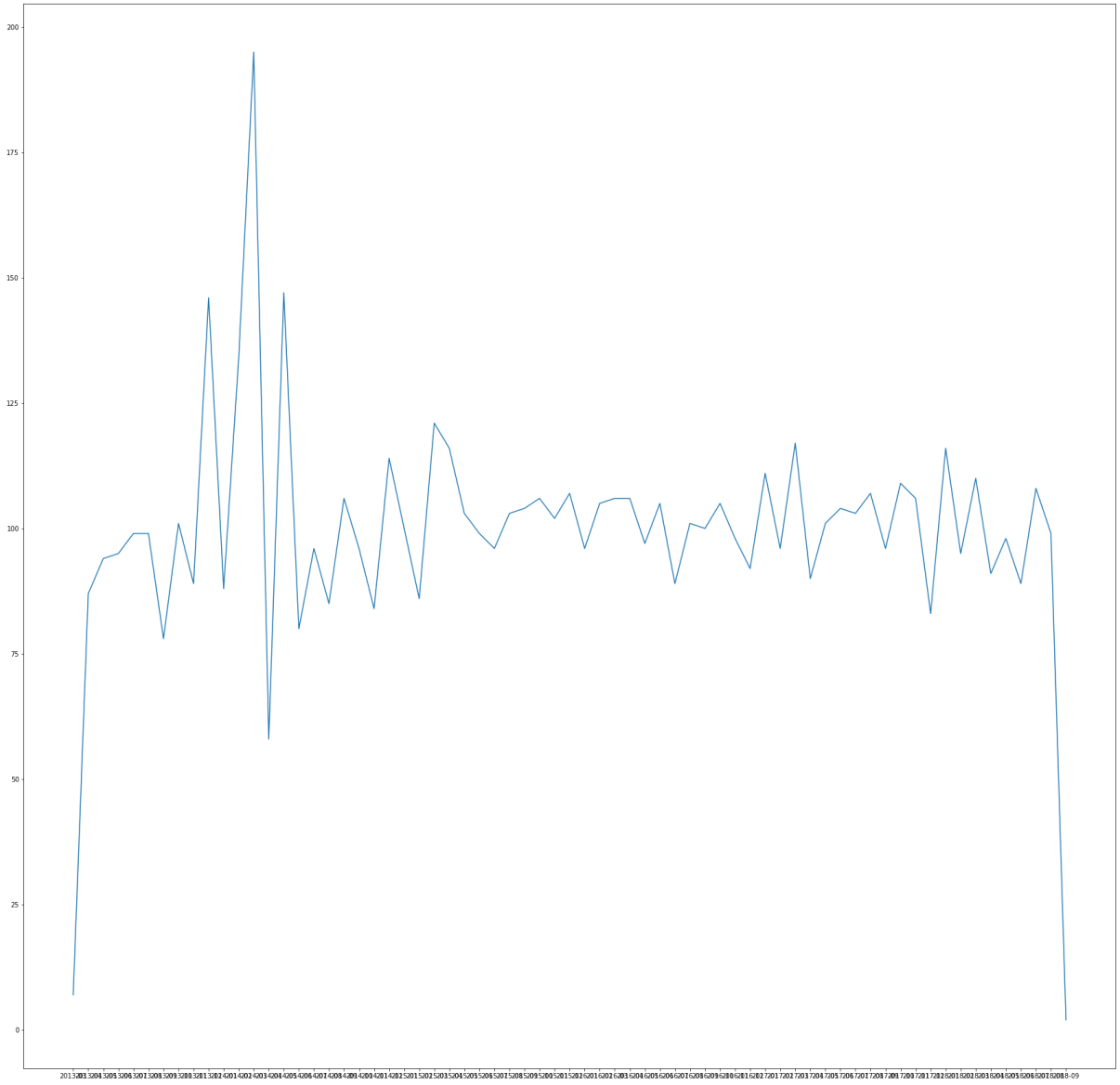
List of top500 tags printed in colab.

5. Determine the ratio of duplicate questions asked in each month

   Consider a ratio of Duplicate questions asked with respect to previous month



Number of Duplicate Questions in one month

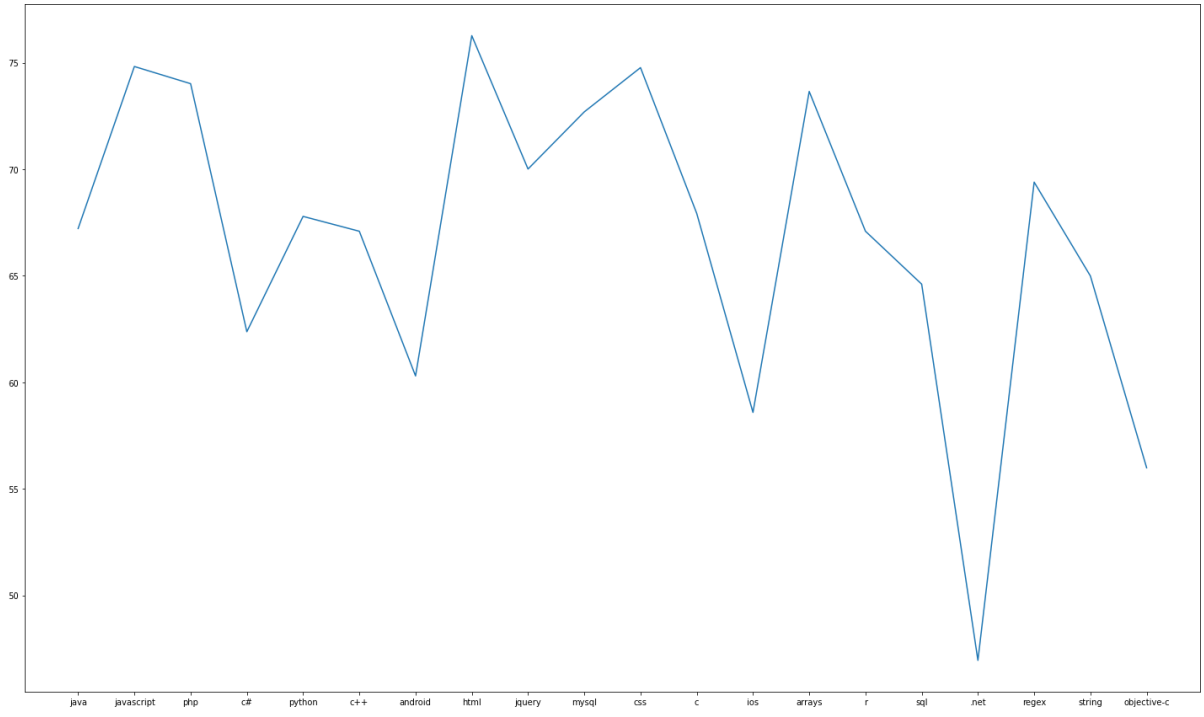


Ratio of Previous Month

6. Determine the percentage of duplicate questions associated with different tags

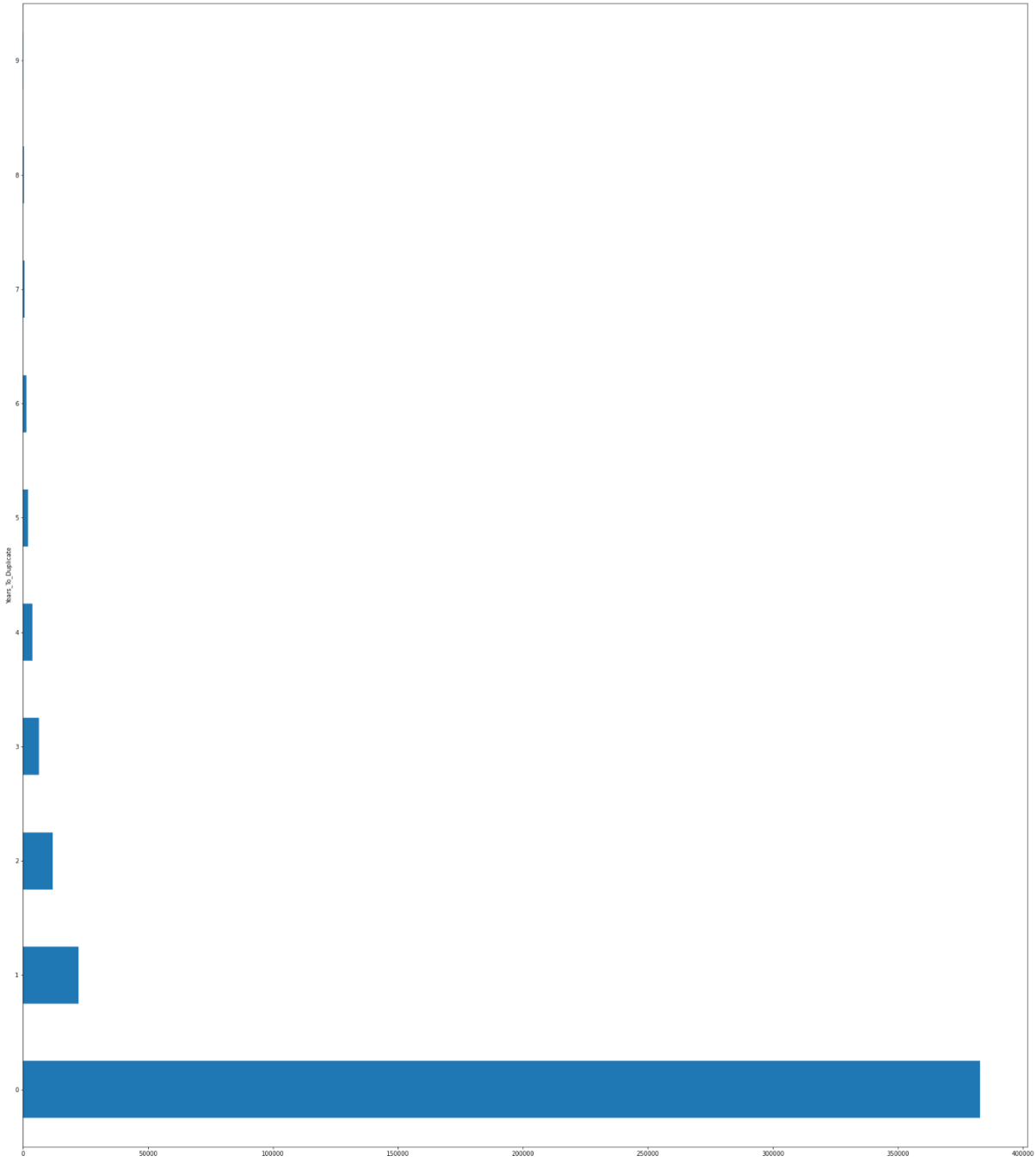
   Here we need to determine that which topic has how much percentage of duplicate questions. Like for example JAVA is included as tag in 94846 questions and in duplicate questions JAVA tag is included 63746 questions. Therefore, we can say that 67 % tags come from Duplicate Questions.

   Analysis of top 20 Tags is shown in Graph :



TOP20 Tags VS Duplicate Ratio for each Tag

7. Determine the time required to close duplicate questions
   Here we need to Calculate the time in between posting the question and marking that question as duplicate questions. I have calculated the time in years to make the graph interpretable.



This shows that most of the duplicate questions are marked as duplicate within a **span of year** and maximum time for a question to be marked as duplicate is **9 years**

```
[30] freq
```

```
Years_To_Duplicate
0    382923
1     22176
2     11884
3      6331
4      3628
5      2045
6      1234
7       623
8       255
9        54
dtype: int64
```

Questions closed as duplicate after the years from which they were published.

8. Distribution of the reputation of users whose questions are closed as duplicates
   (In process and maybe code will be uploaded by me in my git account at the time when faculty checks this file).

9. Consider the first 500 tags and determine how many percentage of questions are been covered
   Here we need to observe that out of total nearly 25 thousand tags top 500 tags covers how much portion of total asked questions.

```
[ ]  que_count = 0
     Total_count = 0
     for i in dataset2['tags']:
       Total_count += 1
       temp = i.split('>')
       que_tag_count = len(temp)-1
       checked_tags = 0
       for j in temp:
         if j[1:] in tags500:
           checked_tags += 1
       if checked_tags == que_tag_count:
             que_count += 1
     print(Total_count)
     print(que_count)
```

```
676204
352349
```

```
[ ]  print("Percentage of Questions covered in top 500 Tags are : ",(que_count/Total_count)*100)
```

```
Percentage of Questions covered in top 500 Tags are :  52.10690856605403
```

Top 500 tags cover **52%** of total questions asked in stack overflow

10. Repeat Q.9 for first 5000 tags
    Determine that out of total nearly 25 thousand tags top 5000 tags covers how much portion of total asked questions.

```
que_count5000 = 0
Total_count5000 = 0
for i in dataset2['tags']:
  Total_count5000 += 1
  temp = i.split('>')
  que_tag_count = len(temp)-1
  checked_tags = 0
  for j in temp:
    if j[1:] in tags5000:
      checked_tags += 1
  if checked_tags == que_tag_count:
        que_count5000 += 1
print(Total_count5000)
print(que_count5000)
print("Percentage of Questions covered in top 5000 Tags are : ",(que_count5000/Total_count5000)*100)
```

```
676204
598323
Percentage of Questions covered in top 5000 Tags are :  88.48261767159022
```

Top 5000 tags cover **88.5%** of total questions asked in stack overflow

GitHub Link : - https://github.com/AKSHAT2802/ML_LHC

Learning Outcomes:

By Long Hour Coding I came to know about data analysis part that is most important while dealing with dataset and before applying ML algorithms to that dataset.

Concepts like List , Dictionaries etc were brushed (Mainly basic Syntax to operate on lists and dictionaries was revised)

Also learned how to filter large datasets and how to read graphs that were plotted and derive conclusions from them.