

## CA675 Cloud Assignment

Name - Sarthak Bhagwat Taru

StudentID - 21261303

Email - sarthak.taru2@mail.dcu.ie

### Task 1 - Acquire the top 2,00,000 posts by ViewCount

Query executed on stackexchange data platform to retrieve the top 2,00,000 posts by viewcount

<https://data.stackexchange.com/stackoverflow/query/new>

(<https://data.stackexchange.com/stackoverflow/query/new>).

To retrieve top 2,00,000 records by viewcount, I have first find out the max viewcount from posts table and which was 10062790.

Query - select max(ViewCount) as viewcount from posts;

With that 4 queries, the sql is executed in following manner to extract top 2,00,000 records.

Query 1 - select top 50000 psts.Id,psts.Score,psts.ViewCount,psts.Body, users.Id as OwnerUserId, users.DisplayName,psts.Title,psts.Tags from Posts psts inner join Users users on psts.OwnerUserId=users.Id where psts.ViewCount>1000 and psts.ViewCount < 10062791 order by psts.ViewCount desc

Query 2 - select top 50000 psts.Id,psts.Score,psts.ViewCount,psts.Body, users.Id as OwnerUserId, users.DisplayName,psts.Title,psts.Tags from Posts psts inner join Users users on psts.OwnerUserId=users.Id where psts.ViewCount>1000 and psts.ViewCount < 124974 order by psts.ViewCount desc

Query 3 - select top 50000 psts.Id,psts.Score,psts.ViewCount,psts.Body, users.Id as OwnerUserId, users.DisplayName,psts.Title,psts.Tags from Posts psts inner join Users users on psts.OwnerUserId=users.Id where psts.ViewCount>1000 and psts.ViewCount < 73139 order by psts.ViewCount desc

Query 4 - select top 50000 psts.Id,psts.Score,psts.ViewCount,psts.Body, users.Id as OwnerUserId, users.DisplayName,psts.Title,psts.Tags from Posts psts inner join Users users on psts.OwnerUserId=users.Id where psts.ViewCount>1000 and psts.ViewCount < 52110 order by psts.ViewCount desc

In output, the 4 csv files were created and combined all that in one combined.csv

**Task 2 & 3 - Use Pig/Hive/MapReduce - Extract, Transform and Load the data as applicable to get:**

This assignment is completed on Google Cloud Platform by creating the cluster with Dataproc service. The google cloud was easy to understand and the cluster creation was easy through DataProc. Instance was configured with Jupyter notebook to execute the python code. I have used the Hive in cluster creation because of the prior hands-on experience in sql query language.

Created the ca675 Database and Table under same database.

Table creation query - CREATE TABLE IF NOT EXISTS ca675.Stackexchange (Id int,Score int,ViewCount int,Body string, OwnerUserId int, OwnerDisplayName string, Title string, Tags string) ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde';

**Data Cleaning**

Before loading data into the Hive, I have cleaned the body column on google colab jupyter notebook with following code snippet.

```
df['Body'] = df['Body'].str.replace(r'<[^<>]*>', '', regex=True)
df["Body"] = df['Body'].str.replace('[^\w\s]','', regex=True)
df["Body"] = df['Body'].str.replace('[^a-zA-Z]', ' ', regex=True)
df["Body"] = df['Body'].str.replace(r'\n', ' ', regex=True)
```

**Data Loading into Hive**

Uploaded the csv file in assigned google cloud storage and loaded the data through below sql query -  
LOAD DATA INPATH 'gs://dataproc-staging-us-central1-1078341724532- t5a0gorr/combined.csv' INTO  
TABLE ca675.Stackexchange;

After Successful data loading, the Jupyter is accessed by cloud public address.

Installed below packages in cloud ubuntu environment -

```
! pip install sasl
! pip install thrift
! pip install thrift-sasl
! pip install PyHive
```

In [20]:

```
from pyhive import hive
from tabulate import tabulate
import pandas as pd
```

In [21]:

```
host_name = "localhost"
port = 10000
user = "sarthak_taru2"
password = "509485636803468691"
database="ca675"
conn = hive.Connection(host=host_name,port=port,username=user,password=password,
database=database,auth='CUSTOM')
cur = conn.cursor()
```

## Task 2.1. The top 10 posts by score

In [30]:

```
cur.execute('''select Id,OwnerUserId,OwnerDisplayName,Score,ViewCount
FROM ca675.stackexchanges order by Score desc LIMIT 10''')
result = cur.fetchall()
df = pd.DataFrame (result, columns =['Id','OwnerUserId','OwnerDisplayName','Score','ViewCount'])
display(df)
```

	Id	OwnerUserId	OwnerDisplayName	Score	ViewCount
0	11227809	87234	GManNickG	25933	1649855
1	927358	89904	Hamza Yerlikaya	23348	10062790
2	2003505	95592	Matthew Rankin	18514	9285139
3	292357	6068	pupeno	12834	3041604
4	231767	18300	Alex. S.	11551	2681330
5	477816	12870	Oli	10921	3269028
6	348170	14069	paxos1977	10079	3985243
7	5767325	364969	Walker	9931	8937271
8	6591213	338204	Forrest	9792	3729583
9	1642028	87234	GManNickG	9560	877861

## Task 2.2 The top 10 users by post score

In [28]:

```

cur.execute('''
select OwnerUserId,max(OwnerDisplayName) , sum(Score) as score
from stackexchanges
group by OwnerUserId
order by Score desc limit 10
''')
result = cur.fetchall()
df1 = pd.DataFrame (result, columns = ['OwnerUserId','OwnerDisplayName','score'
])
display(df1)

```

	OwnerUserId	OwnerDisplayName	score
0	87234	GManNickG	37672
1	4883	readonly	28817
2	9951	e-satis	26878
3	6068	pupeno	25944
4	89904	Hamza Yerlikaya	24024
5	51816	Joan Venge	23763
6	49153	Ali	20203
7	179736	TIMEX	19603
8	95592	Matthew Rankin	19479
9	63051	flybywire	19362

### Task 3 - he number of distinct users, who used the word “cloud” in one of their posts

Here I have considered the Body **and** Title to find out the total count of owner user id who used the word cloud in one of their posts.

In [39]:

```

cur.execute('''
SELECT
COUNT(DISTINCT OwnerUserId) as owner_user_count
FROM stackexchanges
WHERE Title LIKE '% Cloud %' AND Body LIKE '% Cloud %'
''')
result = cur.fetchall()
df2 = pd.DataFrame (result, columns = ['Owner_User_Count'])
display(df2)

```

Owner_User_Count
0
168

**Task 4 - calculate the per-user TF-IDF of the top 10 terms for each of the top 10 users**

So first of all, I have extracted the top 10 owner user id with their titles and created the list of top 10 users.

In [35]:

```
df3 = pd.read_sql("""
SELECT OwnerUserId,OwnerDisplayName,Title,Body
from stackexchanges
WHERE OwnerUserId
IN
(
select OwnerUserId from(select OwnerUserId,max(OwnerDisplayName),sum(Score) as s
core
from stackexchanges
group by OwnerUserId
order by score desc limit 10)stack
)
order by OwnerUserId""", conn)
result = cur.fetchall()
top_10_users = list(df3["owneruserid"].unique())
display(top_10_users)
```

```
[4883, 6068, 9951, 49153, 51816, 63051, 87234, 89904, 95592, 179736]
```

I have taken the reference from below site to implement tfidf

[https://scikitlearn.org/stable/modules/generated/sklearn.feature\\_extraction.text.TfidfVectorizer.html](https://scikitlearn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html)

([https://scikitlearn.org/stable/modules/generated/sklearn.feature\\_extraction.text.TfidfVectorizer.html](https://scikitlearn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html))

for each of the owner and their top 10 words plot the table accordingly.

In [36]:

```
from sklearn.feature_extraction.text import TfidfVectorizer

def calculate_tfidf(title_each_user):
    TitleCopy = title_each_user["title"].copy()
    vectorizer = TfidfVectorizer(stop_words='english', lowercase=True)
    response = vectorizer.fit_transform(title_each_user['body'].str.cat(' '+TitleCopy))
    df_tfidf_words = pd.DataFrame(response.toarray(), columns=vectorizer.get_feature_names()) #calculating tf-idf values
    df_final_result_per_user=df_tfidf_words.sum(axis=0, numeric_only=True) #summing up the tf-idf to get the top 10 words
    top_words = df_final_result_per_user.nlargest(n=10)
    top_10_words = list(top_words.index)
    return(df_tfidf_words[top_10_words])

for each_item in top_10_users: # Iterating through top 10 users
    owneruserid = str(each_item)
    selectTitle=df3.loc[df3['owneruserid'] == each_item]
    selectTitle.insert(0, 'Owneruserid', each_item)
    tfidf=calculate_tfidf(selectTitle)
    tfidf.insert(0, 'owneruserid', owneruserid)
    display(tfidf)
```

	owner	userid	python	use	ruby	list	table	way	difference	br
0		4883	0.000000	0.000000	0.000000	0.000000	0.521156	0.000000	0.000000	0.00
1		4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
2		4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
3		4883	0.072578	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
4		4883	0.160147	0.000000	0.219470	0.438941	0.000000	0.184107	0.000000	0.00
5		4883	0.000000	0.000000	0.000000	0.202044	0.000000	0.169489	0.000000	0.00
6		4883	0.000000	0.198917	0.000000	0.000000	0.255265	0.099459	0.000000	0.00
7		4883	0.121700	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
8		4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.181739	0.000000	0.00
9		4883	0.219948	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
10		4883	0.149597	0.000000	0.000000	0.000000	0.000000	0.000000	0.220697	0.00
11		4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.123956	0.000000	0.34
12		4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
13		4883	0.000000	0.000000	0.413988	0.000000	0.000000	0.000000	0.000000	0.00
14		4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
15		4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
16		4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
17		4883	0.000000	0.120015	0.000000	0.000000	0.308023	0.120015	0.000000	0.00
18		4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
19		4883	0.087249	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
20		4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
21		4883	0.101808	0.234079	0.000000	0.418562	0.000000	0.000000	0.000000	0.00
22		4883	0.000000	0.000000	0.404077	0.000000	0.000000	0.000000	0.000000	0.00
23		4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
24		4883	0.000000	0.000000	0.000000	0.142349	0.000000	0.000000	0.000000	0.00
25		4883	0.195328	0.112276	0.000000	0.000000	0.000000	0.000000	0.288162	0.00
26		4883	0.251336	0.000000	0.000000	0.000000	0.000000	0.144470	0.000000	0.00
27		4883	0.000000	0.000000	0.209780	0.000000	0.000000	0.000000	0.000000	0.00
28		4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
29		4883	0.000000	0.191672	0.000000	0.000000	0.000000	0.000000	0.491935	0.00
30		4883	0.146760	0.168717	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
31		4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.63
32		4883	0.000000	0.231741	0.000000	0.000000	0.000000	0.000000	0.000000	0.00

	owner	userid	file	java	android	git	sql	way	like	t
0		6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00C
1		6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.19E
2		6068	0.000000	0.000000	0.000000	0.728914	0.000000	0.000000	0.000000	0.00C
3		6068	0.000000	0.000000	0.130150	0.000000	0.000000	0.000000	0.000000	0.00C
4		6068	0.000000	0.000000	0.173267	0.000000	0.000000	0.000000	0.000000	0.38E
5		6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.140540	0.000000	0.00C
6		6068	0.000000	0.386646	0.000000	0.000000	0.000000	0.000000	0.000000	0.00C
7		6068	0.000000	0.000000	0.275010	0.000000	0.000000	0.000000	0.000000	0.00C
8		6068	0.000000	0.000000	0.000000	0.000000	0.217416	0.000000	0.000000	0.00C
9		6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.047141	0.00C
10		6068	0.000000	0.000000	0.148172	0.000000	0.000000	0.000000	0.000000	0.00C
11		6068	0.000000	0.000000	0.334555	0.000000	0.000000	0.124201	0.000000	0.00C
12		6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.076296	0.158251	0.00C
13		6068	0.000000	0.000000	0.000000	0.000000	0.105437	0.000000	0.072167	0.00C
14		6068	0.656317	0.000000	0.000000	0.417075	0.000000	0.085457	0.000000	0.00C
15		6068	0.138956	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00C
16		6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00C
17		6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.080992	0.000000	0.00C
18		6068	0.116987	0.000000	0.000000	0.000000	0.692403	0.000000	0.094784	0.00C
19		6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00C
20		6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.037598	0.00C
21		6068	0.166608	0.000000	0.000000	0.000000	0.262958	0.130161	0.000000	0.00C
22		6068	0.371197	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00C
23		6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.089291	0.000000	0.13E
24		6068	0.307285	0.000000	0.000000	0.000000	0.000000	0.120032	0.124483	0.00C
25		6068	0.000000	0.000000	0.352152	0.000000	0.000000	0.000000	0.000000	0.00C
26		6068	0.083346	0.370875	0.000000	0.000000	0.000000	0.000000	0.067528	0.00C
27		6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.029332	0.060840	0.00C
28		6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.093310	0.290312	0.00C
29		6068	0.000000	0.364640	0.000000	0.000000	0.000000	0.000000	0.000000	0.00C
30		6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00C
31		6068	0.000000	0.000000	0.000000	0.142581	0.000000	0.087643	0.090893	0.00C
32		6068	0.000000	0.154458	0.000000	0.000000	0.000000	0.108471	0.000000	0.32E
33		6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.083163	0.000000	0.00C
34		6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00C
35		6068	0.000000	0.390426	0.000000	0.000000	0.000000	0.000000	0.071088	0.00C
36		6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.106695	0.00C



	owneruserid	python	like	git	head	does	using	way	
0	9951	0.000000	0.000000	0.000000	0.000000	0.000000	0.299606	0.000000	0.000
1	9951	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
2	9951	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
3	9951	0.000000	0.000000	0.179128	0.734307	0.000000	0.000000	0.000000	0.000
4	9951	0.000000	0.000000	0.187925	0.000000	0.062642	0.000000	0.059024	0.000
5	9951	0.109076	0.092250	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
6	9951	0.000000	0.033760	0.000000	0.000000	0.000000	0.159673	0.112839	0.119
7	9951	0.000000	0.000000	0.338695	0.000000	0.000000	0.000000	0.000000	0.000
8	9951	0.081895	0.138523	0.000000	0.000000	0.163789	0.000000	0.000000	0.000
9	9951	0.000000	0.000000	0.416684	0.379585	0.138895	0.000000	0.000000	0.139
10	9951	0.155490	0.328761	0.000000	0.000000	0.000000	0.000000	0.146510	0.000
11	9951	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.083623	0.000
12	9951	0.441995	0.000000	0.000000	0.000000	0.000000	0.000000	0.277646	0.000
13	9951	0.000000	0.066630	0.078782	0.000000	0.000000	0.000000	0.000000	0.000
14	9951	0.000000	0.000000	0.000000	0.000000	0.384412	0.000000	0.144885	0.000
15	9951	0.000000	0.000000	0.000000	0.000000	0.227740	0.000000	0.000000	0.113
16	9951	0.000000	0.000000	0.000000	0.000000	0.000000	0.236845	0.000000	0.000
17	9951	0.000000	0.169226	0.000000	0.000000	0.000000	0.000000	0.000000	0.200
18	9951	0.525068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.262
19	9951	0.000000	0.000000	0.075446	0.000000	0.000000	0.150893	0.000000	0.000
20	9951	0.000000	0.156069	0.000000	0.000000	0.000000	0.000000	0.173878	0.000
21	9951	0.000000	0.000000	0.000000	0.000000	0.000000	0.119729	0.000000	0.119
22	9951	0.000000	0.101237	0.000000	0.000000	0.119702	0.000000	0.000000	0.000
23	9951	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
24	9951	0.079724	0.202277	0.000000	0.000000	0.000000	0.079724	0.000000	0.000

	owneruserid	javascript	array	using	php	jquery	file	want	
0	49153	0.000000	0.000000	0.076582	0.000000	0.0	0.000000	0.095828	0.0000
1	49153	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.0000
2	49153	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.0000
3	49153	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.084953	0.0000
4	49153	0.000000	0.524079	0.053427	0.032054	0.0	0.000000	0.033427	0.4528
...	...	...	...	...	...	...	...	...	...
74	49153	0.000000	0.000000	0.000000	0.000000	0.0	0.170775	0.000000	0.0000
75	49153	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.0000
76	49153	0.085136	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.0000
77	49153	0.000000	0.000000	0.067637	0.000000	0.0	0.000000	0.169268	0.0000
78	49153	0.101080	0.000000	0.084238	0.000000	0.0	0.000000	0.000000	0.0000

79 rows × 11 columns

	owneruserid	python	like	list	string	want	class	index	value
0	51816	0.000000	0.075548	0.000000	0.000000	0.000000	0.000000	0.0	0.126344
1	51816	0.000000	0.082816	0.000000	0.000000	0.000000	0.000000	0.0	0.000000
2	51816	0.000000	0.000000	0.143562	0.034883	0.000000	0.000000	0.0	0.000000
3	51816	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.000000
4	51816	0.000000	0.000000	0.000000	0.000000	0.000000	0.033316	0.0	0.000000
...	...	...	...	...	...	...	...	...	...
61	51816	0.000000	0.000000	0.000000	0.000000	0.000000	0.350645	0.0	0.000000
62	51816	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.000000
63	51816	0.393731	0.193425	0.000000	0.295685	0.266721	0.000000	0.0	0.000000
64	51816	0.231039	0.075667	0.000000	0.000000	0.000000	0.000000	0.0	0.000000
65	51816	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.000000

66 rows × 11 columns

	owner	userid	file	python	want	vs	use	standard	command	
0		63051	0.000000	0.328540	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
1		63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
2		63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
3		63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
4		63051	0.000000	0.000000	0.068103	0.000000	0.000000	0.000000	0.000000	0.00
5		63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
6		63051	0.192038	0.000000	0.089639	0.000000	0.000000	0.000000	0.000000	0.00
7		63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
8		63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
9		63051	0.000000	0.146353	0.000000	0.153253	0.306505	0.000000	0.000000	0.00
10		63051	0.077215	0.000000	0.288337	0.000000	0.000000	0.000000	0.000000	0.00
11		63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
12		63051	0.000000	0.151551	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
13		63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
14		63051	0.000000	0.000000	0.000000	0.386070	0.193035	0.000000	0.000000	0.00
15		63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
16		63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
17		63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
18		63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
19		63051	0.000000	0.000000	0.086165	0.000000	0.054446	0.000000	0.303110	0.17
20		63051	0.000000	0.000000	0.000000	0.227035	0.000000	0.000000	0.000000	0.00
21		63051	0.071235	0.000000	0.066501	0.000000	0.000000	0.000000	0.374298	0.17
22		63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
23		63051	0.000000	0.000000	0.172612	0.000000	0.000000	0.000000	0.000000	0.00
24		63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
25		63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
26		63051	0.318679	0.000000	0.099168	0.000000	0.000000	0.000000	0.000000	0.26
27		63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
28		63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
29		63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
30		63051	0.166264	0.000000	0.155217	0.000000	0.000000	0.466327	0.000000	0.41
31		63051	0.243946	0.000000	0.000000	0.000000	0.172682	0.547364	0.000000	0.18
32		63051	0.000000	0.113312	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
33		63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
34		63051	0.000000	0.000000	0.112601	0.000000	0.000000	0.000000	0.000000	0.00
35		63051	0.102145	0.115083	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
36		63051	0.000000	0.537229	0.000000	0.000000	0.000000	0.000000	0.000000	0.00

	owneruserid	file	python	want	vs	use	standard	command	
37	63051	0.000000	0.000000	0.122889	0.000000	0.000000	0.000000	0.000000	0.00
38	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
39	63051	0.124521	0.000000	0.116247	0.000000	0.000000	0.000000	0.163572	0.00
40	63051	0.319440	0.000000	0.149107	0.000000	0.188434	0.000000	0.000000	0.00
41	63051	0.000000	0.321788	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
42	63051	0.000000	0.000000	0.000000	0.232327	0.000000	0.000000	0.000000	0.00
43	63051	0.000000	0.000000	0.000000	0.253717	0.000000	0.000000	0.000000	0.00
44	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
45	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
46	63051	0.000000	0.000000	0.000000	0.194207	0.388413	0.000000	0.000000	0.00
47	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.207214	0.000000	0.00
48	63051	0.278703	0.000000	0.000000	0.000000	0.000000	0.000000	0.366107	0.00

	owneruserid	idiom	int	copy	array	arraysize	data	code	doc
0	87234	0.480997	0.000000	0.360748	0.091453	0.000000	0.000000	0.000000	0.24049
1	87234	0.000000	0.203351	0.000000	0.261452	0.30558	0.30558	0.145251	0.00000
2	87234	0.000000	0.267377	0.000000	0.000000	0.000000	0.000000	0.133688	0.00000

	owneruserid	file	timer	new	java	setpreferredwidth	tablegetcolumnmodel
0	89904	0.000000	0.0000	0.000000	0.040077	0.606105	
1	89904	0.351468	0.0000	0.000000	0.000000	0.000000	
2	89904	0.000000	0.0000	0.000000	0.000000	0.000000	
3	89904	0.463315	0.0000	0.000000	0.000000	0.000000	
4	89904	0.000000	0.0000	0.000000	0.000000	0.000000	
5	89904	0.000000	0.0000	0.000000	0.313903	0.000000	
6	89904	0.000000	0.6912	0.235033	0.091408	0.000000	
7	89904	0.075479	0.0000	0.431366	0.000000	0.000000	
8	89904	0.000000	0.0000	0.000000	0.163094	0.000000	
9	89904	0.000000	0.0000	0.000000	0.000000	0.000000	

	owneruserid	install	python	pip	branch	installed	flask	version	start
0	95592	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0000
1	95592	0.000000	0.373806	0.000000	0.000000	0.000000	0.000000	0.000000	0.5168
2	95592	0.427251	0.000000	0.284834	0.000000	0.000000	0.000000	0.000000	0.0000
3	95592	0.000000	0.000000	0.000000	0.000000	0.436263	0.520551	0.520551	0.0000
4	95592	0.262228	0.104891	0.104891	0.000000	0.121554	0.000000	0.000000	0.0000
5	95592	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0000
6	95592	0.466707	0.262523	0.320861	0.000000	0.000000	0.000000	0.000000	0.0000
7	95592	0.000000	0.000000	0.000000	0.697882	0.000000	0.000000	0.000000	0.0000

	owneruserid	python	want	string	user	dictionary	file	return	
0	179736	0.170174	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.2
1	179736	0.122064	0.000000	0.163895	0.000000	0.172007	0.000000	0.000000	0.0
2	179736	0.000000	0.097942	0.000000	0.302904	0.000000	0.000000	0.000000	0.0
3	179736	0.000000	0.000000	0.000000	0.000000	0.000000	0.272653	0.000000	0.0
4	179736	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
...	...	...	...	...	...	...	...	...	...
110	179736	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
111	179736	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.1
112	179736	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
113	179736	0.000000	0.140351	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
114	179736	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.373771	0.0

115 rows × 11 columns