

## CA675: CLOUD TECHNOLOGIES

### ASSIGNMENT 1: DATA ANALYSIS ON GOOGLE CLOUD AND LOCAL HADOOP DEPLOYMENT

# ASSIGNMENT IMPLEMENTED BOTH ON **GOOGLE CLOUD PLATFORM** AND LOCAL MACHINE

## Contents

- 1) Screenshots
- 2) Code
- 3) Dashboards

# SCREENSHOTS

# GCP SCREENSHOTS

## 1) Hive shell on GCP shell

```
cypherhok@cluster-6504-m: ~ -- Google Chrome
ssh.cloud.google.com/projects/ca675-329118/zones/us-central1-f/instances/cluster-6504-m?authuser=0&hl=en_US&projectNumber=946519243301&useAdminProxy=true&troubleshoot4005Enabled=true&troubleshoot255Enabled=true

at org.apache.hadoop.hive.ql.parse.HiveParser.createTableStatement(HiveParser.java:2942)
at org.apache.hadoop.hive.ql.parse.HiveParser.createTableStatement(HiveParser.java:6175)
at org.apache.hadoop.hive.ql.parse.HiveParserddlStatement(HiveParser.java:3808)
at org.apache.hadoop.hive.ql.parse.HiveParser.createStatement(HiveParser.java:12302)
at org.apache.hadoop.hive.ql.parse.HiveParser.statement(HiveParser.java:1233)
at org.apache.hadoop.hive.ql.parse.ParserDriver.parse(ParserDriver.java:208)
at org.apache.hadoop.hive.ql.parse.ParserDriver.parse(ParserDriver.java:77)
at org.apache.hadoop.hive.ql.parse.ParserDriver.parse(ParserDriver.java:70)
at org.apache.hadoop.hive.ql.Driver.compile(Driver.java:440)
at org.apache.hadoop.hive.ql.Driver.compileInternal(Driver.java:1317)
at org.apache.hadoop.hive.ql.Driver.runInternal(Driver.java:1457)
at org.apache.hadoop.hive.ql.Driver.run(Driver.java:1227)
at org.apache.hadoop.hive.ql.Driver.run(Driver.java:1227)
at org.apache.hadoop.hive.cli.CliDriver.processLocalCmd(CliDriver.java:233)
at org.apache.hadoop.hive.cli.CliDriver.processCmd(CliDriver.java:184)
at org.apache.hadoop.hive.cli.CliDriver.processLine(CliDriver.java:403)
at org.apache.hadoop.hive.cli.CliDriver.executeDriver(CliDriver.java:521)
at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:759)
at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:666)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethod)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethod)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethod)
at java.lang.reflect.Method.invoke(Method.java:498)
at org.apache.hadoop.util.RunJar.run(RunJar.java:244)
at org.apache.hadoop.util.RunJar.main(RunJar.java:158)
FAILED: ParseException line 1:30 cannot recognise input near: ',' 'id' ',' in column type
hive> create table usernames(row_num string, id string, displayname string) ROW FORMAT SERDE 'org.apache.hadoop.hi
ve.serde.OpenCSVSerde';
OK
Time taken: 0.117 seconds
hive> LOAD DATA INPATH 'gs://dataproc-staging-us-central1-946519243301-wm6enp7/data/usernames/username_output_fina
l.csv' INTO TABLE stackexchange;
Loading data to table default.stackexchange
OK
Time taken: 1.387 seconds
hive> select * from usernames limit 10;
WARNING: Hive-on-MR is deprecated as of Hive 2 and may not be available in the future versions. Consider using a diffe
rent execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = cypherhok_20211023141042_9fc49f5e-2f98-4bba-b8e5-11b31a7ccf61
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1634992993441_0011, Tracking URL = http://cluster-6504-m:8080/proxy/application_1634992993441_00
11/
Hadoop job information for Stage=1: number of mappers: 0; number of reducers: 0
2021-10-23 14:10:52,862 Stages map = 0x, reduce = 0x
Ended Job = job_1634992993441_0011
MapReduce Jobs Launched:
Stage=Stage1: MAPS=0x REDUCES=0x SUCCESS=0x
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 14.26 seconds
hive> LOAD DATA INPATH 'gs://dataproc-staging-us-central1-946519243301-wm6enp7/data/usernames/username_output_fina
l.csv' INTO TABLE stackexchange;
FAILED: SemanticException line 1:17 Invalid path 'gs://dataproc-staging-us-central1-946519243301-wm6enp7/data/use
rnames/username_output_final.csv': No files matching path gs://dataproc-staging-us-central1-946519243301-wm6enp7/
data/usernames/username_output_final.csv
hive> LOAD DATA INPATH 'gs://dataproc-staging-us-central1-946519243301-wm6enp7/data/usernames/username_output_fina
l.csv' INTO TABLE stackexchange;
Loading data to table default.usernames
OK
Time taken: 1.322 seconds
hive> SET hive.execution.engine=tez;
hive>
```

## 2) VM instances of cluster

The screenshot shows the Google Cloud Platform console interface. The left sidebar contains navigation options like 'Virtual machines', 'Instance templates', 'Machine images', etc. The main content area is titled 'VM instances' and shows a list of instances. The table has columns for Status, Name, Zone, Recommendations, In use by, Internal IP, and Connect. Three instances are listed: cluster-6504-m, cluster-6504-e0, and cluster-6504-e1, all in the us-central1-f zone. The 'Related actions' section on the right provides links for viewing billing reports, monitoring VMs, exploring VM logs, setting firewall rules, and patch management.

### 3) Dataproc cluster

The screenshot shows the Google Cloud Platform interface for the 'Dataproc' service. The 'Clusters' tab is active, displaying a table with one cluster. The cluster is named 'cluster-6504', is in the 'Running' state, located in the 'us-central1' region, 'us-central1-f' zone, and has 2 total worker nodes. It is scheduled for deletion 'Off' and uses the 'dataproc-staging-us-central1-946519243301-vum6enp7' Cloud Storage staging bucket. The cluster was created on Oct 23, 2021, at 1:41:36 PM. The right sidebar shows 'No clusters selected' with a message to 'Please select at least one resource.' The bottom status bar shows the time as 3:33 PM on 10/23/2021.

Name	Status	Region	Zone	Total worker nodes	Scheduled deletion	Cloud Storage staging bucket	Created
cluster-6504	Running	us-central1	us-central1-f	2	Off	dataproc-staging-us-central1-946519243301-vum6enp7	Oct 23, 2021, 1:41:36 PM

### 4) Storage buckets for data

The screenshot shows the Google Cloud Platform interface for the 'Cloud Storage' service. The 'Bucket details' page for the bucket 'dataproc-staging-us-central1-946519243301-vum6enp7' is displayed. The bucket is located in 'us-central1 (Iowa)', uses 'Standard' storage class, and has 'Subject to object ACLs' for public access. The 'OBJECTS' tab is active, showing a list of objects. The objects are 'output/' and 'usernames/'. The bottom status bar shows the time as 3:33 PM on 10/23/2021.

Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	Encryption	Retention expiration
output/	-	Folder	-	-	-	-	-	-	-
usernames/	-	Folder	-	-	-	-	-	-	-

## 5) Code execution through Jupyter

The screenshot displays a Jupyter Notebook titled "Assignment-CA675" in a web browser. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running code, and saving. The notebook content is as follows:

```
host_name = "localhost"
port = 10000
user = "cypherhawk"
password = "7972163666346360302"
database="default"

def get_hive_connection(host_name, port, user,password, database):
    conn = Hive.Connection(host=host_name, port=port, username=user, password=password,
                           database=database, auth='CUSTOM')
    return conn

conn = get_hive_connection(host_name, port, user,password, database)
cur = conn.cursor()
```

Below the code editor, there is a section titled "Set execution engine to tez" with a button "Add tag".

In [37]:

```
cur.execute('SET hive.execution.engine=tez')
```

Below this, there is a section titled "Task 2 & 3" with a sub-section "2.2.1) Querying top 10 posts by score".

In [35]:

```
cur.execute('select ID, Title, Score from stackexchange_final order by score desc limit 10')
result = cur.fetchall()
print(tabulate(result, tablefmt='orgtbl'))
```

The output of the query is displayed as a table:

ID	Title	Score
11227809	Why is processing a sorted array faster than processing an unsorted array?	25893
077208	How do I undo the most recent local commit in Git?	12174

## LOCAL DEPLOYMENT

### 1) Hadoop Services

```
hadoop@Shubham:~$ jps
2336 Jps
1074 NodeManager
706 SecondaryNameNode
914 ResourceManager
1459 RunJar
283 NameNode
479 DataNode
hadoop@Shubham:~$
```

### 2) Hiveserver2

```
hadoop@Shubham:~/apache-hive-3.1.2-bin$ hiveserver2
2021-10-24 09:44:09: Starting HiveServer2
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop-3.3.1/share/hadoop/common/lib/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = e0aa9409-6a4d-42e2-b22b-e508a993d1c3
Hive Session ID = 58d804ea-6401-4915-8f7d-18cb8dca1861
Hive Session ID = 6970c933-7fb5-4261-a2ba-5b80ceada7c0
Hive Session ID = 4b9599e0-099e-4752-b8c5-f09a00de016b
OK
^C
```

### 3) Jupyter Notebook

...

Add tag

## TF/IDF on data stored on Hive Assignment

CA675

Author : Shubham Rai Student Number : 21261161

...

Add tag

## Data Extraction

Data set extracted from following link <https://data.stackexchange.com/stackoverflow/query/new>

...

Add tag

## Query used to extract data

As data extraction limit is 50000 we need to extract 4 files with help of row number.

```
SELECT
*
FROM
(
SELECT
ROW_NUMBER() OVER (ORDER BY VIEWCOUNT DESC) AS RN,
Id,
PostTypeId,
AcceptedAnswerId,
ParentId,
CreationDate.
```

# Assignment-CA675

October 23, 2021

## 1 Hive Assignment

CA675

Author : Shubham Rai Student Number : 21261161

## 2 Data Extraction

Data set extracted from following link <https://data.stackexchange.com/stackoverflow/query/new>

### 2.0.1 Query used to extract data

As data extraction limit is 50000 we need to extract 4 files with help of row number.

```
SELECT
*
FROM
(
SELECT
ROW_NUMBER() OVER (ORDER BY VIEWCOUNT DESC) AS RN,
Id,
PostTypeId,
AcceptedAnswerId,
ParentId,
CreationDate,
DeletionDate,
Score,
ViewCount,
OwnerUserId,
OwnerDisplayName,
LastEditorUserId,
LastEditorDisplayName,
LastEditDate,
LastActivityDate,
Title,
Tags,
AnswerCount,
CommentCount,
FavoriteCount,
```



```

ClosedDate,
CommunityOwnedDate,
ContentLicense
FROM POSTS WHERE VIEWCOUNT IS NOT NULL
) AS T
WHERE RN BETWEEN 150000 AND 200000

```

### 3 Starting Hive Server

```
bin/hiveserver2 bin/beeline -n hdoop -u jdbc:hive2://localhost:10000
```

### 4 Create Table in HIVE

As delimited file formatted has issues while reading quoted characters we are using SERDE format to read CSV and insert data in HIVE

```

[ ]: create table stackexchange(RN int, Id int, PostTypeId int, AcceptedAnswerId_
    ↪int, ParentId int, CreationDate timestamp, DeletionDate timestamp, Score_
    ↪BIGINT, ViewCount BIGINT, OwnerUserId varchar(255), OwnerDisplayName_
    ↪varchar(255), LastEditorUserId varchar(255), LastEditorDisplayName_
    ↪varchar(255), LastEditDate timestamp, LastActivityDate timestamp, Title_
    ↪timestamp, Tags varchar(255), AnswerCount int, CommentCount int, FavoriteCount_
    ↪int, ClosedDate timestamp, CommunityOwnedDate timestamp, ContentLicense_
    ↪varchar(255)) ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde';

```

### 5 Put Data on HDFS

```
hadoop fs -put < local_path to < hdfs_location >
```

### 6 Task 2 & 3

#### 6.0.1 2.2.1) Querying top 10 posts by score

```

[4]: from pyhive import hive
    from tabulate import tabulate
    import pandas as pd

    host_name = "localhost"
    port = 10000
    user = "hdoop"
    password = "a"
    database="default"

    def hiveconnection(host_name, port, user,password, database):

```

```

    conn = hive.Connection(host=host_name, port=port, username=user,
↳password=password,
                                database=database, auth='CUSTOM')

    return conn

conn = hiveconnection(host_name, port, user,password, database)
cur = conn.cursor()

## Usage example from https://github.com/dropbox/PyHive

```

```

[80]: cur.execute('select ID, Title, Score from stackexchange_view order by score_
↳desc limit 10')
result = cur.fetchall()
print(tabulate(result, tablefmt='orgtbl'))

```

```

| 11227809 | Why is processing a sorted array faster than processing an unsorted
array? | 25893 |
| 927358 | How do I undo the most recent local commits in Git?
| 23274 |
| 2003505 | How do I delete a Git branch locally and remotely?
| 18451 |
| 292357 | What is the difference between 'git pull' and 'git fetch'?
| 12796 |
| 231767 | What does the "yield" keyword do?
| 11512 |
| 477816 | What is the correct JSON content type?
| 10894 |
| 348170 | How do I undo 'git add' before commit?
| 10045 |
| 5767325 | How can I remove a specific item from an array?
| 9877 |
| 6591213 | How do I rename a local Git branch?
| 9747 |
| 1642028 | What is the "-->" operator in C/C++?
| 9539 |

```

## 6.0.2 2.2.2) The top 10 users by post score

```

[8]: ### Join data as usernames data was extracted and added post data pulling

cur.execute("""
SELECT DISTINCT u.displayname,stackexchange.score,stackexchange.owneruserid FROM
(
    select
        OwnerUserId,
        sum(score) as score

```

```

        from stackexchange_view
        group by OwnerUserId
        order by score desc
        LIMIT 11
    )stackexchange
LEFT JOIN usernames u
ON stackexchange.owneruserid = u.id
ORDER BY stackexchange.score DESC
""")
result = cur.fetchall()
print(tabulate(result, tablefmt='orgtbl'))

```

		448906	
	GManNickG	37606   87234	
	readonly	28739   4883	
	e-satis	26728   9951	
	pupeno	25860   6068	
	Hamza Yerlikaya	23949   89904	
	Joan Venge	23632   51816	
	Ali	20156   49153	
	TIMEX	19454   179736	
	Matthew Rankin	19413   95592	
	flybywire	19295   63051	

**6.0.3 2.2.3) The number of distinct users, who used the word “cloud” in one of their posts**

**6.1 – Note :** Using Title and Tags field for counting cloud as we are receiving timeout when trying to pull body field from source

```

[15]: cur.execute("""
SELECT
    COUNT(DISTINCT owneruserid) as user_count
FROM stackexchange_view
WHERE title LIKE '%cloud%' or tags LIKE '%cloud%'
""")
result = cur.fetchall()
print(tabulate(result, tablefmt='orgtbl'))

```

	357
--	-----

```

[77]: ### Get data for top 10 users from above list and pull data

df = pd.read_sql("""
SELECT
    owneruserid,
    title
from stackexchange_view

```

```

WHERE owneruserid
IN
    (
        87234,
        4883,
        9951,
        6068,
        89904,
        51816,
        49153,
        179736,
        95592,
        63051
    )
order by owneruserid""", conn)
top_10_user_id = list(df["owneruserid"].unique())

```

```

[62]: ## References taken and custom function created by me.
## Reference : https://medium.com/@cmukesh8688/
→tf-idf-vectorizer-scikit-learn-dbc0244a911a

from sklearn.feature_extraction.text import TfidfVectorizer

# Calculate sum() of TF-IDF and get top 10 words with highest TF-IDF and select
→only those columns
def calculate_tf_idf(df):
    vectorizer = TfidfVectorizer(stop_words='english', lowercase=True) # Remove
    →Stop Words
    response = vectorizer.fit_transform(df["title"]) # Use title field for TF/
    →IDF
    df_tfidf_sklearn = pd.DataFrame(response.toarray(), columns=vectorizer.
    →get_feature_names_out()
    total_tf_idf = df_tfidf_sklearn.sum(axis = 0) # Remove sum of TF/IDF for
    →getting top 10 most used words
    top_10_list = total_tf_idf.nlargest(10) # Get top 10 words per user
    top_10_words = list(top_10_list.index) # Get list of top 10 words
    df_tfidf_sklearn[top_10_words] # Select only top 10 words as column
    return df_tfidf_sklearn[top_10_words]

```

## 6.2 For each of top 10 users and their top 10 words plot TF/IDF table for their corresponding comments

```
[81]: ## For all 10 users create TF-IDF table for each user for their respective
      ↪ comments.

      for each_user in top_10_user_id: ## Iterate over all top 10 users
          username_id = str(each_user)
          filtered_data = df[(df['owneruserid']==username_id)] # Filter data only for
          ↪ selected user
          tf_idf_df = calculate_tf_idf(filtered_data) # pass on to above function
          print("For Username ID TF/IDF table : "+username_id)
          tf_idf_df.insert(0, 'usernameid', username_id)# attach username ID field to
          ↪ dataframe
          display(tf_idf_df)

      ↪
      ↪ print("*****\n\n\n")
```

For Username ID TF/IDF table : 179736

	usernameid	python	django	javascript	js	node	use	\
0	179736	0.338128	0.000000	0.0	0.000000	0.000000	0.0	
1	179736	0.000000	0.000000	0.0	0.275686	0.275686	0.0	
2	179736	0.000000	0.000000	0.0	0.350733	0.350733	0.0	
3	179736	0.000000	0.000000	0.0	0.000000	0.000000	0.0	
4	179736	0.000000	0.000000	0.0	0.000000	0.000000	0.0	
..	...	...	...	...	...	...	...	
109	179736	0.000000	0.000000	0.0	0.000000	0.000000	0.0	
110	179736	0.000000	0.000000	0.0	0.000000	0.000000	0.0	
111	179736	0.000000	0.288203	0.0	0.000000	0.000000	0.0	
112	179736	0.000000	0.000000	0.0	0.000000	0.000000	0.0	
113	179736	0.288398	0.000000	0.0	0.000000	0.000000	0.0	

	string	dictionary	query	mysql
0	0.000000	0.0	0.0	0.000000
1	0.000000	0.0	0.0	0.000000
2	0.000000	0.0	0.0	0.000000
3	0.000000	0.0	0.0	0.000000
4	0.394849	0.0	0.0	0.000000
..	...	...	...	...
109	0.000000	0.0	0.0	0.000000
110	0.000000	0.0	0.0	0.000000
111	0.000000	0.0	0.0	0.000000
112	0.000000	0.0	0.0	0.398258
113	0.000000	0.0	0.0	0.000000

[114 rows x 11 columns]

\*\*\*\*\*

For Username ID TF/IDF table : 4883

	usernameid	python	git	ruby	table	process	list \
0	4883	0.000000	0.000000	0.000000	0.332494	0.000000	0.000000
1	4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.416967
3	4883	0.000000	0.000000	0.427544	0.000000	0.000000	0.000000
4	4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
5	4883	0.000000	0.000000	0.000000	0.000000	0.350231	0.000000
6	4883	0.000000	0.000000	0.521156	0.000000	0.000000	0.568902
7	4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
8	4883	0.000000	0.402233	0.000000	0.000000	0.000000	0.000000
9	4883	0.000000	0.000000	0.000000	0.521156	0.000000	0.000000
10	4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
11	4883	0.000000	0.280138	0.000000	0.000000	0.000000	0.000000
12	4883	0.249965	0.000000	0.000000	0.000000	0.000000	0.000000
13	4883	0.000000	0.000000	0.000000	0.330545	0.330545	0.000000
14	4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
15	4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
16	4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
17	4883	0.000000	0.301298	0.000000	0.000000	0.000000	0.000000
18	4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
19	4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
20	4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
21	4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
22	4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
23	4883	0.288276	0.000000	0.000000	0.000000	0.000000	0.000000
24	4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25	4883	0.288276	0.000000	0.000000	0.000000	0.000000	0.000000
26	4883	0.000000	0.000000	0.439712	0.000000	0.000000	0.000000
27	4883	0.362262	0.000000	0.000000	0.000000	0.000000	0.000000
28	4883	0.429802	0.000000	0.000000	0.000000	0.000000	0.000000
29	4883	0.234332	0.000000	0.000000	0.000000	0.000000	0.000000
30	4883	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
31	4883	0.000000	0.473839	0.000000	0.000000	0.000000	0.000000
32	4883	0.000000	0.000000	0.000000	0.000000	0.350231	0.000000

	rename	rails	difference	write
0	0.362955	0.362955	0.000000	0.362955
1	0.000000	0.000000	0.000000	0.000000
2	0.000000	0.000000	0.000000	0.000000

3	0.000000	0.000000	0.000000	0.000000
4	0.000000	0.000000	0.000000	0.000000
5	0.000000	0.000000	0.000000	0.000000
6	0.000000	0.000000	0.000000	0.000000
7	0.000000	0.000000	0.000000	0.000000
8	0.000000	0.000000	0.000000	0.000000
9	0.568902	0.000000	0.000000	0.000000
10	0.000000	0.000000	0.000000	0.000000
11	0.000000	0.000000	0.000000	0.000000
12	0.000000	0.000000	0.000000	0.000000
13	0.000000	0.000000	0.000000	0.000000
14	0.000000	0.000000	0.000000	0.000000
15	0.000000	0.000000	0.000000	0.000000
16	0.000000	0.000000	0.000000	0.000000
17	0.000000	0.000000	0.000000	0.000000
18	0.000000	0.000000	0.534435	0.000000
19	0.000000	0.000000	0.000000	0.000000
20	0.000000	0.534435	0.000000	0.000000
21	0.000000	0.000000	0.000000	0.000000
22	0.000000	0.000000	0.000000	0.000000
23	0.000000	0.000000	0.000000	0.000000
24	0.000000	0.000000	0.000000	0.000000
25	0.000000	0.000000	0.000000	0.000000
26	0.000000	0.000000	0.000000	0.479997
27	0.000000	0.000000	0.000000	0.000000
28	0.000000	0.000000	0.000000	0.000000
29	0.000000	0.000000	0.311276	0.000000
30	0.000000	0.000000	0.000000	0.000000
31	0.000000	0.000000	0.000000	0.000000
32	0.000000	0.000000	0.000000	0.000000

\*\*\*\*\*

For Username ID TF/IDF table : 49153

	usernameid	javascript	php	using	jquery	java	array	string	\
0	49153	0.372458	0.000000	0.000000	0.000000	0.0	0.0	0.0	
1	49153	0.000000	0.000000	0.000000	0.290805	0.0	0.0	0.0	
2	49153	0.000000	0.307253	0.000000	0.000000	0.0	0.0	0.0	
3	49153	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.0	
4	49153	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.0	
..	...	...	...	...	...	...	...	...	
74	49153	0.000000	0.337705	0.337705	0.000000	0.0	0.0	0.0	
75	49153	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.0	
76	49153	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.0	
77	49153	0.389987	0.000000	0.000000	0.000000	0.0	0.0	0.0	

78	49153	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.0
----	-------	----------	----------	----------	----------	-----	-----	-----

	file	class	select
0	0.0	0.000000	0.0
1	0.0	0.000000	0.0
2	0.0	0.000000	0.0
3	0.0	0.000000	0.0
4	0.0	0.000000	0.0
..	...	...	...
74	0.0	0.000000	0.0
75	0.0	0.000000	0.0
76	0.0	0.000000	0.0
77	0.0	0.577234	0.0
78	0.0	0.000000	0.0

[79 rows x 11 columns]

\*\*\*\*\*

For Username ID TF/IDF table : 51816

	usernameid	python	string	list	wpf	values	index \
0	51816	0.000000	0.000000	0.000000	0.000000	0.0	0.000000
1	51816	0.000000	0.000000	0.000000	0.328272	0.0	0.000000
2	51816	0.211868	0.000000	0.000000	0.000000	0.0	0.000000
3	51816	0.000000	0.000000	0.362504	0.000000	0.0	0.000000
4	51816	0.000000	0.000000	0.356142	0.000000	0.0	0.000000
..	...	...	...	...	...	...	...
59	51816	0.168840	0.000000	0.000000	0.000000	0.0	0.000000
60	51816	0.000000	0.604756	0.000000	0.000000	0.0	0.000000
61	51816	0.000000	0.000000	0.000000	0.000000	0.0	0.490276
62	51816	0.000000	0.000000	0.378230	0.000000	0.0	0.469521
63	51816	0.314929	0.539368	0.000000	0.000000	0.0	0.000000

	function	class	net	value
0	0.0	0.767028	0.000000	0.000000
1	0.0	0.000000	0.000000	0.000000
2	0.0	0.000000	0.000000	0.417964
3	0.0	0.000000	0.478167	0.000000
4	0.0	0.000000	0.000000	0.000000
..	...	...	...	...
59	0.0	0.000000	0.000000	0.000000
60	0.0	0.000000	0.000000	0.000000
61	0.0	0.000000	0.000000	0.000000
62	0.0	0.000000	0.000000	0.000000
63	0.0	0.000000	0.000000	0.000000



[64 rows x 11 columns]

\*\*\*\*\*

For Username ID TF/IDF table : 6068

	usernameid	file	sql	android	java	asp	mvc	\
0	6068	0.000000	0.000000	0.379839	0.000000	0.000000	0.000000	
1	6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
2	6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
3	6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
4	6068	0.404515	0.000000	0.000000	0.000000	0.000000	0.000000	
5	6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
6	6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
7	6068	0.000000	0.000000	0.000000	0.476325	0.000000	0.000000	
8	6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
9	6068	0.379839	0.000000	0.000000	0.000000	0.000000	0.000000	
10	6068	0.000000	0.000000	0.424029	0.000000	0.000000	0.000000	
11	6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
12	6068	0.591838	0.000000	0.000000	0.000000	0.000000	0.000000	
13	6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
14	6068	0.000000	0.654946	0.000000	0.000000	0.000000	0.000000	
15	6068	0.000000	0.000000	0.000000	0.000000	0.395053	0.395053	
16	6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
17	6068	0.000000	0.000000	0.000000	0.000000	0.319194	0.319194	
18	6068	0.000000	0.000000	0.000000	0.000000	0.395053	0.395053	
19	6068	0.000000	0.635499	0.000000	0.000000	0.000000	0.000000	
20	6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
21	6068	0.000000	0.000000	0.000000	0.370995	0.000000	0.000000	
22	6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
23	6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
24	6068	0.000000	0.000000	0.000000	0.000000	0.358590	0.358590	
25	6068	0.000000	0.000000	0.416058	0.000000	0.000000	0.000000	
26	6068	0.000000	0.000000	0.365621	0.000000	0.000000	0.000000	
27	6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
28	6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
29	6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
30	6068	0.000000	0.000000	0.000000	0.318064	0.000000	0.000000	
31	6068	0.000000	0.000000	0.000000	0.365621	0.000000	0.000000	
32	6068	0.000000	0.351563	0.000000	0.000000	0.000000	0.000000	
33	6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
34	6068	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
35	6068	0.404515	0.000000	0.000000	0.000000	0.000000	0.000000	

net git data difference

0	0.000000	0.000000	0.408078	0.000000
1	0.000000	0.000000	0.000000	0.000000
2	0.000000	0.709227	0.000000	0.354614
3	0.000000	0.000000	0.000000	0.503013
4	0.000000	0.000000	0.000000	0.000000
5	0.000000	0.000000	0.000000	0.000000
6	0.000000	0.000000	0.000000	0.000000
7	0.000000	0.000000	0.000000	0.000000
8	0.000000	0.000000	0.000000	0.000000
9	0.000000	0.000000	0.408078	0.000000
10	0.000000	0.000000	0.000000	0.000000
11	0.000000	0.000000	0.000000	0.000000
12	0.000000	0.317919	0.000000	0.000000
13	0.000000	0.000000	0.000000	0.000000
14	0.000000	0.000000	0.000000	0.000000
15	0.395053	0.000000	0.000000	0.000000
16	0.000000	0.000000	0.000000	0.000000
17	0.319194	0.000000	0.000000	0.000000
18	0.395053	0.000000	0.000000	0.000000
19	0.000000	0.000000	0.000000	0.000000
20	0.000000	0.000000	0.000000	0.000000
21	0.000000	0.000000	0.000000	0.398576
22	0.000000	0.000000	0.000000	0.000000
23	0.000000	0.000000	0.000000	0.000000
24	0.358590	0.000000	0.000000	0.000000
25	0.000000	0.000000	0.000000	0.000000
26	0.000000	0.000000	0.000000	0.000000
27	0.000000	0.000000	0.000000	0.000000
28	0.000000	0.000000	0.000000	0.000000
29	0.000000	0.355684	0.000000	0.000000
30	0.000000	0.000000	0.000000	0.000000
31	0.000000	0.000000	0.000000	0.000000
32	0.000000	0.000000	0.000000	0.000000
33	0.000000	0.000000	0.000000	0.000000
34	0.000000	0.000000	0.441140	0.000000
35	0.000000	0.000000	0.000000	0.000000

\*\*\*\*\*

For Username ID TF/IDF table : 63051

	usernameid	vs	bash	java	list	instance	python \
0	63051	0.000000	0.334417	0.000000	0.000000	0.000000	0.000000
1	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	63051	0.299971	0.000000	0.000000	0.000000	0.000000	0.000000
3	63051	0.000000	0.322478	0.000000	0.000000	0.000000	0.000000

4	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
5	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
6	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
7	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
8	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
9	63051	0.000000	0.000000	0.000000	0.000000	0.801591	0.000000
10	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
11	63051	0.241678	0.000000	0.546575	0.000000	0.000000	0.000000
12	63051	0.000000	0.322478	0.000000	0.000000	0.000000	0.000000
13	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
14	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
15	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
16	63051	0.000000	0.000000	0.000000	0.434222	0.000000	0.000000
17	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
18	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
19	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
20	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
21	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
22	63051	0.000000	0.000000	0.000000	0.000000	0.474023	0.000000
23	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.457567
24	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25	63051	0.422192	0.000000	0.000000	0.477411	0.000000	0.000000
26	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
27	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
28	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
29	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
30	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
31	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.434222
32	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
33	63051	0.000000	0.000000	0.355579	0.000000	0.000000	0.000000
34	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
35	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
36	63051	0.000000	0.000000	0.526650	0.000000	0.000000	0.000000
37	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
38	63051	0.000000	0.422192	0.000000	0.000000	0.000000	0.000000
39	63051	0.000000	0.251242	0.000000	0.000000	0.000000	0.000000
40	63051	0.462809	0.000000	0.000000	0.000000	0.000000	0.000000
41	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
42	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.355579
43	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
44	63051	0.000000	0.000000	0.000000	0.445436	0.000000	0.000000
45	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
46	63051	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
47	63051	0.522192	0.000000	0.000000	0.000000	0.000000	0.000000

	linux	redirect	kill	output
0	0.000000	0.000000	0.000000	0.378156
1	0.000000	0.000000	0.000000	0.000000

2	0.000000	0.339204	0.000000	0.000000
3	0.000000	0.000000	0.000000	0.364656
4	0.445436	0.000000	0.481991	0.000000
5	0.000000	0.000000	0.000000	0.000000
6	0.327146	0.000000	0.000000	0.000000
7	0.000000	0.000000	0.000000	0.000000
8	0.000000	0.000000	0.000000	0.000000
9	0.000000	0.000000	0.000000	0.000000
10	0.000000	0.000000	0.000000	0.000000
11	0.000000	0.000000	0.000000	0.000000
12	0.000000	0.000000	0.000000	0.000000
13	0.000000	0.000000	0.000000	0.000000
14	0.000000	0.000000	0.000000	0.000000
15	0.000000	0.000000	0.000000	0.000000
16	0.000000	0.000000	0.000000	0.000000
17	0.000000	0.000000	0.000000	0.000000
18	0.000000	0.000000	0.000000	0.000000
19	0.000000	0.000000	0.000000	0.000000
20	0.000000	0.000000	0.000000	0.000000
21	0.000000	0.000000	0.000000	0.000000
22	0.000000	0.000000	0.000000	0.000000
23	0.000000	0.000000	0.000000	0.000000
24	0.000000	0.000000	0.000000	0.000000
25	0.000000	0.000000	0.000000	0.000000
26	0.331862	0.000000	0.000000	0.000000
27	0.000000	0.000000	0.000000	0.000000
28	0.000000	0.000000	0.000000	0.000000
29	0.000000	0.000000	0.000000	0.000000
30	0.000000	0.000000	0.000000	0.000000
31	0.000000	0.000000	0.000000	0.000000
32	0.000000	0.000000	0.000000	0.000000
33	0.000000	0.000000	0.000000	0.000000
34	0.000000	0.000000	0.000000	0.000000
35	0.000000	0.000000	0.000000	0.000000
36	0.000000	0.000000	0.569869	0.000000
37	0.000000	0.000000	0.000000	0.000000
38	0.000000	0.477411	0.000000	0.000000
39	0.000000	0.284102	0.000000	0.284102
40	0.000000	0.000000	0.000000	0.000000
41	0.000000	0.000000	0.000000	0.000000
42	0.000000	0.000000	0.000000	0.000000
43	0.000000	0.000000	0.000000	0.000000
44	0.000000	0.000000	0.000000	0.000000
45	0.000000	0.000000	0.000000	0.000000
46	0.000000	0.000000	0.000000	0.000000
47	0.000000	0.000000	0.000000	0.000000

\*\*\*\*\*

For Username ID TF/IDF table : 87234

	usernameid	operator	array	processing	copy	idiom	swap \
0	87234	0.0	0.000000	0.000000	0.57735	0.57735	0.57735
1	87234	1.0	0.000000	0.000000	0.00000	0.00000	0.00000
2	87234	0.0	0.603023	0.603023	0.00000	0.00000	0.00000

	faster	sorted	unsorted
0	0.000000	0.000000	0.000000
1	0.000000	0.000000	0.000000
2	0.301511	0.301511	0.301511

\*\*\*\*\*

For Username ID TF/IDF table : 89904

	usernameid	java	git	undo	jar	appending \
0	89904	0.000000	0.403194	0.403194	0.000000	0.000000
1	89904	0.000000	0.000000	0.000000	0.391176	0.000000
2	89904	0.000000	0.000000	0.000000	0.000000	0.000000
3	89904	0.313903	0.000000	0.000000	0.000000	0.000000
4	89904	0.000000	0.457985	0.457985	0.000000	0.000000
5	89904	0.356655	0.000000	0.000000	0.000000	0.000000
6	89904	0.000000	0.000000	0.000000	0.440595	0.000000
7	89904	0.000000	0.000000	0.000000	0.000000	0.707107
8	89904	0.423549	0.000000	0.000000	0.000000	0.000000
9	89904	0.313903	0.000000	0.000000	0.000000	0.000000

	objectoutputstream	swing	timer	open	profilers
0	0.000000	0.000000	0.000000	0.000000	0.000000
1	0.000000	0.000000	0.000000	0.000000	0.000000
2	0.000000	0.000000	0.000000	0.000000	0.000000
3	0.000000	0.000000	0.000000	0.000000	0.000000
4	0.000000	0.000000	0.000000	0.000000	0.000000
5	0.000000	0.000000	0.000000	0.539382	0.539382
6	0.000000	0.000000	0.000000	0.000000	0.000000
7	0.707107	0.000000	0.000000	0.000000	0.000000
8	0.000000	0.640549	0.640549	0.000000	0.000000
9	0.000000	0.000000	0.000000	0.000000	0.000000

\*\*\*\*\*

For Username ID TF/IDF table : 95592

	usernameid	install	pip	installed	python	virtualenv	cache	\
0	95592	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
1	95592	0.000000	0.000000	0.360493	0.000000	0.360493	0.000000	
2	95592	0.000000	0.000000	0.435556	0.000000	0.000000	0.000000	
3	95592	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
4	95592	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
5	95592	0.347708	0.347708	0.000000	0.347708	0.347708	0.000000	
6	95592	0.454195	0.454195	0.000000	0.000000	0.000000	0.541948	
7	95592	0.000000	0.000000	0.000000	0.386479	0.000000	0.000000	

	local	determining	flask	version
0	0.000000	0.000000	0.000000	0.000000
1	0.000000	0.000000	0.000000	0.000000
2	0.000000	0.519708	0.519708	0.519708
3	0.000000	0.000000	0.000000	0.000000
4	0.000000	0.000000	0.000000	0.000000
5	0.000000	0.000000	0.000000	0.000000
6	0.541948	0.000000	0.000000	0.000000
7	0.000000	0.000000	0.000000	0.000000

\*\*\*\*\*

For Username ID TF/IDF table : 9951

	usernameid	git	javascript	python	string	file	callable	\
0	9951	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	
1	9951	0.000000	0.000000	0.000000	0.455000	0.000000	0.0	
2	9951	0.541841	0.000000	0.000000	0.000000	0.000000	0.0	
3	9951	0.000000	0.493950	0.000000	0.000000	0.000000	0.0	
4	9951	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	
5	9951	0.000000	0.000000	0.000000	0.000000	0.000000	1.0	
6	9951	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	
7	9951	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	
8	9951	0.360453	0.000000	0.000000	0.000000	0.000000	0.0	
9	9951	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	
10	9951	0.258934	0.000000	0.000000	0.000000	0.000000	0.0	
11	9951	0.000000	0.000000	0.000000	0.662735	0.000000	0.0	
12	9951	0.000000	0.372751	0.000000	0.000000	0.000000	0.0	
13	9951	0.000000	0.372751	0.000000	0.000000	0.000000	0.0	
14	9951	0.000000	0.000000	0.000000	0.000000	0.530479	0.0	
15	9951	0.000000	0.000000	0.483550	0.000000	0.000000	0.0	
16	9951	0.288828	0.000000	0.000000	0.000000	0.000000	0.0	

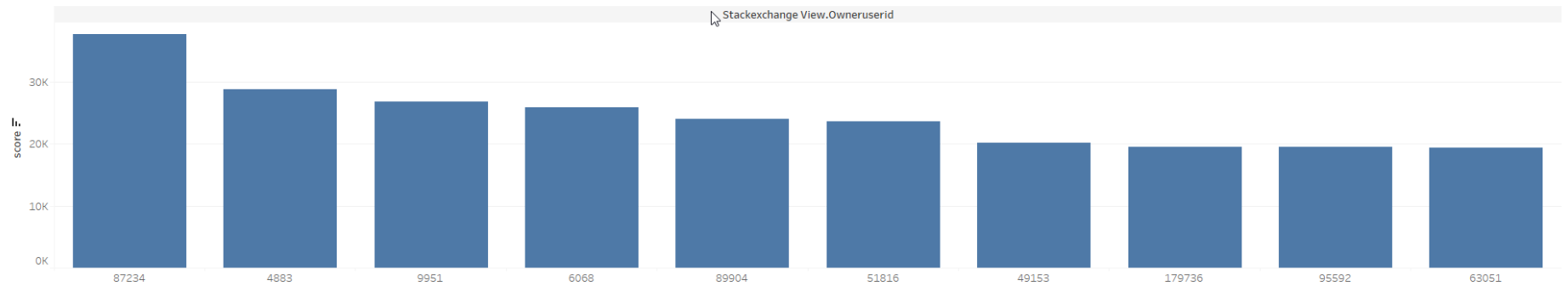
17	9951	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
18	9951	0.000000	0.000000	0.662735	0.000000	0.000000	0.0
19	9951	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
20	9951	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
21	9951	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
22	9951	0.306795	0.000000	0.000000	0.000000	0.000000	0.0
23	9951	0.213534	0.000000	0.000000	0.000000	0.586267	0.0

	android	dictionary	url	way
0	0.00000	0.000000	0.000000	0.000000
1	0.00000	0.000000	0.000000	0.000000
2	0.00000	0.000000	0.000000	0.000000
3	0.00000	0.000000	0.000000	0.000000
4	0.00000	0.000000	0.000000	0.000000
5	0.00000	0.000000	0.000000	0.000000
6	0.00000	0.530479	0.000000	0.000000
7	0.00000	0.000000	0.468624	0.000000
8	0.00000	0.000000	0.000000	0.000000
9	0.00000	0.000000	0.000000	0.000000
10	0.00000	0.000000	0.000000	0.000000
11	0.00000	0.000000	0.000000	0.000000
12	0.00000	0.000000	0.000000	0.000000
13	0.00000	0.000000	0.000000	0.000000
14	0.00000	0.000000	0.000000	0.000000
15	0.48355	0.000000	0.000000	0.483550
16	0.00000	0.000000	0.396495	0.000000
17	0.00000	0.000000	0.000000	0.000000
18	0.00000	0.000000	0.000000	0.000000
19	0.00000	0.000000	0.000000	0.000000
20	0.45500	0.000000	0.000000	0.000000
21	0.00000	0.375106	0.000000	0.375106
22	0.00000	0.000000	0.000000	0.000000
23	0.00000	0.000000	0.000000	0.000000

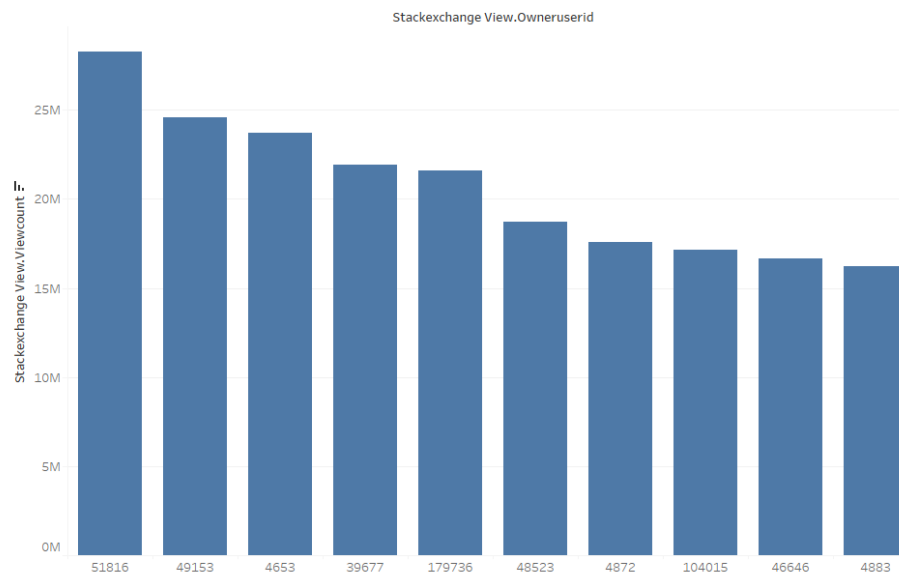
\*\*\*\*\*

## Analytics to verify query output

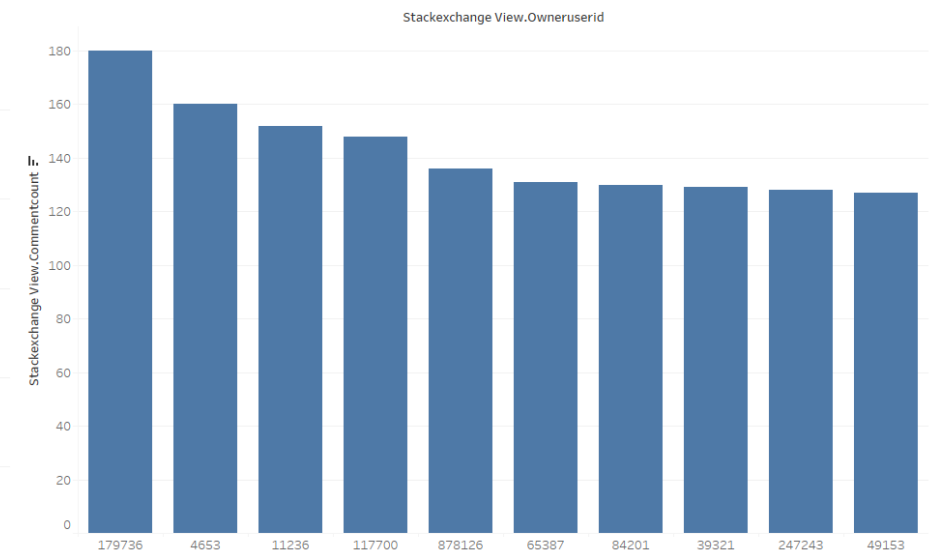
Top 10 user ID with highest score.



Top 10 user ID with highest view count



Top 10 user ID with highest comment count



Top 10 Score Top 10 comment count Top 10 view count Dashboard 1

