Social (Twitter) Data analysis of User Profiling

[Big Data and Social Networks project 2017/2018]

Surbhi Sonkiya
Matricola # 196511
University of Trento
Via Sommarive, 9
Povo TN, Trento - 38123
surbhi.sonkiya@studenti.unitn.it

Syed Zohaib Hassan
Matricola # 196520
University of Trento
Via Sommarive, 9
Povo TN, Trento - 38123
syedzohaib.hassan@studenti.unitn.it

1. DESCRIPTION

The objective of this work is to perform an optimal user profiling by learning from various methodologies and techniques which has been used already in related works, and thereby apply one of the best suited techniques in real [1, 2, 3]. By user profiling we aim to understand how prominently a user tweets about a specific topic from a cleaned dataset of 1.1 million rows.

We also aim to execute the task (code) in a distributed environment set-up, on a Linux machine in order to understand the parallelized and distributed computation concept.

2. RELATED WORK

For user profiling, there are several works performed in the past, mainly in the below mentioned two aspects:

i. Sentiment analysis - It is an analysis of user's sentiments like if the user was happy, sad, angry, upset and so on.

ii. Topic modelling - It is an analysis of the prominent list of topics a user tweets about.

We chose to work on the later aspect to generate user profiles, as we wanted to understand the implementation of Latent Dirichlet Allocation (LDA) algorithm. It's interesting to learn the various possibilities LDA brings in for clustering information and calculating statistics based on word-model to generate user profiles. Our learning about LDA is explained theoretically in subsection 2.3 and its practical implementation is explained in sub-subsection 3.1.8.

In the following subsections, we introduce the concepts, techniques and methodologies that has been used to achieve the objective of this project.

2.1 Spark Dataframes

This project uses Spark Dataframes instead of Hadoop MapReduce, the main reason being Spark uses memory instead of disc to process parallelized data improving the performance by many folds. Spark Dataframes are distributed datasets. The look and feel of dataframes is similar to that of tables in a relational database. In other words, dataframes comprises of table like structures with a name for each column.

Table 1: Advantages of Spark Dataframes for the purpose of user profiling

Features	Spark Data frames	Spark RDD	Spark SQL	Spark Data sets	Hadoop Map Reduce
Compile-time type safety (lambda functions, etc.)	—	Yes	—	Yes	—
Can be created from structured or semi-structured data files	Yes	—	—	—	—
Conceptually like tables in a relational database	Yes	—	—	—	—
Optimized execution engine	Yes	—	Yes	Yes	—
Distributed task execution	Yes	Yes	—	Yes	—
Execute SQL queries	Yes	—	Yes	—	—
Performance issues over other techniques due to more number of intermediate steps	—	Yes	—	—	Yes

For the purpose of this work, to perform user profiling, Spark dataframes has been used as it overweighs the advantages over other techniques as shown in the Table 1 [4, 5, 6, 7, 8, 9, 10, 11].

2.2 Filtering

Filtering is a process to perform selection of the elements of the list to work upon, thereby removing stop words, symbols, and many unwanted data from the dataset dump to prepare more accurate dataset. It uses the basic concept of

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

lambda, map and reduce functions [12, 13].

2.3 Clustering

Clustering is a task to combine entities under one group; where the characteristics of the entities within the same group are similar to each other, however, they vary from the characteristics of the entities of other groups. These groups are called clusters. It is a method of unsupervised learning where these clusters are formed based on observation.

There are several techniques to achieve clustering like K-means, Latent Dirichlet Allocation (LDA) and Gaussian Mixture Model (GMM). However, all these techniques have one common concept i.e. to cluster elements based on "similarity concept" with the other elements present in total to be clustered.

For this project, we perform clustering using Latent Dirichlet Allocation (LDA) technique. LDA is a technique where the algorithm automatically learns and determines topics prevalent in a document. This is also known as document clustering [14, 15, 16]. It provides two advantages for performing topic-modelling based on word-level through which we can retrieve statistics about:

i. How many times did a word appear under various topics.

ii. How many times did a topic appear in the document.

2.4 Distributed Task

A task can produce an output by executing it on a single machine. But, in case the task has millions of rows of data to be computed, the time taken is tremendous to execute it on a single machine. Therefore, to achieve the same task in comparatively less time, Spark introduces Spark Cluster mode.

Cluster mode implements the master and worker nodes which are managed through a cluster manager in between the master and worker nodes. There can be n number of master and worker nodes in one network managed by a single cluster manager.

When a task is submitted to the master node, cluster manager takes care of splitting the entire task into smaller sections and distributing it between all the worker nodes. Once, the task execution is completed by the worker, it returns it's results back to the cluster manager. Cluster manager keeps a check on the pending tasks, and assigns a new task to the worker after the respective worker reports the previous assigned task completed successfully. Thereby making the entire task execute in parallel. It also provides fault-tolerance in case any of the master or worker machines collapse in between the task execution.

Spark supports various types of cluster managers, namely, Standalone, Yarn and Mesos. Standalone cluster manager comes with Spark, while the other two comes from Hadoop and Apache respectively.

For this project, we will use Spark Standalone cluster manager as it is easy to set-up, simple to use and serves the purpose [17, 18].

3. SOLUTION

Figure 1 shows the workflow of this project to generate user profile from the twitter (social network) dataset.

Figure 3 shows the workflow for the set-up of the distributed spark standalone cluster mode.

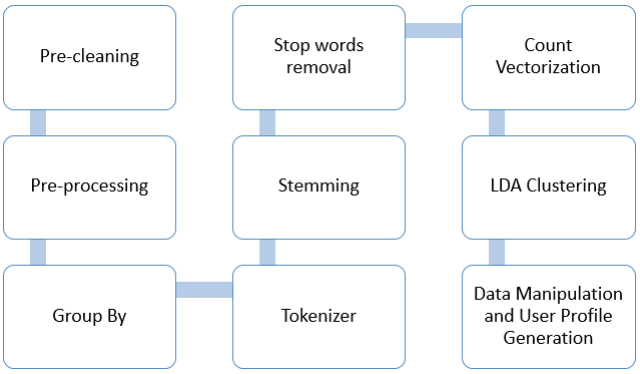


Figure 1: Workflow to generate user profile

3.1 User Profiling

This section explains the step by step process followed to generate user profile from the unstructured twitter dataset.

3.1.1 Pre-cleaning

The dataset used for user profiling was provided by professor and we used tweet-text-26.csv in our project. Since labels for each column was not provided so by looking at twitter API default dataset we manage to identify type of information available in csv file, it had information about tweetid, userid, text, latitude, longitude and two null valued columns at the end, which make up seven columns/values.

While importing the data, we observed that it resulted in more than seven columns for many rows, after further probing it was realized that it was due to the presence of commas in the text of tweet. As file was saved in csv and when we tried to import the data these commas were considered as the start of a new value/column thus creating more than supposed to be seven columns.

We worked upon pre-cleaning of this dataset and merged values of tweets into one value to correct the number of columns to seven and we discarded the latitude, longitude and two null columns as they were of no use. We used **Graphlab** and **Pandas** for this purpose.

3.1.2 Pre-processing

After pre-cleaning of the dataset, there was a need for pre-processing of the dataset. In this process we managed to remove below three items from the tweets in the same order as mentioned. For this purpose, we used **Re library** in python.

i. Hyperlink removal

ii. Symbols (@, $, etc) removal

iii. Digits removal

3.1.3 Group By

In this process, we perform a group by in **Spark** to group the userid and tweets associated with the respective userid.

3.1.4 Tokenizer

Tokenizer is the feature transformation property in Spark that enables us to form tokens with the dataset. **Regex-Tokenizer** module in Spark that allows us to extract words from the tweets of a user [19].

In this project, tokens are an array of strings comprising of the words w_i used by the user u_i.

3.1.5 Stemming

Stemming is a process that cuts down all the conjugated words into one basic form of the word (most likely the verb). **PorterStemmer** is the **NLTK interface** imported in Spark and used for this purpose [20].

For example, given a set of words like play, playing, played, etc are counted as one single word "play". Similarly, words like am, are, is are counted as "be".

3.1.6 Stop Words Removal

This process drops all the words from the array of strings that does not have a specific meaning (like and, or, the, a, had etc). **StopWordsRemover** module in Spark is used for this purpose [21].

Additionally, we appended "https" in the defaults of the word removal. This is done to drop the word "https" from all the tweets since "https" is left after the removal of hyperlinks.

3.1.7 Count Vectorization

The words from the global vocabulary used by the user in every tweet is counted. For this project, we have taken into account the first 1000 words present in the global dictionary. Indexes present in the global dictionary for each of these words used by the user are stored against the respective userid. This is called dense vector [22].

CountVectorizer, a feature extractor module in Spark is used for this purpose.

3.1.8 LDA Clustering

It is typically used to detect underlying topics in text documents by creating a statistical model to discover abstract topics that occur in collection of documents, originally proposed by BLei aet al in 2003. Assumptions taken in LDA algorithm are words carry strong semantic information and documents discussing similar topics will have similar group of words, latent topics are therefore discovered by identifying group of words that frequently occur together. Second assumption is regarding the structure of the documents themselves, it says documents are probabilistic distribution over latent topics and topics are probabilistic distribution over words. Every document contains a number of topics and each topic has distribution of words associated with it. LDA works with probability distribution rather than word frequency.

Plate notations is concise way of visually representing the dependencies among the model parameters, as shown in figure 2 [23]. The larger rectangle represented by D is the total number of documents within the corpus and the smaller rectangle represented by N donates the number of words in the documents.The parameters depending on where they fall inside of those two rectangles that indicates whether they apply at the document level or at word level or both. Two parameters exist outside rectangles are α and β, they are Dirichlet prior. α is the parameter that describe prior per document topic distribution, higher α indicates that each document is to contain a mixture of most of the topics not just one or two topics in particular. Conversely, a low α imply that each document will likely contain just few of the topics. β is the Dirichlet prior parameter which describe per topic word distribution, high β indicates that each topic will contain mixture of most in the words while low β means that each topic may contain a mixture of just a few words. θ is the topic distribution for the document d, $Z_{d,n}$ is the topic

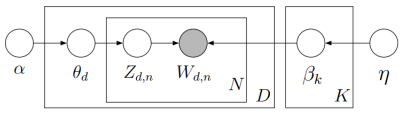


Figure 2: Graphic Model of LDA Algorithm

of the n^{th} word in the document d and $W_{d,n}$ represents each specific word.

LDA, is a module of mlib library of spark which takes in multiple parameters to optimize the model as per user needs such as number of topics (cluster centers), maxIterations which defines the number of expectation maximization iterations.

3.1.9 Data Manipulation and User Profile Generation

In this section, we manipulate the data obtained post LDA clustering. By manipulation it means that we have selected certain columns and rows data from the output of the LDA algorithm to obtain an optimal user profile.

The topic t_i is determined by the five most used terms under a particular topic t_i. Also, their corresponding probabilities related to the topic t_i is calculated.

User profile consists of the topics t_i a user u_i has tweeted about and the term weights of each topic t_i.

We also generate a list of topics t_i and the count of users who used the topic t_i in their tweets.

3.2 Spark Standalone Cluster Mode

There should be passwordless connection set-up between master and worker machines [24, 25, 26, 27]. This is important to do before the start of the cluster mode set-up between master and worker machines [28, 29, 30, 31, 32].

3.2.1 Generate SSH key

From the command line of the master machine, type the command "ssh-keygen -t rs"'

A public-private key is generated inside the folder ".ssh" on the master machine.

3.2.2 Master-worker connection

Create a directory with name ".ssh". Inside this directory, create a file named "authorized_keys" on the worker machine.

Navigate to the folder "./ssh". Copy the key inside the file "id_rsa.pub" from the master machine to the "authorized_keys" file on the worker machine.

Type in command line "chmod 600 authorized_key" to modify the ownership of the file "authorized_keys" present on the worker machine. This is to allow the master machine log into the worker machine without the need to enter the password for the worker machine.

3.2.3 SSH between master-worker

From the command line of master machine, type "ssh username@IP"

Type username and IP address of the worker machine in the above command. Master machine should be able to log into worker machine at this point.

Also, connection from worker to master machine could be tested in the same way. Type the command from worker

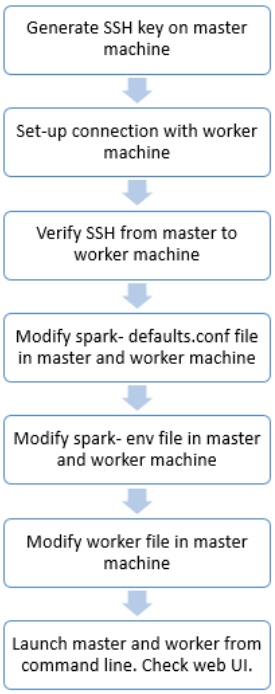


Figure 3: Workflow for Spark Standalone Cluster Mode

machine providing username and IP address of the master machine.

3.2.4 *Modify spark-defaults.conf*

On the master machine, navigate to the Spark Standalone conf folder. Create a copy of "spark-defaults.conf.template" file and rename it as "spark-defaults.conf".

Modify the properties according to the requirement of the project. We mentioned below property in this file.

i. spark.master = spark://<IP or username of the master machine>:7077.

7077 is the default port number for spark. Save and close the file. Copy this file from master machine and paste it inside the same Spark conf folder on all the worker machines.

3.2.5 *Modify spark-env*

On the master machine, navigate to the Spark Standalone conf folder. Create a copy of "spark-env.sh.template" file and rename it as "spark-env.sh".

Modify the properties according to the requirement of the project. We mentioned below properties in this file.

i. SPARK_WORKER_INSTANCES = 2 (If not specified, by default only one worker instance is launched).

ii. SPARK_MASTER_IP = <IP address of master machine>

iii. SPARK_WORKER_IP = <IP address of worker machines>

Save and close the file. Copy this file from master machine and paste it inside the same Spark conf folder on all the worker machines.

3.2.6 *Modify slaves*

On the master machine, navigate to the Spark Standalone conf folder. Create a copy of "slaves.template" file and rename it as "slaves". Mention the IP address of all the worker inside this file in an order, one IP address per each line. Save and close it.

4. EXPERIMENTS

Dataset of 1.13 million rows is taken into account for this project. This dataset has been extracted from the dataset of **total 8.08 million rows** for those **users** who tweeted **more than ten tweets**.

Figures 4 to 17 shows the transition of an unstructured dataset of 1.1 million rows to the final output - user profiling. It contains an example of how the data has been cleaned, processed, grouped, tokenized, and stemmed. It also displays stop words removal, vectorization, LDA clustering and data manipulation for user profile generation.

The entire task has been executed on Spark Standalone Cluster Mode, using different combinations of the configuration properties (executor-memory, etc) of master machine and the number of executors as workers. **Figures 18 to 26** provides further details about the testing in real.

Additionally, time taken for the entire execution for different sizes of dataset has also been tested and noted down. **Figure 27** explains this in a graphical format.

5. CONCLUSIONS

The pre-cleaning and pre-processing of the dataset has been completed outside of spark mainly using pandas. However, grouping the data, tokenization, stemming, stop words removal, vectorization, clustering, data manipulation to obtain users profile has been completed in spark using various spark modules and an external interface (NLTK).

The distributed task has been performed via Spark Standalone Cluster Mode manager. The master and worker nodes were configured and the entire task execution was completed in the master-worker distributed environment.

The major learning from this project are stated below.

i) **Stemming** - We realized that integrating stemming in the process of obtaining users profile could be very powerful in the sense that it chops down the word to its basic form and produces an output that plays a big role to obtain maximum optimum in clustering process. Since spark doesn't have it's own module, we imported NLTK interface to perform this.

ii) **Clustering using LDA** - This project provided an opportunity to the authors to realize the LDA algorithm hands on. Implementing the algorithm gave a clarity about how to declare the number of topics, how the algorithm forms the cluster of words under a topic, and how the number of iterations affect in the process of obtaining the optimal center. Number of iterations is directly proportional to the quality of clustering (the better optimal cluster obtained).

iii) **SSH** - The authors had a first time experience in installing and configuring passwordless SSH in order to enable the master to log into the worker node without the password of the worker node and vice versa. We learned that passwordless connection between both the nodes was required so that there is no prompt asking for password while launching the master and worker nodes.

iv) **Spark Standalone Cluster Mode** - Adding master and worker nodes can be very challenging. However, having been able to do it, gives a sense of satisfaction of achieving the objective of the project. Initially, we tried the connection of linux based master node with windows based worker node,

```
data[index[0]]
```

```
['884756598257704961',
 '2402341338',
 'RT @jayy_mingg: I look down and say some more corny shit like \\ yeen lil yatchy',
 ' but you might need a lil boat\\"  he was older so h\xe2\x80\xa6"',
 '0.000000000000',
 '0.000000000000',
 '',
 '']
```

Figure 4: Unstructured dataset before pre-cleaning

Tweet that has 8 columns because there is a comma in the tweets which is considered as a new column due to input file type as CSV.

```
data[index[0]]
```

```
['884756598257704961',
 '2402341338',
 'RT @jayy_mingg: I look down and say some more corny shit like \\ yeen lil yatchy but you might need a lil boat\\"  he was old
er so h\xe2\x80\xa6"',
 '0.000000000000',
 '0.000000000000',
 '',
 '']
```

Figure 5: Dataset after pre-cleaning

Tweet combined and merged into single column. After pre-cleaning, total 7 columns in the tweet remains.

```
data[40][2]
```

```
'RT @CGTNOfficial: 10-year-old Chinese boy killed, 25 injured in Thailand tour bus crash https://t.co/ioIKRaJDRU'
```

Figure 6: Before Pre-processing

Dataset with hyperlinks, symbols, and numbers before pre-processing

```
data[40][2]
```

```
'RT  CGTNOfficial   year old Chinese boy killed   injured in Thailand tour bus crash '
```

Figure 7: After Pre-processing

Dataset without hyperlinks, symbols, and numbers after pre-processing

user_id	tweet
820694	rt highsnobiety should imprisoned weed dealers be freed when marijuana is legalized
820694	rt mskellymhayes it s crucial to observe that the republican party is not governing like an entity that anticipates losing power they ar
820694	rt sophia_mjones iranian cancer researcher w valid visa was set to start work at boston children s hosp us sent him back to iran
820694	senkamalaharris marinmaven yeeeeeeeeeeeeeeeeup i mean if you re running for president of the united states
820694	mammaloves keithboykin oh you know there s gonna be consequences internal disciplinary actions can be rougher t
820694	rt anamariecox i am as good at photo editing as trump is as presidenting
820694	hbryant i had to do it d
820694	rt maggienyt macron says next question should go to an american jounralist as trump makes the call trump doesn t call on american journ
820694	rt ew see exclusive first look photos of oprah mindykaling rwitherspoon and more in a wrinkle in time
820694	war criminal george bush is a war criminal george bush is a war criminal george bush is a war criminal georg
820694	rt xodanix what better way to erase ndn ppl while celebrating slavery than taking a ndn symbol that s regularly appropriated slap conf
820694	rt sengillibrand dearbetsy i ask you to listen to these survivors and understand just why we re in this fight
820694	rt rationalbassist how do you justify decrying the aca as a great evil while also including exemptions that allow you to receive aca
820694	lawyers weeks
820694	rt blackamazon if you say no one saw this coming you re saying anyone who ain t you is no one
820694	so there were almost as many people at that trumprussia meeting as at trump s inaugural
820694	rt topherspiro breaking in rare joint letter insurers say cruz amendment unworkable in any form and will lead to widespread t
820694	rt swiftonsecurity photographer broke after peta sues him saying monkeys own their pictures
820694	rt keegannyc large crowd gathered in brooklyn for the two year anniversary of the in custody death of sandrabland
820694	rt aaaaaaaaaaronn mxmovement crimesofbrits what the hell twitter so you can be a nazi calling everyone pedos but can t tweet out
820694	you re almost there sean medicaid cuz their target has always been destroying medicare all of social security
820694	blackamazon giiiiiiiiiiiirl i looked like i had swallowed a pound watermelon because i may be thick ish but my
820694	rt adamjkovac metallica comes to town on tuesday but i have to miss it to go on my stupid honeymoon
820694	rt metrouk john mcdonnell stands by claim grenfell tower fire was social murder
820694	america_ _cymatics_ davidgraeber a million time this there is no going back i truly believe it s a pivotal h

Figure 8: Before Group by

One userid with many tweets considered as different rows before Group by

|820694 |rt highsnobiety should imprisoned weed dealers be freed when marijuana is legalized rt mskellymhayes it s crucial to observe that the republican party is not governing like an entity that anticipate
s losing power they ar rt sophia_mjones iranian cancer researcher w valid visa was set to start work at boston children s hosp us sent him back to iran senkamalaharris marinmaven yeeeeeeeeeeeeeeeup i me
an if you re running for president of the united states mammaloves keithboykin oh you know there s gonna be consequences internal disciplinary actions can be rougher t rt anamariecox i am as good at phot
o editing as trump is as presidenting hbryant i had to do it d rt maggienyt macron says next question should go to an american jounralist as trump makes the call trump doesn t call on american journ rt ew
see exclusive first look photos of oprah mindykaling rwitherspoon and more in a wrinkle in time war criminal george bush is a war criminal george bush is a war criminal george bush is a war criminal
georg rt xodanix what better way to erase ndn ppl while celebrating slavery than taking a ndn symbol that s regularly appropriated slap conf rt sengillibrand dearbetsy i ask you to listen to these surv
ivors and understand just why we re in this fight rt rationalbassist how do you justify decrying the aca as a great evil while also including exemptions that allow you to receive aca lawyers weeks rt
blackamazon if you say no one saw this coming you re saying anyone who ain t you is no one so there were almost as many people at that trumprussia meeting as at trump s inaugural rt topherspiro breaki
ng in rare joint letter insurers say cruz amendment unworkable in any form and will lead to widespread t rt swiftonsecurity photographer broke after peta sues him saying monkeys own their pictures rt k
eegannyc large crowd gathered in brooklyn for the two year anniversary of the in custody death of sandrabland rt aaaaaaaaaaronn mxmovement crimesofbrits what the hell twitter so you can be a nazi callin
g everyone pedos but can t tweet out you re almost there sean medicaid cuz their target has always been destroying medicare all of social security blackamazon giiiiiiiiiiiirl i looked like i had swallowed
a pound watermelon because i may be thick ish but my rt adamjkovac metallica comes to town on tuesday but i have to miss it to go on my stupid honeymoon rt metrouk john mcdonnell stands by claim grenfell to
wer fire was social murder america_ _cymatics_ davidgraeber a million time this there is no going back i truly believe it s a pivotal h and in what s probably the most epic of blackwomenbeentellinyall her
e s the pentagon s dirge to the american empi|

Figure 9: After Group by

One userid with many tweets considered as one row after Group by

user_id	concat_ws(, collect_list(tweet))	tokens	token_count
820694	rt highsnobiety should imprisoned weed deale...	[highsnobiety, should, imprisoned, weed, deale...	258
9409672	reuters british government declines to publis...	[reuters, british, government, declines, publi...	139
14731218	oh man meggu on the warpath again glad i hav...	[meggu, warpath, again, glad, have, full, thun...	114
15606755	rt luliortiztv flgovscott officially signi...	[luliortiztv, flgovscott, officially, signing,...	131
16738576	rt jemillerwbal actual email sent to donald ...	[jemillerwbal, actual, email, sent, donald, tr...	202

Figure 10: Tokenizer

Tokens generated from words used by user in the tweets.

user_id	concat_ws(, collect_list(tweet))	tokens	Stemmed_tokens
820694	rt highsnobiety should imprisoned weed deale...	[highsnobiety, should, imprisoned, weed, deale...	[highsnobieti, should, imprison, weed, dealer,...
9409672	reuters british government declines to publis...	[reuters, british, government, declines, publi...	[reuter, british, govern, declin, publish, rev...
14731218	oh man meggu on the warpath again glad i hav...	[meggu, warpath, again, glad, have, full, thun...	[meggu, warpath, again, glad, have, full, thun...
15606755	rt luliortiztv flgovscott officially signi...	[luliortiztv, flgovscott, officially, signing,...	[luliortiztv, flgovscott, offici, sign, hous, ...
16738576	rt jemillerwbal actual email sent to donald ...	[jemillerwbal, actual, email, sent, donald, tr...	[jemillerwb, actual, email, sent, donald, trum...

Figure 11: Stemming

In the first row, “imprisoned” stemmed as “imprison”. Similarly, in the second row, [“reuters”, “government”] stemmed as [“reuter”, “govern”] respectively.

user_id	concat_ws(, collect_list(tweet))	tokens	Stemmed_tokens	filtered	Post_tokens
820694	rt highsnobiety should imprisoned weed deale...	258	[highsnobieti, should, imprison, weed, dealer,...	[highsnobieti, imprison, weed, dealer, freed, ...	226
9409672	reuters british government declines to publis...	139	[reuter, british, govern, declin, publish, rev...	[reuter, british, govern, declin, publish, rev...	125
14731218	oh man meggu on the warpath again glad i hav...	114	[meggu, warpath, again, glad, have, full, thun...	[meggu, warpath, glad, full, thunderbro_, dmel...	94
15606755	rt luliortiztv flgovscott officially signi...	131	[luliortiztv, flgovscott, offici, sign, hous, ...	[luliortiztv, flgovscott, offici, sign, hous, ...	120
16738576	rt jemillerwbal actual email sent to donald ...	202	[jemillerwb, actual, email, sent, donald, trum...	[jemillerwb, actual, email, sent, donald, trum...	176

Figure 12: Stop words removal

Column "filtered" shows removal of the default word "should" from the first row present in the column Stemmed_tokens. The count in the column "Post_tokens" is less than the count in the column "tokens" showing the stop words removal.

user_id	features
820694	(1000,[0,1,3,4,5,...
9409672	(1000,[1,25,33,75...
14731218	(1000,[2,3,4,17,1...
15606755	(1000,[13,18,26,3...
16738576	(1000,[0,1,2,6,22...
20313994	(1000,[1,2,5,9,10...
20698952	(1000,[11,14,21,6...
20977121	(1000,[0,10,17,18...
21027717	(1000,[0,1,3,4,6,...
21116401	(1000,[4,5,19,24,...

Figure 13: Vectorization

Vocabulary size to be used for vectorization is set to "1000".

The topics described by their top-weighted terms:

	topic	Words	termIndices	termWeights
0	1	[know, make, time, sorry, things]	[2, 6, 1, 137, 35]	[0.0101486246179, 0.00924271814018, 0.00866277...
1	6	[want, someone, life, never, know]	[3, 16, 8, 9, 2]	[0.0133477028352, 0.0120494366898, 0.011746551...
2	3	[teenchoice, vote, aldubgetherforever, alduber...	[28, 25, 234, 237, 44]	[0.0455820850455, 0.0380149775152, 0.013464928...
3	5	[trump, russia, nowplaying, realdonaldtrump, w...	[0, 55, 54, 95, 14]	[0.0448528580419, 0.0109153106786, 0.010848465...
4	9	[lady, gaga, veranomtv, pronounce, days]	[20, 33, 36, 261, 102]	[0.142274335337, 0.130792288937, 0.12963214745...
5	4	[porn, please, girls, playing, nude]	[120, 19, 84, 96, 402]	[0.0424305247435, 0.0267229712065, 0.024983899...
6	8	[follow, retweet, everyone, followers, video]	[4, 59, 27, 104, 13]	[0.0998756683199, 0.0332607800836, 0.031839133...
7	7	[free, size, black, nike, code]	[10, 76, 18, 144, 56]	[0.0287866853946, 0.0248494645373, 0.023744206...
8	2	[july, time, marketing, live, read]	[49, 1, 247, 48, 107]	[0.028027139928, 0.0129653136897, 0.0123693725...
9	0	[news, gurmeetramrahim, trump, photos, internet]	[30, 164, 0, 196, 291]	[0.0318997259413, 0.0246610613327, 0.013073992...

Figure 14: Topic Distribution

Column "Words" contains the five most used terms and this forms the corresponding "topic" column. Column "termIndices" shows the index of each word in the column "Words" in the same order. Column "termWeights" shows the probability of each of the words in the column "Words" appearing under the respective topic in the same order.

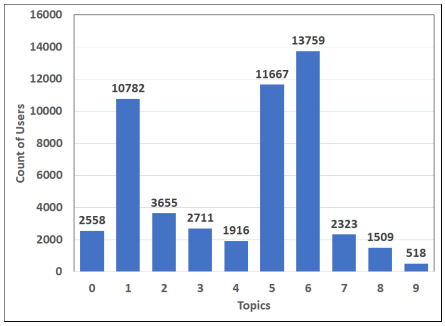


Figure 15: Topic distribution across users

The graph shows the number of users who tweeted about a particular topic. Five words are associated within a topic as shown in figure 16.

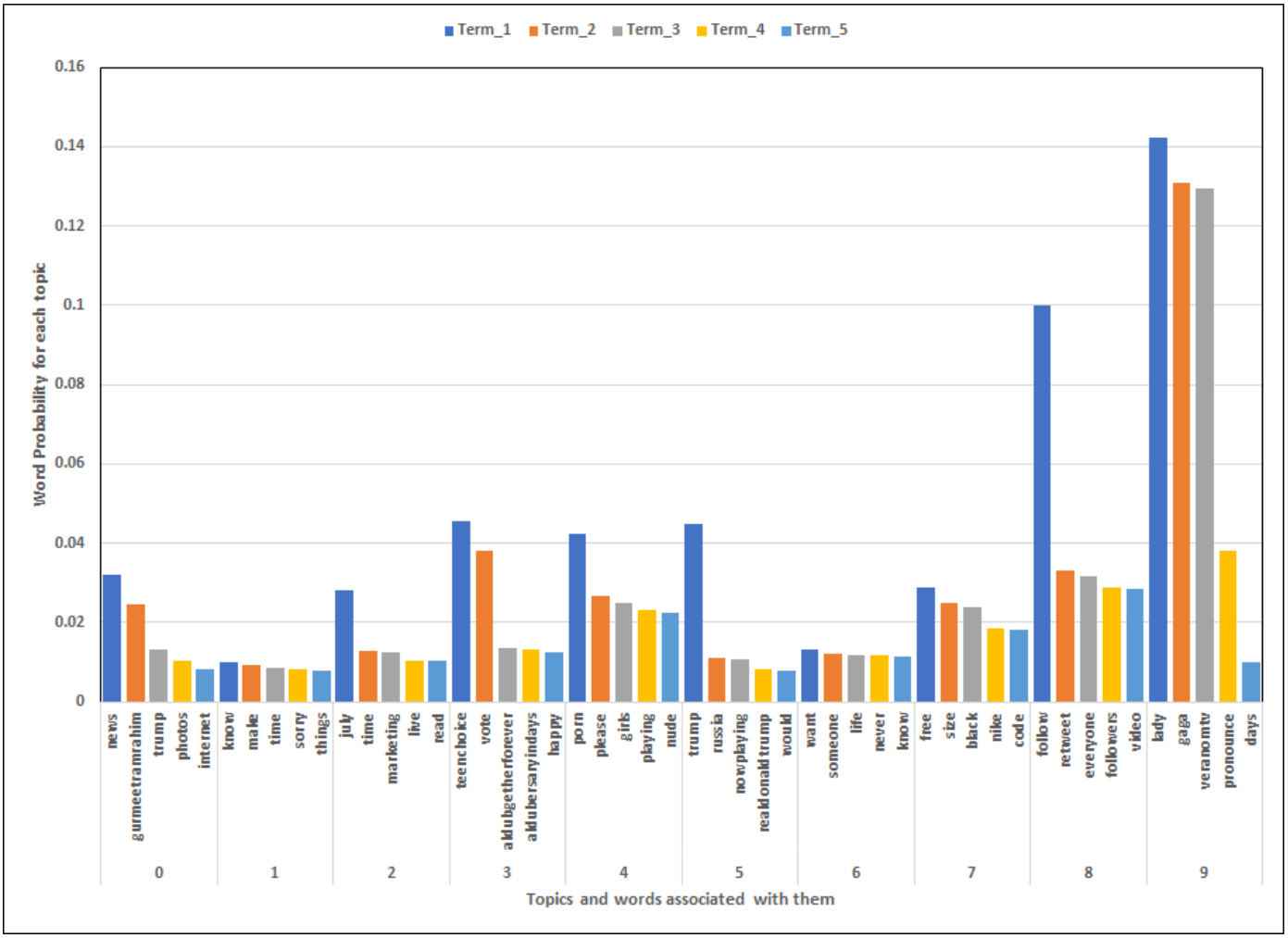
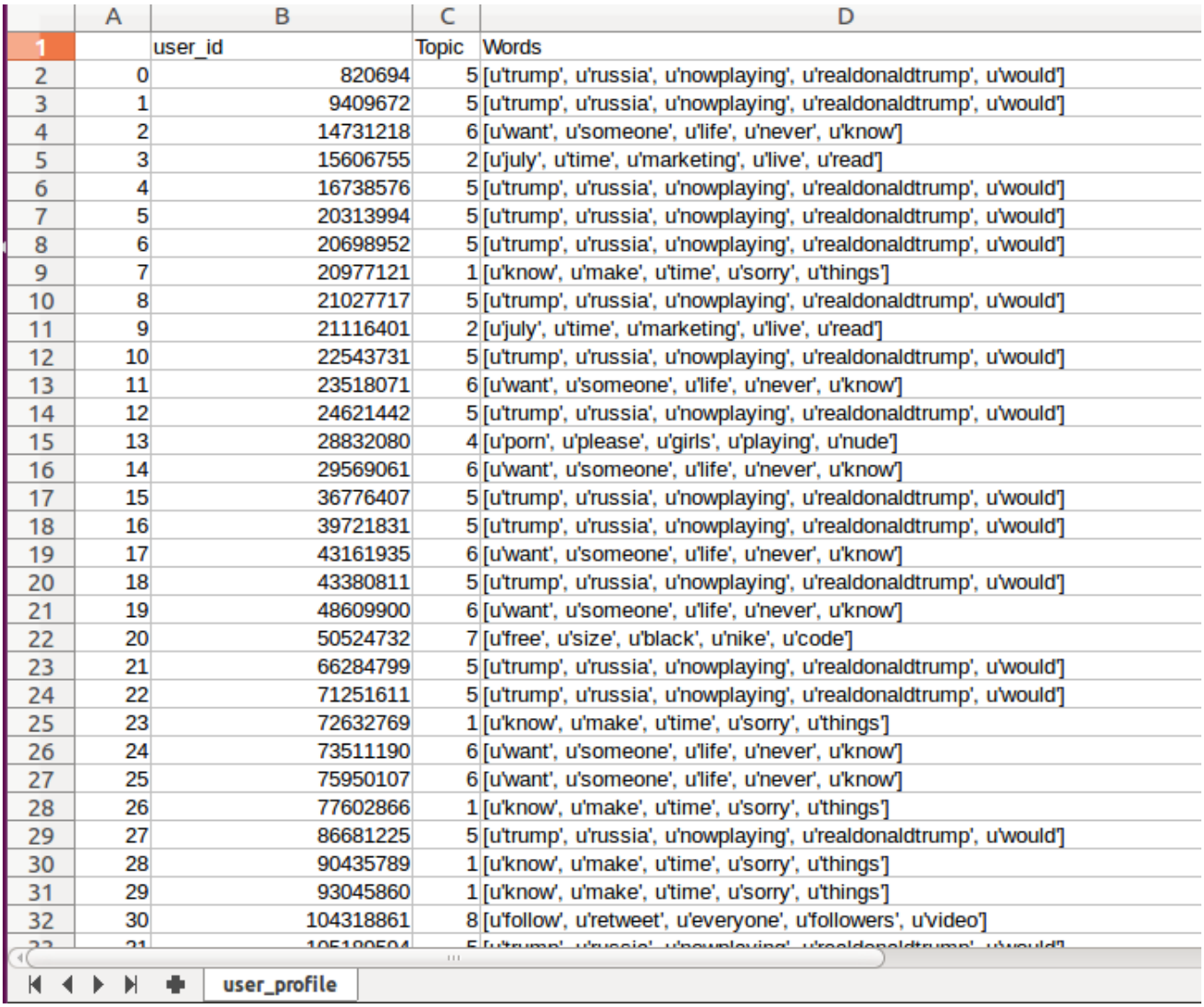


Figure 16: Topic vs Word

The graph shows the words associated under each topic and their probability of having being associated to a topic.



	A	B	C	D
1		user_id	Topic	Words
2	0	820694	5	[u'trump', u'russia', u'nowplaying', u'realdonaldtrump', u'would']
3	1	9409672	5	[u'trump', u'russia', u'nowplaying', u'realdonaldtrump', u'would']
4	2	14731218	6	[u'want', u'someone', u'life', u'never', u'know']
5	3	15606755	2	[u'july', u'time', u'marketing', u'live', u'read']
6	4	16738576	5	[u'trump', u'russia', u'nowplaying', u'realdonaldtrump', u'would']
7	5	20313994	5	[u'trump', u'russia', u'nowplaying', u'realdonaldtrump', u'would']
8	6	20698952	5	[u'trump', u'russia', u'nowplaying', u'realdonaldtrump', u'would']
9	7	20977121	1	[u'know', u'make', u'time', u'sorry', u'things']
10	8	21027717	5	[u'trump', u'russia', u'nowplaying', u'realdonaldtrump', u'would']
11	9	21116401	2	[u'july', u'time', u'marketing', u'live', u'read']
12	10	22543731	5	[u'trump', u'russia', u'nowplaying', u'realdonaldtrump', u'would']
13	11	23518071	6	[u'want', u'someone', u'life', u'never', u'know']
14	12	24621442	5	[u'trump', u'russia', u'nowplaying', u'realdonaldtrump', u'would']
15	13	28832080	4	[u'porn', u'please', u'girls', u'playing', u'nude']
16	14	29569061	6	[u'want', u'someone', u'life', u'never', u'know']
17	15	36776407	5	[u'trump', u'russia', u'nowplaying', u'realdonaldtrump', u'would']
18	16	39721831	5	[u'trump', u'russia', u'nowplaying', u'realdonaldtrump', u'would']
19	17	43161935	6	[u'want', u'someone', u'life', u'never', u'know']
20	18	43380811	5	[u'trump', u'russia', u'nowplaying', u'realdonaldtrump', u'would']
21	19	48609900	6	[u'want', u'someone', u'life', u'never', u'know']
22	20	50524732	7	[u'free', u'size', u'black', u'nike', u'code']
23	21	66284799	5	[u'trump', u'russia', u'nowplaying', u'realdonaldtrump', u'would']
24	22	71251611	5	[u'trump', u'russia', u'nowplaying', u'realdonaldtrump', u'would']
25	23	72632769	1	[u'know', u'make', u'time', u'sorry', u'things']
26	24	73511190	6	[u'want', u'someone', u'life', u'never', u'know']
27	25	75950107	6	[u'want', u'someone', u'life', u'never', u'know']
28	26	77602866	1	[u'know', u'make', u'time', u'sorry', u'things']
29	27	86681225	5	[u'trump', u'russia', u'nowplaying', u'realdonaldtrump', u'would']
30	28	90435789	1	[u'know', u'make', u'time', u'sorry', u'things']
31	29	93045860	1	[u'know', u'make', u'time', u'sorry', u'things']
32	30	104318861	8	[u'follow', u'retweet', u'everyone', u'followers', u'video']

Figure 17: User Profile generated

Generated user profiles are exported to a csv file. This profile shows a unique user id u_i associated with every twitter user. It shows the topic id t_i most frequently twitted by the user id u_i and the words associated under the topic id t_i

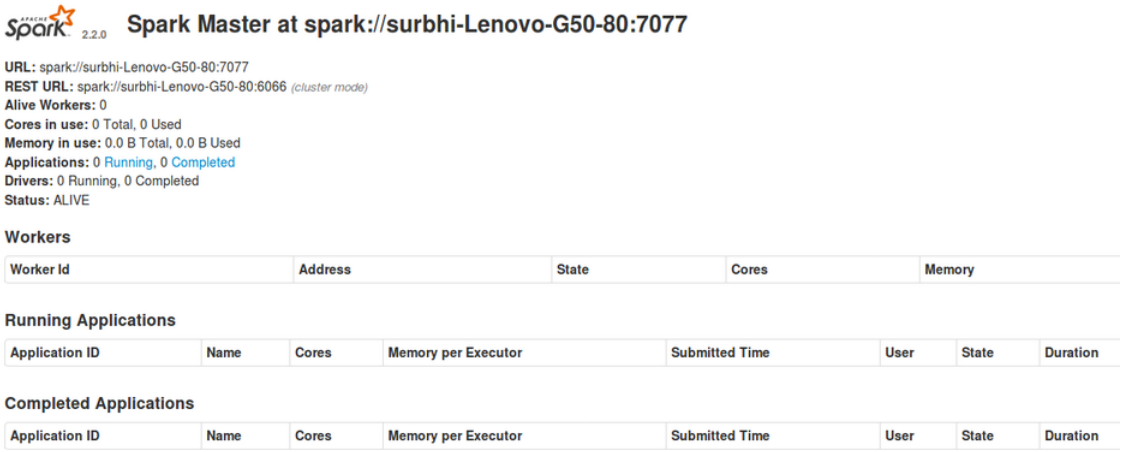


Figure 18: Launch and verify master node on web UI

Launch the master node using "start-master.sh" under SPARK_HOME/sbin folder. Verify the same on the web UI. Default port number is 7077.

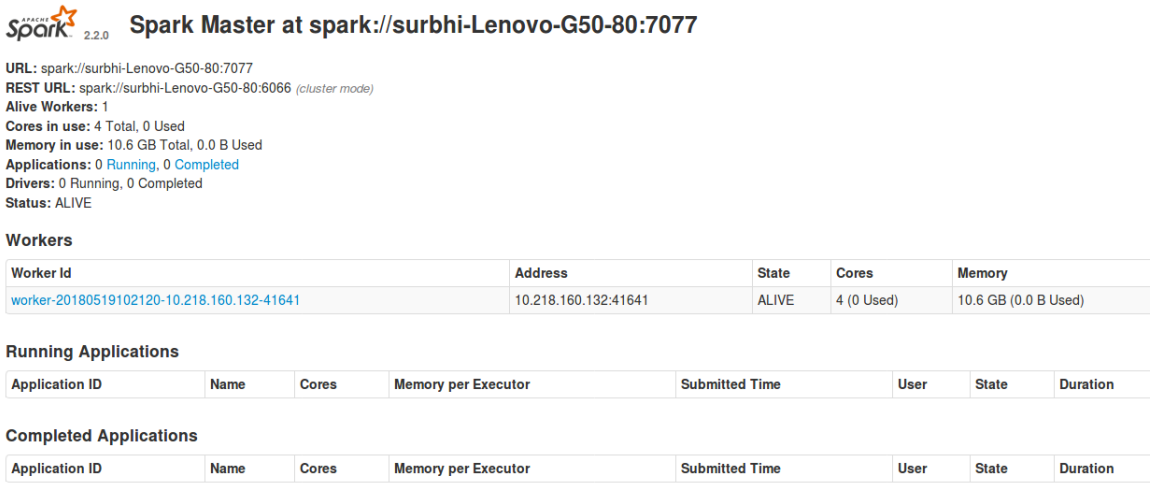


Figure 19: Launch and verify worker node on web UI

Launch the slave using "start-slaves.sh" under SPARK_HOME/sbin folder. Verify the same on the web UI. By default, only one executor of slave node is launched.

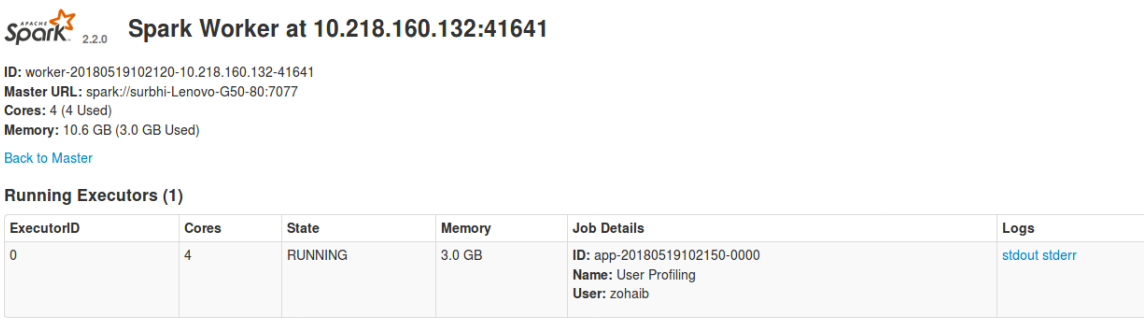


Figure 20: Details of worker node

Submit a job using "spark-submit" command under SPARK_HOME/bin folder, worker node state changes to "RUNNING".

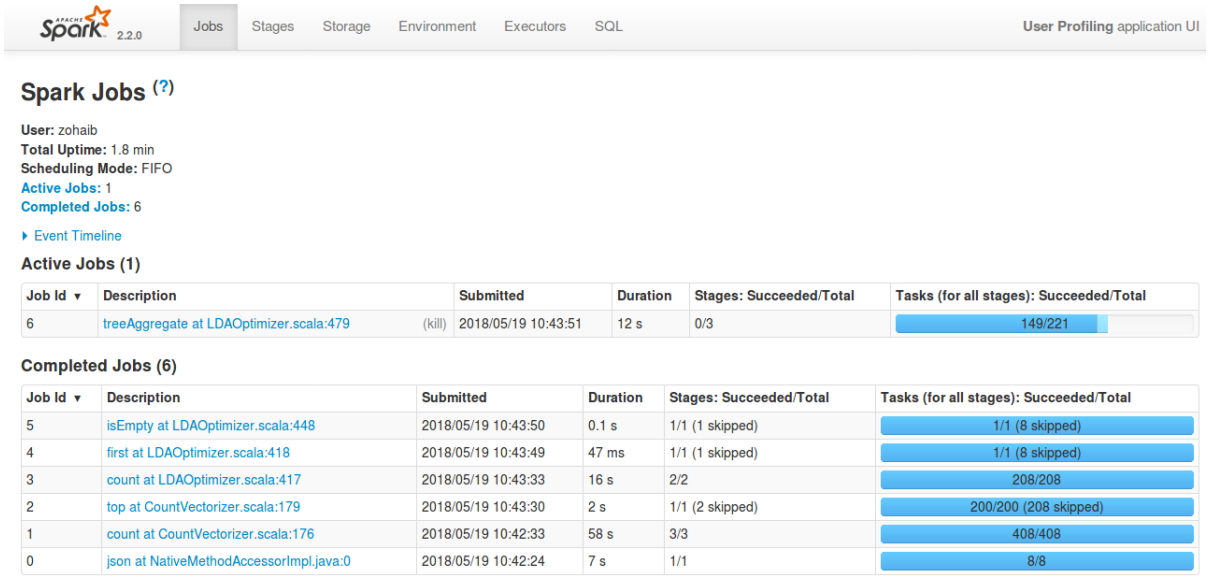


Figure 21: Spark Jobs

The "Jobs" tab on the Spark Standalone cluster web UI displays the details of the various tasks that has been completed successfully and the tasks that are currently active and running.

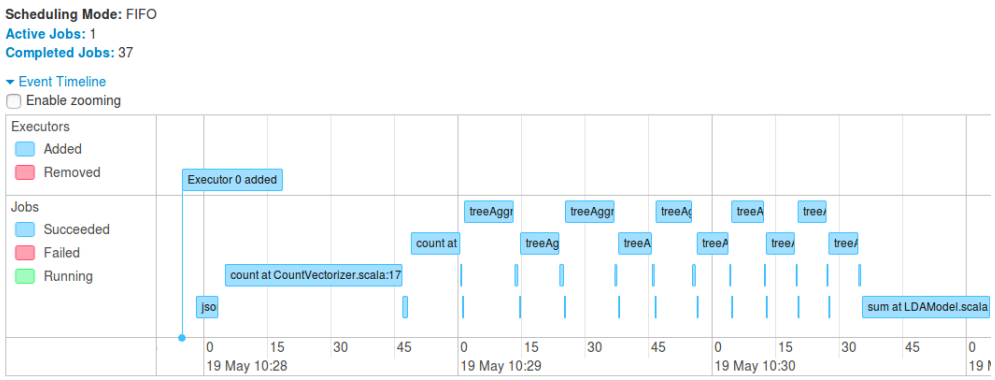


Figure 22: Event Timeline

"Event Timeline" is another feature provided by the Spark Standalone web UI where it shows the graph of various tasks and the time taken for each task. Also, the graph shows LDA algorithm performing ten iterations. "10" was set as the value of the parameter "maxIter" while defining the LDA model.

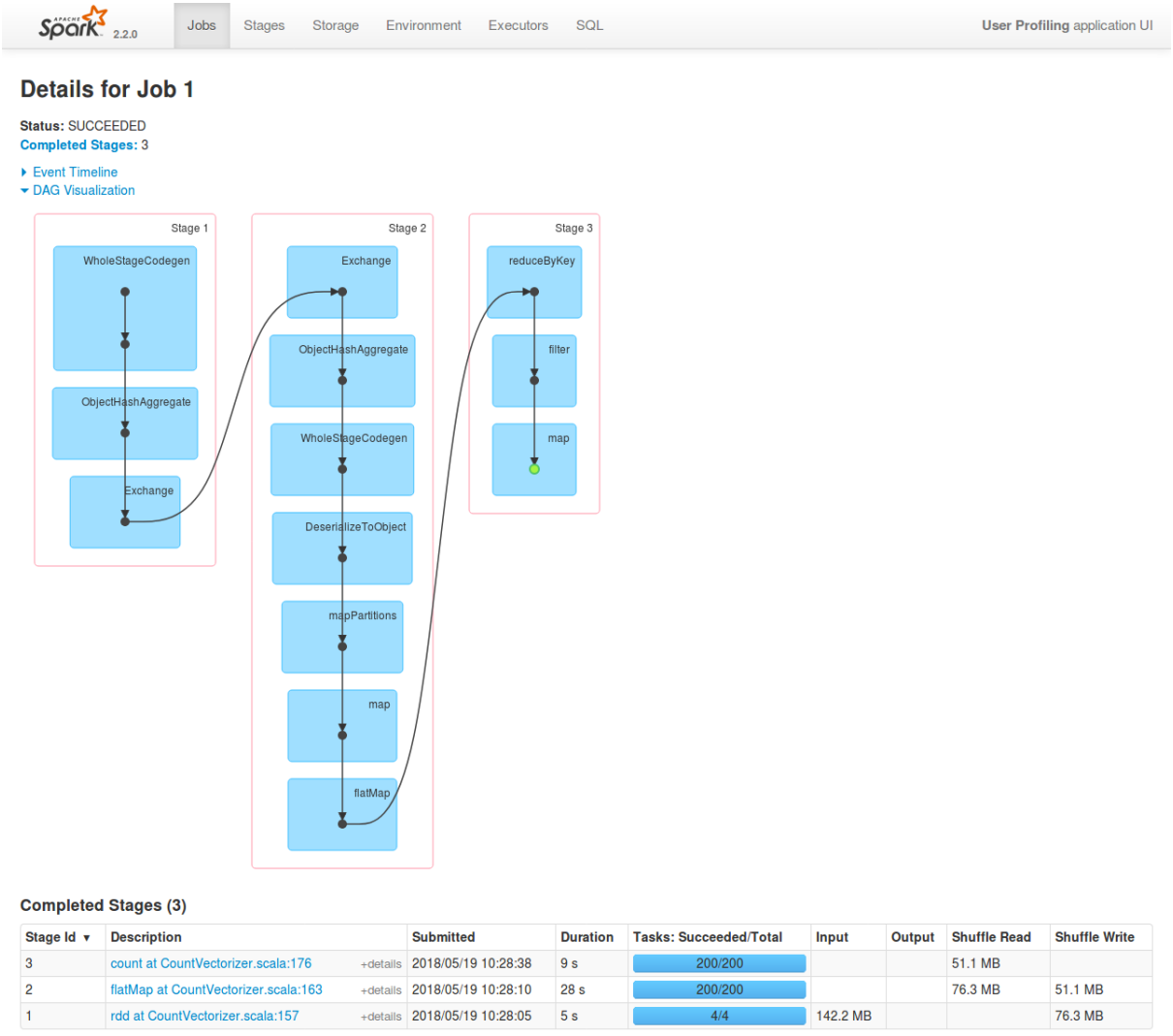


Figure 23: DAG Visualization for first three stages

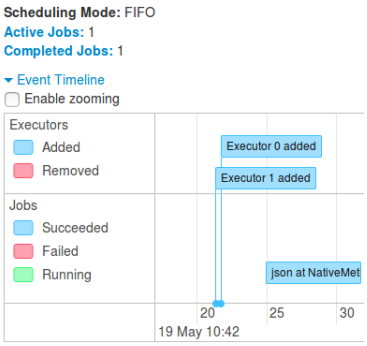
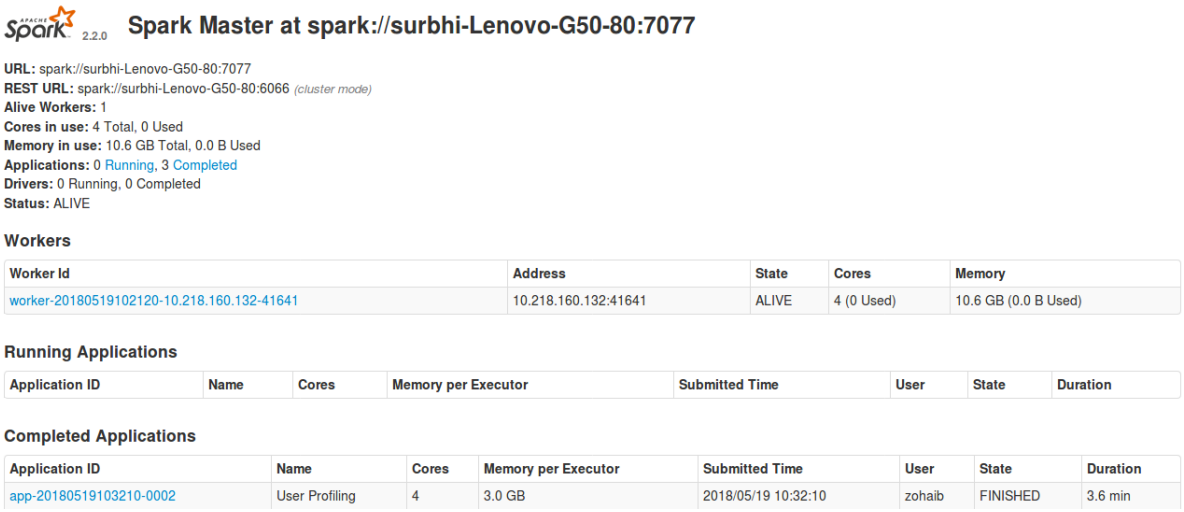


Figure 24: Two executors launched under one master



Spark Master at spark://surbhi-Lenovo-G50-80:7077

URL: spark://surbhi-Lenovo-G50-80:7077
REST URL: spark://surbhi-Lenovo-G50-80:6066 *(cluster mode)*
Alive Workers: 1
Cores in use: 4 Total, 0 Used
Memory in use: 10.6 GB Total, 0.0 B Used
Applications: 0 Running, 3 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers

Worker Id	Address	State	Cores	Memory
worker-20180519102120-10.218.160.132-41641	10.218.160.132:41641	ALIVE	4 (0 Used)	10.6 GB (0.0 B Used)

Running Applications

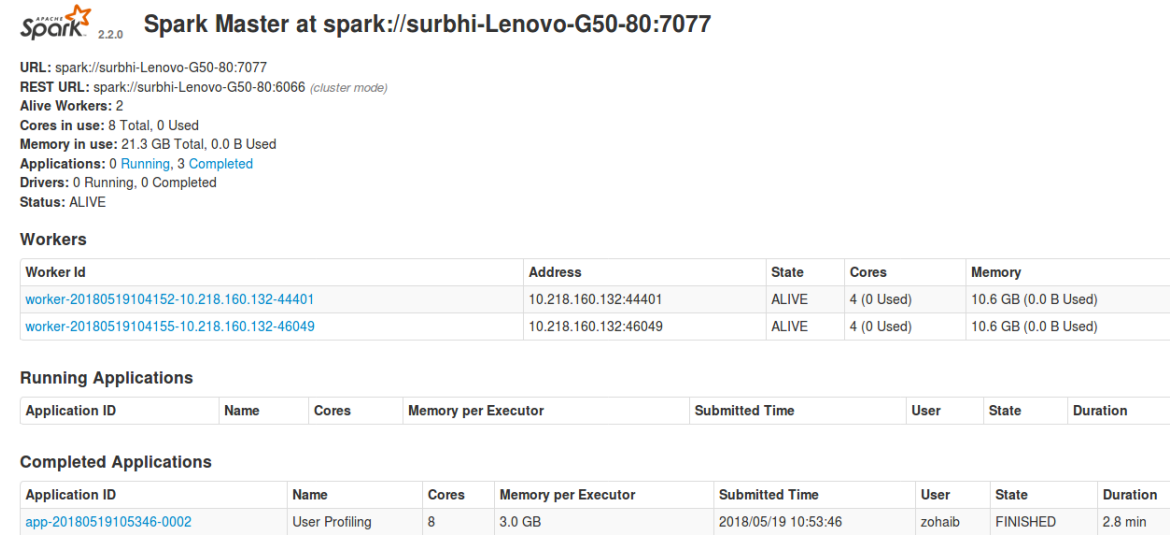
Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration

Completed Applications

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration
app-20180519103210-0002	User Profiling	4	3.0 GB	2018/05/19 10:32:10	zohaib	FINISHED	3.6 min

Figure 25: One executor under one master

One executor was launched and could be seen in the state "ALIVE". The job was submitted to one executor and it completed the execution in 3.6 minutes for 1.13M rows of dataset and LDA iteration as "10".



Spark Master at spark://surbhi-Lenovo-G50-80:7077

URL: spark://surbhi-Lenovo-G50-80:7077
REST URL: spark://surbhi-Lenovo-G50-80:6066 *(cluster mode)*
Alive Workers: 2
Cores in use: 8 Total, 0 Used
Memory in use: 21.3 GB Total, 0.0 B Used
Applications: 0 Running, 3 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers

Worker Id	Address	State	Cores	Memory
worker-20180519104152-10.218.160.132-44401	10.218.160.132:44401	ALIVE	4 (0 Used)	10.6 GB (0.0 B Used)
worker-20180519104155-10.218.160.132-46049	10.218.160.132:46049	ALIVE	4 (0 Used)	10.6 GB (0.0 B Used)

Running Applications

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration

Completed Applications

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration
app-20180519105346-0002	User Profiling	8	3.0 GB	2018/05/19 10:53:46	zohaib	FINISHED	2.8 min

Figure 26: Two executors under one master

Two executors were launched and could be seen in the state "ALIVE". Keeping all the configurations identical as before, this time the job was submitted to two executors instead of one executor and it completed faster, in 2.8 minutes.

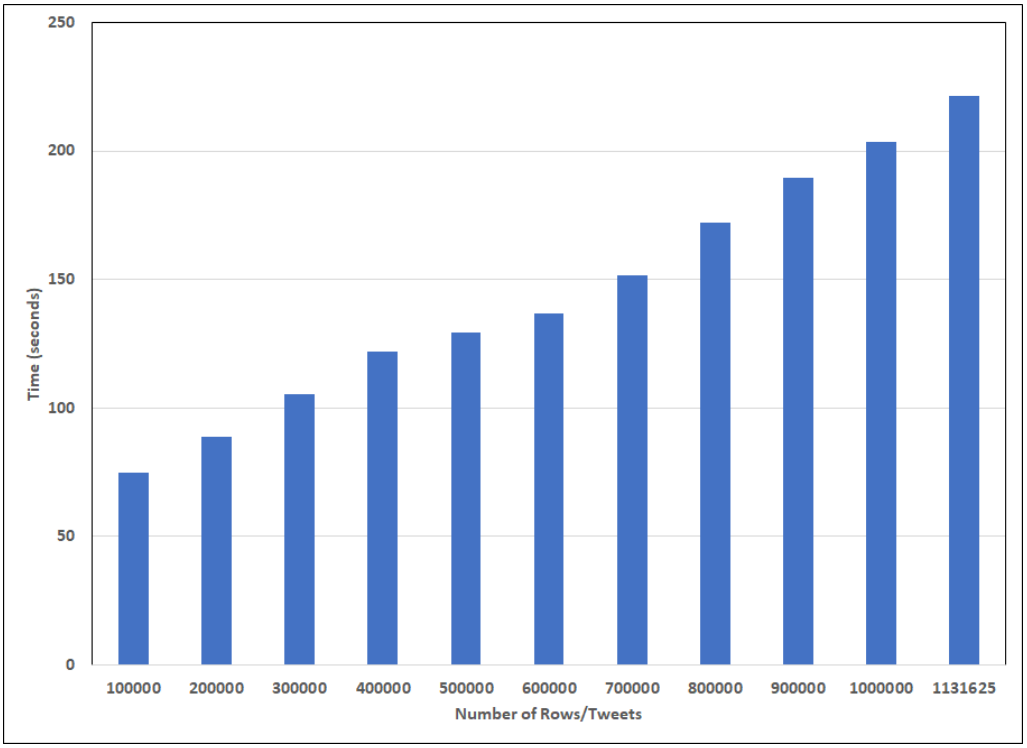


Figure 27: Benchmark

The graph shows the incremental behaviour of the time consumed to execute the task while varying the number of tweets in increasing order.

and vice-versa. We couldn't succeed with this. What worked for us was the master and worker nodes both based on 64-bit Linux operating system. What we learned from this was how the modification in the configurations and properties of the master and worker nodes bring about differences in the time taken to execute the same piece of task.

6. ACKNOWLEDGMENTS

The authors would like to thank Daniele Foroni, Phd University of Trento, for guidance in resolving issues faced during the set-up of distributed environment for this project.

7. REFERENCES

[1] S. Penchikala. (2015) Big data processing with apache spark-part 1: Introduction. [Online]. Available: https://www.infoq.com/articles/apache-spark-introduction

[2] P. Y. Velegrakis. (2017) Big data and social networks.

[3] P. D. A. Iosup. (2017) Principles of distributed systems, cloud computing, and big data processing. [Online]. Available: http://www.ds.ewi.tudelft.nl/~iosup/Presentations/2017/2017-02-13_DistrSysCloudsBigData17course-tud_format-16-9.pdf

[4] J. Dianes. (2015) Spark & python: Sql & dataframes. [Online]. Available: https://www.codementor.io/jadianes/python-spark-sql-dataframes-du107w74i

[5] H. Gupta. (2018) Apache spark: 3 reasons why you should not use rdds. [Online]. Available: https://dzone.com/articles/apache-spark-3-reasons-why-you-should-not-use-rdds

[6] A. Gupta. (2016) Complete guide on dataframe operations in pyspark. [Online]. Available: https://www.analyticsvidhya.com/blog/2016/10/spark-dataframe-and-operations/

[7] A. Spark. Datasets and dataframes. [Online]. Available: https://spark.apache.org/docs/2.1.0/sql-programming-guide.html#sql

[8] K. Willems. (2017) Spark apis: Rdd, dataset and dataframe. [Online]. Available: https://www.datacamp.com/community/tutorials/apache-spark-python#difference

[9] S. Neumann. (2014) Spark vs. hadoop mapreduce. [Online]. Available: https://www.xplenty.com/blog/apache-spark-vs-hadoop-mapreduce/

[10] R. Xin, M. Armburst, and D. Liu. (2015) Introducing dataframes in apache spark for large scale data science. [Online]. Available: https://databricks.com/blog/2015/02/17/introducing-dataframes-in-spark-for-large-scale-data-science.html

[11] B. Mathew. (2016) Spark rdds vs dataframes vs sparksql. [Online]. Available: https://community.hortonworks.com/articles/42027/rdd-vs-dataframe-vs-sparksql.html

[12] B. Klein. Lambda, filter, reduce and map. [Online]. Available: https://www.python-course.eu/lambda.php

[13] P. Tips. 4. map, filter and reduce. [Online]. Available: http://book.pythontips.com/en/latest/map_filter.html

[14] S. Srivastava. (2016) Spark lda: A complete example

of clustering algorithm for topic discovery. [Online]. Available: hhttps://dzone.com/articles/spark-lda-a-complete-example-of-clustering-algorit

[15] Algobeans.com. (2015) Topic modeling with lda introduction. [Online]. Available: https://algobeans.com/2015/06/21/laymans-explanation-of-topic-modeling-with-lda-2/

[16] K. Unyelioglu. (2016) Data clustering using apache spark. [Online]. Available: https://dzone.com/articles/cluster-analysis-using-apache-spark-exploring-colo

[17] D. Lynn. (2016) Apache spark cluster managers: Yarn, mesos, or standalone? [Online]. Available: http://www.agildata.com/apache-spark-cluster-managers-yarn-mesos-or-standalone/

[18] J. Pihony. (2015) Which cluster type should i choose for spark? [Online]. Available: https://stackoverflow.com/questions/28664834/which-cluster-type-should-i-choose-for-spark

[19] A. Spark. Tokenizer. [Online]. Available: https://spark.apache.org/docs/2.1.0/ml-features.html#tokenizer

[20] Stanford.edu. (2009) Stemming and lemmatization. [Online]. Available: https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html

[21] Apache.Spark. Extracting, transforming and selecting features. [Online]. Available: https://spark.apache.org/docs/2.2.0/ml-features.html#stopwordsremover

[22] ApacheSpark. Countvectorizer. [Online]. Available: https://spark.apache.org/docs/2.2.0/ml-features.html#countvectorizer

[23] D. M. Blei, A. Y. Ng, and M. I. Jordan. (2003) Latent dirichlet allocation. [Online]. Available: http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf

[24] Brian. (2011) My adventures in coding. [Online]. Available: https://myadventuresincoding.wordpress.com/2011/12/22/linux-how-to-ssh-between-two-linux-computers-without-needing-a-password/

[25] Clearos.com. Setting up ssh trust between two servers. [Online]. Available: https://www.clearos.com/resources/documentation/clearos/content:en_us:kb_o_setting_up_ssh_trust_between_two_servers#authorizing_the_server_to_trust_itself

[26] Cmidi. (2015) What is actually in knownhosts? [Online]. Available: https://stackoverflow.com/questions/33243393/what-is-actually-in-known-hosts

[27] Gilles. (2012) What is the difference between authorizedkeys and knownhosts file for ssh? [Online]. Available: https://security.stackexchange.com/questions/20706/what-is-the-difference-between-authorized-keys-and-known-hosts-file-for-ssh

[28] K. Singh. (2015) How to setup apache spark standalone cluster on multiple machine. [Online]. Available: http://paxcel.net/blog/how-to-setup-apache-spark-standalone-cluster-on-multiple-machine/

[29] J. Laskowski. (2016) Example 2-workers-on-1-node standalone cluster (one executor per worker). [Online]. Available: https://github.com/jaceklaskowski/mastering-apache-spark-book/blob/master/spark-standalone-example-2-workers-on-1-node-cluster.adoc

[30] D. Canones. (2017) How to configure an apache spark standalone cluster and integrate with jupyter: Step-by-step. [Online]. Available: https://www.davidadrian.cc/posts/2017/08/how-to-spark-cluster/

[31] J. Laskowski. (2016) Spark standalone cluster. [Online]. Available: https://jaceklaskowski.gitbooks.io/mastering-apache-spark/content/spark-standalone.html

[32] E. Guide. How big data and distributed systems solve traditional scalability problems. [Online]. Available: https://www.theserverside.com