



Bharati Vidyapeeth College of Engineering, Navi Mumbai

Department of Information Technology

2021-2022

Facebook Data Analysis with Spark using Pyspark

Submitted in partial fulfillment of the requirements of the degree of

Bachelor of Engineering

By

Rushikesh Mahadik (40)

Omkar Pawar (56)

Bhagyashree Rajguru (59)

Guide:

Prof. V.N.Patil

Mini Project Report Approval for B.E. SEM VIII

This Mini Project Report entitled **Facebook Data Analysis with Spark using Pyspark** by **Rushikesh Mahadik , Omkar Pawar and Bhagyashree Rajguru** is approved for the BDA Lab for the academic year **2021-2022**.

Date: _____

Prof. V.N. Patil
Project Guide

Prof. H .B. Sale
Head of Department

TABLE OF CONTENTS

• ABSTRACT	4
• INTRODUCTION	5
• PROBLEM STATEMENT	6
• RESEARCH METHODOLOGY	7-8
• IMPLEMENTATION	9-12
• RESULT AND CONCLUSION	13

ABSTRACT

The concept of big data has been around for years; most organizations now understand that if they capture all the data that streams into their businesses, they can apply analytics and get significant value from it. But even in the 1950s, decades before anyone uttered the term “big data,” businesses were using basic analytics (essentially numbers in a spreadsheet that were manually examined) to uncover insights and trends. The new benefits that big data analytics brings to the table, however, are speed and efficiency. Whereas a few years ago a business would have gathered information, run analytics and unearthed information that could be used for future decisions, today that business can identify insights for immediate decisions. The ability to work faster – and stay agile – gives organizations a competitive edge they didn’t have before. Through our project we intend to carry out analysis on a preferably large dataset. So we have chosen the dataset obtained from several Facebook users. By carrying out certain operations, we intend to harness their data and use it to identify new opportunities.

INTRODUCTION

A Social network is defined as a network of relationships or interactions, where the nodes consist of people or actor, and the edges or arcs consist of the relationships or interactions between these actors.

Social networks and the techniques to analyse them existed since decades. There can be several type of social networks like email network, telephone network, collaboration network.

But recently online social networks like Facebook, Twitter, LinkedIn, MySpace etc have been developed which gained popularity within very short amount of time and gathered large number of users.

Facebook is said to have more than 2.912 billion users in 2022. The field of social networks and their analysis has evolved from graph theory, statistics and sociology and it is used in several other fields like information science, business application, communication, economy etc.

Analysing a social network is similar to the analysis of a graph because social networks form the topology of a graph. Graph analysis tools have been there for decades. But they are not designed for analysing a social network graph which has complex properties. An online social network graph may be very large. It may contain millions of nodes and edges. Social networks are dynamic i.e. there is continuous evolution and expansion. A node in social network usually has several attributes. There are small and large communities within the social graph. Old graph analysis tools are not designed to manage such large and complex social network graph.

Facebook is a preferred social network by marketers, not only because of the sheer number of users represented but also because of its incredibly insightful analytics suite. It's important to be able to analyze customers and their behavior on a micro level due to Facebook's ever-changing algorithm, and the implications for our content and business. If we refuse to adapt our approach based on these insights, we're doomed to obscurity on the news feed.

A deep Facebook data analysis shouldn't be a one and done situation. Ideally, we'll be auditing our efforts every few months or so at most. This will help us predict the likings, and the general summary of many users as a whole.

PROBLEM STATEMENT

Data mining is sensitive to quality of input data that may be inaccurate having missing information (noise, redundant data). Mapping real data to data mining attributes could be challenging in its own ways. Big data is extensively used to transform large unstructured or structured raw data into crucial and meaningful information which helps in forming a healthy decision support system for business poses the biggest concern of handling, maintaining and analysing huge amount of data generated every day within a minimal time span.

RESEARCH METHODOLOGY

Apache Spark:

Apache Spark is a lightning fast cluster computing system. It provides the set of high-level API namely Java, Scala, Python, and R for application development. Apache Spark is a tool for speedily executing Spark Applications. Spark utilizes Hadoop in two different ways – one is for Storage and second is for Process handling. Just because Spark has its own Cluster Management, so it utilizes Hadoop for Storage objective.

Spark is intended to cover an extensive variety of remaining loads, for example, cluster applications, iterative calculations, intuitive questions, and streaming. Aside from supporting all these remaining tasks at hand in a particular framework, it decreases the administration weight of keeping up isolated apparatuses.

Spark is one of Hadoop's sub venture created in 2009 in UC Berkeley's AMPLab by Matei Zaharia. It was Open Sourced in 2010 under a BSD license. It was given to Apache programming establishment in 2013, and now Apache Spark has turned into the best level Apache venture from Feb-2014. And now the results are pretty booming.

Market rules and big agencies already tend to use Spark for their solutions. Flabbergast to know that the list includes - Netflix, Uber, Pinterest, Conviva, Yahoo, Alibaba, eBay, MyFitnessPal, OpenTable, TripAdvisor and much more. It can be said that Apache Spark Use Case stretch is spread right from Finance, Healthcare, Travel, e-commerce to Media & Entertainment industry.

Spark is not fit for a multi-user environment. Spark as of now is not capable of handling more users concurrency, maybe in future updates this issue will be overcome. Yet an alternate engine like Hive for handling large batch projects.

Spark is accessible, intense, powerful and proficient Big Data tool for handling different enormous information challenges. Apache Spark takes after an ace/slave engineering with two primary Daemons and a Cluster Manager –

- Master Daemon – (Master/Driver Process)
- Worker Daemon – (Slave Process)

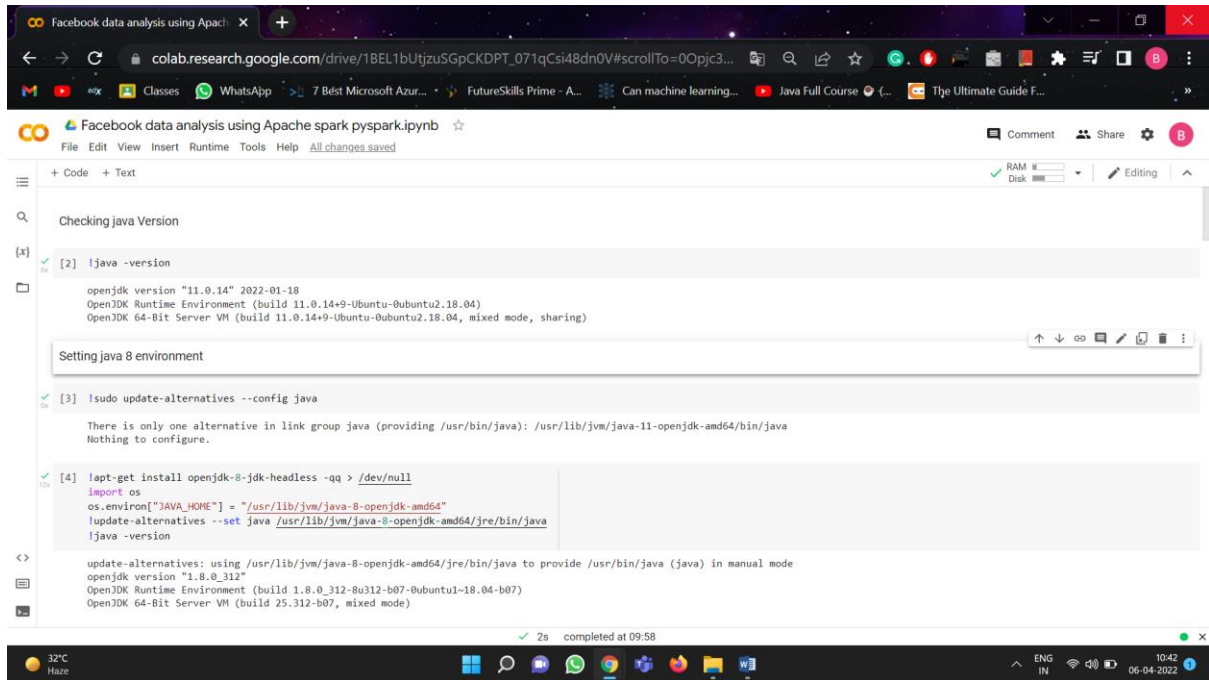
Most advanced and popular product of Apache Community, Spark decreases the time complexity of the system. Fast Computations, increase Performance, structured and unstructured Data Streaming, Graph Analytics, richer Resource Scheduling capabilities ensures smooth, engaging and customer experience compatible with the system.

Pyspark

PySpark is a great language for performing exploratory data analysis at scale, building machine learning pipelines, and creating ETLs for a data platform. If you're already familiar with Python and libraries such as Pandas, then PySpark is a great language to learn in order to create more scalable analyses and pipelines. The goal of this post is to show how to get up and running with PySpark and to perform common tasks.

IMPLEMENTATION

1. Setting up the environment for java 8



```
Checking java Version

[2] !java -version

openjdk version "11.0.14" 2022-01-18
OpenJDK Runtime Environment (build 11.0.14+9-Ubuntu-0ubuntu2.18.04)
OpenJDK 64-Bit Server VM (build 11.0.14+9-Ubuntu-0ubuntu2.18.04, mixed mode, sharing)

Setting java 8 environment

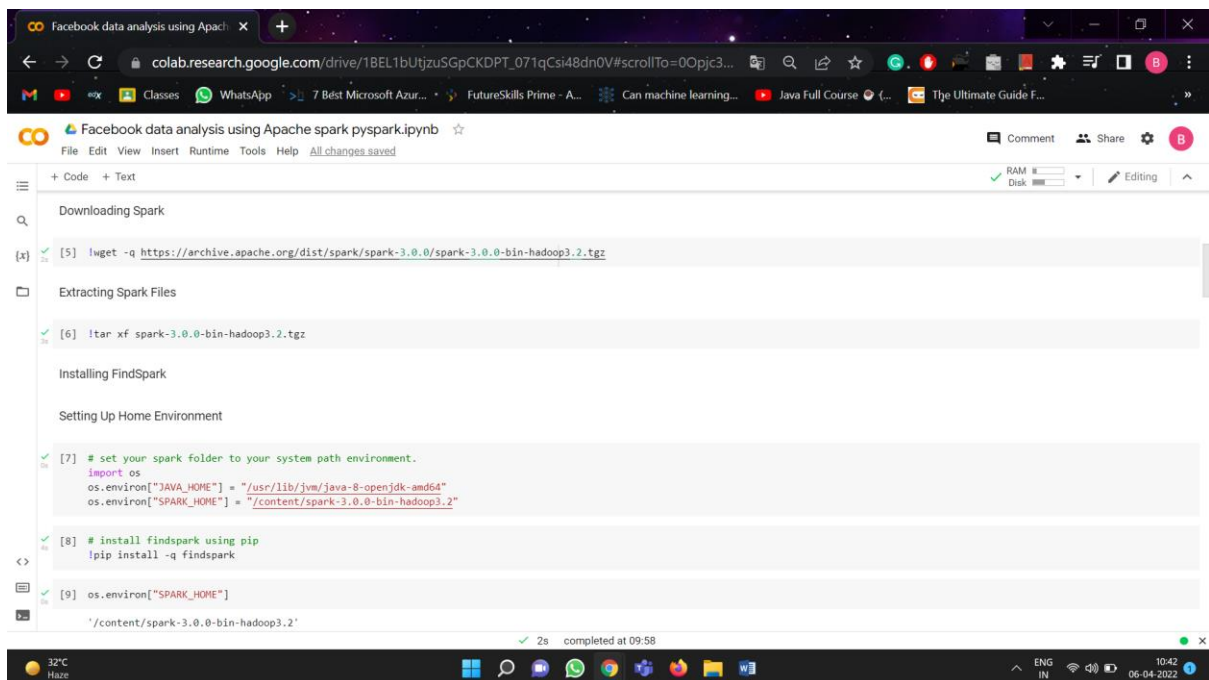
[3] !sudo update-alternatives --config java

There is only one alternative in link group java (providing /usr/bin/java): /usr/lib/jvm/java-11-openjdk-amd64/bin/java
Nothing to configure.

[4] !apt-get install openjdk-8-jdk-headless -qq > /dev/null
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
!update-alternatives --set java /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
!java -version

update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java to provide /usr/bin/java (java) in manual mode
openjdk version "1.8.0_312"
OpenJDK Runtime Environment (build 1.8.0_312-b07-0ubuntu1-18.04-b07)
OpenJDK 64-Bit Server VM (build 25.312-b07, mixed mode)
```

2. Downloading Spark



```
Downloading Spark

[5] !wget -q https://archive.apache.org/dist/spark/spark-3.0.0/spark-3.0.0-bin-hadoop3.2.tgz

Extracting Spark Files

[6] !tar xf spark-3.0.0-bin-hadoop3.2.tgz

Installing FindSpark

Setting Up Home Environment

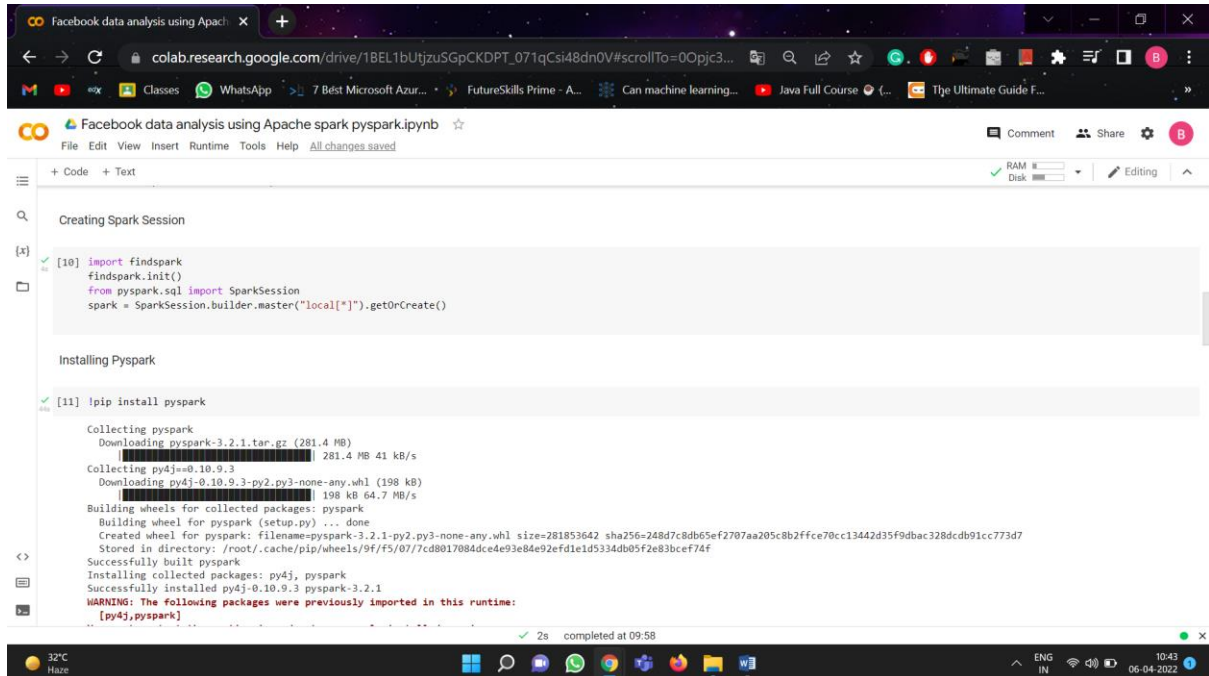
[7] # set your spark folder to your system path environment.
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.0.0-bin-hadoop3.2"

[8] # install findspark using pip
!pip install -q findspark

[9] os.environ["SPARK_HOME"]

"/content/spark-3.0.0-bin-hadoop3.2"
```

3. Installing pyspark



The screenshot shows a Google Colab notebook titled "Facebook data analysis using Apache spark pyspark.ipynb". The notebook is in "Code" view. The first cell, labeled "[10]", contains the following code to create a Spark session:

```
import findspark
findspark.init()
from pyspark.sql import SparkSession
spark = SparkSession.builder.master("local[*]").getOrCreate()
```

The second cell, labeled "[11]", contains the command to install pyspark:

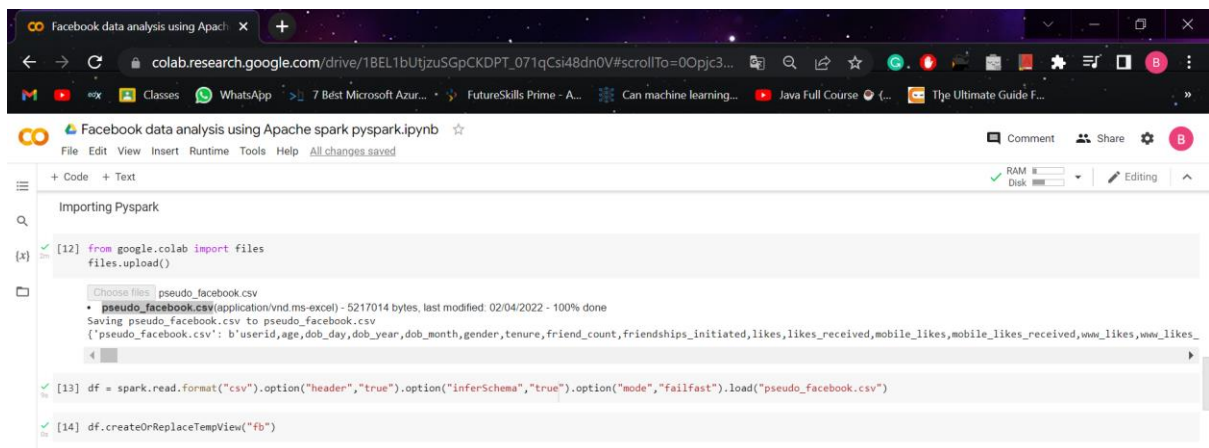
```
!pip install pyspark
```

The output of the second cell shows the installation process:

```
Collecting pyspark
  Downloading pyspark-3.2.1.tar.gz (281.4 MB)
    281.4 MB 41 kB/s
Collecting py4j==0.10.9.3
  Downloading py4j-0.10.9.3-py2.py3-none-any.whl (198 kB)
    198 kB 64.7 MB/s
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.2.1-py2.py3-none-any.whl size=281853642 sha256=248d7c8db65ef2707aa205cbb2ffce70cc13442d35f9dbac328dcdb91cc773d7
  Stored in directory: /root/.cache/pip/wheels/9f/f5/07/7cd017084dce93e84e92efd1e1d5334db05f2e83bcef74f
Successfully built pyspark
Installing collected packages: py4j, pyspark
Successfully installed py4j-0.10.9.3 pyspark-3.2.1
WARNING: The following packages were previously imported in this runtime:
[py4j, pyspark]
```

The bottom of the screenshot shows the Windows taskbar with the system clock at 10:43 on 06-04-2022.

4. Importing dataset from computer.



The screenshot shows the same Google Colab notebook. The third cell, labeled "[12]", shows the process of uploading a file from the local machine:

```
from google.colab import files
files.upload()
```

The output of the third cell shows a file named "pseudo_facebook.csv" being uploaded. The file size is 5217014 bytes, and it was last modified on 02/04/2022. The file is saved as "pseudo_facebook.csv" in the Colab environment.

The fourth cell, labeled "[13]", contains the code to read the CSV file into a DataFrame:

```
df = spark.read.format("csv").option("header", "true").option("inferSchema", "true").option("mode", "failfast").load("pseudo_facebook.csv")
```

The fifth cell, labeled "[14]", contains the code to create or replace a temporary view:

```
df.createOrReplaceTempView("fb")
```

5. Selecting all data from dataset.

```
[15] spark.sql("select * from fb").show()
```

userid	age	dob_day	dob_year	dob_month	gender	tenure	friend_count	friendships_initiated	likes	likes_received	mobile_likes	mobile_likes_received	www_likes	www_likes_received
2094382	14	19	1999	11	male	266	0	0	0	0	0	0	0	0
1192601	14	2	1999	11	female	6	0	0	0	0	0	0	0	0
2083884	14	16	1999	11	male	13	0	0	0	0	0	0	0	0
1203168	14	25	1999	12	female	93	0	0	0	0	0	0	0	0
1733186	14	4	1999	12	male	82	0	0	0	0	0	0	0	0
1524765	14	1	1999	12	male	15	0	0	0	0	0	0	0	0
1136133	13	14	2000	1	male	12	0	0	0	0	0	0	0	0
1680361	13	4	2000	1	female	0	0	0	0	0	0	0	0	0
1365174	13	1	2000	1	male	81	0	0	0	0	0	0	0	0
1712567	13	2	2000	2	male	171	0	0	0	0	0	0	0	0
1612453	13	22	2000	2	male	98	0	0	0	0	0	0	0	0
2104073	13	1	2000	2	male	55	0	0	0	0	0	0	0	0
1918584	13	5	2000	3	male	106	0	0	0	0	0	0	0	0
1704433	13	21	2000	3	male	61	0	0	0	0	0	0	0	0
1932519	13	28	2000	3	female	0	0	0	0	0	0	0	0	0
1751722	13	7	2000	4	female	16	0	0	0	0	0	0	0	0
1470850	13	30	2000	5	female	34	0	0	0	0	0	0	0	0
1001768	13	23	2000	5	female	25	0	0	0	0	0	0	0	0
1537661	13	16	2000	5	female	4	0	0	0	0	0	0	0	0
1020296	13	13	2000	8	male	9	0	0	0	0	0	0	0	0

only showing top 20 rows

6. Query to display total number of users in dataset.

```
[16] spark.sql("select count(*) from fb").show()
```

count(1)
99003

7. Query to select average age of users in dataset

```
[17] spark.sql("select avg(age) from fb").show()
```

```
+-----+
|      avg(age) |
+-----+
| 37.28022383160106 |
+-----+
```

8. Query to select average age of users grouping by gender

```
[18] spark.sql("select avg(age), gender from fb group by gender").show()
```

```
+-----+-----+
|      avg(age) | gender |
+-----+-----+
| 74.77714285714286 |    NA |
| 39.459904605753465 | female |
| 35.67024618431386 |  male |
+-----+-----+
```

9. Query to get average like recieved to each gender

```
spark.sql("select avg(likes_received) as avg_likes , gender from fb group by gender order by avg_likes").show()
```

```
+-----+-----+
| avg_likes|gender|
+-----+-----+
| 67.91154778570697| male|
|157.38285714285715|  NA|
| 251.4354349878273|female|
+-----+-----+
```

10. Query to get data of number of friends user has age between 13 to 35

```
[21] spark.sql("select avg(friend_count) from fb where age>=13 AND age<=35").show()
```

```
+-----+
| avg(friend_count)|
+-----+
|223.54579601745235|
+-----+
```

11. Query to get average like sfrom mobile and website

```
[22] spark.sql("select avg(mobile_likes), avg(www_likes) from fb where age>=13 AND age<=25").show()
```

```
+-----+-----+
| avg(mobile_likes)| avg(www_likes)|
+-----+-----+
|123.98981737425284|55.50010631511801|
+-----+-----+
```

12. Query to select average of nubmer of freidns for each age group

```
df.groupBy("age").avg("friend_count").orderBy("age", ascending = True).show()
```

```
+---+-----+
|age| avg(friend_count)|
+---+-----+
| 13|          164.75|
| 14|251.39012987012987|
| 15|347.69213139801377|
| 16|351.93713545042124|
| 17|350.30063965884864|
| 18| 331.1662817551963|
| 19|333.69209747210203|
| 20|283.49907137171664|
| 21|235.94116044674476|
| 22| 211.3947889182058|
| 23|202.84264305177112|
| 24|185.71206225680933|
| 25|131.02114803625378|
| 26| 144.0081705150977|
| 27|134.14732142857142|
| 28| 125.835448392555|
| 29|120.81818181818181|
| 30|115.20804195804196|
| 31|118.45985832349469|
| 32|114.27997227997228|
+---+-----+
only showing top 20 rows
```

RESULT AND CONCLUSION

Key Findings from the data :

- Gender Interaction :
 - Women interact more with fb then men
 - Women receive & give more likes than men on average
 - Women initiate less friendships than men compared proportionally to friend count
- Likes Split Up :
 - Shows inclination towards mobile apps o More prominence of likes from mobile compared to site though few users still interact with sites
 - There can be a gradual shift from mobile to site in years to come seeing the trend
- User Counts :
 - Age Distribution shows peak between 15-28 years, then small peak between 45-55 years. It seems like ages of parents and kids of a generation