**CALIFORNIA STATE UNIVERSITY, FULLERTON**

**Department of Computer Science**

Final Project Submission
for
CPSC 531- Advanced Database Management System

# HADOOP SPARK-BASED REAL-TIME ONLINE SALES DATA ANALYSIS

Submitted by:

Meghana Bodapati ThirumalaNaidu, CWID: 885206029
Pooja Honneshwari Ravi, CWID: 885237307

# CONTENTS



# LIST OF FIGURES

# PROBLEM STATEMENT

Everyone in the modern world now follows the trend of online purchasing. E-commerce websites have made it easier and is the mainstream market for businesses to connect with clients, regardless of their location, interests, or preferences.

It's crucial to have an examination of solid sales data. Decision-making is aided, sales quality is maintained, and clients are better understood, all of which improve performance going forward.

# AIM OF THE PROJECT

The goal of this project is to gather real-time sales information for goods sold across various platforms in several states. The outcome generated depicts the sales of 5 platforms such as Facebook, Twitter , Google, Affiliate and Organic (prouduct's own website) in different states, which aids the enhancement of cross-platform advertising tactics.

# APPROACH TO DESIGN THE PROJECT

- Kafka is an open-source distributed stream-processing framework that collects real-time data from the e-commerce website and stores it in Kafka-broker(Cluster).
- This stream of data is directed to the processing engine (Spark) through Kafka-consumer.
- The data streams undergo various processes such as Data injection, Data Storage, Data Processing, and Data Analysis.
- The analysis results are stored in MySQL and the raw data is stored in Cassandra (because for further enhancement in the application, where-in we intend to add prediction models, data of any format can be stored as it is a NO-SQL-based database).
- After analyzing, the visualization outcome is depicted in the Superset application.

### Kafka Internal Functioning

- The topic orderstopic is created in the Kafka-Cluster.
- Kafka mainly has two important API's.
- Kafka-Producer collects all the stream of data from the e-commerce website and allows the application to write that into the topic in the cluster.
- From the other side, the kafka-Consumer allows the application to read the stream of data from the topic in the cluster and gives it to spark for further processing.

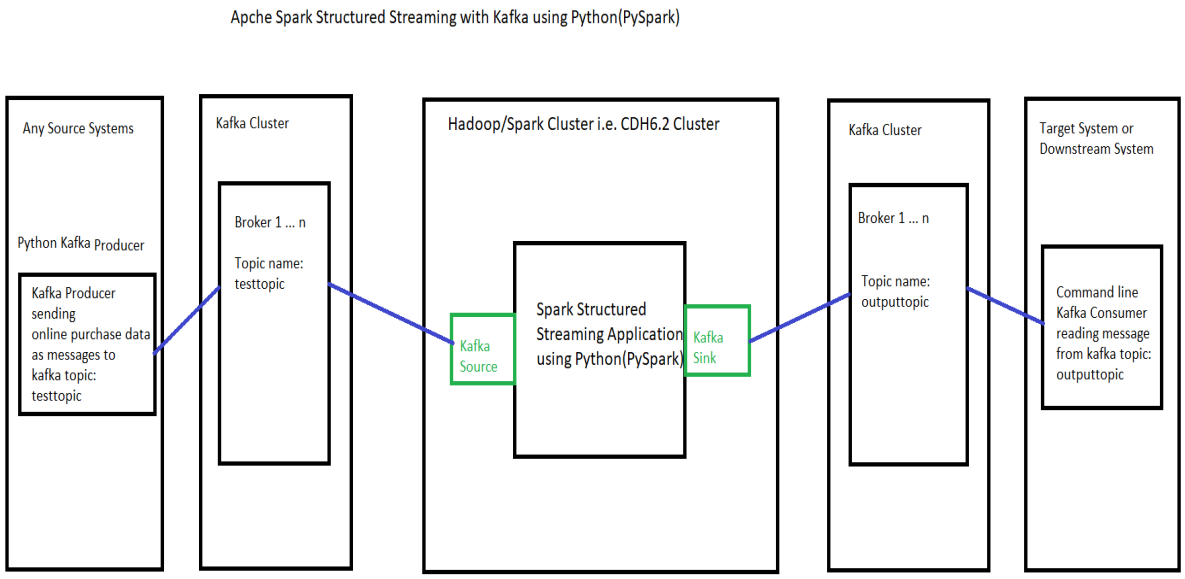


**Fig 1: Internal working of Kafka**

# TOOLS AND TECHNOLOGIES USED

- Hadoop – open-source software utility used for storage and computation of data.
- Kafka – used for capturing and storing streams of real-time data into categories called topics and later gives it to a processing engine for analysis.
- Spark – a data processing engine used for collecting data-stream from Kafka and further storing, processing, and analyzing the data.

- MySQL – is an open-source RDMS used to store the analyzed data.
- Cassandra – NoSQL database management system used to store raw data.
- Superset- an open-source software application for data exploration and data visualization.
- Python (Anaconda, PyCharm) – The programming language used for the analysis.

# ARCHITECTURE

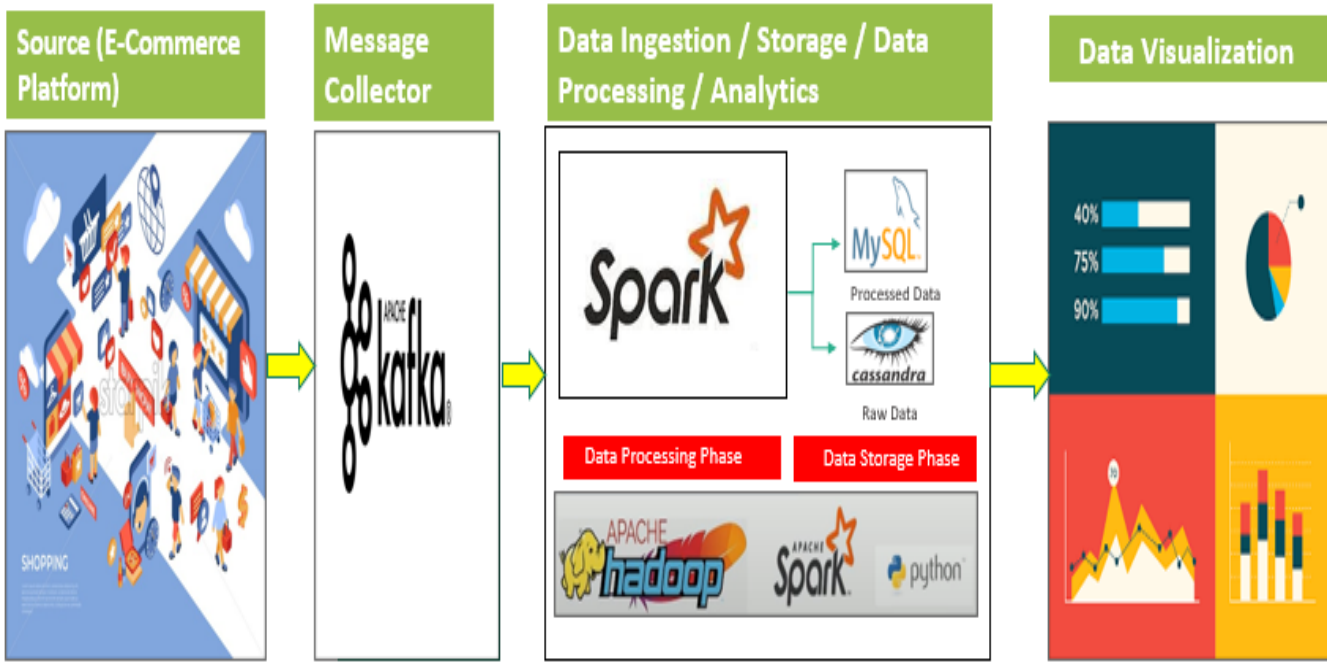


Fig: Real-Time Data Analysis Application Architecture

# DATASET INFORMATION

Three datasets were used:

- Customers dataset
- Orders dataset
- Products dataset

**Data Attributes: customers.csv**

- ID – customer's id
- Name – customer's name
- Address – customer's address
- Birth Date – customer's date of birth
- City – customer's city he lives
- Created At - customer's account created date
- Email - customer's email address
- Latitude - customer's location details
- Longitude - customer's location details
- Password - customer's account password
- Source - customer's source through which he made an order
- State - customer's state
- Zip - customer's area zip code

| ID | Name | Address | Birth Date | City | Created At | Email | Latitude | Longitude | Password | Source | State | Zip |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Hudson Borer | 9611-9809 West Rosedale Ro | 12/12/1986 | Wood River | 2017-10-07T01:34:35 | borer-hudson@yaho | 40.7131489 | -98.5259864 | ccca881f-3e4 | Twitter | NE | 68883 |
| 2 | Domenica Williamson | 101 4th Street | 6/10/1967 | Searsboro | 2018-04-09T12:10:05 | williamson-domeni | 41.5813224 | -92.6991321 | eafc45bf-cf8 | Affiliate | IA | 50242 |
| 3 | Lina Heaney | 29494 Anderson Drive | 12/18/1961 | Sandstone | 2017-06-27T06:06:20 | lina.heaney@yahoo | 46.1197304 | -92.8416108 | 36f67891-34e | Facebook | MN | 55072 |
| 4 | Arnold Adams | 2-7900 Cuerno Verde Road | 8/12/1992 | Rye | 2019-02-21T13:59:15 | adams.arnold@gma | 37.9202933 | -104.972691 | 537a727b-75 | Google | CO | 81069 |
| 5 | Dominique Leffler | 761 Fish Hill Road | 4/20/1974 | Beaver Dams | 2017-09-05T03:36:44 | leffler.dominique@ | 42.348954 | -77.056681 | 6a802b6c-4d | Twitter | NY | 14812 |
| 6 | Rene Muller | 1243 West Whitney Street | 3/27/1983 | Morse | 2016-09-22T10:08:29 | rene.muller@gmail. | 30.1514772 | -92.4861786 | 760495fb-9c3 | Google | LA | 70559 |
| 7 | Roselyn Bosco | 630 Coaker Road | 1/19/1996 | Leakesville | 2018-05-24T06:18:20 | bosco.roselyn@hotr | 31.2341055 | -88.5856948 | 43adf4af-055 | Facebook | MS | 39451 |
| 8 | Aracely Jenkins | 1167 East 570th Avenue | 6/5/1973 | Pittsburg | 2017-08-18T02:55:11 | aracely.jenkins@gm | 37.4347209 | -94.6426865 | 6bb01b7f-64 | Facebook | KS | 66762 |
| 9 | Anais Ward | 5816-5894 280th Street | 10/16/1999 | Ida Grove | 2018-10-13T23:52:56 | ward.anais@gmail.c | 42.2979021 | -95.4673587 | a3de5208-2f | Twitter | IA | 51445 |
| 10 | Tressa White | 13081-13217 Main Street | 1/13/1968 | Upper Sandusky | 2018-08-16T12:16:30 | white.tressa@yahoo | 40.8006673 | -83.2838391 | 81052233-b3 | Google | OH | 43351 |
| 11 | Lolita Schaefer | 495 Juniper Road | 8/20/1982 | Pilot Mound | 2018-03-19T07:17:22 | schaefer-lolita@hot | 42.1394217 | -93.982366 | ceddd766-92 | Facebook | IA | 50223 |
| 12 | Ciara Larson | 16701-16743 449th Avenue | 12/16/1982 | Florence | 2018-06-27T17:25:24 | ciara-larson@hotma | 44.9564152 | -97.2287266 | eb925d11-ea | Facebook | SD | 57235 |
| 13 | Mustafa Thiel | 2993 Hoskings Ranch Road | 7/20/1963 | Santa Ysabel | 2017-09-21T16:21:06 | mustafa.thiel@hotm | 33.08172 | -116.661853 | bb85c7fa-431 | Facebook | CA | 92070 |
| 14 | Lavonne Senger | 3964 Chico River Road | 9/22/1963 | Chico | 2018-06-17T23:29:39 | senger.lavonne@ya | 39.6485802 | -121.934332 | 8beebcc9-a3 | Facebook | CA | 95928 |
| 15 | Bertrand Romaguera | 258 Opal Road | 2/14/2000 | El Paso | 2018-12-20T17:50:32 | romaguera.bertrand | 35.1080351 | -92.0101859 | 2734ae7a-aa | Twitter | AR | 72045 |
| 16 | Khalid Pouros | 3234 County Road 7 | 9/16/1961 | Greensboro | 2016-05-14T23:44:56 | khalid-pouros@yaho | 32.7042678 | -87.5413596 | e60e4c1e-c9 | Google | AL | 36744 |
| 17 | Lora Cronin | 6111 Rogue Riv | 2/16/1990 | Gold Beach | 2017-03-05T20:38:57 | lora-cronin@gmail.c | 42.5048529 | -124.187837 | 1b414387-e9 | Organic | OR | 97444 |
| 18 | Arely Pollich | 13116 Northeast Sandy Boule | 6/27/1990 | Portland | 2017-04-25T20:11:39 | pollich-arely@yaho | 45.554885 | -122.527744 | 251cd4bd-d7 | Affiliate | OR | 97230 |
| 19 | Wyman Kertzmann | 1460 Grays Creek Road | 7/23/1975 | Grants Pass | 2018-12-03T07:27:47 | wyman-kertzmann@ | 42.3222669 | -123.324017 | 89ef82c4-55e | Affiliate | OR | 97527 |

**Fig 3: Dataset: customers.csv**

**Data Attributes: orders.csv**

- ID – order id
- Created At – order created date
- Discount – order discount
- Product ID – product id of the product the customer ordered
- Quantity – quantity of the product ordered
- Subtotal – product amount
- Tax – tax for the product
- Total – total amount including tax
- User Id – customer's id

| ID | Created At | Discount | Product ID | Quantity | Subtotal | Tax | Total | User ID |
|---|---|---|---|---|---|---|---|---|
| 1 | 2019-02-11T21:40:27.892+05:30 | | 14 | 2 | 37.64814539 | 2.07 | 39.71815 | 1 |
| 2 | 2018-05-15T08:04:04.580+05:30 | | 123 | 3 | 110.9314565 | 6.1 | 117.0377 | 1 |
| 3 | 2019-12-06T22:22:48.544+05:30 | 6.416679209 | 105 | 2 | 52.72352144 | 2.9 | 55.62209 | 1 |
| 4 | 2019-08-22T16:30:42.392+05:30 | | 94 | 6 | 109.2186416 | 6.01 | 115.2207 | 1 |
| 5 | 2018-10-10T03:34:47.309+05:30 | | 132 | 5 | 127.8819703 | 7.03 | 134.9419 | 1 |
| 6 | 2019-11-06T16:38:50.134+05:30 | | 60 | 3 | 29.80214752 | 1.64 | 31.44168 | 1 |
| 7 | 2018-09-11T11:22:26.521+05:30 | | 55 | 5 | 95.77128576 | 5.27 | 101.0611 | 1 |
| 8 | 2019-06-17T02:37:41.693+05:30 | 8.65395293 | 65 | 7 | 68.22769726 | 3.75 | 71.9646 | 1 |
| 9 | 2017-05-03T16:00:54.923+05:30 | 3.594742155 | 184 | 3 | 77.39827487 | 4.26 | 81.67427 | 1 |
| 10 | 2020-01-17T01:44:37.233+05:30 | | 6 | 2 | 97.43621265 | 5.36 | 102.7706 | 1 |
| 11 | 2018-07-22T20:31:01.969+05:30 | | 76 | 6 | 63.82421061 | 3.51 | 67.30536 | 1 |
| 12 | 2018-06-26T23:21:13.271+05:30 | | 7 | 7 | 148.2290053 | 10.19 | 158.4454 | 3 |
| 13 | 2019-04-06T01:04:43.973+05:30 | 2.117341034 | 70 | 2 | 57.49300381 | 3.95 | 61.42339 | 3 |
| 14 | 2017-05-25T20:50:37.137+05:30 | | 139 | 4 | 51.18512213 | 3.52 | 54.67137 | 3 |
| 15 | 2018-06-26T02:24:38.715+05:30 | | 116 | 5 | 114.4248513 | 7.87 | 122.3092 | 3 |
| 16 | 2017-12-14T11:28:30.031+05:30 | | 68 | 1 | 76.82895922 | 5.28 | 82.15965 | 3 |
| 17 | 2020-02-06T12:14:36.282+05:30 | | 48 | 1 | 123.2088425 | 8.47 | 131.6599 | 3 |
| 18 | 2020-04-10T23:29:46.804+05:30 | | 12 | 2 | 116.0142758 | 7.98 | 123.9353 | 3 |
| 19 | 2019-02-14T07:28:31.104+05:30 | | 136 | 1 | 105.2040232 | 7.23 | 112.5304 | 3 |

**Fig 4: Dataset: orders.csv**

**Data Attributes: products.csv**

- ID – product id
- Category – product category
- Created At – a product created date
- EAN (European Article Number) – unique number for the product
- Price – product amount
- Rating – product rating
- Title – name of the product
- Vendor – vendor name

| ID | Category | Created At | Ean | Price | Rating | Title | Vendor |
|---|---|---|---|---|---|---|---|
| 1 | Gizmo | 2017-07-19T19:44:56.582+05:30 | 1.01895E+12 | 29.46326 | 4.6 | Rustic Paper Wallet | Swaniawski, Casper and Hilll |
| 2 | Doohickey | 2019-04-11T08:49:35.932+05:30 | 7.66352E+12 | 70.0799 | 0 | Small Marble Shoes | Balistreri-Ankunding |
| 3 | Doohickey | 2018-09-08T22:03:20.239+05:30 | 4.96628E+12 | 35.38874 | 4 | Synergistic Granite Ch | Murray, Watsica and Wunsch |
| 4 | Doohickey | 2018-03-06T02:53:09.937+05:30 | 4.1345E+12 | 73.99178 | 3 | Enormous Aluminum | Regan Bradtke and Sons |
| 5 | Gadget | 2016-10-03T01:47:39.147+05:30 | 5.49974E+12 | 82.7451 | 4 | Enormous Marble Wa | Price, Schultz and Daniel |
| 6 | Doohickey | 2017-03-29T05:43:40.150+05:30 | 2.29334E+12 | 64.95748 | 3.8 | Small Marble Hat | Nolan-Wolff |
| 7 | Doohickey | 2017-06-03T03:07:28.061+05:30 | 1.57967E+11 | 98.81934 | 4.3 | Aerodynamic Linen C | Little-Pagac |
| 8 | Doohickey | 2018-04-30T15:03:53.193+05:30 | 1.07877E+12 | 65.89216 | 4.1 | Enormous Steel Watc | Senger-Stamm |
| 9 | Widget | 2019-02-07T08:26:25.647+05:30 | 7.21747E+12 | 58.31312 | 4.2 | Practical Bronze Com | Keely Stehr Group |
| 10 | Gizmo | 2017-01-09T09:51:20.352+05:30 | 1.80796E+12 | 31.78622 | 4.3 | Mediocre Wooden Ta | Larson, Pfeffer and Klocko |
| 11 | Gadget | 2018-05-28T08:02:54.482+05:30 | 3.64241E+12 | 88.30453 | 0 | Ergonomic Silk Coat | Upton, Kovacek and Halvorson |
| 12 | Gizmo | 2017-11-12T14:51:16.221+05:30 | 9.48247E+12 | 77.34285 | 4.4 | Sleek Paper Toucan | Mueller-Dare |
| 13 | Gizmo | 2016-05-24T23:09:46.392+05:30 | 3.99569E+11 | 75.08617 | 0 | Synergistic Steel Chai | Mr. Tanya Stracke and Sons |
| 14 | Widget | 2017-12-31T14:41:56.870+05:30 | 8.83342E+12 | 25.09876 | 4 | Awesome Concrete Sl | McClure-Lockman |
| 15 | Widget | 2016-09-08T14:42:57.264+05:30 | 5.88165E+12 | 25.09876 | 4 | Aerodynamic Paper C | Friesen-Anderson |
| 16 | Gadget | 2017-01-30T22:00:25.015+05:30 | 9.63314E+12 | 44.07353 | 4 | Incredible Bronze Par | Okuneva, Kutch and Monahan |
| 17 | Doohickey | 2017-01-31T11:05:14.863+05:30 | 6.70464E+12 | 53.29072 | 4.3 | Enormous Wool Car | Hilpert, Jacobs and Hauck |
| 18 | Gadget | 2017-02-11T03:45:40.957+05:30 | 8.90936E+12 | 54.60205 | 4.4 | Lightweight Paper Bot | Wilkinson-Gottlieb |
| 19 | Doohickey | 2016-11-06T23:27:12.642+05:30 | 4.30772E+12 | 42.67117 | 4.2 | Rustic Concrete Bottle | Gibson, Turner and Douglas |

**Fig 5: Dataset: products.csv**

# CODE LINK

https://github.com/meghanabt/HADOOP-SPARK-BASED-REAL-TIME-ONLINE-SALES-DATA-ANALYSIS

# STEPS TO RUN THE APPLICATION

Pre-requisites:

In an Ubuntu system, install Hadoop, Zookeeper and Kafka, Spark, Anaconda, Pycharm, Mysql , Cassandra and Superset.

Steps to Configure Kafka:

1. Go to the folder where the Pycharm Community application is located, then right click and select ‘open in terminal’. Type the command ‘./pycharm.sh’.
2. In the Pycharm application, open the Kafka Producer code which is configured to read the orders dataset and produce streams of data.
3. To configure the kafka consumer to consume and pass the stream data to spark, open a new terminal and change directory to kafka path and then type the command ‘/bin/db-project/kafka/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic orderstopic(*topics name*)’. Keep both ready.
4. Now, first run the kafka_producer to produce stream data and then run the consumer command on the terminal. You can see the streaming of data on pycharm platform which will reflect as it is on the terminal as it is consumed and given as output from the kafka consumer.
5. This stream data from the consumer is fed to the pyspark application via configured jar files .On running this application on pyspark platform the processed output is stored in the mysql database and raw data is stored in cassandra.
6. The processed output is later visualized on Apache superset application in which we initially login using the superset admin USERID and password.
7. Connect the mysql database from superset using the database name and mysql jdbc url.
8. After connecting to the database create charts using the charts tab and select the table in which the processed data is stored.
9. Based on the columns generated a query can be created to obtain the desired graphs.

* Steps to configure kafka, mysql and cassandra is mentioned in a separate document and is added in the git repository *

# TEST RESULTS

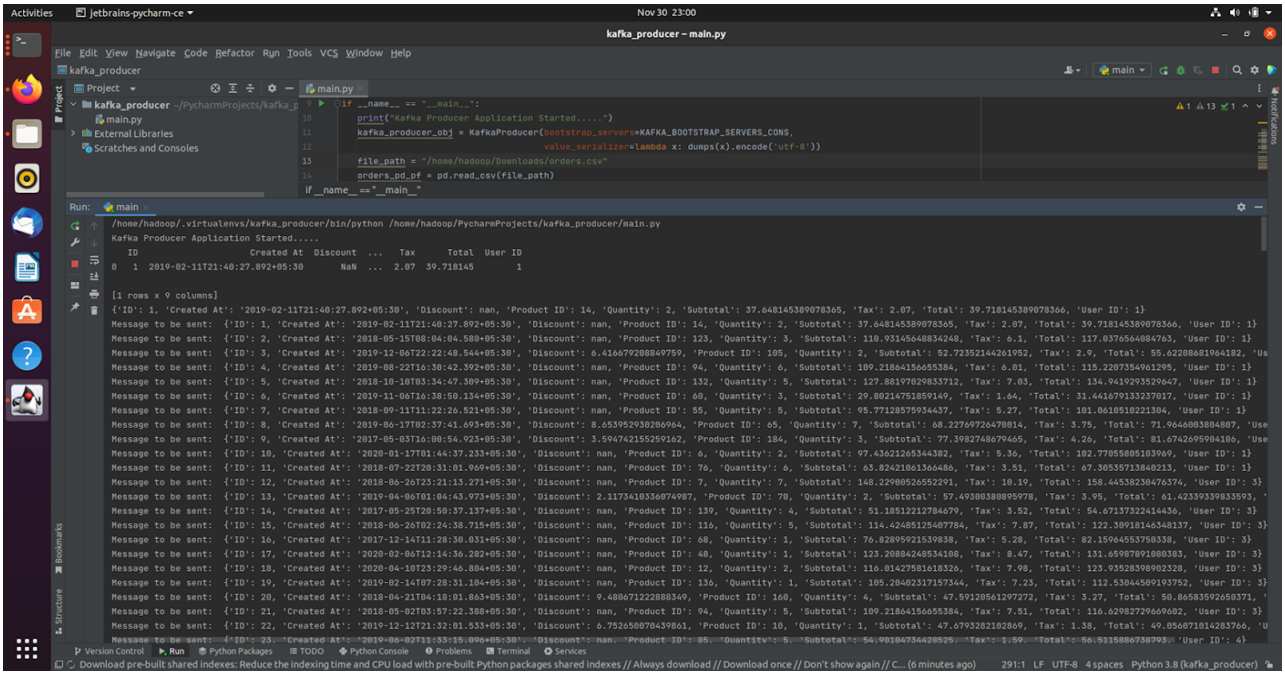


**Fig 6: Streaming of Data in Kafka Producer**

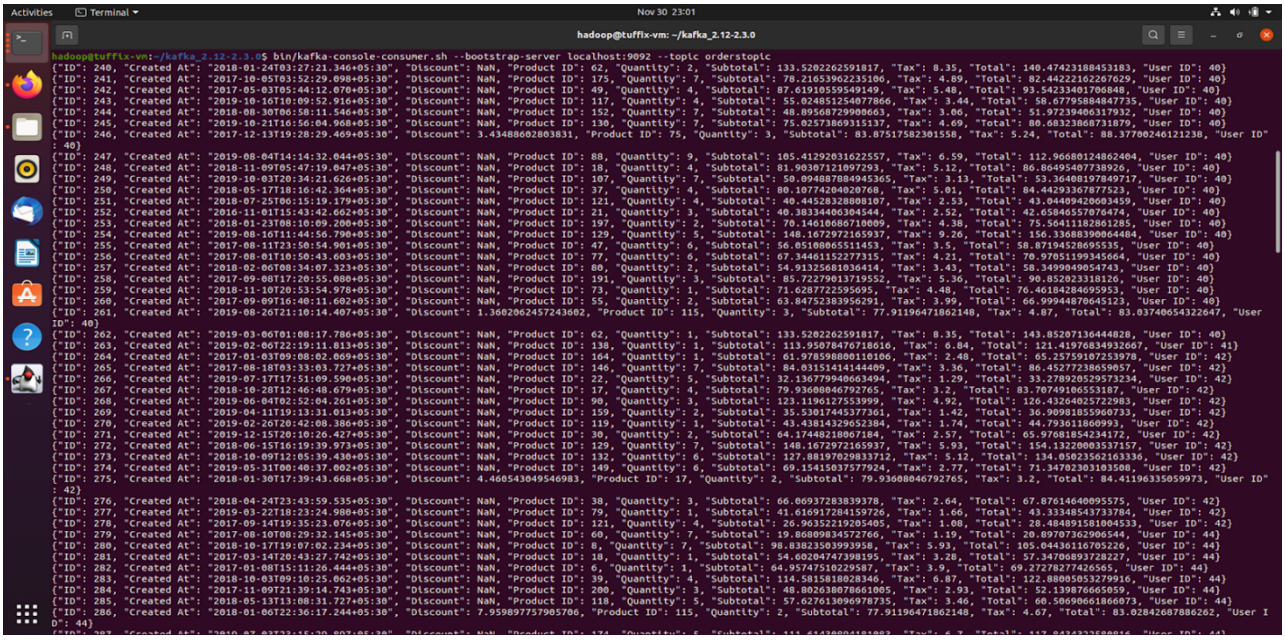

**Fig 7: Streaming of Data in Kafka Consumer**

```
Printing Schema of orders_df:
root
 |-- key: binary (nullable = true)
 |-- value: binary (nullable = true)
 |-- topic: string (nullable = true)
 |-- partition: integer (nullable = true)
 |-- offset: long (nullable = true)
 |-- timestamp: timestamp (nullable = true)
 |-- timestampType: integer (nullable = true)
```

**Fig 8: Schema of data frames generated, of DataStream of orders received from Kafka-consumer**

```
root
 |-- ID: integer (nullable = true)
 |-- Name: string (nullable = true)
 |-- Address: string (nullable = true)
 |-- Birth Date: timestamp (nullable = true)
 |-- City: string (nullable = true)
 |-- Created At: timestamp (nullable = true)
 |-- Email: string (nullable = true)
 |-- Latitude: double (nullable = true)
 |-- Longitude: double (nullable = true)
 |-- Password: string (nullable = true)
 |-- Source: string (nullable = true)
 |-- State: string (nullable = true)
 |-- Zip: integer (nullable = true)
```

**Fig 9: Schema of data frames generated, of static customer data from the dataset**

```
Printing Schema of Order_df4:
root
 |-- order_id: string (nullable = true)
 |-- created_at: string (nullable = true)
 |-- discount: string (nullable = true)
 |-- product_id: string (nullable = true)
 |-- quantity: string (nullable = true)
 |-- subtotal: string (nullable = true)
 |-- tax: string (nullable = true)
 |-- total: string (nullable = true)
 |-- customer_id: string (nullable = true)
 |-- timestamp: timestamp (nullable = true)
 |-- ID: integer (nullable = true)
 |-- Name: string (nullable = true)
 |-- Address: string (nullable = true)
 |-- Birth Date: timestamp (nullable = true)
 |-- City: string (nullable = true)
 |-- Created At: timestamp (nullable = true)
 |-- Email: string (nullable = true)
 |-- Latitude: double (nullable = true)
 |-- Longitude: double (nullable = true)
 |-- Password: string (nullable = true)
 |-- Source: string (nullable = true)
 |-- State: string (nullable = true)
 |-- Zip: integer (nullable = true)

Printing Schema of orders_df5:
root
 |-- Source: string (nullable = true)
 |-- State: string (nullable = true)
 |-- Amount_Paid: double (nullable = true)
```

**Fig 10: Schema of data frames generated, of forked data of customers and orders used for data analysis**

| | | | | |
|---|---|---|---|---|
| Organic | NM | 293.81937092516944 | 2019-10-20 14:02:45 | 412 |
| Facebook | MS | 3319.8868346644153 | 2019-10-20 14:02:45 | 412 |
| Organic | MN | 3978.229446433442 | 2019-10-20 14:02:45 | 412 |
| Facebook | CA | 4684.518476233977 | 2019-10-20 14:03:00 | 413 |
| Facebook | MS | 4407.1372272317 | 2019-10-20 14:03:00 | 413 |
| Facebook | CA | 5032.269939224234 | 2019-10-20 14:03:15 | 414 |
| Google | NE | 712.4077658179535 | 2019-10-20 14:03:15 | 414 |
| Organic | ND | 2905.9799464917774 | 2019-10-20 14:03:15 | 414 |
| Twitter | IL | 3348.821727410377 | 2019-10-20 14:03:30 | 415 |
| Organic | ND | 3674.91247055596 | 2019-10-20 14:03:30 | 415 |
| Twitter | IL | 3568.417520548799 | 2019-10-20 14:03:45 | 416 |
| Twitter | VT | 1037.8921538200907 | 2019-10-20 14:03:45 | 416 |
| Organic | MT | 9385.43902457641 | 2019-10-20 14:04:00 | 417 |
| Twitter | VT | 1464.726981042852 | 2019-10-20 14:04:00 | 417 |
| Affiliate | TX | 8335.897361481708 | 2019-10-20 14:04:15 | 418 |
| Organic | MT | 9496.278196043422 | 2019-10-20 14:04:15 | 418 |
| Twitter | CA | 7506.6223139450385 | 2019-10-20 14:04:15 | 418 |
| Google | MI | 4442.819923353086 | 2019-10-20 14:04:43 | 419 |
| Twitter | CA | 7593.102021168229 | 2019-10-20 14:04:43 | 419 |
| Affiliate | WA | 1129.1928465841045 | 2019-10-20 14:04:59 | 420 |
| Twitter | OH | 1608.691926708404 | 2019-10-20 14:04:59 | 420 |
| Organic | WI | 5012.404838772943 | 2019-10-20 14:04:59 | 420 |
| Affiliate | WA | 1345.4541751066963 | 2019-10-20 14:05:08 | 421 |
| Affiliate | MO | 5828.3170450380285 | 2019-10-20 14:05:08 | 421 |
| Affiliate | MO | 6831.34902858962 | 2019-10-20 14:05:22 | 422 |
| Google | TX | 8200.504406063952 | 2019-10-20 14:05:22 | 422 |
| Google | TX | 9082.447178734588 | 2019-10-20 14:05:31 | 423 |
| Organic | NC | 2558.6695328576693 | 2019-10-20 14:05:52 | 424 |
| Google | TX | 9946.049614318941 | 2019-10-20 14:05:52 | 424 |

**Fig 11: Processed data stored in Mysql Database**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0530 | 84.2394955306831 | | | | | | | | |
| 146 | 2017-09-29T19:13:15.649+05:30 | 22 | "NaN" | 189 | 6 | 62.18492169821006 | 4.66 | 2019-10-20 18:27:23+ | |
| 0530 | 67.40107423105458 | | | | | | | | |
| 69 | 2019-06-28T14:19:18.995+05:30 | 12 | "NaN" | 183 | 8 | 56.697412301919755 | 2.27 | 2019-10-20 18:26:06+ | |
| 0530 | 59.09276090183358 | | | | | | | | |
| 15 | 2018-06-26T02:24:38.715+05:30 | 3 | "NaN" | 116 | 5 | 114.42485125407784 | 7.87 | 2019-10-20 18:25:12+ | |
| 0530 | 122.30918146348135 | | | | | | | | |
| 209 | 2019-01-05T00:03:51.181+05:30 | 36 | "NaN" | 189 | 2 | 93.27738254731509 | 6.06 | 2019-10-20 18:28:26+ | |
| 0530 | 98.64606378422512 | | | | | | | | |
| 22 | 2019-12-12T21:32:01.533+05:30 | 4 | 6.752650070439861 | 10 | 1 | 47.6793282102869 | 1.38 | 2019-10-20 18:25:19+ | |
| 0530 | 49.05607101428377 | | | | | | | | |
| 112 | 2018-06-12T04:48:40.715+05:30 | 17 | "NaN" | 115 | 7 | 77.91196471862148 | 0.0 | 2019-10-20 18:26:49+ | |
| 0530 | 77.94164494454556 | | | | | | | | |
| 98 | 2020-04-06T00:45:35.910+05:30 | 17 | "NaN" | 26 | 3 | 68.12471180754113 | 0.0 | 2019-10-20 18:26:35+ | |
| 0530 | 68.0421431688182 | | | | | | | | |
| 152 | 2018-04-28T23:03:07.994+05:30 | 22 | "NaN" | 61 | 3 | 23.537915510955656 | 1.77 | 2019-10-20 18:27:29+ | |
| 0530 | 25.35326123452555 | | | | | | | | |
| 135 | 2018-01-04T14:19:57.077+05:30 | 21 | "NaN" | 120 | 1 | 83.5020135028928 | 4.59 | 2019-10-20 18:27:12+ | |
| 0530 | 88.08386047261446 | | | | | | | | |
| 54 | 2018-10-26T20:41:23.428+05:30 | 10 | 7.637620601487041 | 68 | 4 | 115.24343882309758 | 6.63 | 2019-10-20 18:25:51+ | |
| 0530 | 122.11637851493805 | | | | | | | | |
| 75 | 2017-04-10T23:33:10.082+05:30 | 12 | "NaN" | 13 | 2 | 75.08616927403709 | 3.0 | 2019-10-20 18:26:12+ | |
| 0530 | 77.95866140384386 | | | | | | | | |
| 115 | 2017-08-30T06:13:52.097+05:30 | 18 | "NaN" | 191 | 6 | 85.72279013719552 | 0.0 | 2019-10-20 18:26:52+ | |
| 0530 | 86.029271263595 | | | | | | | | |
| 203 | 2017-08-17T18:17:48.717+05:30 | 36 | "NaN" | 171 | 6 | 70.05110494330981 | 4.55 | 2019-10-20 18:28:20+ | |
| 0530 | 74.93400185111025 | | | | | | | | |
| 174 | 2020-04-09T12:03:00.755+05:30 | 29 | 3.0797162375864824 | 190 | 3 | 128.58418520579332 | 7.39 | 2019-10-20 18:27:51+ | |
| 0530 | 135.51580933716718 | | | | | | | | |
| 173 | 2018-10-22T18:56:58.539+05:30 | 29 | "NaN" | 91 | 7 | 65.09432810381134 | 3.74 | 2019-10-20 18:27:50+ | |
| 0530 | 68.5948263899836 | | | | | | | | |
| 183 | 2018-08-25T14:09:48.528+05:30 | 32 | "NaN" | 87 | 8 | 117.25536340498041 | 3.4 | 2019-10-20 18:28:00+ | |
| 0530 | 120.5753199106972 | | | | | | | | |
| 58 | 2019-03-09T01:05:15.721+05:30 | 10 | "NaN" | 91 | 1 | 65.09432810381134 | 3.74 | 2019-10-20 18:25:55+ | |
| 0530 | 68.8829031879211 | | | | | | | | |
| 186 | 2018-12-23T21:15:46.547+05:30 | 33 | "NaN" | 108 | 2 | 50.094887884945365 | 1.45 | 2019-10-20 18:28:03+ | |
| 0530 | 51.16706768559577 | | | | | | | | |
| 148 | 2019-01-26T14:21:47.436+05:30 | 22 | "NaN" | 106 | 1 | 52.72352144261952 | 3.95 | 2019-10-20 18:27:25+ | |
| 0530 | 56.56723390171032 | | | | | | | | |

**Fig 12: Raw input data stored in Cassandra Database**

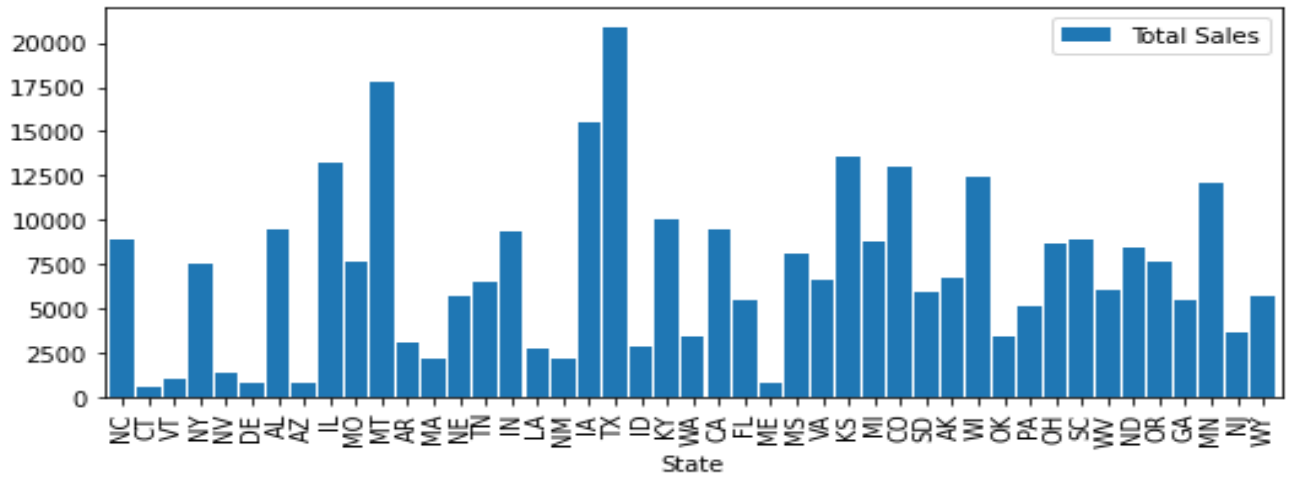


**Fig 13: Total Sales in different states with Source Organic**

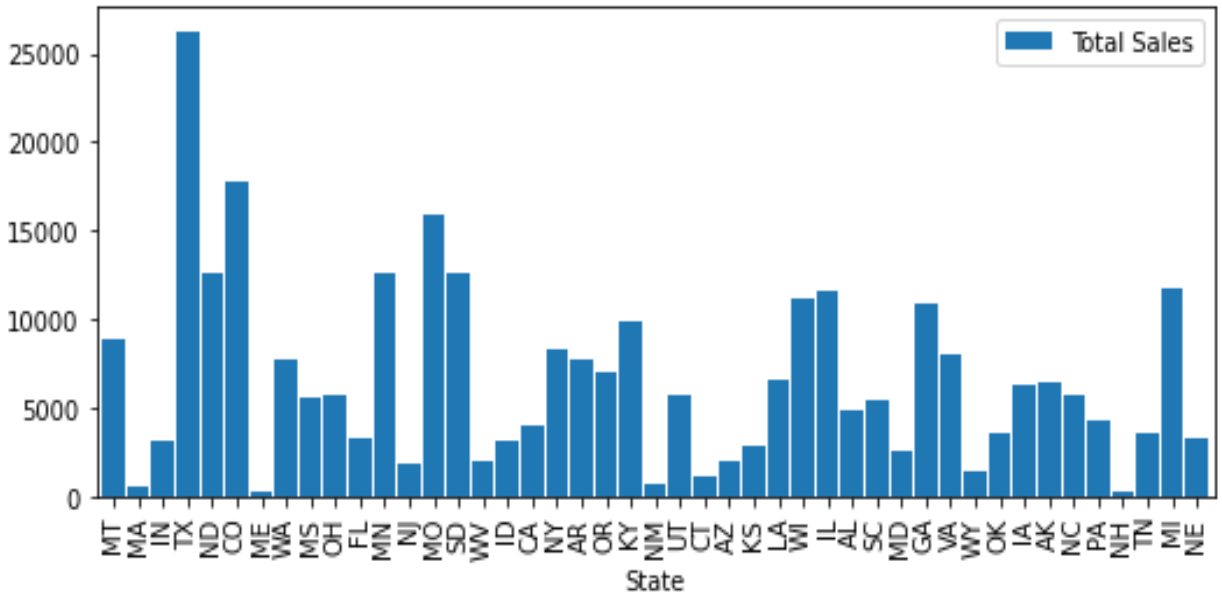


**Fig 14: Total Sales in different states with Source Affiliate**

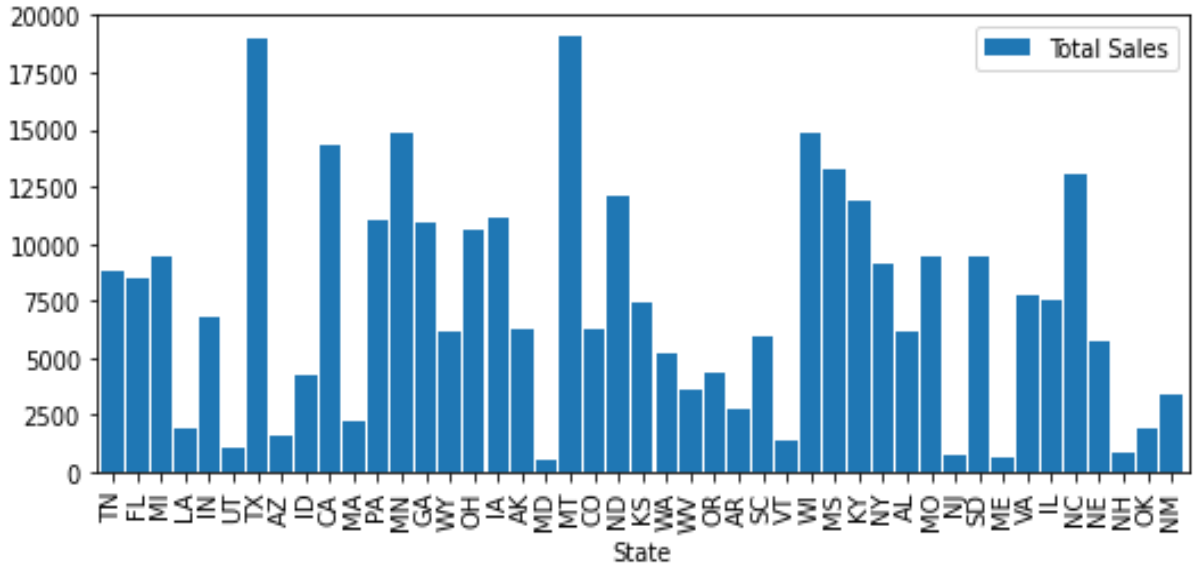


**Fig 15: Total Sales in different states with Source Facebook**

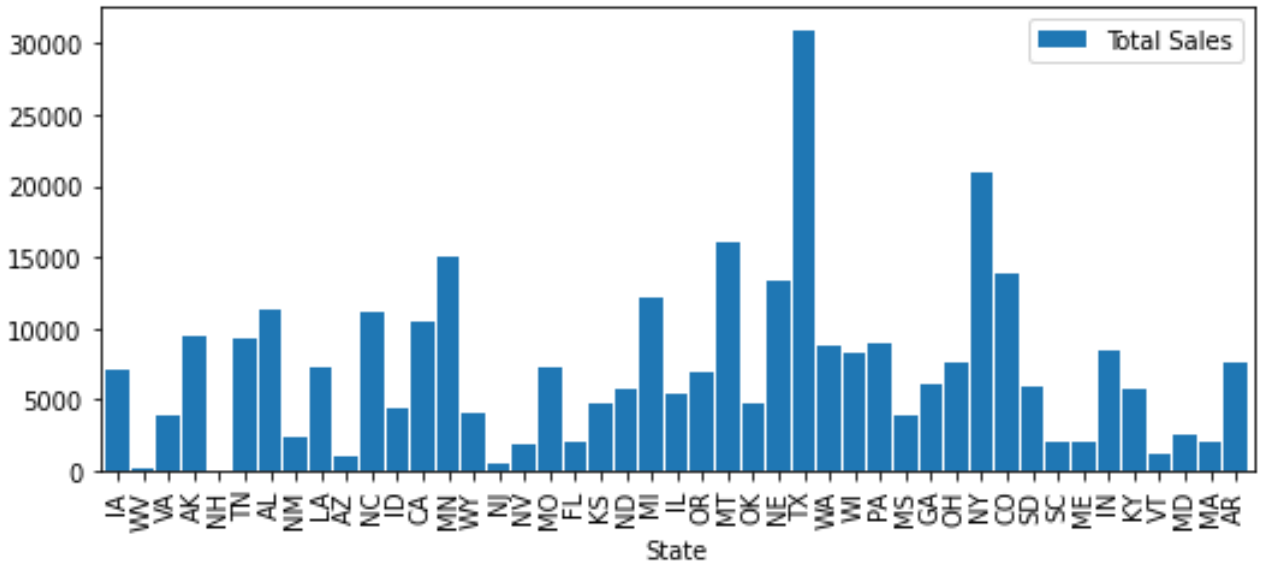


**Fig 16: Total Sales in different states with Source Google**

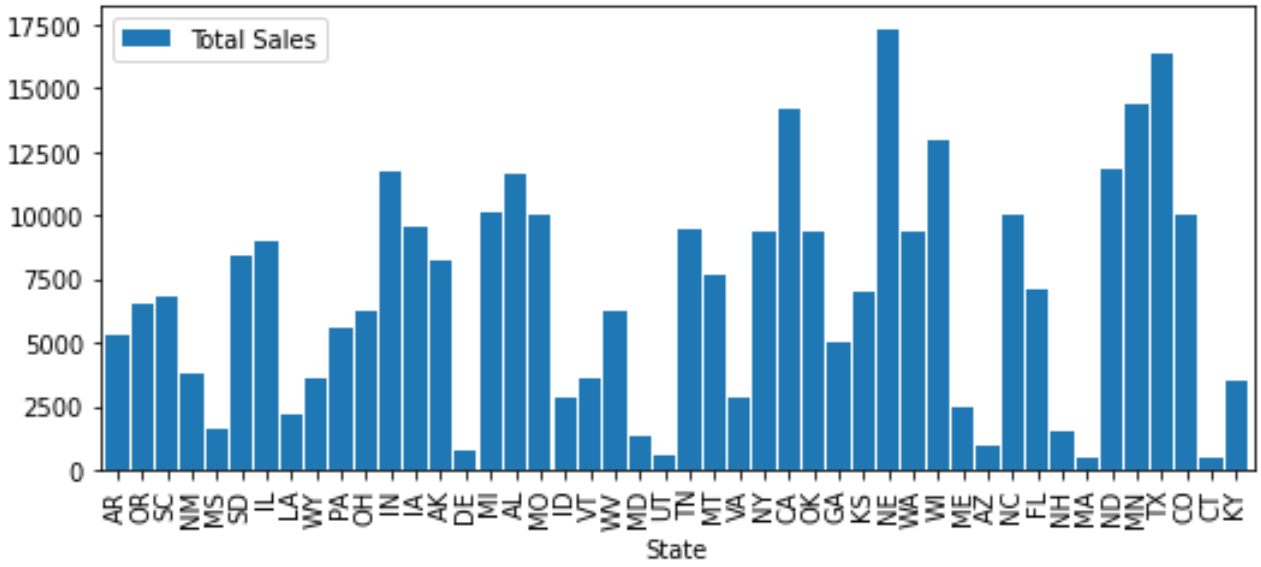


**Fig 17: Total Sales in different states with Source Twitter**

# FUTURE SCOPE

Currently, we have used Cassandra NoSQL data store to store raw data.

As cloud storage becomes more scalable and offers more security, we will be able to employ S3 object storage to develop machine learning models and further complex analytics.

# CONCLUSION

Analyzing the total sales in each state generated by each source will help the e-commerce company choose which marketing platform will maximize its product sales.

Likewise, for the marketing platform, the analysis will aid in increasing their business strategy.