**Building data pipeline**

In this project I have clickstream data which is stored on AWS S3 data lake. I have used Databricks to extract and transform the data and later stored it in Hive table. For visualization, I have linked the Hive table to Tableau. The architecture of the data pipeline I built is shown below, along with a description of how it works from beginning to end.

A picture containing diagram

Description automatically generated

**Data pipeline architecture**

**Extraction of Data from S3**

This notebook shows steps to create Dataframe extracted from data stored in AWS S3. There are two ways to establish access to S3: IAM roles and access keys. I have used access keys for this purpose. Below is the code:

Graphical user interface, text, application, email

Description automatically generated

Once I have mounted our S3 bucket to DBFS, I can access it by using the command below. It is accessing dbutils, available in Scala and Python, so you can run file operations command from your notebooks.

Graphical user interface, application, email

Description automatically generated

Thus, we can load the data into dataframe using below command:

Graphical user interface, text, application, email

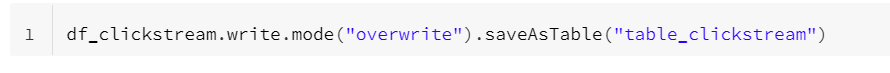
Description automatically generated

The data has 7 columns:

Text

Description automatically generated

Saving in Hive Table, so that can perform queries using SPARK SQL API.



Extraction of Data using spark sql:

**Performing EDA:**

**Total session the dataset has**: 86468

Text

Description automatically generated with medium confidence

**The platforms used by the website:**

There are 6 platforms used by the website are android, pwa(progressive web app), pwa-mobile, mobile\_web, ios, and mobile\_app

Graphical user interface, text, application

Description automatically generated

**Page\_type vs count:**

We can see there are 3 types of pages mostly visited by the users, which are shop-listings, shop-details, shop search.

Graphical user interface, application

Description automatically generated

**Potential customers:**

We can see user\_id bf9ea30021778e68b31afd1aef928967 has 2.45% records, which is maximum. Thus, we can say this customer is a potential customer.

Graphical user interface, text, application

Description automatically generated

**Event\_type vs count:**

Here view\_plp, view\_pdp and view\_srp has the maximum records. This implies these pages are most of the times visited by the customers.

Graphical user interface, application

Description automatically generated

**Preparing the clickstream data:**

Used Spark SQL and Spark functions to build Dataframes with aggregated web metrics.

I have created 3 web metrics that are useful for an e-commerce:

* events\_per\_session: Count of events per session id
* page\_views\_per\_session: Count of page views per session id
* time\_slot: There are 8 time slots I have created 0-3, 3-6, 6-9, 9-12, 12-15, 15-18, 18-21, 21-24. Here 0-3 implies from might till 3am, 3-6 implies 3-6 am and so on. This metrics will check the time and allot one time slot from these.

Text

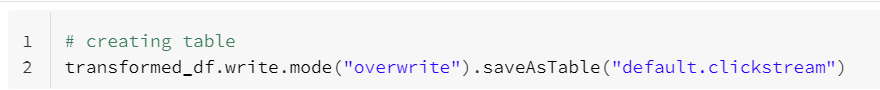
Description automatically generated

**Below shows the schema of the transformed dataframe:**

Text

Description automatically generated

**Loading the transformed dataframe into Hive Table:**



**The below code shows this data is just of 2 days from March 31st and April 1st 2021. But records of 31st March constitute less than .01%. Hence, we can say records are of April 1st**.

Graphical user interface, text, application, email

Description automatically generated

**Connecting to Tableau:**

I selected **Databricks** in **To a Server** section and filled my Databricks cluster details in it. One thing to be noted is, the cluster should be in running status.

Graphical user interface

Description automatically generated

**Visualization:**

**Below chart details:**

Top left shows the count of clicks for each time slot.

Top right chart shows the number of users based on the platform they are using to access the website.

Bottom left chart shows count of clicks for each event type.

Bottom right chart shows count of clicks for each page type.

Chart, bubble chart, treemap chart

Description automatically generated

**We can deduce the following from the above visualization:**

* The dashboard shows that the majority of the customers are involved between the hours of midnight and 3 a.m.
* The top three channels used by users to access the website are progressive web app (pwa), pwa mobile, and ios.
* Testing product descriptions, browsing with filters, and browsing with search keywords are the top three event types that users look at.
* Shop-listings, shop-details, and shop-search are the top three-page forms that users look at.

I have added a new chart to the visualization below that displays the users and the number of clicks they've made. The important thing to note is that in only one day, the user id bf9ea30021778e68b31afd1aef928967 has 203,592 clicks. It means that this is a prospective client. The consumer is searching for the product description page, as seen in the bottom left table. As a result, it is possible to submit ads for items that are close to the quest in order to boost sales.

Chart, waterfall chart

Description automatically generated

**Conclusion**

I have found that there is a lot of traffic on the website between noon and 6 a.m. As a result, we should have enough servers to manage all the requests during these periods. From 6 a.m. to noon, there is also a decrease in traffic. As a result, we can use auto-scaling to manage fluctuations in the number of requests while also saving money.

The website is accessed by many users through three major platforms: pwa, pwa smartphone, and ios. Since mobile devices account for most clicks, the production of mobile-based applications should be prioritized to reach out to potential customers.

One potential consumer has been identified and is currently browsing the product description page. As a result, the consumer indicates that he or she is willing to purchase that product or similar items. Similarly, we can identify more frequent users and give them similar ads based on their product searches, thus increasing sales.