**ASSIGNMENT**

**MSc Data Science Part II**

**Name: Shweta Sandip Kharade**

**Roll No: 10**

**Subject: Data Engineering and Applied Data Science**

**Topic: Mobile Device Usage and User Behavior**

## **Introduction :**

Data engineering involves designing, building, and maintaining systems that collect, store, and process large datasets. Apache Airflow is a key tool in this process, used to automate and orchestrate data pipelines by defining workflows as Directed Acyclic Graphs (DAGs). It ensures tasks run in the correct order and can handle complex data workflows. Data engineers use Airflow to build scalable, efficient data pipelines, integrating data from various sources, maintaining data quality, and supporting analytics, machine learning, and business intelligence efforts.

## 

## **About Dataset :**

This dataset provides a comprehensive analysis of mobile device usage patterns and user behavior classification. It contains 700 samples of user data, including metrics such as app usage time, screen-on time, battery drain, and data consumption. Each entry is categorized into one of five user behavior classes, ranging from light to extreme usage, allowing for insightful analysis and modeling.

## **Process of Creating pipeline using AirFlow :**

**Step 1 :**

Download dataset from kaggle.com and keep it in home directory.

**Step 2 :**

Create DAG with name ‘assignmentdag.py’

detest@deone:~$ sudo nano airflow/dags/assignmentdag.py

import datetime as dt

from datetime import timedelta

from airflow import DAG

from airflow.operators.bash\_operator import BashOperator

from airflow.operators.python\_operator import PythonOperator

import pandas as pd

import sklearn

from sklearn.preprocessing import StandardScaler,LabelEncoder

from elasticsearch7 import Elasticsearch

default\_args = {

'owner': 'detest',

'start\_date': dt.datetime(2024, 10, 4),

'retries': 1,

'retry\_delay': dt.timedelta(minutes=5),

}

def preprocess\_data():

df=pd.read\_csv('/home/detest/deassignment/user\_behavior\_dataset.csv')

df.dropna(inplace=True)

# Feature engineering: Create new features

df['App Usage per Installed App'] =df['App Usage Time (min/day)']/df['Number of Apps Installed']

df['Screen On to App Usage Ratio'] =df['Screen On Time (hours/day)']/df['App Usage Time (min/day)']

# Binning age into categories (age brackets)

bins = [0, 18, 30, 45, 60, 100]

labels = ['0-18', '19-30', '31-45', '46-60', '60+']

df['Age Group'] = pd.cut(df['Age'], bins=bins, labels=labels)

df.drop('Age', axis=1, inplace=True) # Drop original Age column

# Save the preprocessed data

df.to\_csv('preprocessdata.csv')

def validate\_data():

df = pd.read\_csv('preprocessdata.csv')

# Perform simple checks on the dataset to ensure it’s clean

assert df.isnull().sum().sum() == 0, "There are still missing values!"

assert df.shape[0] > 0, "The dataset is empty!"

print("Data validation passed. Preprocessed data is clean.")

df.to\_csv('/home/detest/deassignment/preprocessed\_output.csv')

def insertElasticsearch():

es = Elasticsearch()

df=pd.read\_csv('/home/detest/deassignment/preprocessed\_output.csv')

df=df.drop(columns=['Unnamed: 0'],errors='ignore')

for i,r in df.iterrows():

doc=r.to\_json()

res=es.index(index="mobileusages",doc\_type="doc",body=doc)

print(res)

with DAG('PreprocessedData',

default\_args=default\_args,

schedule\_interval='@once',

# '0 \* \* \* \*',

)as dag:

preprocess\_data\_task = PythonOperator(task\_id='clean',python\_callable=preprocess\_data)

validate\_data\_task = PythonOperator(task\_id='validate',python\_callable=validate\_data)

insertData = PythonOperator(task\_id='InsertDataElasticsearch',python\_callable=insertElasticsearch)

preprocess\_data\_task >> validate\_data\_task >> insertData

**Step 3 :**

**Start the webserver and scheduler**

detest@deone:~$ ~/.local/bin/airflow webserver

detest@deone:~$ ~/.local/bin/airflow scheduler

Go to browser and open the given link: [http://localhost:8080](http://localhost:8080/nifi/)

**Step 4 :**

**Start elasticsearch and kibana**

detest@deone:~$ /usr/lib/elasticsearch/bin/elasticsearch

Go to browser and open the link: [http://localhost:9200](http://localhost:9200/)

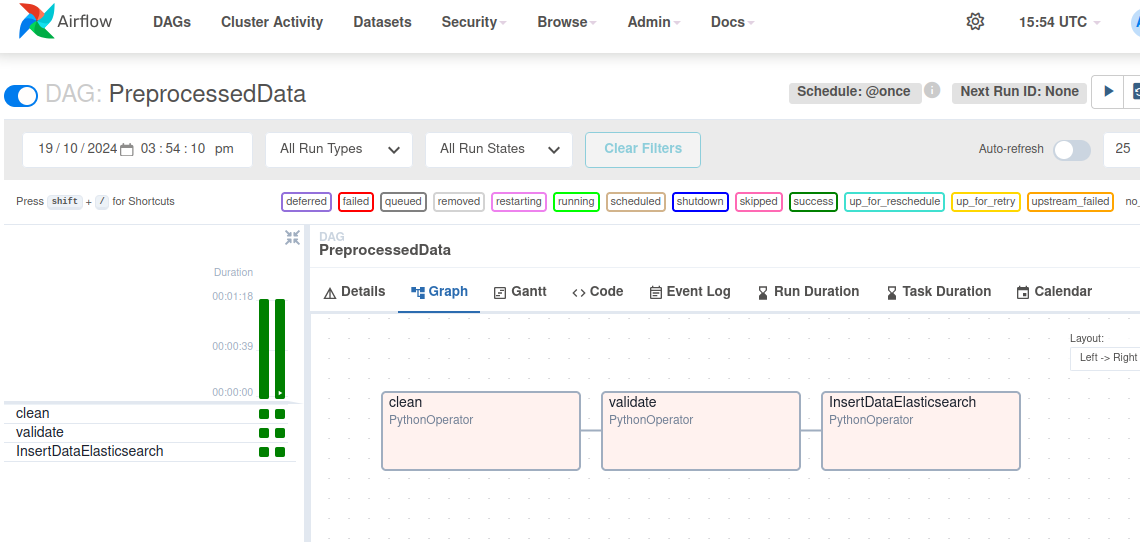
detest@UbuntuDE:~$ /usr/lib/kibana/bin/kibana

Go to browser and open the link: <http://localhost:5601/app/home#/>

**Step 5 :**

Go to the airflow on browser and run DAG:PreprocessedData

DAG is started and finished successfully

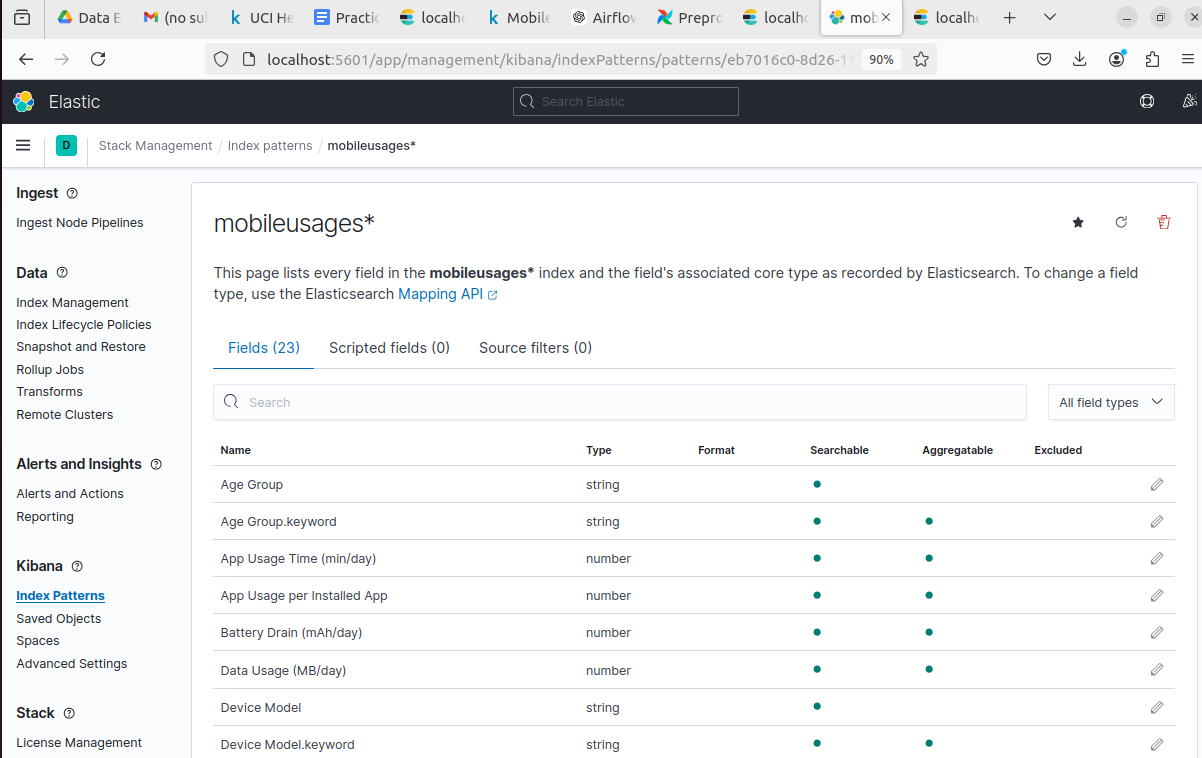
****

**Step 6 :**

Go to kibana -> go to ‘Manage’

In Kibana option select ‘Index Patterns’ -> search for ‘mobileusages’ -> click on ‘Next Step’ and create index

Click on three lines -> In kibana select ‘Discover’ -> From dropdown select ‘mobileusages’



**Step 7 :**

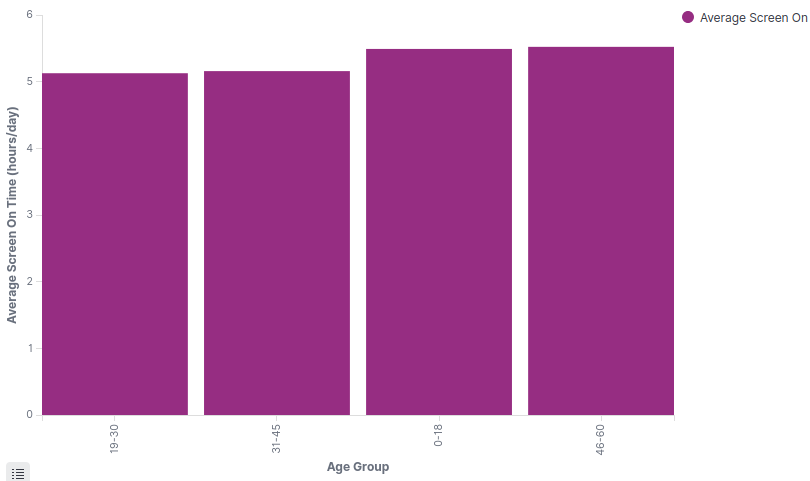
**Now next step is to create visualization that will help to create our dashboard.**

1) Average screen time distribution by different age groups

The graph shows **Average Screen On Time (hours/day)** across different **age groups**. The age groups include **0-18, 19-30, 31-45, and 46-60**. Each bar represents the average time spent with the screen turned on for each group.

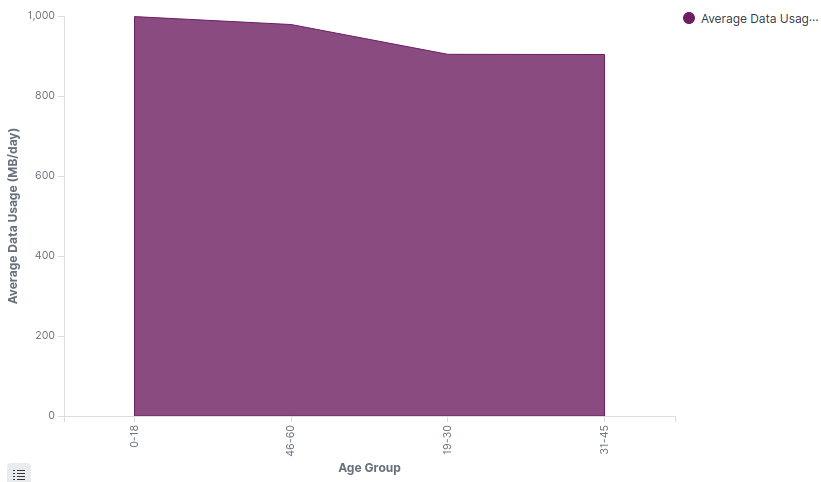
From the graph:

* The average screen-on time is fairly consistent across all age groups, around **5-6 hours per day**.
* There’s no significant difference in screen time between younger and older age groups, indicating similar screen usage patterns across all groups.

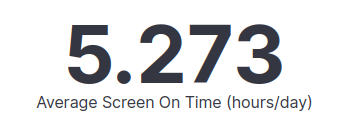


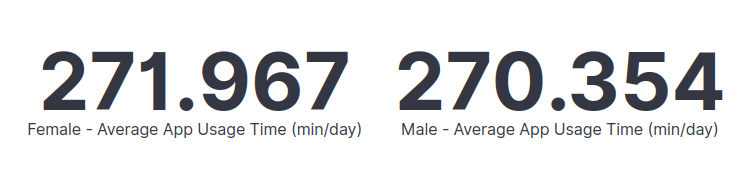
2) Average data usage by different age groups

This graph illustrates the **average data usage (MB/day)** across various **age groups**. It shows that younger age groups, particularly **0-18**, tend to use the most data, with approximately **1,000 MB/day**. As age increases, average data usage gradually decreases, with the **31-45** age group having the lowest usage, just below **800 MB/day**. Younger individuals use more data on average compared to older groups, with a notable decline in usage as age increases.



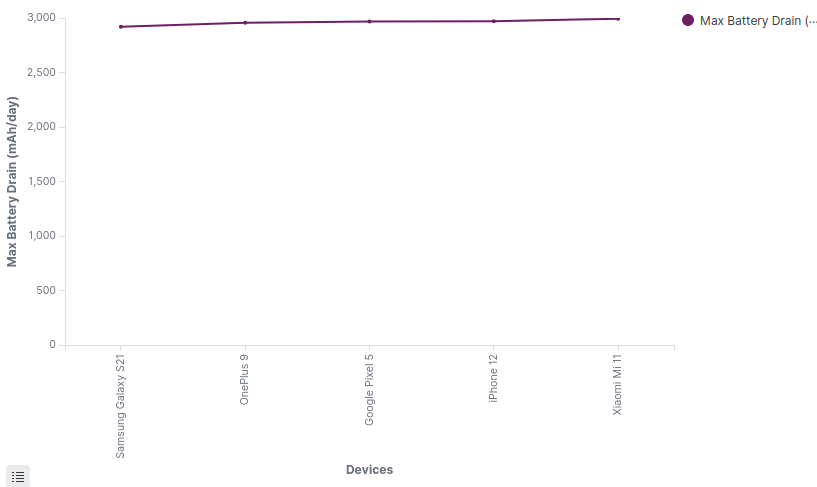
3) KPI - Created two KPIs





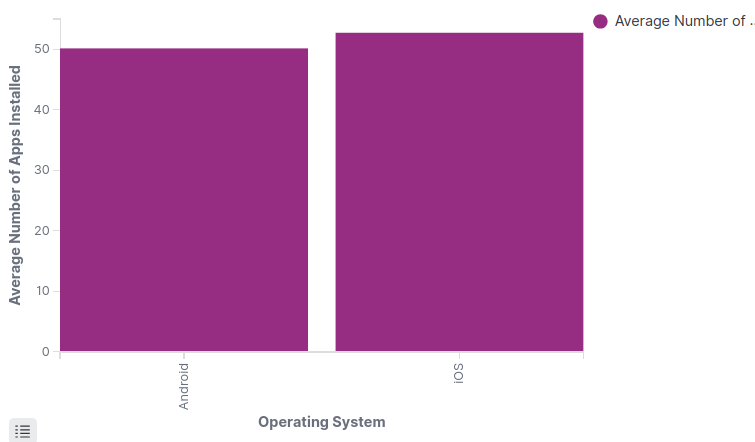
4) Line Chart - Shows Maximum battery drain in different devices

From the graph, we can see that the Samsung Galaxy S21, OnePlus 9, Google Pixel 5, iPhone 12, and Xiaomi Mi 11 have similar maximum battery drain, all around 3000 mAh/day. This suggests that these devices have comparable battery efficiency under heavy usage conditions.



5) Average number of apps installed in different operating system

From the graph, we can see that the average number of apps installed on Android and iOS devices is approximately the same. This suggests that users of both operating systems tend to install a similar number of apps on their devices.



**Step 8 : This is final Dashboard**

