CLOUD ANALYTICS AND DATA WAREHOUSE IMPLEMENTATION FOR A HISTORICAL DATASET

Abstract:

In today's digital age, telecom companies collect vast amounts of data from their customers, which can be analyzed to gain insights into customer behavior and improve services. However, the process of extracting, transforming, and loading (ETL) this data can be tedious and time-consuming. This project aims to automate the ETL process for user data from a telecom company using various tools and techniques such as Python, SQL, and cloud-based services. Automating the ETL process will reduce manual work, increase accuracy, and shorten processing times. The project will also focus on creating data visualizations using software such as Tableau, providing easily understandable insights into the data. The end goal is to establish an efficient and streamlined process for analyzing telecom churn data, and by storing the information in the cloud, present a dashboard that is easily accessible and comprehensible to viewers.

Introduction:

The importance of reporting and analysis in multinational corporations cannot be overstated as it is crucial for improving the customer base and customizing options as per user interests. Our project aimed to enhance the existing system of work in the industry through fine refinements at the edges, by procuring raw data from Kaggle and using it for reporting and analytics in Tableau.

Why this dataset:

This dataset is important because it enables telecom companies to gain insights into customer behavior, develop effective retention strategies, predict churn, improve customer satisfaction, evaluate business performance, and stay competitive in the market.

Data Structure:

The dataset used in our project is divided into seven tables, each with its own specific set of data. The incoming and outgoing tables contain information about all incoming and outgoing calls, respectively, including the total number of minutes used by each customer. The roaming table contains information about the cost of roaming for both incoming and outgoing calls. The user table contains details about each user, such as their phone number and the specific pack they are using. Additionally, the 3G and 2G tables provide information about the volume of data usage and the type of pack used by the user, whether it be monthly or sachet. Finally, the recharge table contains information about the user's last recharge and other related information. By utilizing these tables, we were able to extract valuable insights through reporting and analytics using tools such as Tableau.

Incoming:

idIncoming INT - This is a primary key for this entity, containing the id numbers.
spl_ic_mou INT- Minutes of usage special incoming calls.
isd_ic_mou INT- Minutes of usage of incoming isd calls.
ic_others INT- Other incoming calls.
std_ic_t2t_mou INT- Minutes of usage of incoming calls outside the circle within the same operator.
std_ic_t2m_mou INT- Minutes of usage of incoming calls outside the circle with another operator.
std_ic_t2f_mou INT-Minutes of usage of incoming calls outside the circle with fixed lines of the network.
std_ic_t2o_mou INT-Minutes of usage of incoming calls outside the circle with the other operator fixed line.

loc_ic_t2t_mou INT-Minutes of usage of incoming calls within the same telecom circle with the same operator.
loc_ic_t2m_mou INT-Minutes of usage of incoming calls within the same telecom circle with other operator mobile.
loc_ic_t2o_mou INT-Minutes of usage of incoming calls within the same telecom circle with another operator fixed line.

Roaming:

idRoaming INT- This is a primary key for this entity, containing the id numbers.
RoamingIC INT- Roaming on incoming calls
RoamingOG INT-Roaming on outgoing calls
Incoming_idIncoming INT- it's a foreign key for this entity, which connects with the Incoming table.
Outgoing_idOutgoing INT- it's a foreign key for this entity, which connects with the Outgoing table.
User_idUser INT- This is a foreign key for this entity, which connects with the User table.

User:

idUser INT- This is a primary key for this entity, containing the id numbers.
Onnet INT- All kinds of calls within the same operator network.
Offnet INT- All kinds of calls outside the same operator network.
NightPackUser INT- Scheme to use during specific night hours only.
Fbuser INT- Service scheme to avail services of Facebook and similar social networking sites.
Mobilenumber INT- User's phone number.

Outgoing:

idOutgoing INT- This is a primary key for this entity, containing the id numbers.
spl_og_mou INT- Minutes of usage of special outgoing calls.
isd_og_mou INT- Minutes of usage of special outgoing isd calls.
og_others INT- Other outgoing calls.
std_og_t2t_mou INT- Minutes of usage of outgoing calls outside the circle within the same operator.
std_og_t2m_mou INT- Minutes of usage of outgoing calls outside the circle with other operators.
std_og_t2f_mou INT- Minutes of usage of outgoing calls outside the circle with fixed lines of the network.
std_og_t2o_mou INT- Minutes of usage of outgoing calls outside the circle with the other operator fixed line.
loc_og_t2t_mou INT- Minutes of usage of outgoing calls within the same telecom circle with the same operator.
loc_og_t2m_mou INT- Minutes of usage of outgoing calls within the same telecom circle with other operator mobile.
loc_og_t2f_mou INT- Minutes of usage of outgoing calls within the same telecom circle with the same operator fixed line.
loc_og_t2o_mou INT- Minutes of usage of outgoing calls within the same telecom circle with another operator fixed line.

3g:

id3g INT- This is a primary key for this entity, containing the id numbers.
count_rech_3g INT - represents the count of times a user has recharged their mobile plan with 3G internet services.
vol_3g_mb INT- Mobile 3G internet usage volume.
arpu_3g INT- Average revenue per user on 3G network.
monthly_3g INT- Service with validity equivalent to a month.
sachet_3g INT- Service with validity smaller to a month.

Recharge:

idRecharge INT- This is a primary key for this entity, containing the id numbers.
max_rech_amt INT- maximum recharge amount.
date_of_last_rech DATETIME- Date of last recharge.
last_day_rch_amt INT- Amount of last recharge.
date_of_last_rech_data DATETIME- Date of last data recharge.
total_rech_data INT- Total data recharge.
max_rech_data INT- Maximum data recharge.

2g_id2g INT- it's a foreign key for this entity, which connects with the 2g table.
3g_id3g INT- it's another foreign key for this entity, which connects with the 3g table.
User_idUser INT- This is a foreign key for this entity, which connects to the user table.

2g:

id2g INT- This is a primary key for this entity, containing the id numbers.
count_rech_2g INT- represents the count of times a user has recharged their mobile plan with 2G internet services.
vol_2g_mb INT- Mobile 2G internet usage volume.
arpu_2g INT- Average revenue per user on 2G network.
monthly_2g INT- Service with validity equivalent to a month.
sachet_2g INT- Service with validity smaller to a month.

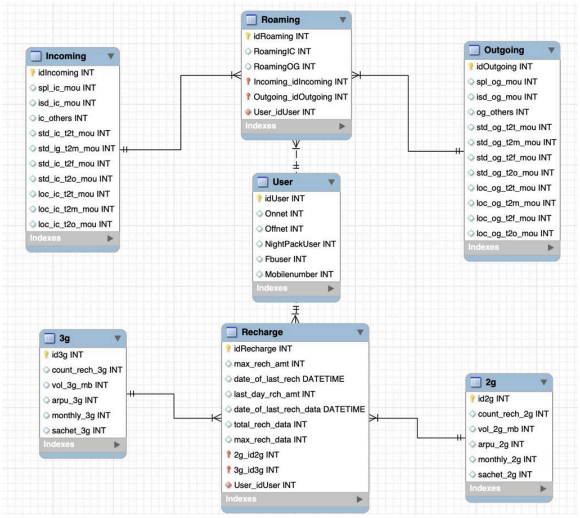


Fig: ER Diagram

Data Preparation and Exploration

The Telecom data was originally provided as a comma-separated file in CSV format. We uploaded this file to Google Bucket storage and then processed it through Google Bigquery to create a data warehouse. As part of this process, we divided the CSV file into seven tables, based on our Data Model.

To gain an initial understanding of the data structure, we used Python. We also connected Google Bigquery to Tableau for data visualization, and conducted various operations to obtain meaningful insights from the data. Overall, our efforts allowed us to effectively explore the data and gain valuable insights.

Project Flow:

For the implementation of cloud automation we have downloaded telecom dataset on cloud. The dataset consists of voice and usage of different customers for a period of a month in the year of 2014.

- **Overview of the Project**

  Our project involved procuring raw data from a large dataset obtained from Kaggle. We then dumped the files in CSV format into Google Bucket without any further processing. This dataset was cleaned and processed in the subsequent stages and used for analytics and reporting. To facilitate ETL processing, we decided to use the Apache Airflow tool and constructed the platform in Google with the support of Google for the warehouse.

- **Implementation of Apache Airflow Pipeline**

  To run the Apache Airflow pipeline, we utilized Google Cloud Composer as an instance creator to secure an output from the Airflow pipeline. We then wrote code/scripts to aid in the construction of the pipeline, which were saved as Python .py files. These scripts were imported into the Google Cloud warehouse to create corresponding .dags. The completion of .dags without errors indicates successful implementation of the Airflow pipelines. Upon completion of the .dags, the data was successfully transferred to Google BigQuery, which served as the warehouse in our implementation.

- **Reporting and Analytics using Tableau**

  Once the data was entered into the warehouse, we were able to perform reporting and analytics. We utilized Tableau for visualizations in analytics. The querying was not complex in this type of data as most corporations depend more on the analytical visualizations part. However, we also included prominence here for querying out important information and describing them in terms of reporting.

- **Business Insights and KPIs**

  The end-user finds the final reporting and analytics to be the most useful and uses the information retrieved to make decisions. We also utilized the concept of KPIs in our work, which benefits in defining business insights. The business insights are much proficient and helpful for the user as more information can be derived and are very intuitive in decision making. Overall, our project serves as a business-friendly service that provides valuable insights to end-users.

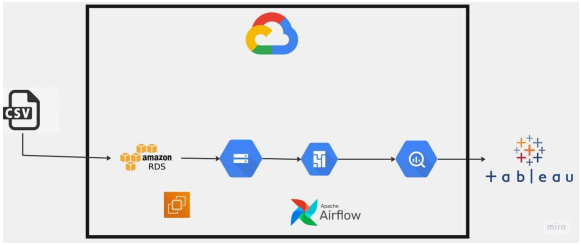
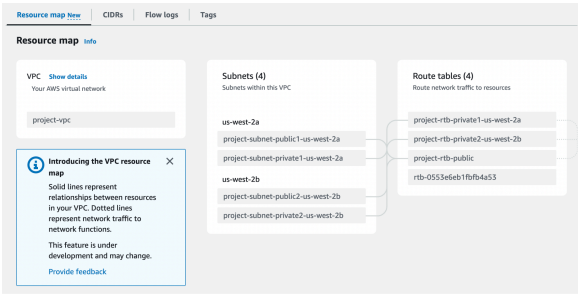


Fig: Cloud Architecture Workflow

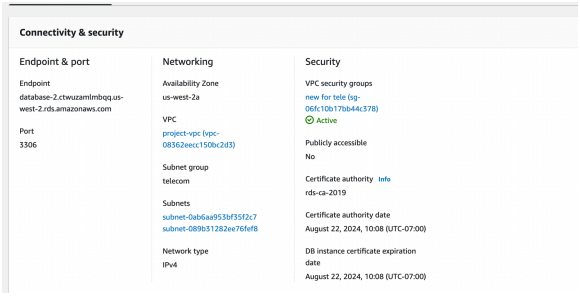
Methodology:

For the project implementation we have downloaded telecom churn dataset from kaggle that consists of revenue per user, voice and data per user that will be used for analysis further in the project.

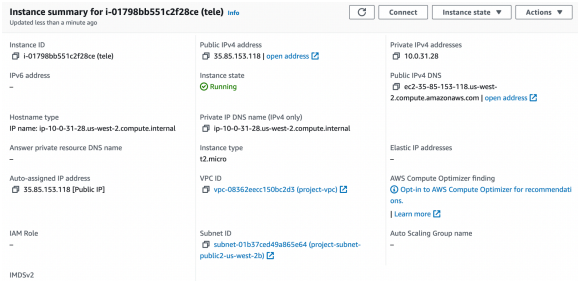
Virtual Private Cloud with 2 private networks over different regions have been created over AWS using AWS VPC service. The network has been created for a CIDR range of 10.0.0.0/16 which provides 65K different sub IP addresses for usage.



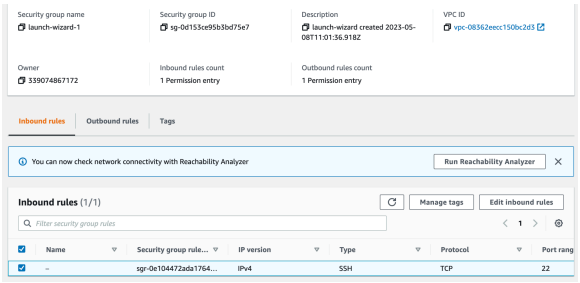
New subnet groups have been created for the 2 private addresses that have been established in the previous step for the RDP to have access over them. Using a free tier version we have created a RDS for the project. The created subnet groups are configured to the RDS.



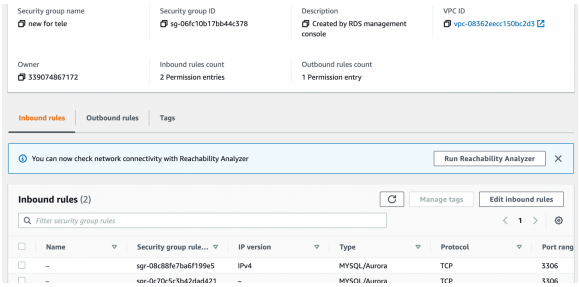
As the RDS is configured on a private IP it cannot be connected directly hence we have connected to it through a cloud virtual machine. EC2 service in AWS facilitates the creation of virtual machines. The private IP for the VM is in the range of provided IP address (10.0.0.0/16) that has been configured during start of the project.



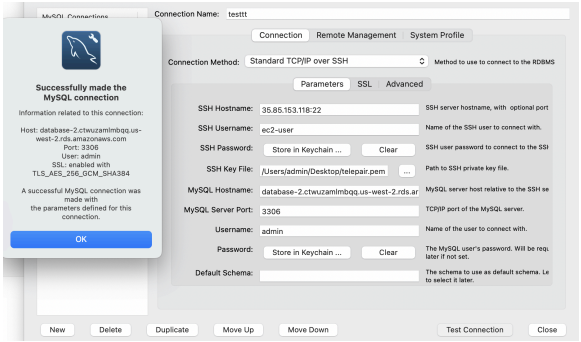
The security of the VM is configured to a particular IP such that others cannot publicly access the VM and the database.



The RDS has been configured such that the VM IP can access the RDS.

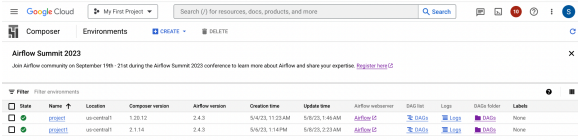


To connect over private VPC we have used VM IP as a bastion host. The VM's public IP is given in an inbound rule for RDS such that data from that particular IP is allowed to communicate with RDS. The privately configured subnets are configured for the VM to establish a link between RDS and EC. The security of the VM is set such that a only particular Ip can have access to it. The connection to the VM is made using the SSH key pair created on AWS for secured connection.

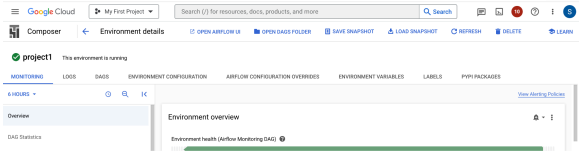


Now the dataset has first been cleaned of nulls from date columns and then loaded into RDS connected through MySQL workbench.

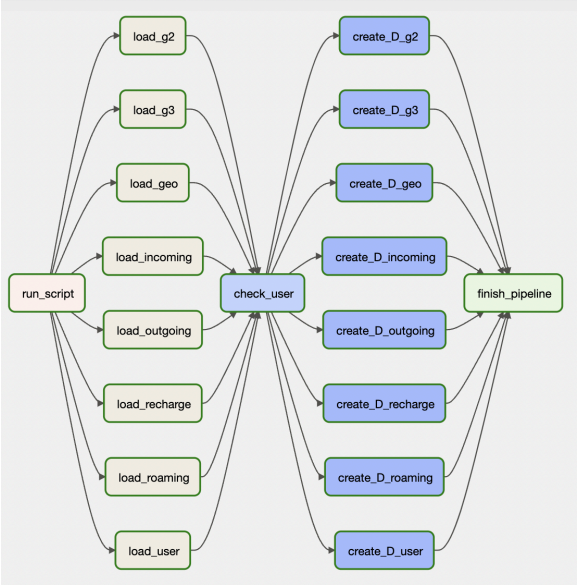
Google Cloud provides Cloud composer service which is built upon Apache Airflow to manage and orchestrate the airflow. The Cloud Composer provides 2 versions for the project we have chosen version 2, The version 2 has auto scaling automatically enabled while the version 1 does not, scaling needs to be done manually. The version 2 allows creation of 3 nodes for allocating computing resources to run multiple tasks and process simultaneously.



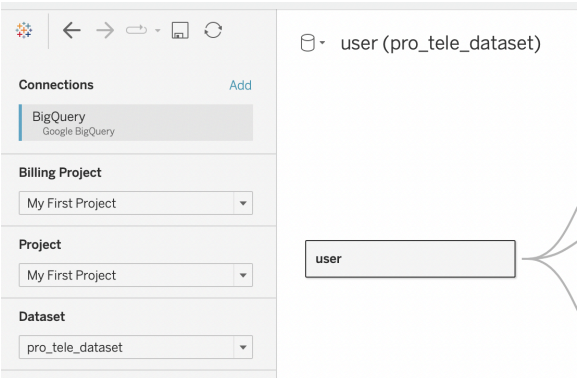
The instance provides Airflow UI and dag folder where scripts are stored.



The idea behind using airflow is to perform ETL operations on cloud over python scripts and schedule them according to the requirement. The below figure represents the workflow of the ETL process. The CSV file is accessed and data stored into big query using DAG which are written in python. Data is taken from RDS and stored into the cloud bucket as a CSV file using python library mysql.connector to connect to RDS database and google.cloud to store the retrieved data in google bucket. The retrieved CSV files are loaded into BIG Query by making use of airflow.providers.google.cloud packages. Once the DAG is success data is loaded into BIG Query.



Once data is loaded into a data warehouse it is accessible for users on cloud whenever required. Tableau is connected to the bg query to perform visualization and analyze the graphs



Queries:

1) **Retrieve the top 5 users with the highest average revenue per user on the 3G network, who have used 3G services with validity less than a month:**

```
SELECT id, AVG(arpu_3g_9) AS
avguser_revenue, sachet_3g_9
FROM
lithe-bazaar-385717.pro_tele_dataset.g3
WHERE sachet_3g_9 > 0
GROUP BY id, sachet_3g_9
ORDER BY AVG(arpu_3g_9) DESC
LIMIT 5;
```

```
1  SELECT id, AVG(arpu_3g_9) AS avguser_revenue, sachet_3g_9
2  FROM lithe-bazaar-385717.pro_tele_dataset.g3
3  WHERE sachet_3g_9 > 0
4  GROUP BY id, sachet_3g_9
5  ORDER BY AVG(arpu_3g_9) DESC
6  LIMIT 5;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	id	avguser_revenue	sachet_3g_9
1	24356	13884.31	4
2	12014	3519.19	1
3	6158	2512.91	36
4	623	2339.38	49
5	16127	1899.28	8

2) **Query for getting top 100 recharge ids, last recharge amount and last recharge date.**

```
SELECT ID as Recharge_id,
last_day_rch_amt_9 as
Last_recharge_amount, date_of_last_rech_9
as Last_recharge_date
FROM
lithe-bazaar-385717.pro_tele_dataset.rechar
ge
ORDER BY last_day_rch_amt_9 DESC
LIMIT 100;
```

```
1  SELECT ID as Recharge_id, last_day_rch_amt_9 as Last_recharge_amount, date_of_last_rech_9 as Last_recharge_date
2  FROM lithe-bazaar-385717.pro_tele_dataset.recharge
3  ORDER BY last_day_rch_amt_9 DESC
4  LIMIT 100;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH | PREVIEW

Row	Recharge_id	Last_recharge_amount	Last_recharge_date
1	19858	3000	2014-09-26
2	22701	1699	2014-09-04
3	21106	1699	2014-09-04
4	20762	1555	2014-09-19
5	25621	1555	2014-09-26
6	21906	1555	2014-09-08
7	1314	1555	2014-09-27
8	9572	1555	2014-09-26
9	17313	1500	2014-09-30
10	25005	1500	2014-09-30

3) **Query for getting the top 10 states with the highest number of users.**

```
SELECT State, COUNT(*) AS num_users
FROM
lithe-bazaar-385717.pro_tele_dataset.geo
GROUP BY state
ORDER BY num_users DESC
LIMIT 10;
```

```
SELECT State, COUNT(*) AS num_users
FROM lithe-bazaar-385717.pro_tele_dataset.geo
GROUP BY state ORDER BY num_users DESC
LIMIT 10;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETA

Row	State	num_users
1	Telangana	969
2	Sikkim	966
3	Arunachal Pradesh	960
4	Uttar Pradesh	957
5	Punjab	955
6	Gujarat	955
7	Bihar	950
8	Tamil Nadu	949
9	Karnataka	944
10	Tripura	937

4) **Retrieve the average revenue generated by 3g users who make calls within the same network (onnet), have a call minutes greater than 500, and are subscribed to the night pack service.**

SELECT g.ID, g.arpu_3g_9 AS avgrevenue_user,u.fb_user_9,u.Onnet_mou_9
FROM
lithe-bazaar-385717.pro_tele_dataset.g3 g
JOIN
lithe-bazaar-385717.pro_tele_dataset.user u
ON g.ID = u.ID
WHERE u.fb_user_9 = 1 AND
u.Onnet_mou_9 >= 500
ORDER BY avgrevenue_user DESC;

```
SELECT g.ID, g.arpu_3g_9 AS avgrevenue_user,u.fb_user_9,u.Onnet_mou_9
FROM lithe-bazaar-385717.pro_tele_dataset.g3 g
JOIN lithe-bazaar-385717.pro_tele_dataset.user u
ON g.ID = u.ID
WHERE u.fb_user_9 = 1 AND u.Onnet_mou_9 >= 500
ORDER BY avgrevenue_user DESC;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTI

Row	ID	avgrevenue_use	fb_user_9	Onnet_mou_9
1	12014	3519.19	1	509.04
2	24288	2929.18	1	1003.94
3	6158	2512.91	1	751.63
4	20221	2358.18	1	937.16
5	14440	1798.93	1	686.73
6	3377	1706.76	1	929.89
7	17524	1579.27	1	1094.03
8	20059	1570.16	1	1012.76

5) **Retrieve the total minutes of usage of incoming and outgoing calls for each user.**

SELECT u.id,
SUM(r.roam_ic_mou_9) AS total_roaming_incoming_mou,
SUM(r.roam_og_mou_9) AS total_roaming_outgoing_mou
FROM
lithe-bazaar-385717.pro_tele_dataset.roaming r
JOIN
lithe-bazaar-385717.pro_tele_dataset.user u
ON
r.id = u.id
WHERE r.roam_ic_mou_9 > 0 AND
r.roam_og_mou_9 > 0
GROUP BY u.id;

```
SELECT u.id,
SUM(r.roam_ic_mou_9) AS total_roaming_incoming_mou,
SUM(r.roam_og_mou_9) AS total_roaming_outgoing_mou
FROM
lithe-bazaar-385717.pro_tele_dataset.roaming r
JOIN lithe-bazaar-385717.pro_tele_dataset.user u ON
r.id = u.id
WHERE r.roam_ic_mou_9 > 0 AND r.roam_og_mou_9 > 0
GROUP BY u.id;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH | PREVIEW

Row	id	total_roaming_incoming_mou	total_roaming_outgoing_mou
1	24901	2.0	38.91
2	12894	4.0	6.73
3	25690	4.25	1.58
4	24945	4.25	14.2
5	1945	4.5	8.18
6	740	4.75	0.95
7	24779	5.5	4.05

Results per page: 50 1 – 50 of 3124

6) **Retrieve the top 10 users with the highest 2G internet usage volume:**

SELECT id as user_id, vol_2g_mb_9
FROM
lithe-bazaar-385717.pro_tele_dataset.g2
ORDER BY vol_2g_mb_9 DESC
LIMIT 10;

```
SELECT id as user_id, vol_2g_mb_9
FROM lithe-bazaar-385717.pro_tele_dataset.g2
ORDER BY vol_2g_mb_9 DESC
LIMIT 10;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	user_id	vol_2g_mb_9
1	10043	8993.95
2	1853	8680.13
3	10841	8525.19
4	24039	7050.48
5	11400	5554.25
6	6278	5503.25
7	9690	5458.13
8	10779	4540.25
9	22274	4448.7
10	21648	4427.14

7) Retrieve the top 10 users with the highest 3G internet usage volume:

SELECT id as user_id, vol_3g_mb_9
FROM
lithe-bazaar-385717.pro_tele_dataset.g3
ORDER BY vol_3g_mb_9 DESC
LIMIT 10;

```
SELECT id as user_id, vol_3g_mb_9
FROM lithe-bazaar-385717.pro_tele_dataset.g3
ORDER BY vol_3g_mb_9 DESC
LIMIT 10;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DI

Row	user_id	vol_3g_mb_9
1	7143	39221.27
2	23428	26857.04
3	12053	19851.32
4	12203	19794.09
5	14440	19604.01
6	20221	18627.51
7	16905	18247.22
8	24288	17152.26
9	6158	16718.42
10	21906	16326.23

8) Retrieve the average number of minutes of usage for onnet and offnet calls.

SELECT AVG(Onnet_mou_9) AS avg_onnet_calls, AVG(Offnet_mou_9) AS avg_offnet_calls FROM lithe-bazaar-385717.pro_tele_dataset.user;

```
SELECT AVG(Onnet_mou_9) AS avg_onnet_calls, AVG(Offnet_mou_9) AS avg_offnet_calls
FROM lithe-bazaar-385717.pro_tele_dataset.user;
```

Query results | SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH | PREVIEW

Row	avg_onnet_calls	avg_offnet_calls
1	139.064988...	222.106691...

9) Retrieve the top 5 users with the highest average revenue per user on the 2G network, who have used 2G services with validity less than a month:

SELECT id, AVG(arpu_2g_9) AS avguser_revenue, sachet_2g_9
FROM
lithe-bazaar-385717.pro_tele_dataset.g2
WHERE sachet_2g_9 > 0
GROUP BY id, sachet_2g_9
ORDER BY AVG(arpu_2g_9) DESC
LIMIT 5;

```
SELECT id, AVG(arpu_2g_9) AS avguser_revenue, sachet_2g_9
FROM lithe-bazaar-385717.pro_tele_dataset.g2
WHERE sachet_2g_9 > 0
GROUP BY id, sachet_2g_9
ORDER BY AVG(arpu_2g_9) DESC
LIMIT 5;
```

Query results

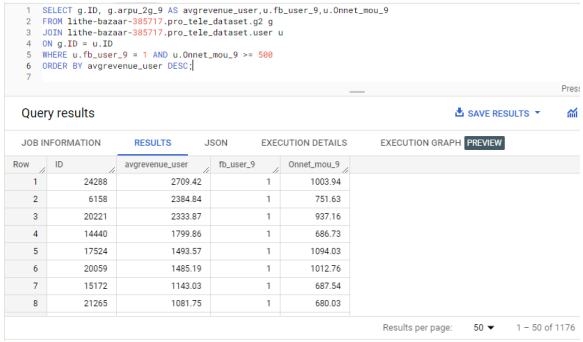
JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXE

Row	id	avguser_revenue	sachet_2g_9
1	6158	2384.84	2
2	623	2123.86	35
3	23994	1635.86	1
4	17524	1493.57	1
5	4306	1300.34	1

10) Retrieve the average revenue generated by 2g users who make calls within the same network (onnet), have a call minutes greater than 500, and are subscribed to the night pack service.

SELECT g.ID, g.arpu_2g_9 AS avgrevenue_user,u.fb_user_9,u.Onnet_mou_9

```
FROM
lithe-bazaar-385717.pro_tele_dataset.g2 g
JOIN
lithe-bazaar-385717.pro_tele_dataset.user u
ON g.ID = u.ID
WHERE u.fb_user_9 = 1 AND
u.Onnet_mou_9 >= 500
ORDER BY avgrevenue_user DESC;
```



Query results

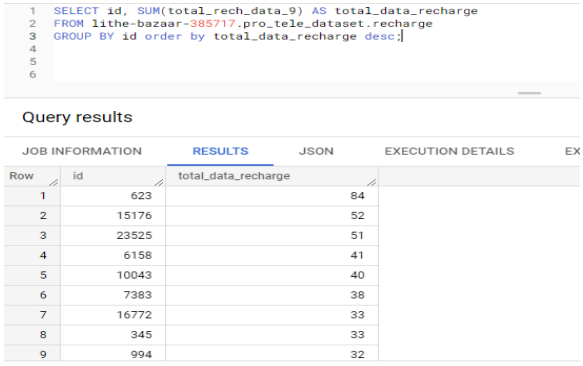
JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	ID	avgrevenue_user	fb_user_9	Onnet_mou_9
1	24288	2709.42	1	1003.94
2	6158	2384.84	1	751.63
3	20221	2333.87	1	937.16
4	14440	1799.86	1	686.73
5	17524	1493.57	1	1094.03
6	20059	1485.19	1	1012.76
7	15172	1143.03	1	687.54
8	21265	1081.75	1	680.03

Results per page: 50 1 – 50 of 1176

11) To retrieve the total amount of data recharged for each user:

```
SELECT id, SUM(total_rech_data_9) AS
total_data_recharge
FROM
lithe-bazaar-385717.pro_tele_dataset.rechar
ge
GROUP BY id order by total_data_recharge
desc;
```



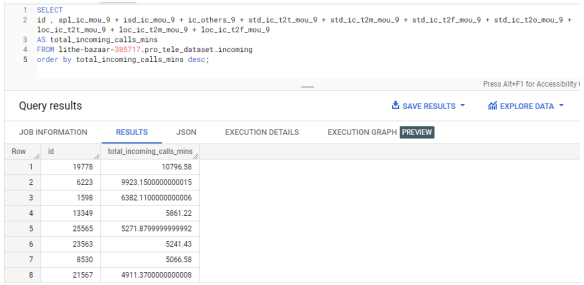
Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	id	total_data_recharge
1	623	84
2	15176	52
3	23525	51
4	6158	41
5	10043	40
6	7383	38
7	16772	33
8	345	33
9	994	32

12) Query to get the total number of incoming call minutes per user:

```
SELECT id , spl_ic_mou_9 + isd_ic_mou_9
+ ic_others_9 + std_ic_t2t_mou_9 +
std_ic_t2m_mou_9 + std_ic_t2f_mou_9 +
std_ic_t2o_mou_9 + loc_ic_t2t_mou_9 +
loc_ic_t2m_mou_9 + loc_ic_t2f_mou_9 AS
total_incoming_calls_mins FROM
lithe-bazaar-385717.pro_tele_dataset.incomi
ng order by total_incoming_calls_mins desc;
```



Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	id	total_incoming_calls_mins
1	19778	10796.58
2	6223	9923.1500000000015
3	1598	6382.1100000000006
4	13349	5861.22
5	25565	5271.8799999999992
6	23563	5241.43
7	8530	5066.58
8	21567	4911.3700000000008

13) Query to get the top 3 total number of incoming calls operator network:

```
SELECT
  CASE
    WHEN std_ic_t2t_mou_9 > 0 THEN
'Same Operator'
    WHEN std_ic_t2m_mou_9 > 0 THEN
'Other Operator Mobile'
    WHEN std_ic_t2f_mou_9 > 0 THEN
'Fixed Line'
    WHEN std_ic_t2o_mou_9 > 0 THEN
'Other Operator Fixed Line'
    WHEN loc_ic_t2t_mou_9 > 0 THEN
'Same Circle'
    WHEN loc_ic_t2m_mou_9 > 0 THEN
'Other Operator Mobile'
    WHEN loc_ic_t2f_mou_9 > 0 THEN
'Other Operator Fixed Line'
    WHEN isd_ic_mou_9 > 0 THEN 'ISD
calls'
    WHEN spl_ic_mou_9 > 0 THEN 'Special
calls'
    WHEN ic_others_9 > 0 THEN 'Other
calls'
```

END AS incoming_network,COUNT(*) AS total_incoming_calls
FROM lithe-bazaar-385717.pro_tele_dataset.incoming
GROUP BY incoming_network order by total_incoming_calls desc LIMIT 3;

```
SELECT
  CASE
    WHEN std_ic_t2t_mou_9 > 0 THEN 'Same Operator'
    WHEN std_ic_t2m_mou_9 > 0 THEN 'Other Operator Mobile'
    WHEN std_ic_t2f_mou_9 > 0 THEN 'Fixed Line'
    WHEN std_ic_t2o_mou_9 > 0 THEN 'Other Operator Fixed Line'
    WHEN loc_ic_t2t_mou_9 > 0 THEN 'Same Circle'
    WHEN loc_ic_t2m_mou_9 > 0 THEN 'Other Operator Mobile'
    WHEN loc_ic_t2f_mou_9 > 0 THEN 'Other Operator Fixed Line'
    WHEN isd_ic_mou_9 > 0 THEN 'ISD calls'
    WHEN spl_ic_mou_9 > 0 THEN 'Special calls'
    WHEN ic_others_9 > 0 THEN 'Other calls'
  END AS incoming_network,COUNT(*) AS total_incoming_calls
FROM lithe-bazaar-385717.pro_tele_dataset.incoming
GROUP BY incoming_network order by total_incoming_calls desc LIMIT 3;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECU

Row	incoming_network	total_incoming_calls
1	Same Operator	11307
2	Other Operator Mobile	7505
3	Same Circle	4768

14) Retrieve the top 5 users who have the highest difference between their outgoing call minutes within the same operator network (loc_og_t2t_mou) and outside the operator network (std_og_t2m_mou).

SELECT
id, SUM(loc_og_t2t_mou_9) AS loc_og_t2t_mou_sum,
SUM(std_og_t2m_mou_9) AS std_og_t2m_mou_sum,
(SUM(loc_og_t2t_mou_9) - SUM(std_og_t2m_mou_9)) AS difference
FROM lithe-bazaar-385717.pro_tele_dataset.outgoing
GROUP BY id
ORDER BY difference DESC
LIMIT 5;

```
SELECT
  id, SUM(loc_og_t2t_mou_9) AS loc_og_t2t_mou_sum, SUM(std_og_t2m_mou_9) AS std_og_t2m_mou_sum, (SUM(loc_og_t2t_mou_9) - SUM
  (std_og_t2m_mou_9)) AS difference
FROM lithe-bazaar-385717.pro_tele_dataset.outgoing
GROUP BY id
ORDER BY difference DESC
LIMIT 5;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH | PREVIEW

Row	id	loc_og_t2t_mou_sum	std_og_t2m_mou_sum	difference
1	654	10389.24	0.0	10389.24
2	16783	8618.93	293.89	8325.04
3	7131	7781.34	256.01	7525.33
4	24009	5768.38	0.0	5768.38
5	23596	5678.41	0.0	5678.41

15) Retrieve the top 10 users with the highest total outgoing call minutes outside the operator network (std_og_t2m_mou).

SELECT
id,SUM(std_og_t2m_mou_9) AS total_outgoing_mins
FROM lithe-bazaar-385717.pro_tele_dataset.outgoing
GROUP BY user_id
ORDER BY total_outgoing_mins DESC
LIMIT 10;

```
SELECT
id,SUM(std_og_t2m_mou_9) AS total_outgoing_mins
FROM lithe-bazaar-385717.pro_tele_dataset.outgoing
GROUP BY id
ORDER BY total_outgoing_mins DESC
LIMIT 10;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DET

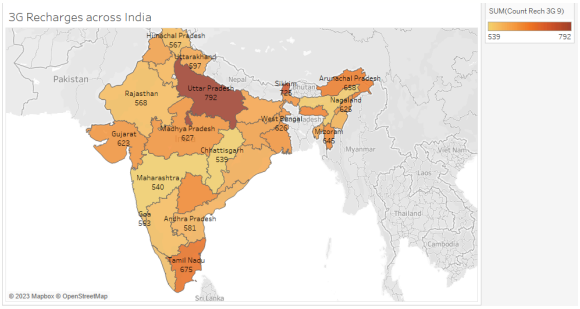
Row	id	total_outgoing_mins
1	23991	10223.43
2	18686	6067.73
3	15253	5805.06
4	24314	5540.89
5	12614	5457.41
6	7226	4843.64
7	13361	4738.91
8	13153	4617.19
9	11136	4356.79
10	25595	4287.89

Data Visualization:

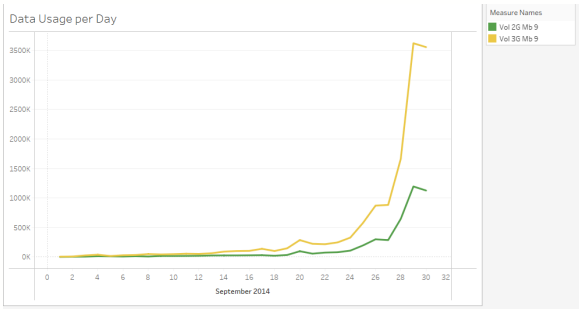
Tableau is a powerful business intelligence and data visualization tool that allows us to analyze and visualize data in a variety of ways. By utilizing the

functionality of Tableau, we were able to gain valuable insights from our data and present our findings in a meaningful and actionable way.

1. 3G Recharges done across India.

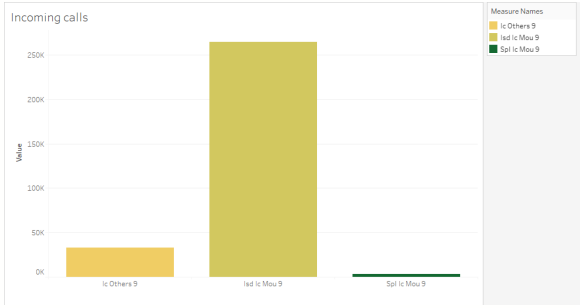


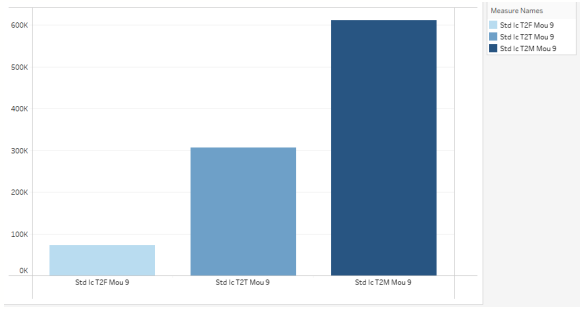
2. Total Data Usage throughout the Month



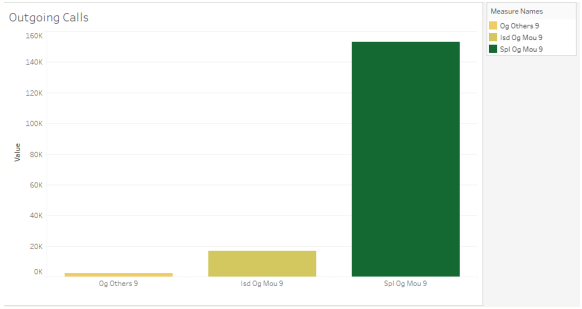
3. Usage of 2G and 3G network

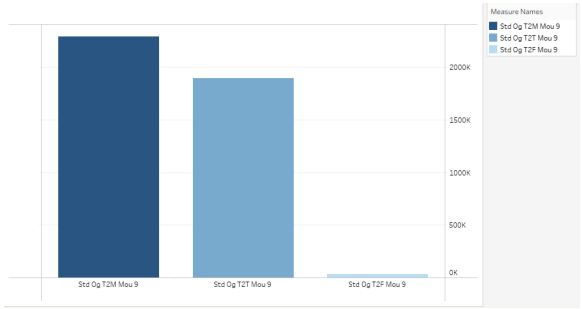
4. Analysis of Incoming calls



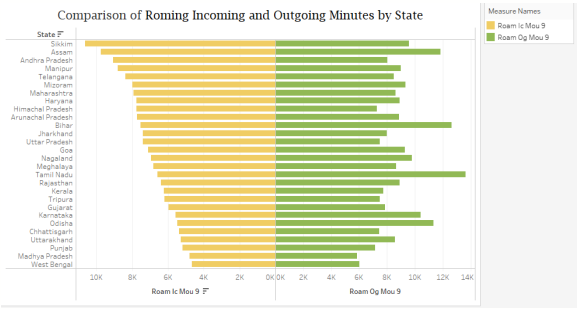


5. Analysis of Outgoing Calls

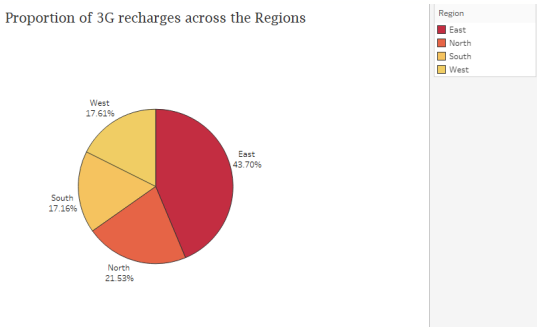




6. Comparison of Roaming Incoming and Outgoing Minutes by State.



7. Proportion of 3G Recharges across the region.



KPI Dashboards:

A KPI dashboard is a graphical representation of an organization's key performance indicators (KPIs) that provides a quick and easy-to-understand summary of the most critical metrics used to measure the organization's progress and performance towards achieving its goals and objectives. In more general terms, KPI means key performance indicators, the business managers make business intuitive decisions through these factors. We from our visualizations try coming up with detailed concepts. The KPIs we came up with during the analysis part are as follows.

1. Incoming calls summaries
2. Outgoing calls summaries
3. Data usage management
4. Differential analysis of data usage between 3g and 2g
5. Recharge factors and amounts across different states

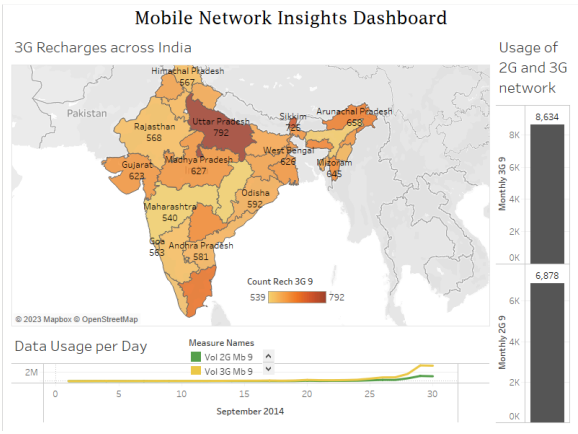


Fig: Mobile Network Insights Dashboard

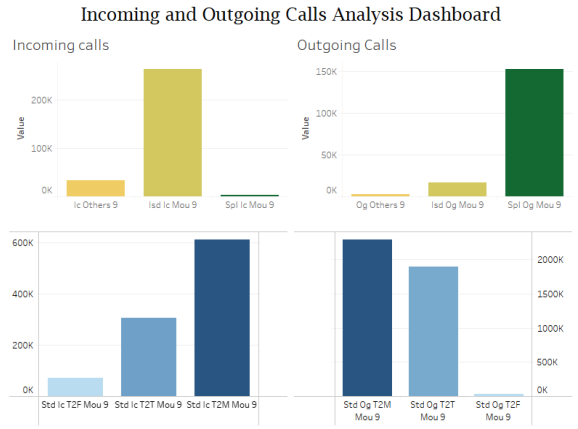


Fig: Incoming and Outgoing Call Analysis

Conclusion:

We have observed from the above that we are automating most of the service side hassle of the multinational corporations. The process from procuring the dataset to securing meaningful business insights through reporting and analytics has been very well designed and structured through a series of processes using the concepts of relational databases and cloud technologies, even the security measures

are being considered and taken care of by VPC network establishments. We can conclude by saying that this project opens a wide scope in deriving useful information from raw data through an automated process which benefits companies in avoiding human errors and reducing a huge financial crop down. Usage of all these concepts enhance the productivity of the insights and aid companies in taking intuitive decisions.