

## Plantix Case Study

### Q1: Which KPIs would you use to measure the performance of our app?

- The following KPIs can be used to measure the performance:
1. **User Growth:** As most of the Plantix's feature, utilities, and offerings are available through the app, the user growth and user registration can be first KPI, as this would tell us that how many users have started using Plantix's app in comparison in each set of periods. We can use the following mathematical approach for percentage growth in user base:

$$\text{User in the beginning} - \text{User at the End} / \text{User in the beginning} * 100$$

2. **Downloads:** The app downloads can also present important insights for us. Basically, we can use this metric in conjunction with the Plantix's marketing plan. Let's say if we do A/B testing and based on promotion idea, we can compare the percentage increase in downloads of app from marketplace while running Idea A and Idea B.
3. **Retention Rate:** The retention rate is one of metrics which can tell us about consumer/user stickiness. It can provide us with the utility or the value which user is able to find via the app and it can tell us about how often the user is using the app and how consistent he is for the usage. This can be used in multiple ways like if the app has better retention rate, we can get the idea of how properly the new features, which are more value in the app is being seen from the user's PoV.
4. **Sessions:** While tracking the retention rate, it also opens to track the session time. Basically, the longer the session, the more value our user can derive from the app. These two metrics (retention rate and session) can help in multiple way. In a way if a particular user is repeat user and is having longer app session, then he becomes one of the valuable customers and the marketing team can reach out for premium subscription or personalized approach for adding more value.
5. **Active users:** The app/platform's success can be defined by the daily and monthly active user. While getting this metrics, we can also figure out the new logins or new users registered in each period.
6. **Time taken to make the first purchase:** With the data being collated, we can figure the average number of time taken by a user in making the first purchase, this enables us to figure out the user decision making process and it will help us in designing our welcome offers and program our app to add more value right from the first session. We can use the following formula

$$\text{Date of First purchase} - \text{Date of registration}$$

7. **Churning Rate:** The churning rate is another metrics, which helps us with finding out the user who have stopped using the app out of the total number of users acquired. Mathematically, we can represent and use the metrics in the following way:

**(Customers Left/ Customers at starting + New customers acquired) \* 100**

**Q2: Prepare a report regarding our growth between the 2 years. Please try to answer the following questions:**

- a) Did our business grow?** : Yes, the business did make a significant progress i.e., more than 90%. The following SQL code can be used to get the answer.

```
1. Create TEMP Table Output AS
2. SELECT * from sales_orders
3. JOIN sales_orders_items ON sales_orders_items.fk_order_id = Sales_orders.Order_id
4. WHERE sales_orders_items.Order_quantity_accepted <> 0
5.
6. SELECT * from output
7.
8. Create TEMP TABLE Yearly_revenue AS
9. SELECT rate, ordered_quantity, Rate* Ordered_Quantity AS Revenue, creation_time,
10. EXTRACT(YEAR FROM creation_time) AS Year_Sales, SUM(Rate* Ordered_Quantity)
11. OVER (Partition by (EXTRACT(YEAR FROM creation_time))) as Yearly_Revenue
12. From output
13.
14. SELECT DISTINCT (yearly_Revenue) from yearly_revenue
15. ORDER BY Yearly_revenue
```

- b) Does our app perform better now?** : Yes, the apps performance has increased basis the fact that yearly revenue has increased, which translates that the supplier are now ordering more from app i.e., the data shows significant growth in number of orders received in 2022

- c) Did our user base grow?** Yes, in comparison between the year 2021 and 2022. The user base for the app has increased by close to 20%. The code to fetch the details is:

```
1. SELECT COUNT(DISTINCT (USER_ID)) FROM login_logs
2. WHERE EXTRACT(YEAR FROM Login_time) = '2021'
3.
4. SELECT COUNT(DISTINCT (USER_ID)) FROM login_logs
5. WHERE EXTRACT(YEAR FROM Login_time) = '2022'
```

**Q3: What are our top-selling products in each of the two years? Can you draw some insight from this?**

```
1. Create TEMP Table Top_Product_21 AS
2. SELECT fk_product_id, COUNT(*) FROM (SELECT * FROM Sales_orders_items
3. JOIN Sales_orders ON sales_orders.order_id = Sales_orders_items.fk_order_id
4. Where Order_quantity_accepted <> 0 and sales_order_status LIKE 'Shipped'
5. AND EXTRACT(YEAR FROM creation_time) = '2021') x
6. GROUP BY Fk_product_ID
7. ORDER BY COUNT(*) DESC
8. LIMIT 5
9.
10. Create TEMP Table Top_Product_22 AS
11. SELECT fk_product_id, COUNT(*) FROM (SELECT * FROM Sales_orders_items
12. JOIN Sales_orders ON sales_orders.order_id = Sales_orders_items.fk_order_id
13. Where Order_quantity_accepted <> 0 and sales_order_status LIKE 'Shipped'
14. AND EXTRACT(YEAR FROM creation_time) = '2022') x
15. GROUP BY Fk_product_ID
```

```
16. ORDER BY COUNT(*) DESC
17. LIMIT 5
```

**Logic:** Considered the products where the sales has been made and Order is accepted and shipped (Top 5 products for each year).

#### Top selling products

2021 Products	2022 Products
10235 (280)	8444 (233)
1548 (146)	3610 (198)
8219 (133)	10235 (184)
9925 (123)	8425 (153)
8444 (114)	1041 (142)

#### Insights:

- Products with id 10235 and 8444 has performed consistently in terms quantity sold as these two products are in Top 5 for both the years.
- The demand for the 8425 is on increasing trend as in the year the products was on 6<sup>th</sup> position and this could mean that retailers are getting increasing demand from end users for this products and hence presents a potential for increasing supplies

**Q4: Looking at July 2021 data, what do you think is our biggest problem and how would you recommend fixing it?**

```
1. Create TEMP TABLE Buyer_data As
2. Select * FROM Sales_orders_items
3. JOIN Sales_orders ON sales_orders.order_id = Sales_orders_items.fk_order_id
4. WHERE EXTRACT(YEAR FROM creation_time) = '2021'
5.
6. --Point 1
7. SELECT Sales_order_status, Count(*) FROM (SELECT * FROM Sales_orders_items
8. JOIN Sales_orders ON sales_orders.order_id = Sales_orders_items.fk_order_id
9. WHERE EXTRACT(YEAR FROM creation_time) = '2021')X
10. GROUP BY Sales_order_status
11. ORDER BY COUNT(*) DESC
12.
13. --Point 2 - Repeat orders or Customer Retention
14.
15. SELECT Fk_Buyer_id, Count(*) FROM (SELECT * From buyer_data
16. WHERE Sales_order_status = 'Shipped') x
17. GROUP BY fk_Buyer_ID
18. HAVING COUNT(*) >= '6'
19. ORDER BY COUNT (*) DESC
```

- In the time period July 2021, the total number of rejected orders is 35% more than the order which were shipped. So, this has resulted in a loss of revenue for the company. I.e., Company can focus on improving the order fulfillment rate

- Out of the total buyers in July 2021 only 150 buyers have placed orders more than 5 times which means that company needs to focus more on customer retention and can present unique offers and preposition to make the repeat purchase more lucrative. This insight can also mean that the majority of revenue of company comes from concentrated buyers hence, company can also look to acquire new customers. ( Considered the shipped orders only for the precise analysis)

**Q5: Does the login frequency affect the number of orders made?**

```

1. CREATE TEMP TABLE Login_Count AS
2. SELECT User_id, Count(*) AS Total_logins FROM Login_logs
3. Group BY user_ID
4. ORDER BY COUNT(*) DESC
5.
6. CREATE TEMP TABLE Order_Counts AS
7. SELECT fk_buyer_id, COUNT(*) as Total_orders_made FROM Sales_orders
8. GROUP BY fk_buyer_id
9. ORDER BY Count(*) DESC
10.
11. SELECT * FROM Login_count
12. JOIN Order_counts ON order_counts.fk_buyer_id = Login_count.User_ID
13. ORDER BY total_logins DESC

```

- As per the analysis there is not direct relation exists between the total frequency of login and total number of orders made. The below given output confirms that there is no direct relation exists between the two metrics.

	user_id integer	total_logins bigint	fk_buyer_id integer	total_orders_made bigint
1	63652	3868	63652	2
2	209276	2471	209276	3
3	51192	2097	51192	4
4	215084	1293	215084	1
5	43180	1286	43180	4
6	216168	1273	216168	3
7	62468	1268	62468	1
8	218084	1255	218084	7
9	98452	1190	98452	11
10	87136	1164	87136	18
11	38752	1134	38752	17
12	97116	1107	97116	17
13	196852	1081	196852	2
14	197680	1062	197680	1
15	83720	997	83720	5
16	57952	971	57952	16

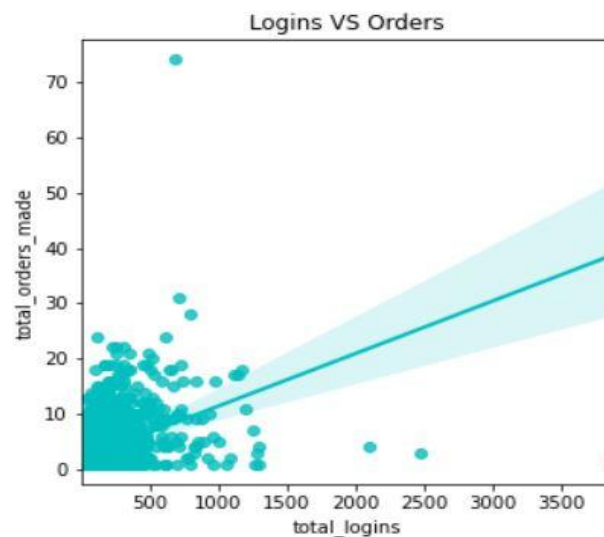
- Python Code for Regression to confirm, if the above data has any Relation between them

```

1. import numpy as np
2. import pandas as pd
3. import matplotlib.pyplot as plt
4. import seaborn as sns
5. import psycopg2
6.
7. conn = psycopg2.connect("dbname=Plantix user=postgres password=12345678")
8.
9. cursor = conn.cursor()
10.
11. cursor.execute("SELECT * FROM Login_logs")
12. cursor.fetchmany(5)
13.
14. cursor.execute("Create TEMP TABLE LOGIN_COUNT AS SELECT User_id, Count(*) AS Total_logins
    FROM login_logs Group BY user_ID ORDER BY COUNT(*) DESC")
15. cursor.execute("SELECT * FROM Login_count")
16. cursor.fetchmany(5)
17.
18. cursor.execute("CREATE TEMP TABLE Order_Counts AS SELECT fk_buyer_id, COUNT(*) as
    Total_orders_made FROM Sales_orders GROUP BY fk_buyer_id ORDER BY Count(*) DESC")
19. cursor.execute("SELECT * FROM Order_counts")
20. cursor.fetchmany(5)
21.
22. cursor.execute("Create TEMP TABLE Combined_Data AS SELECT * FROM Login_count JOIN
    Order_counts ON order_counts.fk_buyer_id = Login_count.User_ID ORDER BY total_logins DESC")
23. cursor.execute("SELECT * FROM Combined_data")
24. cursor.fetchmany(5)
25. Combined_Data = pd.read_sql("SELECT * FROM Combined_data", con= conn)
26. Combined_Data.head()
27.
28. plt.figure(figsize=(10,4))
29. sns.regplot(data = Combined_Data, y = "total_orders_made", x = "total_logins", color =
    "c").set(title = "Logins VS Orders")

```

- Output :



- As can be seen there is no defined relation between the two data points and hence the above graph confirms our understanding that there is no direct relation between the number of orders and login frequency.