Does failing increase risk of churn?

We would get the answer by finding following factors -

* Calculate no. of players failing and immediately quitting
* Calculate no. of players failing on last session they played
* Calculate no. of players failing on last stage they played
* Calculate no. of players failing on last level they played

Metrics calculated -

- 1.DAU (daily active users)

- 2.Maximum days most players left

- 3.Maximum level players are playing

- 4.No. of fails on particular level and stage

- 5.No. of fails on particular level

- 6.Failing ratio (total fails per total starts )

-Fails by last event\_datetime

7.No. of players failed after which they never played the game (players who failed and immediately quit the game)

-Fails by last session\_id

8.No. of players failed in a session after which they never played the game (at least 1 fail in last session)

-Fails by last level

9.No. of players failed in their last level after which they never played the game (at least 1 fail in their last level played)

-Fails by last stage

10.No. of players failed in a stage after which they never played the game (at least 1 fail in last stage)

- 11.Levels on which most players failed and never played the game (by last event\_datetime)

- 12.Frequency of count of fails on their last level (so,most people fail 1 time in their last level played )

- 13. Exploring Players dataset

CREATE DATABASE kwalee;

USE kwalee;

CREATE TABLE levels (

event\_datetime DATETIME,

player\_id VARCHAR(35),

levelno INT,

stageno INT,

status\_ VARCHAR(10),

session\_id VARCHAR(35) );

#### **# Extract csv file into table**

LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/level\_progress.csv"

INTO TABLE levels

COLUMNS TERMINATED BY ','

OPTIONALLY ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 LINES

( event\_datetime,player\_id,levelno,stageno,status\_,session\_id );

SELECT \* FROM levels;

#### **1. DAU (daily active users)**

# SELECT AVG(c) FROM (

SELECT DATE(event\_datetime)AS d ,DAYNAME(event\_datetime),COUNT(player\_id) AS c FROM levels

GROUP BY DATE(event\_datetime),DAYNAME(event\_datetime)

ORDER BY DATE(event\_datetime) ;

#)AS table1;

#### **2. Maximum days most players played the game**

SELECT days\_played ,COUNT(days\_played) FROM

(

SELECT player\_id, COUNT(DISTINCT DAY(event\_datetime)) AS days\_played , COUNT(session\_id) ,MAX(levelno) AS maxlevel

FROM levels

GROUP BY player\_id

ORDER BY 2,3 DESC,4 DESC

) AS table1

GROUP BY days\_played;

days\_played count(days\_played)

1 2955

2 1146

3 433

4 203

5 87

6 55

#### **3. Maximum levels players played**

SELECT max\_level, COUNT(max\_level) AS player\_count FROM

(

SELECT player\_id, MAX(levelno) AS max\_level FROM levels

GROUP BY player\_id

ORDER BY 2 DESC

)AS table1

GROUP BY max\_level

ORDER BY 1,2 DESC;

max\_level player\_count

1 1231

2 1514

3 927

4 504

5 306

#### **4.No. of fails on particular level and stage**

SELECT levelno, stageno,COUNT(status\_)

FROM levels

WHERE status\_='fail'

GROUP BY levelno, stageno

ORDER BY 1,2;

#### **5.No. of Fails on level**

SELECT levelno,

COUNT(status\_)

FROM levels

WHERE status\_='fail'

GROUP BY levelno

ORDER BY 2 DESC;

levelno count(status\_)

4 1623

1 1256

2 1243

5 1144

3 838

6 498

7 279

8 255

#### 

#### **6. Failing ratio (total fails per total starts )**

WITH

table1 AS (

SELECT levelno, stageno, COUNT(status\_) AS fail FROM levels

WHERE status\_='fail'

GROUP BY levelno, stageno, status\_

ORDER BY 1 ),

table2 AS (

SELECT levelno, stageno, COUNT(status\_) AS total FROM levels

WHERE status\_='start'

GROUP BY levelno, stageno, status\_

ORDER BY 1 )

SELECT t1.levelno as Level\_number, t1.stageno as Stage\_number, t1.fail as Total\_fails ,t2.total as Total\_starts,CAST( (t1.fail/t2.total)\*100 AS DECIMAL(10,2) ) AS Fails\_ratio

FROM table1 AS t1

JOIN table2 t2

ON t1.levelno=t2.levelno AND t1.stageno=t2.stageno

ORDER BY 1,2;

### **Failing and Quitting**

#### 

#### **7.No. of players failed after which they never played the game (by last event\_datetime) (players who failed and immediately quit the game)**

WITH table1 AS

(

SELECT player\_id, status\_, levelno, stageno, session\_id, event\_datetime,

ROW\_NUMBER() OVER ( PARTITION BY player\_id ORDER BY event\_datetime DESC) AS rnk

FROM levels

)

SELECT status\_ AS players\_last\_event\_time,COUNT(status\_) AS no\_of\_players FROM table1

WHERE rnk BETWEEN 0 AND 1

GROUP BY status\_;

players\_last\_event\_time no\_of\_players

start 3123

fail 838

complete 978

#### **8.No. of players failed in their last session after which they never played the game (at least 1 fail in last session)**

WITH table1 AS

(

SELECT player\_id, levelno, stageno, status\_, session\_id, event\_datetime,

DENSE\_RANK() OVER ( PARTITION BY player\_id ORDER BY player\_id,session\_id) rnk

FROM levels

)

SELECT DISTINCT player\_id

FROM table1

WHERE status\_='fail' AND rnk BETWEEN 0 AND 1;

*Players count - 963*

#### **9.No. of players failed in their last level after which they never played the game (at least 1 fail in their last level played)**

WITH table1 AS (

SELECT player\_id, levelno, stageno, status\_, session\_id, event\_datetime,

DENSE\_RANK() OVER( PARTITION BY player\_id ORDER BY player\_id,levelno DESC) rnk

FROM levels

)

#select count(\*) from table1

SELECT DISTINCT player\_id FROM table1

WHERE status\_='fail' AND rnk=1;

*Players count - 1743*

#### **10.No. of players failed in a stage after which they never played the game (at least 1 fail in last stage)**

WITH table1 AS (

SELECT player\_id, levelno, stageno, status\_,

DENSE\_RANK() OVER ( PARTITION BY player\_id ORDER BY player\_id, levelno DESC,stageno DESC) rnk

FROM levels

)

SELECT DISTINCT player\_id

FROM table1

WHERE status\_='fail' AND rnk=1;

*Players count - 1350*

#### 

#### 

#### **11.Count of level/stage on which players failed and never played the game (by last event\_time)**

SELECT levelno, count(levelno) AS players\_failing FROM (

WITH table1 AS

(

SELECT player\_id, status\_, levelno, stageno, event\_datetime,

ROW\_NUMBER() OVER ( PARTITION BY player\_id ORDER BY event\_datetime DESC) AS rnk

FROM levels

)

SELECT player\_id, status\_, levelno, stageno, event\_datetime,rnk

FROM table1

WHERE status\_='fail' AND rnk=1

)AS table2

GROUP BY levelno

ORDER BY 2 DESC;

levelno players\_failing

4 168

2 160

1 140

3 126

5 111

6 47

7 21

#### 

#### 

#### 

#### 

#### 

#### **12.Frequency of count of fails on their last level ( so, people mostly fail 1 time in their last level played )**

fails\_on\_last\_level frequency

1 1208

2 328

3 120

4 52

5 13

select fails\_on\_last\_level, count(fails\_on\_last\_level) as frequency\_of\_fails from (

WITH table1 AS (

select player\_id,levelno,stageno,status\_,

DENSE\_RANK() OVER(PARTITION BY player\_id ORDER BY player\_id,levelno DESC) rnk

FROM levels

)

SELECT player\_id ,levelno AS last\_level,COUNT(status\_)AS fails\_on\_last\_level

FROM table1

WHERE status\_='fail' AND rnk=1

GROUP BY player\_id,levelno

)AS table3

GROUP BY fails\_on\_last\_level

ORDER BY 2 DESC;

#### 

#### 

#### 

#### 

#### 

#### **13.Exploring Players dataset**

create database kwalee;

use kwalee;

create table players (

install\_datetime datetime,

player\_id varchar(35),

platform varchar(10),

country varchar(5),

screen\_size double,

system\_memory int

);

desc players;

LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/players.csv"

INTO TABLE players

COLUMNS TERMINATED BY ','

OPTIONALLY ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 LINES

(install\_datetime,player\_id,platform,country,screen\_size,system\_memory);

**# EXPLORING DATA**

select \* from players;

select distinct country from players;

select country,count(player\_id) from players group by country order by 2 desc;

select platform,count(platform) from players group by platform;

select system\_memory,count(system\_memory) as count\_memory

from players

where system\_memory between 2000 and 4000

group by system\_memory order by 2 desc;

select sum(count\_memory) from

(

select system\_memory,count(system\_memory) as count\_memory

from players

where system\_memory between 2600 and 4000

group by system\_memory order by 2 desc

)

as table1;

select screen\_size,count(screen\_size) from players group by screen\_size order by 2 desc;

select date(install\_datetime) from players order by install\_datetime ;

##### **New daily installs**

select date(install\_datetime), count(install\_datetime) as daily\_installs

from players

group by DATE(install\_datetime)

order by date(install\_datetime);

##### **Average new daily installs**

with table2 as

(

select date(install\_datetime), count(install\_datetime) as daily\_installs

from players

group by DATE(install\_datetime)

order by date(install\_datetime)

) select avg(daily\_installs) from table2;

##### **Total players did not opened the app after installing (60 players)**

select p.player\_id ,l.event\_datetime

from players p

left join levels l

on l.player\_id=p.player\_id

where event\_datetime is NULL;

##### **Failing and quitting relationship with platform**

select platform, count(platform) from

(

with table1 as (

select player\_id,levelno,stageno,status\_,

dense\_rank() over(partition by player\_id order by player\_id,levelno desc,stageno desc) rnk

from levels

)

select distinct t.player\_id , p.platform as platform, system\_memory

from table1 t

join players p

on t.player\_id=p.player\_id

where status\_='fail' and rnk=1

)as table2

group by platform;

##### 

##### 

##### 

##### 

##### 

##### **Failing and quitting relationship with system\_memory**

select system\_memory,count(system\_memory) from #platform, count(platform) from

(

with table1 as (

select player\_id,levelno,stageno,status\_,

dense\_rank() over(partition by player\_id order by player\_id,levelno desc,session\_id) rnk

from levels

)

select distinct t.player\_id , p.platform as platform, system\_memory

from table1 t

join players p

on t.player\_id=p.player\_id

where status\_='fail' and rnk=1

)as table2

group by system\_memory

order by 2 desc;

#group by platform;