How can we increase revenue from Catch the Pink Flamingo?

Muhammed Abdul-Fattah Salem

Context

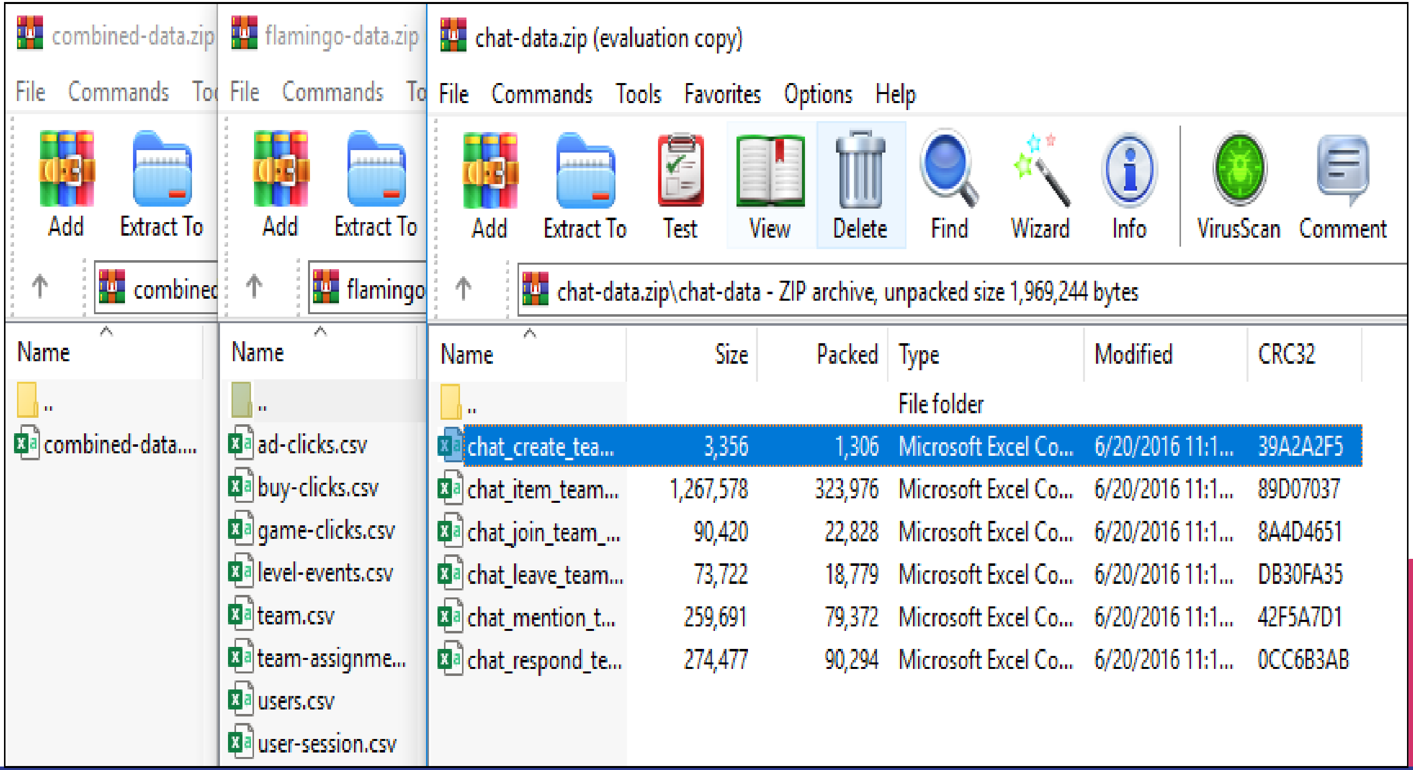
CATCH THE PINK FLAMINGO GAME| REVENUE GROWTH STRATEGY

- ❑ Eglence Inc. game .
- ❑ **Platform**: Mobile.
- ❑ Multi-users game.
- ❑ **Objective**: following missions provided by real-time prompts and cover each level grid map hitting as many flamingos as possible.

Script

- Eglence Inc.| Catch the Pink Flamingo Game. It is a multi-user mobile game where players have to catch Pink Flamingos that randomly pop up on a gridded world map based on missions changing in real-time, reaching top ranks.
- The game objective is to catch as many Pink Flamingos as possible by following the missions provided by real-time prompts in the game and cover the map provided for each level. The levels get more complicated in mission speed and map complexity as the users move from level to level.
- In the upcoming sections, we are insightfully figuring out some revenue growth strategies for the game using different data analytics tools.

Problem Statement

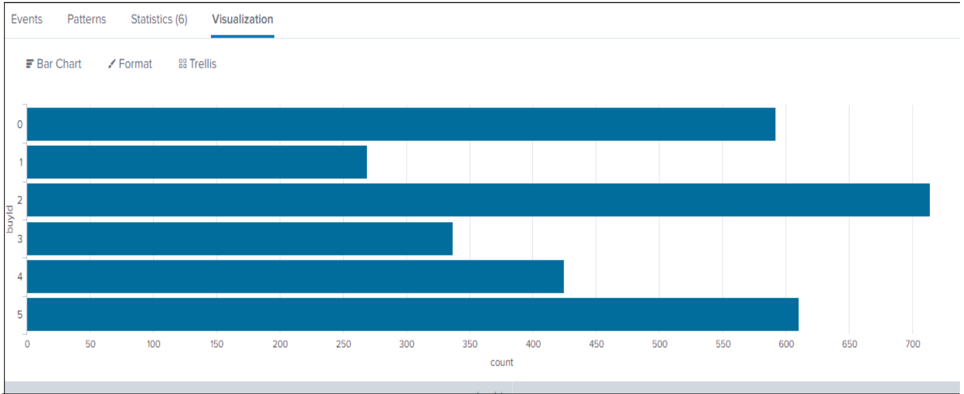


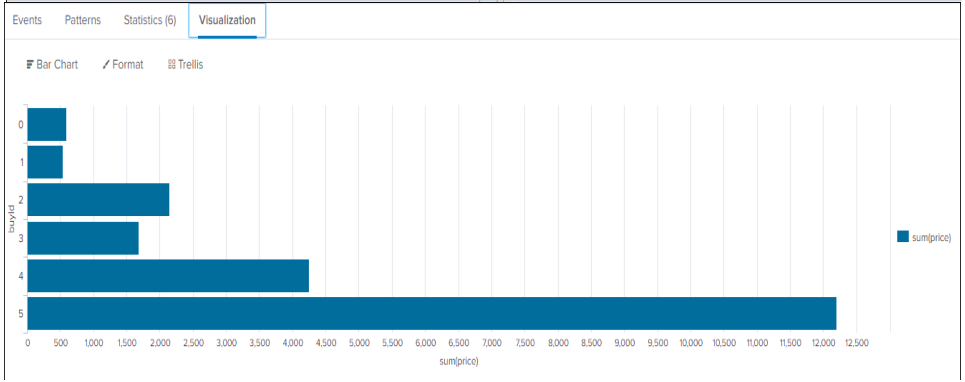
Script

For our game data analytics revenue growth strategy, we have its 3 subjects of different data sources to work on:

- ❑ The “**flamingo-data**” contains 8 CSV files containing simulated game data and log data for the Catch the Pink Flamingo Simulated Game Data. These data have been used for data exploration in Splunk〉.
- ❑ The “**combined-data**” contains a single CSV file created by aggregating data from several game data files. It has been used for identifying big spenders with Knime SparkMLib.
- ❑ The “**chat-data**” contains 6 CSV files representing simulated chat data related to the Catch the Pink Flamingo game to be used in Graph Analytics with neo4j.
- ❑ Combining and analyzing these datasets will contribute to new insights from these data and recommendations for revenue growth strategy.

Data Exploration Overview



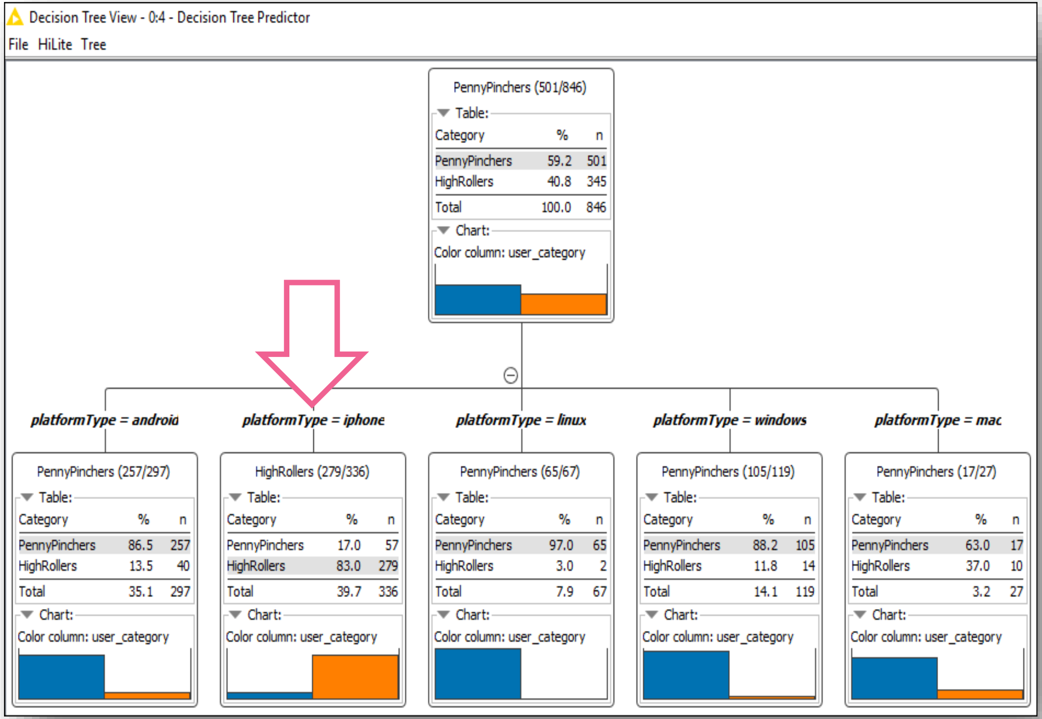


- Item “2” is the most purchased, but item “1” is the least.
- Item “5” has the highest revenue, but item “1” has the lowest impact on revenue.
- Iphone platform users are the top in purchases.

Script

- For better understanding, let’s import theses datasets of ”flamingo-data” into **Splunk** 〉, using aggregation functions to make histograms as follows:
- The first histogram is figured using “buy-clicks” dataset, shows how many times each unique item has been purchased. the item “2” is the only most purchased between 6 items, and the item “1” is the least.
- The second histogram is made by the same dataset. This graph shows that the item “5” had the highest revenue, and the item “1” had also the least record; the item “1” is not preferred by most of players. In this case, providers should pay attention to such not worth products.

Classification Analysis



- ❑ HighRollers: **$]5, ∞ [**

  PennyPinchers: **$]- ∞,5]**

- ❑ Results:

  Iphone platform users are **HighRollers**.

Script

- **For classification analysis**, we used the "combined-data" dataset, and classified users with Decision Tree in KNIME to identify HighRollers in the game. **The objective** is to predict which user is likely to purchase big-ticket items while playing game.
- Using the file "combined_data", we defined two categories of users: **HighRollers**, who spend more than $5.00 to buy the items, and **PennyPinchers**, who spend $5.00 or less to buy the items. In this graph, blue stands for the PennyPinchers, and orange for the HighRollers.
- The resulting decision tree shows that the predicted user_category is different in various platforms, the users on the platform android, linux, windows and mac are almost **PennyPinchers**. However, most iphone platform users are **HighRollers**.
- Furthermore, thanks to the confusion matrix, our decision tree model has an accuracy rate = 88.5%.

Clustering Analysis

Cluster	adClicks	gameClicks	Revenue
1	25.12037037	362.50308642	35.35802469
2	32.05	2393.95	41.20
3	36.47486034	953.82122905	46.16201117

Low_level_spenders

Neutral_users

High_level_spenders

- High_level_spenders' adClicks is 1.45 times more than for Low_level_spenders and 1.14 times more than for Neutral_users.
- High_level_spenders' gameClicks is 2.63 times more than for Low_level_spenders.
- High_level_spenders' revenue is 1.31 times higher than for Low_level_spenders and 1.12 times higher than Neutral_users.

Script

- **<u>For the clustering analysis</u>**, we are using **iPython Notebook** with **Spark Mllib**:
- There are 3 attributes to focus on: amount of ad-clicking per user, amount of game-clicking per user and total price spent by each user. The training dataset is created by combining these 3 attributes in one table.
- Finally, training to create cluster centers and getting results as follows:

Cluster 1 is different from the others in that the users' adClicks, gameClicks and revenue are all less than others, this kind of users can be called "**Low_level_spenders**".

Cluster 2 is different from the others in that the adClicks is not the least, gameClicks is the most, but their revenue is not the most, this kind of users can be called "**Neutral_users**"

Cluster 3 is different from the others in that the users' ad-clicks, game-clicks and cost are all more than others, this kind of users can be called "**High_level_spenders**".

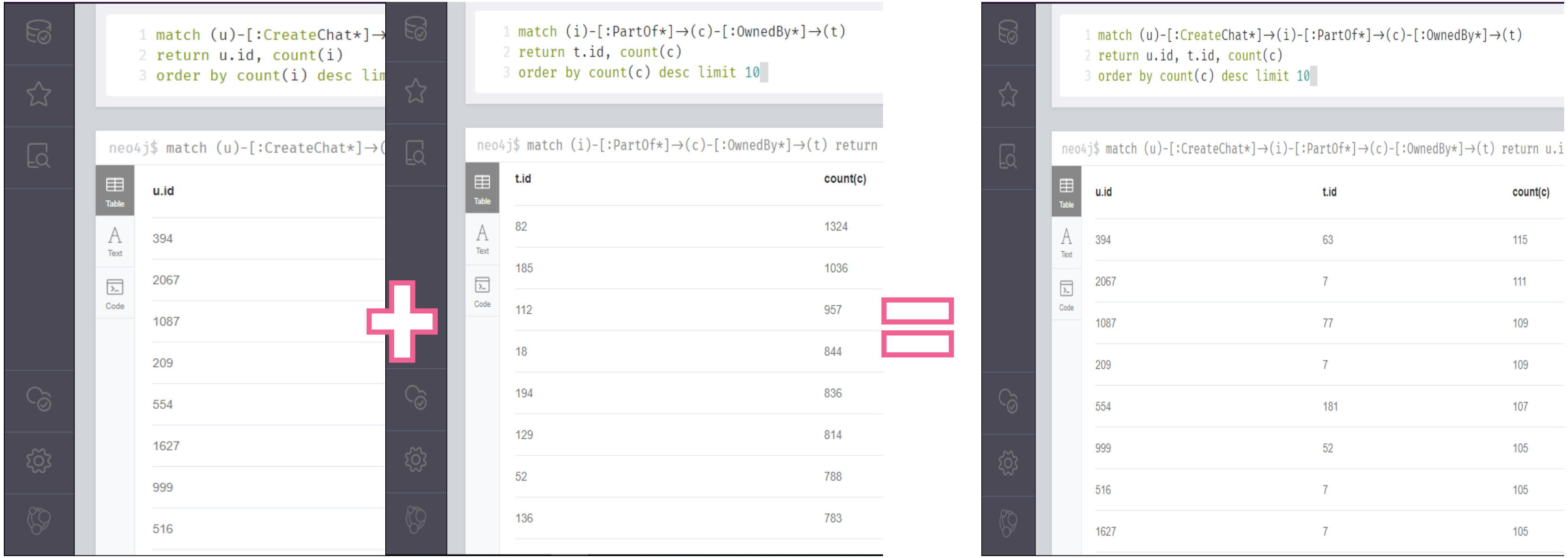
<u>These results reflect</u>:

High_level_spenders' adClicks is 1.45 times more than for Low_level_spenders' and 1.14 times more than for Neutral_users.

High_level_spenders' gameClicks is 2.63 times more than for Low_level_spenders.

High_level_spenders' revenue is 1.31 times higher than for Low_level_spenders and 1.12 times higher than Neutral_users.

Graph Analytics: Chat Data



```
match (u)-[:CreateChat*]→
return u.id, count(i)
order by count(i) desc lim
```

neo4j$ match (u)-[:CreateChat*]→(

u.id
394
2067
1087
209
554
1627
999
516

```
match (i)-[:PartOf*]→(c)-[:OwnedBy*]→(t)
return t.id, count(c)
order by count(c) desc limit 10
```

neo4j$ match (i)-[:PartOf*]→(c)-[:OwnedBy*]→(t) return

t.id	count(c)
82	1324
185	1036
112	957
18	844
194	836
129	814
52	788
136	783

```
match (u)-[:CreateChat*]→(i)-[:PartOf*]→(c)-[:OwnedBy*]→(t)
return u.id, t.id, count(c)
order by count(c) desc limit 10
```

neo4j$ match (u)-[:CreateChat*]→(i)-[:PartOf*]→(c)-[:OwnedBy*]→(t) return u.i

u.id	t.id	count(c)
394	63	115
2067	7	111
1087	77	109
209	7	109
554	181	107
999	52	105
516	7	105
1627	7	105

- Most of the chattiest users are not a part in the chattiest teams, but only the user with U.id = 999.

Script

- For the chat-data Graph Analytics in **Neo4J**:
- 6 CSV data files of “ chat-data” are loaded into Neo4J. Firstly after that, we have figured out the top 10 chattiest users by matching all users with a CreateChat edge, returning the users’ ID and the count of the users.
- Secondly, we have figured out the top 10 chattiest teams by matching all ChatItems with PartOf edge, connecting them with a TeamChatSession node and the TeamChatSession nodes must have an OwnedBy edge, connecting them with any other node, and returning the count of teams.
- Finally, we have figured out the third table, combining previous results, shows that only one of the top chattiest users with U.id = 999 belongs to one of th top chattiest teams with T.id= 52, but other 9 users are not part of the top 10 chattiest teams.
- Furthermore, If we could identify these highly interactive neighborhoods, we should potentially target **most active users** of the neighborhood, **based on cluster coefficient**, for direct advertising

Recommended Actions

- Offer iPhone users more packages and limited promotions$.
- Add more products with higher-price ads$ to “High_level_spenders” & “HighRollers”.
- Encourage low-level spenders and PennyPinchers with fixed pay packages and limited promotion to users.

Script

Following previous results, here are some recommendations to be considered:

❑Offer iPhone users more packages and limited promotions$.

Explain:

Splunk> : the data exploration analysis reflects that most users with iphone platform are high-level spenders; offering them more products with higher price ads will increase revenue/user.

❑Add more products with higher-price ads$ to “High_level_spenders” & “HighRollers”.

Explain:

Knime SparkMLib : according to classification and clustering analysis, high-level spenders clicked less and buy most; adding more ads for higher-price products will increase revenue/user.

❑Encourage low-level spenders and PennyPinchers with fixed pay packages and limited promotion to users.