**CHAPTER 1**

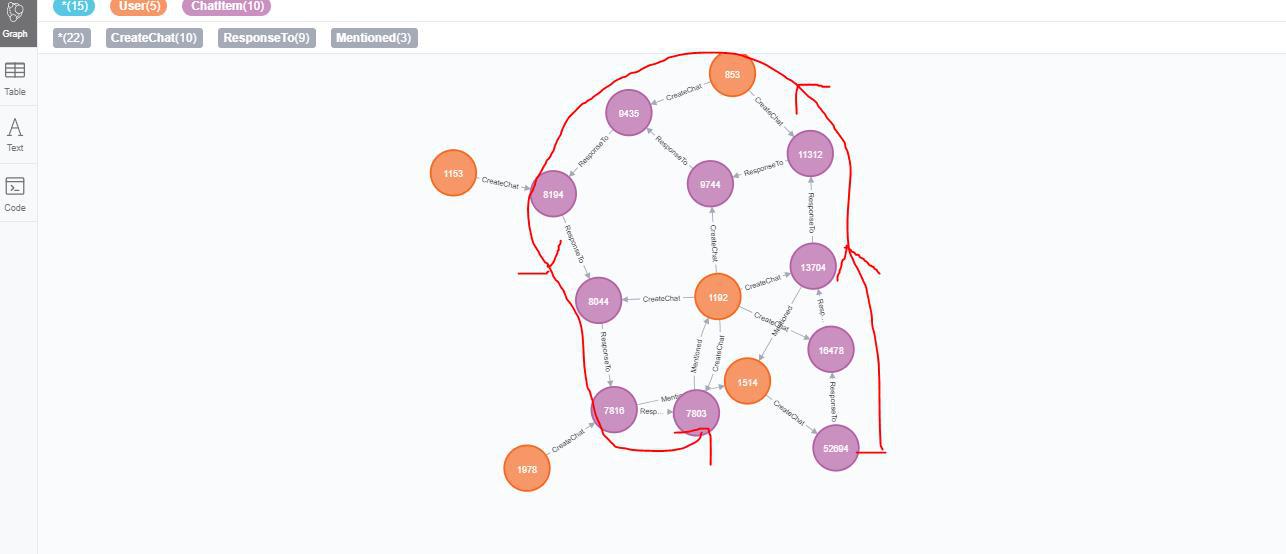
**Introduction**

* This project builds a big data ecosystem using various tools and methods.
* It analyzes a data set simulating big data generated from a large number of users who are playing our imaginary game “Catch the Flamingo”.
* Then, walk through the typical big data science steps for acquiring, exploring, preparing, analyzing and reporting.

**1.1** **Catch the Flamingo**

* It’s a ​ multi-­user game​where the players have to catch Pink Flamingos that randomly pop up on a gridded world map based on missions that change in real-time.
* For the player or team to move to the next complexity level, they need to have at least one point in every map grid cell, i.e., cover the whole world map.
* After the initial sign up, a player (user) is asked to play the Level 1 individually without joining any team. This is where the user gets trained as a player and starts building a game history.
* Level 1 is an easy entry to the game composed of only 64 (8x8) grid cells and longer, more obvious, fun missions.
* Upon completion of Level 1, the player gets asked if she/he wants to join any team or form a team and will continue the rest of the time as a team player even if that means the user is a 1­-person team of her/his own.
* At the beginning of each level, the game creates a brand new map with more cells than the level before. The complexity of the missions also increases.
* The missions change more frequently as the levels increase. The players keep in touch via **​chat boards ​**assigned to the teams and also via social media, e.g., Twitter.

**1.2 Data Set Overview**

* The table below lists each of the files available for analysis with a short description of what is found in each one.

|  |  |  |
| --- | --- | --- |
| **File Name** | **Description** | **Fields** |
| **ad-clicks.csv** | A line is added to this file when a player clicks on an advertisement in the Flamingo app. | timestamp: when the click occurred.  txId: a unique id (within adclicks.log) for the click.  userSessionid: the id of the user session for the user who made the click.  teamid: the current team id of the user who made the click.  userid: the user id of the user who made the click.  adId: the id of the ad clicked on.  adCategory: the category/type of ad clicked on. |
| **buy-clicks.csv** | A line is added to this file when a player makes an in-app purchase in the Flamingo app. | timestamp: when the purchase was made.  txId: a unique id (within buyclicks.log) for the purchase  userSessionId: the id of the user session for the user who made the purchase  team: the current team id of the user who made the purchase  userId: the user id of the user who made the purchase  buyId: the id of the item purchased.  price: the price of the item purchased |
| **users.csv** | This file contains a line for each user playing the game. | timestamp: when user first played the game.  userId: the user id assigned to the user.  nick: the nickname chosen by the user.  twitter: the twitter handle of the user.  dob: the date of birth of the user.  country: the two-letter country code where the user lives. |
| **team.csv** | This file contains a line for each team terminated in the game. | teamId: the id of the team  name: the name of the team  teamCreationTime: the timestamp when the team was created  teamEndTime: the timestamp when the last member left the team  strength: a measure of team strength, roughly corresponding to the success of a team  currentLevel: the current level of the team |
| **team-assignments.csv** | A line is added to this file each time a user joins a team. A user can be in at most a single team at a time. | timestamp: when the user joined the team.  team: the id of the team  userId: the id of the user  assignmentId: a unique id for this assignment |
| **level-events.csv** | A line is added to this file each time a team starts or finishes a level in the game | timestamp: when the event occurred.  eventId: a unique id for the event  teamId: the id of the team  teamLevel: the level started or completed  eventType: the type of event, either start or end |
| **user-session.csv** | Each line in this file describes a user session, which denotes when a user starts and stops playing the game. Additionally, when a team goes to the next level in the game, the session is ended for each user in the team and a new one started. | timestamp: a timestamp denoting when the event occurred.  userSessionId: a unique id for the session.  userId: the current user's ID.  teamId: the current user's team.  assignmentId: the team assignment id for the user to the team.  sessionType: whether the event is the start or end of a session.  teamLevel: the level of the team during this session.  platformType: the type of platform of the user during this session. |
| **game-clicks.csv** | A line is added to this file each time a user performs a click in the game. | timestamp: when the click occurred.  clickId: a unique id for the click.  userId: the id of the user performing the click.  userSessionId: the id of the session of the user when the click is performed.  isHit: denotes if the click was on a flamingo (value is 1) or missed the flamingo (value is 0)  teamId: the id of the team of the user  teamLevel: the current level of the team of the user. |

**CHAPTER 2**

**Motivation**

The idea of the project has been devised by taking in mind the current generation: Today’s youth. Today’s generation is very much occupied with online games. They are very freak with gaming. So, considering that we have tried to choose that type of project in which we can enhance our skills and can be helpful for the companies that develop games. We have tried to recommend actions to increase the revenue of the game by analyzing their behavior, etc.

Using the latest tools to predict results and analyze the data to extract that meaningful information that not only gives us interesting insights but also helps us in making data driven decisions that previously were not possible.

Nowadays data is driving the world as is evident from the fact that big data is used almost in every big corporation and we will know that data scientist is considered one of the hottest jobs in the digital 21st century. Therefore, we wanted to work on this technology and see ourselves what we can learn from it, how we can analyze huge amounts of data and last but not the least to know how data influences decision making in these big MNCs.

**CHAPTER 3**

**Literature Review**

The aim of this study was to explore the relationship between online gaming motivation, self-concept clarity and problematic online gaming. More specifically, the study investigated the mediating role of gaming motives between self-concept clarity and problematic online gaming. Data from young adult video game players from Croatia were analyzed. Problematic online gaming was positively correlated with social, competition, coping, fantasy and escape motives for playing online video games, and negatively with self-concept clarity. Hierarchical regression analyses revealed that escape motives and self-concept clarity were significant predictors of problematic online gaming after controlling for age and weekly gaming time. The results of the mediation model showed that self-concept clarity was both directly and indirectly (via escape motive) associated with problematic online gaming. The discussion addresses the issue of escapism in relation to self-concept clarity and as a factor in predicting problematic online gaming.

**CHAPTER 4**

**Objective**

We will extract and analyze the following type of queries:

* Player chat behavior to find the ways of improving the game.
* We can pin point the countries which have the most number of players.
* Analyze their playing habits like how many hours they spend every week online, the popular time during which the most of the players are online both during a day and a week.
* The main objective is to increase the revenue of the game so we will recommend the actions after analyzing the dataset that how can we increase the revenue.

**CHAPTER 5**

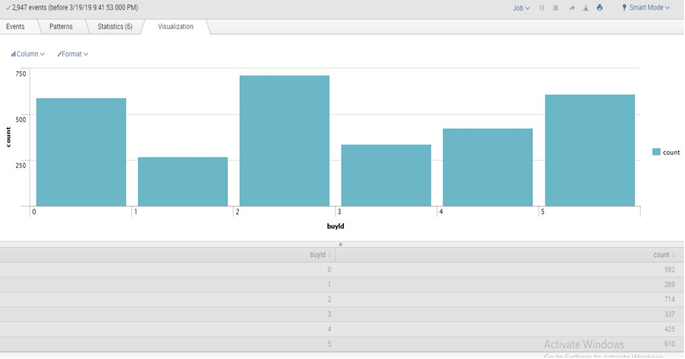
**Implementation Overview**

**5.1 Exploring Data with SPARK**

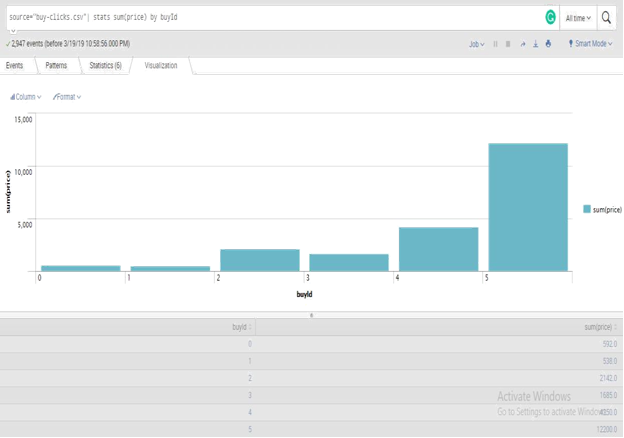
**5.1.1 Aggregation**

|  |  |
| --- | --- |
| Amount spent buying items | 21407.0 |
| Number of unique items available to be purchased | 6 |

**A histogram showing how many times each item is purchased:**



**A histogram showing how much money was made from each item:**

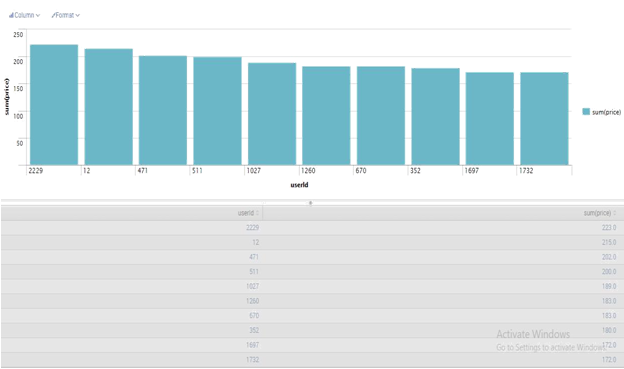
****

**5.1.2 Filtering**

The following table shows the user id, platform, and hit-ratio percentage for the top three buying users:

|  |  |  |  |
| --- | --- | --- | --- |
| **Rank** | **User Id** | **Platform** | **Hit-Ratio(%)** |
| 1 | 2229 | iPhone | 11.5970 |
| 2 | 12 | iPhone | 13.0682 |
| 3 | 471 | iPhone | 14.5038 |

**A histogram showing total amount of money spent by the top ten users (ranked by how much money they spent).**

****

**5.2 Data Classification with KNIME**

**5.2.1 Data Preparation**

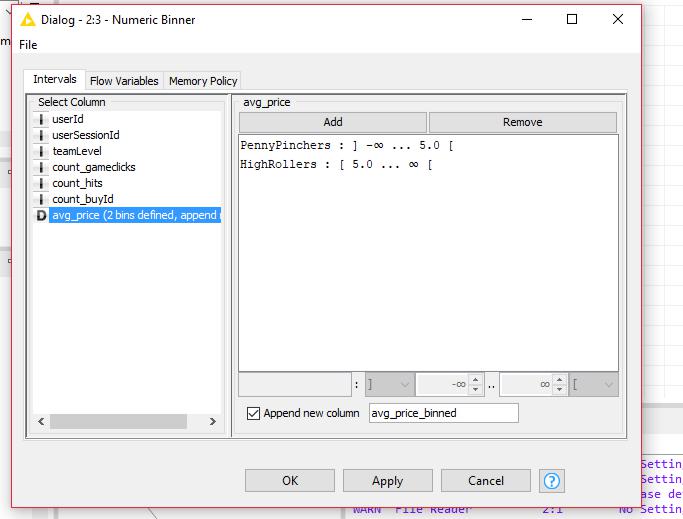
Analysis of combined\_data.csv

**Sample Selection**

|  |  |
| --- | --- |
| **Item** | **Amount** |
| # of Samples | 4619 |
| # of Samples with Purchases | 1411 |

**Attribute Creation**

A new categorical attribute was created to enable analysis of players as broken into 2 categories (HighRollers and PennyPinchers). A screenshot of the attribute follows:



The creation of this attribute was guided by the avg\_price atribute, when the value was more than $5.00 the user was set with category as High Roller, and when the value was $5.00 or less the user was set with category as PennyPinchers. The creation of this new categorical attribute was necessary because it make things easier to analyze and find potential buyers for determined item.

**Attribute Selection**

The following attributes were filtered from the dataset for the following reasons:

|  |  |
| --- | --- |
| **Attribute** | **Rationale for Filtering** |
| avg\_price | We don’t need the average price anymore since we have a new categorized attribute avg\_price\_binned based on this. |
| user\_Id | Don’t need this since it’s just a computer generated number. |
| user\_Session\_Id | Don’t need this since it’s just a computer generated number. |

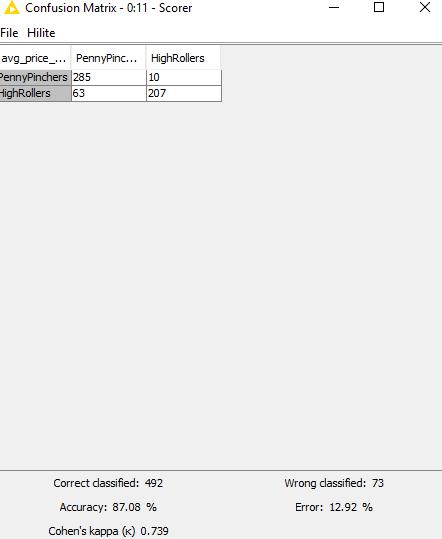
**5.2.2 Data Partitioning and Modeling**

The data was partitioned into train and test datasets. The train data set was used to create the decision tree model. The trained model was then applied to the test dataset. This is important because decision tree uses the train dataset to know how to predict, so it uses what was learned to use in the test dataset to make the prediction, and compares both results (from train and test datasets) to check the accuracy and error. When partitioning the data using sampling, it is important to set the random seed to make sure the partition is the same every time you run the program. This is needed when you need a reproducible result.

**A screenshot of the resulting decision tree can be seen below**:

**5.2.3 Evaluation**

**A screenshot of the confusion matrix can be seen below:**

****

As seen in the screenshot above, the overall accuracy of the model is **87.08%**

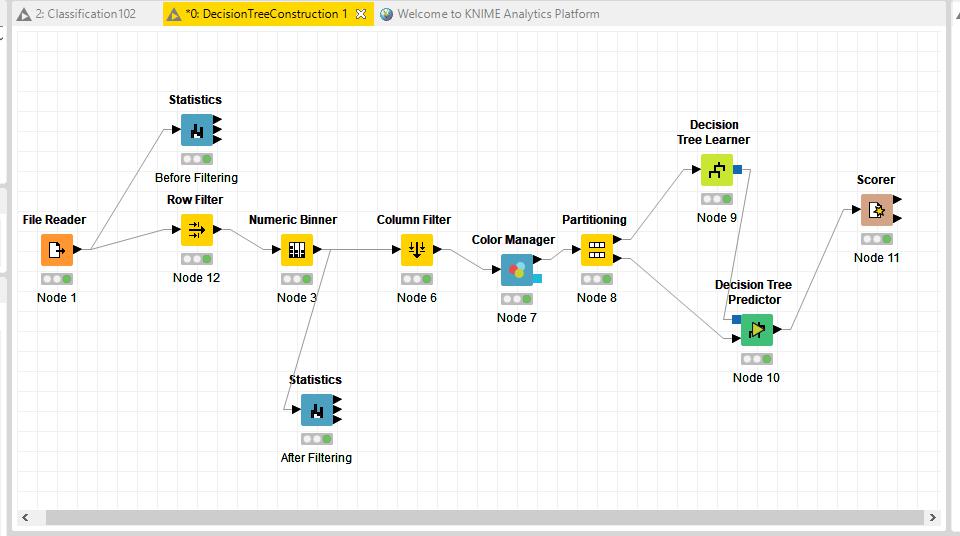
High Rollers correctly predicted – 207

Penny pitchers correctly predicted – 285

High Rollers as Penny Pinchers in-correctly predicted – 63

Penny pitchers as High Rollers in-correctly predicted – 10

**5.2.4 Analysis Conclusions**

The final KNIME workflow is shown below:

What makes a HighRoller vs. a PennyPincher ?

* Gamers using iPhone has more probability to be a High Roller, while gamers using Windows, Linux, Mac and Android has more probability to be Penny Pinchers.

|  |
| --- |
| **Specific Recommendations to Increase Revenue** |
| 1. Show more ads to iPhone users and increase ads price for the same platform device. |
| 1. Offer temporary discounts to motivate Penny Pinchers to buy items that cost more than five dollars. |

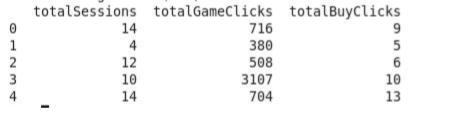
**5.3. Clustering with Spark**

**5.3.1 Attribute Selection**

|  |  |
| --- | --- |
| **Attribute** | **Rationale for Selection** |
| totalBuyClicks | New attribute with total purchase of each user, grouped by userId attribute, and the file used is buy-clicks.csv. Used to analyze how many times a user make in-app purchases. |
| totalGameClicks | New attribute with total clicks each user made inside the app, grouped by userId attribute from the file game-clicks.csv. Used to analyze how many “time” the user spends playing the game. |
| totalSessions | New attribute with total sessions each user made, grouped by userId atribute from the file user-session.csv. Used to analyze how often the user plays the game. |

**5.3.2 Training Data Set Creation**

The training data set used for this analysis is shown below (first 5 lines):



Dimensions of the training data set (rows x columns): 531x3

# of clusters created: 3

**5.3.3 Cluster Centers**

|  |  |
| --- | --- |
| **Cluster #** | **Cluster Center(totalSessions, totalGameClicks, totalBuyClicks)** |
| Cluster 1 | Array([7.27922078, 369.18831169, 4.95779221]) |
| Cluster 2 | Array([11.19101124, 927.70224719, 6.45505618]) |
| Cluster 3 | Array([11.24444444, 2310.64444444, 5.53333333]) |

These clusters can be differentiated from each other as follows:

Cluster 1 is different from the others in that it has the lower values in all fields. We can conclude that, few sessions means the user is new or has no Interest in the game, and eventually, buy less from other players.

Cluster 2 is different from the others in that it has higher values in all fields from cluster 1 and lower values in two fields from cluster 3. We can conclude that more time the user plays, but less clicks he made inside the app (spend playing), causes the user to buy more time.

Cluster 3 is different from the others in that it has higher values in all fields from cluster 1 and lower value in one field from cluster 2. We can conclude that more time the user plays the game, and more clicks he made inside the app (spend playing), the less he will buy (doesn’t mean he has the lowest totalBuyClicks, but he has less interest on buying items).

**5.3.4 Recommended Actions**

|  |  |
| --- | --- |
| **Action Recommended** | **Rationale for the action** |
| Make items promotions based on how much clicks they make. | Suppose a user is playing a session, if he/she reach 100 clicks in that session, he/she get a promotion on an item (only on that session and to use on that session). This incentivizes the user to spend more time playing, to get more promotions. |
| Make items promotions based on how much sessions they start | Suppose the user start a session, and is his 1000 session, he/she can buy an item (only on that session and to use on that session) with less prices. This incentivizes the users to come play more time so he/she can get more promotions. |

**5.4 Graph Analytics of Simulated Chat Data With Neo4j**

**5.4.1 Modeling Chat Data using a Graph Data Model**

The graph model is composed by 45463 nodes and 118502 relationships (after all steps concluded).

**The nodes are:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Property** | **Explanation** |
| User | ID | Represent the users that interacts in the chat. |
| Team | ID | Represent the team composed by the users that interacts in the chat. |
| TeamChatSession | ID | Represent the session of a chat based on the team of the user that created the chat. |
| ChatItem | ID | Represent a “message” send inside the chat, this message is called item. |

**The relationships are:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Property** | **Explanation** |
| CreateSession | timeStamp | Composed by TeamChatSession and User, is generated every time a user start a new session in the chat |
| OwnedBy | timeStamp | Composed by TeamChatSession and Team, is generated everytime a new session is started in the chat, and vinculated to the TeamChatSession vinculated to a user from that team |
| Joins | timeStamp | Composed by User and TeamChatSession, is generated everytime a user join an already created team chat session |
| Leaves | timeStamp | Composed by User and TeamChatSession, is generated everytime a user leaves a team chat session |
| Mentioned | timeStamp | Composed by ChatItem and User, is generated everytime a user is mentioned in a item (message) in a chat |
| PartOf | timeStamp | Composed by ChatItem and TeamChatSession, is generated eveytime a item (message) is send inside the chat, is vinculated to that specific TeamChatSession because it was generate on that session |
| ResponseTo | timeStamp | Composed only by ChatItem, is generated everytime a item (messages) is send as a response to another item (message) |
| IneractsWith | timeStamp | Composed only by User, is generated everytime a user has directed interacted with another user in the chat |

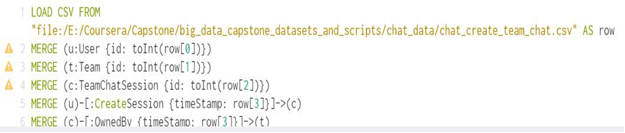
**5.4.2 Creation of the Graph Database for Chats**

Schema of the 6 csv files

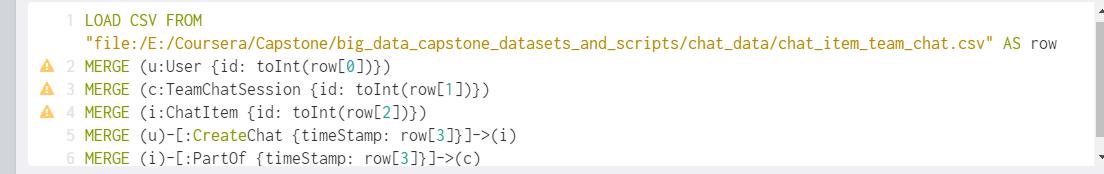
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FILE NAME** | **COLUMN 0** | **COLUMN 1** | **COLUMN 2** | **COLUMN 3** |
| chat\_create\_tea m\_chat.csv | userId | teamId | teamChatSessionId | timeStamp |
| chat\_item\_team \_chat.csv | userId | teamChatSessionId | chatItemId | timeStamp |
| chat\_join\_team\_ chat.csv | userId | teamChatSessionId | timeStamp |  |
| chat\_leave\_team \_chat.csv | userId | teamChatSessionId | timeStamp |  |
| chat\_mention\_te am\_chat.csv | chatItemId | userId | timeStamp |  |
| chat\_respond\_te am\_chat.csv | chatItemId | chatItemId | timeStamp |  |

**5.4.3 The loading process**

chat\_create\_team\_chat.csv – Load the csv file. It created two edge labels the CreateSession from User to TeamChatSession, and the OwnedBy from TeamChatSession to Team. The columns are 0 to the user as id, 1 to team as id, 2 to teamChatSession as id, and 3 as timeStamp for both CreateSession and OwnedBy. These loads can be done using the MERGE statement.



chat\_item\_team\_chat.csv – Load the csv file. It Create two edges labels, the CreateChat from User to ChatItem, and the PartOf from ChatItem to TeamChatSession. The columns are 0 to the user as id, 1 to the TeamChatSession as id, 2 to ChatItem as id, 3 to both CreateChat and PartOf as timeStamp. These loads can be done using the MERGE statement.



chat\_join\_team\_chat.csv – Load the csv file. This file creates one edge label, Joins, from User to TeamChatSession. The columns are 0 to user as id, 1 to teamChatSession as id, and 2 to Joins as the timeStamp. This load can be done using MERGE statement.

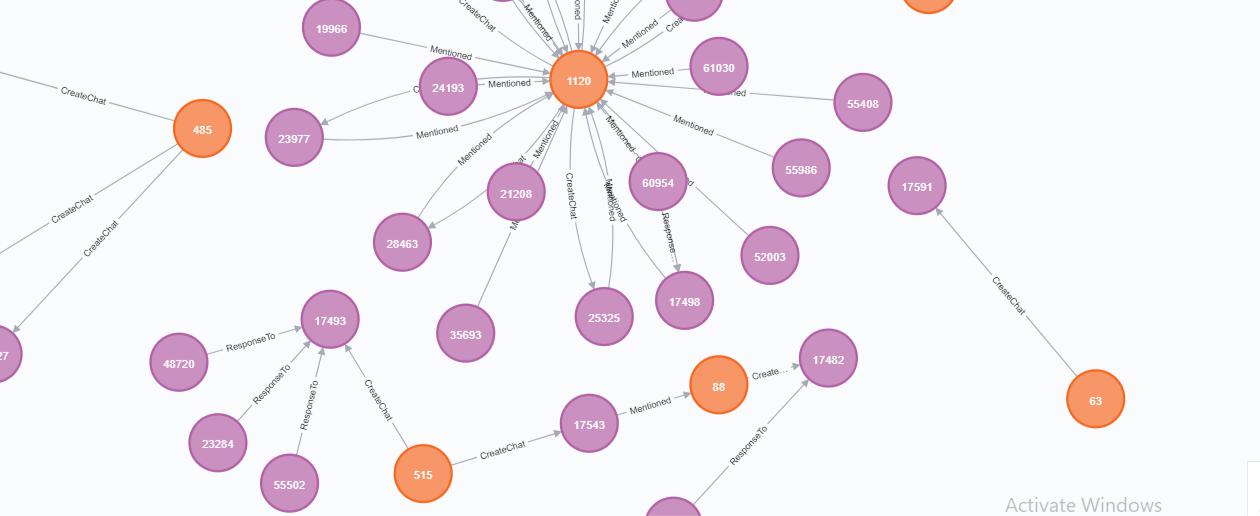


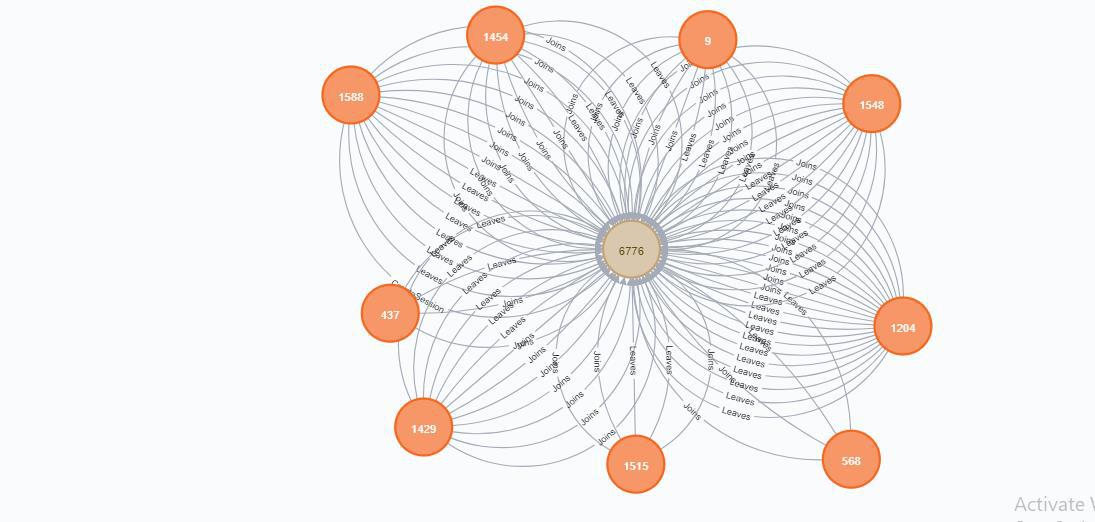
chat\_leave\_team\_chat.csv – Load the csv file. This file creates one edge label, Leaves, from User to TeamChatSession. The columns are 0 to user as id, 1 to teamChatSession as id, and 2 to Leaves as the timeStamp. This load can be done using MERGE statement

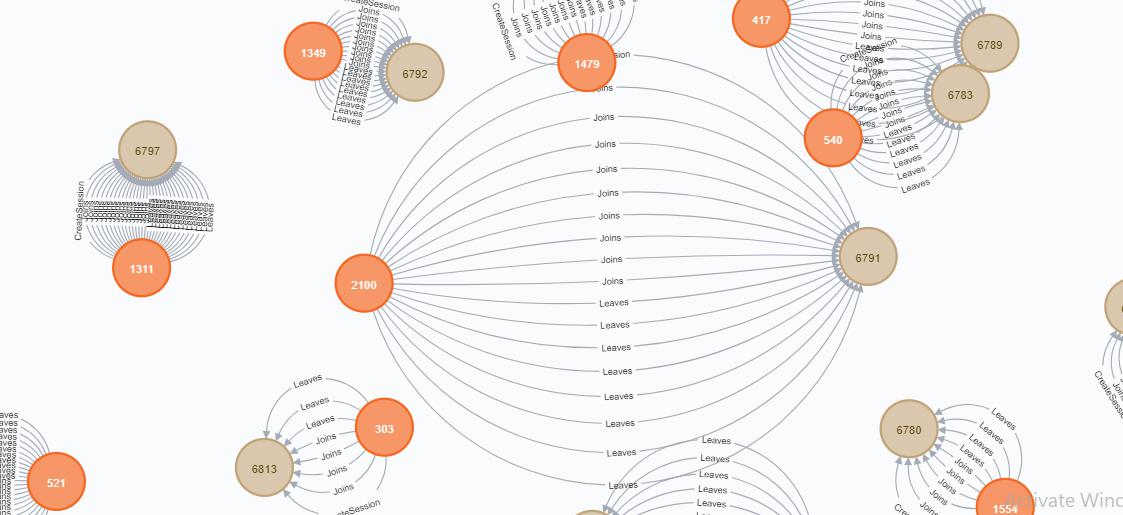
chat\_mention\_team\_chat.csv – Load the csv file. This file creates one edge label, Mentioned, from ChatItem to User. The columns are 0 to chatItem as id, 1 to user as id, and 2 to Mentioned as the timeStamp. This load can be done using MERGE statement.



chat\_respond\_team\_chat.csv – Load the csv file. This file creates one edge label, ResponseTo, from one ChatItem to another ChatItem. The columns are 0 to chatItem as id, 1 to chatItem as id, and 2 to ResponseTo as the timeStamp. This load can be done using MERGE statement

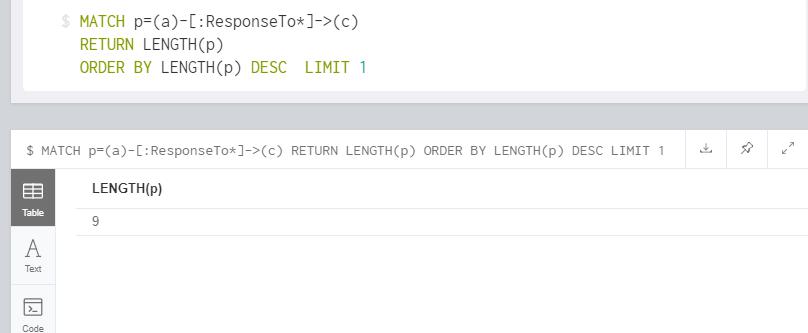
**Graph Example**





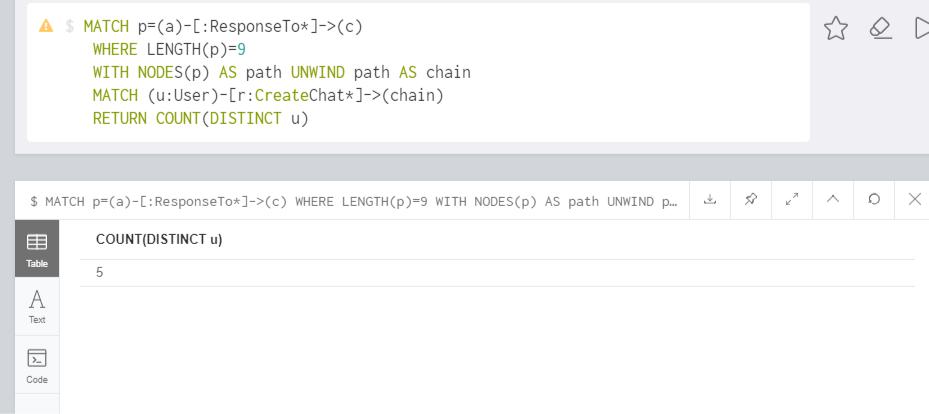
**5.4.4. Finding the longest conversation chain and its participants**

To analyze the longest conversation chain and its participants was made with two queries, were first was found the longest conversation chain, and second was found who interacts in this chain.

**The first query:**

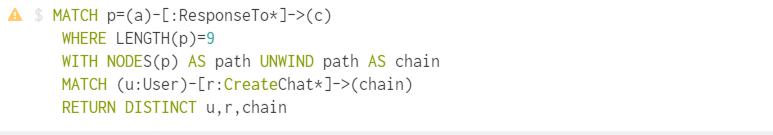
Return as result a LENGTH(p) = 9.

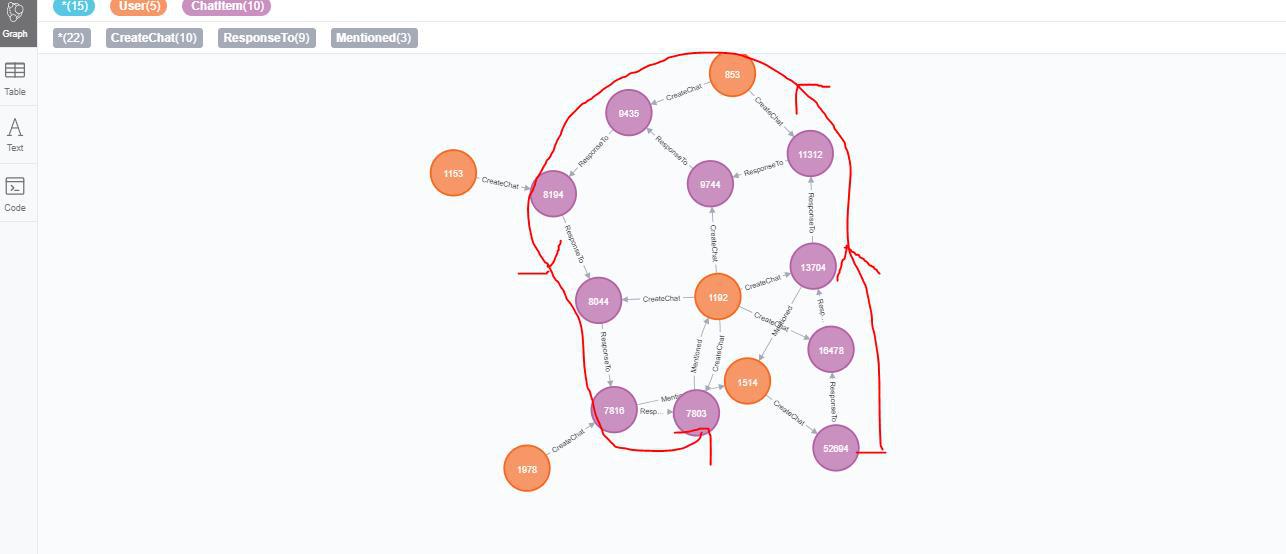
**The second query:**



Return as result a COUNT(DISTINCT u) = 5

**The resulting graph (and the query) is shown below:**



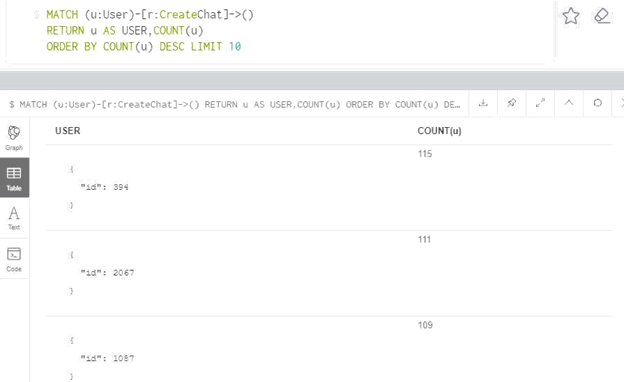
****

These users are in the longest path, which means, their chats are “the most relevant”, with mentions and chats being created. It can be useful to make a rank of the most influent users in chats, and eventually, show relevant ads (maybe, social network apps), and also, the “download” of the game by people that is not playing the game (It can be done by incentivizing the users of this chain to bring more people from their social networks).

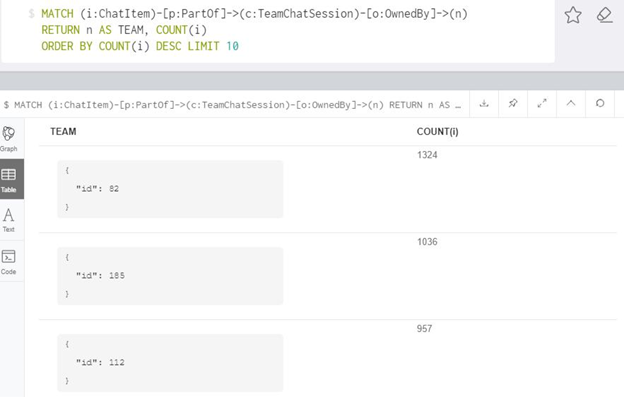
**5.4.5 Analyzing the relationship between top 10 chattiest users and top 10 chattiest teams**

There were used two queries, one to find the chattiest users and other to find the chattiest teams.

**Query to find the chattiest users:**



**Query to find the chattiest teams:**

****

**Chattiest Users**

|  |  |
| --- | --- |
| Users | Number of Chats |
| 394 | 115 |
| 2067 | 111 |
| 1087 | 109 |

**Chattiest Teams**

|  |  |
| --- | --- |
| Teams | Number of Chats |
| 82 | 1324 |
| 185 | 1036 |
| 112 | 957 |

By analysis, on searching the team that the TOP 3 users belong, was concluded that none of the TOP 3 chattiest users belong to neither the TOP 3 chattiest teams. A good approach using this information is to make specific ads to the chattiest teams (the most clicked ads can appear here). And make specific ads to the chattiest users (the most relevant ads, which have specific public).

**5.4.6 How Active Are Groups of Users?**

The first step to find the cluster coefficient of the top 3 users is to identify the list of nodes related to them.

Below is the query:



Substitute the “TOP 3 USER ID” by the id of the user (394, 2067, 209). The second step was to find the number of edges of each neighbor node from the User.

Below is the query:

****

Substitute “LIST” by the list of neighbor nodes from the specified User.

To calculate the cluster coefficient the formula used was:

(Total Edges from neighbors)/(Total of neighbors node)\*(Total number of neighbors node – 1)

**Most Active Users (based on Cluster Coefficients)**

|  |  |
| --- | --- |
| **User ID** | **Coefficient** |
| 209 | (20/5\*4)/100=1.00 |
| 2067 | (28/6\*5)/100=0.93 |
| 394 | (07/4\*3)/100=0.58 |

The only cluster that has full connectivity is the cluster of the user 209. This node has full connection, and high probability to gain more connections, which mean, high potential to show the most paid ads.

**5.5 Recommended Actions**

By analysis, we could conclude that there are three different ways to increase Eglence Inc. revenue.

* The first recommendation is to create some item packs or special offers based on the platform the user is playing the game, as we saw previously, iPhone users has a potential to spent money with expensive items and more times than users from other platforms, so special offers to iPhone is a good way to increase revenue.
* The second recommendation is to target ads to users based on their behavior or even make pack of items from the most purchased by some kind of user. We can use the information gathered from clustering, for example, user with many clicks, many sessions, but few purchases are a better target to show ads, users with many session, many purchases, but a few clicks in the game, are a better target to sell expensive items.
* The third recommendation is to take the users that has more interaction in the chats (the most influential) to show most paid ads inside these chats.

These three approaches are good choices to increase Eglence Inc. revenue using ads and purchases as source.

**CHAPTER 6**

**Possible Advantages**

* It helps to analyze user’s playing and purchasing behavior.
* Increasing the Overall revenue.
* Allows us in finding the users who are more influential than others.
* If the company wants to expand its game to a different country then we can predict the total number of users in that nation who will spend more than $5 or less than $5 with an accuracy of 88%.
* Helps us to pin-point the chat items that are mostly talked about in the chat between users in the game.
* How geography plays a role in the quantity of the users can also be analyzed further.
* Those users who belong to developed countries tend to spend more than those who belong to developing countries.

**References**

1. [www.coursera.org](http://www.coursera.org)
2. <https://docs.splunk.com/Documentation/Splunk/6.1.4/SearchReference/Commonstatsfunctions>
3. [www.stackoverflow.com](http://www.stackoverflow.com)
4. [www.neo4j.com](http://www.neo4j.com)
5. [www.knime.com](http://www.knime.com)
6. <https://community.cloudera.com/>