Technical Appendix
Catch the Pink Flamingo Analysis

Produced by: <Ruchi Sahu>

Acquiring, Exploring and Preparing the Data

Data Exploration

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	Records of player clicks on an advertisement	Timestamp: when the click occurred. txId: a unique id (within ad-clicks.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	Records of all in app purchase entries.	timestamp: when the purchase was made. txId: a unique id (within buy-clicks.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	Records of each user player in 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 records of 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

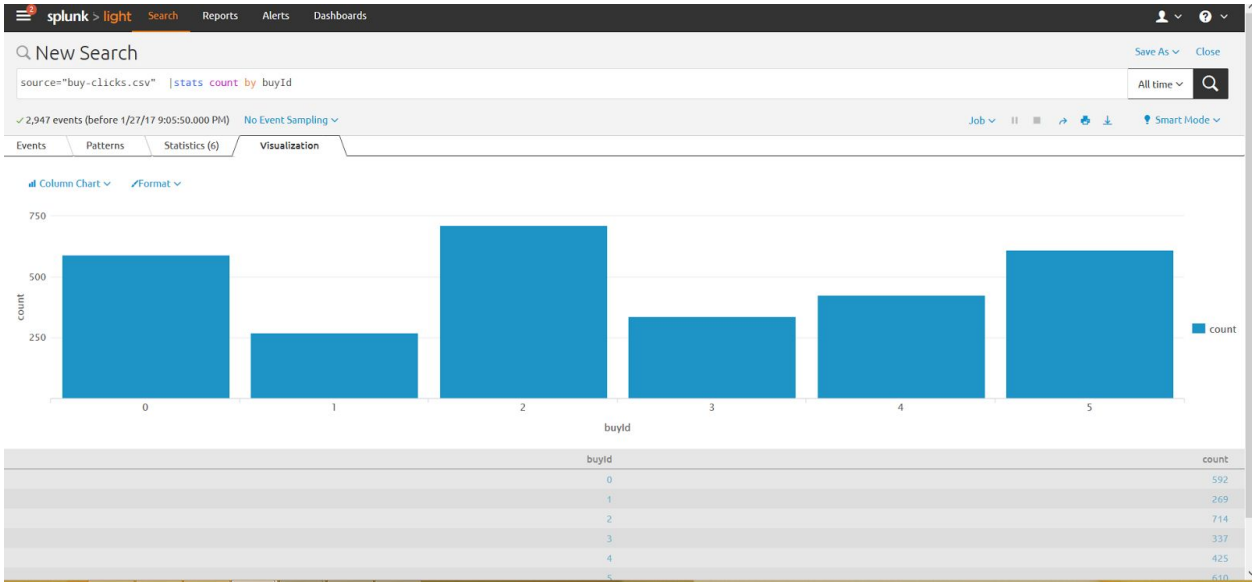
		currentLevel: the current level of the team
user-session.csv	Each line in this file describes a user session, which denotes when a user starts and stops playing the game. Additionally, when a team goes to the next level in the game, the session is ended for each user in the team and a new one started.<Fill in short phrase>	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
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

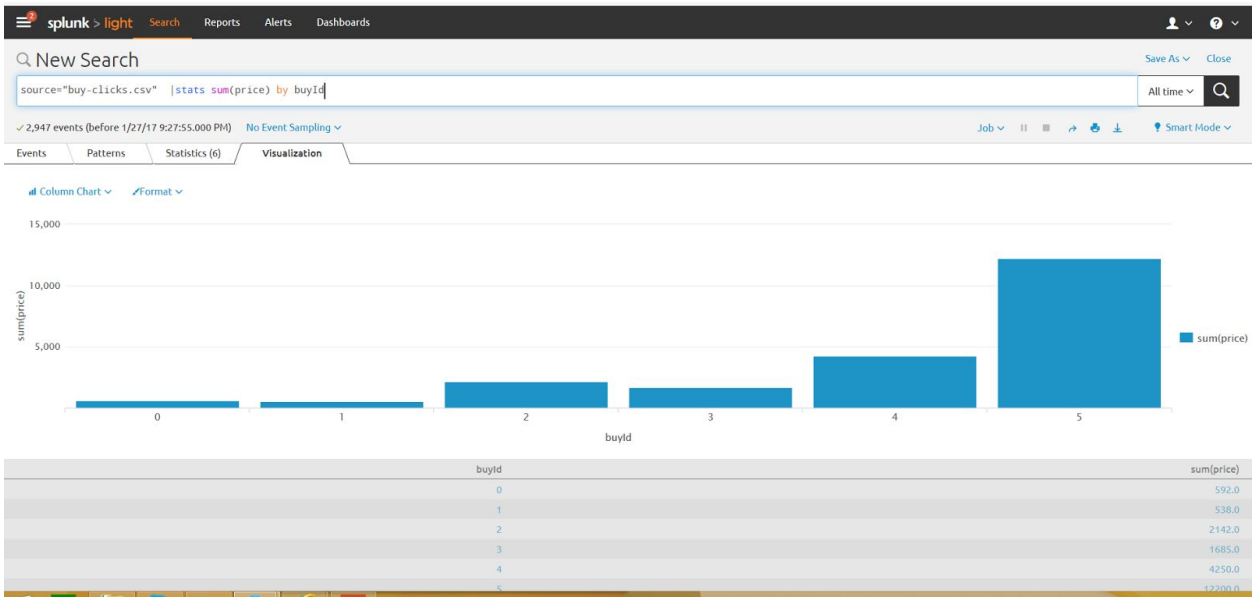
Aggregation

Amount spent buying items	21407
# Unique items available to be purchased	6

A histogram showing how many times each item is purchased:

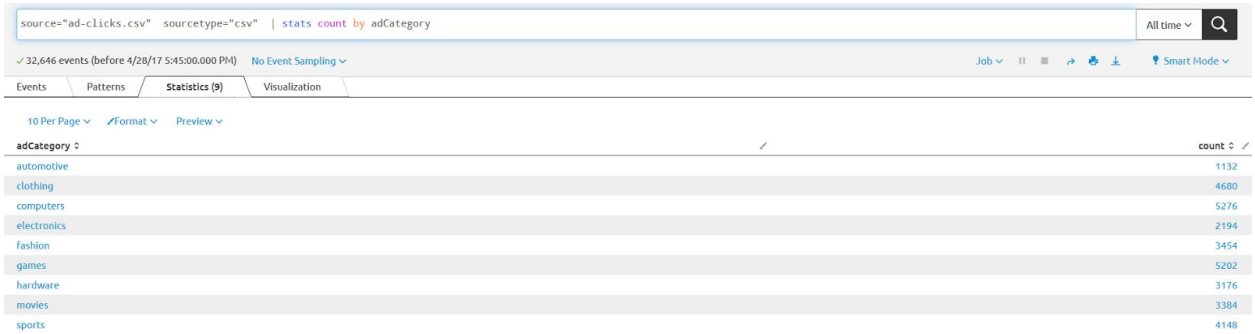


A histogram showing how much money was made from each item:



Filtering

A histogram showing how many times each category of advertisement was clicked-on:



The following table shows the total amount of ad-click revenue for a set of specific values based on the advertisement category. All non-listed categories generate .25 revenue.

Scenario #	Electronics	Fashion	Automotive	Total Revenue
1 - even	0.50	0.50	0.50	9221.5
2 - uneven	0.55	0.60	0.55	9606.7

Data Classification Analysis

Data Preparation

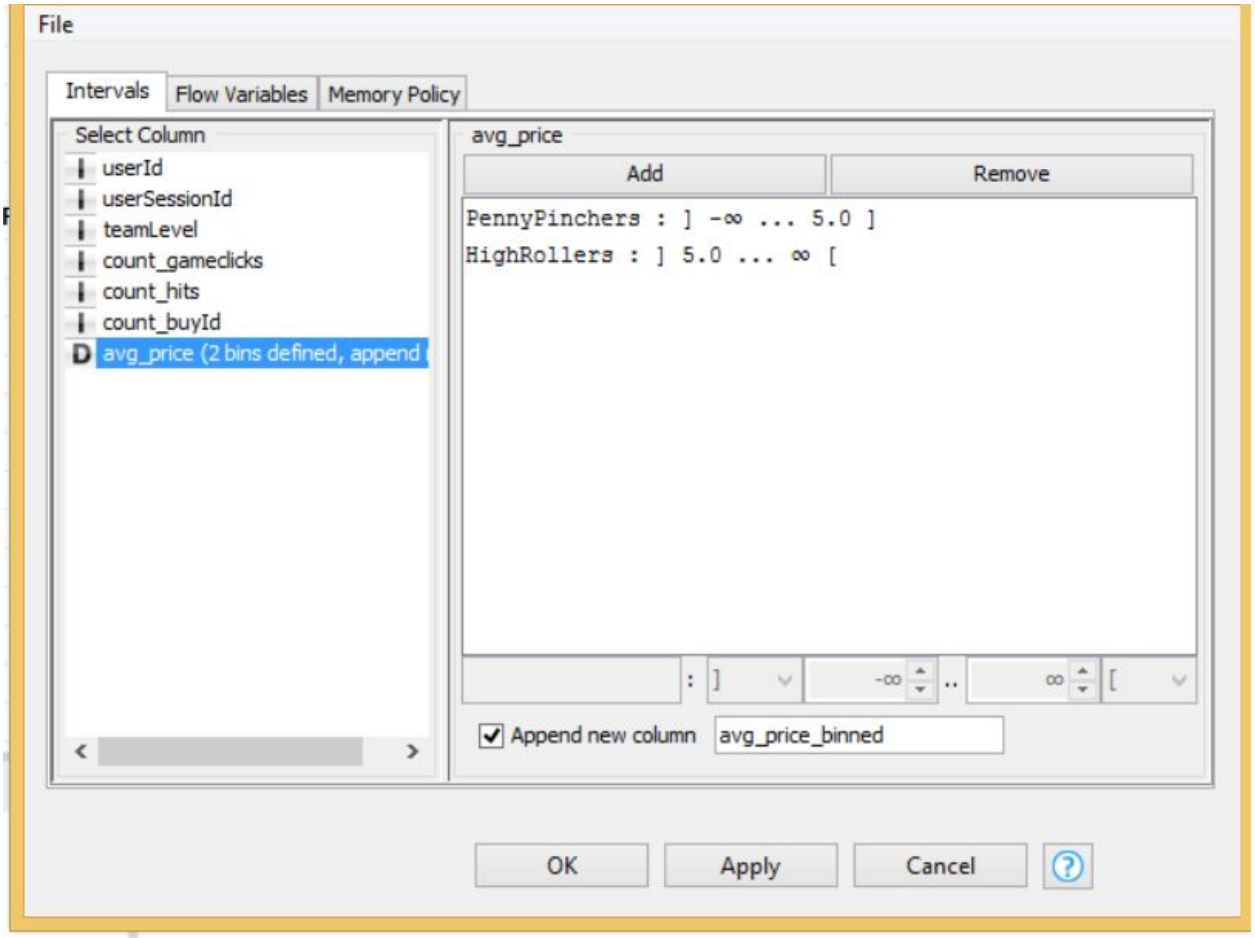
Analysis of combined_data.csv

Sample Selection

Item	Amount
# of Samples	4619
# of Samples with Purchases	1411

Attribute Creation

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



The new attribute avg_price_binned is added,it enabled the analysis of the player's spending behavior.The Numeric Binner nodes use two bins: the PennyPincher bin has a range of less or equal to $5.00.The HighRoller bin has a range of $5:00 and more.

The creation of this new categorical attribute was necessary because it will facilitate the creation of a decision tree that can separate those who buy big ticket items from those who buy inexpensive items.This will allow us to identify these two groups for further analysis and marketing.

The creation of this new categorical attribute was necessary because <Fill in 1-2 sentences>.

Attribute Selection

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

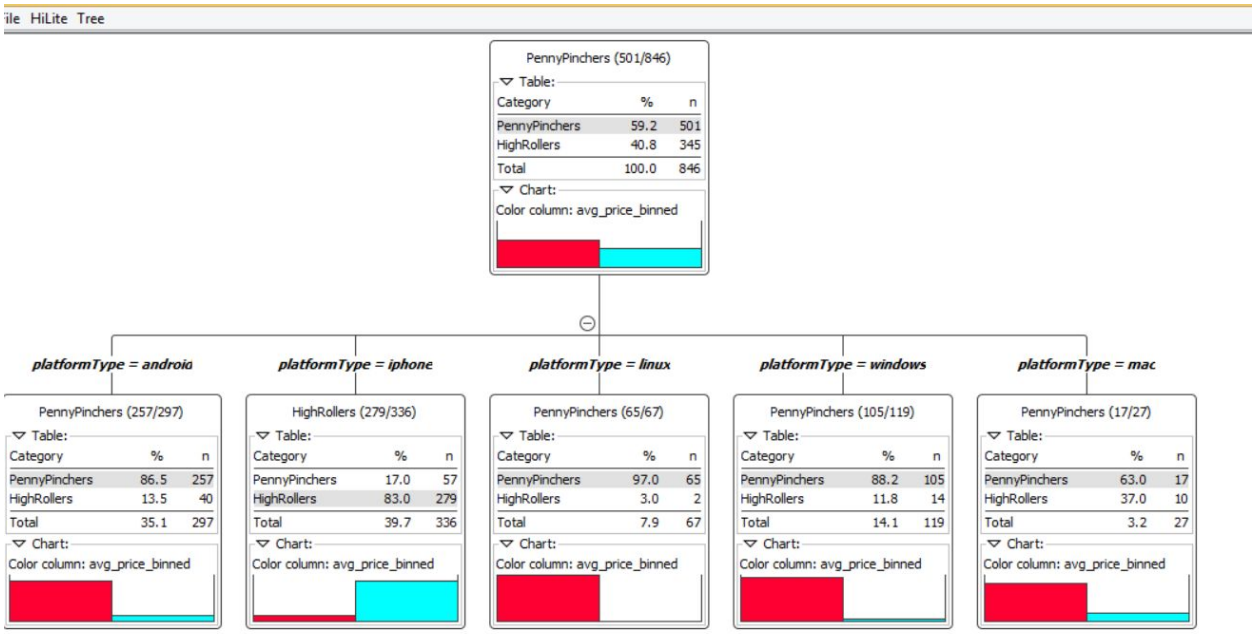
Attribute	**Rationale for Filtering**
userId	Each user has a unique Id,this will not help us to predict if a new user will be a big spender or not.
userSessionId	Each user has a unique Id but a previous session Id will not help us predict if a new session will be related to spending.
avg_price	We have binned the avg_price into HighRollers and PennyPinchers,so we don't want to keep this original class attribute.
<Optional Fill in>	<Optional Fill in 1-3 sentences>

Data Partitioning and Modeling

The data was partitioned into train(60%) and test(40%) 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 we avoid overfitting and can calculate a level of confidence in predicting the outcome of new instances of game players.

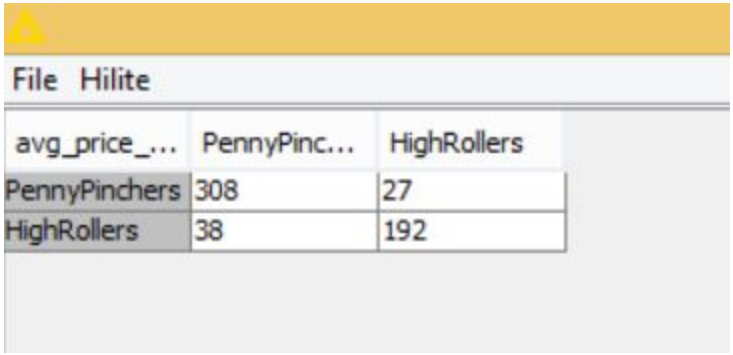
When partitioning the data using sampling, it is important to set the random seed because… it ensures to get reproducible results upon re -executions.If there is no fixed seed defined, a new random seed is taken for each execution.

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



Evaluation

A screenshot of the confusion matrix can be seen below:

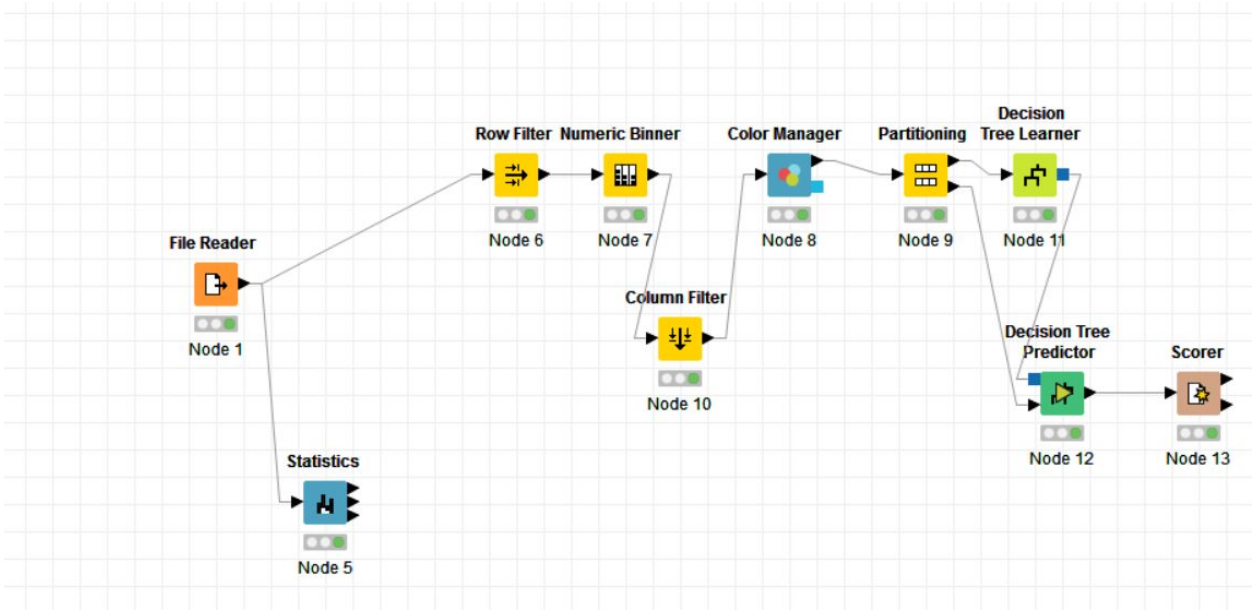


Confusion matrix provide comparison between actual and predicted values of the test attribute. Out of the test data which is 40% of the total 1411. Our model predicted:

- 308 PennyPinchers were correctly predicted but
- 27 PennyPinchers were incorrectly predicted as HighRollers.
- 192 HighRollers were correctly predicted but
- 38 HighRollers were incorrectly predicted as PennyPinchers.

Analysis Conclusions

The final KNIME workflow is shown below:



What makes a HighRoller vs. a PennyPincher?
From the analysis it appears that plateformType predicts HighRoller vs PennyPinchers.iphone users are more likely to be HighRollers.

What makes a HighRoller vs. a PennyPincher? From the analysis it appears that plateformType predicts HighRoller vs PennyPinchers.iphone users are more likely to be HighRollers.
What makes a HighRoller vs. a PennyPincher? From the analysis it appears that plateformType predicts HighRoller vs PennyPinchers.iphone users are more likely to be HighRollers.
What makes a HighRoller vs. a PennyPincher? From the analysis it appears that plateformType predicts HighRoller vs PennyPinchers.iphone users are more likely to be HighRollers.

Clustering Analysis

Attribute Selection

Attribute	Rationale for Selection
Total game_clicks	Total number of clicks by each user
Total ad_click	Total number of clicks on ads
Revenue per user	Amount of money earned from each user spending on their purchases

Training Data Set Creation

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

	totalAdClicks	totalGameClicks	revenue
0	44	716	21.0
1	10	380	53.0
2	37	508	80.0
3	19	3107	11.0
4	46	704	215.0

Dimensions of the training data set (rows x columns) : 543 X 3

# of clusters created: 3

Cluster Centers

Cluster #	Cluster Center
1	[25.12037037, 362.50308642, 35.35802469]
2	[32.05, 2393.95, 41.2]

3	[36.47486034, 953.82122905, 46.16201117]

These clusters can be differentiated from each other as follows:

One Cluster is centered at array([25.12037037, 362.50308642, 35.35802469])

The second Cluster is centered at array([32.05, 2393.95, 41.2])

The third Cluster is centered at array([36.47486034, 953.82122905, 46.16201117])

First number (field1) in each array refers to amount of game-clicking per user ,second no in array refer to ad_clicks and the third number is the cost on this game per user.

Compare the 1st number of each cluster to see how differently users in each cluster behave when it comes to game_click.

Compare the 2nd number of each cluster to see how differently users in each cluster behave when it comes to buying stuff.

Recommended Actions

Action Recommended	**Rationale for the action**
Release much more in-app ads to Increase game's revenue	Company can work on more tools to catch flamingo that will cost more to user. by providing more products to "high level spending user.
promotion to the users	Company can providing some fixed pay packages or promotion to users, in order to stimulate consumption.

Graph Analytics Analysis

Modeling Chat Data using a Graph Data Model

This chat data model is created using Neo4j graphical tool from six csv files.The data contains information regarding users,team,chat session,chat item and timestamp to support node creation.It provide data to build relationships for creating chats,joining teams,leaving teams, mention teams and responding to chats.The analysis can help Eglence by targeting the user for revenue generation in the Flamingo game.

Creation of the Graph Database for Chats

Describe the steps you took for creating the graph database. As part of these steps

i) Write the schema of the 6 CSV files

Chat_create_team_chat.csv: userid, teamid, TeamChatSessionID, timestamp

Chat_item_team_chat.csv: userid, TeamChatSessionID, chatitemid timestamp

Chat_join_team_chat.csv: userid, TeamChatSessiionID, timestamp

Chat_leave_team_chat.csv:userid, TeamChatSessionID, timestamp

Chat_mention_team_chat.csv: ChatItem, userid, timestamp

Chat_respond_team_chat.csv: chatid1, chatid2, timestamp

ii) Explain the loading process and include a sample LOAD command

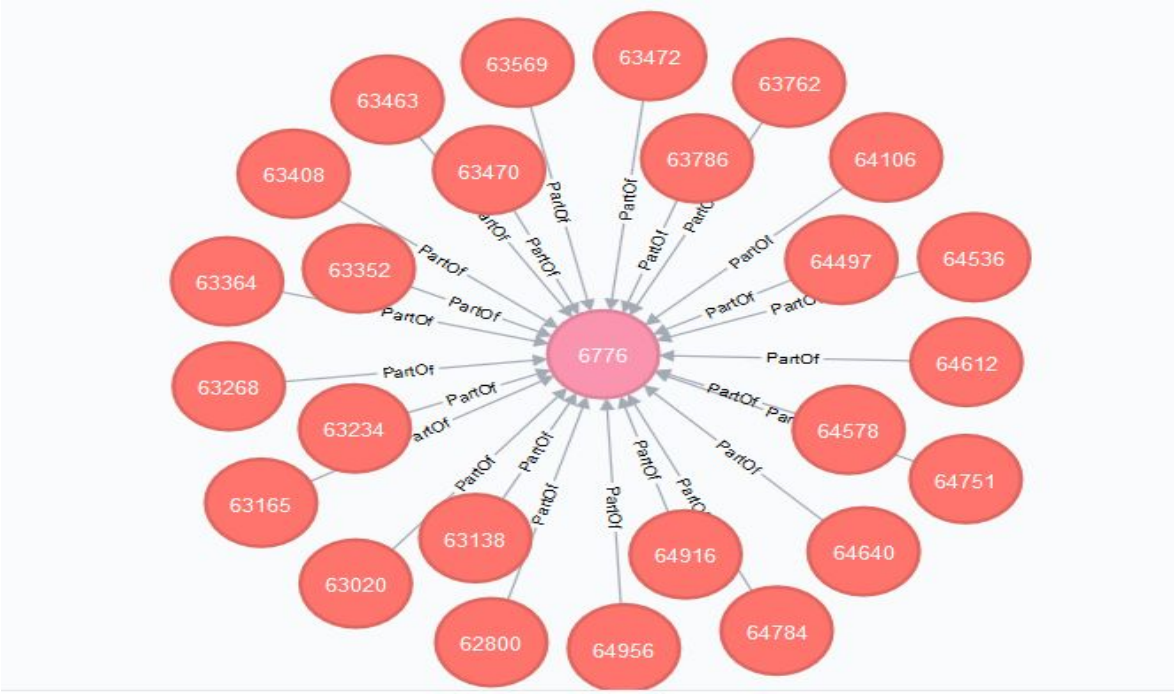
Six csv files are loaded into Neo4j one-by-one:

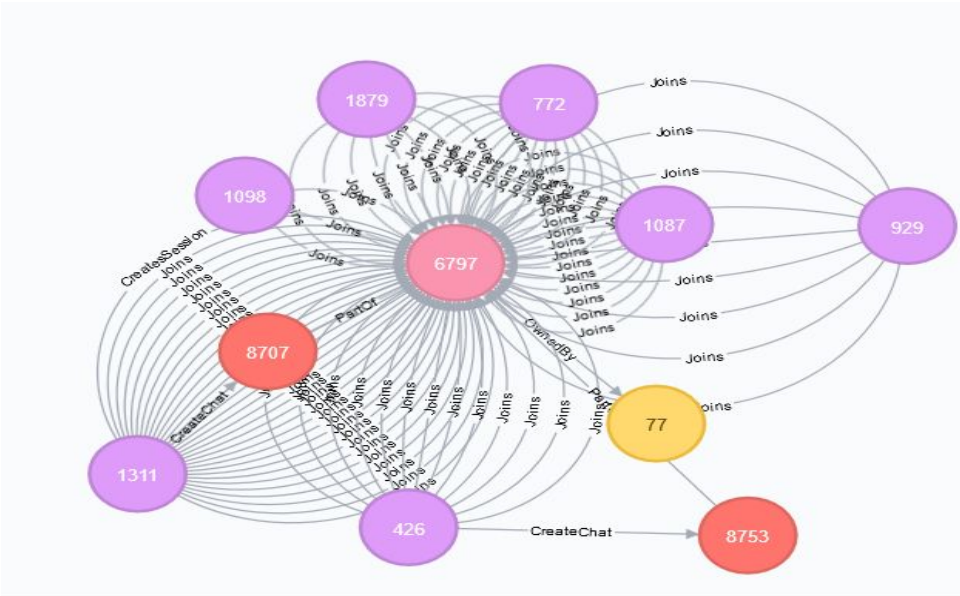
- Loading Chat_create_team_chat.csv,->three nodes and two edges are created. Nodes are user, Team and TeamChatSession. Edges are CreateSession and OwnedBy.
- Loading Chat_join_team_chat.csv->two nodes and one edge are created. Nodes are user and TeamChatSession. Edge is Join;
- Loading Chat_leave_team_chat.csv->two nodes and one edge are created. Nodes are user and TeamChatSession. Edge is Leave;
- Loading Chat_iteam_team_chat.csv-> three nodes and two edges are created. Nodes are user, TeamChatSession and chatitem. Edges are CreateChat and part of;

- Loading Chat_mention_team_chat.csv,->two nodes and one edge are created. Nodes are chatiteam user. Edge is Mentioned;
- Loading Chat_respond_team_chat.csv-> two nodes and one edge are created. Nodes are chatiteam1 chatitem2. Edge is ResponseTo.

```
LOAD CSV FROM "file:///C:/coursera/data/chat_create_team_chat.csv" AS row
MERGE (u: User {id: toInt(row[0])}) MERGE (t: Team {id: toInt(row[1])})
MERGE (c: TeamChatSession {id: toInt(row[2])})
MERGE (u)-[:CreatesSession{timeStamp: row[3]}]->(c)
MERGE (c)-[:OwnedBy{timeStamp: row[3]}]->(t);
```

iii) Present a screenshot of some part of the graph you have generated. The graphs must include clearly visible examples of most node and edge types. Below are two acceptable examples. The first example is a rendered in the default Neo4j distribution, the second has had some nodes moved to expose the edges more clearly. Both include examples of most node and edge types.





Displaying 35 nodes, 101 relationships (completed with 2 additional relationships).

Finding the longest conversation chain and its participants

Report the results including the length of the conversation (path length) and how many unique users were part of the conversation chain. Describe your steps. Write the query that produces the correct answer.

The longest conversation chain in the chat data using the "ResponseTo" edge label = 10 There are 10 ChatItems nodes and 9 ResponseTo edges in the conversation.

1: Get the longest length of the ResponseTo edge

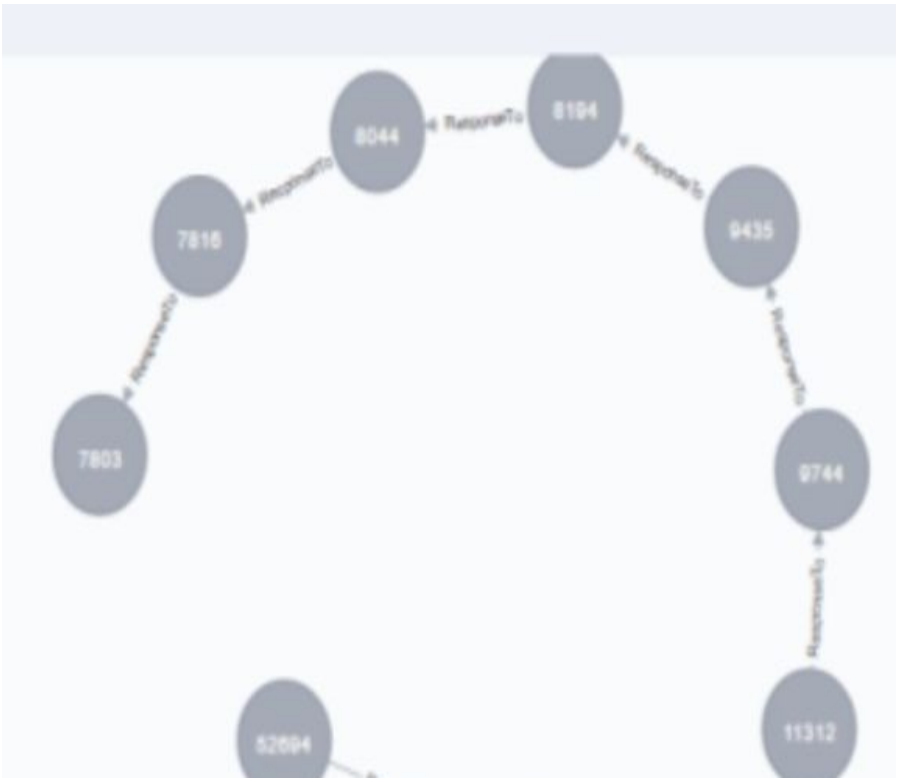
match p=(i:ChatItem)-[:ResponseTo*]->(j:ChatItem) return p, length(p) AS

LenLongestPath, extract(n in nodes(p)|n.id) AS longestPath order by length(p) desc limit 1

Result: There are 5 users in this longest conversation

2: Get the users who are in the longest path

match p=(i:ChatItem)-[:ResponseTo*]->(j:ChatItem) where length(p)=9 with extract(n in nodes(p)|n.id) AS LongestPath match (u:User)-[:CreateChat]->(k:ChatItem) where k.id in LongestPath return count(distinct u) as NumUsers



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

2: Get the users who are in the longest path

match p=(i:ChatItem)-[:ResponseTo*]->(j:ChatItem) where length(p)=9 with extract(n in nodes(p)|n.id) AS LongestPath match (u:User)-[:CreateChat]->(k:ChatItem) where k.id in LongestPath return count(distinct u) as NumUsers

Users	Number of Chats
394	115
2067	111
209	109

Chattiest Teams

Teams	Number of Chats
82	1324
185	1036
112	957

How Active Are Groups of Users?

Report the top 3 most active users in the table below.

Most Active Users (based on Cluster Coefficients)

User ID	Coefficient
209	0.95238
554	0.90476
1078	0.8

Recommended Actions

Finally, make recommendations to Eglence, Inc. and include examples of how your findings support them. Include this information in Slide 6 of your final presentation.

More tools to iPhone users and cheap tools for android and windows user:

Understand the users who play the games is critically important to enhance the revenue .

- According to the decision tree classification, it reflects that most users which on the platform iPhone are HighRollers, so offering more products to them can increase our revenue.

- But significant number of users do not play on iphone so provide a low spend tools to android and windows .

- Increase the complexity of game and introduce more tools to deal with the complexity.If user will buy these tools can finish work immediately with tools otherwise it will take more days,if user wanna wait.

- Add more ads to increase Revenue.