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	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	line is added to this file when a player makes an inapp 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
User.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 twoletter 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

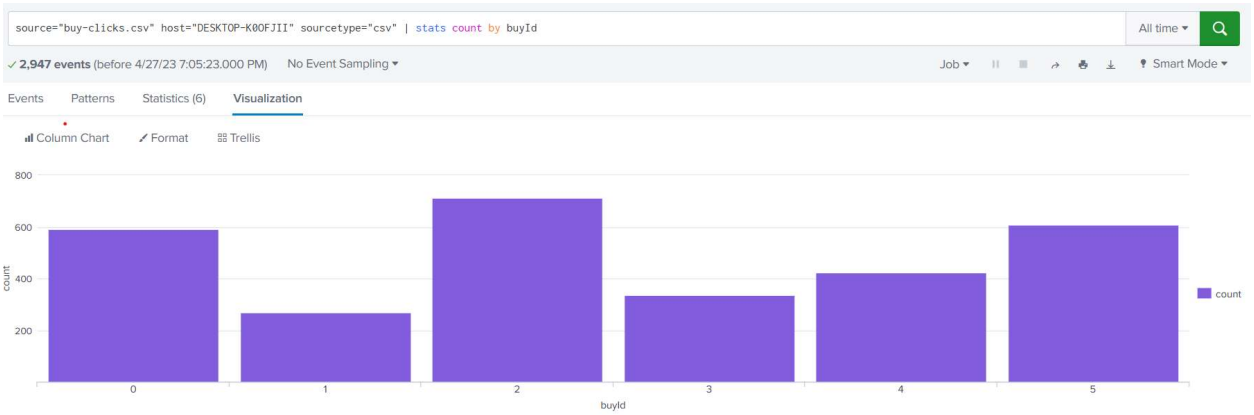
		when the team was created teamEndTime: the timestamp when the last member left the team strength: a measure of team strength, roughly corresponding to the success of a team currentLevel: the current level of the team
Team-assignments.csv	A line is added to this file each time a user joins a team. A user 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	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

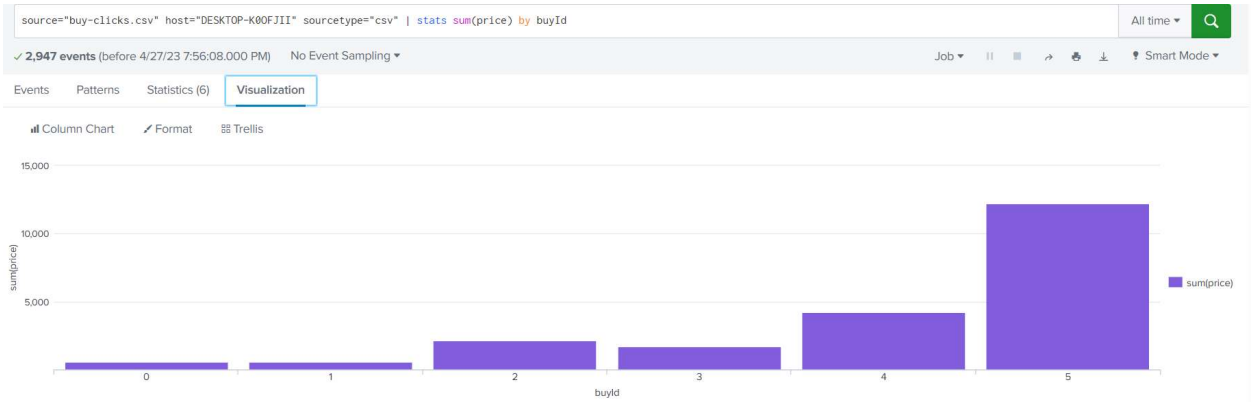
Aggregation

Amount spent buying items	21407
Number of unique items available to be purchased	6

A histogram showing how many times each item is purchased:

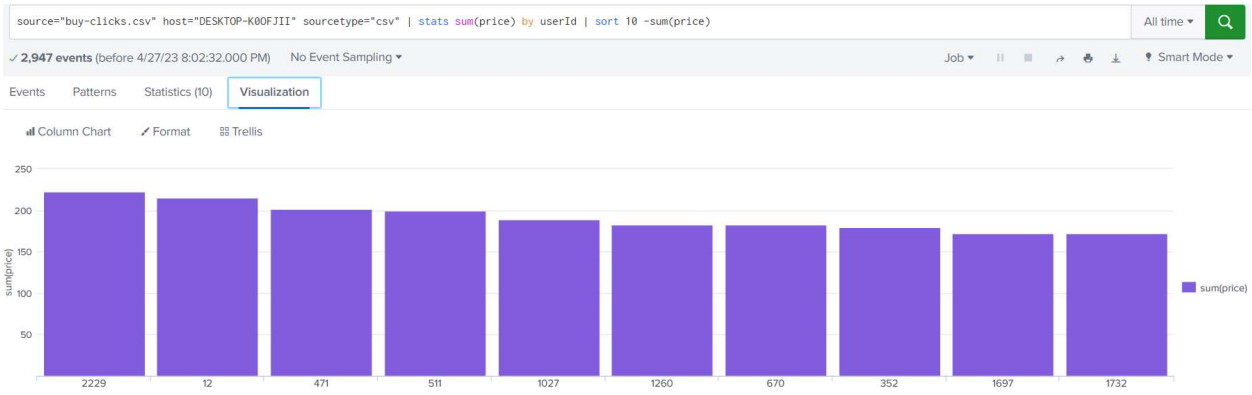


A histogram showing how much money was made from each item:



Filtering

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



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	0.11596958
2	12	iphone	0.130681
3	471	iphone	0.1450381

Data Preparation

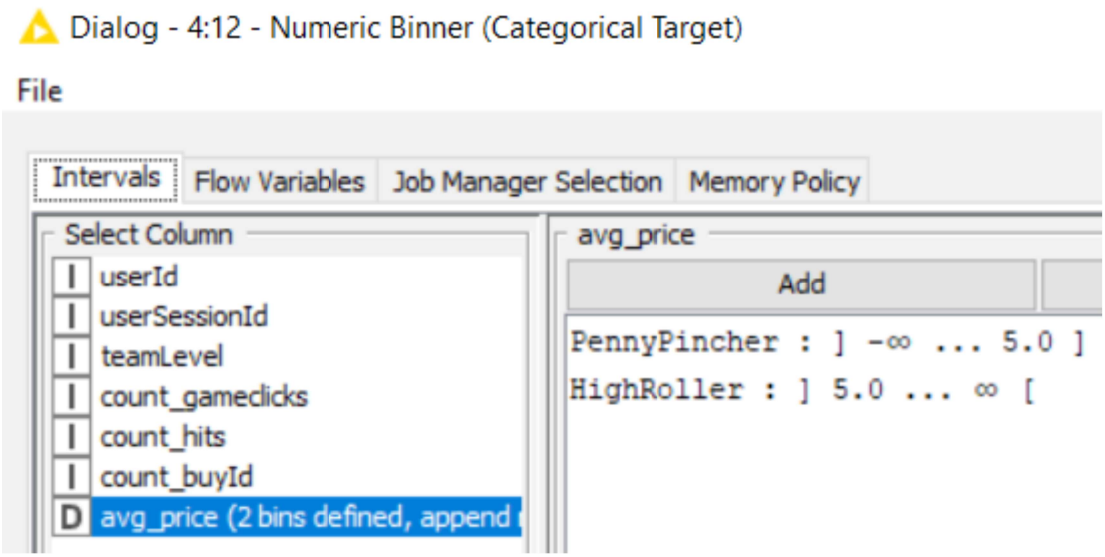
Analysis of combined_data.csv

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:



A Categorical attribute named *'User_Category'* is created with given conditions. PennyPincher are those with avg_price <= 5, and HighRoller are those with avg_price > 5.

The creation of this new categorical attribute was necessary because solving a classification problem requires the target variable to be categorical but avg_price is numerical.

Attribute Selection

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

Attribute	Rationale for Filtering
avg_price	Target variable is derived from this attribute, so it can not become part of model features otherwise we'd see model overfitting

userId	Id field adds no information hence can be excluded from modelling
userSessionId	Id field adds no information hence can be excluded from modelling
<Optional Fill in>	<Optional Fill in 1-3 sentences>

Data Partitioning and Modeling

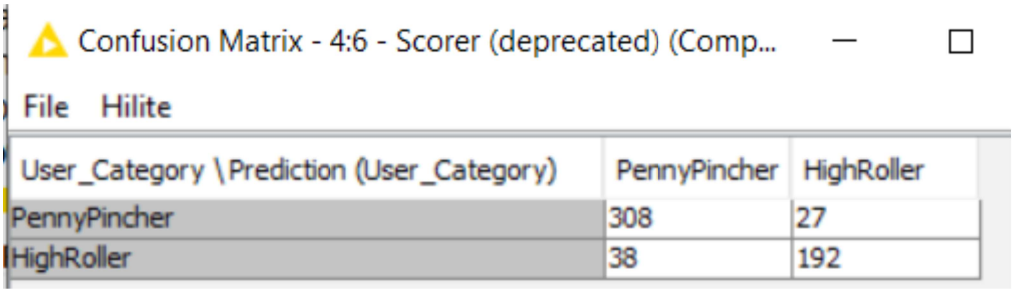
The data was partitioned into train and test datasets.
The <training> data set was used to create the decision tree model.
The trained model was then applied to the <test> dataset.
This is important because we need to evaluate the performance of our model on unseen dataset.

When partitioning the data using sampling, it is important to set the random seed because we want to be able to regenerate the same results while comparing with others.

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

Evaluation

A screenshot of the confusion matrix can be seen below:

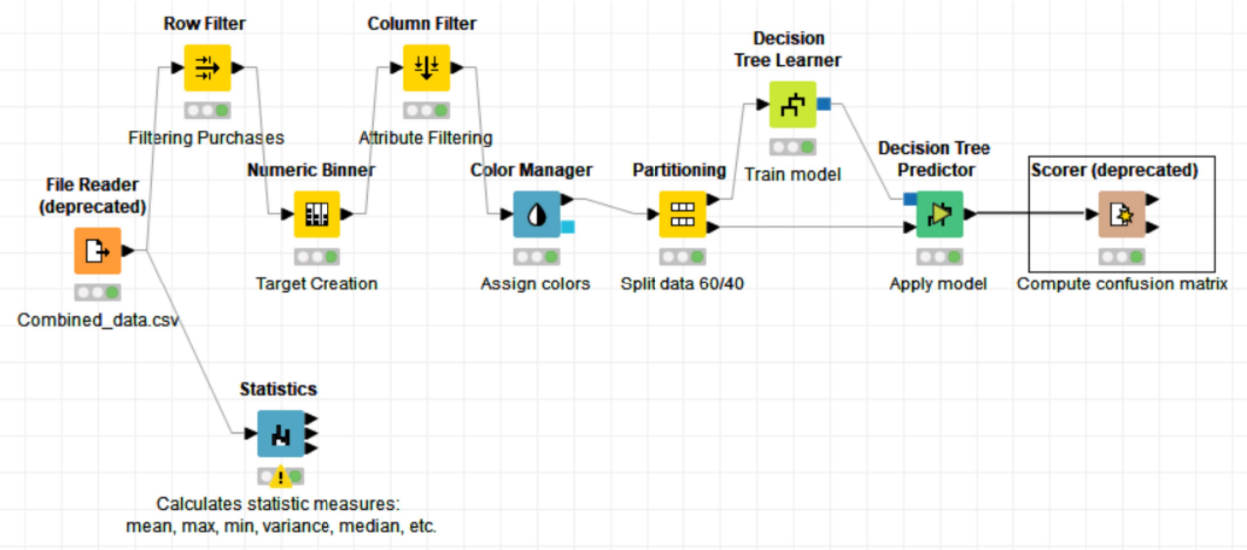


As seen in the screenshot above, the overall accuracy of the model is <88.496%>

38 HighRollers were incorrectly predicted as PennyPincher by the Decision Tree model.
Likewise, 27 PennyPinchers were incorrectly predicted as HighRolers.
Decision Tree model correctly predicts 308 PennyPincher and 192 HighRoller.
Overall, 500 accurate predictions with 65 incorrect predictions.

Analysis Conclusions

The final KNIME workflow is shown below:



What makes a HighRoller vs. a PennyPincher?
Users with iphone platformType are mostly HighRoller.
Users with Linux platformType are all (but two) PennyPinchers.
Mac users are more inclined to spend more than linux/windows/android.

Specific Recommendations to Increase Revenue
1. Iphone and Mac users may be targeted with expensive items to generate more revenue out of them
2. Other platform users can be targeted with a wider variety of cheaper options to generate more revenue from them.

Attribute Selection

features_used = ["totalAdClicks",”total_spend”, “total_gameclicks"]

Attribute	Rationale for Selection
totalAdClicks	It captures users’ inclination to click on ads which is important while targeting
total_spend	It captures users’ purchase behavior and tell us more about the budget
total_gameclicks	It captures users’ playing behavior by counting the number of clicks on the grids in a game.

Training Data Set Creation

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

```
trainingDF.head(5)
```

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

Dimensions of the final data set: (543,3)
# of clusters created: 3

Cluster Centers

The code used in creating cluster centers is given below:

print(my_kmmodel.centers)

Cluster centers formed are given in the table below

Cluster #	Center
1	[36.44134078, 46.96648045, 926.11731844]
2	[24.98746082, 35.06583072, 357.95924765]
3	[32.35555556, 39.42222222, 2310.64444444]

Adclicks Revenue Gameclicks

These clusters can be differentiated from each other as follows:

Cluster 1 is different from the others in that users with the highest revenue and adclick are not the ones who are playing the most but an intermediate result in game clicks.

Cluster 2 is different from the others in that the users who play the less also produces the less ad revenue and click count

Cluster 3 is different from the others in that the users who play the most are not the ones who produce the most revenue, the revenue in the middle along with the ad click count

Below you can see the summary of the train data set:

```
trainingDF.describe()
```

	totalAdClicks	total_spend	total_gameclicks
count	543.000000	543.000000	543.000000
mean	29.373849	39.349908	707.077348
std	15.216343	41.221737	612.140845
min	1.000000	1.000000	8.000000
25%	16.000000	10.000000	319.000000
50%	30.000000	25.000000	555.000000
75%	42.000000	55.000000	877.500000
max	67.000000	223.000000	4599.000000

Recommended Actions

Action Recommended	Rationale for the action
Send ads to users from their favorite ads category to increase chances of ad clicks	Since, total amount spent (or revenue per user) is directly proportional to the number of ads he/she clicked, sending each user ads from their favorite ads category will enable purchase and increase company's revenue
Encourage players in Cluster 2 to spend more time playing the game	Players in Cluster 2 are 5x more likely to click on ads (25/358 vs 32/2310) while playing the game than players in Cluster 3. (adclicks/gameclicks). This recommendation will generate more adClicks from Cluster 2 Players and hence more Revenue for Eglence.Inc
Increase prices for ads items shown to Cluster 3 Players	Cluster 3 Players have lowest (1.21) revenue per ad clicked i.e. ratio of total_spend and totalAdClicks. Since they are playing the game the most (most gameclicks), company can generate more revenue from them by increasing prices of ads shown to these players.

Graph Analytics

Modeling Chat Data using a Graph Data Model

Graph Data Model is perfectly suited for evaluating or analyzing various hypotheses based on Chat Data generated out of Catch the pink flamingo game.
There are following node types and edge types in the graph model of Chat Data.

- Node Types: User node, Team node, Chat Item node, Team Chat Session node
- Edge Types: Joins, Leaves, Mentioned,CreateChat, CreateSession, OwnedBy, ResponseTo, PartOf

CreateSession edge is created from User node to TeamChatSession node when a User initiates a TeamChatSession.
OwnedBy edge is created from TeamChatSession node to Team node whenever any User from a Team initiates a TeamChatSession.
Joins edge is created from User node to TeamChatSession node when a User joins a TeamChatSession.
Leaves edge is created from User node to TeamChatSession node when a User leaves a TeamChatSession.
CreateChat edge is created from User node to ChatItem node when a User creates a ChatItem.
PartOf edge is created from TeamChatSession node to Team node for all TeamChatSessions belonging to a Team
Mentioned edge is created from ChatItem node to User node when a User is mentioned in a ChatItem.
ResponseTo edge is created between two ChatItem nodes when one ChatItem is a response to another ChatItem

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:

File Name	Description	Example
1. chat_create_team_chat.csv	A line is added to this file when a player creates a new chat with their team	userid, teamid, TeamChatSessionID, timestamp 559,48,6288,14567
2. chat_item_team_chat.csv	Creates nodes labeled ChatItems. Column 0 is	userid, teamchatsessionid, chatitemid, timestamp

	User id, column 1 is the TeamChatSession id, column 2 is the ChatItem id (i.e., the id property of the ChatItem node), column 3 is the timestamp for an edge labeled "CreateChat". Also create an edge labeled "PartOf" from the ChatItem node to the TeamChatSession node. This edge should also have a timeStamp property using the value from Column 3	1956,6299,6305,1464235803
3. chat_join_team_chat.csv	Creates an edge labeled "Joins" from User to TeamChatSession. The columns are the User id, TeamChatSession id and the timestamp of the Joins edge	Example: userid, TeamChatSessionID, teamstamp 559,6288,12345
4. chat_leave_team_chat.csv	Creates an edge labeled "Leaves" from User to TeamChatSession. The columns are the User id, TeamChatSession id and the timestamp of the Leaves edge	userid, teamchatsessionid, timestamp 1244,6821,1464241204.0
5. chat_mention_team_chat.csv	Creates an edge labeled "Mentioned". Column 0 is the id of the ChatItem, column 1 is the id of the User, and column 2 is the timeStamp of the edge going from the chatItem to the User	ChatItem, userid, timeStamp 6349,2508
6. chat_respond_team_chat.csv	A line is added to this file when a player responds	chatid1, chatid2,timestamp 6326,6305,21564

	to a chat post	

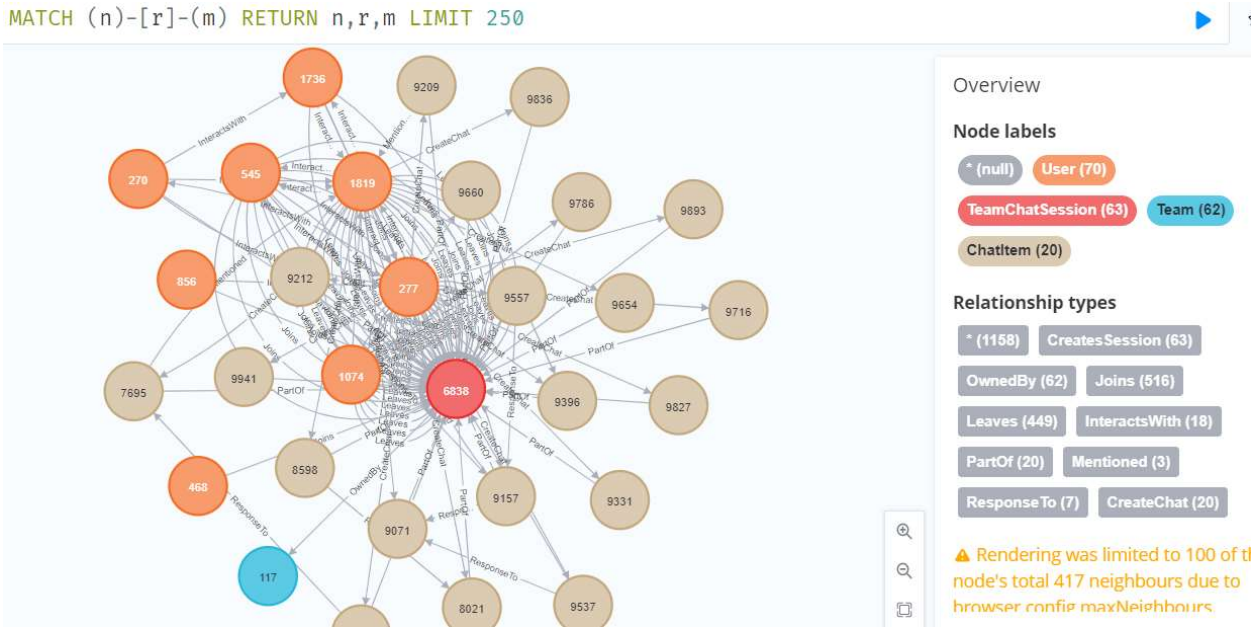
ii) Explain the loading process and include a sample LOAD command:

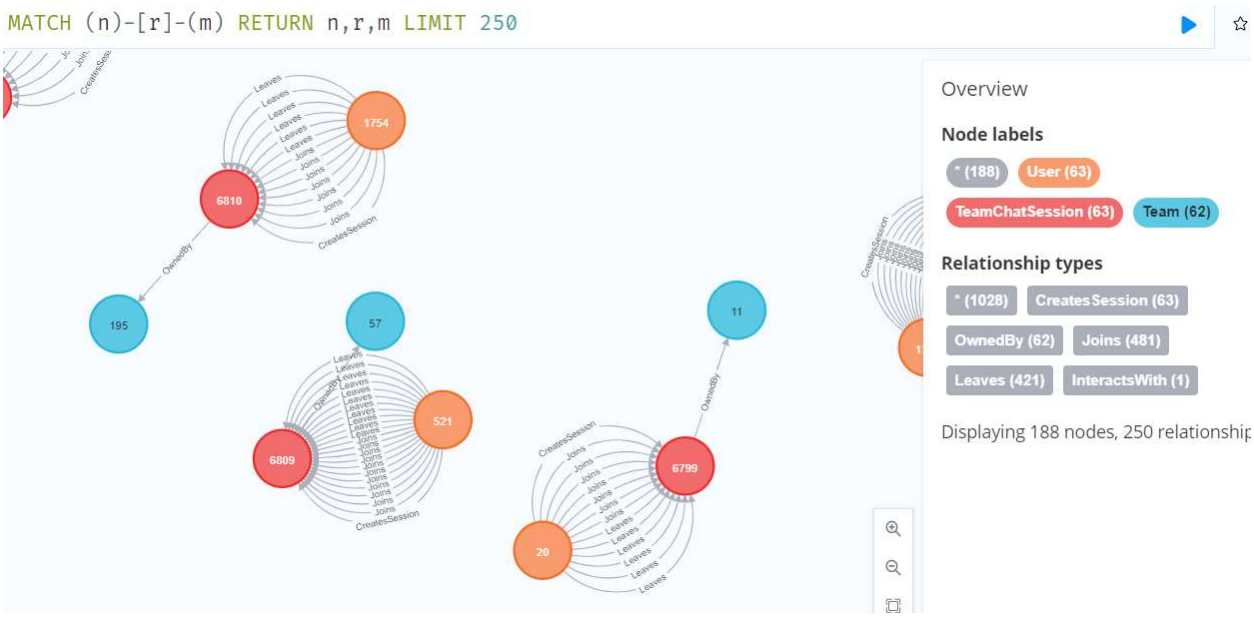
First, we defined the constraints for all the node types to be unique so that they are not duplicated when multiple relationships are created.
Following example snippet shows loading command for chat_join_team_chat.csv file.

```
LOAD CSV FROM "file:///chat_data/chat_join_team_chat.csv" AS row
MERGE (u:User {id: toInteger(row[0])})
MERGE (c:TeamChatSession {id: toInteger(row[1])})
MERGE (u)-[:Joins {timeStamp: row[2]}]->(c)
```

We are reading the CSV file from the import directory, row by row . In each row of this file, first value converted to integer is attributed to User node ID, second value converted to integer is attributed to TeamChatSession ID while third value indicates timestamp of the *Join* relationship i.e. when User ID joined the TeamChatSession ID. Merging commands for all the three values defines the nodes and creates the specified relationship *Joins* between them. Approach is similar in every file.

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 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.





Finding the longest conversation chain and its participants

The longest conversation is 9 nodes long that includes 5 unique Users as part of the conversation.

Following query produces the longest path where we can see the ChatItem ids. Then select the path and count the distinct nodes in second query.

match p=(i)-[:ResponseTo]->(j) return i.id,j.id,length(p) as len order by len desc limit 1*
match p=(i:ChatItem)-[:ResponseTo]->(j:ChatItem)*
where i.id=52694 and j.id=7803
with p
match (u:User)-[:CreateChat]->(k:ChatItem)
where k in nodes(p)
return count(distinct u)

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

Chattiest users are the ones having the most number of relationships of type ‘CreateChat’.

Chattiest Users

Users	Number of Chats
394	115
2067	111
1087	109

Chattiest teams are the ones with the most number of nodes of type ‘ChatItem’ that are part of TeamChatSessions owned by the team.

Chattiest Teams

Teams	Number of Chats
82	1324
185	1036

112	957

User ID 999 is the 6th most chattiest User and is part of Team ID 52 which is the 7th most chattiest Team.

How Active Are Groups of Users?

To find how active the groups of users are, first we created a new edge type *'InteractsWith'* between users which are responding to each other or who mention one another. Then self loops are deleted. Then we compute how dense the neighborhoods are for each of the chattiest Users.

There is only one *InteractsWith* edge between two users even if one interacts with another multiple times via *RespondTo* or *Mentioned* relationships. This is ensured by not duplicating the edge while creating it.

Cluster Coefficient is computed as the number of edges present amongst the neighbors divided by the total number of edges possible amongst them. E.g. if chattiest user has 5 neighbors and there are 15 edges amongst these 5 users (excluding edges with the chattiest user), then cluster coefficient is 15/20 = 0.75

Most Active Users (based on Cluster Coefficients)

User ID	Coefficient
394	1.00
461	1.00
209	0.95