Technical Appendix
Catch the Pink Flamingo Analysis

Produced by: <PRANAV BHARGAVA>

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	A line added to this file whenever an ad is clicked.	adId: Id of clicked ad adCategory: Type of ad clicked on teamId: current team id of user timestamp:when the click occurred txId: Unique id for each click userId: Id of user userSessionId: Id of user session in which user clicked
buy-clicks.csv	A line is added to this file on each app-purchase in Flamingo app.	buyId: Id of purchased item price: Cost of item purchased timestamp: time of click team: current team id of user who made purchase txId: Unique id for each click userId: Id of user userSessionId: Id of user session in which user clicked

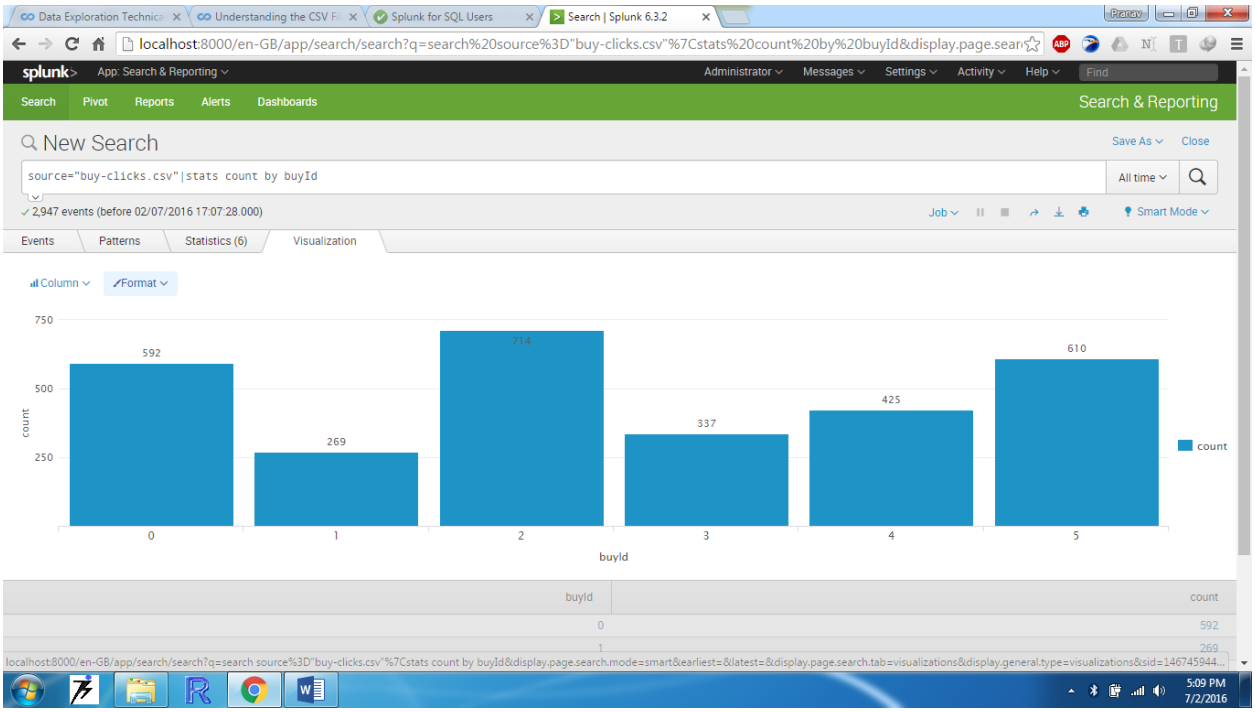
users.csv	This file contains a line for each user in the game.	timestamp: when user first played the game. id: the user id assigned to the user. nick: the nickname chosen by the user. twitter: the twitter handle of the user. dob: the date of birth of the user. country: the two-letter country code where the user lives.
team.csv	This file contains a line for each terminated team 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 the file each time a use joins the team.	time: 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 whenever a team starts or ends a new level.	time: when the event occurred. eventid: a unique id for the event teamid: the id of the team level: the level started or completed eventType: the type of event, either start or end

user-session.csv	Each line describes the user session of a player i.e. when session starts and ends.	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.	time: 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

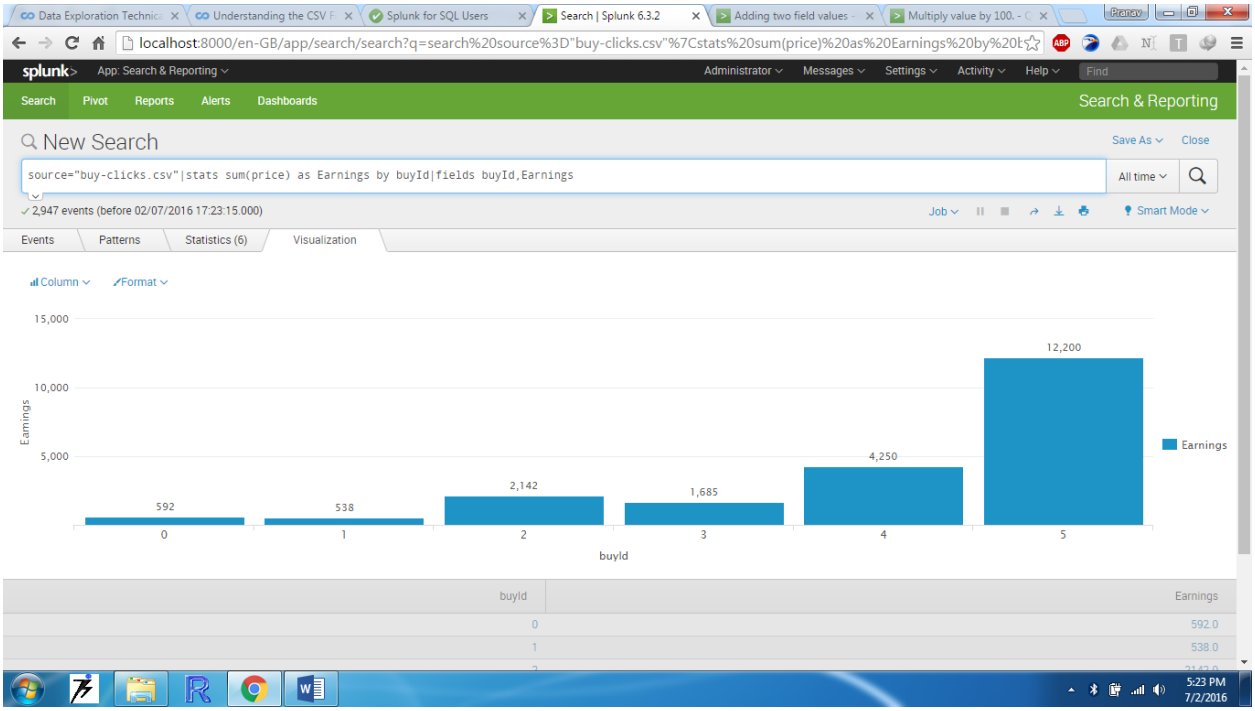
Aggregation

Amount spent buying items	21407.0
# Unique items available to be purchased	6

A histogram showing how many times each item is purchased:

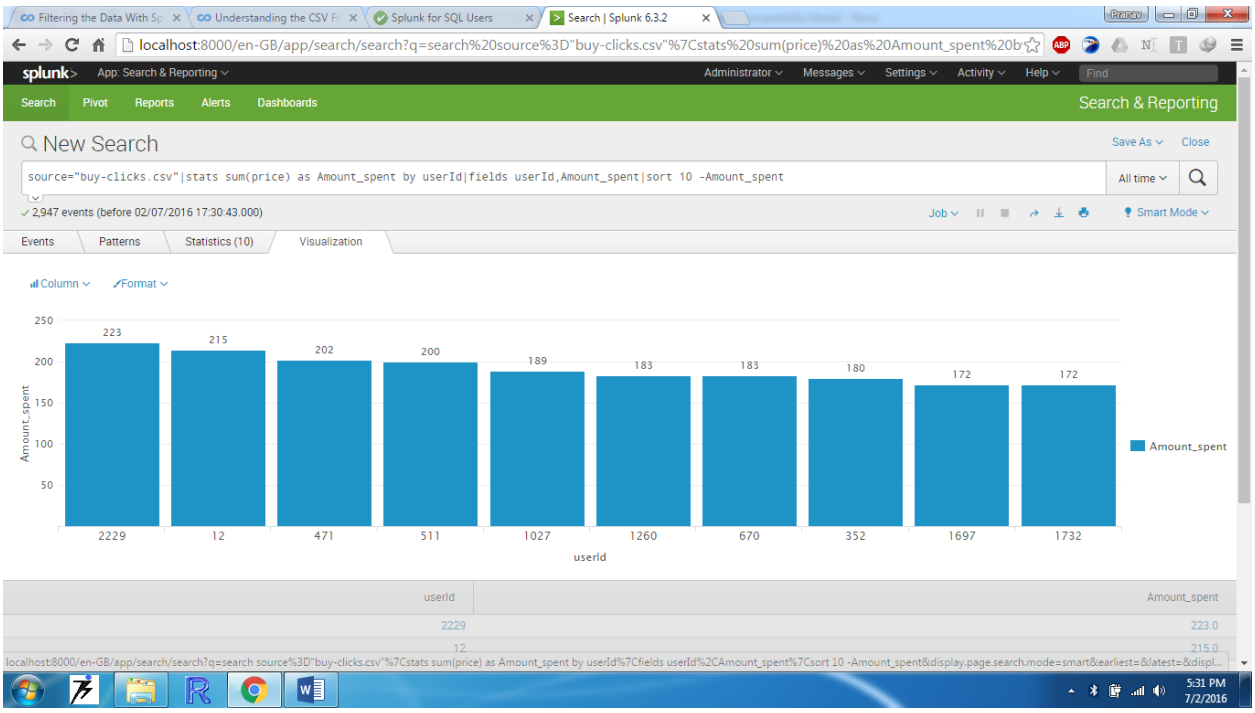


A histogram showing how much money was made from each item:



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	0.115970
2	12	iphone	0.130682
3	471	iphone	0.145038

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

Row ID	userId	userSe...	teamLevel	platfor...	count_...	count_...	count_...	avg_price	avg_price_binned
Row4	937	5652	1	android	39	0	1	1	PennyPinchers
Row11	1623	5659	1	iphone	129	9	1	10	HighRollers
Row13	83	5661	1	android	102	14	1	5	PennyPinchers
Row17	121	5665	1	android	39	4	1	3	PennyPinchers
Row18	462	5666	1	android	90	10	1	3	PennyPinchers
Row31	819	5679	1	iphone	51	8	1	20	HighRollers
Row49	2199	5697	1	android	51	6	2	2.5	PennyPinchers
Row50	1143	5698	1	android	47	5	2	2	PennyPinchers
Row58	1652	5706	1	android	46	7	1	1	PennyPinchers
Row61	2222	5709	1	iphone	41	6	1	20	HighRollers
Row68	374	5716	1	android	47	7	1	3	PennyPinchers
Row72	1535	5720	1	iphone	76	7	1	20	HighRollers
Row73	21	5721	1	android	52	2	1	3	PennyPinchers
Row101	2379	5749	1	android	62	9	1	3	PennyPinchers
Row122	1807	5770	1	iphone	177	25	2	7.5	HighRollers
Row127	868	5775	1	iphone	54	5	1	10	HighRollers
Row129	1567	5777	1	android	27	4	2	4	PennyPinchers
Row131	221	5779	1	iphone	37	2	1	20	HighRollers
Row135	2306	5783	1	android	67	5	1	1	PennyPinchers
Row137	1065	5785	1	iphone	37	5	2	11.5	HighRollers
Row140	827	5788	1	iphone	75	5	1	20	HighRollers
Row150	1304	5798	1	mac	71	9	2	11.5	HighRollers
Row158	1264	5806	1	linux	81	12	1	5	PennyPinchers
Row159	1026	5807	1	iphone	52	10	1	20	HighRollers
Row163	649	5811	1	linux	51	9	1	1	PennyPinchers
Row169	1958	5817	1	android	40	3	1	20	HighRollers
Row172	1300	5820	1	android	58	1	2	3	PennyPinchers
Row186	178	5834	1	iphone	54	6	1	20	HighRollers
Row196	670	5844	1	iphone	38	3	2	20	HighRollers
Row207	208	5855	1	iphone	32	3	1	20	HighRollers
Row210	157	5858	1	iphone	32	2	1	10	HighRollers
Row212	2221	5860	1	iphone	191	18	2	11.5	HighRollers
Row215	471	5863	1	iphone	45	6	2	15	HighRollers
Row218	1234	5866	1	android	46	3	1	10	HighRollers
Row222	371	5870	1	android	53	9	1	3	PennyPinchers
Row232	2146	5880	1	linux	46	7	1	2	PennyPinchers
Row239	935	5887	1	iphone	57	2	1	10	HighRollers
Row241	165	5889	1	iphone	49	3	1	5	PennyPinchers

The attribute is called avg_price_binned. The users who buy with average price above 5 are HighRollers and below or equal to 5 are PennyPinchers. It helps us to categorize users who buy more or buy expensive items from the less spending ones.

The creation of this new categorical attribute was necessary because **it helps determine which users have contributed most to the revenue of the company.**

.

<u>Attribute Selection</u>

The following attributes were filtered from the dataset for the following reasons: **It means I kept these attributes because rest didn't make any special impact.**

Attribute	Rationale for Filtering
userId	It was filtered because it is the unique key for every single user and is thus always required to identify individual user.
platformType	Will help us in analyzing whether certain platform users make majority of “HighRollers” or “PennyPinchers”
avg_price_binned	This is the only dependent variable which is to be predicted and is thus required in the dataset so that predictions can be made later.
<Optional Fill in>	<Optional Fill in 1-3 sentences>

Data Partitioning and Modeling

The data was partitioned into train and test datasets.

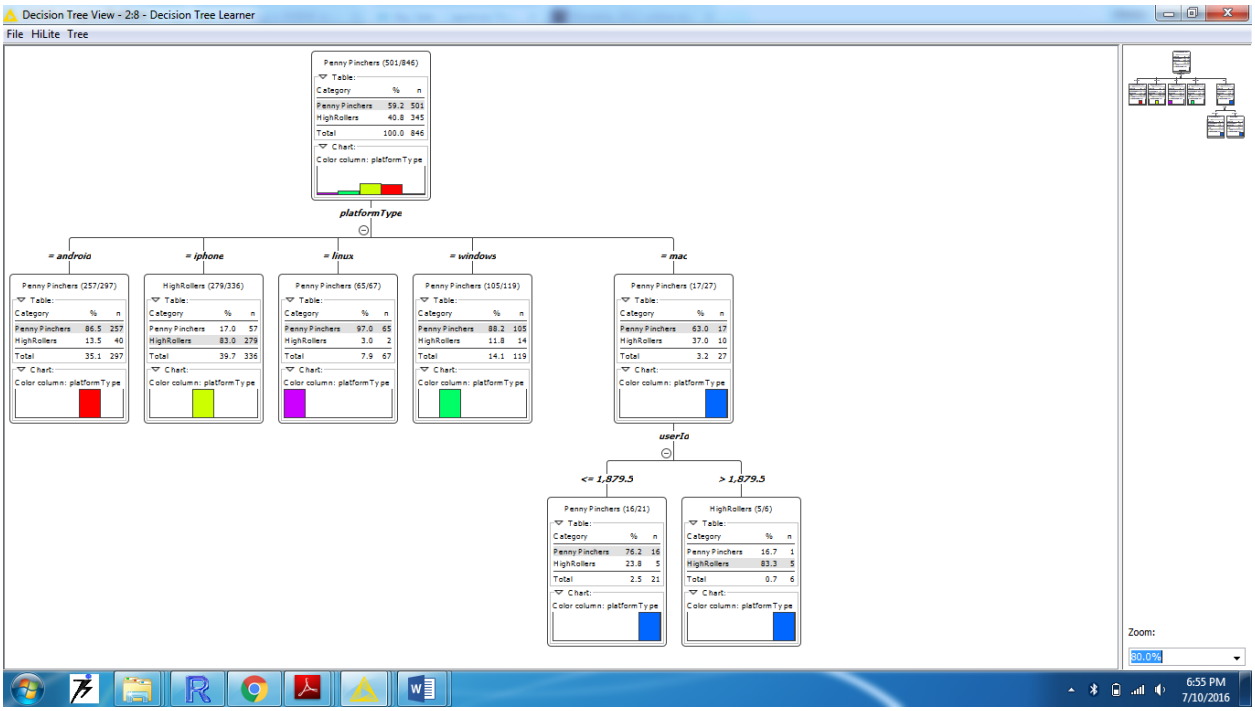
The **train** data set was used to create the decision tree model.

The trained model was then applied to the **test** dataset.

This is important because **the training set makes a model while the test dataset verifies the accuracy of the model created using training set.**

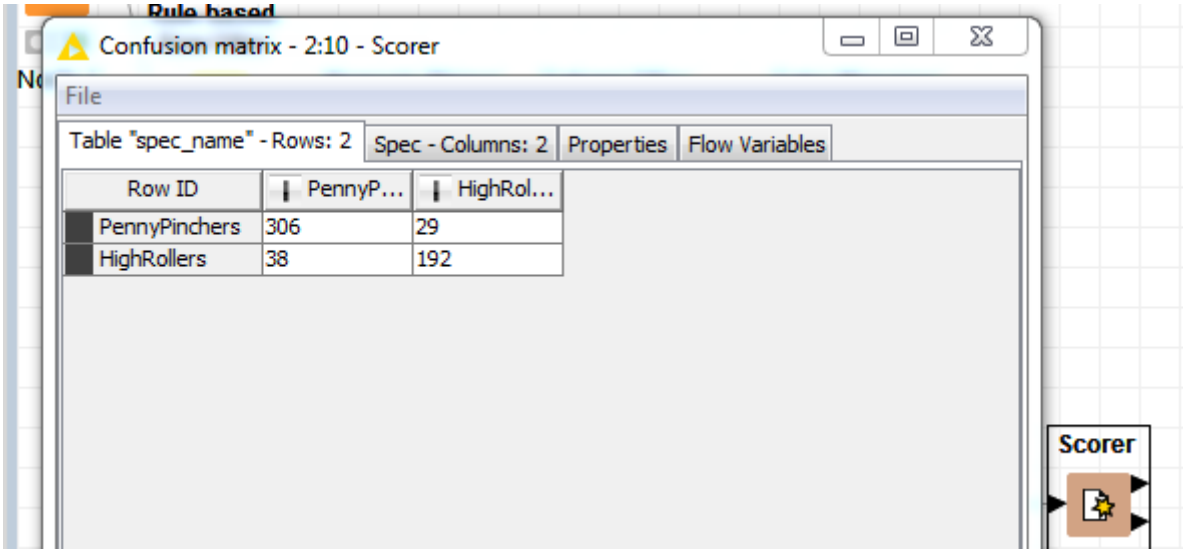
When partitioning the data using sampling, it is important to set the random seed because **a fixed random seed helps our model to become deterministic i.e. every time same result for a test set.**

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 **0.8814159 (which is mentioned in flow variables in the above screenshot or we can calculate based on correct/total i.e. (306+192)/565)**

306: These are the penny pinchers which were correctly predicted i.e. true positive.

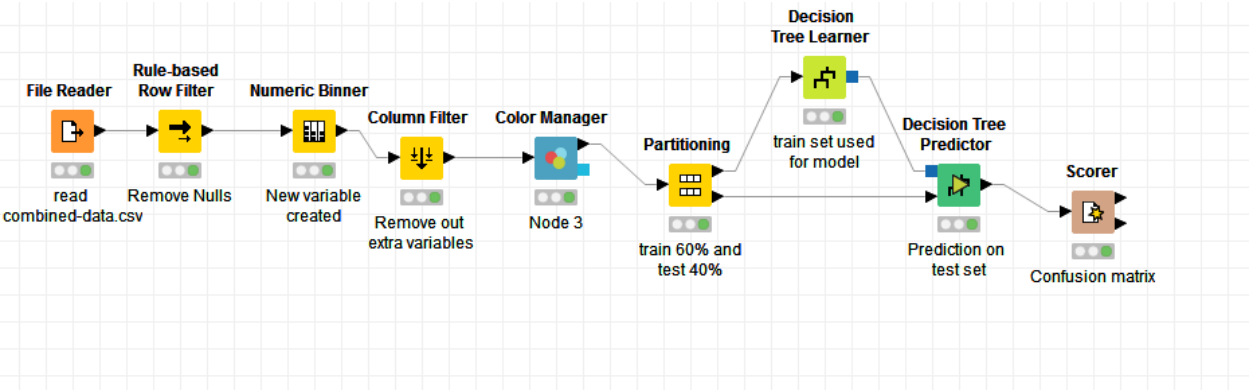
29: These are the high rollers which were predicted as penny pinchers i.e. false positive.

38: These are the penny pinchers which were predicted as high rollers i.ie false positive.

192: These are the high rollers which were predicted correctly i.e. true positive.

Analysis Conclusions

The final KNIME workflow is shown below:



What makes a HighRoller vs. a PennyPincher?

According to the decision tree, the platform used to play this game is the main independent variable which affects our depending variable. Thus a user using **iphone** has greater chances of being a high roller.

Specific Recommendations to Increase Revenue
1. Users who tend to spend more should be given more opportunities to spend i.e. more in app purchases for them in comparison to pennypinchers.
2. As more people advance towards a higher level its in app purchases can be made more expensive but only when a lot of people reach that level.

Clustering Analysis

Attribute Selection

Attribute	Rationale for Selection
hit_percentage	It helps us to find average hits based on userId
adCount	Number of ad clicks based on userId
price	Amount spent by user on in-app purchases
<Optional Fill in>	<Optional Fill in 1-3 sentences>

Training Data Set Creation

The training data set used for this analysis is shown below (first 5 lines): <Fill In Screenshot>

```
>>> trainingDF.head(n=5)
   totalAdClicks  revenue  hit-percentage
0             44     21.0       13.407821
1             10     53.0       10.000000
2             37     80.0       12.204724
3             19     11.0       10.943032
4             46    215.0       13.068182
>>>
```

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

# of clusters created: 3

Cluster Centers

Cluster #	Cluster Center
1	[34.65972222, 59.60416667, 11.93105959]
2	[25.36811594, 15.4173913 , 10.98206254]
3	[40.87037037, 138.24074074, 12.67545308]

These clusters can be differentiated from each other as follows:

Cluster 1 is different from the others in that… **total adClicks though being mediocre i.e. 34.65 generate revenue of 59.60 which is 4 times more than in cluster 2. Also the increase in hit-percentage means a higher revenue.**

Cluster 2 is different from the others in that… **total adClicks is the lowest in this cluster and generates lowest revenue and has users with least hit ratio.**

<Optional Fill In> Cluster 3 is different from the others in that… **total adClicks is the highest i.e. 40.87 and the revenue is at least 4 times more than any other cluster along with the highest hit-percentage.**

Recommended Actions

Action Recommended	**Rationale for the action**
Users in the 3rd cluster should be shown more ads and given more app purchases	This is because these users have the maximum adClicks and generate most revenue along with the highest hit-percentage. So these users can be offered more expensive products to buy just after they achieve a certain number of hits and be shown more ads.
Users in 2nd cluster should be given cheaper products and more ads	Users with lowest adClicks and least revenue should be shown cheaper products as they would be motivated to buy whenever they hit a flamingo.

Graph Analytics Analysis

Graph Analytics

1. Modeling Chat Data using a Graph Data Model

In this analysis, the graph model for chat is used. The graph model is a model in which the data is modeled as a graph. The graph model gives us some characteristic features. For example, we can see who is the chattiest person or how long conversation chains are or how active specific groups of users are along how active a team is.

2. Creation of the Graph Database for Chats

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

i) The schema of the 6 CSV files

There are 6 files named

(1) “chat_create_team_chat.csv”,

(2) “chat_join_team_chat.csv”,

(3) “chat_leave_team_chat.csv”,

(4) “chat_item_team_chat.csv”,

(5) “chat_mention_team_chat.csv”,

(6) “chat_respond_team_chat.csv”.

(1) chat_create_team_chat.csv

chat_create_team_chat.csv file creates an edge labeled "CreatesSession" from User to TeamChatSession. The columns are the User id, Team id and the timestamp of the CreatesSessio edge.

(2) chat_join_team_chat.csv

chat_join_team_chat.csv file creates an edge labeled "Joins" from User to TeamChatSession. The columns are the User id, TeamChatSession id and the timestamp of the Joins edge.

(3) chat_leave_team_chat.csv

chat _leave_team _chat.csv file creates an edge labeled "Leaves" from User to TeamChatSession. The columns are the User id, TeamChatSession id and the timestamp of the Leaves edge.

(4) chat_item_team_chat.csv

chat_item_team_chat.csv file creates nodes labeled ChatItems. Column 0 is 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.

(5) chat_mention_team_chat.csv

chat_mention_team_chat.csv file 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.

(6) chat_respond_team_chat.csv

chat_respond_team_chat.csv file creates an edge labeled "ResponseTo" from a ChatItem node to another ChatItem node. Column 0 has the ID of the first ChatItem node, column 1 has the ID of the second ChatItem node and column 2 has the timeStamp of the edge.

ii) The loading process and a sample LOAD command

Process

(1) Load CSV file as row

(2) Create nodes

(3) Create edges

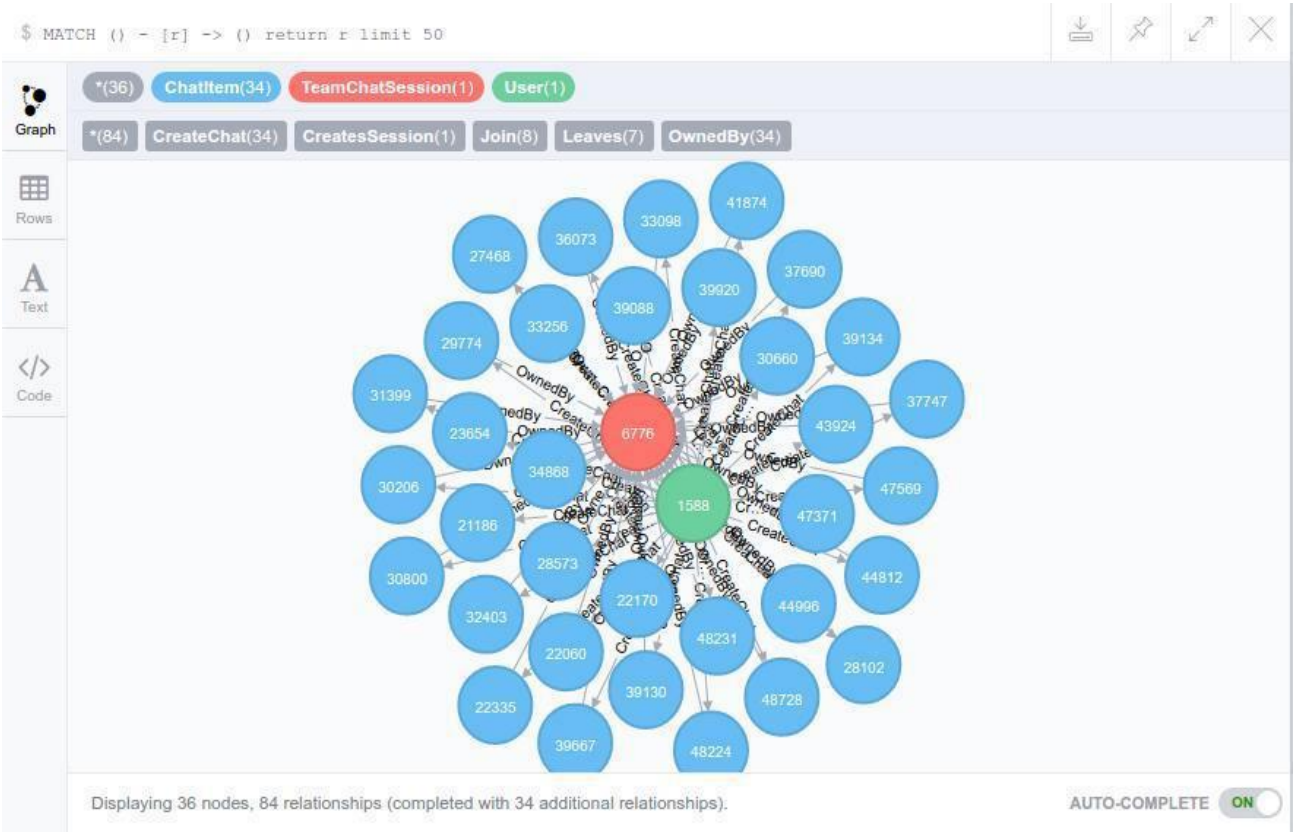
A sample LOAD command

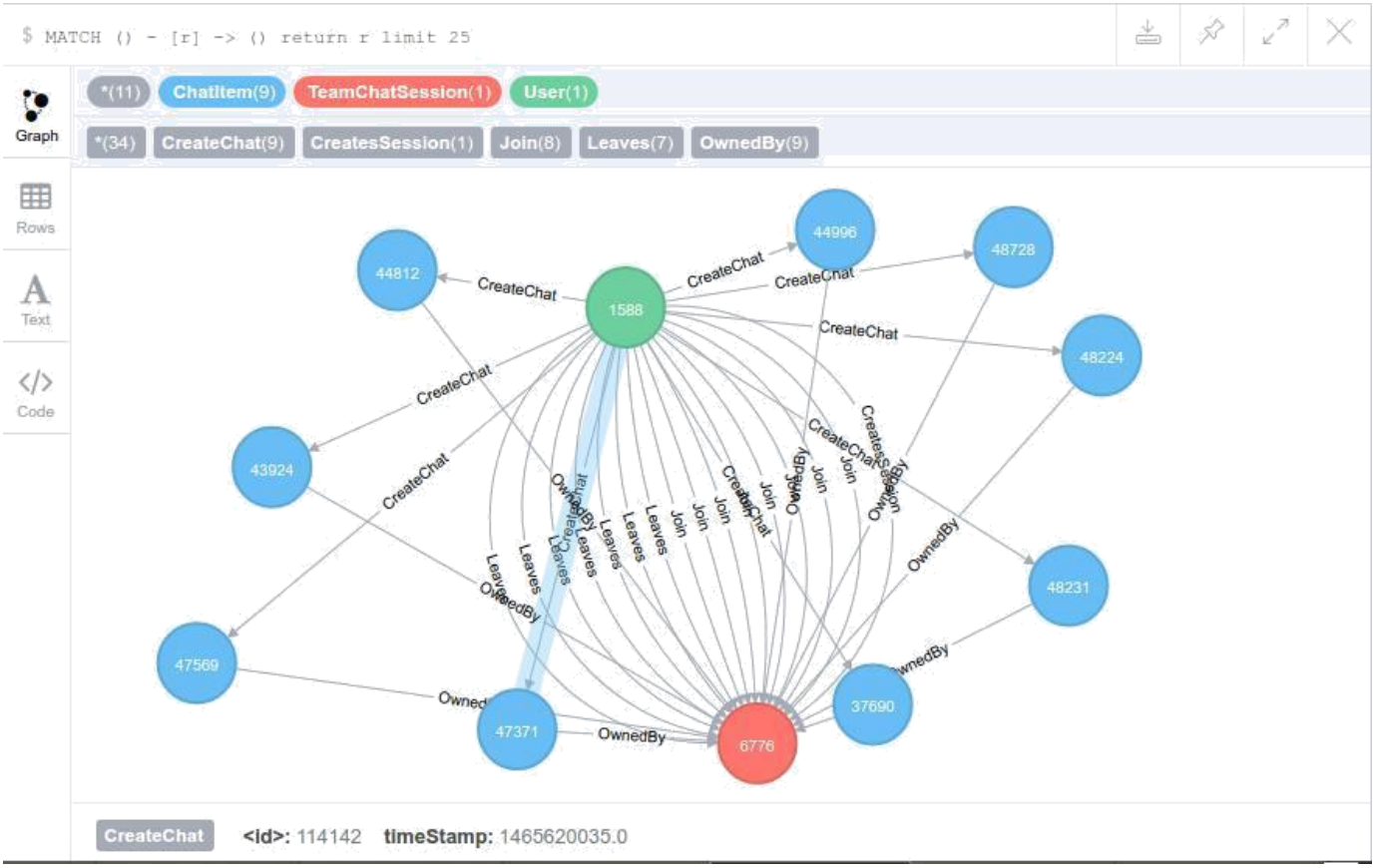
LOAD CSV FROM "file:/ C:\Users\Pranav\Desktop\New folder\Big data project\big_data_capstone_datasets_and_scripts\big_data_capstone_datasets_and_scripts\chatdata/chat_join_team_chat.csv" AS row

MERGE (u:User {id: toInt(row[0])})

MERGE (c:TeamChatSession {id: toInt(row[1])}) MERGE (u)-[:Join{timeStamp: row[2]}]->(c)

iii) Screenshots of some part of the graph .





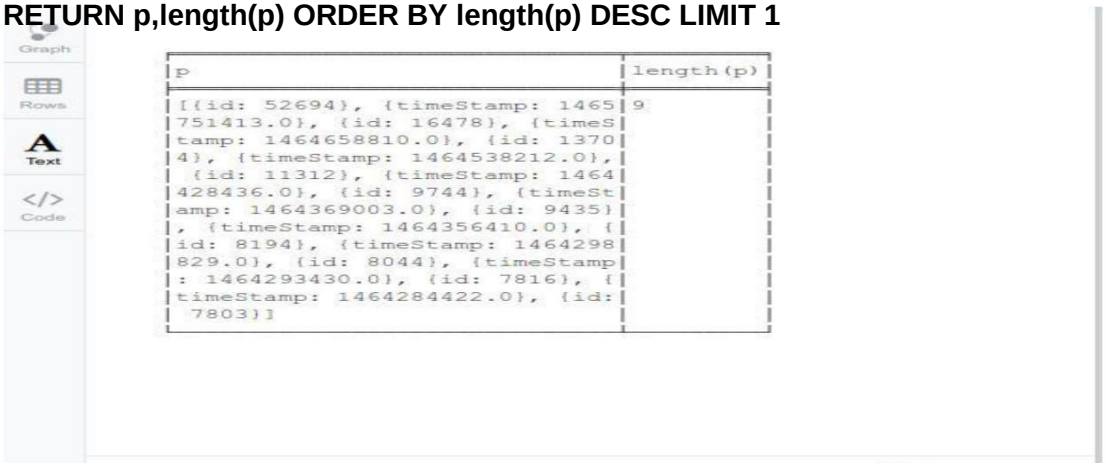
3. Finding the longest conversation chain and its participants

Here, I 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.

3.1 Query for Part1

MATCH p= (a) - [:ResponseTo*] -> (c)

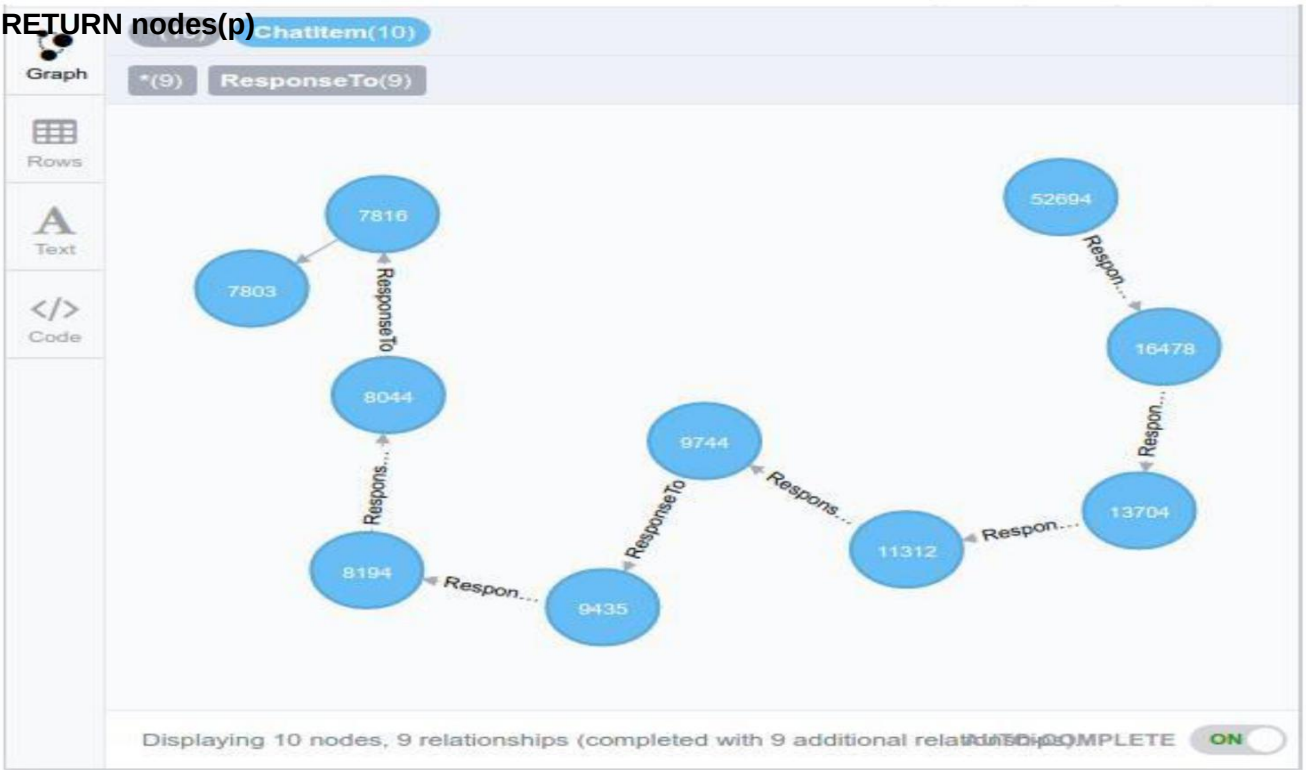
RETURN p,length(p) ORDER BY length(p) DESC LIMIT 1



3.2 Query for Part2

MATCH p=(a)-[:ResponseTo *]->(c) WHERE length(p) = 9

RETURN nodes(p)



Query to count the number of users in the longest conversation chain

MATCH p=(a)-[:ResponseTo *] ->(c) WHERE length(p) = 9

MATCH (u) - [:CreateChat*] -> (ci) WHERE ci IN

nodes(p) RETURN count(distinct u)

The query returns 5 as the number of unique users in the longest chat.

3.3 Answer

Q. What is the number of items in the longest chat?

A. The number of chats in the longest chat is 9. Chat id is this: {52694, 16478, 13704, 11312, 9744, 9435, 8194, 8044, 7816, 7803}

Q. What is the number of unique users in the longest chat?

R. The number of unique users in the longest chat is 5.

Q. How the results of this kind of search may be relevant to Elegance Inc?

Elegance Inc might want to know how the users are active. The longer the longest chat is, the more the users are active, that is to say, Elegance Inc can use the number I got above as an index to estimate how the users are active.

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

Steps for analysis

1. Find the chattiest users using MATCH command
2. Obtain the user id and number of chats using RETURN, ORDER BY, DESC, and LIMIT

Queries for analysis

(I) As for chattiest users

```
MATCH p= (u) - [:CreateChat*] -> (c)

RETURN u.id, count(*) AS appearances

ORDER BY appearances DESC LIMIT 10
```

(II) As for chattiest teams

```
MATCH (ci) - [:PartOf*] -> (c) - [:OwnedBy] ->

(t) RETURN t.id,count(distinct ci) AS
```

appearances ORDER BY appearances DESC

LIMIT 10

4.1 Chattiest Users

Users	Number of Chats
394	115
2067	111
1087	109

4.2 Chattiest Teams

Teams	Number of Chats
82	1324
185	1036
112	957

4.3 Discussion

I did not report above, but a user whose id is 999 is one of the chattiest user in 10 chattiest users list. The user is included in team id 52 which is one of the chattiest team (This is also not reported above, but team id 52 is 7th chattiest team).

This kind of search may be useful to Elegance Inc because we can see the chattiest users as target for direct marketing. They seem to enjoy this game with players. Also, we can see the chattiest group as target for advertising. They want to be able to chat with other players more effectively. If so, we could recommend them fee-charging service for their demands.

5. How Active Are Groups of Users?

Steps

1. Construct the neighborhood of users
2. Make connection between users and call it “InteractsWith” as a edge
3. Delete self-loops
4. Calculate cluster coefficient for chattiest users

Most Active Users (based on Cluster Coefficients)

User ID	Coefficient
209	0.952
554	0.904
1087	0.8