Technical Appendix

Catch the Pink Flamingo Analysis

Produced by: Urvi Kalia

Acquiring, Exploring and Preparing the Data

Data Exploration

The objective of this document is to showcase , key findings from data exploration phase. Here the data into consideration is the data generated from the highly popular mobile game called "**Catch The Pink Flamingo**". A short brief description about the game and its data structure , is followed. After which key finding and its analysis is documented.

Catch The Pink Flamingo

The objective of the game is to catch as many Pink Flamingos as possible by following the missions provided by realtime 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. Its a multi-user game where each user can be a member of a single team at any given time. Users are ranked based on their accuracy and speed and are categorized as "rising star", "veteran", "coach", "social butterfly" and "hot flamingo".Teams can be of size from one to thirty.Users are allowed in game purchases .Finally the game never ends, that is there will always be a more complicated next level.

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	An entry is added in this file when a 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
buyclicks.csv	An entry is added to this file when a player makes an inapp purchase.	**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	It contains a line for each user playing 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	**teamid**: the id of the team

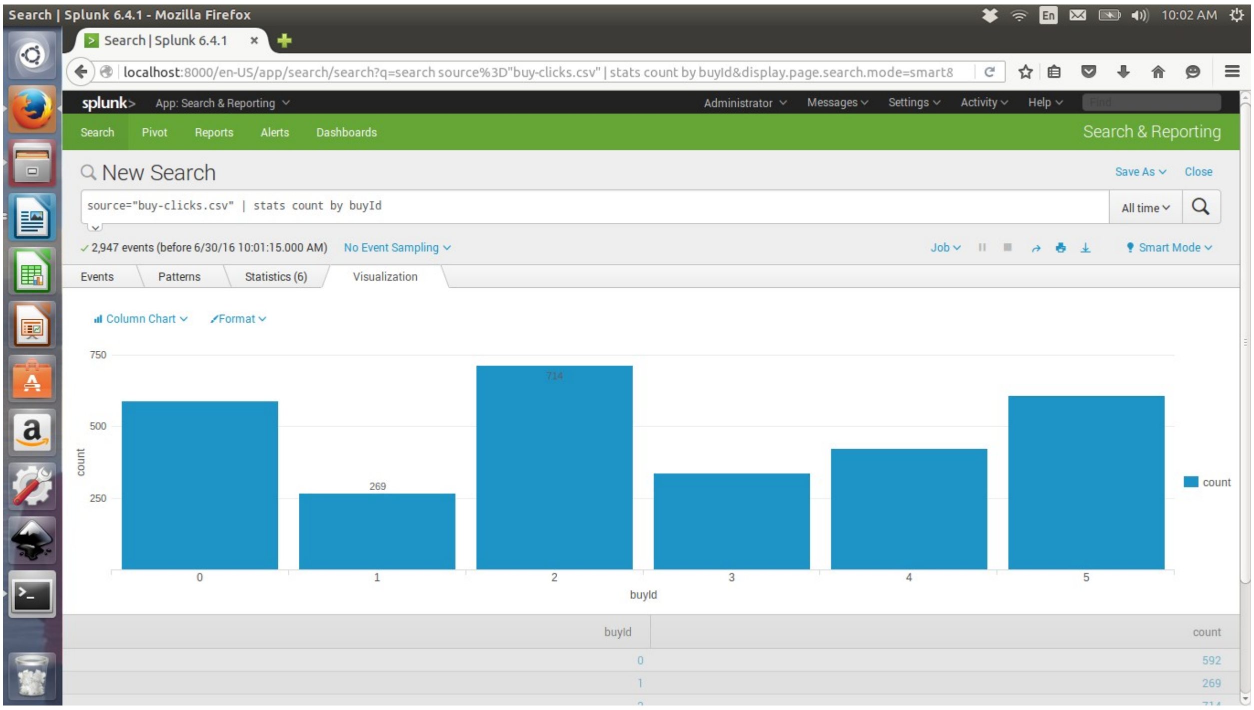
	each team terminated in the game	name: the name of the team **teamCreationTime**: the timestamp when the team was created **teamEndTime**: the timestamp when the last member left the team **strength**: a measure of team strength, roughly corresponding to the success of a team **currentLevel**: the current level of the team
team-assignments.csv	An entry 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.	**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	An entry is added to this file each time a team starts or finishes a level in the game	**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 in this file describes a user session, which denotes when a user starts and stops playing the game.	**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	An entry 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

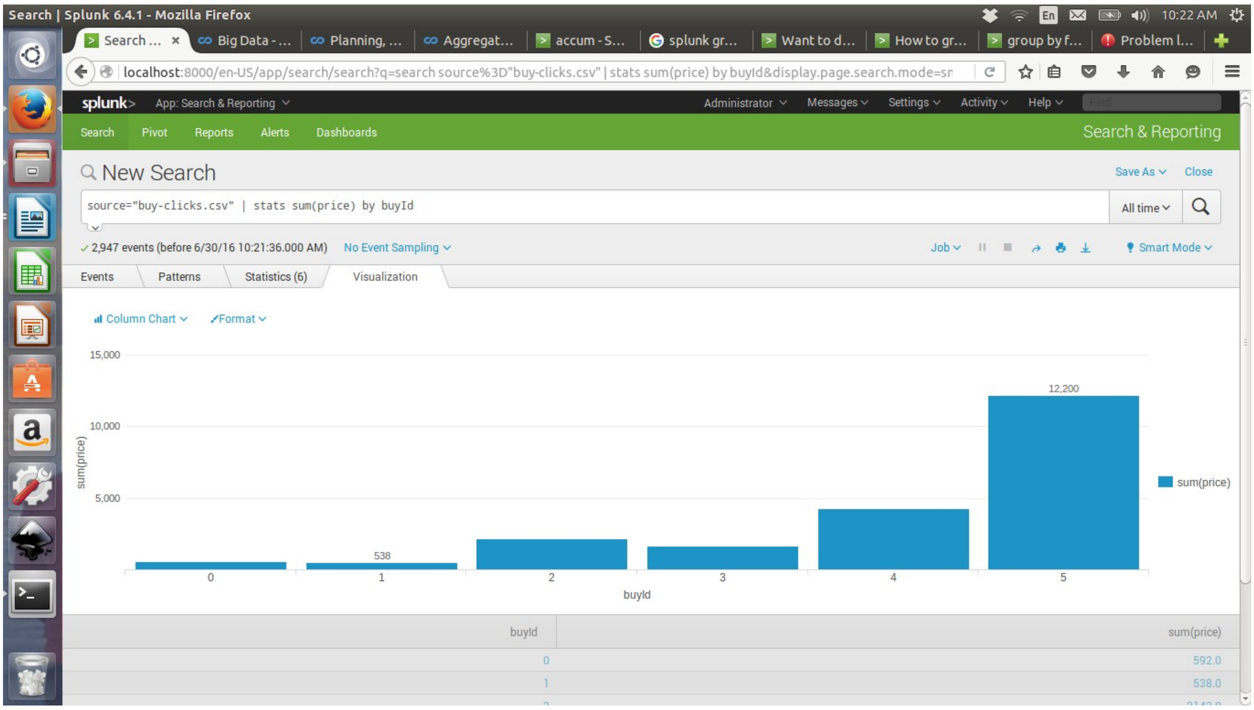
A histogram showing showing frequency of each item being purchased.



Analysis

1. Item2 is purchased most often , 714 times.Followed by Item5 and Item0.

Histogram given below, depicts money made from each item:

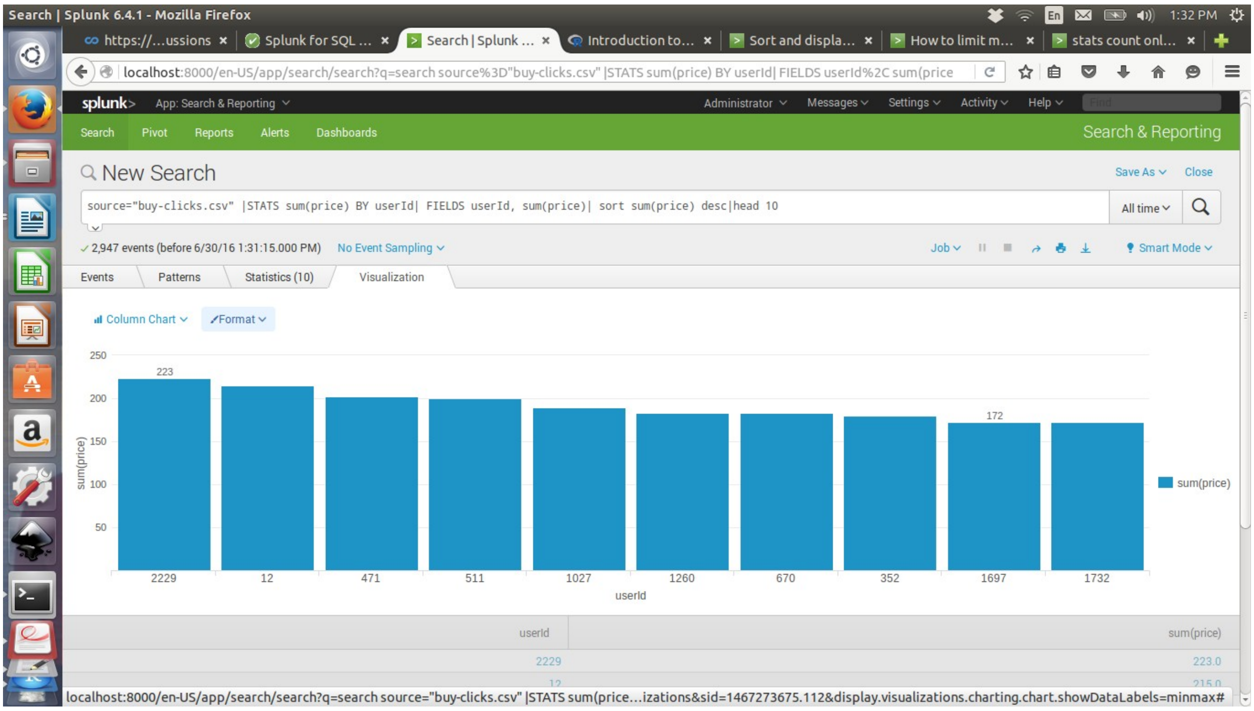


Analysis

1. Item 5 makes the maximum money with the total of about 12,200.
2. Item 0 and Item 1 contribute the least , less than 3% of the total purchase done.
3. Though Item 2 is puchased most often, its price is just 3 as compared to item 5 which is 20.Hence while considering contribution in terms of total price amount, Item 5 stands out.

Filtering

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



Analysis :

1. The top buyer spends around 223.
2. Top 10th buyer spends around 172.
3. Doing further analysis , one can infer the variation among the top ten buyers is less as compared to variation across the data.

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

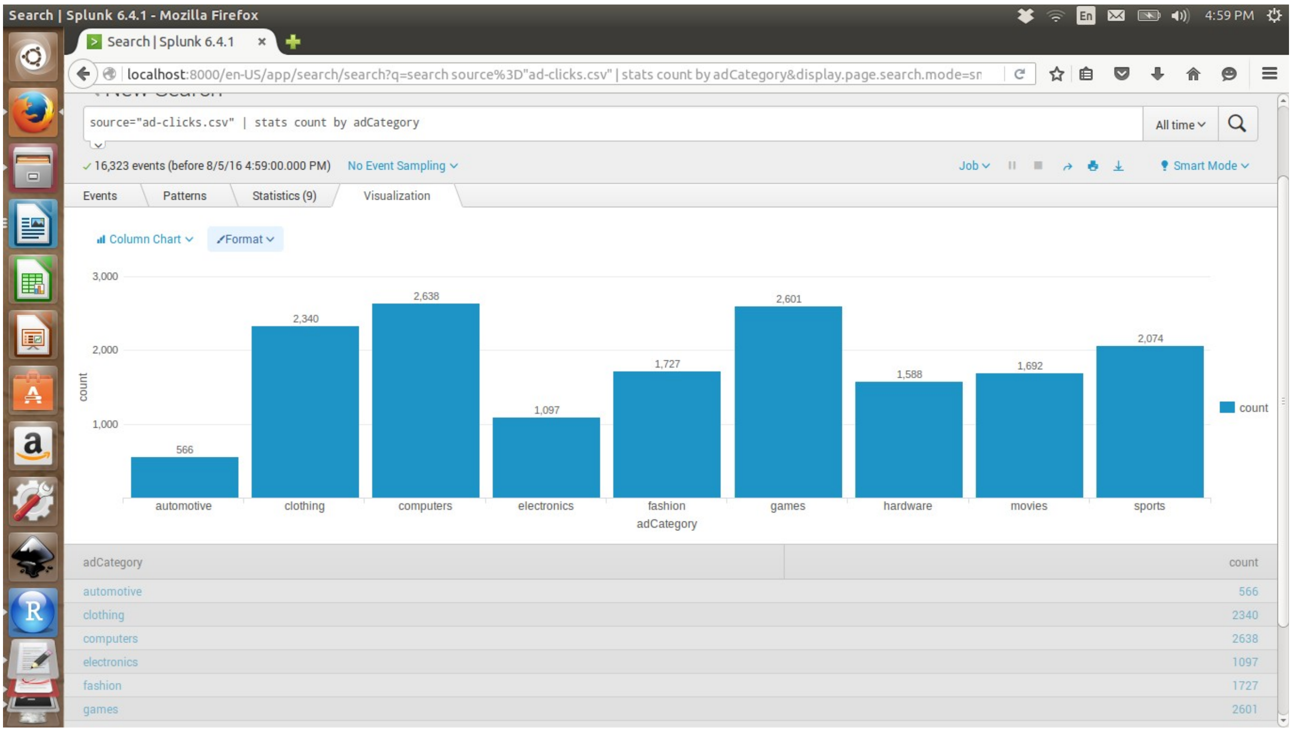
Rank	**User Id**	**Platform**	**Hit-Ratio (%)**
1	2229	iphone	11.596
2	12	iphone	13.068
3	471	iphone	14.503

Analysis

1. Hit-Ratio varies from 0 to 50%. with a mean of 11.09% and median at 10.95%.Three top users in Hit-Ratio have hit ratio more than 25% where as rest have below 25%.Hence can be interpreted as there is a large difference in hit ratio between top three users (in hit-ratio) and others.
2. 75% of times the platform used is iphone or android.Linux and mac platform is not much used for the game. Its recommened , that we further analyze the issue as to why there is such a huge variation based on the platform.
3. From the above table , one can conclude that top buyers in the game are not the ones with best hit ratio.No relationship between hit-Ratio and buying patterns.

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	4928.25
2 - uneven	0.55	0.60	0.55	5184.1

Data Preparation

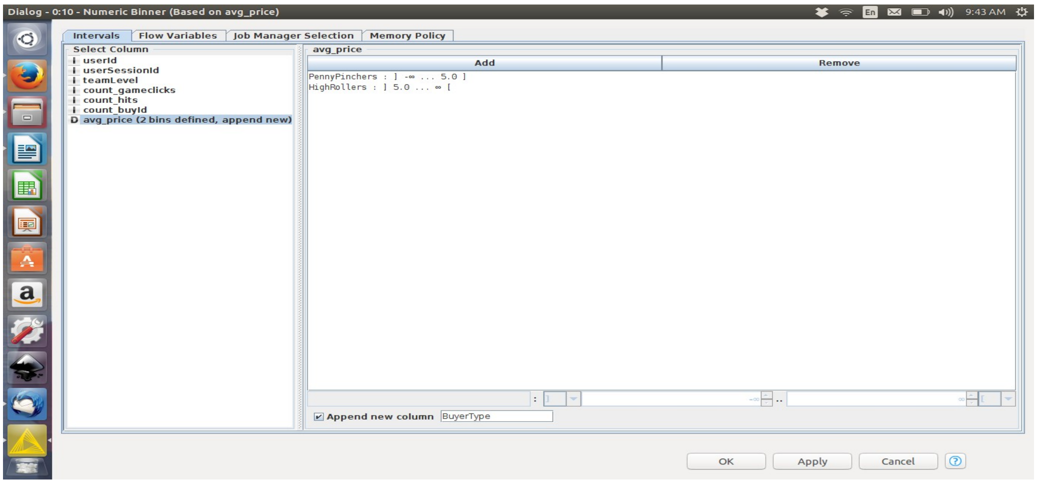
Analysis of combined_data.csv

<u>Sample Selection</u>

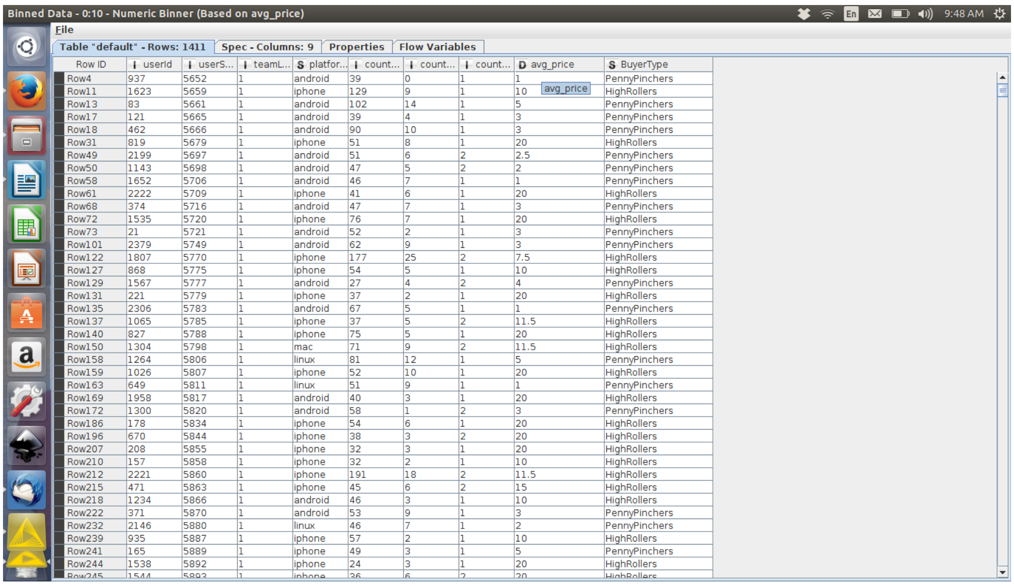
Item	Amount
# of Samples	4619
# of Samples with Purchases	1411

<u>Attribute Creation</u>

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



Based on avg_price , another categorical vaiable named BuyerType is created . BuyerType is PennyPinchers if avg_price is less than or equal to 5. If avg_price is greater than 5 , its value is HighRollers.



The creation of this new categorical attribute was necessary due to the following :

1. Here we had to predict the user that are likely to buy big-ticket items.
2. In the given data, average purchase price is given.This is a numerical variable.
3. Decision tree , requires the response variable to be categorical.
4. Hence created a categorical variable named BuyerType based on avg_price, which will be used as the response variable while building a model using decision tree.

Attribute Selection

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

Attribute	**Rationale for Filtering**
avg_price	As the response variable “buyerType” is derived from avg_price.Its variable has to be excluded from the selection.
user_id	userId is just an identifier for the user.It won't contribute anything the the model.
userSession_Id	userSession_id is also just an identifier.Hence filtering from the dataset.

With the given dataset , following variables are being considered for modeling :

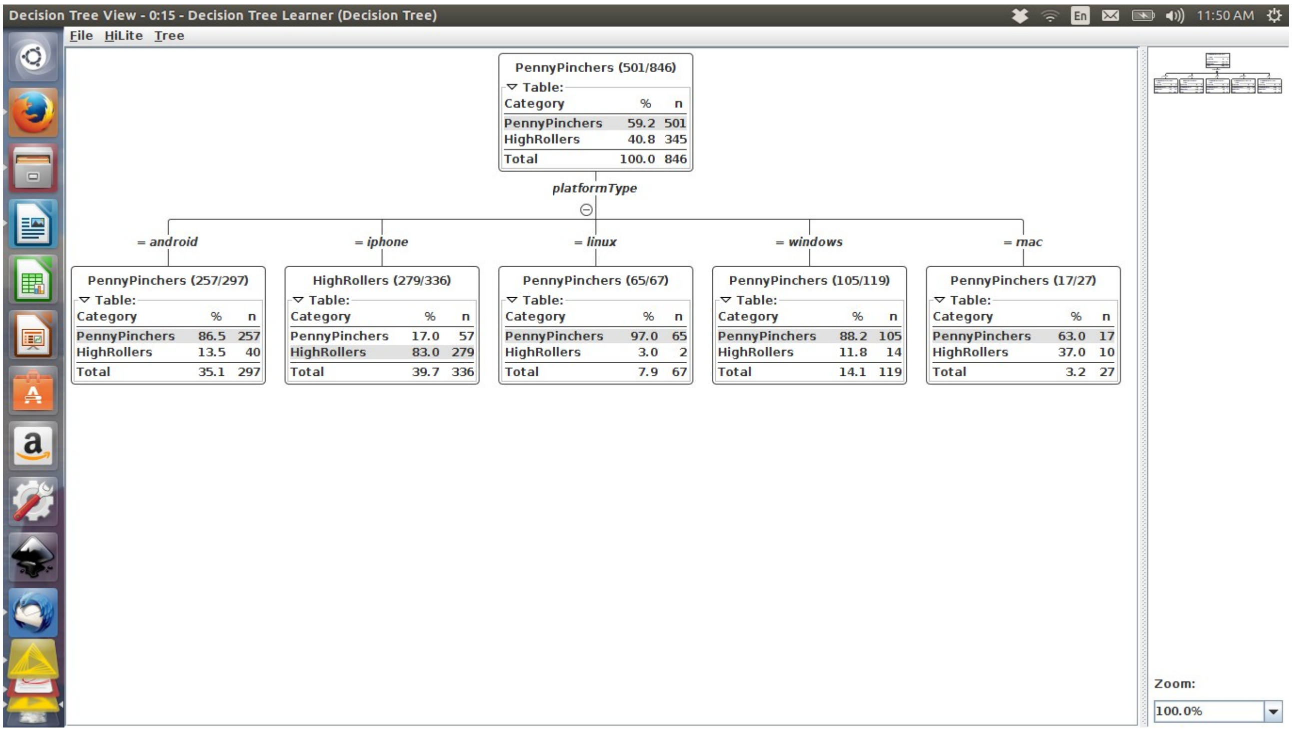
4. Response Variable : BuyerType
5. Exploratory Variables : teamLevel, platformType,count_gameclicks,count_hits,count_buyid.

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 **testing** dataset.
This is important because **the model has to be tested on the data that was not used to train the model**.

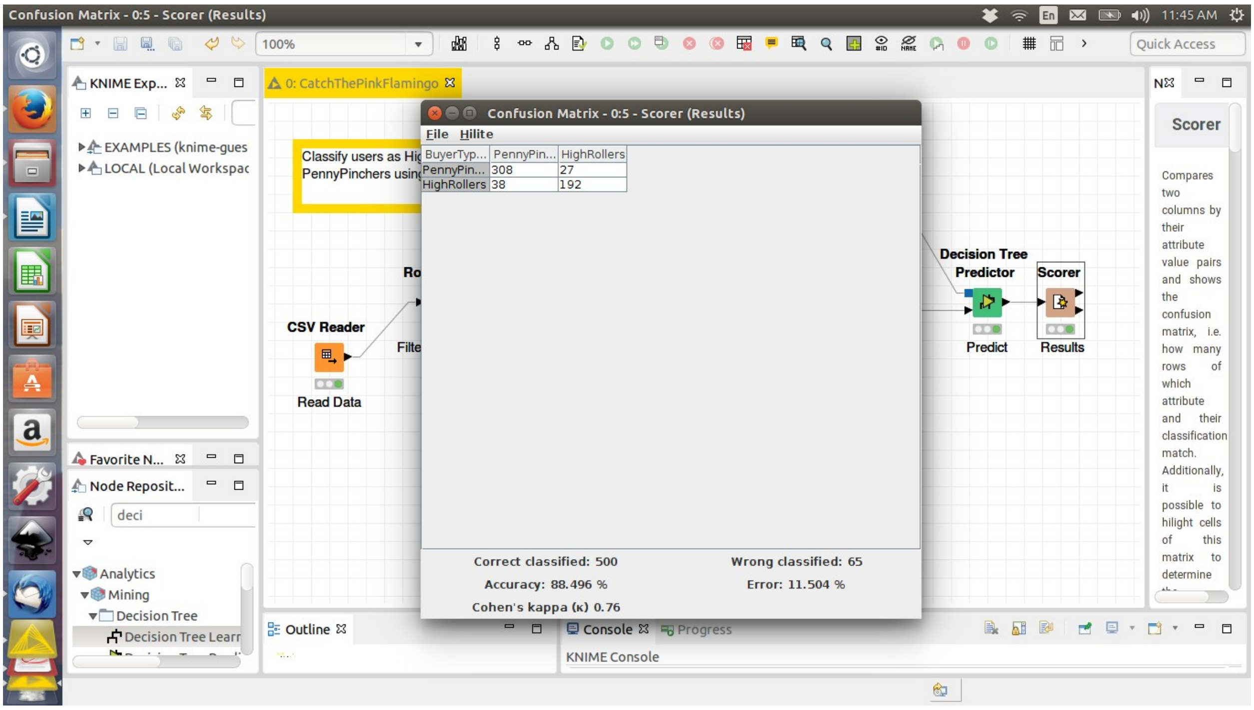
When partitioning the data using sampling, it is important to set the random seed for the following reason :
To ensures that same partition are formed every time the partition node is executed.Hence get reproducible results.

Screenshot of the resulting decision tree is as shown below:



Evaluation

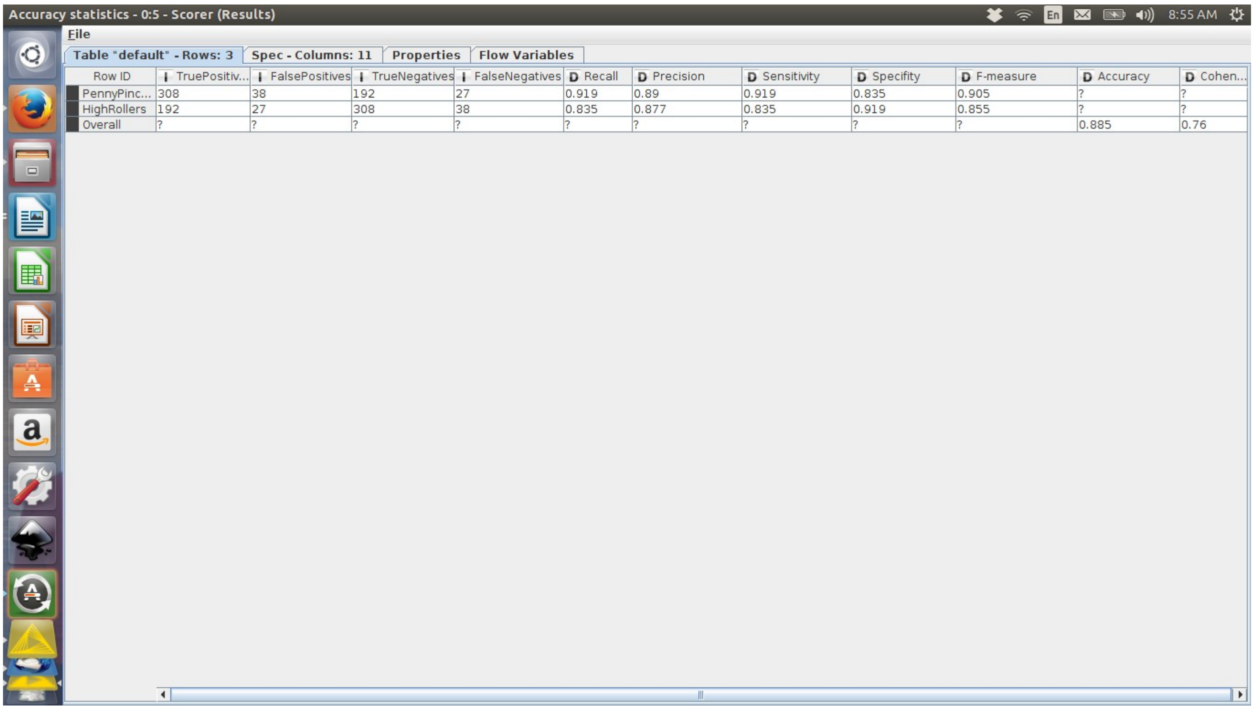
Screenshot of the confusion matrix can be seen below:



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

500 are correctly classified , where as 65 are wrongly classified.89% of times users were correctly classified as PennyPinchers. Where as 87% times they were correclty identified as HighRollers.

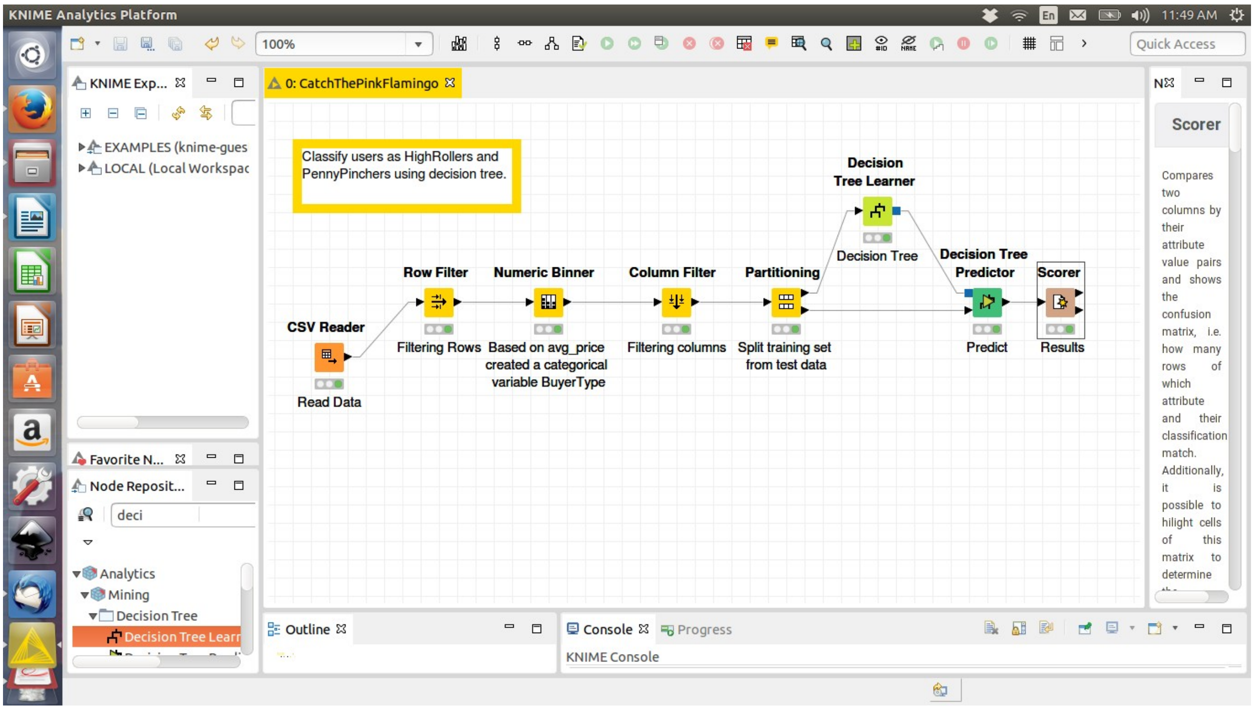
Other accuracy statistics are as below :



Row ID	TruePositiv...	FalsePositives	TrueNegatives	FalseNegatives	Recall	Precision	Sensitivity	Specifity	F-measure	Accuracy	Cohen...
PennyPinc...	308	38	192	27	0.919	0.89	0.919	0.835	0.905	?	?
HighRollers	192	27	308	38	0.835	0.877	0.835	0.919	0.855	?	?
Overall	?	?	?	?	?	?	?	?	?	0.885	0.76

Analysis Conclusions

The final KNIME workflow is shown below:



What makes a HighRoller vs. a PennyPincher?

Iphone users are HighRoller where as all other platform (i.e. android, linux, windows and mac) users are PennyPincher. Iphone forms 39.7% of user population, Android users are around 35.1 % , windows contribute 14.1% of total users,users using linux or mac are 7.9 and 3.2% respectively.

Specific Recommendations to Increase Revenue
1. Promote in-app purchases on android platform as it makes 35.1% of users.This will lead to increase in revenue proportionally. It is also recommended to investigate the reason for android users being PennyPincher
2. Promote in-app purchases to users who are classified as PennyPincher and are on iphone platform.As our findings suggest that iphone users are majorly HighRoller , there is a higher probability of them puchasing.Investigate further as to if there is any other reason for PennyPincher iphone users for not making in-app purchases.

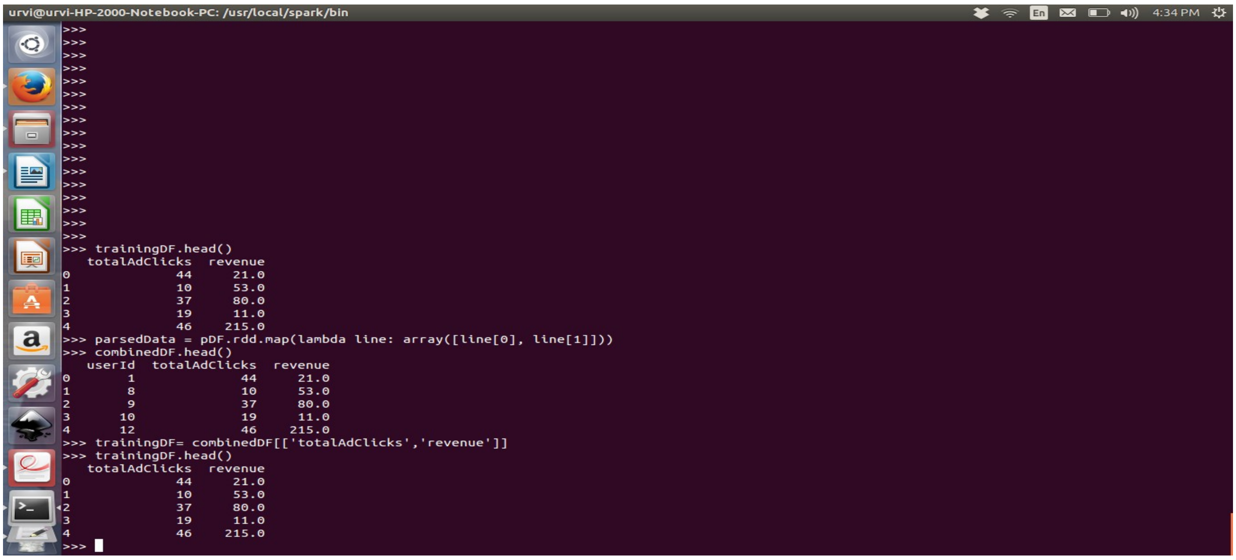
Clustering Analysis

Attribute Selection

Attribute	Rationale for Selection
Number of Ads clicked	As Eglence Inc. Gets paid for ads clicked. It stands out as a good attribute to be consider while clustering.
Amount spend on purchasing inapp items	This attribute seams to be contributing to the revenue and its a numeric value.

Training Data Set Creation

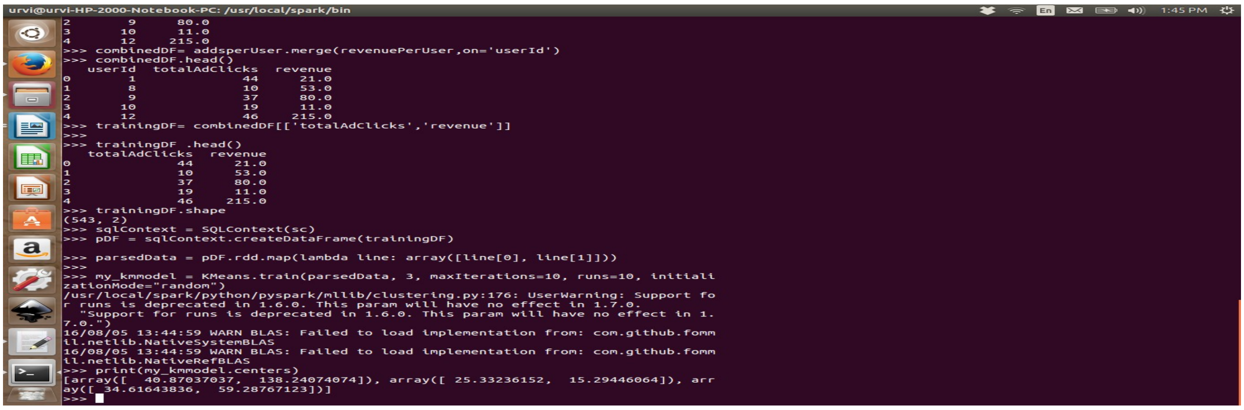
The training data set used for this analysis is shown below (first 5 lines):



Dimensions of the training data set (rows x columns) :
(543, 2)

# of clusters created: 3 clusters created

my_kmmodel = KMeans.train(parsedData, 3, maxIterations=10, runs=10, initializationMode="random")



Cluster Centers

Cluster #	Cluster Center
1	[40.87037037,138.24074074]
2	[34.61643836, 59.28767123]
3	[25.33236152, 15.29446064]

First number refers to number of ad-clicks and second number is the amount spent per user.
These clusters can be differentiated from each other as follows:

1. Cluster1 spends almost 9 times more than cluster3 on in-app purchases.
2. Cluster2 spends 3 times more than cluster3 on in-app purchases.
3. Players in cluster1 clicks on adds 1.5 times more than the players on cluster3.
4. The cluster size for the given clusters is 53,139 and 351 respectively.
5. Cluster1 where in players clicks the most and spends the maximum,is comparitively of much smaller size , as compared to others.
6. Cluster2 and Cluster3 are large in size. This can be considered as an opportunity and targeted.

Recommended Actions

Action Recommended	Rationale for the action
Charge higher fees for hosting frequently clicked ads.i.e. ads of category clothing , computers and games are clicked the most.	The distribution of number of ads clicked per user is almost normally distributed with mean equal to 29.37 and median is 30 and standard deviation of 15.All the three cluster centers fall within one standard deviation. Increasing the fee for hosting ads which are often clicked can improve the overall revenue generation from the game.
Campaign to promote in-app products /items to players of cluster 2 and 3.	In-app purchases are quite low in cluster 2 and 3 as compared to cluster1. Inspecting the size of the two clusters (cluster 2 and 3) , its a potential market for In-app purchases.

Graph Analytics Analysis

Modeling Chat Data using a Graph Data Model

The chat data in the game is modeled as graph data. This can give us some valuable insights about the interactions done amongst team/user during the game.

Creation of the Graph Database for Chats

Approach

There are 6 CSV files which describe the graph database.Process followed is as below :

7. Identify the Nodes and the edges.
8. Idetify the properties for each of the node and edge.
9. Load the given csv files accordingly.Before that add a unique constraint for id's for each.

Schema of the csv files

Schema of the csv files is show below :

chat_create_team_chat.csv
UserId TeamId TeamChatSessionId timeStamp

chat_join_team_chat.csv
UserId TeamChatSessionId timeStamp

chat_leave_team_chat.csv
UserId TeamChatSessionId timeStamp

chat_item_team_chat.csv
UserId TeamChatSessionId ChatItemId timeStamp

chat_mention_team_chat.csv
ChatItemId UserId timeStamp

chat_respond_team_chat.csv
ChatItemId1 ChatItemId2 timeStamp

The graph model for chats contains four nodes and eight edges as mentioned below.

Nodes
User
Team
ChatItem
TeamChatSession

Edges
CreateChat
PartOf
CreateSession
OwnedBy
Joins
Leaves
Mentioned
ResponseTo

All the nodes have **id** as the property whereas all the edges have **timestamp** as the property.

Loading commands

LOAD CSV FROM "file:////home/urvi/Documents/Coursera/Bigdata/capstoneProject/week4/chat-data/chat_create_team_chat.csv" AS rowMERGE (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)

LOAD CSV FROM "file:////home/urvi/Documents/Coursera/Bigdata/capstoneProject/week4/chat-data/chat_join_team_chat.csv" AS row MERGE (u:User {id: toInt(row[0])}) MERGE (c:TeamChatSession {id: toInt(row[1])}) MERGE (u)-[:Joins{timeStamp: row[2]}]->(c)

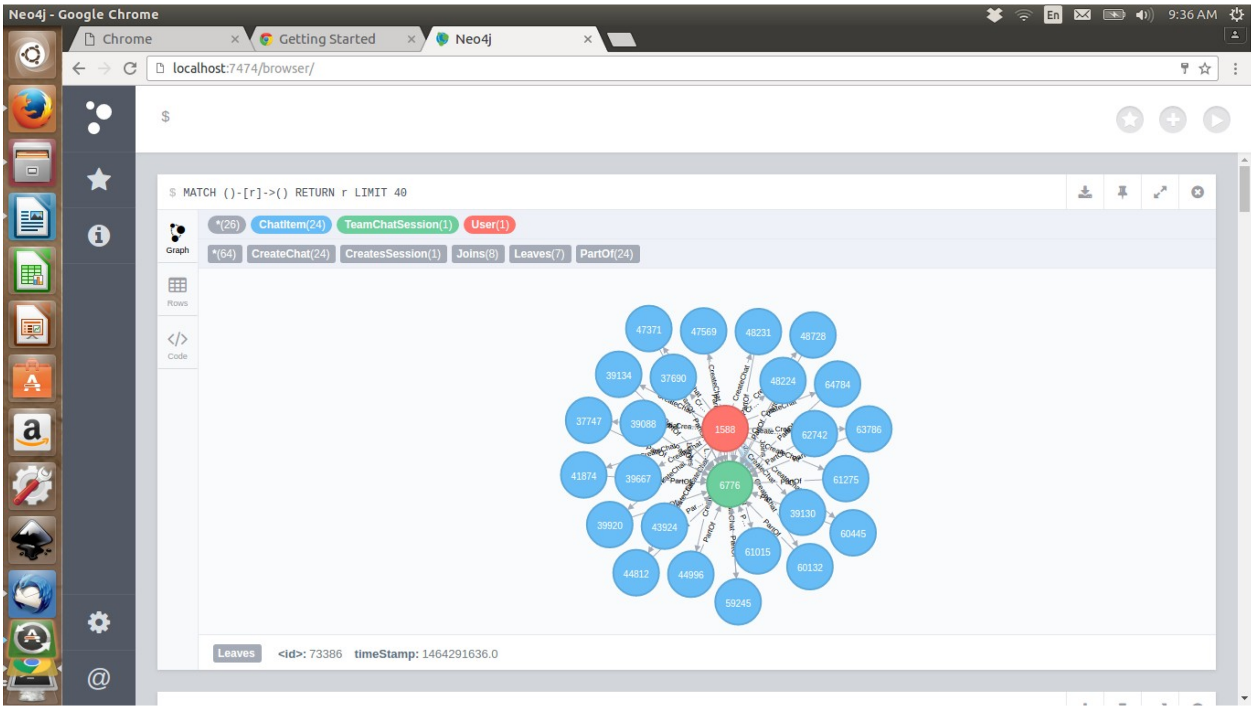
LOAD CSV FROM "file:////home/urvi/Documents/Coursera/Bigdata/capstoneProject/week4/chat-data/chat_leave_team_chat.csv" AS row MERGE (u:User {id: toInt(row[0])}) MERGE (c:TeamChatSession {id: toInt(row[1])}) MERGE (u)-[:Leaves{timeStamp: row[2]}]->(c)

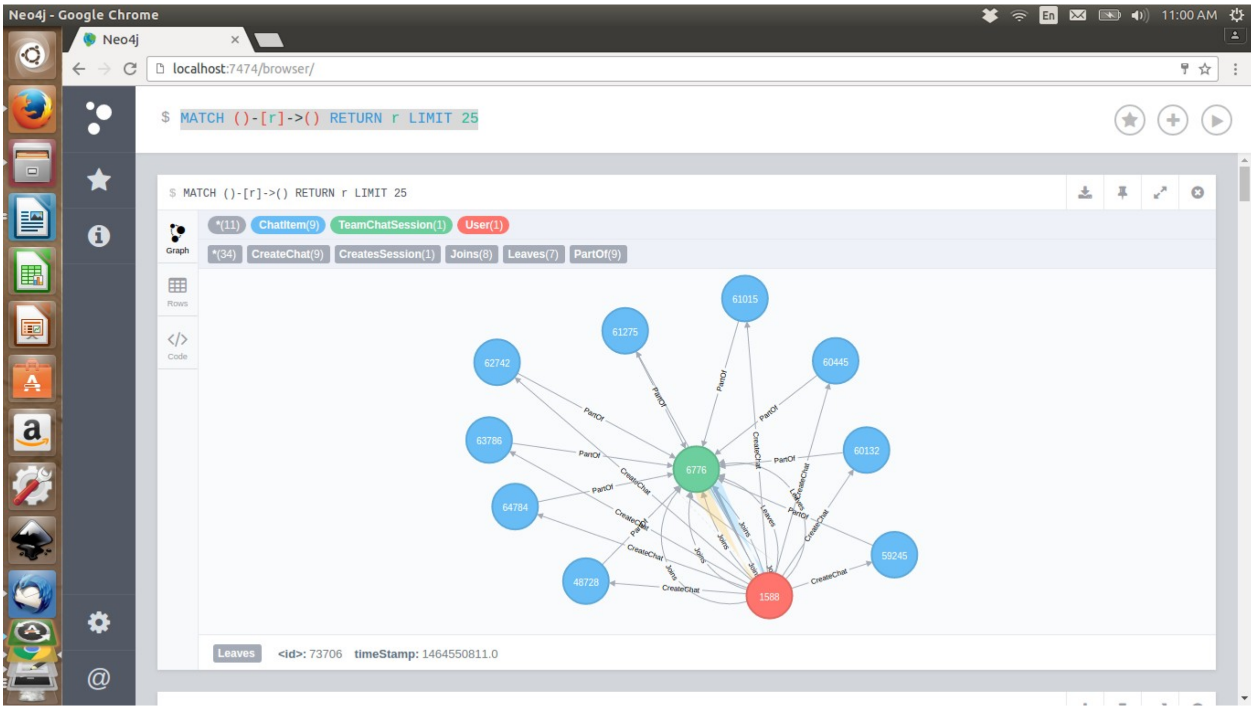
LOAD CSV FROM "file:////home/urvi/Documents/Coursera/Bigdata/capstoneProject/week4/chat-data/chat_item_team_chat.csv" AS row MERGE (u:User {id: toInt(row[0])}) MERGE (c:TeamChatSession {id: toInt(row[1])}) MERGE (i:ChatItem {id: toInt(row[2])}) MERGE (u)-[:CreateChat{timeStamp: row[3]}]->(i) MERGE (i)-[:PartOf{timeStamp: row[3]}]->(c)

LOAD CSV FROM "file:////home/urvi/Documents/Coursera/Bigdata/capstoneProject/week4/chat-data/chat_mention_team_chat.csv" AS row MERGE (i:ChatItem {id: toInt(row[0])}) MERGE (u:User {id: toInt(row[1])}) MERGE (i)-[:Mentioned{timeStamp: row[2]}]->(u)

LOAD CSV FROM "file:////home/urvi/Documents/Coursera/Bigdata/capstoneProject/week4/chat-data/chat_respond_team_chat.csv" AS row MERGE (i1:ChatItem {id: toInt(row[0])}) MERGE (i2:ChatItem {id: toInt(row[1])}) MERGE (i1)-[:ResponseTo{timeStamp: row[2]}]->(i2)

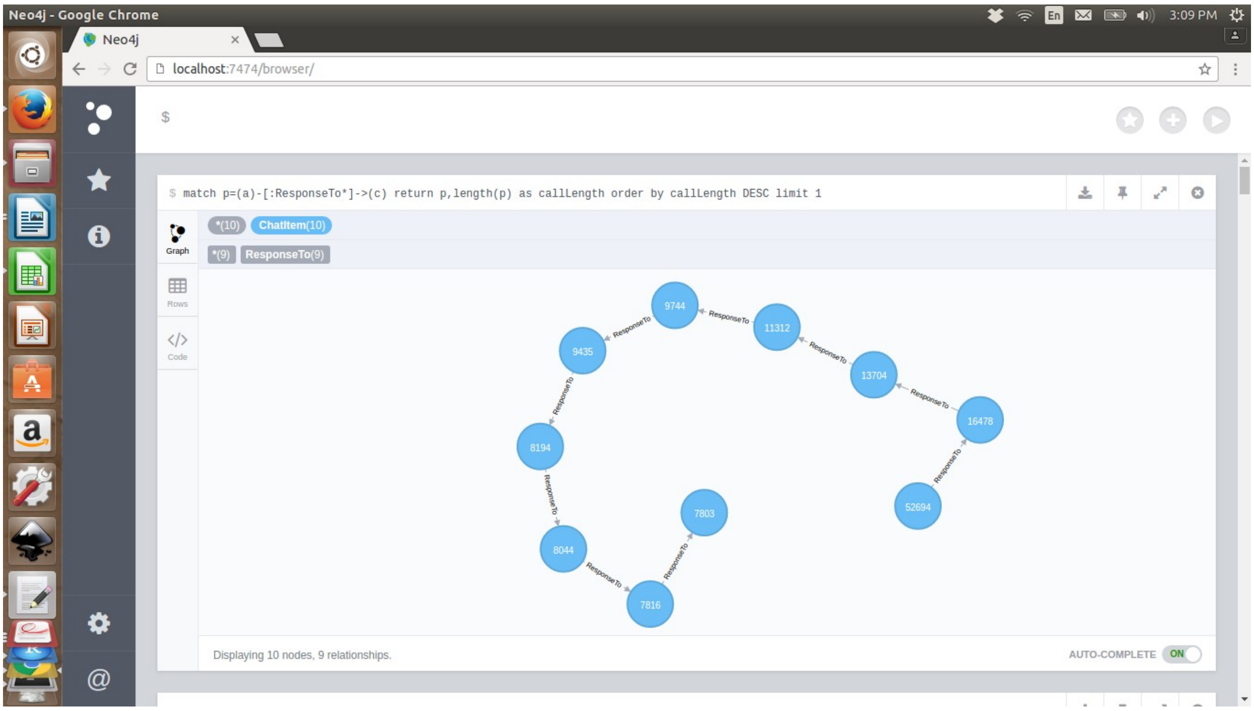
Sample snapshots of the graph





Longest Conversation chain and its participants

The longest conversation path length is 10.



Query :

match p=(a)-[:ResponseTo]->(c) return p,length(p) as callLength order by callLength DESC limit 1*

Distinct users involved in the longest conversation are **5**.

Their userId's are 1192 1978 1153 853 1514

Further this information can be used in the following ways :

1. For the users and team involved in longest conversation , we can identify user ranking and the team ranking and level at which the team is. Can further investigate if its a specific level where team needs to interact.Is the team using chat option to decide the stratergies for the difficult levels , general interaction or learning about the game in general.
2. What is the time duration the coversation spans.
3. Looking at the details further , comparing length of conversations one can infer , more than 85% of chat conversations are just one or two threads long.

Conversation Length	No. of Conversations
9	1
8	2
7	5
6	13
5	47
4	163
3	656
2	2733
1	11073

We can investigate the chat option's ease of usability , since there are only handful of long conversations.Also can something be introduced in the game , which will require teams to interact and discuss stratergy amonst themselves while playing the game. This will keep the users engaged in the game.

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

To find the top Chattiest user , we need to find the outdegree for each user with createChat edge (number of chats created by each user). Arrange them in decending order of number of edges , further limit them to 10.

Query *: match (n)-[r:CreateChat]->()*

return n.id as user, count(r) as outDegree

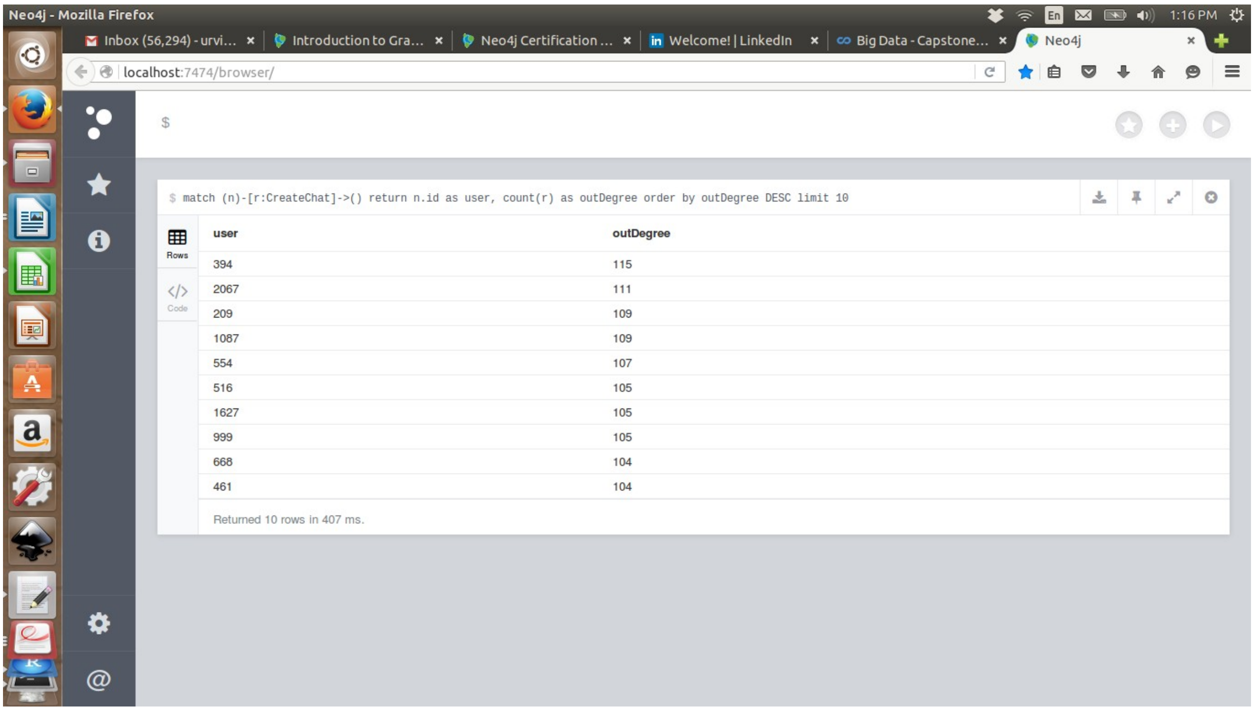
order by outDegree DESC limit 10

Chattiest Users

Users	Number of Chats
394	115
2067	111
209	109

Chattiest Teams

Teams	Number of Chats
82	1324
185	1036
112	957



Relation between Chattiest users and Chattiest teams

4. None of the top 3 users have been in the top 3 teams but user with id 999 was in the top ranked team.

How Active Are Groups of Users?

First we need to create a new edge among users who satisfy either of the following condition :

2. One mentioned another user in the chat.
3. One created a chatItem in response to another user's chatItem.

To do this we execute the following queries :

Match (u1:User)-[:CreateChat]-()-[:Mentioned]->(u2:User)

create (u1)-[:InteractsWith]->(u2)

Match (u1:User)-[:CreateChat]-()-[:ResponseTo]-()-[:CreateChat]-(u2:User)

create (u1)-[:InteractsWith]->(u2)

Match (u1)-[r:InteractsWith]->(u1) delete r

Most Active Users (based on Cluster Coefficients)

User ID	Coefficient
394	1
2067	0.8571
209	0.9523