Technical Appendix

Catch the Pink Flamingo Analysis

Alessandro Corradini, Dec 2017, Coursera Big Data Specialization Capstone Final Assignment

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	Database of clicks on ads	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	Database of purchases.	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
game-clicks.csv	A record of each click a user performed during 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
level-events.csv	A record of each level event for a team. Level events are recorded when a team ends or begins a new level	timestamp: when the event occurred. eventId: a unique id for the event

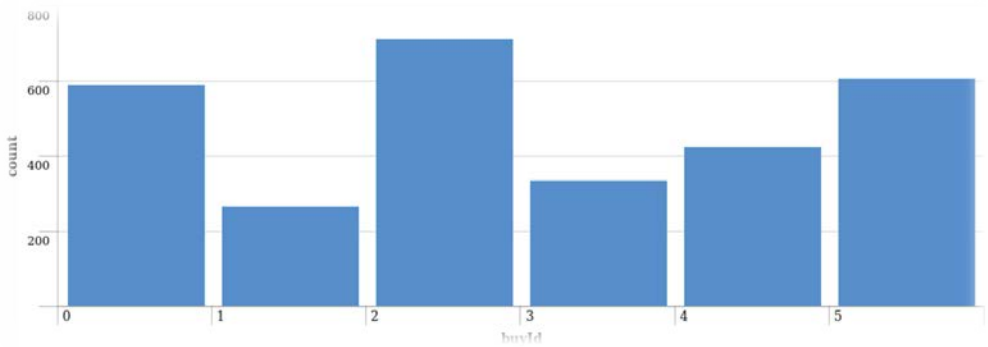
		teamId: the id of the team teamLevel: the level started or completed eventType: the type of event, either start or end
team-assignments.csv	A record of each time a user joins a team.	timestamp: when the user joined the team. team: the id of the team userId: the id of the user assignmentId: a unique id for this assignment
team.csv	A record of each 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
user-session.csv	A record of each session a user plays.	timestamp: a timestamp denoting

	When a team levels up, each current user session ends and a new session begins with the new level.	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.
users.csv	Database of the game users	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.

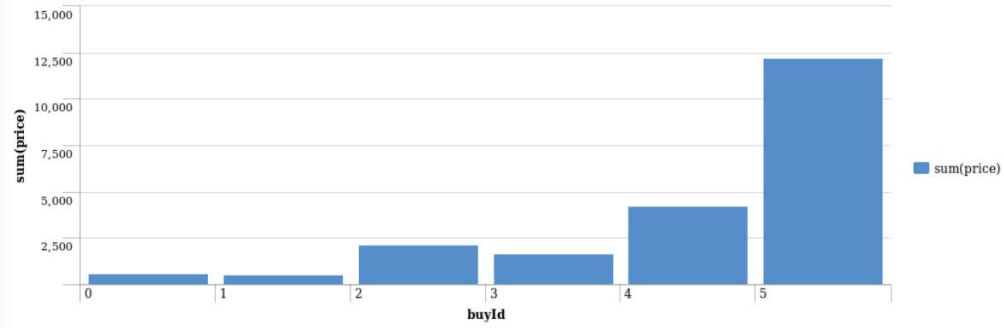
Aggregation

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

A histogram showing how many times each item is purchased:

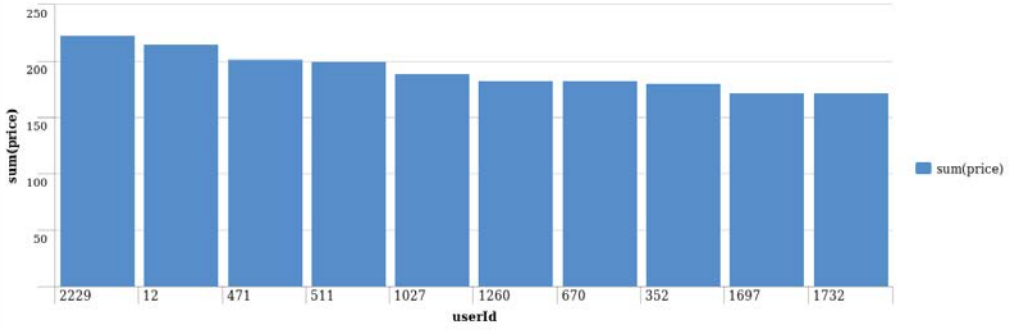


A histogram showing how much money was made from each item:



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	11.5%
2	12	iPhone	13%
3	471	iPhone	14.5%

Data Classification Analysis

Data Preparation

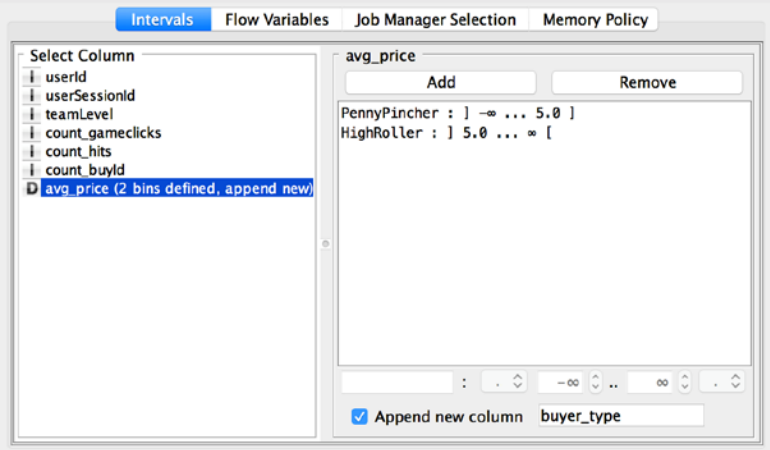
Analysis of combined_data.csv

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:



Describe the design of your attribute in 1-3 sentences:
High rollers are defined as those who purchase items over $5.00. Defining a new column based on the avg_price allows us to classify users accordingly.

The creation of this new categorical attribute was necessary because **our goal is to understand the attributes of who makes large purchases. This categorical variable is what we are going to base our decision tree upon.**

Attribute Selection

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

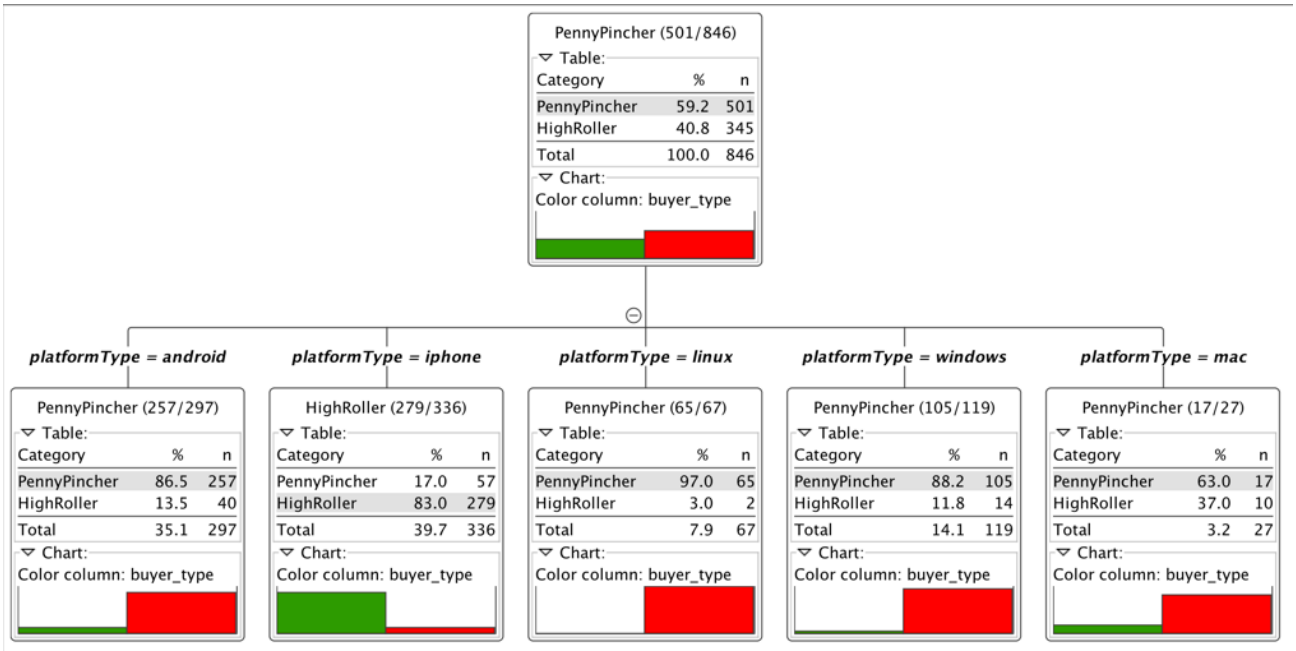
Attribute	**Rationale for Filtering**
userId	Not relevant for the model.
userSessionId	Not relevant for model.
avg_price	This feature was used to create the categorical feature "buyer_type", the variable we are trying to predict based on other elements. We do not want to include this in our model.

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 **partitioning the data set into training and test data allows us to verify the accuracy of the trained model.**

When partitioning the data using sampling, it is important to set the random seed because **it allows you to obtain reproducible results each time you run the partition.**

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



Cluster Analysis

Attribute Selection

Attribute	**Rationale for Selection**
totalAdClicks	Total of ad-clicks per user. This attribute is correlated to the profit's company.
totalBuyClicks	Total money of in-app purchase per user. This attributes is correlated to the profit's company.
totalRevenue	Total money spent on in-app purchase items per user.

Training Data Set Creation

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

Create the training data set for clustering and show the first 5 rows

```
trainingDF = combinedDF[['totalAdClicks','totalBuyClicks','totalRevenue']]
trainingDF.head(n=5)
```

	totalAdClicks	totalBuyClicks	totalRevenue
0	44	9	21.0
1	10	5	53.0
2	37	6	80.0
3	19	10	11.0
4	46	13	215.0

Show the dimension of the training data set

```
trainingDF.shape
```

```
(543, 3)
```

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

# of clusters created: **3**

Cluster Centers

Cluster #	Cluster Center
1	[41.07, 10.29, 145.51]
2	[34.28, 6.45, 67.22]
3	[26.30, 4.48, 17.07]

These clusters can be differentiated from each other as follows:

Cluster 1 is different from the others in that **the players in the cluster have the highest 'totalAdClics', 'totalBuyClicks' and 'totalRevenue'.**

Cluster 2 is different from the others in that **the players in the cluster have the second highest 'totalAdClics', 'totalBuyClicks' and 'totalRevenue'.**

Cluster 3 is different from the others in that **the players in the cluster have the lowest 'totalAdClics', 'totalBuyClicks' and 'totalRevenue'.**

Recommended Actions

Action Recommended	Rationale for the action
Increase the prices for advertisements showed to players into first cluster	Players into the first cluster are frequent ad-clickers and increase the price of their ad, could increase the company's revenue.
Charge players into third cluster lower fees for the price of the in-app purchase items	Players into the third cluster only purchase items with lower prices. Lowering the price of the in-app purchase or giving them coupons could encourage them to spend more.

Graph Analytics

Modeling Chat Data using a Graph Data Model

The graph model is a network based on chat interactions between users. A chat session can be initiated by a user, other users on the same team are able to join and leave the session. Interactions between users begins when a user create a post. It's possible for a user, mention another user. All relationship between entities are logged with a timestamp.

Creation of the Graph Database for Chats

Describe the steps you took for creating the graph database.

Write the schema of the 6 CSV files

chat_create_team_chat.csv	userID teamID teamChatSessionID timestamp
chat_join_team_chat.csv	userID teamChatSessionID timestamp

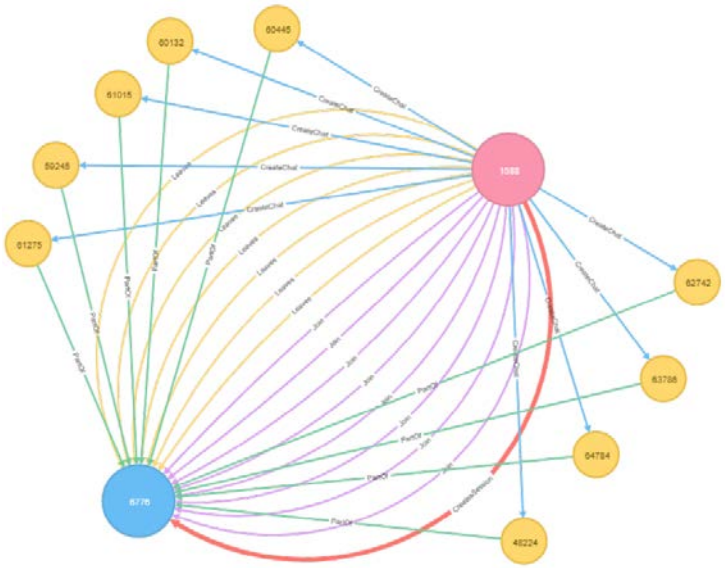
chat_leave_team_chat.csv	userID teamChatSessionID timestamp
chat_item_team_chat.csv	userID teamChatSessionID chatItemID timestamp
chat_mention_team_chat.csv	chatItemID userID timestamp
chat_respons_team_chat.csv	chatItemID_1 chatItemID_2 timestamp

Explain the loading process and include a sample LOAD command

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

The first line load the csv from the specific location one row at a time. From the second line to fourth, create the nodes for User, Team, TeamChatSession with a specific column converted to integer, this field is used by the id attribute. The fifth and sixth lines create CreatesSession and OwnedBy edges and link the nodes previously created. The edges have a timestamp property filled by the fourth column of schema.

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.

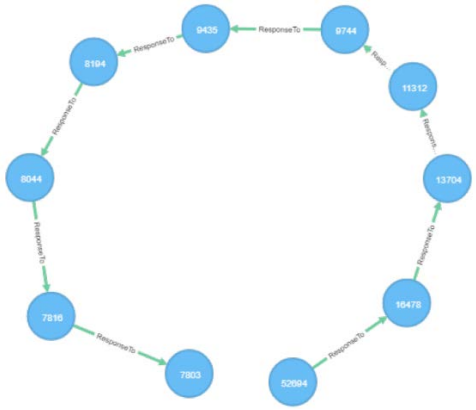


Finding the longest conversation chain and its participants

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.

How many cats are involved in it?

```
MATCH p=(a)-[:ResponseTo*]->(b)
RETURN p, length(p)
ORDER BY length(p) desc limit 1
```



The longest conversation chain in the chat data has path length 9, therefore 10 chats are involved in it.

How many users participated in this chain?

```
match p=(c:ChatItem)-[:ResponseTo*]->(j:ChatItem)
where length(p)=9
with p
match q=(u:User)-[:CreateChat]-(c:ChatItem)
where (c IN NODES(p))
return count(distinct u)
```

With 9 as longest path, count the number of distinct users who create ChatItem in this longest path. The query returns 5.

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

Describe your steps from Question 2. In the process, create the following two tables. You only need to include the top 3 for each table. Identify and report whether any of the chattiest users were part of any of the chattiest teams.

Chattiest Users

Determine the number of chats created by a user from the CreateChat edge

```
match (u:User)-[:CreateChat]-(i:ChatItem)
return u.id as Users, count(u.id) as Num_Chats
order by count(u.id) desc limit 10
```

Users	Num_Chats
394	115
2067	111
209	109
1087	109
554	107
516	105
1627	105
999	105
668	104
461	104

Users	Number of Chats
394	115
2067	111
209	109

Chattiest Teams

Match all ChatItem with a PartOd edge and connect them with a TeamChatSession node that have an OwnedBy edge connection them with any other node.

```
match (:ChatItem)-[:PartOf]->(:TeamChatSession)-[:OwnedBy]->(t:Team)
return t.id as Teams, count(t.id) as Num_Chats
order by count(t.id) desc limit 10
```

Teams	Num_Chats
82	1324
185	1036
112	957
18	844
194	836
129	814
52	788
136	783
146	746
81	736

Teams	Number of Chats
82	1324
185	1036
112	957

Finally, present your answer, i.e. whether or not any of the chattiest users are part of any of the chattiest teams.

```
match (u:User)-[:CreateChat]->(:ChatItem)-[:PartOf]->(:TeamChatSession)-[:OwnedBy]->(t:Team)
where u.id IN [394, 2067, 209, 1087, 554, 516, 1627, 999, 668, 461]
and t.id IN [82, 185, 112, 18, 194, 129, 52, 136, 146, 81]
return distinct u.id as User, t.id as Team
```

This query is used to investigate if the most chattiest user are part of any chattiest team and it return one result, userID 999 is part of teamID 52.

How Active Are Groups of Users?

Describe your steps for performing this analysis. Be as clear, concise, and as brief as possible. Finally, report the top 3 most active users in the table below.

Connect mentioned users

```
match (u1:User)-[:CreateChat]->(:ChatItem)-[:Mentioned]->(u2:User)
merge (u1)-[:InteractsWith]->(u2)
```

Connect users responses with the chat creator

```
match (u1:User)-[:CreateChat]->(:ChatItem)-[:ResponseTo]-(:ChatItem)<-[:CreateChat]-(u2:User)
merge (u1)-[:InteractsWith]->(u2)
```

Eliminate all self interaction

```
match (u1)-[r:InteractsWith]->(u1) delete r
```

Calclulate the cluster coefficient.

```
match (u1:User {id:394})-[:InteractsWith]->(u2:User)
with collect(u2.id) as neighbours, count(u2) as k
match (u3:User)-[iw:InteractsWith]->(u4:User)
where (u3.id in (neighbours)) and (u4.id in (neighbours))
return count(iw)/(k * (k - 1) * 1.0) as clusteringCoefficient
```

Most Active Users (based on Cluster Coefficients)

User ID	Coefficient
394	0.9167
2067	0.7679
209	0.9524