**Technical Appendix**

**Catch the Pink Flamingo Analysis**

**Produced by: zhan jiefan**

Acquiring, Exploring and Preparing the Data

Data Exploration

Data Set Overview

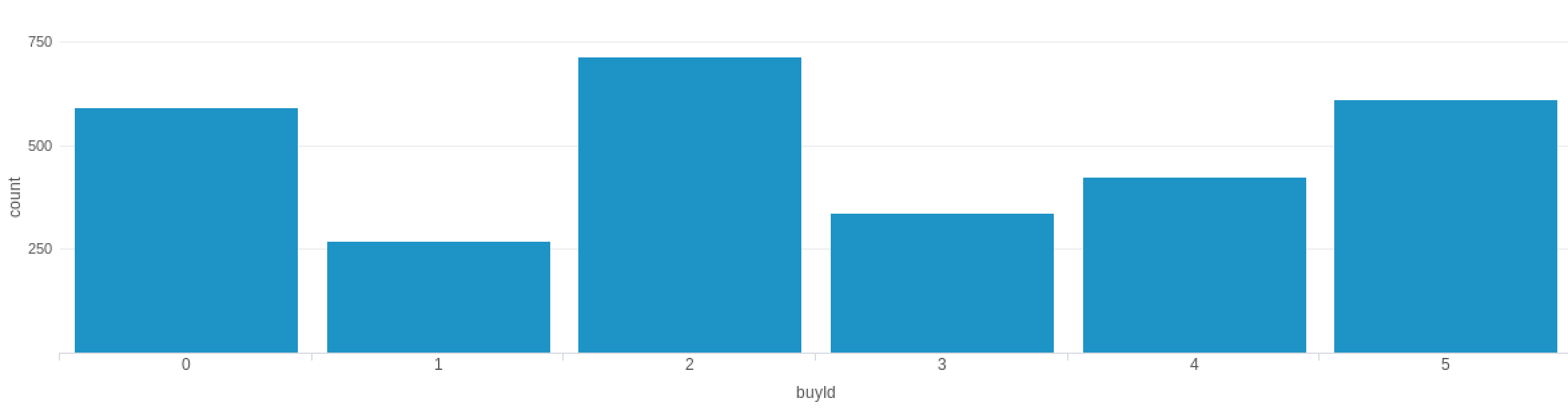
The table below lists each of the files available for analysis with a short description of what is found in each one.

|  |  |  |
| --- | --- | --- |
| **File Name** | **Description** | **Fields** |
| ad-clicks.csv | A line 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 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 | A line is added to this file when a player makes an in-app purchase in the Flamingo app. | **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 | 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 two-letter 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 | 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 |
| 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 |
| 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 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 |

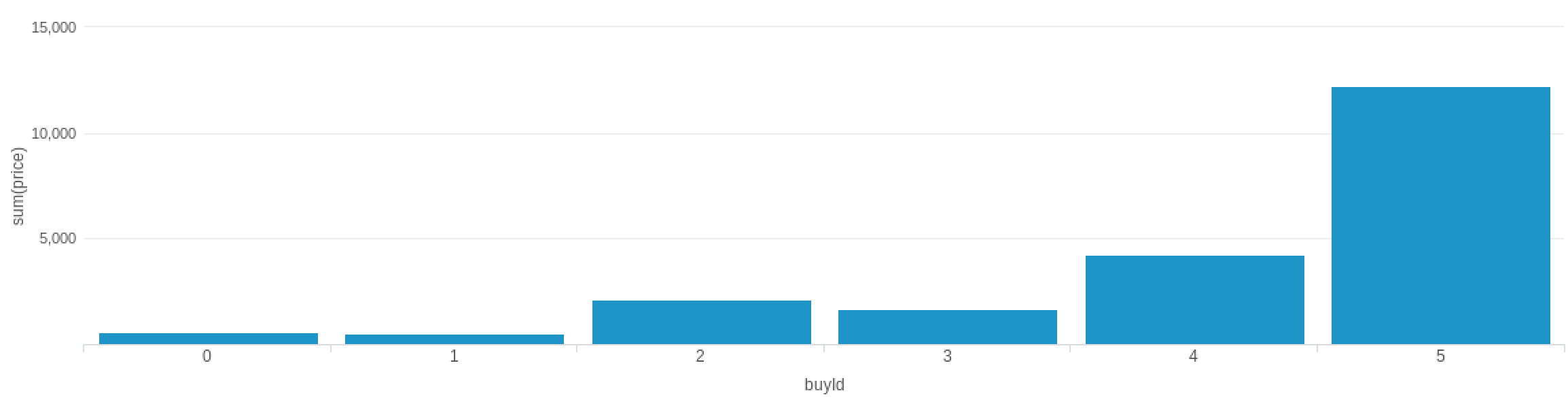
Aggregation

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

A histogram showing how many times each item is purchased:

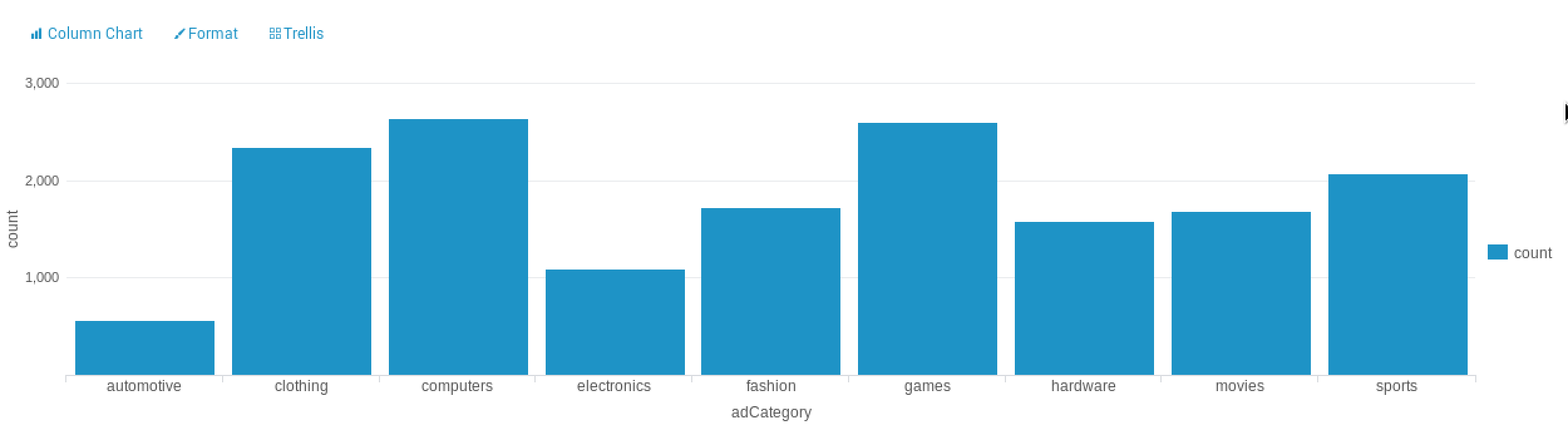


A histogram showing how much money was made from each item:



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 | 4928.25 |
| 2 - uneven | 0.55 | 0.60 | 0.55 | 5184.1 |

Data Classification Analysis

**Data Preparation**

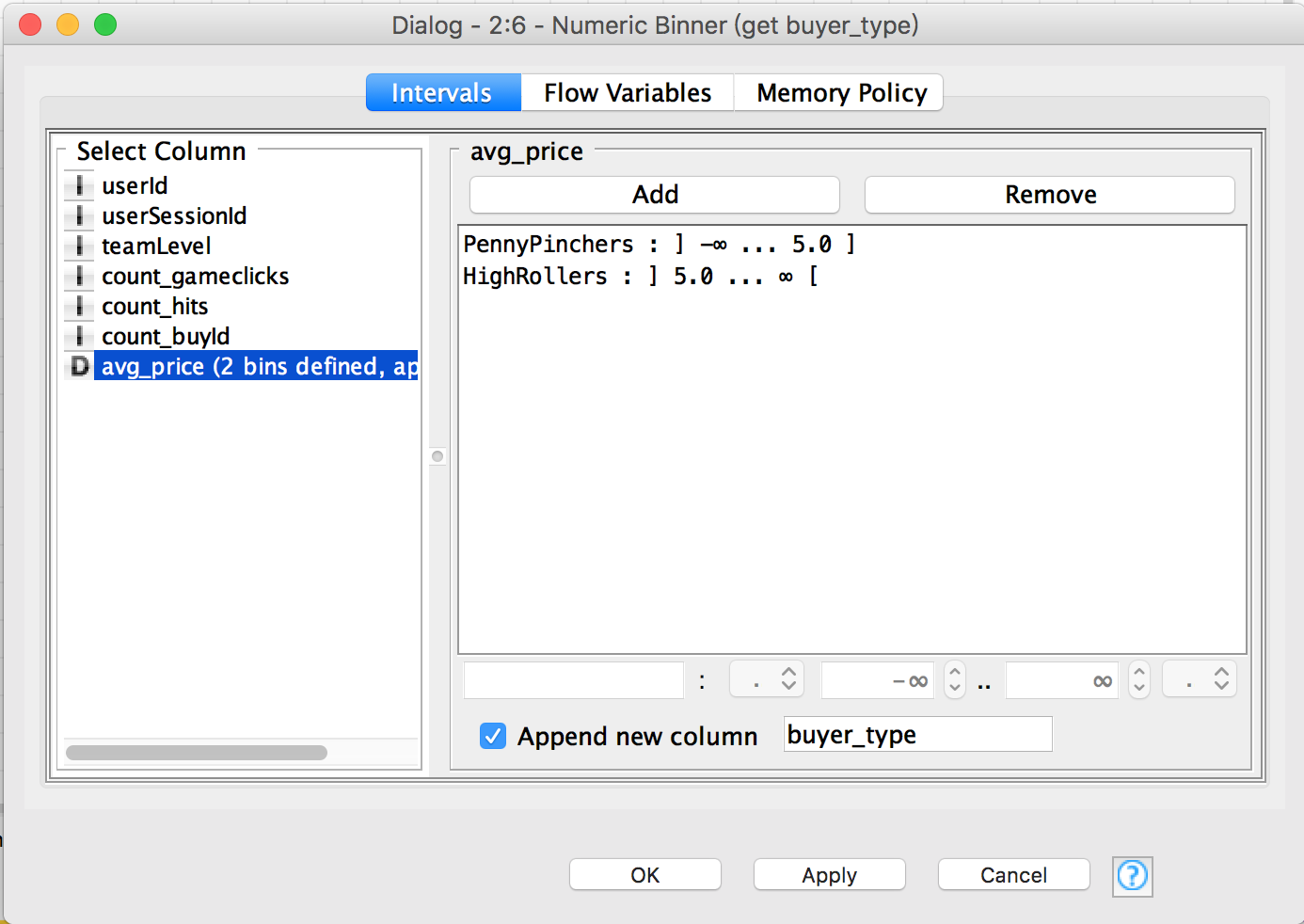
Analysis of combined\_data.csv

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 column of avg\_price represent average per orders of a user. If avg\_price greater than 5, then the user is HighRollers. If avg\_price less than or equal 5, then the user is PennyPinchers.

The creation of this new categorical attribute was necessary because the task is classifying users as HighRollers or PennyPinchers, we need categorial variate, instead of continuous variate.

.

Attribute Selection

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

|  |  |
| --- | --- |
| **Attribute** | **Rationale for Filtering** |
| userID | It is uniquely user id, has no sense to classify. |
| userSession | It is uniquely user session id, has no sense to classify. |
| avg-price | The target label derived from it |

**Data Partitioning and Modeling**

The data was partitioned into train and test datasets.

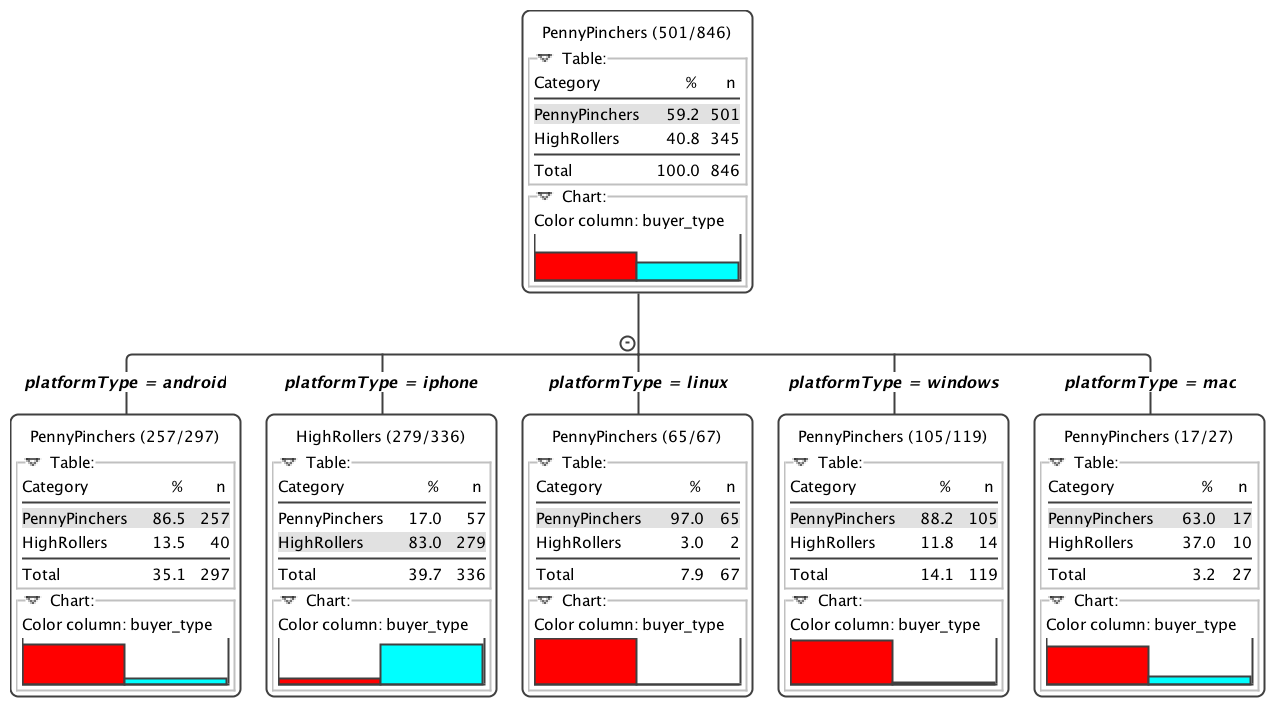
The train data set was used to create the decision tree model.

The trained model was then applied to the test dataset.

This is important because preventing model overfitting.

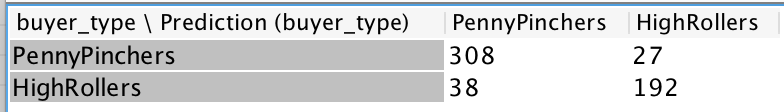
When partitioning the data using sampling, it is important to set the random seed because the train/test dataset will not change regardless rerun all workflow.

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%**

row1, col1: 308, real label is PennyPinchers, predict label is PennyPinchers, correctly predicted

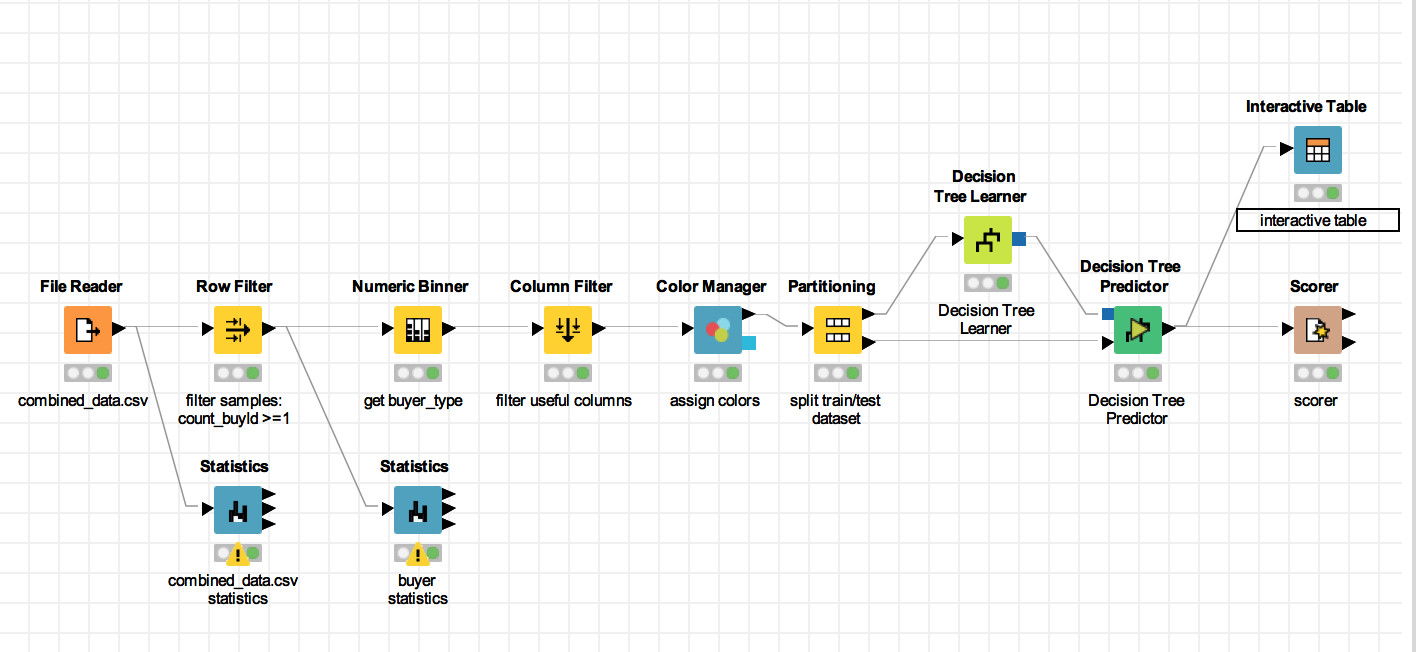
row1, col2: 27, real label is PennyPinchers, predict label is HighRollers, incorrectly predicted

row2, col1: 38 real label is HighRollers, predict label is PennyPinchers, incorrectly predicted

row2, col2: 192, real label is HighRollers, predict label is HighRollers, correctly predicted

**Analysis Conclusions**

The final KNIME workflow is shown below:



What makes a HighRoller?

PlatformType is a key attribute to classify users. If user use iphone, he or she high probability is a HighRoller. And if users not use iphont, he or she high probability is a PennyPicher.

|  |
| --- |
| **Specific Recommendations to Increase Revenue** |
| 1. Advertising to iphone users, to attract more new iphone users. |
| 2. Offering discount items to users who are not use iphone. |

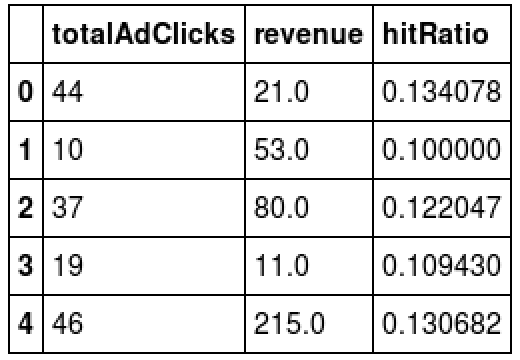
Clustering Analysis

**Attribute Selection**

|  |  |
| --- | --- |
| **Attribute** | **Rationale for Selection** |
| totalAdClicks | Total number of ad-clicks per user |
| revenue | Sum of money spent on item by each user |
| hitRatio | Hit ratio of game-clicks per user |

**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 \* 3

# of clusters created: 3

**Cluster Centers**

|  |  |
| --- | --- |
| **Cluster #** | **Cluster Center** |
| 1 | [ 34.144 , 67.448 , 0.1198328] |
| 2 | [ 41.0666667, 145.511111, 0.128167091] |
| 3 | [ 26.36461126, 17.12600536, 0.1103672 ] |

These clusters can be differentiated from each other as follows:

The features is totalAdClicks, revenue, hitRatio.

Cluster 1 is different from the others in that median totalAdClicks, median revenue, median hitRatio.

Cluster 2 is different from the others in that high totalAdClicks, high revenue, high hitRatio.

Cluster 3 is different from the others in that low totalAdClicks, low revenue, low hitRatio.

**Recommended Actions**

|  |  |
| --- | --- |
| **Action Recommended** | **Rationale for the action** |
| Add ads to cluster 3 | Cluster 3 buy little items, so we can increase ads revenue from them. |
| Deep research cluster 2, keep them retention | Cluster 2 are high value users, we should attention to their churn tendency, keep them retention to get more income. |

Graph Analytics Analysis

**Modeling Chat Data using a Graph Data Model**

It’s a chat graph. It’s contain user create chat session, user joins a chat session and user leaves a chat session. It’s also contain user chats in a chat session, and the chat may mentions other user or responds to other user.

**Creation of the Graph Database for Chats**

Describe the steps you took for creating the graph database.

1. **Write the schema of the 6 CSV files**

**chat\_create\_team\_chat.csv:** userid, teamid, TeamChatSessionID, timestamp

A line is added to this file when a player creates a new chat with their team.

**chat\_join\_team\_chat.csv:** userid, TeamChatSessionID, timestamp

Creates an edge labeled "Joins" from User to TeamChatSession. The columns are the User id,

TeamChatSession id and the timestamp of the Joins edge.

**chat\_leave\_team\_chat.csv:** userid, TeamChatSessionID, timestamp

Creates an edge labeled "Leaves" from User to TeamChatSession. The columns are the User id,

TeamChatSession id and the timestamp of the Leaves edge.

**chat\_item\_team\_chat.csv:** userid, TeamChatSessionID, chatitemid, timestamp

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. This edge should also have a timeStamp property using the value from Column

3.

**chat\_mention\_team\_chat.csv:** ChatItem, userid, timeStamp

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.

**chat\_respond\_team\_chat.csv:** chatid1, chatid2,timestamp

A line is added to this file when player with chatid2 responds to a chat post by another player with

chatid1.

1. **Explain the loading process and include a sample LOAD command**

The first line gives the path of the file.

Then create nodes and attributes through MERGE.

Finally, create edges (source node, destination node, and relative attribute) through MERGE.

LOAD CSV FROM "file:/Users/hahadsg/Downloads/tmp/z/big\_data\_capstone\_datasets\_and\_scripts/chat-data/chat\_create\_team\_chat.csv" AS row

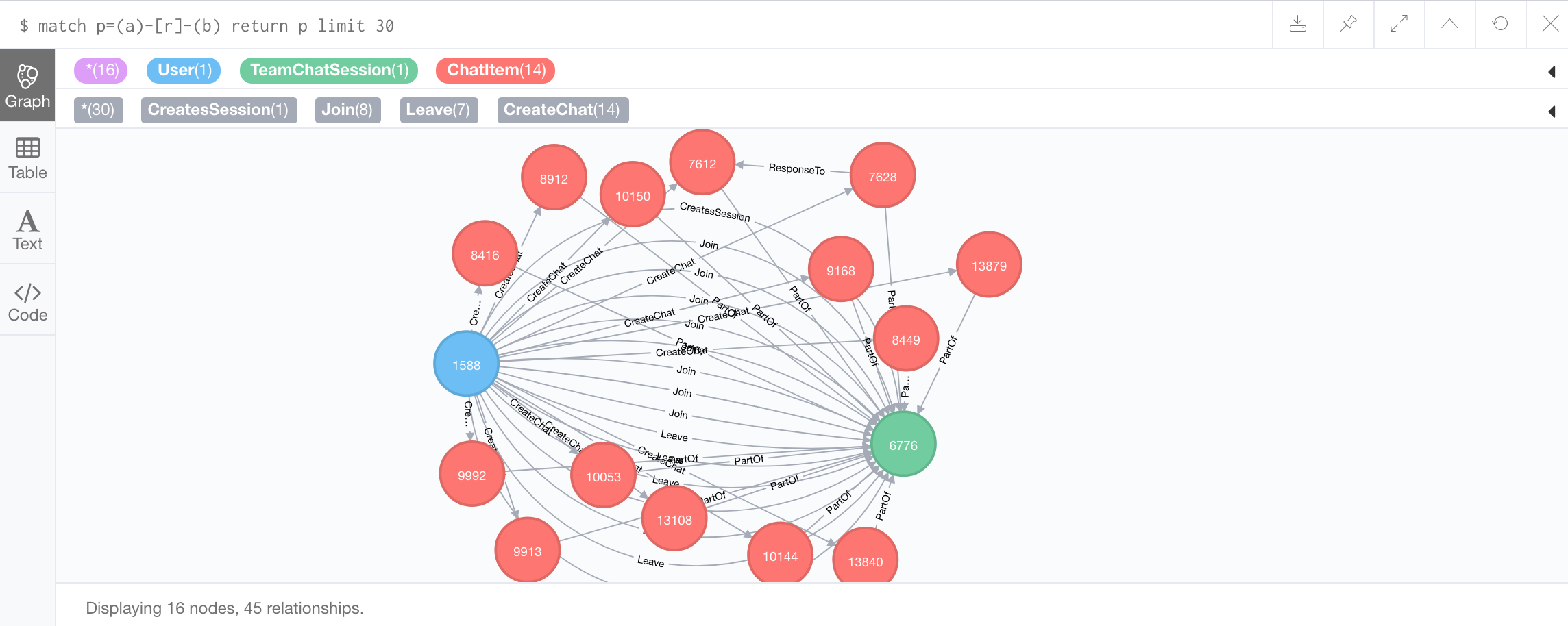
MERGE (u:User {id: toInt(row[0])}) MERGE (t:Team {id: toInt(row[1])})

MERGE (c:TeamChatSession {id: toInt(row[2])})

MERGE (u)-[:CreatesSession{timeStamp: row[3]}]->(c)

MERGE (c)-[:OwnedBy{timeStamp: row[3]}]->(t)

1. **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.**

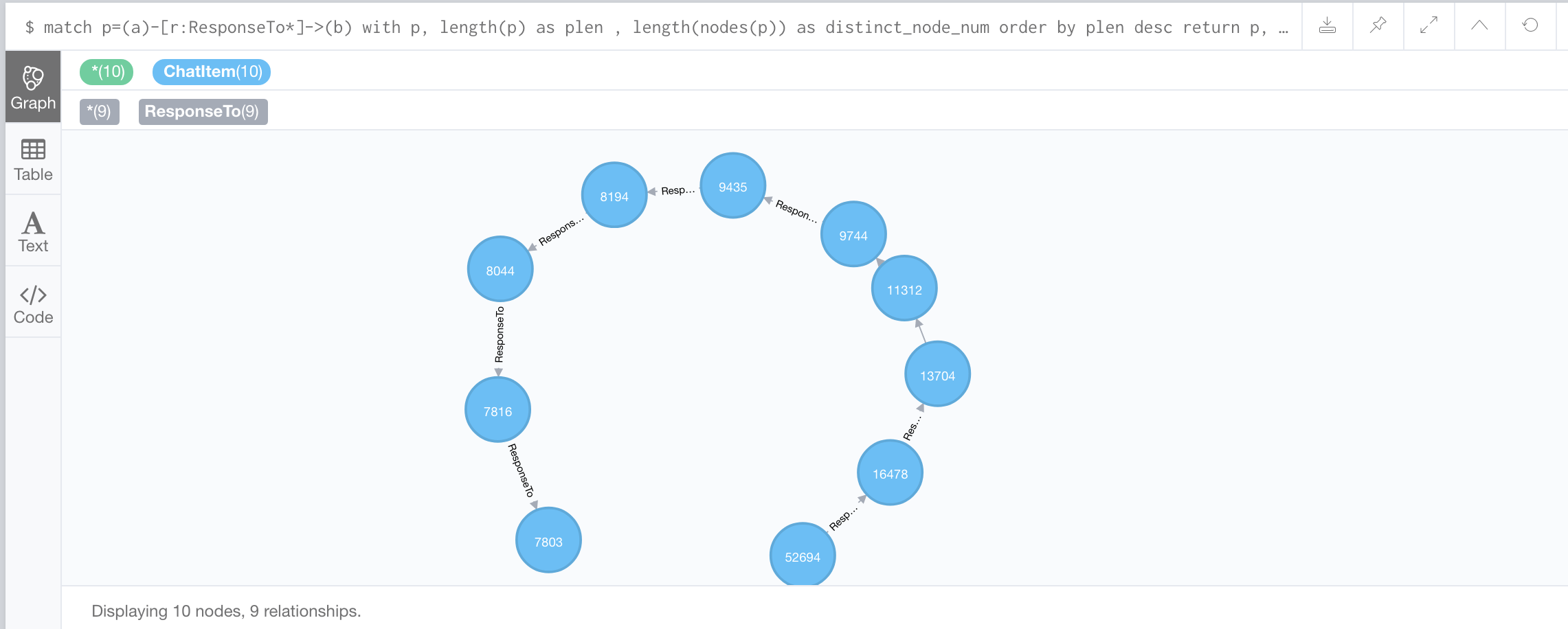


**Finding the longest conversation chain and its participants**

Report the results including the length of the conversation (path length) and how many unique users were part of the conversation chain. Include an image of the graph with the longest conversation chain.

The length of longest conversation is 9.

And 5 unique users were part of the longest conversation chain

****

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

Include your table containing the top 3 chattiest users and teams below, and report whether or not any of the chattiest users are part of any of the chattiest teams.

**Chattiest Users**

|  |  |
| --- | --- |
| **Users** | **Number of Chats** |
| 394 | 115 |
| 2067 | 111 |
| 1087 | 109 |

**Chattiest Teams**

|  |  |
| --- | --- |
| **Teams** | **Number of Chats** |
| 82 | 1324 |
| 185 | 1036 |
| 112 | 957 |

The User 999 in Team 52

**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 | 0.9523809523809523 |
| 554 | 0.9047619047619048 |
| 1087 | 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.

**Recommendation (learn from classification)**

1. Advertising to iphone users, to attract more new iphone users.
2. Offering discount items to users who are not use iphone.

**Recommendation (learn from classification)**

1. Add ads to cluster 3: Cluster 3 buy little items, so we can increase ads revenue from them.
2. Deep research cluster 2, keep them retention: Cluster 2 are high value users, we should attention to their churn tendency, keep them retention to get more income.