**Kedir Nasir Omer Date: Nov. 28/2022**

**UC SanDiego- Big Data Specialization**

***5. Technical Appendix***

* ***Catch the Pink Flamingo Analysis***

1. **Data Exploration**

* Overview of the Data Set

* The available files are listed in the table below, along with a brief summary of what can be 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 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 | Database of purchases. | 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 |

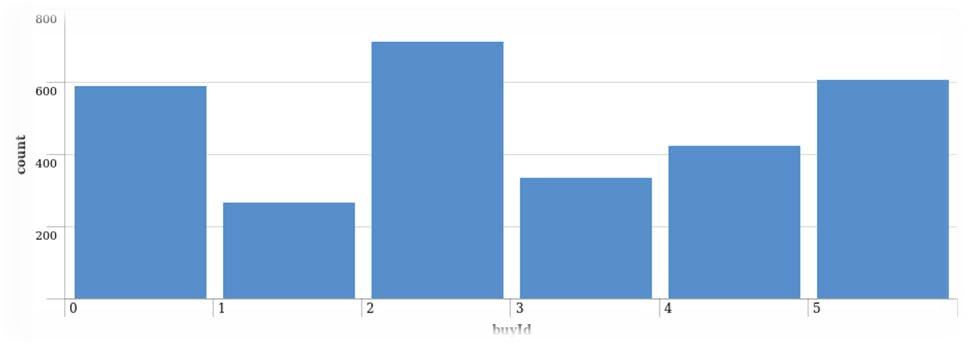
|  |  |  |
| --- | --- | --- |
|  |  | team: the current team id of the user who made the purchase  userId: the user id of the user who made the purchase buyId: the id of the item purchased  price: the price of the item purchased |
| game-clicks.csv | A record of each click a user performed during the 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.    When a team levels up, each current user session ends and a | timestamp: a timestamp denoting when the event occurred. |
|  | new session begins with the new level. | 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. |

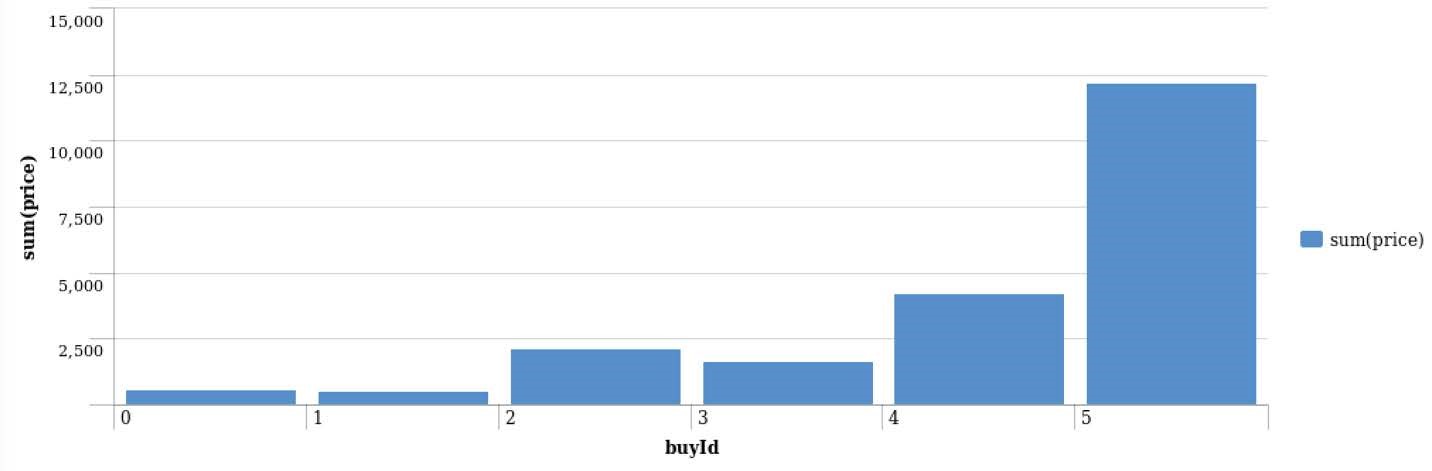
* Aggregation

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

* A histogram displaying the frequency of purchase for each item:

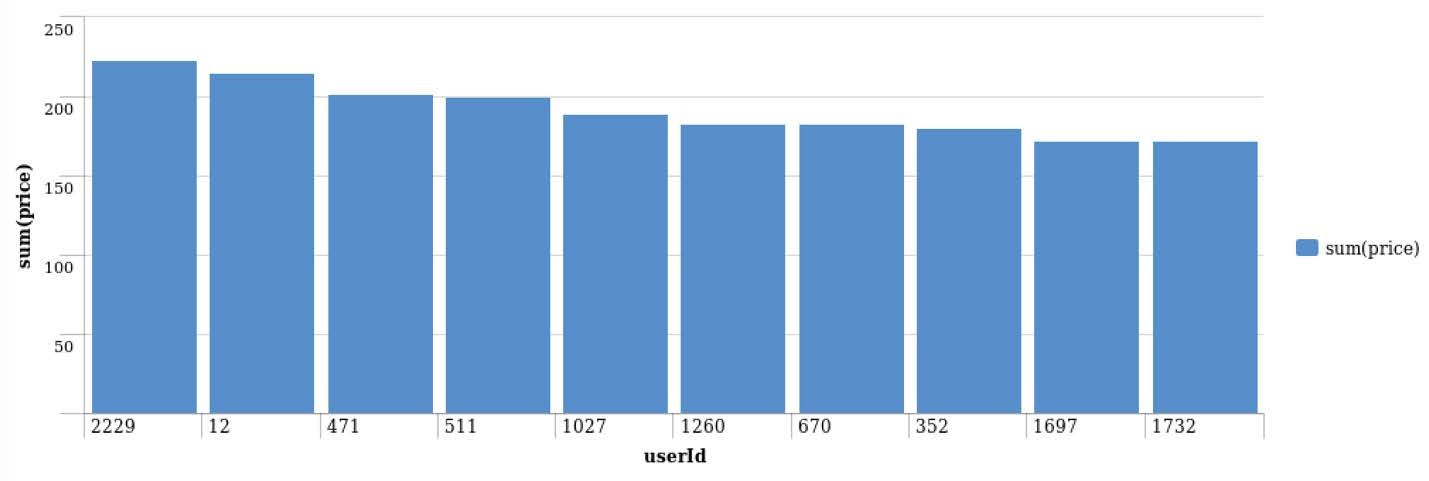


* A histogram displaying the revenue generated by each item:



* Filtering

* A histogram of the top ten users' combined spending amounts (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% |

**2.1. Data Preparation**

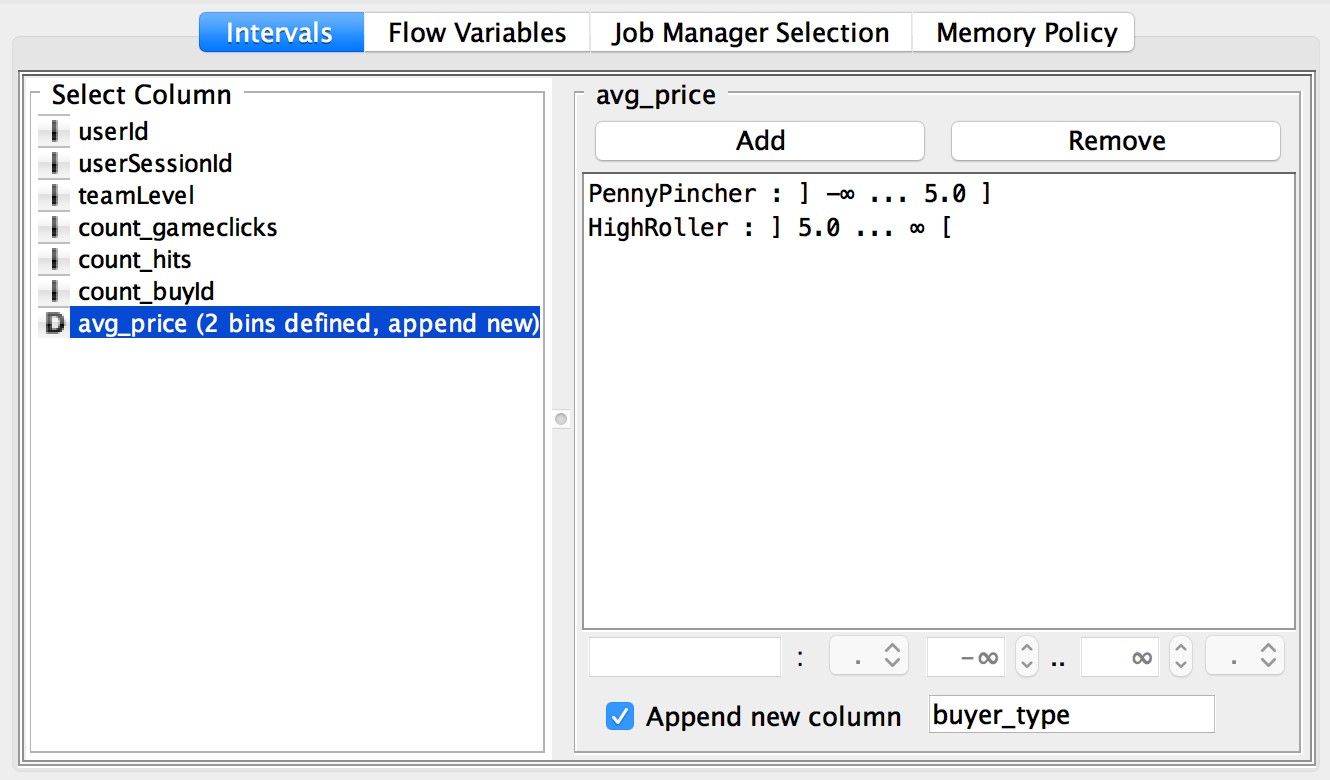
* **Classifying in KNIME to identify big spenders in Catch the Pink Flamingo**
* Analyzing the combined\_data.csv

**Sample Selection**

|  |  |
| --- | --- |
| **Item** | **Amount** |
| # of Samples | 4619 |
| # of Samples with Purchases | 1411 |

* **Attribute Creation**

For the purpose of categorizing players into two groups for analysis, a new categorical attribute was developed (**HighRollers** and **PennyPinchers**).



* **Describing the design attribute**

High rollers are those who make purchases that cost more than $5. We can categorize users in the appropriate way by creating a new column based on the avg price.

Because our objective is to comprehend the characteristics of those who make significant purchases, the development of this new categorical attribute was necessary. We will build our decision tree around this categorical variable.

* **Attribute Selection**

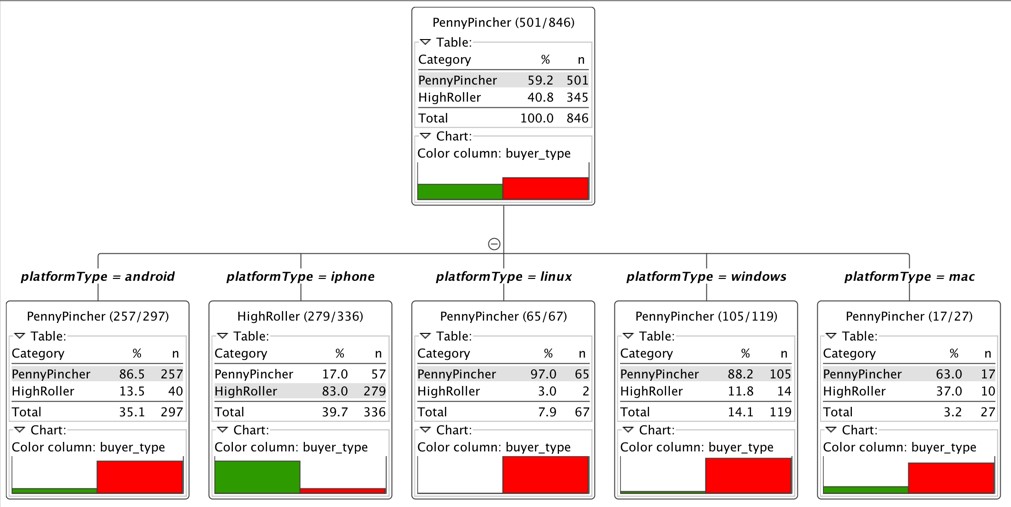
For the reasons listed below, the following attributes were excluded from the dataset

|  |  |
| --- | --- |
| **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. |

**2.2. Data Partitioning and Modeling**

* **Classifying in KNIME to identify big spenders in Catch the Pink Flamingo**
* The training data set was used to build the decision tree model, and the data was divided into test and training datasets.
* The test dataset was then fed into the trained model.
* This is significant because it enables us to check the precision of the trained model by dividing the data set into training and test data.
* Setting the random seed is crucial when using sampling to partition the data because it enables you to get consistent results every time you run the partition.

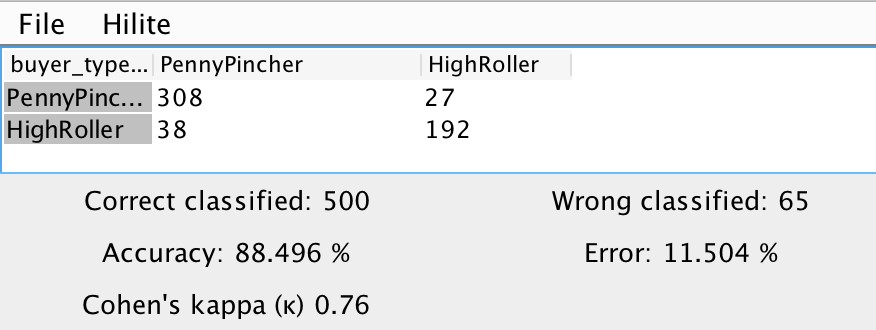
Below is a screenshot of the resulting decision tree:



**2.3. Evaluation**

* **Classifying in KNIME to identify big spenders in Catch the Pink Flamingo**

Below is a screenshot of the confusion matrix

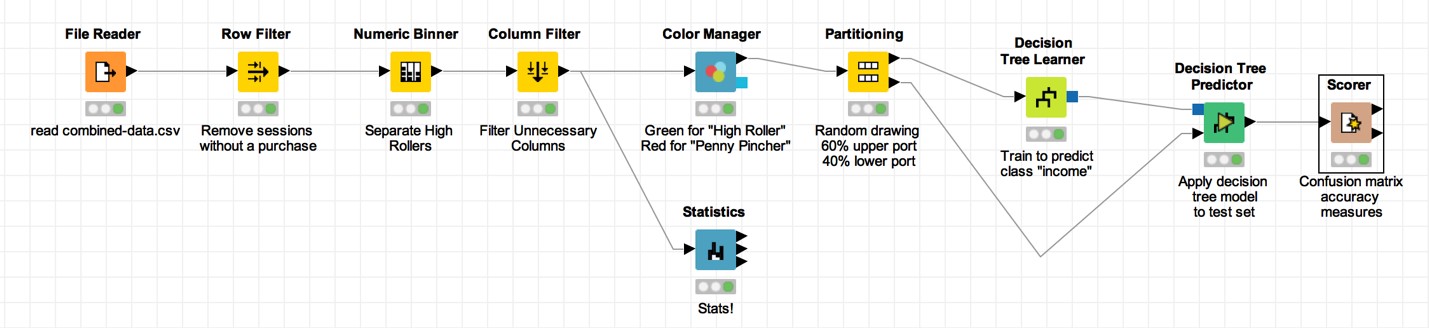


* The above screenshot demonstrates that the model's **overall accuracy** is **88.496%.**
* **PennyPincher** buyers were classified correctly **308 times** and incorrectly 27 times by the model. When the byer type is a **HighRoller**, the model correctly identified the byer type **192 times** while misclassifying it 38 times.

**2.4. Analysis Conclusions**

* **Classifying in KNIME to identify big spenders in Catch the Pink Flamingo**

Below is the completed KNIME workflow



* **PennyPincher vs HighRoller**
* iOS is used by HighRoller users.
* PennyPincher uses Linux, Mac, Android, and Windows.
* **Specifc Recommendation to Increase Revenue**
* Promote iOS HighOller
* Future product creation ought to concentrate on iOS.

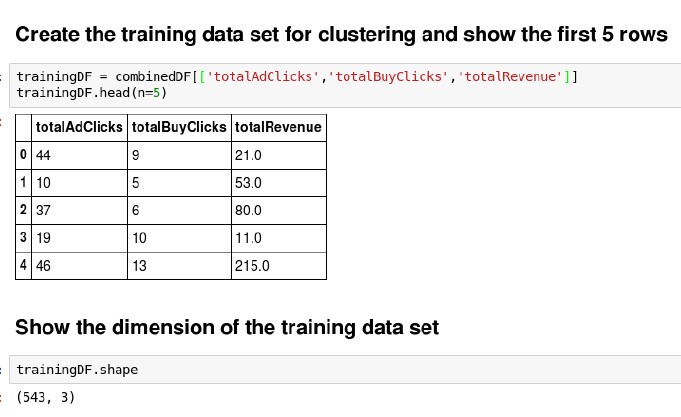
**3.1. Attribute Selection**

* **Recommending Actions from Clustering Analysis**

|  |  |
| --- | --- |
| 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. |

**3.2. Training Data Set Creation**

* **Recommending Actions from Clustering Analysis**

The first five lines of the training data set used for this analysis are displayed below

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

**3.3. Cluster Centers**

* **Recommending Actions from Clustering Analysis**

|  |  |
| --- | --- |
| **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 distinguished from one another in the following ways
* In contrast to the other clusters, **Cluster 1's** participants have the **highest** "totalAdClics," "totalBuyClics," and "totalRevenue."
* The players in **Cluster 2** differ from the others in that they have the **second-highest** "totalAdClics," "totalBuyClics," and "totalRevenue."
* The players in **Cluster 3** differ from the others in that they have the **lowest** "totalAdClics," "totalBuyClics," and "totalRevenue" numbers.

**3.4. Recommended Actions**

* **Recommending Actions from Clustering Analysis**

|  |  |
| --- | --- |
| **Recommended Action** | **Justification for the action** |
| Increase the cost of the ads that are displayed to players in the first cluster. | Players in the first cluster click on advertisements frequently, which raises the cost of their ads and potentially boosts revenue for the business. |
| Charge players a lower fee in the third cluster of in-app purchase items. | Only lower-priced items are purchased by players in the third cluster. Giving them coupons or reducing the cost of the in-app purchase could persuade them to spend more. |

**4. Graph Analytics with Chat Data Using Neo4j**

* **A Graph Analytics Approach to Simulated Chat Data**

1. Modeling Chat Data using a Graph Data Model

A network based on user chat interactions makes up the graph model. Users on the same team can join and leave a chat session, which can be started by one of them. When a user creates a post, user interaction starts. Another user may be mentioned by a user. With a timestamp, every relationship between entities is recorded.

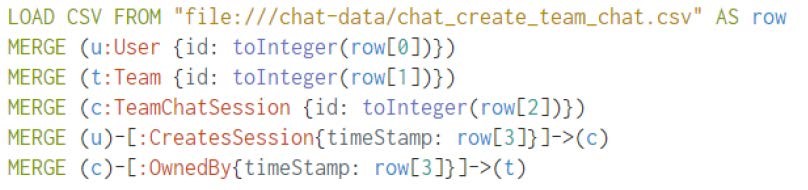
1. Creation of the Graph Database for Chats

Describe the steps you took for creating the graph database.

* Write the schema of the 6 CSV files

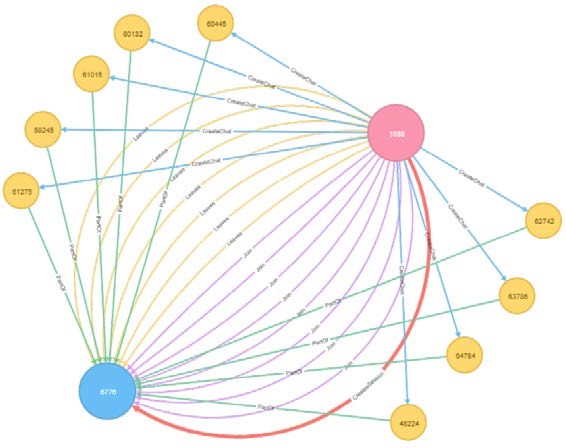
|  |  |
| --- | --- |
| chat\_create\_team\_chat.csv | userID teamID teamChatSessionID timestamp |
| chat\_join\_team\_chat.csv | userID teamChatSessionID timestamp |
| chat\_leave\_team\_chat.csv | userID teamChatSessionID timestamp |
| chat\_item\_team\_chat.csv | userID teamChatSessionID chatItemID 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



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.

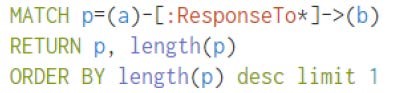
Present a screenshot of some part of the graph you have generated. The graphs must include clearly visible examples of most node and edge types.

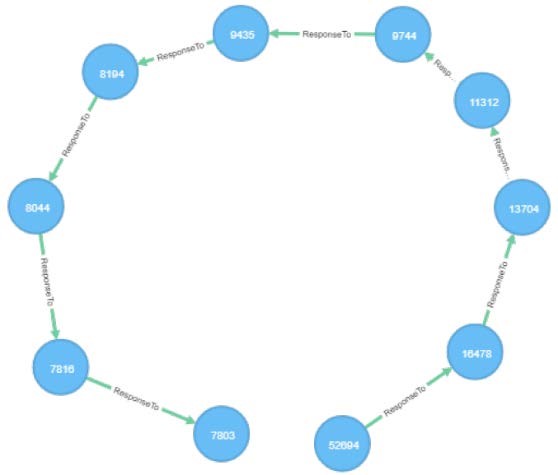


1. Finding the longest conversation chain and its participants

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 chats are involved in it?





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?



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

1. 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

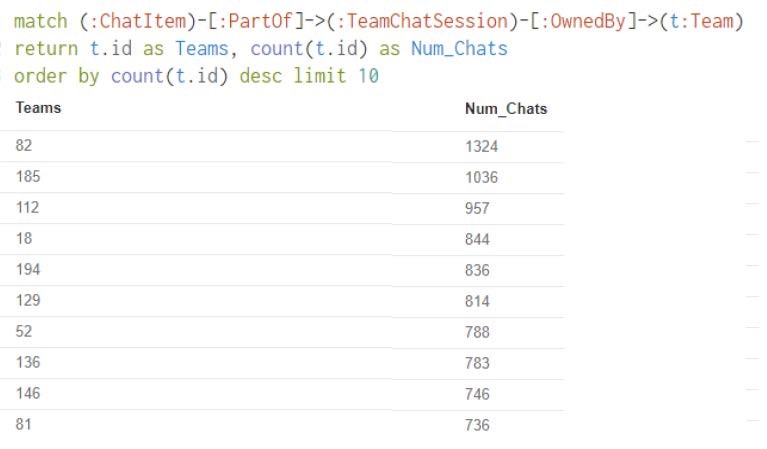
* Determine the number of chats created by a user from the CreateChat edge



|  |  |
| --- | --- |
| Users | Number of Chats |
| 394 | 115 |
| 2067 | 111 |
| 209 | 109 |

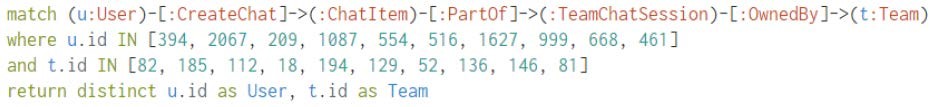
Chattiest Teams

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.



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

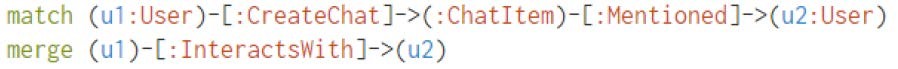
* Finally, present your answer, i.e. whether or not any of the chattiest users are part of any of the chattiest teams.



This query is used to investigate if the chattiest user is 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



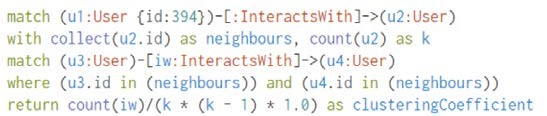
* Connect users’ responses with the chat creator



* Eliminate all self interaction



* Calculate the cluster coefficient.



Most Active Users (based on Cluster Coefficients)

|  |  |
| --- | --- |
| User ID | Coefficient |
| 394 | 0.9167 |
| 2067 | 0.7679 |
| 209 | 0.9524 |