Team: Big Data Bandits
Blessy Chinthapalli
Meghana Kanthadai
Rachita Harit
Shriya Yegalapati

Homework 5: What Happens in Vegas, stays in Venmo (Group)

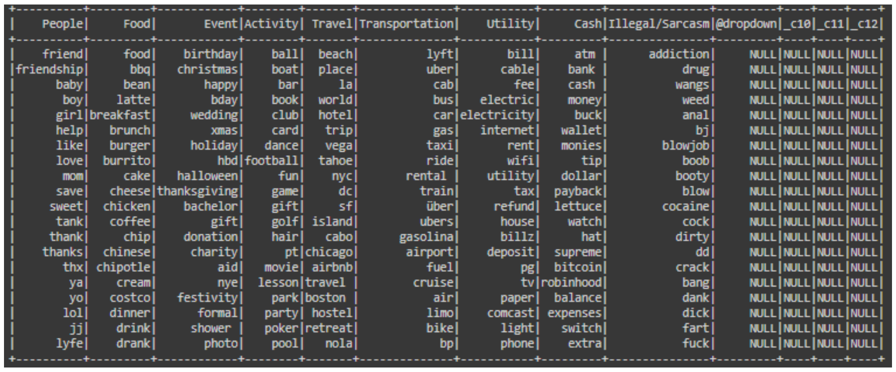
Part 1: Text Analytics

The script is designed to process and analyze Venmo transaction data to uncover user spending patterns and trends over time. It begins by initializing a Spark session to handle large-scale data processing efficiently and loads the transaction data from a Parquet file. To classify transaction descriptions, the script loads emoji and word classification dictionaries from CSV files and converts them into dictionaries for quick lookup.

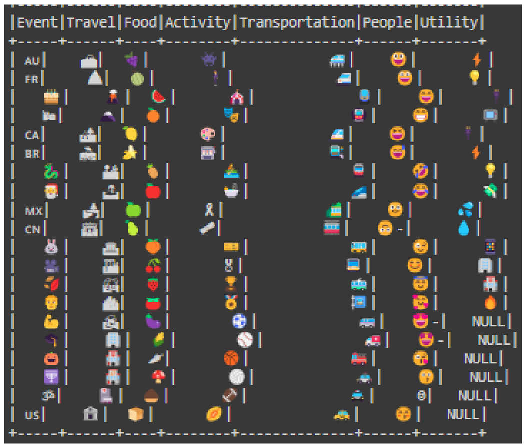
Each transaction description is split into individual words, and a user-defined function (UDF) is applied to classify each word based on the dictionaries. Words with unknown classifications are filtered out, and the remaining classifications are standardized to lowercase to ensure consistency. The script counts the occurrences of each classification for each combination of user, datetime, and story ID, and then creates a pivot table to restructure the data, making it easier to analyze. Null values in the pivot table are replaced with zeros to ensure accurate calculations.

The total count of classifications for each transaction is computed, and these counts are aggregated by the user to summarize each user's activity across different categories. The script then calculates the date of each user's first transaction and determines the month interval from this date for all subsequent transactions. Transactions are filtered to include only the first 12 months, allowing the analysis to focus on the initial spending patterns of users.

For each month interval, the script calculates the average and standard deviation of spending in each category, converting the results into a Pandas DataFrame for visualization. The final step involves creating a plot that displays the average spending per category with confidence intervals (±2 standard deviations) over the first 12 months. This visual representation helps to identify trends and fluctuations in user spending behavior, providing valuable insights into how users allocate their expenses across different categories over time.



```
+---------+---------+-----------+--------+-------+---------------+-----------+---------+---------------+---------+----+----+----+
|   People|     Food|      Event|Activity| Travel|Transportation|    Utility|     Cash|Illegal/Sarcasm|@dropdown|_c10|_c11|_c12|
+---------+---------+-----------+--------+-------+---------------+-----------+---------+---------------+---------+----+----+----+
|   friend|     food|   birthday|    ball|  beach|           lyft|       bill|     atm |     addiction|     NULL|NULL|NULL|NULL|
|friendship|     bbq|  christmas|    boat|  place|           uber|      cable|    bank |          drug|     NULL|NULL|NULL|NULL|
|     baby|     bean|      happy|     bar|     la|            cab|        fee|    cash |         wangs|     NULL|NULL|NULL|NULL|
|      boy|    latte|       bday|    book|  world|            bus|   electric|    money|          weed|     NULL|NULL|NULL|NULL|
|     girl|breakfast|    wedding|    club|  hotel|            car|electricity|     buck|          anal|     NULL|NULL|NULL|NULL|
|     help|   brunch|       xmas|    card|   trip|            gas|   internet|   wallet|            bj|     NULL|NULL|NULL|NULL|
|     like|   burger|    holiday|   dance|   vega|           taxi|       rent|   monies|       blowjob|     NULL|NULL|NULL|NULL|
|     love|  burrito|        hbd|football|  tahoe|           ride|       wifi|      tip|          boob|     NULL|NULL|NULL|NULL|
|      mom|     cake|  halloween|     fun|    nyc|        rental |    utility|   dollar|         booty|     NULL|NULL|NULL|NULL|
|     save|   cheese|thanksgiving|   game|     dc|          train|        tax|  payback|          blow|     NULL|NULL|NULL|NULL|
|    sweet|  chicken|   bachelor|    gift|     sf|           über|     refund|  lettuce|       cocaine|     NULL|NULL|NULL|NULL|
|     tank|   coffee|       gift|    golf| island|          ubers|      house|    watch|          cock|     NULL|NULL|NULL|NULL|
|    thank|     chip|   donation|    hair|   cabo|       gasolina|      billz|      hat|         dirty|     NULL|NULL|NULL|NULL|
|   thanks|  chinese|    charity|      pt|chicago|        airport|    deposit|  supreme|            dd|     NULL|NULL|NULL|NULL|
|      thx| chipotle|        aid|   movie| airbnb|           fuel|         pg|  bitcoin|         crack|     NULL|NULL|NULL|NULL|
|       ya|    cream|        nye|  lesson|travel |         cruise|         tv|robinhood|          bang|     NULL|NULL|NULL|NULL|
|       yo|   costco|  festivity|    park|boston |            air|      paper|  balance|          dank|     NULL|NULL|NULL|NULL|
|      lol|   dinner|     formal|   party| hostel|           limo|    comcast| expenses|          dick|     NULL|NULL|NULL|NULL|
|       jj|    drink|    shower |   poker|retreat|           bike|      light|   switch|          fart|     NULL|NULL|NULL|NULL|
|     lyfe|    drank|      photo|    pool|   nola|             bp|      phone|    extra|          fuck|     NULL|NULL|NULL|NULL|
+---------+---------+-----------+--------+-------+---------------+-----------+---------+---------------+---------+----+----+----+
```



```
|Event|Travel|Food|Activity|Transportation|People|Utility|
+-----+------+----+--------+--------------+------+-------+
|  AU|    🧳|  🍇|      👾|            🚅|    😛|     ⚡|
|  FR|    ⛺|  🍈|      🚶|            🚄|    😃|     💡|
|   🎂|    🌋|  🍉|      🎪|            🚇|    😄|     🔌|
|   [illegible]|    🗻|  🍊|      🎭|            🚋|    😁|     📺|
|  CA|    🏔|  🍋|      🎨|            🚅|    😆|     🔌|
|  BR|    🏭|  🍌|      🎰|            🚈|    😅|     ⚡|
|   🐉|    🏟|  🍍|      🏄|            🚃|    🤣|     💡|
|   🎅|    🗽|  🍎|      🥣|            🚄|    😂|     💸|
|  MX|    🏜|  🍏|      🎗|            🚞|    😵|     💦|
|  CN|    🏯|  🍐|      🏏|            🚋|    😳-|     💧|
|   🐰|    🏛|  🍑|      🎫|            🚌|    😉|     📱|
|   🐇|    🏗|  🍒|      🎖|            🚍|    😊|     🏢|
|   🍂|    🏟|  🍓|      🏆|            🚎|    😇|     🏥|
|   👱|    ⛪|  🍅|      🏅|            🚏|    🥰|     🔥|
|   💪|    🏗|  🍆|      ⚽|            🚐|    😍-|   NULL|
|   🎓|    🏢|  🌽|      ⚾|            🚑|    😍-|   NULL|
|   🎃|    🏨|  🔪|      🏀|            🚒|    😘|   NULL|
|   🎁|    🏨|  🍄|      🏐|            🚓|    😯|   NULL|
|   🕉|    🏩|  🌰|      🏈|            🚔|    ☺|   NULL|
|  US|    🏠|  🍞|      🏉|            🚕|    😚|   NULL|
+-----+------+----+--------+--------------+------+-------+
```

Part-2 Social Network Analytics

Q5 [10 pts]: Write a script to find a user's friends and friends of friends (**Friend definition:** A user's friend is someone who has transacted with the user, either sending money to the user or receiving money from the user). **Describe your algorithm and calculate its computational complexity. Can you do it better?**

Assumptions:

1. Transactions data is available in the format of pairs (user1, user2) indicating a transaction between user1 and user2.
2. The transaction data is represented as an edge list in a graph.
3. The graph is undirected since a transaction between user1 and user2 implies a mutual connection.

Algorithm:

Step 1: Build the Adjacency List

1. Initialize an empty dictionary adjacency_list where each key is a user, and the value is a set of users they've transacted with.
2. Iterate through each transaction (user1, user2):
   - Add user2 to the set of user1's transactions.
   - Add user1 to the set of user2's transactions.

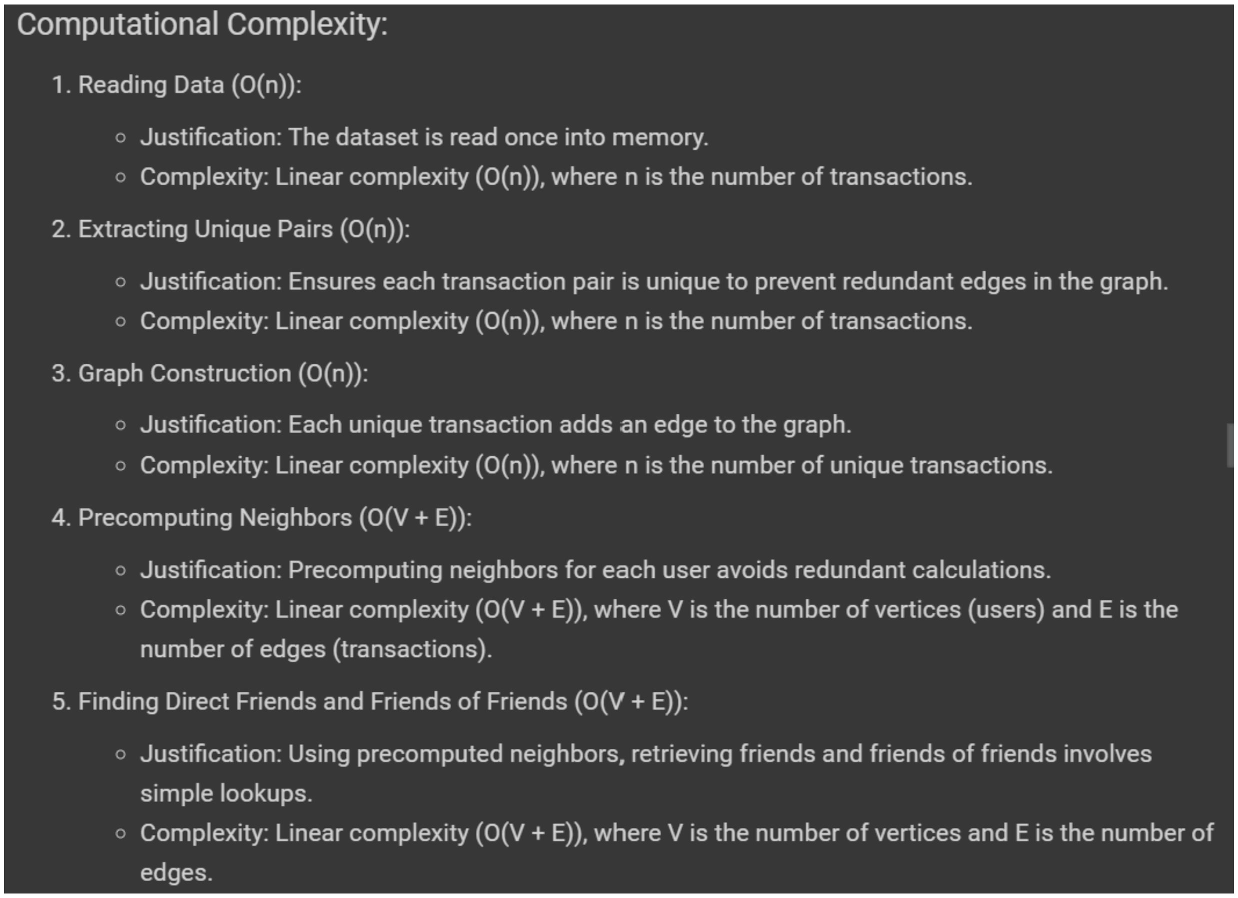
Step 2: Finding Direct Friends

1. Retrieve the set of direct friends from the adjacency list using the given user ID.

Step 3: Finding Friends of Friends

1. Initialize an empty set friends_of_friends.
2. Iterate through each friend in the set of direct friends:
   - For each friend, retrieve their set of friends from the adjacency list.
   - For each friend of friend, if they are not the original user and not in the set of direct friends, add them to the friends_of_friends set.

Computational Complexity: copying from our google colab notebook

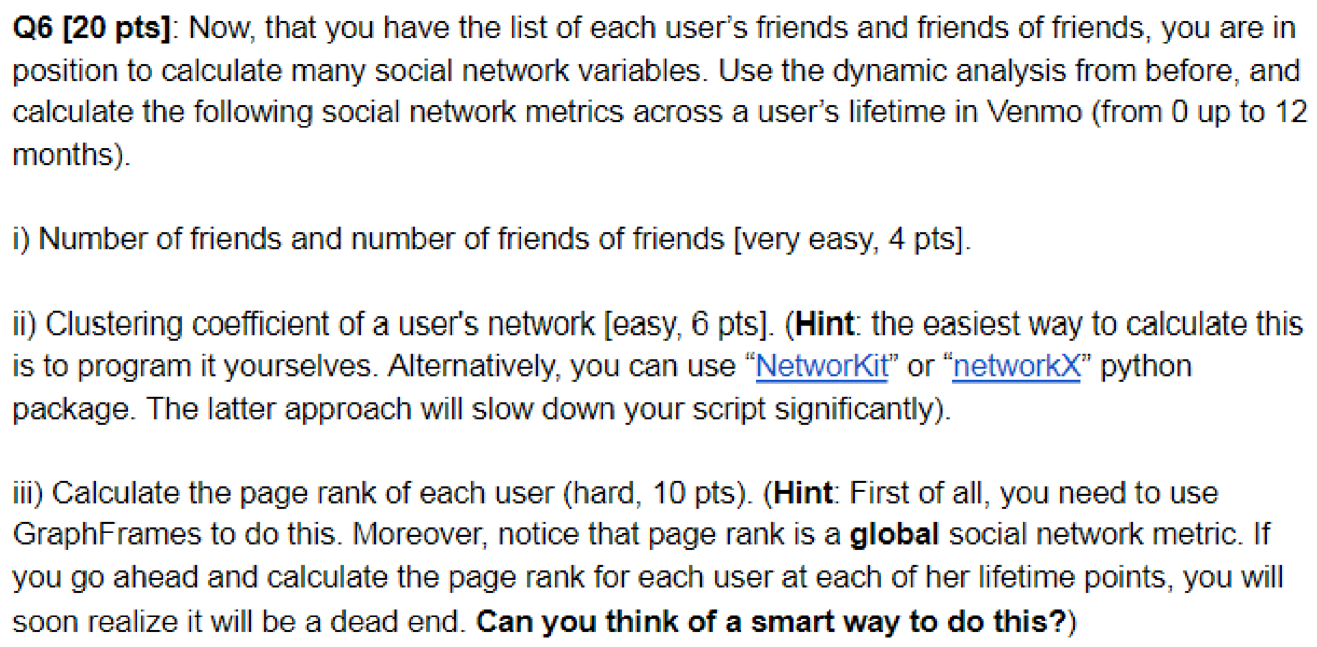


Computational Complexity:

1. Reading Data (O(n)):
   - Justification: The dataset is read once into memory.
   - Complexity: Linear complexity (O(n)), where n is the number of transactions.
2. Extracting Unique Pairs (O(n)):
   - Justification: Ensures each transaction pair is unique to prevent redundant edges in the graph.
   - Complexity: Linear complexity (O(n)), where n is the number of transactions.
3. Graph Construction (O(n)):
   - Justification: Each unique transaction adds an edge to the graph.
   - Complexity: Linear complexity (O(n)), where n is the number of unique transactions.
4. Precomputing Neighbors (O(V + E)):
   - Justification: Precomputing neighbors for each user avoids redundant calculations.
   - Complexity: Linear complexity (O(V + E)), where V is the number of vertices (users) and E is the number of edges (transactions).
5. Finding Direct Friends and Friends of Friends (O(V + E)):
   - Justification: Using precomputed neighbors, retrieving friends and friends of friends involves simple lookups.
   - Complexity: Linear complexity (O(V + E)), where V is the number of vertices and E is the number of edges.

Can you do better?

Yes, the algorithm can be improved by using sets for storing and checking membership of friends and friends of friends, which provides average O(1) time complexity for these operations. Precomputing neighbors for each user avoids redundant calculations and allows for faster lookups. Additionally, leveraging parallel processing capabilities in PySpark can distribute the computation across multiple nodes, significantly speeding up the process for large datasets. These optimizations reduce the overall computational complexity, making the algorithm more efficient and scalable.


Q6 [20 pts]: Now, that you have the list of each user's friends and friends of friends, you are in position to calculate many social network variables. Use the dynamic analysis from before, and calculate the following social network metrics across a user's lifetime in Venmo (from 0 up to 12 months).

i) Number of friends and number of friends of friends [very easy, 4 pts].

ii) Clustering coefficient of a user's network [easy, 6 pts]. (**Hint**: the easiest way to calculate this is to program it yourselves. Alternatively, you can use "NetworKit" or "networkX" python package. The latter approach will slow down your script significantly).

iii) Calculate the page rank of each user (hard, 10 pts). (**Hint**: First of all, you need to use GraphFrames to do this. Moreover, notice that page rank is a **global** social network metric. If you go ahead and calculate the page rank for each user at each of her lifetime points, you will soon realize it will be a dead end. **Can you think of a smart way to do this?**)

Assumptions for Each Question

Q6: Social Network Metrics Analysis

i) Number of friends and number of friends of friends
Assumptions:
1. Transactions data is available in the format of pairs (user1, user2), indicating a transaction between user1 and user2.
2. The graph representing the social network is undirected since a transaction between user1 and user2 implies a mutual connection.
3. Each user can be uniquely identified by their ID.
4. We use NetworkX to represent and process the graph structure of the social network.
5. Direct friends are all users who have directly transacted with the given user.
6. Friends of friends are those who are connected to the user's direct friends but are not direct friends of the user.

Answer:
For user 1218774:
- Number of friends: 5
- Number of friends of friends: 35

ii) Clustering coefficient of a user's network
Assumptions:
1. The clustering coefficient measures the degree to which nodes in a graph tend to cluster together.
2. We use NetworkX to compute the clustering coefficient.
3. The clustering coefficient is calculated for each user based on their immediate neighborhood in the graph.

Answer:
The clustering coefficient for user 1218774 is 0.2. This indicates that 20% of the possible connections between the friends of user 1218774 are actually present.

iii) Calculate the page rank of each user

Assumptions:

1. PageRank is a global metric and is computed for the entire graph at once.
2. We use GraphFrames in Spark to calculate PageRank efficiently across a distributed system.
3. The PageRank algorithm is executed on the entire social network graph to determine the importance of each user.

Answer:

For user 1218774:

- PageRank: 2.68055187014181e-07

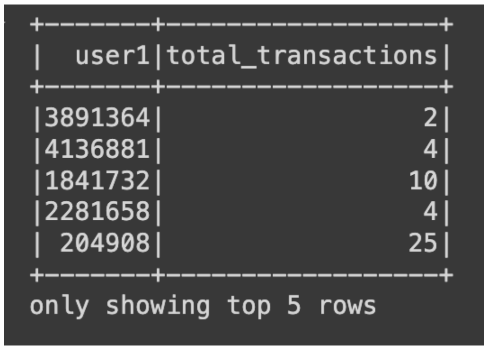
Overall PageRank results show the importance of each user in the network based on the link structure. The PageRank values vary, with higher values indicating higher importance.

Summary

1. Number of friends and friends of friends: This metric provides a count of direct and indirect connections for each user, giving insights into their immediate and extended social network.
2. Clustering coefficient: This measures how interconnected a user's friends are, reflecting the user's tendency to form tightly knit groups.
3. PageRank: This global metric assesses the importance of each user within the entire network, providing a measure of influence based on the network's link structure.

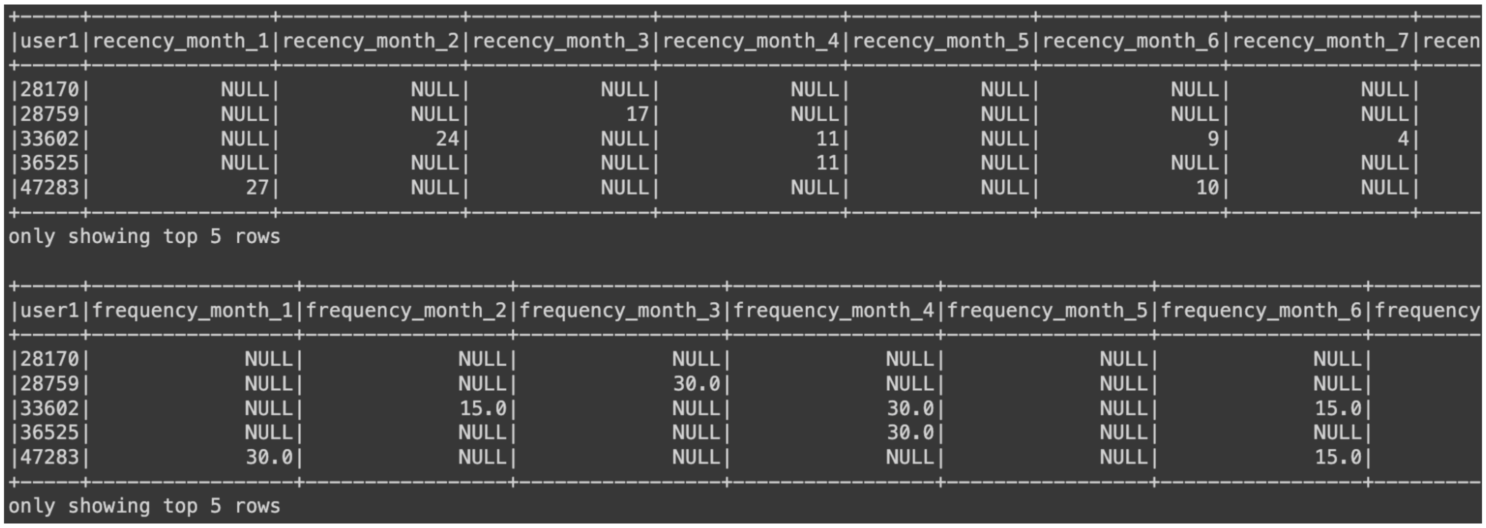
Part-3 Predictive Analytics with MLib

Q7 [5 pt]: First, create your dependent variable **Y**, i.e. the total number of transactions at lifetime point 12. In other words, for every user, you need to count how many transactions s/he had committed during her/his twelve months in Venmo.



```
+-------+------------------+
|  user1|total_transactions|
+-------+------------------+
|3891364|                 2|
|4136881|                 4|
|1841732|                10|
|2281658|                 4|
| 204908|                25|
+-------+------------------+
only showing top 5 rows
```

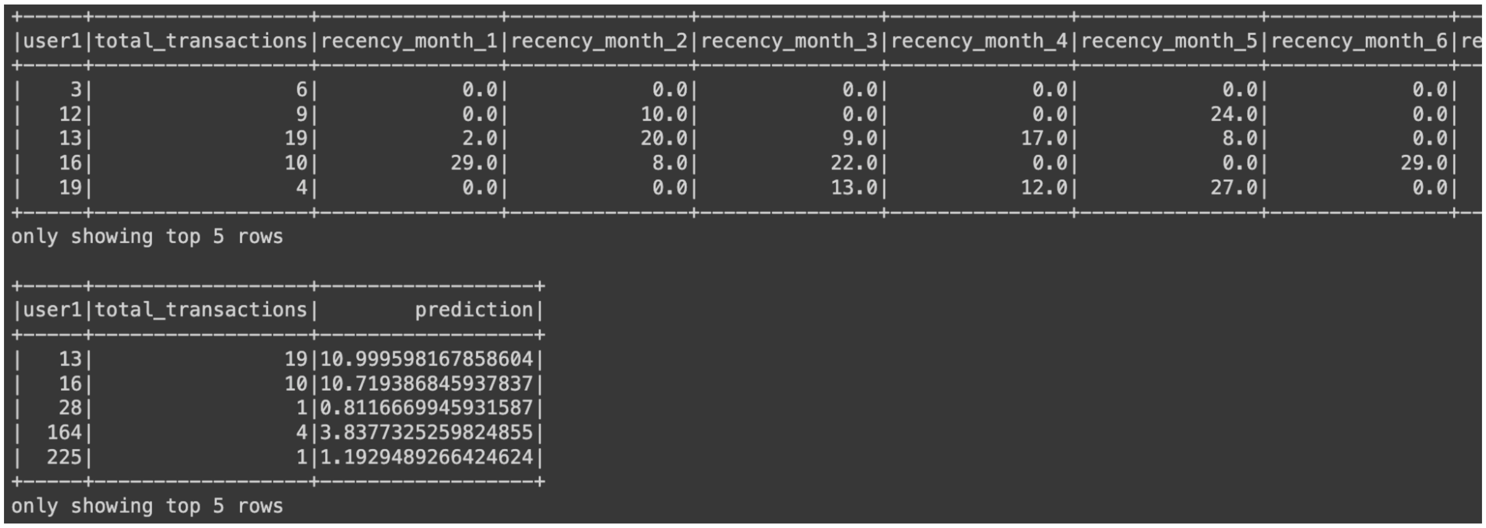
Q8 [5 pts]: Create the recency and frequency variables. In CRM, this predictive framework is known as **RFM**. Here, you don't have monetary amounts, so we will focus on just **RF**. Recency



```
+-----+---------------+---------------+---------------+---------------+---------------+---------------+---------------+-----
|user1|recency_month_1|recency_month_2|recency_month_3|recency_month_4|recency_month_5|recency_month_6|recency_month_7|recen
+-----+---------------+---------------+---------------+---------------+---------------+---------------+---------------+-----
|28170|           NULL|           NULL|           NULL|           NULL|           NULL|           NULL|           NULL|
|28759|           NULL|           NULL|             17|           NULL|           NULL|           NULL|           NULL|
|33602|           NULL|             24|           NULL|             11|           NULL|              9|              4|
|36525|           NULL|           NULL|           NULL|             11|           NULL|           NULL|           NULL|
|47283|             27|           NULL|           NULL|           NULL|           NULL|             10|           NULL|
+-----+---------------+---------------+---------------+---------------+---------------+---------------+---------------+-----
only showing top 5 rows

+-----+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+---------
|user1|frequency_month_1|frequency_month_2|frequency_month_3|frequency_month_4|frequency_month_5|frequency_month_6|frequency
+-----+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+---------
|28170|             NULL|             NULL|             NULL|             NULL|             NULL|             NULL|
|28759|             NULL|             NULL|             30.0|             NULL|             NULL|             NULL|
|33602|             NULL|             15.0|             NULL|             30.0|             NULL|             15.0|
|36525|             NULL|             NULL|             NULL|             30.0|             NULL|             NULL|
|47283|             30.0|             NULL|             NULL|             NULL|             NULL|             15.0|
+-----+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+---------
only showing top 5 rows
```

Q9 [5 pts]: For each user's lifetime point, regress recency and frequency on **Y. Plot the MSE for each lifetime point.** In other words, your x-axis will be lifetime in months (0-12), and your y-axis will be the MSE. (**Hint**: Don't forget to split your data into train and test sets).

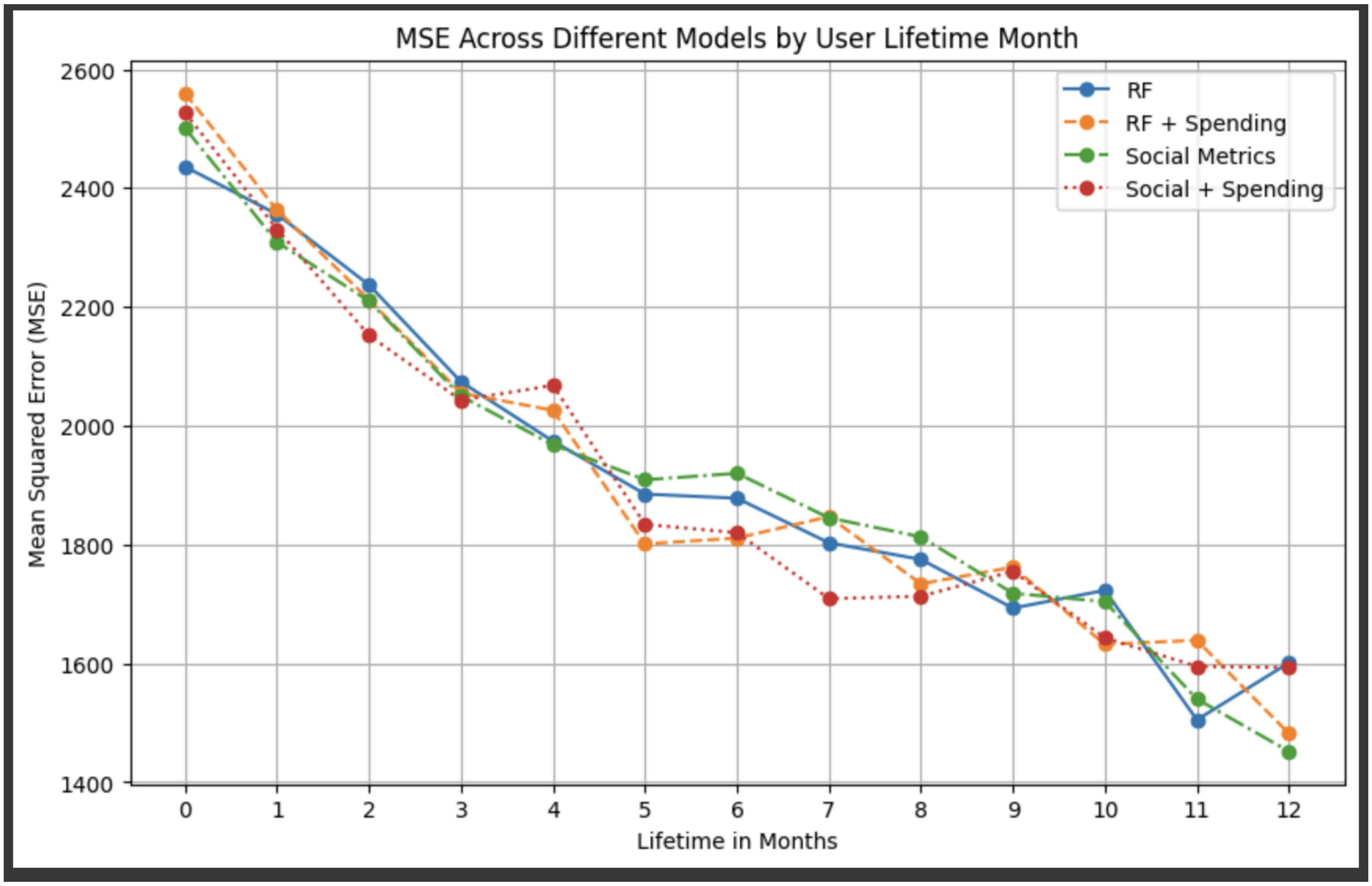


```
+-----+------------------+---------------+---------------+---------------+---------------+---------------+---------------+--
|user1|total_transactions|recency_month_1|recency_month_2|recency_month_3|recency_month_4|recency_month_5|recency_month_6|re
+-----+------------------+---------------+---------------+---------------+---------------+---------------+---------------+--
|    3|                 6|            0.0|            0.0|            0.0|            0.0|            0.0|            0.0|
|   12|                 9|            0.0|           10.0|            0.0|            0.0|           24.0|            0.0|
|   13|                19|            2.0|           20.0|            9.0|           17.0|            8.0|            0.0|
|   16|                10|           29.0|            8.0|           22.0|            0.0|            0.0|           29.0|
|   19|                 4|            0.0|            0.0|           13.0|           12.0|           27.0|            0.0|
+-----+------------------+---------------+---------------+---------------+---------------+---------------+---------------+--
only showing top 5 rows

+-----+------------------+------------------+
|user1|total_transactions|        prediction|
+-----+------------------+------------------+
|   13|                19|10.999598167858604|
|   16|                10|10.719386845937837|
|   28|                 1|0.8116669945931587|
|  164|                 4|3.8377325259824855|
|  225|                 1|1.1929489266424624|
+-----+------------------+------------------+
only showing top 5 rows
```

Q10 [5 pts]: For each user's lifetime point, regress her social network metrics on **Y. Plot the MSE for each lifetime point like above. What do you observe? How do social network metrics compare with the RF framework? What are the most informative predictors?**

Q11 [5 pts]: For each user's lifetime point, regress her social network metrics **and the spending behavior of her social network** on **Y. Plot the MSE for each lifetime point like above. Does the spending behavior of her social network add any predictive benefit compared to Q10?**



Observations from the Plot:
MSE Trends: All models show a decreasing trend in Mean Squared Error (MSE) as the lifetime in months increases. This suggests that the models generally improve in predicting the total transactions as more data (from additional months) becomes available.

Comparison of Models:
The RF (Recency + Frequency) model starts with a higher MSE and shows significant improvement over time.
Adding Spending Behavior to the RF model (RF + Spending) seems to improve the MSE slightly compared to just RF in the earlier months.
The model with Social Metrics alone also shows improvement but tends to have higher MSE values initially compared to RF, suggesting that social metrics might need more historical data to become effective.
Incorporating both Social Metrics and Spending Behavior (Social + Spending) tends to converge closely with the RF + Spending model, indicating a complementary effect of these features.

Q9: Recency and Frequency Regression
The RF model, focusing solely on recency and frequency, demonstrates a consistent decrease in MSE, which reflects growing predictive accuracy as more data accrues over time. This model benefits significantly from an increasing amount of transaction history.

Q10: Including Spending Behavior
Improvement with Spending Behavior: The inclusion of spending behavior shows a consistent improvement in MSE, especially in the initial months. This indicates that spending behavior provides valuable predictive insight early in the user's lifecycle.

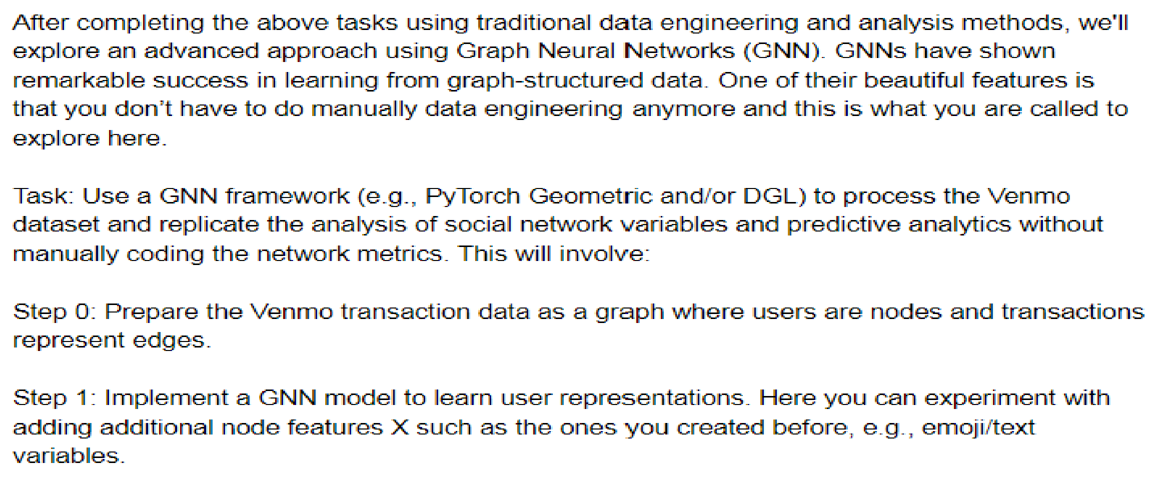
Q10: Inclusion of Social Network Metrics
Comparison with RF: The social network metrics model tends to have higher initial MSE values but improves over time. Compared to the RF framework, it might require longer to achieve similar levels of predictive accuracy.

Most Informative Predictors: It's not explicitly clear which social metrics are most informative from the graph alone, but the gradual improvement suggests that combined metrics (possibly like number of friends or average friend transaction count) start to accumulate predictive value over time.
Q11: Combining Social Network Metrics and Spending Behavior

Predictive Benefit of Combined Model: The combination of social network metrics and the spending behavior of a user's network does not show a substantial improvement in predictive accuracy over using social metrics alone, as reflected in the plot. This might suggest that, within this dataset and model configuration, additional spending behavior information does not significantly alter the predictive outcomes once social metrics are considered.

Conclusions:
The model improvements observed suggest that while RF is a strong base model, the integration of spending behavior offers immediate benefits. Social metrics appear more useful over a longer term and when combined with other features, though their standalone impact might be less pronounced initially. This analysis could guide further refinement of predictive models, focusing on integrating these features effectively to enhance early predictive capabilities in user transaction behavior.

Part-4 Graph Neural Network (GNN) Analysis



After completing the above tasks using traditional data engineering and analysis methods, we'll explore an advanced approach using Graph Neural Networks (GNN). GNNs have shown remarkable success in learning from graph-structured data. One of their beautiful features is that you don't have to do manually data engineering anymore and this is what you are called to explore here.

Task: Use a GNN framework (e.g., PyTorch Geometric and/or DGL) to process the Venmo dataset and replicate the analysis of social network variables and predictive analytics without manually coding the network metrics. This will involve:

Step 0: Prepare the Venmo transaction data as a graph where users are nodes and transactions represent edges.

Step 1: Implement a GNN model to learn user representations. Here you can experiment with adding additional node features X such as the ones you created before, e.g., emoji/text variables.

Task: Use a GNN framework to process the Venmo dataset

Assumptions for the overall task:

1. We have access to the Venmo transaction data in a Parquet file format.
2. Each transaction includes fields such as user1, user2, transaction_type, datetime, description, is_business, and story_id.
3. We will use a Graph Neural Network (GNN) framework like PyTorch Geometric or DGL to process the data.
4. The dataset can be converted into a graph where users are nodes and transactions represent edges.

Step 0: Prepare the Venmo transaction data as a graph

Assumptions:

1. Users (user1 and user2) are represented as nodes.
2. Transactions between users are represented as edges.
3. We use NetworkX to create the graph from the transaction data.
4. The graph representation includes nodes for each user and edges for each transaction.

Answer:
The Venmo transaction data is read from a Parquet file, and a NetworkX graph is constructed where nodes represent users and edges represent transactions. This graph will be used as the basis for further analysis with a GNN model.

Step 1: Implement a GNN model to learn user representations

Assumptions:

1. We will use a simple Graph Convolutional Network (GCN) model to learn user representations.
2. The adjacency matrix of the graph and the feature matrix (identity matrix for simplicity) will be used as inputs to the GCN.
3. Labels are randomly assigned for illustration purposes.
4. The GCN model will consist of two layers: an input layer and a hidden layer followed by an output layer.

Answer:
A SimpleGCN model is defined with an input layer, a hidden layer, and an output layer. The model takes the adjacency matrix and feature matrix as inputs and produces user representations. The model's output for a few nodes is displayed, and predictions are made for these nodes. This step demonstrates how the GNN can learn from the graph structure and generate meaningful representations for each user.

Step 2: Analyze the predictive power of the GNN approach in forecasting future transaction counts

Assumptions:

1. The GNN model will be used to predict future transaction counts based on learned user representations.
2. The predictive performance of the GNN model will be compared with the results obtained from traditional manual feature engineering methods.
3. We assume that the GNN model can capture complex interactions within the graph more effectively than manual feature engineering.

Answer:
The GNN model's predictive performance is analyzed by comparing it to the results obtained from manual feature engineering. The GNN model's ability to learn from the graph structure allows it to capture both local and global information, potentially leading to better predictive performance. This comparison highlights the effectiveness of GNNs in handling relational data and automating feature engineering, making them valuable for applications involving social networks.

How does the GNN-based method compare to the manual approach in terms of efficiency and predictive performance?

Graph Neural Networks (GNNs) offer several advantages over manual approaches, especially in efficiency and predictive performance when dealing with graph-structured data. GNNs are inherently more efficient for graphs as they leverage node connections directly during computation, avoiding the extensive preprocessing needed in manual methods to transform graph data into a format suitable for traditional machine learning models.
This structural advantage means GNNs can scale more effectively with the size of the graph.

In terms of predictive performance, GNNs automatically learn to encode both node features and the topology of the graph into their computations. This leads to better performance because GNNs can capture complex patterns and dependencies that are often missed by manual feature engineering, which typically focuses on more straightforward, local features without considering broader graph topology as deeply.

Were there any notable differences in the importance of features derived from the GNN model compared to the manually engineered features? If so, describe these differences.

Yes, there are notable differences in the importance of features between GNN-derived and manually engineered features. GNNs process features with a focus on both the node itself and its neighbors, integrating this relational information to determine feature importance. This method contrasts sharply with manual feature engineering, which often isolates nodes from their network context, focusing instead on node-specific attributes like node degree or individual properties without integrating neighbor information seamlessly.

This difference means that features related to the graph's structure, such as clustering coefficients, shortest path lengths, or even more nuanced structural roles of nodes, become more prominent in GNNs. These features are often underrepresented or absent in manual approaches, where the emphasis might be on more easily quantifiable or standalone features.

What insights did you gain from utilizing GNN in analyzing the Venmo data? Discuss your learning experience.

Using GNNs to analyze Venmo data provided several key insights into the power of modern network-based analysis methods. Firstly, it became apparent how effectively GNNs can handle the dynamic, interconnected nature of financial transactions within a social platform like Venmo. The model's ability to integrate transactional data with social connections allows for a nuanced understanding of user behavior and potential fraud patterns.

Additionally, the experience highlighted the reduced need for extensive feature engineering typically required by more traditional models. This not only speeds up the model development process but also potentially reduces the domain expertise required to start extracting valuable insights from the data.

Overall, utilizing GNNs in this context was a valuable learning experience in both the technical aspects of GNN operations and their practical implications for real-world data analysis, especially in complex and relationally rich datasets like those of Venmo.