CS 6350

ASSIGNMENT 3

Names of students in your group:

Thien Nguyen

DXN210021

Raaid Rizwan

RXR200043

Number of free late days used: 1   
Note: You are allowed a **total** of 4 free late days for the **entire semester**. You can use at most 2 for each assignment. After that, there will be a penalty of 10% for each late day.

Please list clearly all the sources/references that you have used in this assignment.

Problem 1: Attached zip file.

Problem 2: <https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/2253767106815136/2562041412224435/8178819525474631/latest.html>

Problem 2 Dataset: <https://snap.stanford.edu/data/soc-Epinions1.html>

Report and summary on following pages.

**Assignment 3 – Problem 1: Report**

Where the Data Comes From

We retrieve financial news articles through the Finnhub API (using the finnhub-python library). A script named finnhub\_producer.py pulls general-news items, checks that each article hasn’t been processed before, and then publishes the raw news data—headline, summary, source, and so on—to a Kafka topic called news\_raw.

Interpreting the Results

The objective is to discover which named entities (companies, people, places) appear most often in the news. Spark extracts the entities and Kibana visualizes their counts.

In the Kibana “Top 10 Entities” chart (captured in Top 10 Entities Result.png) you can see the entities that occurred most frequently during the period analyzed. Company names such as AbbVie appear, showing they were featured in recent coverage. General concepts like AI are present as well, reflecting their popularity in the financial sector. An “Other” bucket groups together less common entities, which is typical in this kind of aggregation. A few items that are not true entities—percentages or date fragments such as “A decade” or “5 %”—still slip through, suggesting that the Spark-side filtering could be tightened so it counts only organizations, people, and locations (the ORG, PERSON, and GPE labels).

Potential Improvements

• Stricter filtering: Modify spark\_app.py to include only the desired entity types (ORG, PERSON, GPE) and ignore MONEY, PERCENT, DATE, and similar labels.

• Kibana adjustments: Refine the visualization—for example, change the “Top N” setting or apply additional filters—to emphasize the most relevant entities.

Overall, the dashboard offers a quick view of trending topics and entities in financial news; an extra round of filtering would make the results even cleaner.

**Assignment 3 – Problem 2: Explanation**

We used the SNAP soc-Epinions1 dataset, which collates a sample of “X trusts Y” directed relationships from the Epinions.com customer review site. We created a GraphFrame from the dataset (the specifics of which are explained in the code) and showed the top five results from every query in the homework. The results we attained make sense. We will only be describing the top one or two in the following result analysis. In order,

1. The vertices with the highest number of outDegrees would be the users who trust the most other users. In our results, user 645 trusted the most other users with 1,805 outward edges.
2. The vertices with the highest number of inDegrees would be the users who have the most users who trust in them, i.e. the most trusted users. In our results, user 18 was the most trusted user with 3,035 inward edges.
3. The vertices with the highest PageRank are the vertices that have the highest number of inward edges with the lowest number of outward edges. In other words, the most trusted users that don’t also trust many other users (thus, they aren’t trusted simply because they know many other users). This is user 18 once again with a PageRank of 345.
4. Connected components in a graph are a collection of vertices which are connected through edges. We only have two connected components in the graph according to our results, one of which has a count of 75877 vertices while the other has 2. This means that almost all users are connected to each other except for two who are only connected to each other.
5. TriangleCount lists the number of “triangles” (3-vertex relationships) a vertex is involved in. Our first result is user 645 (the top outDegree vertex) 48,674 triangles. In second place is user 18 (the top inDegree and PageRank user) with 47,203 triangles.