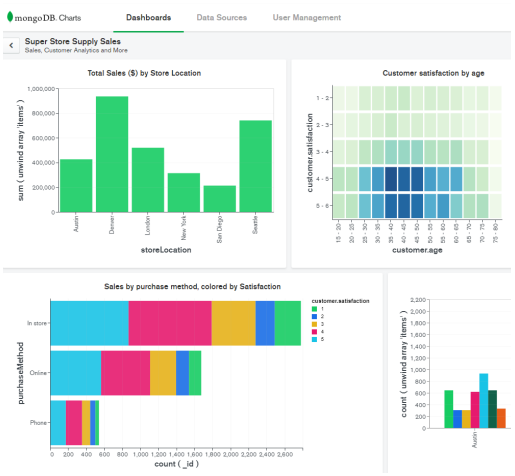
COURSE PROJECT — Rumour Spreading Analysis on twitter

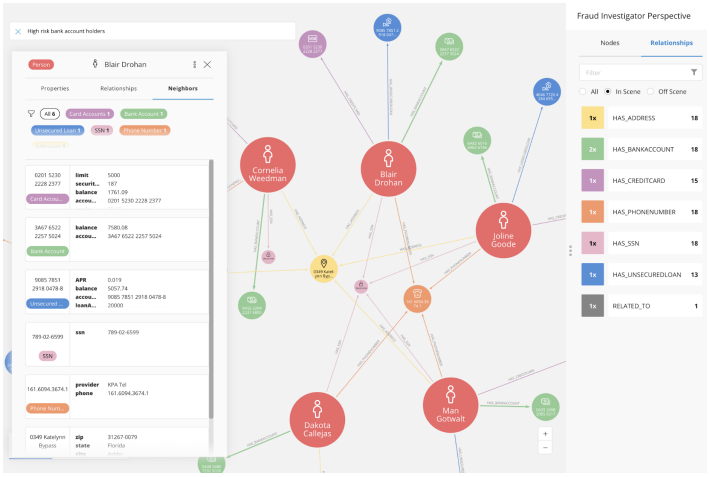
Big data on social media is a valuable resource only if we know how to analyze it effectively and with the right tools. In this project, you will learn how social media data (on Twitter) can be analyzed with guided steps and select data-intensive tools w.r.t the nature of the given problem. The provided data is a set of rumour tweets about eight different topics (breaking news) as follows: charliehebdo, ebola-essien, ferguson, germanwings-crash, ottawashooting, prince-toronto, putinmissing, sydneysiege (click here for more details). You will apply various data-intensive concepts/tools to parse, transform, analyze, and visualize how rumours spread on social media by comparing the data of rumours and non-rumours on Twitter. The tools include but are not limited to: MongoDB server/Atlas, MongoDB Charts, MongoDB Database Tools, Tableau or PowerBI, and Neo4J Graph Database. Each rumour dataset contains networks of tweets, including: original `source tweets` (and their labelled `annotations`: `misinformation:1` (rumour) or `misinformation:0` (real news)), `reactions` (and their replying hierarchical `structure`: who replied to whom), and `retweets` by who (no hierarchical structure for retweets). The goal of this project is to (1) help you get familiar with the big data tools and (2) also expose you with other potential tools both on-premise and cloud services when analyzing non-relational data. (Click here for tweet data dictionary info.) The following is a guideline to complete the project:

STEPS:

1) Pick ONE of the 8 topics (breaking news) above in the rumour tweet datasets and write an ingesting script (in any programming language) to import source-tweets and retweets into MongoDB using MongoDB Database Tools (`mongoimport`) as one collection for each source: tweets and retweets, respectively.
2) Update each of the source-tweets with labelled annotations specified in the datasets using CRUD commands.
3) Perform the **1st analysis** on *the key words and their frequencies* using Word Cloud to see what words and their frequencies there are in the rumour and non-rumour data. The Word Cloud can be done using methods mentioned in the word cloud project assignment in the Big Data Analysis Coursebook, MongoDB Charts (click here for demo), or Tableau, etc.
4) Perform the **2nd analysis** on *how many times each tweet (both rumour and non-rumour) gets retweeted on social media* by using MapReduce or Aggregation functionalities available on MongoDB.

5) Perform the **3rd analysis** on *how fast each tweet (both rumour and non-rumour) gets retweeted on social media* by counting the accumulated number of tweets over time using MongoDB Charts, Tableau, Power BI or other visualization tools. The running total (accumulative) line graphs should also include the overall cumulative frequencies such as the AVG, MIN, and MAX frequencies. Hint: Attribute `"retweeted_status.user"` stores info. of the original twitter user while attribute `"user"` stores retweeting user info.
6) Perform the **4th analysis** on *who tweeted the tweets, each of them was retweeted by whom* by converting json objects into nodes and edges, and visualizing them on a graph database Neo4J. Hint: Step-by-step video on how to load json to neo4j, and tutorial.
7) **(EXTRA POINTS, optional)** Perform the **5th analysis** on *where each tweet is tweeted by visualizing users' locations on the map* using aforementioned tools or other visualization tools such as Google Maps API.
8) **(EXTRA POINTS, optional)** Write a converting script to convert a `structure` file into the corresponding tweet and its hierarchical replies json file, and import it into MongoDB as a collection called `'reactions.'`
9) **(EXTRA POINTS, optional)** Perform the **6th analysis** on *who tweeted the tweets, each of them was replied by whom in hierarchy* by visualizing it on a graph database Neo4J.





What & How to Submit:

- Submit: 1) all the source codes for database scripts (including CRUD commands and mapreduce functions), ingesting scripts, and/or converting scripts (with comments),
  2) databases both MongoDB and Neo4J (either on-premise or a link to cloud version),
  3) resulting outputs of **the analyses** (screen captures, databases, and/or files), and associated dashboard from MongoDB Charts, Tableau, Power BI, or Google Maps, and
  4) powerpoint slides describing the project steps, results, and demo; via Google Classroom.
- The project can be done in a group of three to four students. The project demo will be conducted on the next day of the project due.