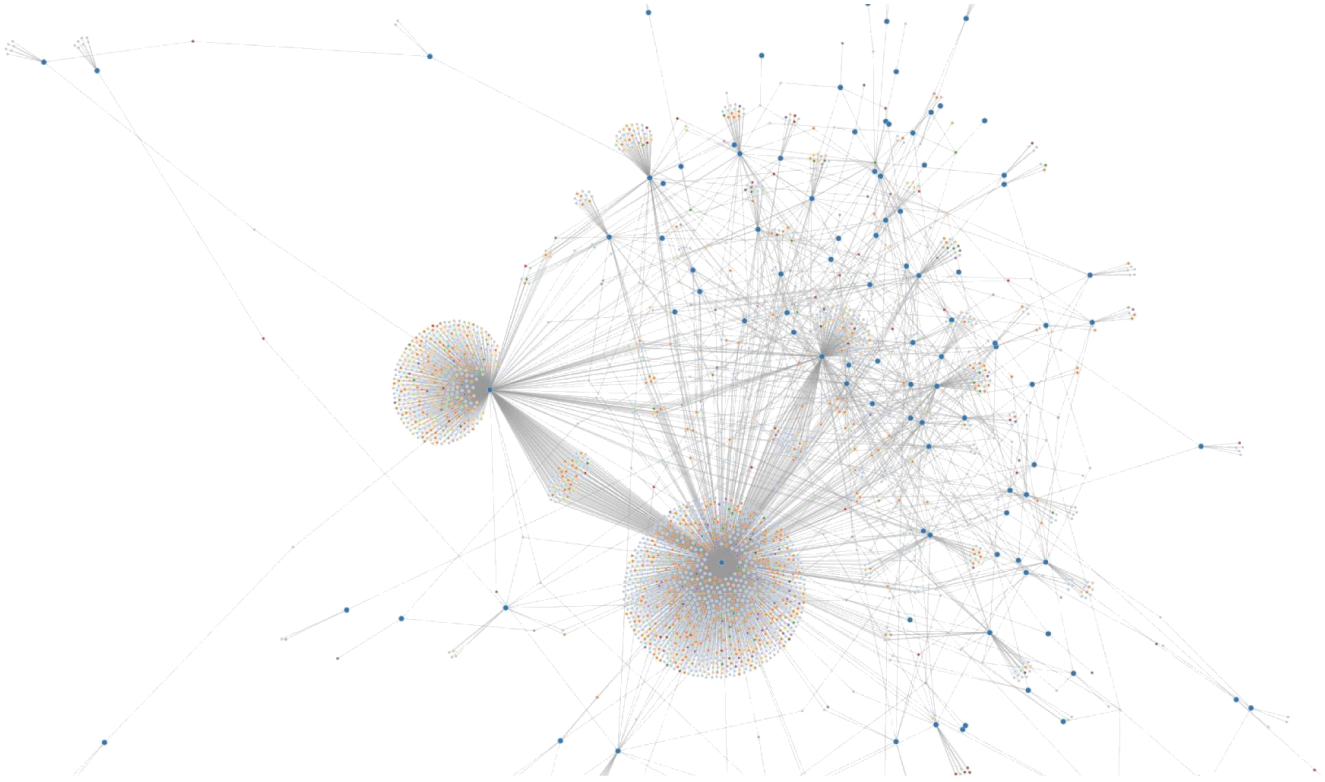


Yelp User Pattern Analysis and Recommender System Report

Department of Computing Science
Simon Fraser University

Ruoting Liang, Tianpei Shen, Yuyi Zhou

Table of Contents

Problem Definition

❖ Project Scope

Yelp is a web platform that collects and hosts data based on user reviews on services provided by local businesses. As the term project is to utilize big data computational tools to conduct data analysis, our team members initiated a project to analyze yelp user pattern and develop a recommender system to assist yelp users in finding services. The system is built based on the yelp data challenge data set [1] which is publicly available on Yelp's website. In the dataset, there are a total of 6 json files: business.json (132.3MB), checkin.json (60.1MB), photo.json (24.2MB), review.json (3.82GB), tip.json (184.9MB), and user.json (1.57GB). After completing the initial analysis of the dataset, the project team decided to focus the analysis on user behavior patterns. Particularly, the system's main focus is to analyze users who provided ratings for businesses in the Great Toronto area.

This project comprises of two major components: pre-processing data on spark to analyze user patterns, and building a recommender system for users. For the first part of this project, the following aspects were analyzed: user-business relationship, user rated business categories and rating distributions, number of compliments user received, number of votes users sent out, and keywords used in users' positive reviews vs negative reviews. Based on the results from the first part of this project, a recommender system was built to recommend new businesses to any single user based on his or her personal preferences.

❖ Potential Challenges

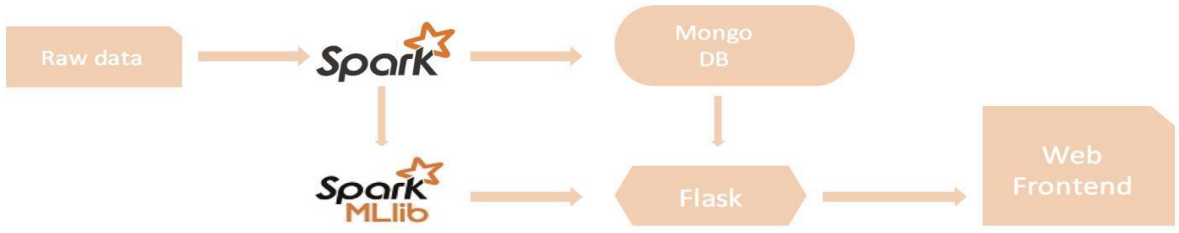
The project team encountered numerous challenges in this project. In particular, optimizing processing big files such as review.json (3.82GB) and user.json (1.57GB) and presenting informative visualization summarizing large amounts of data were some of the main challenges experienced.

- ➢ Challenges in data processing:
  - ○ Identifying all businesses in the Great Toronto area with user ratings;
  - ○ Re-assigning a new category for each business, whereas the original business categories list contains multiple categories;
  - ○ Analyzing user reviews: extracting key words from each user's review;
  - ○ Processing data to fit visualization;
  - ○ Converting string ids to int ids where int ids are used to train recommender system models; and,
  - ○ Training the ML model to predict new ratings for users.
- ➢ Challenges in visualizing data:
  - ○ Hard to create informative visualization to summarize user-business relationships that contains 15k businesses, 80k users, and 400k links between businesses and users; and,
  - ○ Identify the most suitable tools, based on several new tools, to create a decent visualization.

Methodology

In order to create informative displays while allowing users to interact with the data, the project team decided to build a web application user interface that places an emphasis on the user feedback. The initial layer of the web service was built to visualize the yelp dataset based on business categories with the functionality for the end user to easily retrieve and filter data from various categories.

Our application pipeline:



❖ Application of Tools and Techniques

➢ Data Processing: Spark, Spark SQL, Spark MLlib

Spark is very powerful in parallel processing large data. Especially, it supports DataFrame and SQL. With using Spark, the dataset gets computed efficiently.

The web service, which is compatible with spark, invokes the ML model to make predictions. This provides a straightforward and interactive way to search for recommended businesses.

➢ Web backend: flask, spark, cherrypy

Flask is a lightweight web service. There are two primary reasons flask was chosen: it is easy to use and learn; it is python based which allows flask to integrate easily with pyspark. After reviewing the available tutorials[2], the project team successfully deployed cherrypy as WSGI web server to get pyspark to communicate with flask.

➢ Web frontend: D3.js, DC.js, crossfilter.js, Leaflet.js, keen.js, bootstrap v4

D3.js, DC.js, and crossfilter.js are famous JavaScript libraries used to build data visualization dashboard. They provide interactive charts and their user interface is "clean". The Leaflet.js was the chosen JavaScript library used for mapping. The team chose Leaflet over Google Map because leaflet responds quicker, more user friendly, and is compatible with numerous plugins. Moreover, Google API can be down occasionally based on previous experience. The kean.js and bootstrap v4 were used to style the web user interface.

➢ Data Storage: MongoDB

The MongoDB is a distributed non-relational database. It is well scaled, which is perfect for big data storage. Data is stored in MongoDB as documents. As a result, it is not required to define schema before creating data. For this project, after data is processed using spark, there are several json output files that are directly imported into MongoDB. Querying data in MongoDB with pymongo can also be completed quickly without additional efforts.

➢ Other tools/technologies:

Gephi for user-business relationship network visualization: the full dataset for graphing is roughly 35MB, which web browsers cannot handle. As a result, Gephi, a network analysis and visualization tool, was used to draw the complete relationship diagram between users and businesses.

Yelp GraphQL for retrieving additional information: yelp provided GraphQL to allow the recommender system to query additional data such as external URLs and user uploaded pictures.

Problems

❖ Technical Challenges Encountered

➢ Re-assign new category to each business

Original business category list contains multiple categories, and many categories overlap with each other. For example, categories that contain 'food' are 'food stands', 'Food trucks', 'Live/Raw Food', 'Soul Food', 'Seafood Markets', etc. In order to find most common categories to give a category representation for each business, the top 20 most common categories was identified and each business was re-assigned with a new category.

➢ Review text visualization

As it is difficult to visualize text in reviews for any given user, the team decided to build a word cloud to give a visual representation of text data. The more important and frequent a word is used, the larger and bolder it is displayed.

Additionally, in order to distinguish between what words a user used in a positive review versus a negative review, the text in reviews was first cleaned by tokenization segments into its atomic element while removing stop words in English. For each user, two documents were built based on the rating score, where positive review are user ratings equal to 4 or 5 while negative reviews are user ratings equal to 1 or 2.

Subsequently, term frequency-inverse document frequency (TF-IDF)[3] was used to measure how important a word is to the document.

A highly activated user may have up to 10,000 words in his review corpus. As a result, it is difficult to visualize such word document without generating too much clutter. In order to visualize positive/negative review document for each user using the word cloud, each word cloud was limited to the top 100 largest TF-IDF value words.

➢ Building a recommendation system

To build a recommendation system for users, we decided to use Spark machine learning recommendation library, Alternating Least Square algorithm (ALS)[4], to train the model. Initially, we need to convert the given string ids into int ids and use them to train the recommender system. The whole dataset was split into training data and testing data to initially train the model using the training dataset. The best model was then selected based on the best performance and was then applied against the testing data. The pre-train model is loaded and new predictions are made based on each web request.

➢ Analyzing user-business relationship

In analyzing user-business relationship, we have 15k businesses and 80k users. As the full data set is 35MB, common web browsers are incapable of loading such big files. Even though FireFox can eventually display the graphs, it takes over 15 minutes and the browser will prompt the user whether to continue loading the page that has become non-responsive for long durations. Additionally, a graph containing 90k nodes and 400k links is complicated and difficult to visualize each relationship clearly. Therefore, the top 100 users with the most number of reviews and the corresponding businesses were selected to be shown (Even though the front page only show a subset of the users, other user analysis pages contain all users' information). Using this methodology, the web service can display clearly which user rated the most number of businesses as well as which businesses have the most user ratings. Graph of the full dataset (i.e. all Great Toronto business and users rated these businesses) was also drawn using Gephi to give the end users a sense of the big business network within the Greater Toronto area.

Results

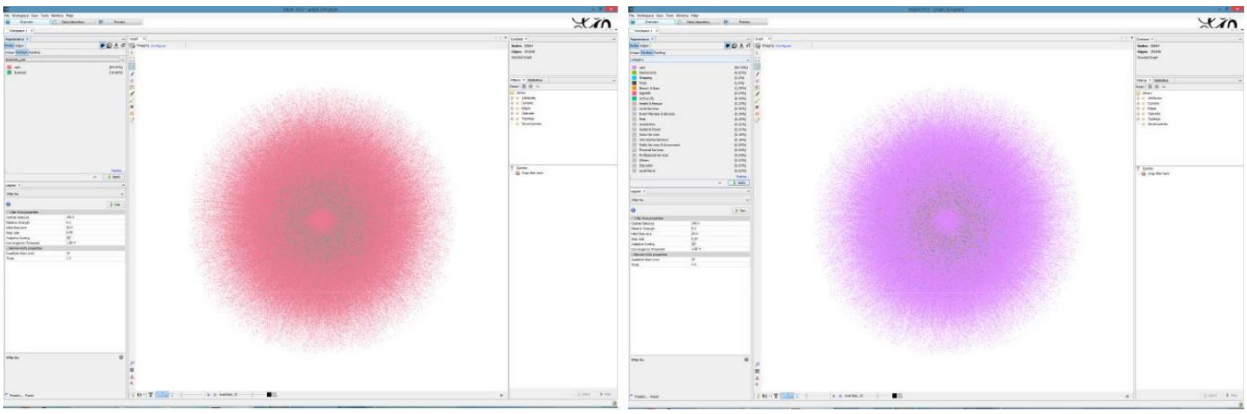
❖ Outcomes of the project

The following URL contains a short video showing a quick overview of the web service interface as well as the recommender system: https://www.youtube.com/watch?v=hoNKMHuzXDM

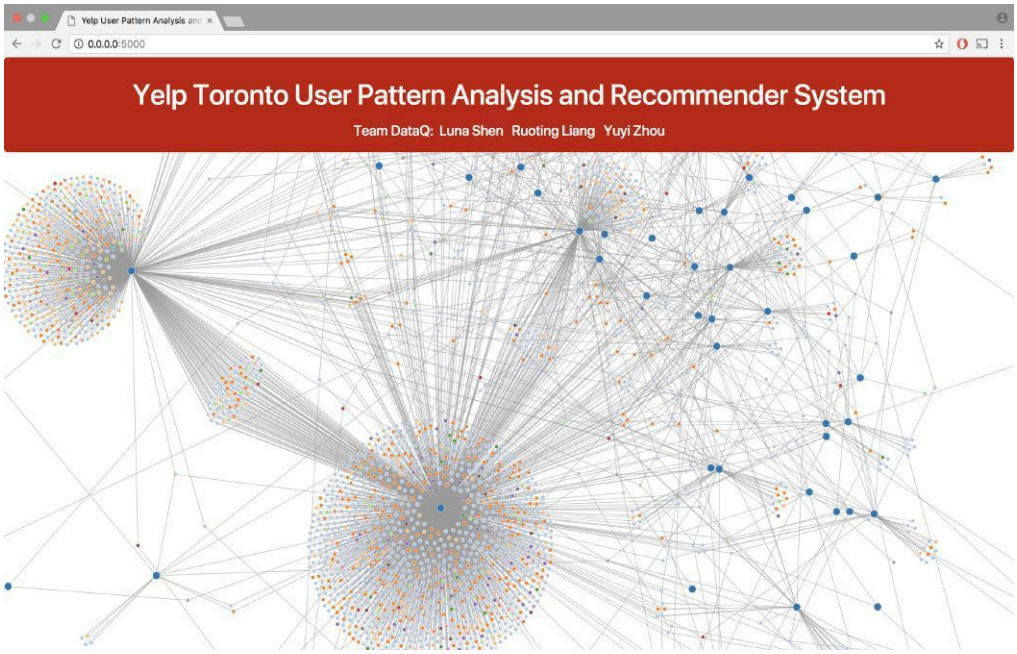
➢ Link graph

User-business link (without splitting businesses into different categories, left diagram): Each user is a pink dot and each business is a green dot. The total number of nodes is approximately 100k with approximately 400k links bridging connection between the nodes. As seen from the left diagram, there are more users (84%) than business (16%), with each link representing a user review for a particular business.

User-business link (splitting the businesses to different categories, right diagram): With the businesses assigned to 20 major categories, the most popular business is restaurant which is shown as green dots.



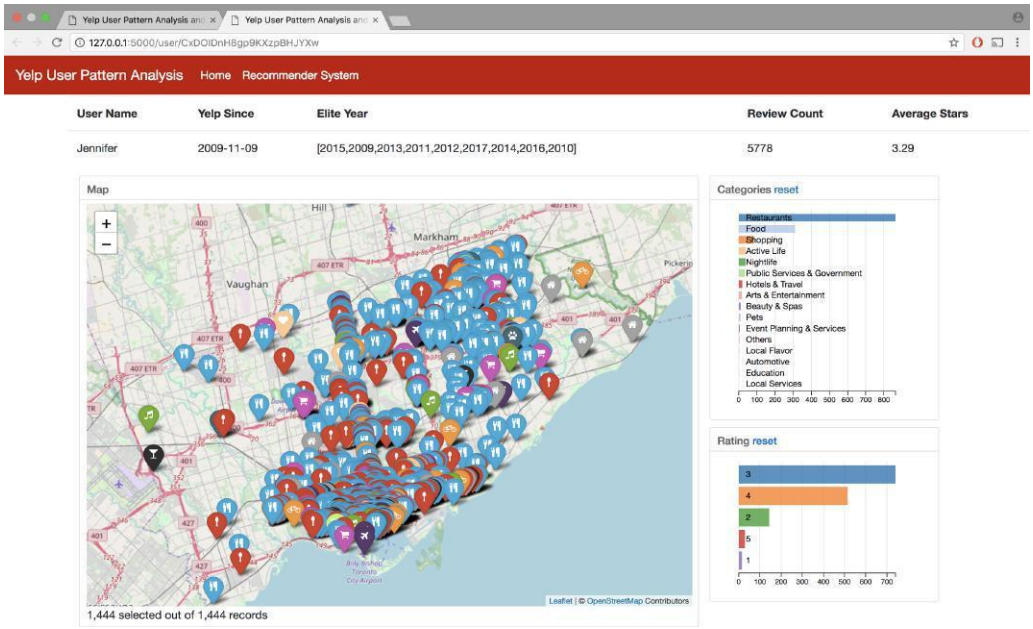
The main page of the website only shows the top 100 users and the businesses they have rated. The blue nodes represent the user and other nodes represent different categories of businesses.



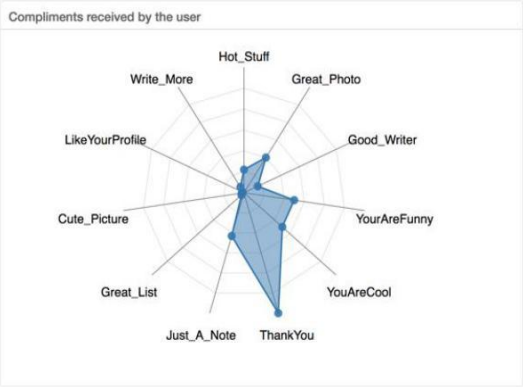
➢ Map

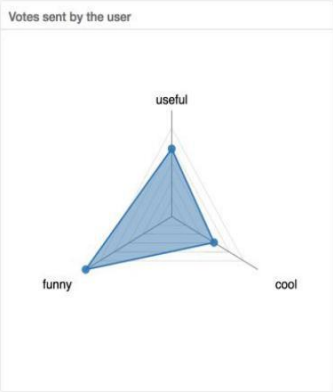
This part shows three interactive charts: map, business categories rating distribution, and rating count distribution. The map shows all the business locations a selected user has submitted a user rating. When the mouse hovers over a particular business, it shows the timestamp and rating of the business review.

When selecting one or more categories in the left business category chart, the filter functionality is enabled and the map will show only the businesses under the selected categories. The rating chart shows the rating count distribution for the selected categories. Similarly, selecting a particular rating will update map and category chart to only contain businesses with the selected ratings.

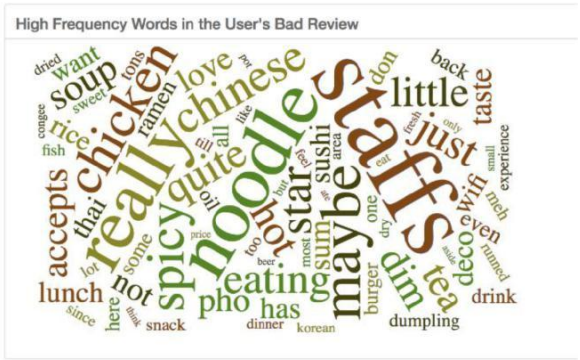


➢ Radar charts of compliments received by the user



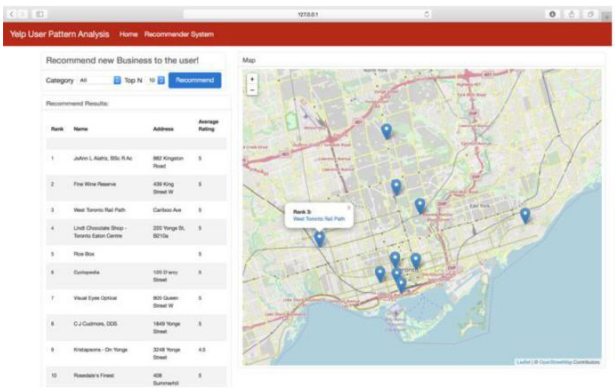


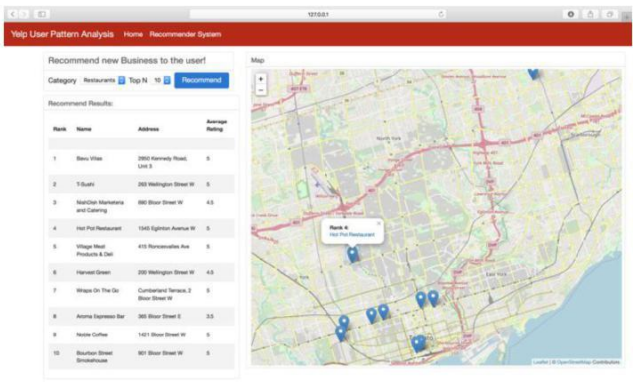
➢ Word cloud of good/bad reviews by the selected user



➢ Recommendation system

Two models were trained, with one model strictly used for recommending restaurants while the other model is used for all other business categories. The query is first sent to web backend where the web backend constructs a test set and sends it to the spark model. The spark model predicts a rating on each user-business pair, sorts the businesses by rating score, and returns top N businesses to web backend. The web frontend populates the businesses information in a table based on their ranking. Additionally, the business locations are shown on the map. When selecting the marker in the map, it provides an external Yelp URL retrieved by Yelp's GraphQL. Each URL links to a particular Yelp webpage containing information on the selected business.





❖ Lessons Learned - Data Analysis

In this project, the team conducted data analysis on a sub dataset within the Great Toronto Area of *391,848* reviews of *15,495* businesses, written by *83,383* different users.

This dataset was analyzed to identify user behavior patterns within Toronto. Several interesting discoveries were found in this project:

- ➢ By analyzing the number of businesses reviews, it was discovered that some users are more active than others in providing reviews for numerous businesses;
- ➢ Preliminary results can be derived based on analyzing the text written in each review. For example, if there was a user that always mentioned 'staffs' in a review, whether it is either a good or bad review, then we may conclude that the user strongly values the services received from the business. In other words, if a business wants to increase its review rating received from this particular user, the business must provide a better service.
- ➢ A user's personality may be understood by analyzing the different types of compliments the user received. For example, if a user receives more 'compliment_fun', we may conclude that the user is a very fun person. We can also get to know what kind of business the user cares more about and whether he or she is a strict person based on the number of each rating level.

Therefore, data analysis is very important to acquire meaningful insights from the dataset. Moreover, we can apply the knowledge that we gain from dataset to make knowledgeable business decisions and to establish strategic goals to improve or expand business opportunities.

In addition, data visualization is important in data analysis. As the main goal in big data analysis is to simplify large amounts of data with complex connections for people to understand and identify useful patterns, this cannot be achieved without decent visualization tools. As a result, the project team has invested a lot of effort in this project in the visualization component. The team has converted the raw data collected from user reviews into simplified graphs and visuals that is easy to understand while not losing emphasis on key information from each user.

❖ Lessons Learned - Implementation

Before implementing the web application, it is essential to make a strategic plan with a roadmap outlining key deliverables and performance goals. By identifying all of the key features to be delivered, each team member can pre-process the data such that no time is wasted on re-doing any work. In this project, each member was drawing different graphs to visualize a user pattern/behavior, while doing pre-processing data in spark. Each member kept the end goal in mind and constructed the dataset such that is can be implemented in the next phase of the project.

The project team has gained an understanding on handling big data through completing the project using Spark to process data. Additionally, the team has explored and learned new technologies including MongoDB for storing data, web backend collaborating with spark, and available JavaScript libraries in creating visuals for large dataset.

In addition to establishing a strategic plan, other data analysis tools were examined to assess if there are tools available to use for the purpose of cleaning, processing and visualizing the data. While we were drawing the relationship between businesses and users, we want to plot the full graph of Toronto businesses and users through flask. However, the file is too large for browser to handle. Subsequently, the team tried to reduce the file size by replacing the string id to integer, which still did not work due to the large size of the file. We tried several other approaches and finally decided to use Gephi, an open-source network analysis and visualization software package written in Java. Gephi is capable to show the full graph with about 90k nodes and 400k links.

Project Summary

Getting the data	0
ETL:Extract-Transform-Load work and cleaning the data set	0
Problem: Work on defining problem itself and motivation for the analysis.	4
Algorithmic work: Work on the algorithms needed to work with the data, including integrating data mining and machine learning techniques	3
Bigness/parallelization	2
UI	5
Visualization	4
Technologies	2

References

[1] *Yelp*. From https://www.yelp.com/dataset

[2] Dianes, J. A. (n.d.). *Building a Movie Recommendation Service with Apache Spark & Flask Part 2*. From www.codementor.io/jadianes/building-a-web-service-with-apache-spark-flask-example-app-part2-du1083854.

[3] Kwok., H. W. (2008). *Interpreting TF-IDF term weights as making relevance decisions.* ACM Transactions on Information Systems.

[4] Yehuda Koren, R. B. (2009-08-01). Matrix Factorization Techniques for Recommender Systems. CA, USA: IEEE Computer Society Press Los Alamitos.