Big Data Analytics
Project proposal
2020 - 2021

Issam Falih

As part of the course requirement, students are required to complete a class project. The class project accounts for 30% of your final grade. Each project can be done individually or in groups of 2 students.
The project should contains at least the following parts:

- Data collection and preprocessing & data transformation (i.e. dimensional reductions techniques).
- Data analysis & Visualisation.
- Make the link between this problem and the need to use the big data techniques.
- Apply a machine learning model on the problem i.e. clustering, classifications, ...
- Project reports & presentation(see the details about what report to submit below).

The timeline and deliverables for the project are as follows. All deliverables are due before midnight on the due date:

- **December 13, 2020:** Email your team members and project topic you wish to pursue .
- **December 20, 2020:** Upload on Moodle a 2-page project proposal. The proposal must contain the following information:
  - Project title along with a list of team members
  - List of data sources that will be used (provide a URL for the data). State whether the data needs to be further preprocessed.
  - A short abstract describing the goals of the project.
  - Project timeline.
  - Role of team members (who will be responsible for doing what).
- **January 15 , 2020:** Upload on Moodle a Project Presentation.
- **January 17, 2020:** Upload on Moodle your final project report. Make sure you submit a zip file that includes the source code as well (you don't need to submit the raw data). If you're hosting the project on a server, make sure you make the web site accessible until January 30, 2021.

You may choose one of the topics below for the project. If you have other suggestions, you are welcome to discuss it.

1 Sport Analytics

There are many publicly available databases for professional sports players and teams. Examples are:

1. https://openfootball.github.io
2. NFL (http://www.databasefootball.com/)
3. NBA (http://www.databasebasketball.com)
4. MLB (http://www.databasebaseball.com/)
5. NHL (http://www.databasehockey.com)

You can use the database to perform various types of analysis. For this project, you will need to do extensive preprocessing to convert the raw data into their appropriate formats. The following is an example of a prediction task you can perform using the sports database:

Given a pair of teams, (X,Y), where X is the home team and Y is the away team, predict who will win the game or what is the point difference at the end of the game. You will need to extract features for the home team and the away team (e.g., their offensive statistics over the last, say, 10 games, the defensive statistics over the last 10 games, statistics about their players, etc). You will need to create a data set containing examples of games where X had beaten Y, Y had beaten X, or X drew with Y. Train a classifier to make the prediction for future games. Report the accuracy of your classifier. For visualization, you can show how your predictions changes week by week for various games.

Instead of predicting the game outcome, you can also try predict teams that will make it to the playoffs, who are the best players, the final rankings of all teams, etc.

2 Sentiment Detection on Twitter Data

The goal of this project is to identify trends in user sentiment on a specific topic and to monitor their changes over time. You need to use the Twitter streaming API to monitor a set of keywords that capture all the tweets about the given topic. The data collection should take at least 6-8 weeks to make sure you have enough event data to do your analysis. You should retrieve only tweets that have geolocations and remove all the retweets. The tweets should be preprocessed.

You also need to develop a sentiment analysis method that will predict whether a tweet has positive, negative, or neutral sentiment on the topic. To do this, you must first manually label some of the tweets as positive, negative, or neutral classes, and then train a classifier to predict the sentiment of the rest of the tweets. To improve the classifier, you may add other features for classification, such as the number of positive or negative words in the tweet. A list of words with positive and negative polarity is available at `https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html`.

Use the geolocation to group together tweets from the same "region" (city, county or state, depending on whichever granularity you prefer). Aggregate the tweets for each region per day and

create a time series that shows the net daily sentiment (#positive tweets - #negative tweets) in the region. If you want, you may also cluster the regions based on similarity of their time series.

3 Recommender system Website

here are many user ratings data sets available online. This includes:

a. MovieLens movie ratings (https://grouplens.org/datasets/movielens/)

b. lastFM million song dataset (http://labrosa.ee.columbia.edu/millionsong/lastfm)

c. other data sets (https://gist.github.com/entaroadun/1653794)

In addition to the ratings data, you should try to incorporate other information as well to improve the recommendation (e.g., in movie recommendation, add information about movie genre, actors, directors, etc). Below, is a summary of the tasks you need to perform to complete the project:

1. Apply user-based and item-based recommendation algorithm as follows.

   For each user, calculate its top-k nearest neighbors (i.e., other users who share the most similar item preferences). Use the weighted average ratings of the neighbors to estimate whether the user likes an item he/she has not rated.

   For each item, calculate its top-k nearest neighbors (i.e., other items whose ratings are most correlated to it). Use the weighted average ratings of the user on the most similar items to estimate whether the user likes an item he/she has not rated.

   The similarity between every pair of users/items should be computed using the Hadoop framework.

2. Create training and test sets from the data. Compare the performance of the two approaches described above in terms of their accuracy on the test set. You are free to consider other approaches as well (e.g., Mahout's recommender system).

3. Develop a web-based interface that provides the following functionalities [Optional]:

   Allows a user to login and logout from the system.

   Displays items the user has rated.

   Provide recommendation to items the user has not rated.

   Provide recommendation of other users who share similar preference.

4 Real Time Sentiment Analysis of Twitter Data Using Hadoop

Social media for many people has become integral part of their daily life. Social media metrics are now considered parts of altmetrics, which are non-traditional metrics proposed as an alternative to more traditional metrics.

Twitter is an online social networking service that enables users to send and read short 140-character messages called "tweets". Registered users can post and read tweets, but general public can also read them. This is unlike Facebook, where social interactions are often private. Users access Twitter through the website interface, SMS, or mobile device app.
You can develop the application based on Apache Storm, a distributed computation framework, which adds reliable real-time data processing capabilities to Apache Hadoop. It is fast, scalable, reliable and can be programmed using a variety of programming languages (Python, Java, Scala).

Algorithm Sentiment analysis or opinion mining refers to the use of natural language processing and text analysis to identify and extract subjective information in source materials. Normally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual of a document(s).

Possible tools

- Big Data: Apache Strom, Apache HBase, Twitter's Search and Streaming APIs,
- Visualization tools: D3 Visualization, Tableau visualization.
- Development tools: Python, R, Java, and Scala.
- Natural Language Processing Algorithms: Python Natural Language Toolkit (NLTK) and AlchemyAPI Service

5 Predicting Airline Delays with Hadoop

One of the main goals is using machine learning algorithms to build predictive models with Python packages and data analysis programs. Training the original datasets is important to build models with its performance. Finding a good combination of technologies and programming languages would be cruicial to make a successful project.

Dataset The data can be downloaded from Bureau of Transportation Statistics where it is described in detail. An other link to more detailed data can be found here.

Possible tools

- Apache Pig
- Hadoop
- Python
- scikit-learn

6 Streaming Text Analytics using Python

This project proposition consist of 2 part :

A. Identifying Trends in Twitter ´

Twitter is one of the main online social networks where users post and interact with messages known as "tweets". Tweets allow for instant, short, and frequent communication and they have been proved an effective way to communicate news and other timely information. Therefore, a practical use for Twitter's functionality is to be used for identifying trends in real-time. Identifying trends is important for several industries and services, including marketing, customer service, and crisis response.

Your task is to design and implement a Twitter streaming application that tracks specific hashtags and reports their popularity (# occurrences) in real-time. In particular, you need to:

- Identify 5 related #hashtags (e.g., political parties, companies, product brands, stocks, etc.)
- Collect tweets mentioning any of the 5 #hashtags in real-time
- Compute the number of occurrences of each of the mentioned hashtags
- Plot the results of your analysis in real-time. Alternatively, you can decide to store the results in a file, post-process them as a batch (offline) and create a plot based on the post-process analysis. The results are based on the time window that your application is running (from the time it begins, until it is killed or interrupted/stopped).

For the needs of your assignment, you will need to stitch together a number of technologies that can enable the analysis to be performed, including:

A Twitter client: This is an application that connects to the Twitter service and obtains tweets as they become available. It requires to create your own credentials to access the Twitter APIs. See Appendix A.

Apache Spark Streaming: This is an apache spark streaming application that connects to your twitter client, receives the tweets as a stream, performs real-time processing of the incoming tweets, extracts useful information, and computes the quantities of interest (i.e., number of occurrences of a #hashtag).

Real-time reporting: This is a visualization component that reports through a plot the results computed by the apache spark streaming application in real-time. This can be implemented using AJAX (asynchronous HTTP calls); see the resources of the Appendix B for examples. Alternatively, results can be stored in a file in real-time, post-processed as a batch (offline), and presented as a plot.

Notes:

- Several implementation approaches exist for this application. You are encouraged to make assumptions, make decisions and follow the technical path that you feel is more appropriate. You will have a chance to explain your approach during the marking session
- Appendix A provides instructions on how to setup a Twitter application
- Appendix B provides several online resources related to the assignment

B. Real-time Sentiment Analysis of Twitter Topics

In the field of computational linguistics and natural language processing, *sentiment analysis* aims to determine the attitude of a speaker, writer, or other subject with respect to some topic or the overall contextual polarity or emotional reaction to a document, interaction, or event. A basic task in sentiment analysis is classifying the **polarity** of a given text at the document, sentence, or feature/aspect level—whether the expressed opinion in a document, a sentence or an entity feature/aspect is *positive*, *negative*, or *neutral*. For example, consider the following three text inputs:

"I love ice cream a lot"

"I dislike ice cream a lot"

"ice cream is made from milk"

One would expect that the polarity of the first is (rather) *positive*, of the second is (rather) *negative* and of the third is (rather) *neutral*. The word "rather" is used here to express subjectivity, since humans not always agree about the polarity of a sentence. We rely on an out-of-the-shelf library to perform sentiment analysis. The analysis will be on the level of a document, where the document is a tweet (i.e., all the words in a single tweet).

Your task is to design and implement a Twitter streaming application that performs sentiment analysis of tweets related to competitive topics and provides a real-time monitoring of the polarity.

- Identify 5 competitive topics (e.g., political parties, companies, product brands, stocks, etc.)
- Manually select a set of 10 hashtags that better describe each of the topic identified above
- Collect tweets related to the 5 topics in real-time and perform sentiment analysis for each topic
- Plot the results of your analysis in real-time. Alternatively, you can decide to store results in a file, post-process them and create a plot based on the post-process analysis

Notes:

- The implementation approach for the streaming application should be similar to the one followed in Part A
- For the sentiment analysis you should employ Python's Natural Language Toolkit (NLTK) library
- Appendix A provides instructions on how to setup a Twitter application
- Appendix B provides several online resources related to the assignment

Appendix A – Setting up a Twitter Application & Installing Tweepy

To start collecting tweets, you need to set up a Twitter application and get credentials that allow you to pull tweets out of the twitter streaming API. Then, you need to develop a Twitter client that connects to Twitter and acquires Twitter data. You can do that using Tweepy.

Create a Twitter Application and Obtain OAuth Access Keys

Briefly, you need to:

- Create a Twitter developer account: https://developer.twitter.com/en/apply-for-access
- Create a New Application
- Fill in your Application Details
    - *Name*: Your app name. It needs to be a unique name across all twitter applications
    - *Description*: A short description for your app
    - *Website*: The website address where the app will be hosted. Use a placeholder for now
    - *Callback URL*: Ignore this field
- Create Your Access Token
- Choose what Access Type You Need (choose 'Read only')
- Make a note of your OAuth Settings

Once you've done this, you will have the following OAuth settings.

- Consumer Key
- Consumer Secret
- OAuth Access Token
- OAuth Access Token Secret

You should keep these secret, since anyone with the keys, could effectively access your Twitter account.

Detailed information is provided here: http://docs.inboundnow.com/guide/create-twitter-application/

Install Tweepy

Tweepy is a python library for accessing the Twitter API. You can install Tweepy using pip:

```
$pip install tweepy
```

You may also use Git to clone the repository directly from Github and install it manually:

```
$git clone https://github.com/tweepy/tweepy.git
$cd tweepy
$python setup.py install
```

The next step is to use Tweepy to create a Twitter application that uses your Twitter credentials.
More information: https://github.com/tweepy/tweepy

Appendix B – Useful Online Resources and Tutorials

Spark Streaming Programming Guide
http://spark.apache.org/docs/latest/streaming-programming-guide.html

Python Streaming Examples
https://github.com/apache/spark/tree/master/examples/src/main/python/streaming

An easy-to-use Python library for accessing the Twitter API
http://www.tweepy.org/

Apache Spark General Tutorial
https://www.toptal.com/spark/introduction-to-apache-spark

Apache Spark Streaming Tutorial: Identifying Trending Twitter Hashtags
https://www.toptal.com/apache/apache-spark-streaming-twitter

Twitter Trends Analysis using Spark Streaming
http://www.awesomestats.in/spark-twitter-stream/

Apache Spark Streaming with Twitter (and Python)
https://www.linkedin.com/pulse/apache-spark-streaming-twitter-python-laurent-weichberger/

Twitter-Sentiment-Analysis-Using-Spark-Streaming-And-Kafka
https://github.com/sridharswamy/Twitter-Sentiment-Analysis-Using-Spark-Streaming-And-Kafka

Twitter Sentiment Analysis using Python
https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/

Sentiment Analysis on Reddit News Headlines with Python's Natural Language Toolkit (NLTK)
https://www.learndatasci.com/tutorials/sentiment-analysis-reddit-headlines-pythons-nltk/