Lyon Eye

Application using Big Data Analytics
over the Open Data of Grand Lyon-France

Bastrakova E.[†], Garcia-Calderon S.[†], Ledesma R.[†], Lopez-Roa C.[†], Kalyan K.[†], and Millan J.[†]

[†] DMKM, Universite Lumiere Lyon 2, Lyon, France

June 11, 2016

Abstract

As part of an effort to use big data analytics and data warehousing in the context of the open data of Grand Lyon, France. We designed and implemented an open-source web-application to explore the touristic points of interest as well as to do sentiment analysis in the related social network data. Here we describe from design to implementation of the application, the ETL & ELT[1] jobs, text mining algorithms, OLAP processes, user interface and deployment in a cloud server. The source code as well as the documentation can be found in Github and the live application in available under this URL. Tableau Dashboards are available in this link. A demo of the application can be found in this video. You can find all the technical documentation in this link

Keywords— big data analytics, data warehouse, OLAP, text mining, sentiment analysis, open data

Contents

[*]Different from traditional ETL (Extraction, Transformation and Loading) jobs, the more recent ELT approach will first Load and then Transform, thereby keeping as much data as possible

1 Introduction

Web applications to support travel experience are part of the emerging *collaborative* economy. In this, the value is provided by the users rather than by the producer. Following the courses of Complex Data Warehousing and Big Data Analytics we got motivated to develop an application to harness the potential of mining social data to provide value in a local context, in this case, to the Grand Lyon Metropolitan Area. Also, the philosophy was kept as open source, both in the developed code as in the components and APIs used. Data collection both in the form of scheduled traditional ETL jobs and real time ELT processes. A variety of text mining tools were then applied in the analytical pipeline of the application, where Sentiment analysis plays a central role. Data is consolidated in a data warehouse and then data visualization is possible thanks to traditional OLAP tools and a custom made, mobile-ready, user interface to provide added value.

In this paper we present the design and implementation of a map application to display points of interest in Grand Lyon adding value by aggregating social network sentiment (Twitter) and ratings (Foursquare & Yelp) and forecasting future ratings over time. The same data warehouse that powers the application can be exploited using traditional OLAP analysis.

Through out the development of the application we followed the EMC's Data Analytics Life Cycle to correctly identify, test and evaluate analytical questions

and hypothesis.

The paper is organized in the following way: First a exploration of the current proposal is made, then the full contribution from design to implementation is described, a few results in the analysis built and some discussions are drawn. In the end we present the conclusions and future work perspectives.

2 Related Work

Web services such as AirBnB, TripAdvisor, Foursquare, Yelp are examples of where the value, in this case taking the form of information, is provided directly by the user, and the enterprise behind the development of the application only takes care of the user experience and the availability of the application.

But not only commercial value has been extracted of collaborative economy application to support travelers, also, governments are interested in positioning their touristic offer and services available in the web to potential travelers and citizens. Examples of this are interactive map applications like: Paris Lyon Saint-Priest, TCL Lyon, and Velo-v Grand-Lyon. However this applications do not incorporate more information than the available to the governments, that is, these applications are not part of collaborative economy.

3 Contribution

Here we present the design of the components involved in the application. This means, from data sources, data collection, data exploration, data transformation and value extraction as well as data visualization.

3.1 General Components

A basic diagram of the main components of the application can be seen in figure 1.

In the following sections we describe each component design extensively.

3.2 Data Sources

As part of a joint effort to take part in the *Smart City* movement and to *accelerate innovation and encourage citizen participation*, the Metropolis of Grand Lyon has released several repositories of Open Data. The data encompasses around 622 data sets divided 14 categories, both in static and real-time feed.

For this project we chose the Point d'intérêt touristique data set, that contains several features regarding points of touristic interest in the following categories:

1. Cultural Heritage

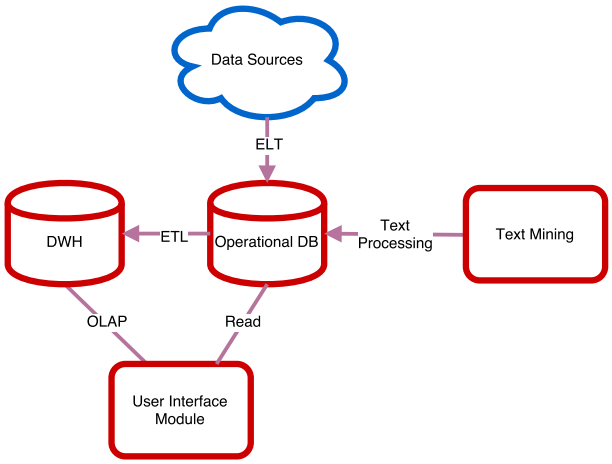

Figure 1: Basic schema of the components involved in the application. As well as their interactions. The data is first collected from Data Sources using ELT jobs and written to the operational database in this text processing is done. Once processed a ETL job writes into the data warehouse. From this data can be exploited using OLAP tools, The data is also read from the operational database for being displayed in a custom made User Interface.

2. Tasting
3. Natural Heritage
4. Business and Services
5. Equipment
6. Restaurants

Also, two social networks, Yelp & Foursquare, which are specialized in ratings and reviews of public places were used as data sources, cross referencing the sites of the Grand Lyon Data set. Some additional metadata of the places was queried from these sources. Several ELT jobs were scheduled to query the data source of: Grand-Lyon, Yelp and Foursquare, and load the data into the operational database.

The social network Twitter was used to provide a massive free text corpus to perform relevant text mining techniques and algorithms. A real time ELT job was running 24/7 to collect and store data coming from Twitter data source into the operational Database. A graph showing the number of tweets collected per day is shown in figure 2.

Twitter data source was queried for certain keywords, those that are a super set of the interest points.

3.3 Text Mining

To correctly extract value of the unstructured data collected via twitter, we designed a pipeline to as-

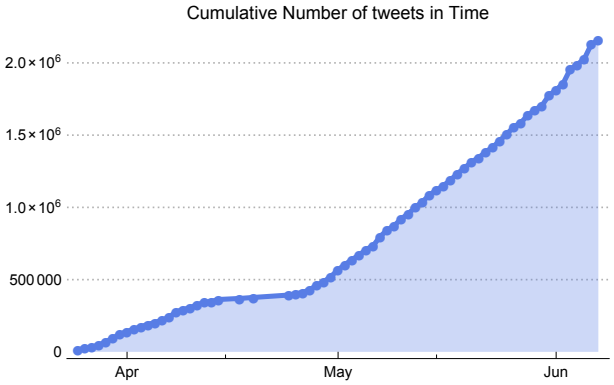

Figure 2: Here we can visualize the total number of tweets collected in a time windows of 68 days. Reaching more than two million and counting.

sign sentiment to each tweet and also, to identify the touristic point that is being referenced. Sentiment is the attitude, opinion or feeling toward something, such as a person, organization, product or location. Here we assigned a five star-like sentiment with the following equivalence:

Rating	Sentiment
⋆	Very Negative
⋆⋆	Negative
⋆⋆⋆	Neutral
⋆⋆⋆⋆	Positive
⋆⋆⋆⋆⋆	Very Positive

Figure 3: Equivalence of rating with sentiment. In this case a three star rating is equivalent to neutral sentiment and a five star rating to a very positive sentiment.

Tweets

As a first step pre-processing in the tweets text is done using regular expressions, namely: converting to lower case, removing URLs, removing user names, removing additional white spaces, converting hashtags into words, etc.

Several methods were tested to assign sentiment to the text corpus contained in every tweet. A private API acquired by IBM, AlchemyAPI, a Bag of Words model and also a Support Vector Machine Model were used.

The AlchemyAPI is specialized in Natural Language processing, including sentiment analysis. It supports several languages, namely: English, French, Italian, German, Portuguese, Russian and Spanish. It provides a numeric response for a given document that represents the overall sentiment of the document.

The Bag of words model can be seen as a weigthed vectorial representation of a document. That is a document d represented as the multiset W of words and their frequency, $\{(w_i, f_i)\}$. A separate document, called a *dictionary* (δ) contains weights for specific words that represent the sentiment of this words. Taking the summation:

$$s = \sum_{i}^{n} W_i \cdot \delta \tag{1}$$

we can find the sentiment contained in the document d. In this case we used two dictionaries, both, in English and French.

The Support Vector Machine takes a vectorial representation of the documents weighted by TF-IDF frequency and is trained to separate multidimensional regions of the labeled vectors. In this case the labels where taken as a vote of both previous models, taking the tweets in which the sentiment of each model is such that

$$|s_b - s_a| < 1, \tag{2}$$

where s_b is the sentiment given by the bag of words model and s_a is the sentiment given by the AlchemyAPI model. This means, the disagreement would not be more than one unit.

As baseline a Random Forest and Naive Bayes models were trained with the same training set and results evaluated.

Interest Points

First, all touristic points are pre-processed, substituting unwanted characters and removing stopwords both in french and english, these are treated as *keywords* for each site. Then a many-to-many relation between tweets and interest points is created, that is, a tweet can be related to several interest points and, naturally, each interest point can be related to several tweets. We say that a interest point is related to a tweet if

$$\frac{|K \cap T|}{|K|} \geq \alpha, \tag{3}$$

where K is the set of keywords, T is the set of words in the pre-processed tweet, and α is a parameter such that $0 \leq \alpha \leq 1$.

3.4 Data Warehouse

With the results from the text processing steps, an ETL process is performed from the operational database to the data warehouse. An overview of this process can be seen in figure 4. From this process, a data warehouse is created so the information of the different ratings can be analyzed per interest point, per location and per date. The schema of the data warehouse can be appreciated in figure 5.

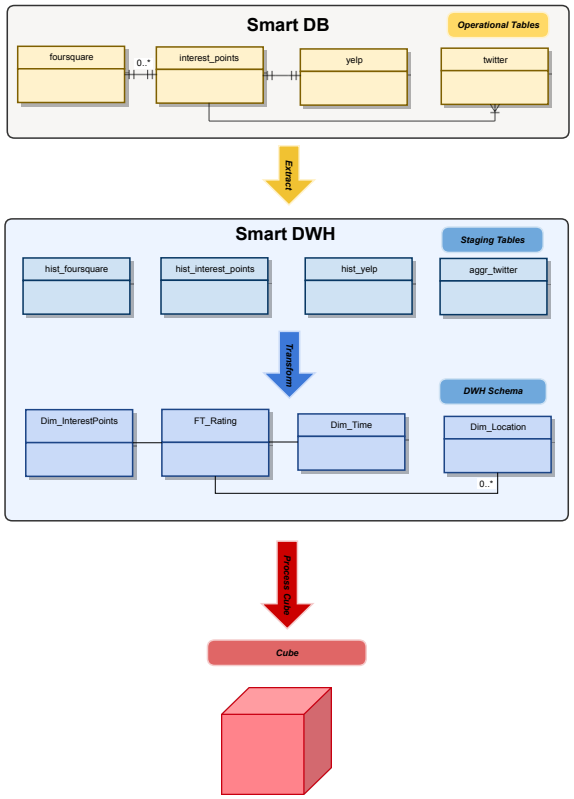

Figure 4: ETL Job. After each data source has been written to the respective operational table, a extraction is performed to populate historical staging tables in the database. Then changes are calculated to write the facts and dimensions into the data warehouse schema. From this we can construct a ROLAP cube to explore and exploit data.

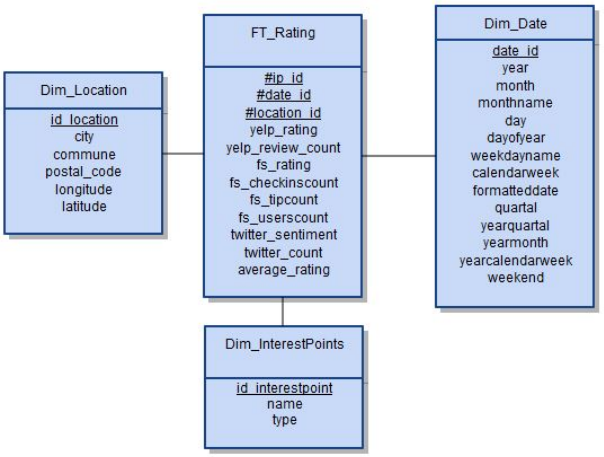

Figure 5: Schema of the data warehouse. We defined five measures that can be observed by three dimensions in a star schema. The dimensions: Location, Time, Interest Points, are calculated during ETL to update them if necessary.

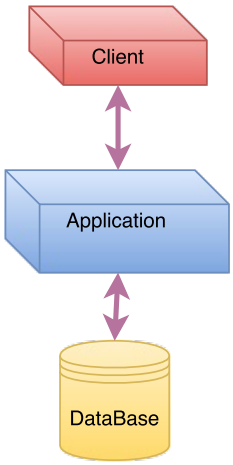

Figure 6: In a 3-tier architecture, the data persistence is done in the database, all the business logic, data access and user interface is done in the application layer, and the client renders the assets in a web browser.

3.5 OLAP

After constructing a data warehouse, OLAP analysis was made possible. Using the commercial software Tableau we were able to explore the data cube generated and produce some custom made Dashboards. These Dashboards are available in the following link.

3.6 User Interface

A custom made, multilingual, responsive, user interface was built from scratch, having as objective: To present all the data points in a mobile-ready web application. The web application was developed as a 3-tier architecture. A general diagram of this architecture is shown in figure 6.

Several features were included in the user interface, a screen-shot of the application front end is shown in figure 7.

1. **Multilingual:** All texts can be shown in any of seven languages to choose: English, French, Spanish, Russian, Hindi, Chinese, or German.

2. **Map showing interest points:** Each interest point is plotted in a map having precise geolocalization having the marker color set according to the five-star ranking. Colors vary from red, to green. Points are grouped in clusters when many points are too close. The cluster points shown the number of points contained in the cluster and the color of the cluster marker is also color coded. The tooltip of a point is it's name and the tooltip of a cluster point is the number of points contained and the average rating of them.

3. **Filter sidebar:** It provides the possibility to

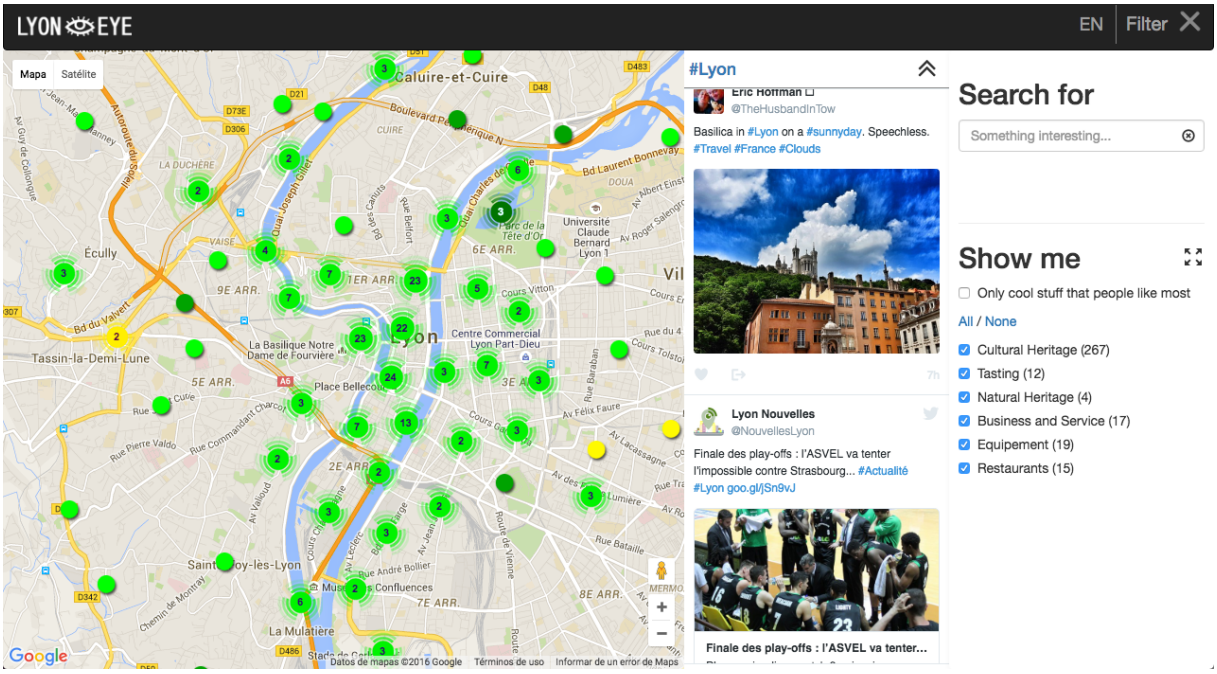

Figure 7: Front end of the application as shown in the client side. Several points as well as cluster points can be seen color coded by their ranking. In the right we can see the

filter the points by category, to center the map around the shown points, to filter only the top rated points (more than 4 stars), and it's also possible to search points by name. While the search is typed a list of possible matches is shown, when the enter key is pressed a subset of the points matching the introduced search term is displayed.

4. **Social Feed:** An additional sidebar showing the real time tweets relevant to the Lyon is also available to the user.

5. **Point tooltip:** When clicking a point the relevant information of the point is shown in a tooltip, mainly: Name, category, star-rating, address, web page, schedule and more information link.

6. **Point details:** When clicking in the more information link of the point tooltip a new screen is shown where we can see more information of the point: Name, rating, category, address, web page, phone and schedule, also a photo thumbnail of the place. A graph showing the rating as a function of time for the data sources: Yelp, Foursquare and Yelp is shown as well as the overall average of these three. A forecast of the expected overall rating is done for a period of one week.

7. **Related tweets:** Also the related tweets to each data point can be seen in this second window, feedback of the assigned sentiment is possible using a positive and negative vote buttons.

The forecasting of the rating over time is done using the Triple Exponential Smoothing method. This is specially adequate for data showing both trending and seasonality. The forecast can be expressed as:

$$F_{t+m} = (S_t + mb_t)I_{t-L+m} \tag{4}$$

where

$$\begin{aligned} S_t &= \alpha \frac{y_t}{I_t - L} + (1-\alpha)(S_{t-1} + b_{t-1}) \\ b_t &= \gamma(S_t - S_{t-1}) + (1-\gamma)b_{t-1} \\ I_t &= \beta \frac{y_t}{S_t} + (1-\beta)I_{t-L} \end{aligned} \tag{5}$$

and y is the observation, S is the smoothed observation, b is the trend factor, I is the seasonal index, F is the forecast at m periods ahead, t is an index denoting a time period. α, β and γ are constants that mus be estimated in such a way that the MSE of the error is minimized.

3.7 SmartLabs

As a part of a continuous integration and improvement of the analytical model a data laboratory was set up in development environment, in it we deployed a REST service that returns a sentiment to a given text using the analytical model in production.

Also a exploration in a Word2Vec model was done with promising results. This model maps the occurrence of words within a document to a high dimensional vector space. Some properties that are conserved is that similar words, or words with frequent co-occurrence are mapped *close*, while, non related words are mapped *far* from each other.

4 Implementation

Now we describe the components and technologies used during implementation. A detailed description of the infrastructure as well as all the components and their interactions is can be seen in figure 8

4.1 Infrastructure

A cloud virtual private server running Ubuntu 14.10 in Amazon Web Services (AWS) was used to support the application for public access and constituted the Production environment. A virtual machine running the same Linux version was used inside a VPN as development environment. All code was transported and versioned using Git client and a GitHub public repository (available here).

Database and data collection

The database choice was PostgreSQL 9.5, which is an advanced open source RDBMS [1]. This database is responsible for both the operational DB and the data warehouse components.

The module for retrieving the source data from the available sources has been developed mainly using Python to connect to the sources and Pentaho's Kettle [2] to perform the synchronization process with the operational database.

For the application developed the GrandLyon Open Data, more specifically the Touristic Interest Points (Point d'intérêt touristique) available online, is exploited with the given REST API [3].

Using the Yelp Python library [4], we search in Yelp for the interest points we already retrieved from the GrandLyon Open Data and bring the information of them when a match occurs.

By connecting to the Foursquare REST API [5], we search in Foursquare for the interest points we already retrieved from the GrandLyon Open Data and bring the information of them when a match occurs

For collecting tweets in real time and 24/7 the `tweetpy` Python library [6] was used, grabbing all tweets that contained at least one of the following keywords: 'lyon', 'villeurbanne', 'bron', 'bellecour', 'fourviere', 'gerland', 'venissieux'.

ETL and Data Warehouse

For the ETL jobs, the Pentaho's Kettle Data Integration tool [2] was used to schedule and perform the ETL jobs between the operational database and the data warehouse, that is, to perform table compares, delta merges and updates and inserts from one set of table to the other.

Considering the asynchronous process of processing and rating tweets, along with the nature of the information we use to feed the data warehouse, an out-of-standard update process of the data warehouse is performed. This allows us to include within the metrics of the data warehouse the possible information that was processed after the previous update, which otherwise would be never taken into account. It is important to understand that with this decision it is possible that a reading of the ratings at one moment is different from a previous reading, nevertheless, and considering that this variance has no impact over the possible analysis that can be performed, we prioritize having all the possible information reflected over the data warehouse.

OLAP

Tableau [7] was used in order to do OLAP analysis. Tableau is business intelligence software that allows to easily connect to data, visualize and create interactive, sharable dashboards. The version we used was Tableau 9.2.6

The following dashboards were developed, which allows to solve the following business queries:

- **Sentiment by Time Dashboard:**

  Sentiment by Week: Displays a comparison between the sentiment average of the different data sources by week.

  Sentiment by Day of Week: Displays a comparison between the sentiment average of the different data sources by day of the week.

  A sample of this dashboard can be seen in figure 9.

- **Activity by Location Dashboard:**

  Top 10 Tweets by Commune in Lyon: Displays the Sectors of Lyon with the highest mentions of Interest Points in tweets.

  Bottom 10 Tweets by Commune in Lyon: Displays the Sectors of Lyon with the lowest mentions of Interest Points in tweets.

  Sentiment by Area: Displays a map of the city of Lyon with the interest points in proportion to mentions and the color according to the average sentiment

  A sample of this dashboard can be seen in figure 10.

- **Type of IP Dashboard:**

  Popular Type of Interest Points: Displays a proportion of data for each type if Interest Point.

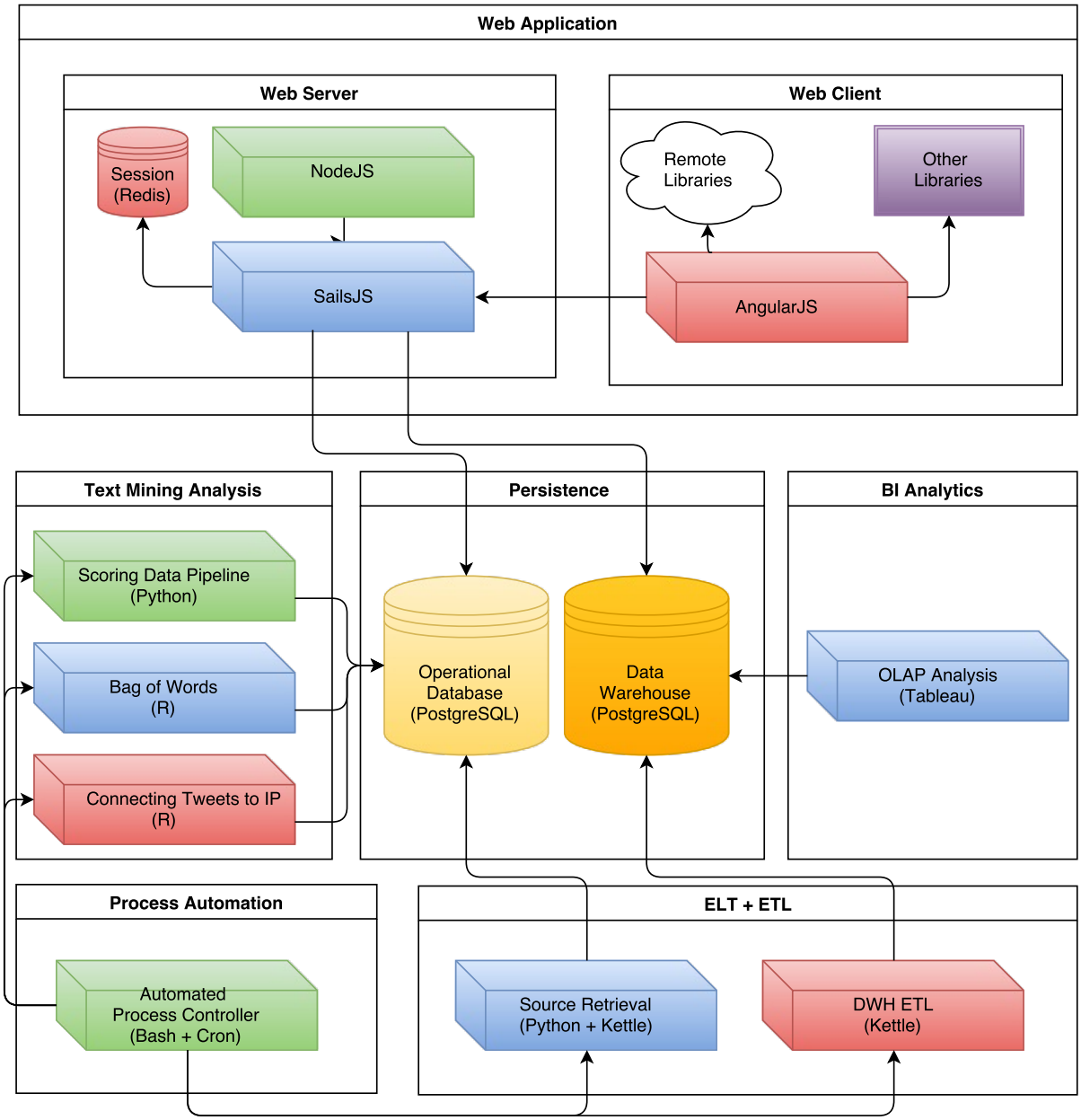

Figure 8: As we can see the database plays a central role in the application. The automated process controls the different tasks of the application. Source Retrieval populates the operational database, the Text Mining jobs are launched synchronously to process the ever arriving Twitter feed. The data warehouse ETL job is constantly and automatically generating the facts in the DWH from the operational data. Also, the OLAP analysis tool can read the DWH to exploit it in rich and easy to use interactive web-based Dashboards. Ultimately the web-application reads both the operational database and the data warehouse to display the relevant information in the described interface. It controls all the logic from in the server side using JavaScript frameworks and keeps the session information of the client in the in-memory Redis data-store. On the client side, more JavaScript frameworks and libraries are loaded to correctly display the application in the client's web browser.

Sentiment by Type of Interest Point: Displays the average sentiment for the types of Interest Points.

A sample of this dashboard can be seen in figure 11.

4.2 Processing pipeline

The data pipeline for processing and scoring the tweets consists of three steps

1. **Reading raw data from database:** Chunks of 700 tweets are pulled from the Database.

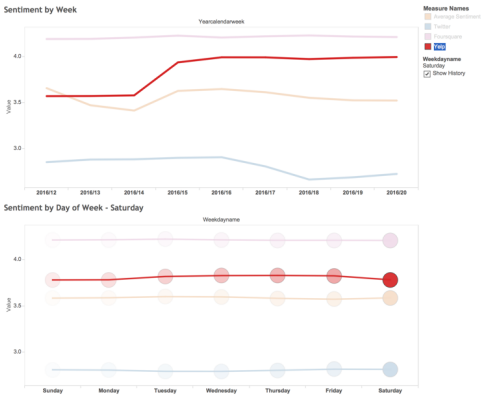

Figure 9: Tableau Dashboard for Sentiment by Time.

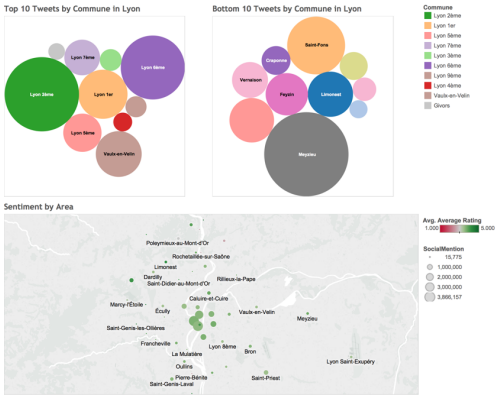

Figure 10: Tableau Dashboard for Activity by Location.

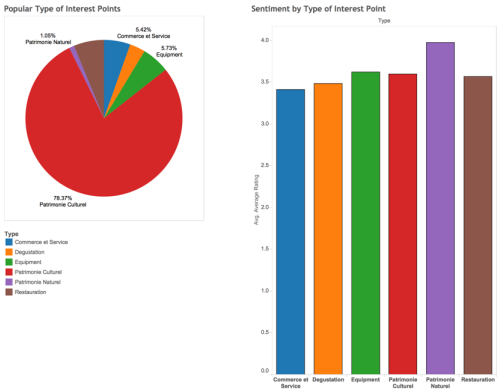

Figure 11: Tableau Dashboard for Type of IP.

2. **Preprocessing raw data:** To provide accurate results, text preprocessing is done as previously described.

3. **Scoring:** The scoring is done in Python using the `sklearn` [8] library using the models previously described.

4.3 Web-Stack

The Web User Interface is built on NodeJS using the framework SailsJS on the server side and AngularJS on the client side. It is connected to PostgreSQL for managing the data and to Redis for managing sessions. The versions are: NodeJS: 4.4.1, PostgreSQL: 9.5, Redis: 2.8.4.

A general diagram showing the components of the web-stack and their interaction in the 3-tier architecture is shown in figure 12

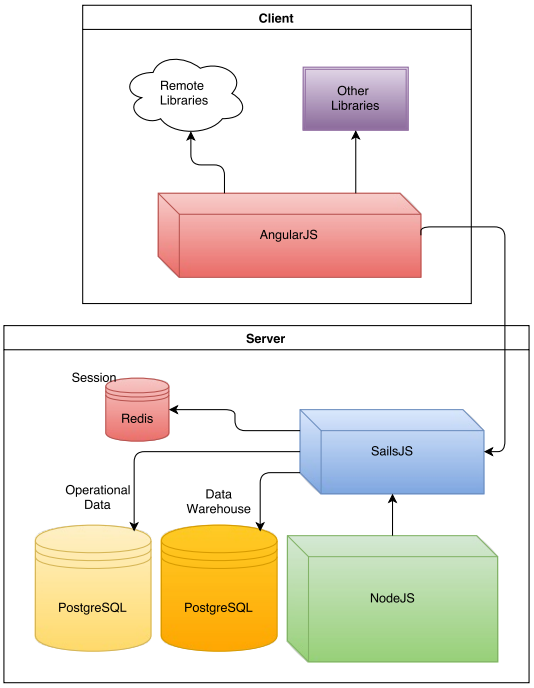

Figure 12: Detailed architecture of the web stack.

5 Model Evaluation

To evaluate the accuracy of the generated model and to compare it to other models several test were carried out, from which the following provided the most insightful results.

Unsupervised learning

A manually disambiguate data set was used to evaluate the performance of unsupervised learning models, i.e., the bag of words (BOW) and AlchemyAPI (Alch) models.

The data set consists in around 1,200 tweets of costumers speaking about the Apple iPhone, which were manually labeled with a sentiment using the Amazon Mechanical Turk work marketplace. These labels where used as ground-truth for the BOW and Alch models, and both precision and recall evaluated as follows

$$F_1 = 2\frac{p \cdot r}{p + r}, \tag{6}$$

where p is the precision and r is the recall, i.e. the harmonic mean of the two.

Both the labeled data and the model output where grouped in binary classes, as, positive and negative sentiment to avoid intraclass errors.

Test2: Supervised learning

To evaluate the performance of the supervised learning models, i.e., the Support Vector Machine (SVM), and the baselines, Random Forests (RF) and Naive Bayes (NB); the testing data was splitted in training and testing sets in the proportion 80% and 20% respectively. The average F_1 measure was again evaluated.

6 Results

Results of the Test1 are shown in figure 13

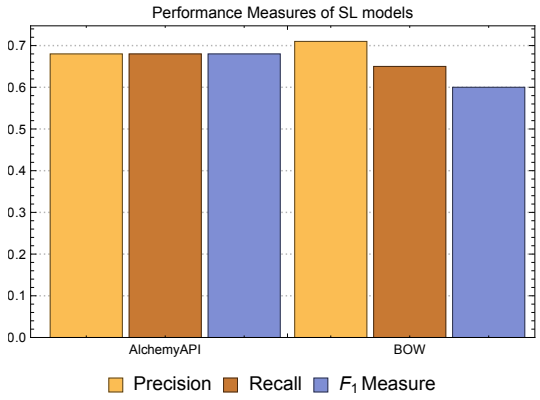

Figure 13: Results of the Test1 on the Supervised Learning models. We can see that overall the average F_1 score of the Alchemy API is slightly superior as the one of the BOW model.

Results of the Test2 are shown in figure 14

Results of Test1 and Test2 can be seen in table 6

	Precision	Recall	$< F_1 >$
Alchemy	0.68	0.68	0.68
BOW	0.71	0.65	0.60
NB	0.65	0.64	0.63
RF	0.72	0.71	0.70
SVM	0.72	0.73	0.72

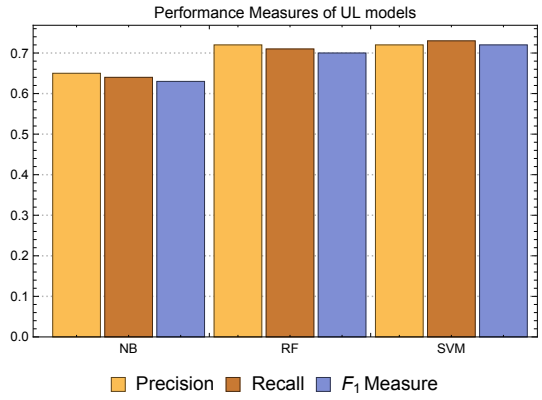

Figure 14: Results of the Test2 on the Unsupervised Learning models. We can see that the NB model provides the baseline followed by the RF and above all is the SVM model.

7 Conclusions & Future Work

As part of an effort to use big data analytics and data warehousing in the context of the open data of Grand Lyon, France. We designed and implemented an open-source web-application to explore the touristic points of interest as well as to do sentiment analysis in the related social network data. Here we describe from design to implementation of the application, the ETL & ELT jobs, text mining algorithms, OLAP processes, user interface and deployment in a cloud server.

We were able to aggregate the different sources of information (GrandLyon Open Data, Yelp, Foursquare and Twitter), process their information and make it useful for a single web user, by processing and implementing data mining techniques to identify the most current sentiment of each interest point in the city.

Even though the web client interface provides the features required to visualize and interact with the application, further features can be considered in order to enrich and facilitate the user experience. Among this we can find: user session and personalized experience, full integration with the Tableau OLAP Dashboards, user feedback about the information displayed and social networks integration.

The implementation of a modular application allowed us to experiment and try different approaches for the different analysis, while keeping the information consistent and each module independent. Considering this we created a scalable and modifiable application that can be reproduced for any other mayor city in the world.

References

[1] The PostgreSQL Global Development Group. (2016) Postgresql 9.5.3 documentation - fuzzystrmatch. [Online]. Available: https://www.postgresql.org/docs/9.5/static/fuzzystrmatch.html

[2] Pentaho community. (2016) Kettle. [Online]. Available: http://wiki.pentaho.com/display/ServerDoc2x/Kettle

[3] Métropole de Lyon. (2016) Données métropolitaines du grand lyon. [Online]. Available: http://data.grandlyon.com/

[4] Yelp. (2016) Yelp/yelp-python: A python library for the yelp api. [Online]. Available: https://github.com/Yelp/yelp-python

[5] Foursquare. (2016) foursquare for developers. [Online]. Available: https://developer.foursquare.com/

[6] P. Rivera. (2016) Tweepy. [Online]. Available: http://www.tweepy.org/

[7] T. Software. (2016) Business intelligence and analytics | tableau software. [Online]. Available: http://www.tableau.com/

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.