|  |  |
| --- | --- |
| Trust  It |  |
|  |  |
| 5.2.2020  Authored by: Marija Stankovic  Mentor: Manuel Garcia Vega |  |

# Table of contents

Description of the practice........................................................................................3

List of the problems raised and the procedure followed for resolution......................7

Identification of the contributions that, in terms of learning, have involved practices

Assessment of the tasks developed with the knowledge and skills acquired in relation to university studies.........................................................................................................**8**

Evaluation of practice and suggestions for improvement...........................................9

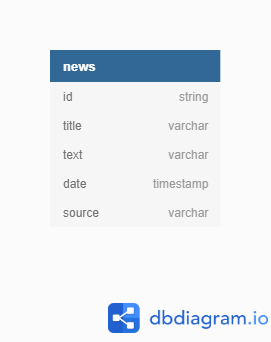
# Description of the practice

Our goal was to make an application that could detect a fake news and, in that manner, help the users distinguish lie from truth. The making of an application consisted from 3 phases:

1. building a database of news for which we could claim it is reliable or not,
2. making an algorithm that uses the database, google search results and a neural network and
3. making an interface for the users

## Building a database

Our database consists of two parts: reliable and unreliable news. At first, we used MongoDB for holding our data but because our database has only one table (image 1) we decided later to use a csv files for storing the data.



*Image 1*

The reliable and unreliable news are saved into different files, so the attribute on weather a news is unreliable is unnecessary.

My part was to get the unreliable (fake or sarcastic) news into the database. The sources considered as such are *Daily Squib, The Chaser, The New Republic, The Telegraph, Weekly World News, Fox News, Fox News, The Hard Times, The Sun, The Babylon Bee, The Beaverton, The Rochdale Herald.* Our intention is not to discretize or any of the mentioned sources and we apologize if we offended anyone. This assumption is based on thorough research on lists and web-pages on this topic such as <http://www.fakenewscodex.com/>.

The news is inserted from internet into the database using python scripts. Each python script scrapes news from one source. The main script calls a scraping function from each python script, so once it is running, news from all the sources are scraped and written into the database.

The *ReadWriteCSV* script handles reading and writing in csv file. Once all the news are scraped and loaded into an array of program, they are written in the file all at once using a *dump* function.

## Algorithm for evaluating the news

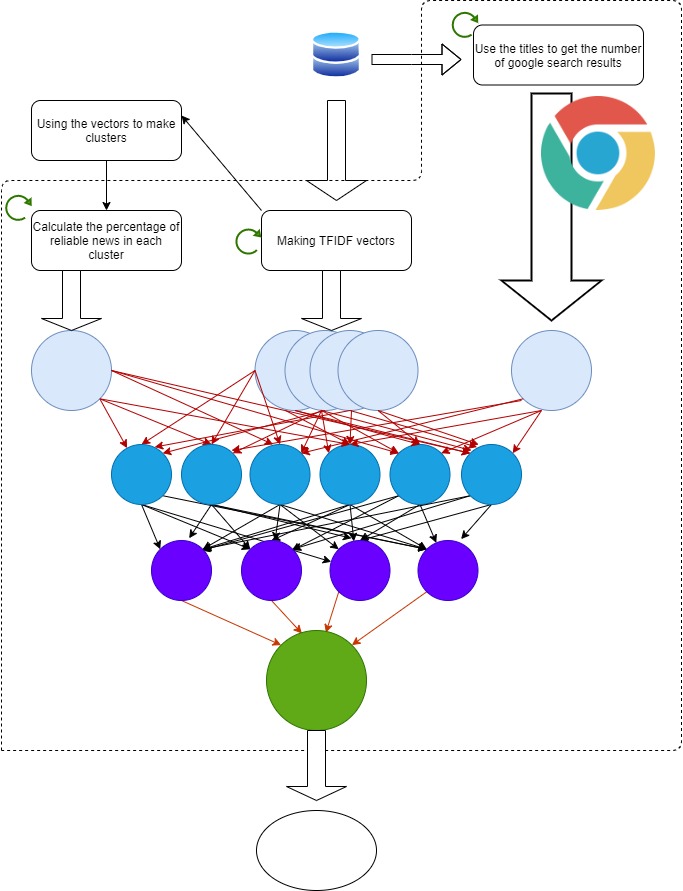
For making a prediction on weather a news is unreliable or not, we use two factors: the database and the google search results.

The algorithm first loads all the news, calculates the TFIDF (Term Frequency Inverse Data Frequency) vectors and creates clusters using Kmeans classifier. Then for each cluster the percentage of reliable news in the cluster is calculated and a neural network (multilayer perception) is trained on the vectors.

When a news is represented for evaluation, a vector is made from this text. From all the words in the vectors, only 2% of the most important ones are taken into consideration. Also, a cluster prediction is made.

The part with Google API was done by my teammate.

Finally, using **the 2% of the most important words**, **percentage of the reliable news in a cluster predicted for the news** and **number of research results on Google for the topic** as inputs for neural network, a final conclusion is made on whether a news can be considered as true or not.



The interface is a web application written with html, js and css. The user should enter the date, title and text of the news for which he wants the evaluation. Then, after pressing a button, the data entered is sent to the server using fetch method. Server then evaluates the news using the already described algorithm and sends the evaluation back to front-end in JSON format. The result is rendered.

The result is described with status *reliable* or *unreliable.*

On the web page, the user can also see the news recently scraped from reliable news sources and can visit the source of those news.

# List of the problems raised and the procedure followed for resolution

The problems that occurred with the projects are mostly in relation with encountering with, for us, new technologies. The most time we spent was for learning how to work with python, CSV files or connecting python and front-end web page. With this, we had problems with making the best decision in the new environment. Also, what made a big problem for us was using MongoDB in the first phase of our project and later deciding to switch to CSV files. First, we chose to work with database because it already had efficient mechanisms for checking key value and writing. Later, when we realized that our database is just one entity and that we could manage it on our own, we transited to CSV file. Because of this we had to change all of our scripts and to test everything again and of course new bugs appeared which required extra time to solve. To check if the key already exists, at the beginning of the scraping process all ids from the file are loaded into an array and then while scraping, new ids are compared with those from the array and, if for a news is decided to be in a database, its id is also added in the array of ids. As for writing, first, we tried to write in the CSV file each time a news is scraped. That meant opening and closing the file each time, which is around 300 times in total per day. This required a tremendous amount of CPU time. That is why writing in the database file now consists of two parts: call of *write* function which just adds a news object in a global array and at the end of the scaping all the news from all the sources a call of *dump* function writes a series of objects from the array into the database file.

All of these problems weren’t difficult to solve, but they consumed a lot of our time.

In conclusion, I can say that small mistakes at the beginning of the project and bad decisions in the first phase make big problems later in the development.

Identification of the contributions that, in terms of learning, have involved practices

While building a database in MongoDB, learning how to use this database was for us not so difficult since we already had the knowledge from database subjects, although we had more experience with standard SQL bases and this is a NoSQL database. Later, when making a CSV file for database, our mechanisms for management of writing and reading were based on what we learned how regular database functions work.

For writing the algorithm that evaluates the news, the knowledge obtained from other subjects like *Data Mining* helped us to choose the best scikit-learn classificators (KMeans for clustering and Multilayer Perception for neural network) and how to use them. Basic knowledge from general subjects like Mathematics helped us understand vectors in multidimensional space and how to use them.

Since we already made web applications within our web subjects, making interface was not a big problem. We never used python as back-end but we learned quickly what to use to connect the front- and back-end.

For rendering a diagram a bokeh in python is used.

Organization of the teamwork was made with experience from previous subjects that required team projects and with theoretical knowledge from subjects about the management of large groups.

# **Evaluation of practice and suggestions for improvement**

The practice was very interesting. The research both on a technical part of realization of the application and on choosing sources of news, was informative, refreshing and of course will be useful for future work.

The experience of working in a different country and speaking on another language then our own teaches us to be flexible and how to work under various circumstances.

The topic is from a Natural Language Processing subfield opened for us new horizons in a way that we would like to continue the research in that direction.

We have two suggestions for improvement:

The firs would be a more specific classification of the news. The news should be categorized in more groups like false, true but with some false interpretations, pseudoscientific, opinions, satirical and news composed from tweets and quotes. Granulating the classification would give more precise data and hence the prediction would be more precise.

The second suggestion is to use different, more soothing machine learning models special for NLP. We don’t have more knowledge on those so we used usual algorithms.