**Distributed Data Processing and Sentiment Analysis of a Twitter Dataset**

Author: Jefferson William Teixeira

e-mail: [sba22201@student.cct.ie](mailto:sba22201@student.cct.ie)

Student ID: sba22201

Contents

[1 Introduction 3](#_Toc135767520)

[2 Materials and Methods 3](#_Toc135767521)

[3 Data Preparation and Visualization 3](#_Toc135767522)

[4 Distributed Data Processing and NoSQL Storage 3](#_Toc135767523)

[5 Sentiment Analysis and Time Series Forecast 3](#_Toc135767524)

[5.1 Experiment Setup 3](#_Toc135767525)

[5.2 Results 3](#_Toc135767526)

[5.3 Agriculture Import and Export Prediction 3](#_Toc135767527)

[5.3.1 Experiment Setup 3](#_Toc135767528)

[5.3.2 Results 3](#_Toc135767529)

[6 Conclusions 3](#_Toc135767530)

[7 References 3](#_Toc135767531)

# Introduction

In this project, distributed data processing and machine learning (ML) were applied to a large Twitter dataset that contains comments about the war in Ukraine. The dataset with the Twitter user’s comments was collected from a Kaggle [repository](https://www.kaggle.com/code/ssaisuryateja/eda-and-sentiment-analysis/input) and contains about 364,875 tweets on this topic. The dataset was stored using distributed and NoSQL databases and processed for sentiment analysis using time-series to forecast the user’s sentiment in different time intervals as per the experiments presented in this report.

This report is organized as follows: Section 2 describes the tools used to complete this project. In Section 3, the dataset, the data preparation steps, and the interactive dashboard are described. In Section 4, the distributed environment and NoSQL storage experiments are presented, while Section 5 covers the sentiment analysis and time series forecast experiments. Finally, in Section 6, the conclusions are summarized.

# Materials and Methods

In this project, the Cross Industry Standard Process (CRISP-DM) framework (Chapman *et al.*, 2000) was adapted. The advantage of using such a framework is that it can be applied to any domain, so the main tasks are known before the project starts, which contributes to the organization of the required steps. Thus, this project was divided into the following parts:

**Data preparation and visualization**: In this part, the exploratory data analysis (EDA) was performed on the datasets, where they were pre-processed for the statistical and ML analysis. This step was carried on separately, so the output data could be consumed by both analyses, avoiding rework and code duplication. In this step, it was also created an interactive dashboard with the main graphs that describe the information collected from the datasets.

**ML analysis**: In the ML part, it was performed sentiment analysis on a curated dataset collected from Twitter’s platform with recent user’s comments about the agriculture topic. More specifically, the sentiment of the tweets was extracted, and classification models were evaluated on these data. Moreover, forecasting analysis was performed on the crops and livestock import/export dataset, where different models were tested.

The EDA steps were implemented in the accompanying Jupyter notebook called **DataPrepVis**. The statistical logic and ML analysis can be found in the **Statistics** and **ML** Jupyter notebooks, respectively.

The source code was mainly implemented in the Jupyter notebooks, where each one has its own set of auxiliary functions. However, the following Python modules and helpers were also developed under the *jupyter/modules* folder:

**TextProcessor:** It contains the text processing methods used in this project.

**JsonHelper**: It was used to convert dictionaries into JSON format.

**Constants**: It contains all the shared constants used by all notebooks.

The reason to separate these modules from the Jupyter notebooks was to keep the code organised and to follow the best programming practices with regards to reuse and code modularization.

The source code and files used in this project are hosted on GitHub under a public organization called [CCT-MastersDA](https://github.com/CCT-MastersDA). The project’s repository is called [cct-ca2](https://github.com/CCT-MastersDA/cct-ca2), which can be accessed with the following command in any terminal: *git clone* [*https://github.com/CCT-MastersDA/cct-ca2.git*](https://github.com/CCT-MastersDA/cct-ca2.git).

In this project, an Excel file was used to organize the tasks and project requirements due to its simplicity. This way, the *CA2-Planning.xlsx* file under the *project-mngmt* folder was used to keep track of the deliverables and mark the items as completed using checkboxes.

# Data Preparation and Visualization

Finally, a dashboard called **DataPrepVisDashboard** was also created using the Voila tool containing the static and interactive graphs created in the EDA steps. The previously extracted HTML version of the dashboard can be found in the *jupyter/dashboard* folder.

# Distributed Data Processing and NoSQL Storage

# Sentiment Analysis and Time Series Forecast

The analysis described in this section was implemented and discussed in more detail in the accompanying Jupyter notebook called **ML**.

In section 5.1, it is discussed the sentiment analysis performed on the tweet’s dataset about agriculture. In section 5.2, the ML models used to predict the import and export of agriculture products in Ireland and other countries are presented.

Using the dataset 4, the sentiment analysis described in this section focuses on the classification of user’s comments about agriculture, extracted from the Twitter platform, into negative, neutral, or positive categories. This analysis was implemented in the Section 3 of the **ML** Jupyter notebook.

## Experiment Setup

As described in the Section 6 of the **DataPrepVis** Jupyter notebook, four versions of the tweets were generated from the raw text: (1) cleaned with stop words, (2) cleaned without stop words, (3) *lemmatized,* and (4) *stemmerized* tweets. Stop words do not add much information to the text, so their frequency could bias the models, therefore, they are usually removed from the dataset. *Lemmatization* and *Porter Stemmer* are common ways to extract the core meaning from the words, this way these techniques were applied to the datasets to evaluate their impact in the performance of the classifiers.

This dataset was collected from a live platform, so it was likely to be unbalanced in terms of the distribution of the sentiment classes, which was confirmed during the experiments, as shown in Section 3.1.2 of the **ML** Jupyter notebook. Therefore, SMOTE technique was used to oversample the dataset, turning it into a balanced one, because it is known that unbalanced data can led to a poor classification performance.

The tweets dataset cannot be fed directly to the ML algorithms themselves as most of them expect numerical feature vectors with a fixed size rather than the raw text documents with variable length (Scikit-learn, 2022). As a result, The TF-IDF and Count vectorizers techniques were used to extract feature vectors from the tweets, generating the independent variables used for the classification. In this case, the target variable was the sentiment.

Every feature in the vectorized tweets can be treated as independent and makes equal contribution to the result, this way the Naïve Bayes (NB) model was used. On the other hand, Logistic Regression (LGR) algorithm was also tested due to its efficiency in predicting classes based on the features relationships.

## Results

# Conclusions

In this project, it was discussed several ML and statistical methods to acquire insights from the datasets. The main challenges faced in this project was the number of datasets needed to attend the brief. Each dataset had different characteristics and needed specific EDA discipline. Multiple datasets also complicated the organization of the Jupyter notebooks that had to be split to avoid rework and code duplication, while keeping good programming practices.

It was also difficult think of all different questions in order to apply the required techniques. In the statistical part, seeing the aggregated data graphs was very helpful to guide the discussions on the results of the inferential tests applied. For the ML, several datasets had to be extracted from the import/export dataset, which made the analysis complex and hard to explain. This problem could be simplified by focusing on a single aspect of the dataset (e.g., Import quantity in Ireland) not trying to model the whole dataset.

It was also very laborious to keep several Jupyter files and the report synchronized so the references to the code from this report are correct. Knowing which content to keep in each document was another challenge.

# References

Chapman, P. *et al.* (2000) *CRISP-DM 1.0 Step-by-step data mining guide*. Available at: http://www.crisp-dm.org/CRISPWP-0800.pdf.

Eurostat (2022) *Facts and figures on life in the European Union*, *Eurostat*. Available at: https://european-union.europa.eu/principles-countries-history/key-facts-and-figures/life-eu\_en (Accessed: 24 December 2022).

Scikit-learn (2022) *Scikit-learn: Machine Learning in Python*, *https://scikit-learn.org*.