Configuration Manual

MSc Research Project
MSc in Data Analytics

Giorgia Luzia Pscheidt
Student ID: x22184261

School of Computing
National College of Ireland

Supervisor: Athanasios Staikopoulos

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Giorgia Luzia Pscheidt		
Student ID:	x22184261		
Programme:	MSc in Data Analytics	**Year:**	2023
Module:	MSc Research Project		
Lecturer:	Athanasios Staikopoulos		
Submission Due Date:	14/12/2023		
Project Title:	Sentiment Analysis of Anti-LGBTQ+ laws in Brazil using Comparative Analysis Models		
Word Count:	830	**Page Count:**	7

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……

Date: ……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Giorgia Luzia Pscheidt
Student ID: x22184261

1 Introduction

In this configuration manual, it is explained in detail the execution, procedure and information about the system requirements to run the codes, library versions and the storage capacity needed for the project: "Sentiment Analysis of Anti-LGBTQ+ laws in Brazil using Comparative Analysis Models".

2 Local Machine Details

In Figure 1, you can see the processor and RAM meet the requirements suggested by the college as an ideal device for this master.

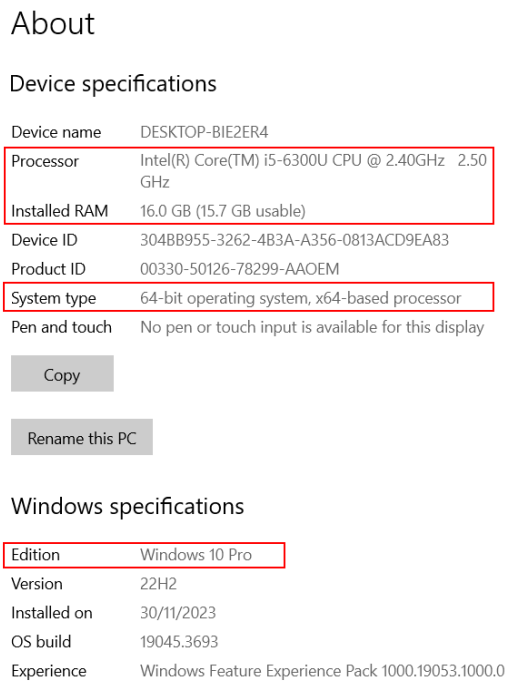
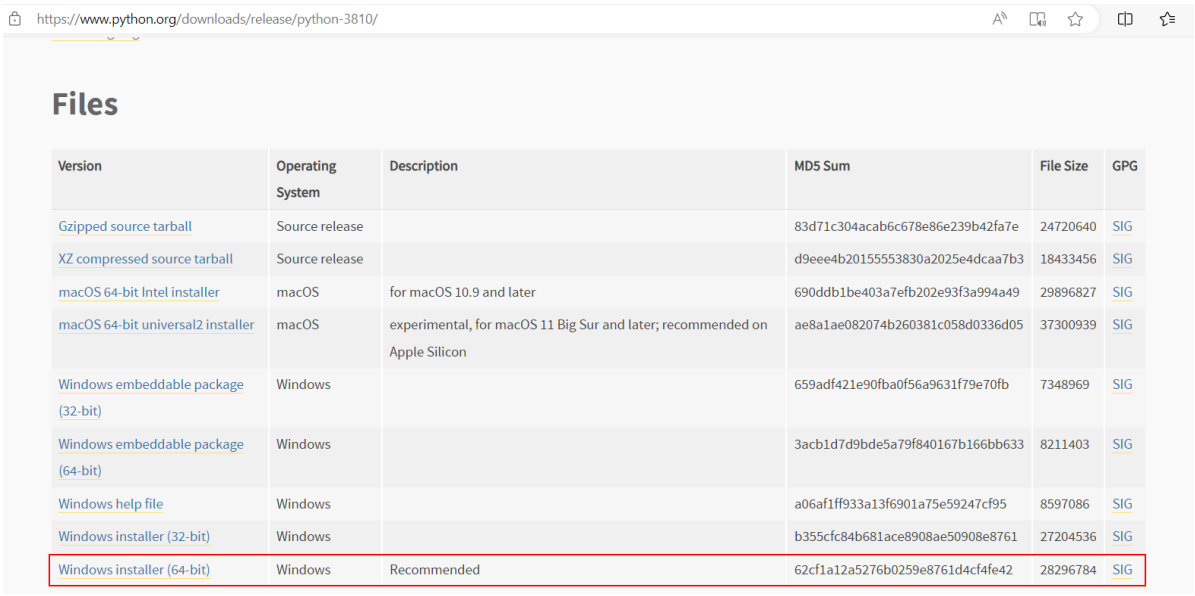


Figure 1 - Local Machine Settings

3 Comment Extraction

In this section you will find the configuration necessary to perform the comments extraction from YouTube.

- Download Python 3.8[1] and select the version based on your laptop configuration (Figure 2).



https://www.python.org/downloads/release/python-3810/

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		83d71c304acab6c678e86e239b42fa7e	24720640	SIG
XZ compressed source tarball	Source release		d9eee4b20155553830a2025e4dcaa7b3	18433456	SIG
macOS 64-bit Intel installer	macOS	for macOS 10.9 and later	690ddb1be403a7efb202e93f3a994a49	29896827	SIG
macOS 64-bit universal2 installer	macOS	experimental, for macOS 11 Big Sur and later; recommended on Apple Silicon	ae8a1ae082074b260381c058d0336d05	37300939	SIG
Windows embeddable package (32-bit)	Windows		659adf421e90fba0f56a9631f79e70fb	7348969	SIG
Windows embeddable package (64-bit)	Windows		3acb1d7d9bde5a79f840167b166bb633	8211403	SIG
Windows help file	Windows		a06af1ff933a13f6901a75e59247cf95	8597086	SIG
Windows installer (32-bit)	Windows		b355cfc84b681ace8908ae50908e8761	27204536	SIG
Windows installer (64-bit)	Windows	Recommended	62cf1a12a5276b0259e8761d4cf4fe42	28296784	SIG

Figure 2 - Python 3.8

- Install Chrome Browser[2]
- After installing Chrome Browser, make sure to check which version you have installed: open Chrome → Click in the 3 dots in the top right corner → Go to "Help" → Select "About Google Chrome" (Figure 3).

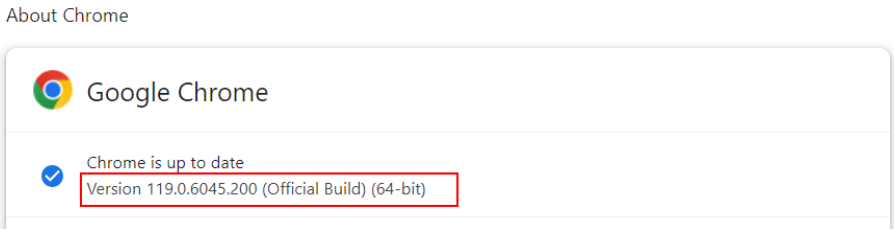


Figure 3 - Google Chrome version

- Installing ChromeDriver[3] for version newer than 155 (Figure 4).
- Run the link that follows your settings. This will download the zip file containing the application.

[1] https://www.python.org/downloads/release/python-3810/
[2] Google Chrome - The Fast & Secure Web Browser Built to be Yours
[3] ChromeDriver - WebDriver for Chrome - Downloads (chromium.org)

Stable

Version: 119.0.6045.105 (r1204232)

Binary	Platform	URL	HTTP status
chrome	linux64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/119.0.6045.105/linux64/chrome-linux64.zip	200
chrome	mac-arm64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/119.0.6045.105/mac-arm64/chrome-mac-arm64.zip	200
chrome	mac-x64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/119.0.6045.105/mac-x64/chrome-mac-x64.zip	200
chrome	win32	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/119.0.6045.105/win32/chrome-win32.zip	200
chrome	win64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/119.0.6045.105/win64/chrome-win64.zip	200
chromedriver	linux64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/119.0.6045.105/linux64/chromedriver-linux64.zip	200
chromedriver	mac-arm64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/119.0.6045.105/mac-arm64/chromedriver-mac-arm64.zip	200
chromedriver	mac-x64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/119.0.6045.105/mac-x64/chromedriver-mac-x64.zip	200
chromedriver	win32	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/119.0.6045.105/win32/chromedriver-win32.zip	200
chromedriver	win64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/119.0.6045.105/win64/chromedriver-win64.zip	200

Figure 4 – ChromeDriver

- Copy the file selected and replace in the archive downloaded from GitHub in the path: "…\Thesis\Comments Extraction".
- After you replaced the file, open the command prompt in this folder and run the command: ***pip install -r requirements.txt*** (Figure 5). This step will download the libraries necessary to run the python file, such as rich (version 13.6.0)[4], scrapy (version 2.7.1)[5], selenium (version 4.0.0)[6] and webdriver_manager (version 3.8.5)[7]

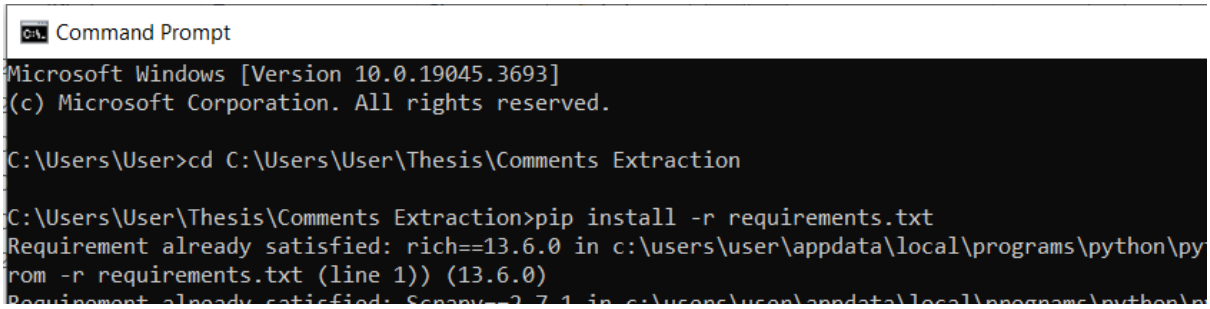


Figure 5 - Installing requirements

- Running the python file: Right click in the "youtube_comment" file → Go to "Edit with IDLE → Click in "Edit with IDLE 3.8 (64-bit) (Figure 6).

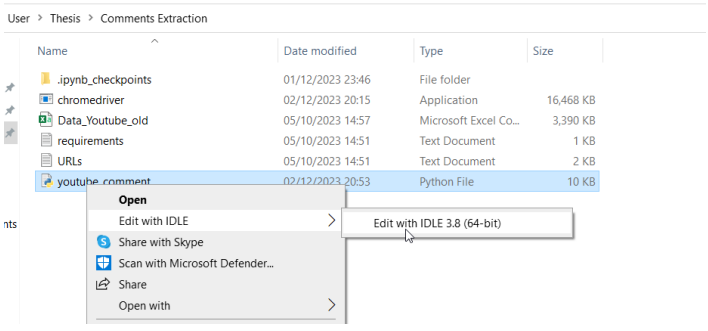


Figure 6 - Running py. File

[4] rich · PyPI
[5] Scrapy · PyPI
[6] selenium · PyPI
[7] webdriver-manager · PyPI

- Once the file opens, go to Run → Run Module (Figure 7).

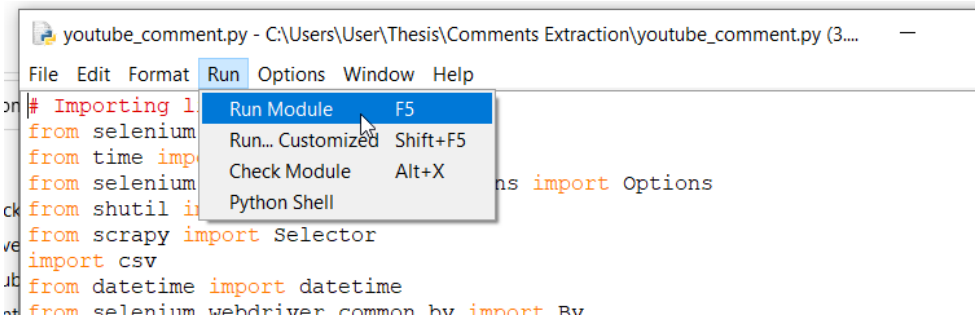


Figure 7 - Running py. file (part2)

- Once the python file starts running, chrome browser will open in youtube.com in the links provided for the extraction. It will automatically collect the comments and save them in a csv file. Note this process will take time since it is collecting data from 44 URLs, for me took 6 hours.

4 Data cleaning, Translation and Sentiment Analysis

Here I will explain which applications are needed to run the rest of the code.

- Anaconda Version 23.3.1[8]
- Jupyter Lab Version 1.23.4[9]

4.1 Data Cleaning

For the data cleaning process, access the file in the Jupyter lab: **cleaning_raw_data.ipynb**. For this file you will need the following libraries: pandas (version 1.5.2)[10], re (version 2.2.1)[11], collections[12] and spellchecker (version 0.7.2)[13].
Then you can just restart the kernel and run all the cells (Figure 8).

[8] Free Download | Anaconda
[9] Project Jupyter | Installing Jupyter
[10] pandas - Python Data Analysis Library (pydata.org)
[11] re — Regular expression operations — Python 3.12.1 documentation
[12] collections — Container datatypes — Python 3.12.1 documentation
[13] pyspellchecker · PyPI

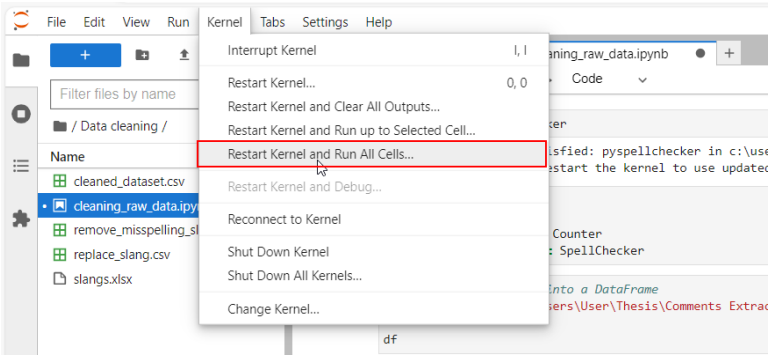


Figure 8 - Running Data Cleaning

4.2 Translation

After completing to run this code, go to the folder Translation and open the file **translating.ipynb**.
Before running this code, be aware that for this step it was used the Google Cloud Translation API[14], which you will need to create your own account and replace this file: **googleapi_key.json** which will contain your credentials to use this tool.
The libraries used in this step were: os (version 10.0.19)[15], google.cloud (version 3.12.1)[16], wordcloud (version 1.9.2)[17] and matplotlib (version 3.6.2)[18].
Then you can just restart the kernel and run all the cells (Figure 8). Note that this process will take time to finish, between 1 to 2 hours.

4.3 Sentiment Analysis

After completing to run the translation, go to the folder Sentiment Analysis and open the file **SA1.ipynb**. Here it is the first algorithm performed, with used Sentiment Intensity Analyzer (SIA) and VADER (Valence Aware Dictionary and Sentiment Reasoner) to make the text classification without labels (Vencer, Bansa, & Caballero, 2023). In Figure 9, those might be some necessary installations to perform this task.
In this code, it was used scattertext (version 0.1.19)[19], nltk (version 3.7)[20], numpy (version 1.23.5)[21], scikit-learn (version 1.0.2)[22] and seaborn (0.12.2)[23].
Then you can just restart the kernel and run all the cells (Figure 8).

[14] Cloud Translation API | Google Cloud
[15] os — Miscellaneous operating system interfaces — Python 3.12.1 documentation
[16] Python Cloud Client Libraries | Google Cloud
[17] wordcloud · PyPI
[18] matplotlib · PyPI
[19] scattertext · PyPI
[20] NLTK :: Natural Language Toolkit
[21] NumPy Documentation
[22] scikit-learn: machine learning in Python — scikit-learn 1.3.2 documentation
[23] seaborn: statistical data visualization — seaborn 0.13.0 documentation (pydata.org)

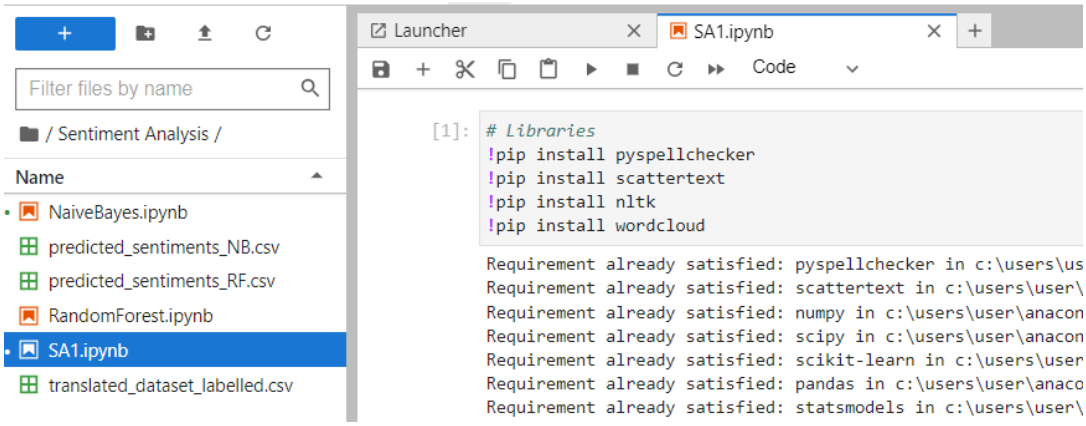


Figure 9 - NLTK process

4.4 Random Forest

Still in the same folder, open the file RandomForest.ipynb. In this code the first machine learning[24] is performed using the labelled dataset created manually by the author. Figure 10 shows the model being trained. You can just restart the kernel and run all the cells (Figure 8).

```
# Preprocess and split your dataset into training and testing sets
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
import pandas as pd

# Load your labeled dataset
labeled_df = pd.read_csv('translated_dataset_labelled.csv')

# Split the dataset into training and testing sets
train_data, test_data, train_labels, test_labels = train_test_split(
    labeled_df['translated_comment'], labeled_df['label'], test_size=0.2, random_state=42
)

# Preprocess the text data
vectorizer = CountVectorizer()
X_train = vectorizer.fit_transform(train_data)
X_test = vectorizer.transform(test_data)

# Train a Random Forest Classifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
rf_classifier.fit(X_train, train_labels)

# Predict sentiment labels for the test set
predicted_labels = rf_classifier.predict(X_test)

# Evaluate the accuracy of the model
accuracy = accuracy_score(test_labels, predicted_labels)
print(f'Accuracy: {accuracy}')
```

Figure 10 - Random Forest

[24] sklearn.ensemble.RandomForestClassifier — scikit-learn 1.3.2 documentation

4.5 Naïve Bayes

Still in the same folder, open the file NaiveBayes.ipynb. In this code the second machine learning[25] is performed using the labelled dataset created manually by the author. Figure 11 shows the model being trained. You can just restart the kernel and run all the cells (Figure 8).

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

# Load labeled dataset
labeled_df = pd.read_csv('translated_dataset_labelled.csv')

# Split the dataset into training and testing sets
train_data, test_data, train_labels, test_labels = train_test_split(
    labeled_df['translated_comment'], labeled_df['label'], test_size=0.2, random_state=42
)

# Preprocess the text data using CountVectorizer
vectorizer = CountVectorizer()
X_train = vectorizer.fit_transform(train_data)
X_test = vectorizer.transform(test_data)

# Train a Multinomial Naive Bayes classifier
nb_classifier = MultinomialNB()
nb_classifier.fit(X_train, train_labels)

# Predict sentiment labels for the test set
predicted_labels = nb_classifier.predict(X_test)

# Evaluate the accuracy of the model
accuracy = accuracy_score(test_labels, predicted_labels)
print(f'Accuracy: {accuracy}')

# Display confusion matrix and classification report
cm = confusion_matrix(test_labels, predicted_labels)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=labeled_df['label'].unique(), yticklabels=labeled_df['label'].unique())
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()

print(classification_report(test_labels, predicted_labels))
```

Figure 11 - Naive Bayes

References

Vencer, L. V., Bansa, H., & Caballero, A. R. (2023). Data and Sentiment Analysis of Monkeypox Tweets using Natural Language Toolkit (NLTK). *2023 8th International Conference on Business and Industrial Research (ICBIR)* (pp. 392--396). IEEE.

[25] 1.9. Naive Bayes — scikit-learn 1.3.2 documentation