



## Documentation

# Hack The Future 2020 @ XploData

Jean-Joseph Adjizian, Reinert Roux & Peter Boschmans, XploData

November 24<sup>th</sup>, 2020

## Table of Contents

1. Introduction .....	3
1.1. Hack the future .....	3
1.2. XploData .....	3
1.3. The challenge .....	4
1.4. Skills .....	4
2. Tools .....	4
2.1. Python .....	4
2.1.1. Installation: .....	4
2.1.2. Libraries .....	5
2.2. NLP .....	5
2.3. Web Application .....	5
• Streamlit: .....	5
• Flask: .....	5
3. To get started: .....	6
4. Further Reading .....	6
4.1. Scraping: .....	6
4.2. Natural language processing: .....	6
4.3. Streamlit: .....	6
4.4. Examples: .....	6
5. Demo: .....	7

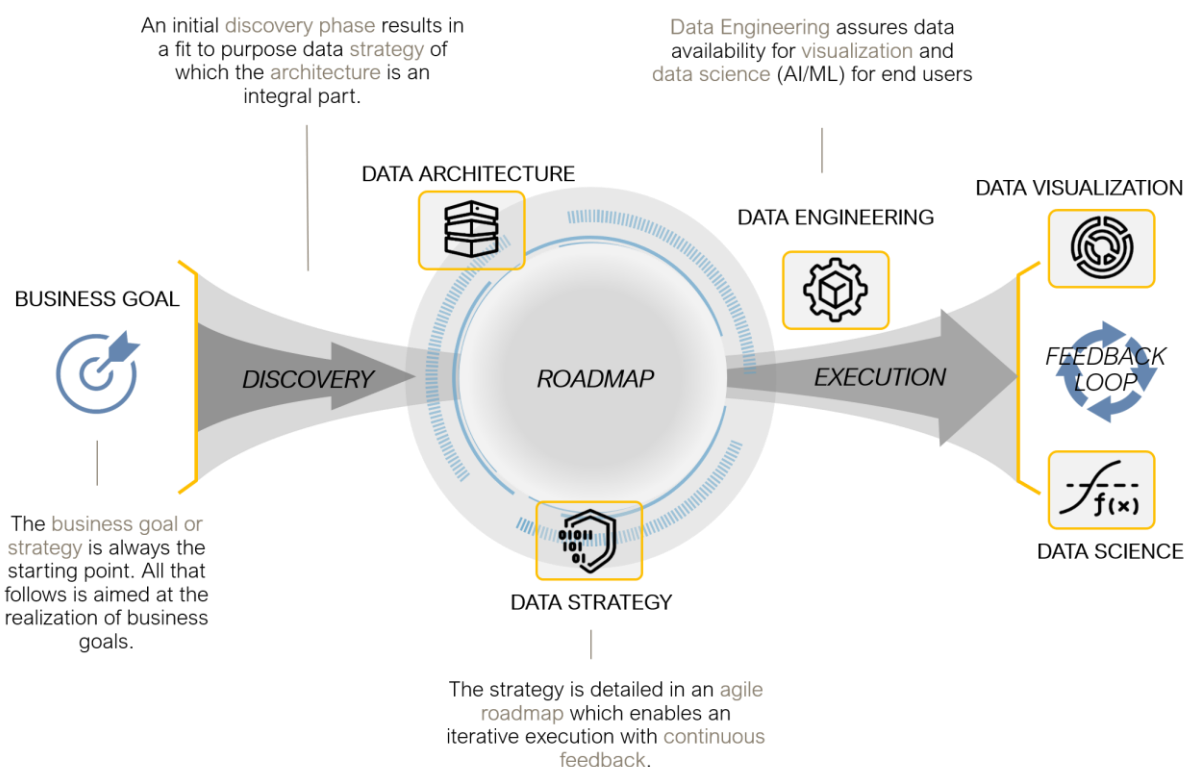
# 1. Introduction

## 1.1. Hack the future

First welcome to Hack the Future 2020. We, at XploData, are very happy that you are taking part of this adventure. Remember, this day should be all about fun and learning.

## 1.2. XploData

XploData (part of the Cronos Group) has been helping its customers to make better decisions faster with the help of data since 2007. We guide and support our customers on their data journey. We have 5 business units (see figure) that ensure a smooth end-to-end approach: from strategy to implementation and support afterwards. At XploData we put our people first. Our mission is to help our customers maximize the impact and value of their data by building a stable and lasting relationship with our people. In this way we ensure the development of data-driven organizations and the realization of our clients' strategy.



### 1.3. The challenge

The Twitosphere is going crazy over our latest product and we need to help our center to find some clarity in this twitter mess. Are you not afraid to scrap the web? Are you not afraid to extract people sentiments from their tweets? Do you want to kick some bots that create fake tweets out? If yes, then be ready to apply the latest models in Natural Language Processing (NLP) to help the people responsible for our twitter account to find some clarity in this twitter storm.

You will learn how you can apply and develop a machine learning application in only a few single steps. Your analysis and coding skills will be needed.

At the end the challenge is “plain and simple”: Code an application that can retrieve tweets, analyses them using NLP, and show the outcome into a web interface.

Each team has flexibility depending on skills and time. In other words, you can make your challenge easy as well as complicated. The more creative you will be, the more challenging it will be. Remember, the goal is also to have fun.

In this project, there are different phases involved that you would encounter during a project with a customer in real life (preparation, execution, test, put in production) and you’ll learn how to face those. The end goal of this challenge is also to get inspired by the potential that machine learning can have if you build a good case with it.

### 1.4. Skills

Knowledge of coding and understanding of how to build an application to apply machine learning.

## 2. Tools

Each participant can choose their own tools for the challenge. However, we recommend using some of the tools below to facilitate the execution of the challenge in time. Also **remember**, we are here to help, so if you have questions please do not hesitate to ask.

### 2.1. Python

#### 2.1.1. Installation:

<https://www.python.org/downloads/>

and follow the instructions: <https://docs.python.org/3/using/index.html>

### 2.1.2. Libraries

To install libraries with python you can use the command 'pip install'. If you are using anaconda, you can either use the anaconda GUI or the conda command.

- *Twint:*  
Twint is an advanced Twitter scraping tool written in Python that allows for scraping Tweets from Twitter profiles **without** using Twitter's API.  
<https://github.com/twintproject/twint>
- *Pandas:*  
**Pandas** is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool.  
<https://pandas.pydata.org/>

## 2.2. NLP

There are many python libraries that help do NLP. Below you can find a list of them:

- Spacy: <https://spacy.io/>
- NLTK: <https://www.nltk.org/>
- Transformers: <https://huggingface.co/transformers/>
- TextBlob: <https://textblob.readthedocs.io/en/dev/>
- Scikit-learn: <https://scikit-learn.org/stable/>
- ...

## 2.3. Web Application

There are also many ways to create a web application, and we only list here two of them.

- Streamlit:  
Turn data scripts into sharable web apps in minutes. All in Python. All for free. No front-end experience required.  
<https://www.streamlit.io/>
- Flask:  
Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications.

<https://palletsprojects.com/p/flask/>

### 3. To get started:

If you are a bit in the blue in the challenge, here are some ideas that you could work on and modify to your liking.

The Twint library allows you to extract valuable information, not only about people sayings, but also for example about their location, their followers, who are they following, etc, and this can be important information you can work on within your web interface. You can find more details about Twint capabilities following this [link](#).

Using an NLP library there also many exciting applications that you could choose from: Sentiment Analysis, Text Classification, Text Extraction, Machine Translation, Text Summarization, Market Intelligence, Auto-Correct, Intent Classification, Urgency Detection. You will find some code example in the links below in part 4.

### 4. Further Reading

There is a good chance that you will find all your answers on the web, sometimes information is harder to dig out though. Below you can find a few links that can get be of interest and get you started.

#### 4.1. Scraping:

- <https://realpython.com/beautiful-soup-web-scraper-python/>

#### 4.2. Natural language processing:

- <https://sunscrapers.com/blog/8-best-python-natural-language-processing-nlp-libraries/>

#### 4.3. Streamlit:

- <https://www.streamlit.io/gallery>
- <https://github.com/MarcSkovMadsen/awesome-streamlit> (App paragraph)

#### 4.4. Examples:

- Tweet EDA: <https://github.com/cotraak/trump-tweets-eda>
- Twitter Analytics: <https://towardsdatascience.com/simple-twitter-analytics-with-twitter-nlp-toolkit-7d7d79bf2535>
- Fake tweets: <https://becominghuman.ai/real-vs-fake-tweet-detection-using-a-bert-transformer-model-in-few-lines-of-code-ccc33ecb1a2>

- Bot detection: <https://blog.quantinsti.com/detecting-bots-twitter-botometer/>
- Fake News: <https://www.cvpcorp.com/detecting-fake-news-with-a-bert-model/>

## 5. Demo:

You can find our simple demo and code using this link: