

PROJET X-AE-A12

Détection de FakeNews sur Twitter

*Alexandre Baux, Sohel Dinnoo,
Mathis Rouget, Vianney Saint Georges-Chaumet,
Zein Sakkour, Philémon Wehbe*



SOMMAIRE

01

Besoins et MVP

02

Structure du projet

03

Organisation

04

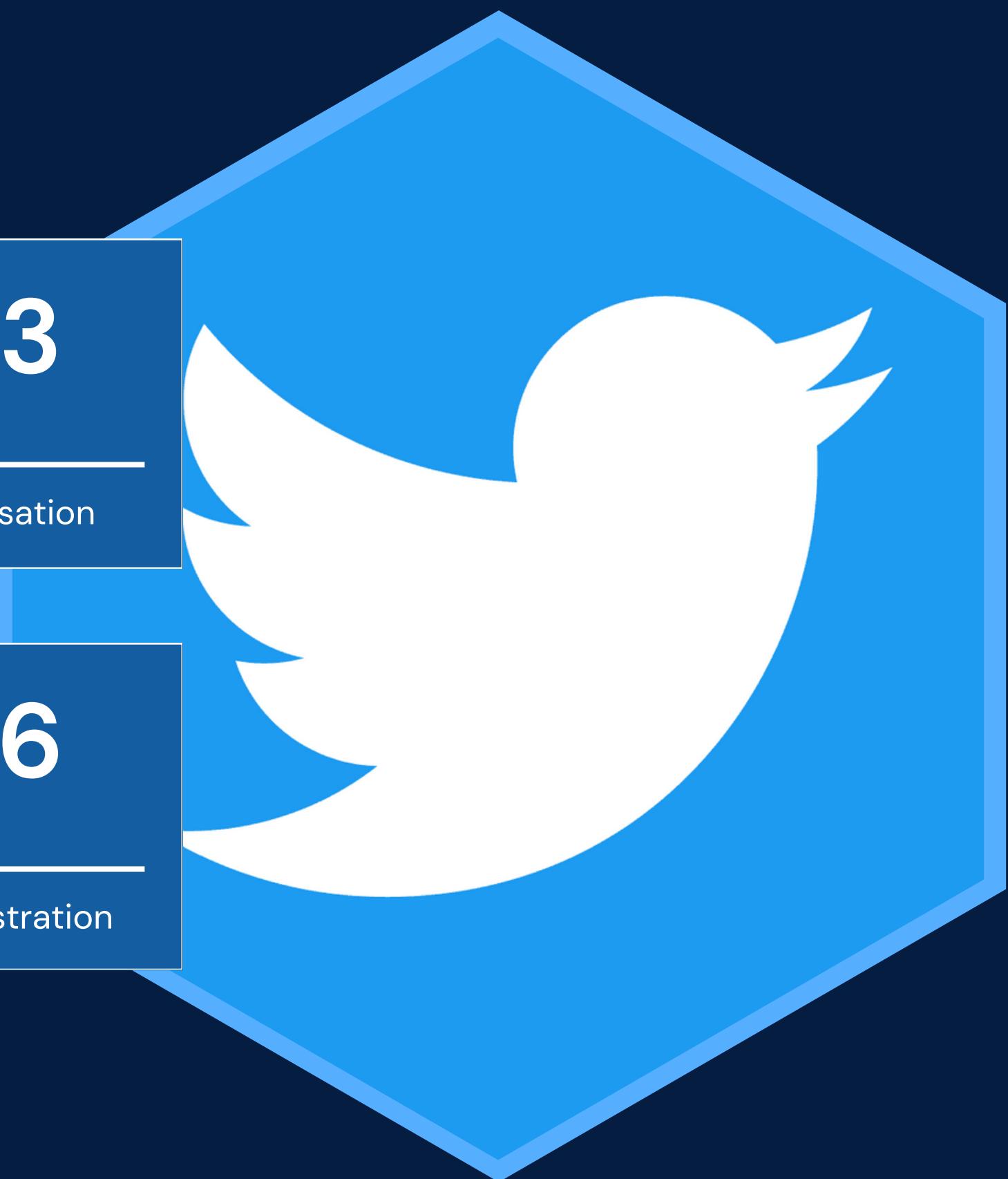
Tests

05

Documentation

06

Démonstration





IDENTIFICATION DES BESOINS MVP

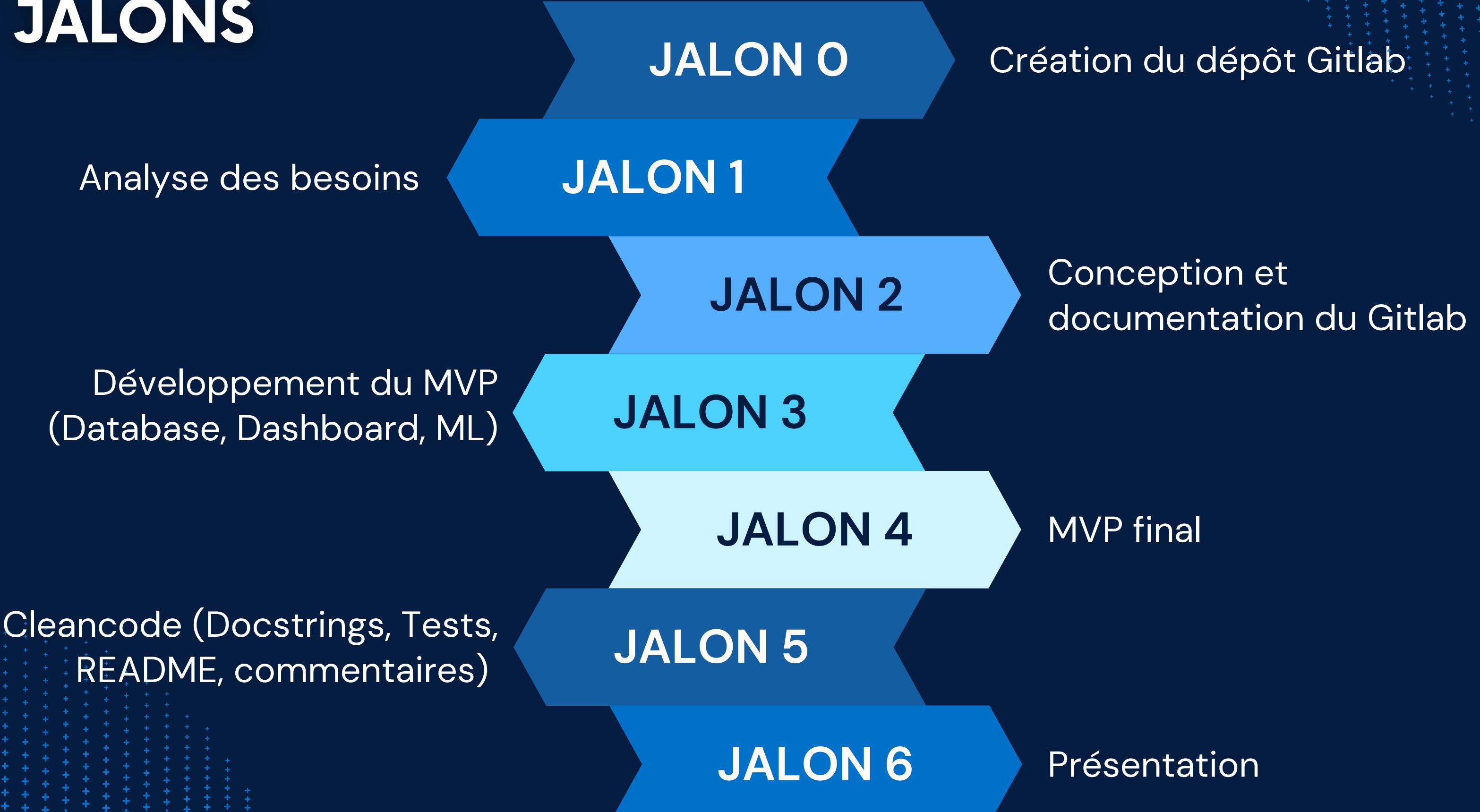
Analyse des besoins

En tant qu'entreprise ou personnalité publique, je dois pouvoir me faire une idée de la désinformation qui circule à mon égard.

Minimum Viable Product

Élaboration d'un tableau de bord interactif permettant de visualiser des statistiques sur les FakeNews d'un ensemble de tweets.

JALONS



SPRINTS

SPRINT 1

Collecte de données par le scraper

SPRINT 2

Mise en forme des données avec Pandas

SPRINT 3

Implémentation du Machine Learning

SPRINT 4

Création du graphe avec Cytoscape

SPRINT 5

Affichage global grâce au Dashboard

PROGRESSION DU MVP

Dashboard classique
avec graphiques
simples sur des
tweets d'une
database donnée

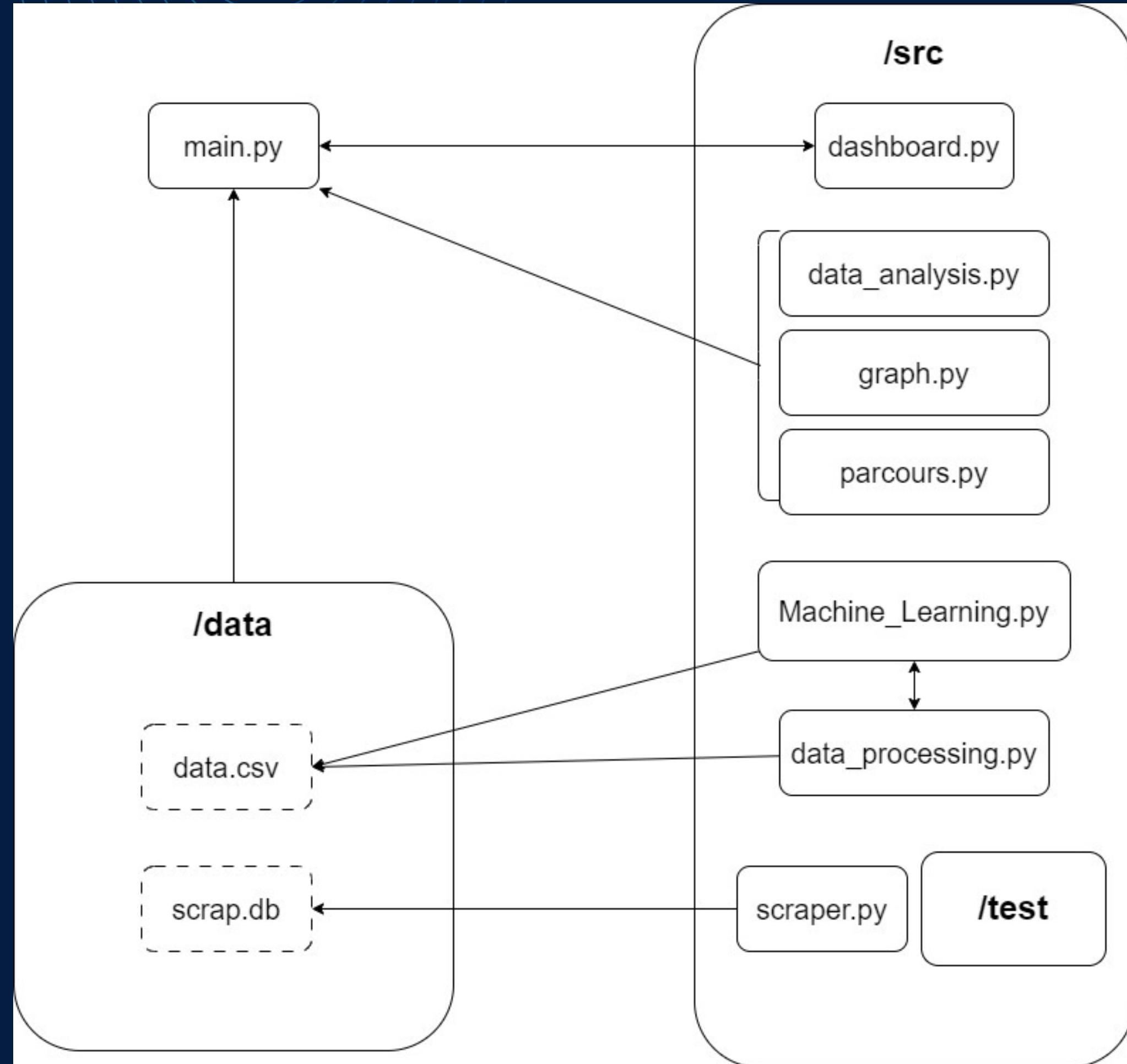
Implémentation du
Machine Learning
pour détecter les
FakeNews et
validation avec la
base de données
LIAR

Collecte de données
avec un scraper,
mise en place d'une
base de données,
analyse

Amélioration du
Dashboard avec des
graphes plus évolués

Architecture du code

- Séparation du code en différents fichiers qui interagissent entre eux, articulés autour de main.py
- Architecture arborescente (code, sources, data, ...)



RÉPARTITION DES FONCTIONNALITÉS

DASHBOARD

Alexandre

MACHINE LEARNING

Zein
Philémon

TRAITEMENT DES DONNÉES

Sohel

GRAPHES ET ANALYSE DES DONNÉES

Vianney
Mathis

DÉMONSTRATION

- README
- Tests (Gitlab CI Pipeline en ligne)
- Main Loop
- Structuration à gros et petits grains du code



```
    = None
fingerprint = set()
duplicates = True
debug = debug
logger = logging.getLogger()
path:
self.file = open(os.path.join(job_dir, 'fingerprint'), 'w')
self.file.seek(0)
self.fingerprints = set()

@ssmethod
def from_settings(cls, settings):
    debug = settings.getboolean('debug', False)
    return cls(job_dir=job_dir, debug=debug)

def request_seen(self, fp):
    if fp in self.fingerprints:
        return True
    self.fingerprints.add(fp)
    if self.file:
        self.file.write(fp + '\n')

def request_fingerprint(self, request):
    return request['fp']
```

AMÉLIORATIONS POSSIBLES

- Autofill sur la barre de recherche
- Confondre majuscules et minuscules dans l'username
- Implémentation d'un réseau de neurones

MERCI

