ISEN ALL IS DIGITAL! OUEST | yncréa

Projet Année 3
Big Data/IA/Web

Partie Intelligence Artificielle

Projet encadré par :

- Ayoub KARINE <ayoub.karine@isen-ouest.yncrea.fr>
- Zakaria ABOU EL HOUDA <zakaria.abou-el-houda@isen-ouest.yncrea.fr>

Contexte du projet

Sujet

Objectif général

Concevoir et développer une application d'étude des accidents de la route

Approfondir les compétences acquises dans les modules *Big Data, Intelligence Artificielle, Développement Web et Base de Données* à travers une application complète de traitements et de visualisation de données concernant les accidents corporels de la circulation routière en France.

Sujet

Objectif général

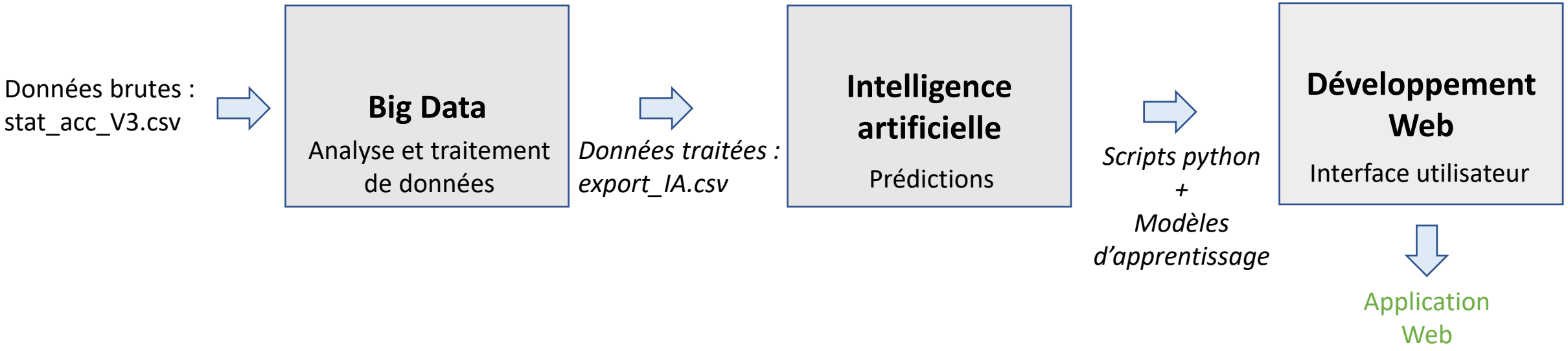
Concevoir et développer une application d'étude des accidents de la route

Approfondir les compétences acquises dans les modules *Big Data, Intelligence Artificielle, Développement Web et Base de Données* à travers une application complète de traitements et de visualisation de données concernant les accidents corporels de la circulation routière en France.

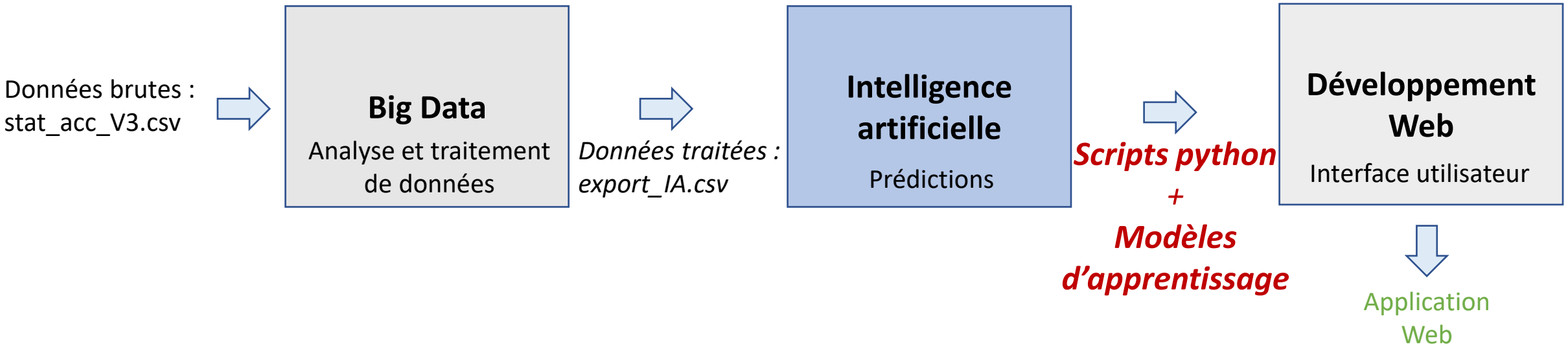
Objectifs de la partie Intelligence Artificielle :

- Comprendre les étapes d'un projet d'apprentissage automatique
- Apprendre à évaluer une méthode d'apprentissage automatique
- Manipuler les différentes techniques d'apprentissage automatique :
  - Apprentissage non supervisé : réduction de dimensionnalité, clustering
  - Apprentissage supervisé : régression, classification

Déroulement du projet



Déroulement du projet



Cahier des charges

Cahier des charges

Fonctionnalités

5 fonctionnalités principales sont attendues :

1. Découverte et préparation des données
2. Apprentissage non-supervisé
3. Apprentissage supervisé
4. Métriques
5. Création de scripts utilisables en ligne de commande

Attention

Les fonctionnalités additionnelles que vous développerez seront évaluées uniquement si les fonctionnalités principales sont en place.

Fonctionnalité 1 : Découverte et préparation des données

Préparation de l'environnement de travail et gestion de projet :

Outils de développement, installation des packages, Gantt…

→ **Attendus : diagramme de Gantt**

Organisation des données :

CSV → Pandas

Découverte des données :

- Valeur cible (gravité de l'accident) pour la classification = classe
- Nombre d'instances
- Nombre d'instances par classe
- Taille des features

→ **Attendus : graphiques de visualisation avec leur interprétation**

Préparation des données :

- Déjà réalisé en big data :
  - Conversion des valeurs non-numériques
  - …
- Conversion des dates et heures

→ **Attendus : fonctions de préparation des données avec leur explication**

Fonctionnalité 2 : Apprentissage non-supervisé

Réduction de dimension :

En utilisant deux méthodes différentes :

- Corrélation entre attributs et suppression manuelle de quelques features
- Utilisation de PCA avec la bibliothèque *scikit-learn* (en bonus)

→ **Attendus : graphiques de visualisation avant/après, pourcentage de réduction avec les interprétations**

Partitionnement (clustering) :

Clustering sur la latitude et la longitude en utilisant *k-means* selon les deux procédés suivants :

- *k-means* « *from scratch* » avec :
  - Des distances spécifiques (L1, L2, Haversine…)
  - Un nombre de clusters qui varie
- *k-means* en utilisant les fonctions mises à disposition dans la bibliothèque *scikit-learn* avec :
  - Un nombre de clusters qui varie

→ **Attendus : graphiques de visualisation des *clusters*, des centroïdes avec les 2 procédés et avec N *clusters* (où N varie) , comparatif entre les 2 procédés avec les interprétations**

Il est fortement conseillé d'utiliser la bibliothèque *plotly* : https://plotly.com/python/ pour l'affichage des données sur une carte.

Évaluation quantitative des résultats « non supervisé » :

Évaluation des performances de la méthode de *clustering k-means* en utilisant les métriques suivantes :

- Silhouette Coefficient
- Calinski-Harabasz Index
- Davies-Bouldin Index

→ **Attendus : tableaux et graphiques, étude comparative**

Fonctionnalité 3 : Apprentissage supervisé

Répartition des données :

Répartition des données en base d'apprentissage et de tests en utilisant les algorithmes *holdout (répété 5 fois)* et *leave-one-out* en utilisant les fonctions mises à disposition dans la bibliothèque *scikit-learn (Bonus :* réalisation « *from scratch* »)

→ **Attendus : fonctions de création des jeux de données, comparatif entre les deux algorithmes avec les interprétations**

Classification avec KNN :

Utilisation de la méthode *K-Nearest Neighbors* pour *r*echercher les accidents les plus similaires *+* prédiction de la gravité de l'accident selon les deux procédés suivants :

- « *from scratch* »
- en utilisant les fonctions mises à disposition dans la bibliothèque *scikit-learn*

→ **Attendus : résultats de classification, comparatif entre les 2 procédés, variation de K et de plusieurs distances avec les interprétations**

Classification avec trois algorithmes de « haut niveau » :

Prédiction de la gravité d'un accident selon les 3 méthodes suivantes (en utilisant *scikit-learn*) :

- *Support Vector Machine* (SVM)
- *Random Forest*
- *Multilayer Perceptron* (MLP)

Pour chaque méthode, recherche des valeurs optimales en utilisant *GridSearch* et enregistrement des modèles entrainés ayant les meilleurs paramètres trouvés

Fusion des 3 classifieurs ci-dessus : vote majoritaire, …

→ **Attendus : résultats de classification, tableaux/graphes d'optimisation des paramètres, comparatif entre les 3 méthodes avec les interprétations, 3 modèles d'apprentissages (3 fichiers : lien utile)**

Fonctionnalité 3 : Apprentissage supervisé

Évaluation quantitative des résultats « supervisé » :

Évaluation des performances de chaque méthode de l'étape « apprentissage supervisé utilisant les méthodes de haut niveau » en utilisant les métriques suivantes :

- Taux d'apprentissage
- Matrice de confusion
- Précision/Rappel
- Courbe ROC

→ **Attendus : tableaux et graphiques, étude comparative entre les différentes approches supervisées**

Fonctionnalité 4 : Création de scripts utilisables en ligne de commande (préparation pour la partie web)

Script pour *l'apprentissage non-supervisé* :

Création d'un script pour la méthode *k-means* prenant comme paramètres d'entrées :

- La latitude et la longitude de l'accident
- Les centroides

→ **Sortie : le cluster d'appartenance de l'accident (en *JSON*)**

Script pour *l'apprentissage supervisé* :

Création d'un script pour la méthode *KNN* prenant comme paramètres d'entrées :

- Les informations de l'accident
- Le fichier CSV contenant la liste des accidents

→ **Sortie : la grabité de l'accident (en *JSON*)**

Script pour les méthodes de classification de « haut niveau » :

Création d'un script pour les méthodes de classification de « haut niveau » prenant comme paramètres d'entrées :

- Les informations de l'accident
- La méthode de classification à utiliser et les modèles de prédiction entraînés

→ **Sortie : la gravité de l'accident (en *JSON*)**

Attention

Les informations de l'accident en entrée des scripts devront être prétraitées avec la même fonction (et les mêmes paramètres) que celle utilisée pour les données d'apprentissage.

Fonctionnalité 4 : Création de scripts utilisables en ligne de commande (préparation pour la partie web)

Exemple d'utilisation de sys.argv :

```python
import sys

def somme(a, b):
    return float(a) + float(b)

s = somme(sys.argv[1], sys.argv[2])
print(s)
```

```
PS D:\OneDrive - yncréa\Enseignement\2019-x\A3\Projet_2022-2023\Tests> python somme.py 5 6
11.0
```

Il est possible d'utiliser argparse à la place de sys.argv

Livrables et évaluations

Déroulement

Organisation

Travail en trinôme :

- Chaque étudiant dans le trinôme connaît l'ensemble du projet
- Attention à bien se répartir le travail en prévoyant les tâches de chacun avec un **diagramme de Gantt**

Ressources externes :

- Tous les documents sont autorisés
- Attention à utiliser avec une grande précaution tout document extérieur : site de vulgarisation, forum, code d'autrui

Documentation du projet :

- Au fur et à mesure
- Standardisée
- Livraison de code ou de documents :
  - Ne pas attendre la dernière minute pour poster un livrable
  - Préparer des livrables intermédiaires (surtout pour les sources)
  - Sauvegarder régulièrement vos scripts et résultats

Livrable

Format de l'archive :

Archive *ZIP*, *TGZ*, *7ZIP*, pas de *RAR :* `projetia_groupeX.zip` (remplacer X par votre numéro de trinôme)

Le rendu final doit être déposé avant le 23/06 à 12h. Il doit contenir :

- L'intégralité de vos scripts Python commentés avec vos éventuelles ressources (fichier de données initiales…)
  - .ipynb (GoogleColab)
  - .py
- Présentation au format PDF + Rapport au format PDF :
  - gestion de projet (planning + qui a fait quoi)
  - attendus

Remarques :

- Malus possible sur l'un des membres du groupe si l'investissement est jugé trop faible
- Possibilité d'être interrogé durant le projet de façon individuelle
- Plagiat sévèrement sanctionné pour TOUS les membres du/des groupe(s)

Attention

Les livrables seront à poster sur l'intranet. Tout retard sera sanctionné (l'heure du réseau faisant foi).
Les fichiers au mauvais format ou avec un mauvais nommage seront pénalisés.

Évaluation

QCM le 22/06/2023 à partir de 11h40 :

- QCM de 20 minutes à points négatifs
- Évaluation individuelle

Présentation orale le 23/06/2023 à partir de 13h30 :

- Soutenance de 15 minutes (strict) + 5 minutes de questions
- Présentation en trinôme (pensez à vous répartir la parole)
- Présentez l'essentiel de votre projet

Code et rapport :

- Rendu de l'intégralité de vos codes sources
- Rapport du trinôme présentant les attendus du projet
- Aucun code ne doit apparaître dans le rapport

Barème indicatif :

- Présentation 40% -- QCM individuel 20% -- Code et Rapport 40%