# Projet chef d’oeuvre

Cerebro: un aide au diagnostic d’Alzheimer

Silvia Ciappelloni

SIMPLON.CO

Microsoft

# Sommaire

# Introduction

Selon le rapport de la Fondation pour la Recherche Médicale[1], la maladie d'Alzheimer est la forme de démence la plus courante. Plus de 50 millions de personnes en sont aujourd'hui atteintes dans le monde et l'Organisation Mondiale de la Santé prévoit même que ce nombre pourrait augmenter et atteindre les 152 millions d'ici 2050. La France compte aujourd'hui environ 900 000 malades avec une projection de 2,1 millions de malades estimés en 2040 chez les plus de 65 ans. La démence est la 1ère cause de dépendance lourde du sujet âgé et la 1ère cause d'entrée en institution atteignant environ 3 millions de personnes qui sont directement concernées par la maladie, si on inclut les aidants aussi.
Plusieurs études menées dans des pays occidentaux indiquent une diminution du nombre de nouveaux cas de démence, donc de maladie d'Alzheimer, au cours des dernières années[2] . Cette diminution est corrélée avec une meilleure prise en charge et une meilleure prévention des maladies cardiovasculaires. Cependant, du fait de l'allongement de l'espérance de vie et du vieillissement, le nombre total de cas de malades atteints de maladie d'Alzheimer devrait probablement continuer à augmenter dans les années à venir.

## Problématique

Il n'existe actuellement aucun traitement efficace contre cette maladie. Quand un radiologue analyse une radiographie, il est impossible pour lui de dire si une personne va développer la maladie. L'Alzheimer est une pathologie qui évolue différemment d'un patient à l'autre, les personnes atteintes ne présentent pas toutes les mêmes symptômes et l'aggravation peut être plus rapide chez certaines patients plutôt que chez d' autres. La première manifestation clinique de la maladie est une altération sélective de la mémoire qui peut évoluer jusqu'à arriver à un déficit cognitif léger. Ce stade appelé MCI (Mild Cognitive Impairment) est caractérisé par l'apparition des symptômes qui permettent de diagnostiquer chez certaines personnes, mais pas toutes, le stade précoce de la maladie d'Alzheimer.
L'un des outils diagnostiques les plus utilisés par les médecins pour identifier l'apparition de la maladie d'Alzheimer est l'imagerie cérébrale (IRM - Imagerie par Résonance Magnétique). En effet, les images permettent de visualiser une diminution de la taille des structures situées sous le cortex cérébral. Plus l'atrophie est détectée à un stade précoce, plus la maladie bénéficie d'une bonne prise en charge.
Actuellement, l'utilisation de l'imagerie cérébrale est principalement limitée aux études de recherche et aux essais cliniques, afin de s'assurer que des nouveaux médicaments potentiels sont testés chez les bonnes personnes. Toutefois, l'imagerie est un outil puissant, mais elle est coûteuse et nécessite des installations et une expertise spécialisées.
L'application des algorithmes d'intelligence artificielle pourrait améliorer la capacité de l'imagerie cérébrale à prédire la maladie d'Alzheimer des années avant un diagnostic réel, ce qui permettrait aux chercheurs de trouver de meilleures façons de ralentir, voire d'arrêter, le processus de la maladie. À l'heure actuelle, il n'existe aucun moyen de diagnostiquer rapidement et efficacement la maladie d'Alzheimer. L'IA pourrait donc s'avérer être un outil utile pour les radiologistes.

1 https://www.frm.org/recherches-maladies-neurologiques/maladie-d-alzheimer/alzheimer-en-chiffres
2 Caroll Brayne et al. Dementia in Western Europe: epidemiological evidence and implications for policy making. Policy view. Lancet Neurology 2015, publié en ligne le 21 août 2015.

Pour atteindre ce stade, les cliniciens et les chercheurs devront utiliser des techniques d'apprentissage automatique et profond qui peuvent prédire avec précision la progression d'un patient atteint d'une déficience cognitive légère à la démence.
Pour cela, je propose une solution reposant sur l'IA et spécifiquement sur la reconnaissance d'image, qui permettrait aux équipes soignantes de prédire les premiers stades de la maladie d'Alzheimer ainsi que sa progression, afin d'empêcher des changements irréversibles dans le cerveau.

## Analyse des besoins

Le service de Radiologie et de Neuroimagerie diagnostique et thérapeutique de l'Hôpital Pellegrin de Bordeaux souhaiterait bénéficier d'une Preuve de Concept (POC) d'une solution innovante utilisant l'IA.
L'objectif sera de livrer une application dans laquelle le médecin pourra charger l'image 3D du cerveau, obtenue à partir d'un examen en IRM, et retourner le stade de la maladie. Les données insérées dans l'application, ainsi que la prédiction du stade de la maladie, seront exportées vers un tableau suivi de patients, qui permettra d'analyser la progression de la maladie du patient.
La solution doit également être facilement adaptable afin de pouvoir prendre en compte un certain nombre de variables (données cliniques, données de vie réelle, etc..) dans le modèle prédictif, préalablement validées avec les experts du domaine.

# Gestion de projet

Le projet a été réalisé en 7 étapes principales (Fig.1).

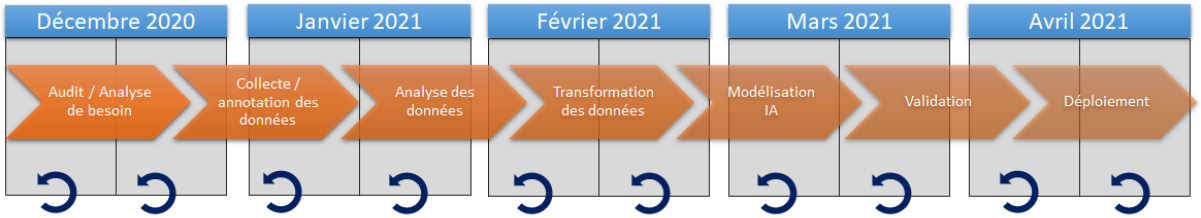



*Fig 1. Description en 7 étapes de la réalisation du projet cerebro avec la fréquence de reporting.*

- ❑ Analyse du besoin: le moment où le projet a été monté, avec l'expression du besoin du client. Dans ce cas, le service du CHU demande un outil innovant d'intelligence artificielle qui permet d'accélérer le diagnostic de la maladie d'Alzheimer ainsi que la prise en charge de la part du corps hospitalier.
- ❑ La collecte et la simulation des données. Une fois le projet identifié comme pouvant être résolu par une solution à base d'IA, la donnée a été collectée.
- ❑ L'analyse des données. Étape indispensable avant quoi que ce soit : se familiariser avec les données. C'est aussi l'occasion de compléter les connaissances métiers acquises lors de l'étape d'analyse du besoin.
- ❑ Transformation des données. Il s'agit ici de créer et de mettre à disposition les « features » pour la modélisation.
- ❑ Modélisation. Il s'agit principalement de se concentrer sur la partie technique de l' IA du projet.

- ❑ Validation. Cette étape concerne la validation des modèles mais aussi des données via notamment une analyse de biais.
- ❑ Intégration et déploiement. Cette dernière étape permet d'intégrer l'IA dans une application web afin de pouvoir la déployer chez le client. Elle comprend une analyse de qualité et du monitoring.

La méthode Kanban a été utilisée pour la réalisation de ce projet car son système à flux "tirés" s'adapte en permanence au besoin du client. Afin de visualiser l'avancement de projet, le tableau Kanban dans l'outil Trello a été utilisé tout au long du processus de développement (Fig.2). Le tableau est divisé en plusieurs sections:

- Backlog: rapresent les améliorations à faire;
- Conception & Recherche: il s'agit de la section qui rassemble les idées et l'acculturation métiers;
- A faire: les tâches clefs à faire pour les développement du projet décidées avec la team entière et mise à jour à chaque meeting;
- En cours: les tâches en cours de réalisation;
- Révision du code:étape de validation du code de part des autres développeurs;
- Terminé: section où on retrouve les tâches terminées après leur validation.

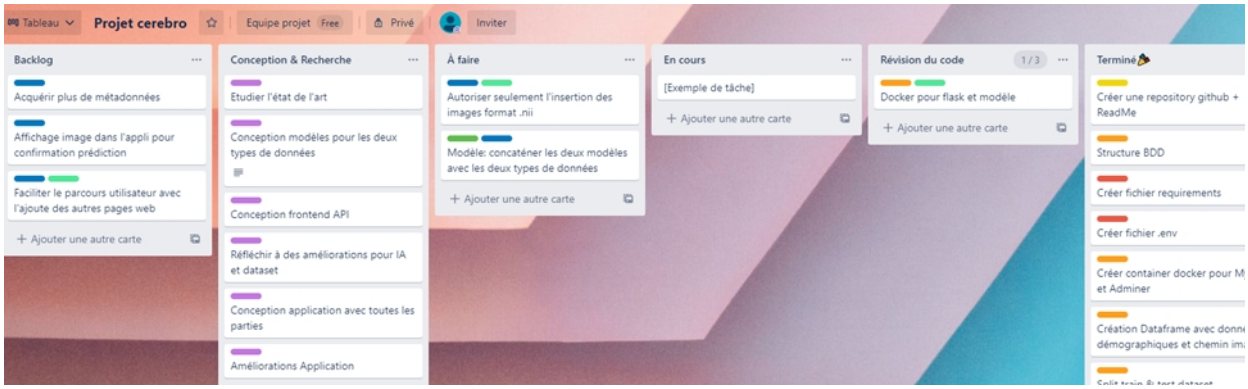



Fig 2. Instantané du suivi projet à fin Mars.

Chaque jour, une réunion est programmée afin d'évaluer la progression du projet. C'est le moment aussi où on peut ajouter d' autres tâches indispensables au développement du projet.

De plus, la communication avec les experts métiers a été au centre de ce projet. Pour cela, une équipe scientifique et médicale a supervisé le projet tout au long du POC, avec des reportings chaque deux semaines (une fréquence de deux fois par mois) ou des ateliers visant à acculturer le métier à la technologie et inversement (Fig.1).

# Intelligence artificielle

L’algorithme d’intelligence artificielle développé permet de prédire le stade de la maladie d’Alzheimer à partir des images cérébrales obtenues par résonance magnétique (IRM). Il sera donc capable de reconnaître les premiers signes d’une détérioration cognitive menant à la maladie d’Alzheimer. Une classification en classes multiples a été utilisée comme modèle d’apprentissage automatique et profond: **CN** (control) qui correspond à des personnes saines, **MCI** (Mild Cognitive Impairment) ou déficit cognitif léger, il représente un trouble cognitif particulier, intermédiaire entre les changements cognitifs liés à l'âge (comme l'oubli bénin lié à l'âge) et les pathologies dégénératives responsables de troubles cognitifs évolutifs et le groupe **AD** (Alzheimer Disease) qui représente le stade sévère ou avancé de la maladie d’Alzheimer.

## Dataset

Le dataset utilisé pour ce projet a été collecté à travers la Alzheimer's Disease Neuroimaging Initiative (ADNI)[3].

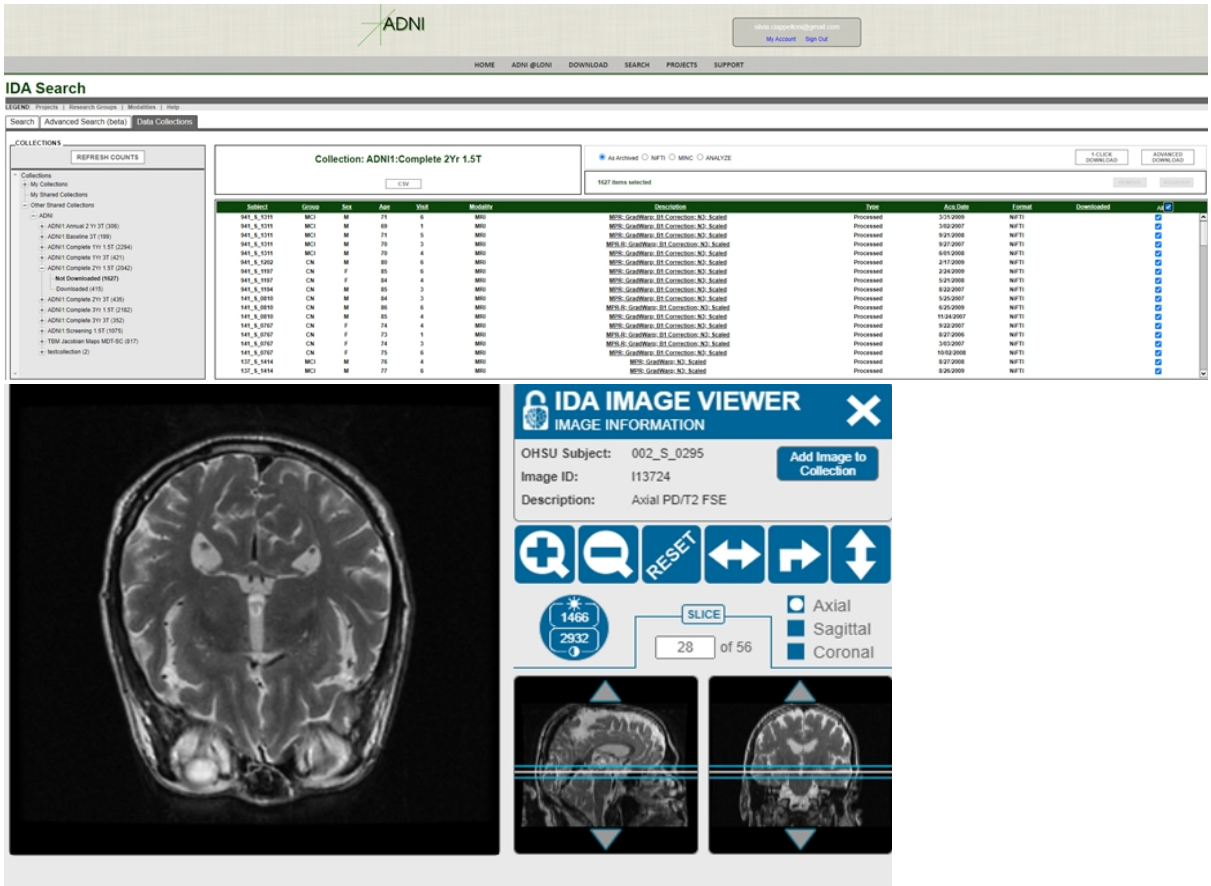



*Fig.3 Vue du site ADNi avec l’exemple d’une image.*

Cette initiative a été créée par un groupe de laboratoires de recherche à l’Université de la Californie du Sud. Elle permet d'accéder librement à un grand volume de données sur la maladie d'Alzheimer après envoi d’une lettre de motivation. Les données recueillies incluent

3 http://adni.loni.usc.edu/data-samples/access-data/

les images et un fichier csv avec des informations annexes, comme les données démographiques, pour un total de 306 patients.

## Création & Nettoyage

Certaines informations du fichier csv d'origine (Fig.3) ont été éliminées lors du nettoyage car pas nécessaires à la prédiction finale (Fig.4).

| | Image Data ID | Subject | Group | Sex | Age | Visit | Modality | Description | Type | Acq Date | Format | Downloaded |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | I143856 | 136_S_1227 | MCI | F | 67 | 6 | MRI | MPR; ; N3; Scaled | Processed | 3/23/2009 | NiFTI | NaN |
| 1 | I99265 | 136_S_1227 | MCI | F | 66 | 4 | MRI | MPR; ; N3; Scaled | Processed | 3/06/2008 | NiFTI | NaN |
| 2 | I66824 | 136_S_1227 | MCI | F | 65 | 2 | MRI | MPR; ; N3; Scaled | Processed | 2/21/2007 | NiFTI | NaN |
| 3 | I40404 | 136_S_0579 | MCI | F | 66 | 2 | MRI | MPR-R; ; N3; Scaled | Processed | 7/10/2006 | NiFTI | NaN |
| 4 | I119735 | 136_S_0579 | MCI | F | 66 | 2 | MRI | MPR-R; ; N3; Scaled_2 | Processed | 7/10/2006 | NiFTI | NaN |

*Fig 3. Visualization du fichier csv brut téléchargé depuis le site ADNI avec les données démographiques.*

La colonne “Subject” a été éliminée aussi car seulement deux sujets sur l’entier dataset, ont eu plus d’une visite.

| | Image Data ID | Group | Sex | Age |
|---|---|---|---|---|
| 0 | I143856 | MCI | F | 67 |
| 1 | I99265 | MCI | F | 66 |
| 2 | I66824 | MCI | F | 65 |
| 3 | I40404 | MCI | F | 66 |
| 4 | I119735 | MCI | F | 66 |

*Fig 4. Dataset après nettoyage.*

## Exportation

Afin d’avoir un dataset final qui ressemblait les données démographiques aux images, le chemin vers les images stockées en local a été ajouté. (Fig.5). Le choix d’enregistrer que le chemin vers l’image a été fait pour privilégier la rapidité des requêtes vers la base de données. Ainsi les données seront mobiles, si les fichiers stockées en local seront déplacés dans un service cloud il y aura  juste à changer le chemin dans le code.

| Image_data_id | Group | Sex | Age | Image_path |
|---|---|---|---|---|
| I101432 | MCI | M | 72 | ../data/ADNI_data/ADNI1_Annual_2_Yr_3T/images\... |
| I101541 | AD | F | 71 | ../data/ADNI_data/ADNI1_Annual_2_Yr_3T/images\... |
| I101566 | MCI | M | 87 | ../data/ADNI_data/ADNI1_Annual_2_Yr_3T/images\... |
| I102675 | MCI | M | 67 | ../data/ADNI_data/ADNI1_Annual_2_Yr_3T/images\... |
| I103276 | AD | F | 74 | ../data/ADNI_data/ADNI1_Annual_2_Yr_3T/images\... |
| ... | ... | ... | ... | ... |

*Fig 5. Dataset finale.*

Depuis le début, les données ont été séparées en deux dataset: train et test (Fig.6) et insérées dans les tables respectives de la base de données (voir chapitre base de

données). Cette étape d'échantillonnage est très importante afin d'éviter le data leakage et permettre une généralisation du modèle sur des données que le modèle n'a jamais vu.

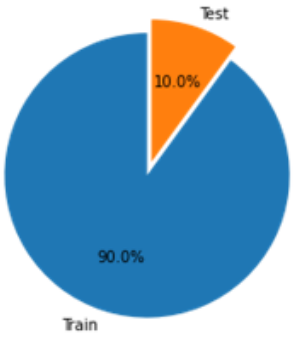



Fig 6. Split du dataset final: 90% des données seront utilisées pour l'entraînement du modèle et le 10% restant pour évaluer sa performance.

Les données ensuite ont été importées depuis la base de données lors de l'exploration et l'entraînement de modèles (Fig.7).

```
path='../BDD/.env'

load_dotenv(dotenv_path=path)
user=os.getenv("MYSQL_USER")
password=os.getenv("MYSQL_PASSWORD")
host=os.getenv("MYSQL_HOST")
database=os.getenv("MYSQL_DATABASE")
port=os.getenv("MYSQL_PORT")

engine = create_engine("mysql+pymysql://{user}:
{pw}@localhost/{db}"
                       .format(user=user,
                               pw=password,
                               db=database))

data = pd.read_sql("select * from Patient_data_train", con =
engine)
```

*Fig 7. Extrait du code avec la connexion à la base de données et l'importation de la table Patient_data_train.*

# Exploration

## Données démographiques

Le dataset ne montre pas la présence de valeurs nulles.

Bien qu'il n'y ait pas beaucoup d'informations démographiques, l'exploration du dataset indique que les femmes sont les plus atteintes de la maladie d'Alzheimer (Fig.8), comme confirmé dans la littérature [1].

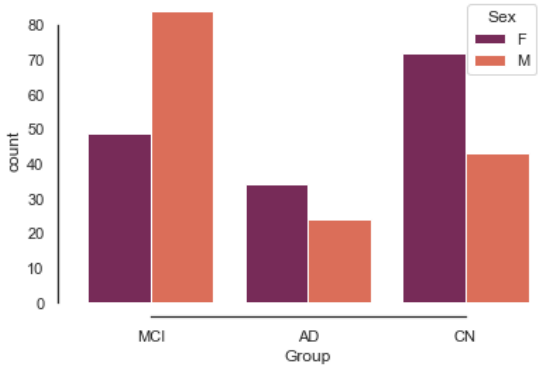



*Fig 8. Nombre d' hommes et de femmes par groupe (MCI, AD, CN).*

Au contraire, les hommes semblent les plus susceptibles à l'apparition des premiers symptômes, déduction qui n'est pas cohérente avec la réalité.
Le fait que le dataset ne soit pas bien balancé pourrait expliquer cette problématique. En effet le groupe AD n'est pas assez représenté (Fig.9).

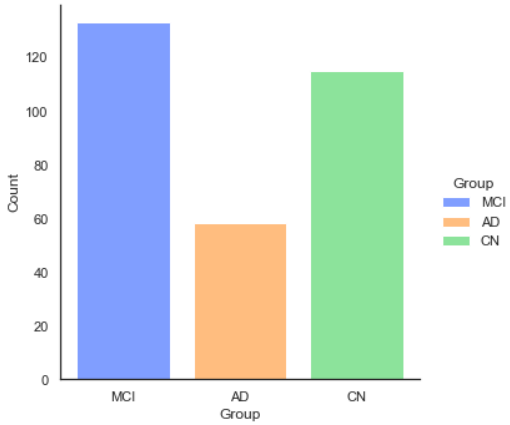



*Fig 9. Distribution de données par les trois différents groupes.*

L'âge est un autre facteur de risque, en effet la possibilité de développer la maladie augmente avec le vieillissement [1]. La moyenne d'âge pour la maladie d'Alzheimer est de 75 ans, qui correspond à l'état de l'art [1] (Fig 10). Les données sur le groupe avec un léger déficit, ne sont pas représentatives de la réalité car normalement les premiers symptômes de la maladie apparaissent autour de l'âge de 65 ans et pas à 74 ans comme dans notre cas. De plus, le group control n'est pas représenté dans toutes les tranches d'âge.

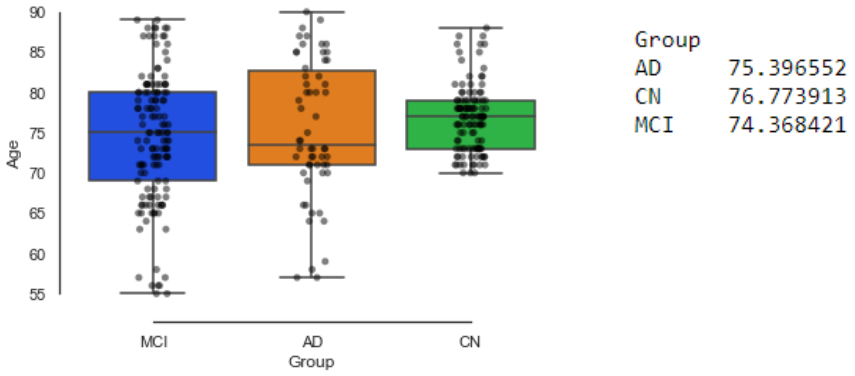



*Fig 10. Distribution de l'âge en fonction du stade de la maladie.*

En général, les informations les plus importantes sont (Fig 11 ):

- le groupe CN est représenté par une petite tranche d'âge comparé aux autres deux groups;
- le groupe MCI est composé en majorité par des hommes avec une apparition plus tardive des premiers signes, comparés aux femmes;
- le groupe AD est représenté en majorité des femmes avec une apparition plus précoce de la maladie (moyenne autour de 70 ans). Au contraire, les hommes ont une apparition plus tardive de la maladie, qui confirme la déduction du groupe MCI.

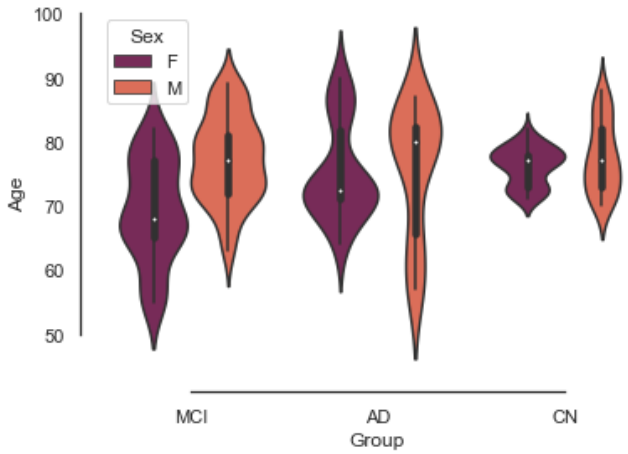



*Fig 11. Graph qui permet de comparer simultanément les trois variables: distribution de l'âge en fonction du stade de la maladie et du genre.*

Après la création des dataset de train et de test, une exploration de données a été réalisée afin d'évaluer la distribution de données. Le dataset de train est bien représentatif de toutes les données (Fig 12). Au contraire dans le dataset de test est visible une légère modification: le group CN est le plus représenté et dans le group AD une difference des moyennes d'age entre homme et femme a été remarquée(Fig 12).

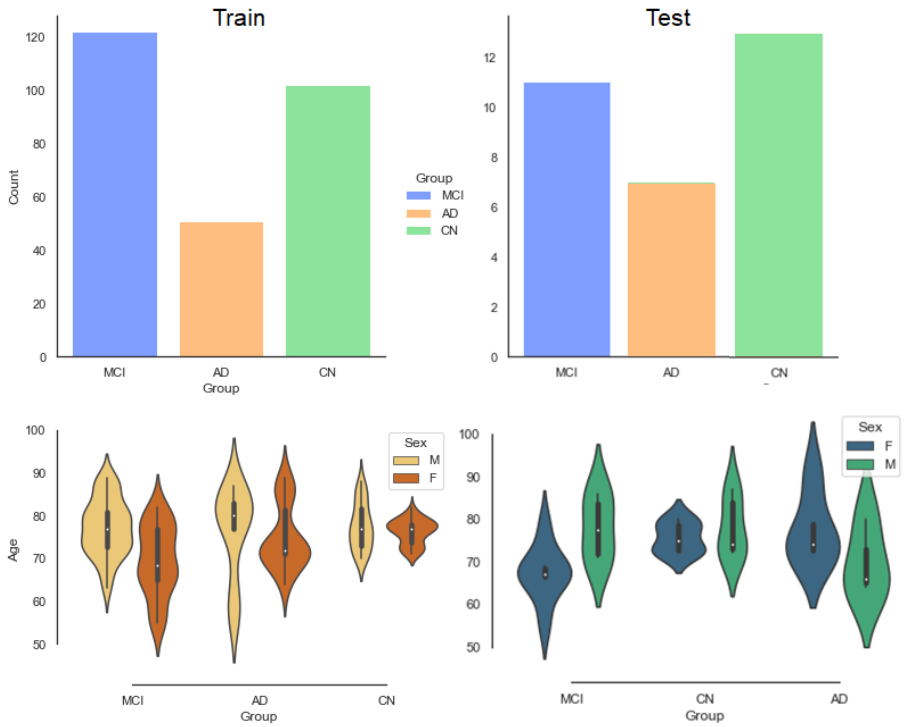



*Fig 12. Distribution des variables entre train  et test*

## Images

Les images téléchargées depuis le site ADNI ont été  organisées en trois dossiers séparés en fonction de leur groupe.Les images sont en 3D et en format Nifti (Fig 13). L'import des images a été réalisé avec la library Nibabel, spécifique à ce type de format.

```
im =
nib.load(adni_data[10]).get_fdata()
plt.imshow(im[85])
print(im[85].shape)
print(len(im))
```

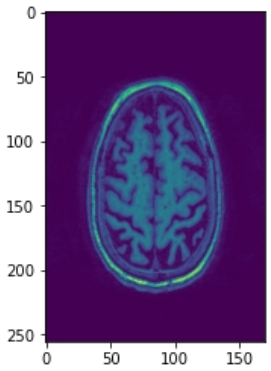


*Fig 13. Extrait de code pour l'import des images. Dans ce cas, la section 85  du patient 10 est visualisée.*

Pour chaque patient nous avons 256 images de taille 256*170 (Fig 14). Au total, 78 336 images ont été analysées.

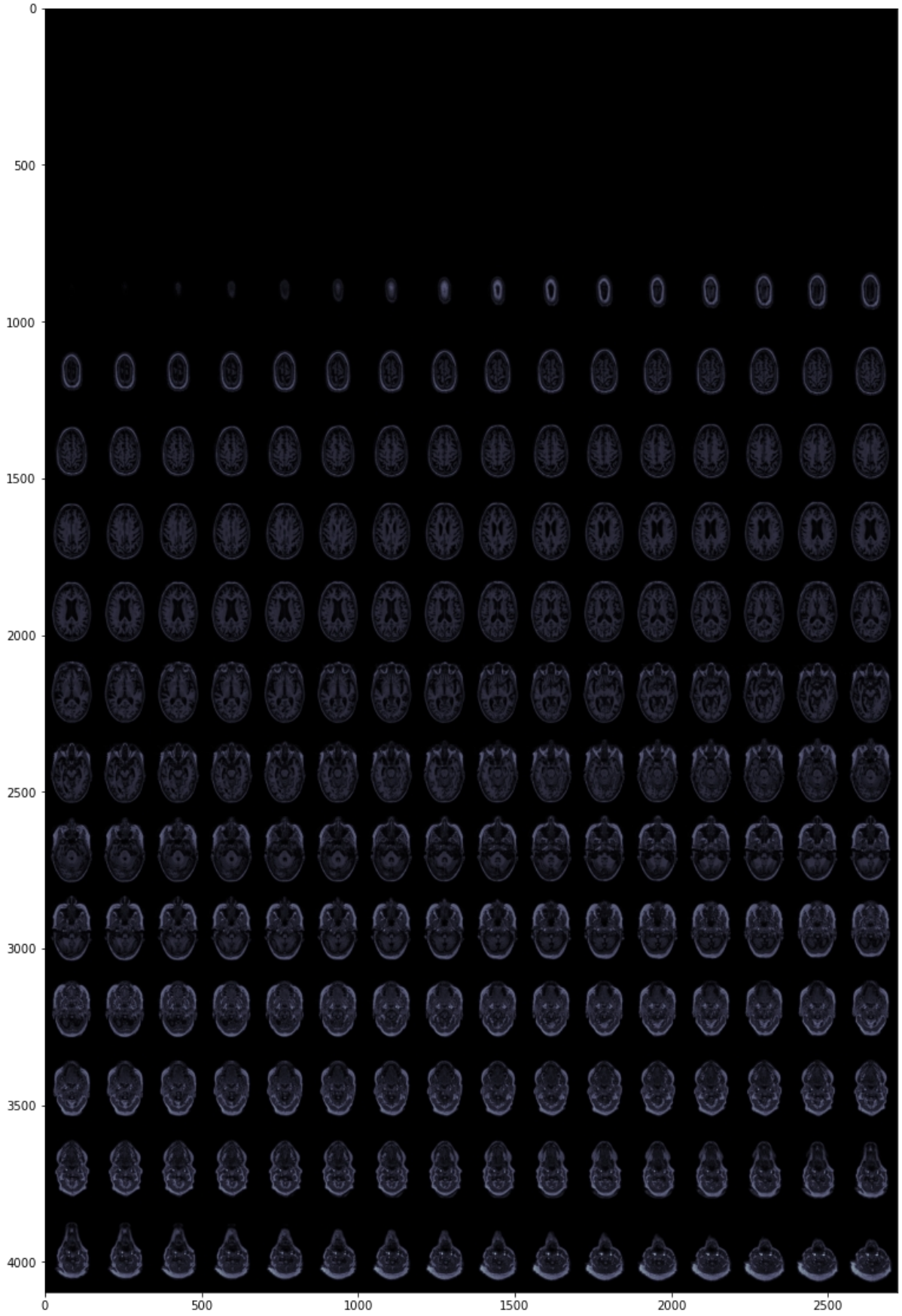



*Fig 14. Totalité des sections pour un patient.*

La visualisation de l’atrophie cérébrale n’est pas si facile à détecter à l'œil humain (Fig 15 et 16).

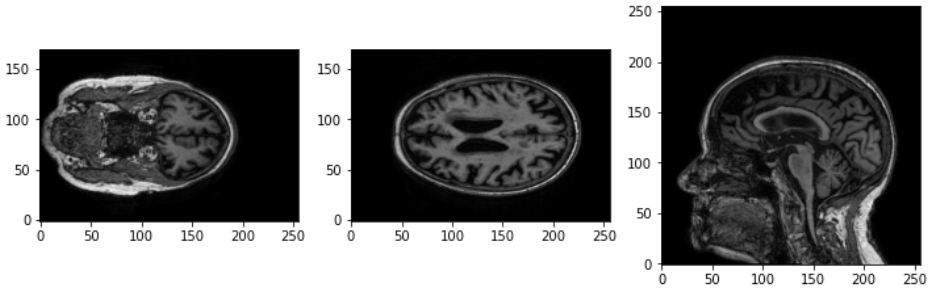


*Fig 15. Exemple de visualisation d’une section sur les trois axes.*

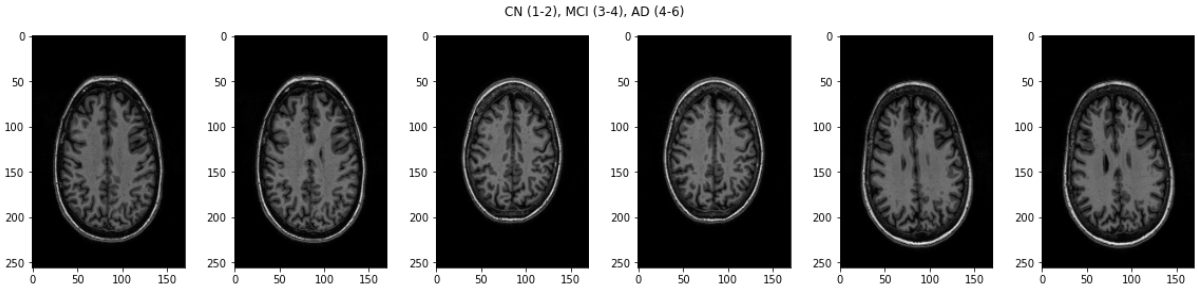



*Fig 16. Rendu visuel de l'évolution du stade de la maladie: deux images pour le CN, deux pour le MCI et deux pour AD.*

## Prétraitement d’images

### 1. Redimensionnement

Afin de trouver un compromis entre la vitesse de calcul et la perte d’informations, une taille d’image de 21x42x42 a été sélectionnée (Fig 17 ).

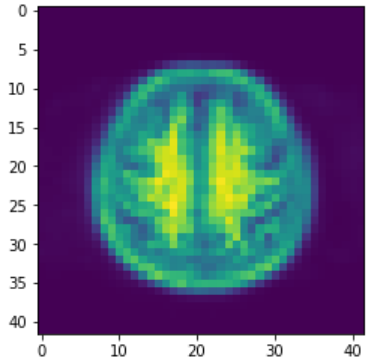


*Fig 17. Visualisation de l’image après redimensionnement*

### 2. Normalisation

Afin de redimensionner les valeurs de pixels de l’intervalle de 0-255 à un intervalle de  0-1, préféré pour les modèles de réseaux neuronaux, une normalisation a été réalisée (Fig 18 et 19). Enfin les images pré-traitées ont été enregistrées sous format numpy.

```
[6.90782323e-03, 1.28059625e-02, 2.00182415e-02, ...,
 4.19063187e-02, 3.95687973e-02, 3.95492577e-02],
[1.85434169e-04, 5.54369816e-04, 1.56699139e-03, ...,
 3.00263355e-02, 3.14064800e-02, 3.11690981e-02],
[5.87481463e-08, 6.96348524e-07, 5.04891461e-06, ...,
 1.46318519e-02, 1.47725281e-02, 1.04587328e-02]]])
```

*Fig 18. Valeurs de pixels des images après redimensionnement et normalisation*

```python
nummean = 138
nummax = 5185
converted_shape = (21,42,42)
for file in os.listdir(ad_data_dir):
    img = nib.load(os.path.join(ad_data_dir,file))
    img = img.get_fdata()
    # resize
    img = skimage.transform.resize(img, converted_shape)
    #normalize
    img = (img - nummean) / nummax
    #rename with id
    id =  file.split('_')[-1].split('.')[0]
    np_filename = id + '.npy'
    #save to a new folder
    np.save(os.path.join(ad_write_dir, np_filename),img)
```

*Fig 19. Extrait du code avec l'import des images, le redimensionnement, la normalisation et la sauvegarde. Le valeur de "nummean" correspond à la moyenne de pixels par image et "nummax" à la valeur maximale des pixels, préalablement calculées.*

Au moment de l'exploration des données, un déséquilibre dans les groupes a été montré. C'est un problème assez courant dans le machine learning. En effet, lorsque la taille des groupes diffère considérablement, les groupes sous-représentés seront vus moins souvent par le modèle et ne seront donc pas appris ainsi que leurs homologues surreprésentés. Afin d'atténuer ces problèmes, l'augmentation des données est conseillée comme axe d'amélioration.

## Méthodes

Comme déjà mentionné plus tôt, il s'agit d'une problématique de classification multiclasse: trois classes sont utilisées afin de prédire l'évolution de la maladie d'Alzheimer. Des méthodes de classification ont été réalisées selon le type de données analysées (Fig 20).

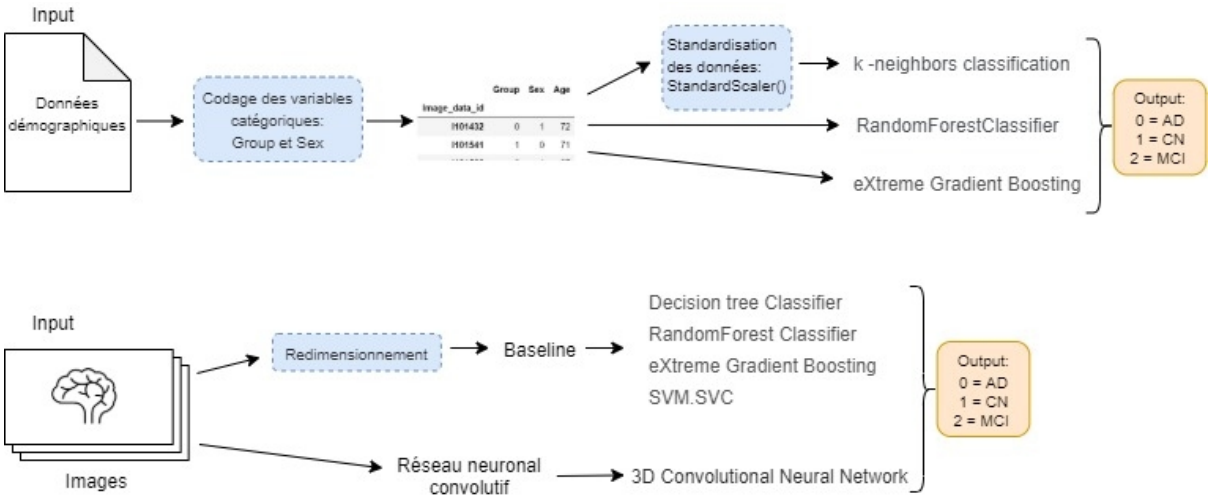



*Fig 20. Schéma récapitulatif de tous les modèles utilisés lors de ce projet.*

Pour tous les modèles, la méthodologie d'entraînement prévoit une séparation entre dataset de train et de test (comme déjà vu dans le paragraphe de création du dataset). Ensuite l'ensemble de train a été divisé en deux dataset: train et validation (Fig 21)

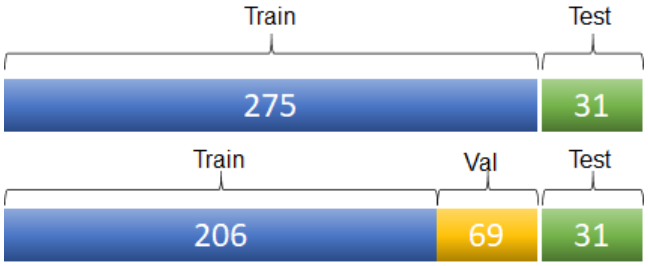



*Fig 21. Définition des trois datasets.*

Pour le traitement des classes, un *Encodeur* pour transformer les labels catégoriels en chiffre a été appliqué: AD : 0, CN : 1, MCI : 2.
Les métriques d'évaluation utilisées pour tous les modèles sont les suivantes : *Accuracy, Precision, Recall,F1 Score* et *matrice de confusion*. Dans le domaine de la santé il est important d'étudier le taux de faux positifs et de faux négatifs. Dans ce cas, le nombre de faux négatifs, personnes malades qui sont prédits comme saines devra être le plus bas possible.
Dans la table "Model's_list" de notre base de données, un résumé des meilleurs modèles et de leurs performances est affiché.

## Données démographiques

### Modèles

Pour les données démographiques trois modèles de classification ont été choisis: le Random Forest, l'arbre de décision avec la méthode boosting (XGBoost) et le k-nearest neighbors (Fig 20). Dans ce dernier cas une standardisation de données a été réalisée en amont car généralement, les modèles qui utilisent la distance comme mesures entre les

variables, sont sensibles aux échelles de valeurs numériques. Les hyperparamètres par défaut ont été utilisés tout au début. (Fig 22)

k-nearest neighbors

```
{'algorithm': 'auto',
 'leaf_size': 30,
 'metric': 'minkowski',
 'metric_params': None,
 'n_jobs': None,
 'n_neighbors': 5,
 'p': 2,
 'weights': 'uniform'}
```

XGBoost

```
(base_score=0.5, booster='gbtree', colsample_bylevel=1,
 colsample_bynode=1, colsample_bytree=1, gamma=0,
 learning_rate=0.1, max_delta_step=0, max_depth=3,
 min_child_weight=1, missing=None, n_estimators=100, n_jobs=1,
 nthread=None, objective='multi:softprob', random_state=0,
 reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
 silent=None, subsample=1, verbosity=1)
```

Random Forest

```
(bootstrap=True, ccp_alpha=0.0, class_weight=None,
 criterion='gini', max_depth=None, max_features='auto',
 max_leaf_nodes=None, max_samples=None,
 min_impurity_decrease=0.0, min_impurity_split=None,
 min_samples_leaf=1, min_samples_split=2,
 min_weight_fraction_leaf=0.0, n_estimators=100,
 n_jobs=None, oob_score=False, random_state=None,
 verbose=0, warm_start=False)
```

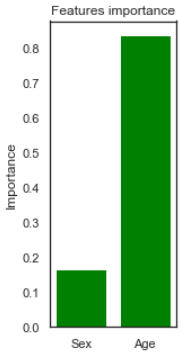
*Fig 22. Hyperparamètres finals choisis pour chaque modèle en utilisant les librairies scikit-learn et xgboost.*

## Résultat

Les performances des différents modèles sont comparées dans la table ci-dessous. Une analyse des caractéristiques les plus importantes pour le modèle montre que l'âge joue un rôle majeur au moment de la prédiction (Table 1 et Fig 23).

| Modèle | Train Acc % | Test Acc % | Recall | Précision | f1-score |
|---|---|---|---|---|---|
| KNeighbors | 71% | 52 % | 0.47 | 0.46 | 0.45 |
| XGBoost | 71 % | 48 % | 0.46 | 0.46 | 0.46 |
| RandomForest | 72 % | 52 % | 0.49 | 0.49 | 0.48 |

*Table 1. Performances des modèles avec les métriques utilisées.*




```
Variable: Age      Importance: 0.83
Variable: Sex      Importance: 0.17
```

*Fig 23. Les variables les plus importantes pour la prédiction*.

Un overfitting rapproche tous les modèles, sûrement causé par la taille réduite du dataset. Comme déjà mentionné, dans notre cas le meilleur modèle sera celui avec le plus bas nombre de faux négatifs. On considère que toutes les valeurs qui font partie des groupes AD et MCI et qui sont prédits comme CN, sont de faux négatifs, ainsi les valeurs du groupe AD qui sont prédit comme MCI (Fig 24 et Table 2.)

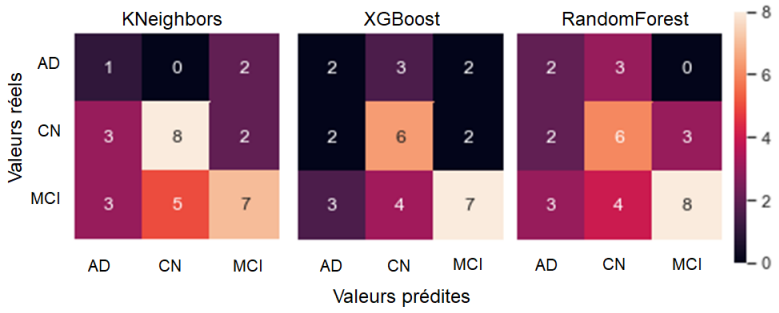



*Fig 24. Matrices de confusion sur le test pour les trois modèles.*

| Modèle | AD / CN | MCI / CN | AD / MCI | Tôt FN | Tôt VP | Tôt VN | Tôt FP |
|---|---|---|---|---|---|---|---|
| KNeighbors | 0 | 5 | 2 | 7 | 8 | 8 | 8 |
| XGBoost | 3 | 4 | 2 | 9 | 9 | 6 | 7 |
| RandomForest | 3 | 4 | 0 | 7 | 10 | 6 | 8 |

*Table 2. Représentation des taux de FN, VP, VN, FP. Les trois premières colonnes représentent les trois types de faux négatifs dans notre cas: les malades sévères qui sont prédits comme sains, les malades légers prédits comme sains et les malades sévères qui sont prédits comme légers.*

Le k-neighbors et le Random Forest ont la meilleure valeur d'accuracy. Afin de choisir le modèle le plus performant, le F1-score a été pris en compte. En effet, le F1-score est un moyen de représenter le recall et la précision ensemble.
Selon le F1-score et l'analyse de la matrice de confusion, le modèle Random Forest est le meilleur modèle.
Afin d'obtenir une meilleure précision pour le RandomForest, deux types de recherche d'hyperparamètres ont été réalisés:

- une cross-validation par *StratifiedKFold* et un *RandomizeSearch:* 60% d'accuracy
- une cross-validation par *StratifiedKFold* et un *GridSearch*: 67% d'accuracy

Effectivement l'application d'hyperparamètres améliore la précision de la prédiction surtout grâce à une diminution des nombres de faux négatifs et faux positifs (FN=5, FP=5, VP=15, VN=5).

En général, la performance de tous les modèles n'est pas exceptionnelle. Le dataset est assez limité, avec seulement deux caractéristiques (Age et Sex) et un nombre de données réduits. De plus comme déjà mentionné, les classes sont déséquilibrées, et dans notre cas, la classe minoritaire c'est justement la classe "AD" qui présente le plus fort intérêt et que l'on aimerait pouvoir identifier. La librairie *imbelearn* permet d'afficher un rapport contenant notamment les résultats sur l'ensemble des métriques du package (Fig 25) Ce tableau montre effectivement que la précision, le recall et le f1-score de la classe 0 (AD) sont mauvais, tandis que pour les autres sont plus élevés, surtout pour la classe 2 (MCI) qui correspond à la classe majoritaire.

| | pre | rec | spe | f1 | geo | iba | sup |
|---|---|---|---|---|---|---|---|
| 0 | 0.29 | 0.40 | 0.81 | 0.33 | 0.57 | 0.31 | 5 |
| 1 | 0.46 | 0.60 | 0.67 | 0.52 | 0.63 | 0.40 | 10 |
| 2 | 0.82 | 0.56 | 0.87 | 0.67 | 0.70 | 0.47 | 16 |

*Fig 25. Rapport des métriques pour les trois classes avec le modèle Random Forest*.

De plus, car le dataset de test n'est pas représentatif de toutes les données, on augmente le biais et le modèle devient moins représentatif de la réalité.
Afin d'améliorer la performance, différentes solutions pourraient êtres envisagées:

- une collecte de données supplémentaires afin de rééquilibrer les classes;
- enrichir le dataset avec d' autres informations: données biologiques, donnée de vie, etc..;
- utiliser des méthodes de rééchantillonnage afin d'avoir des données plus équilibrées. Il existe deux méthodes: le sur-échantillonnage (augmente le nombre d'observations de la classe minoritaire afin d'arriver à un ratio classe minoritaire/ classe majoritaire satisfaisant) et les sous-échantillonnage (diminue le nombre d'observations de la classe majoritaire afin d'arriver à un bon ratio).

# Images

## Modèles

Afin d'avoir une référence de performance pour les modèles futures, des baselines avec des méthodes de machine learning, ont été réalisées. Pour cela, quatre modèles de classifications ont été utilisés (Fig 20). Toutefois avant le début de l'entraînement, un redimensionnement de la taille des images a été nécessaire (Fig 26).

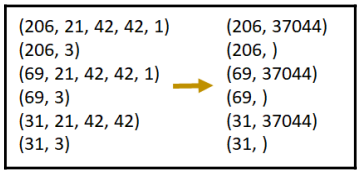



*Fig 26. Redimensionnement des images afin de pouvoir les entraîner avec des modèles de machine learning.*

Pour la baseline, des modèles avec des hyperparamètres par défaut ont été utilisés (Fig 27). Ensuite sur le meilleur modèle, une recherche des hyperparamètres a été réalisée.

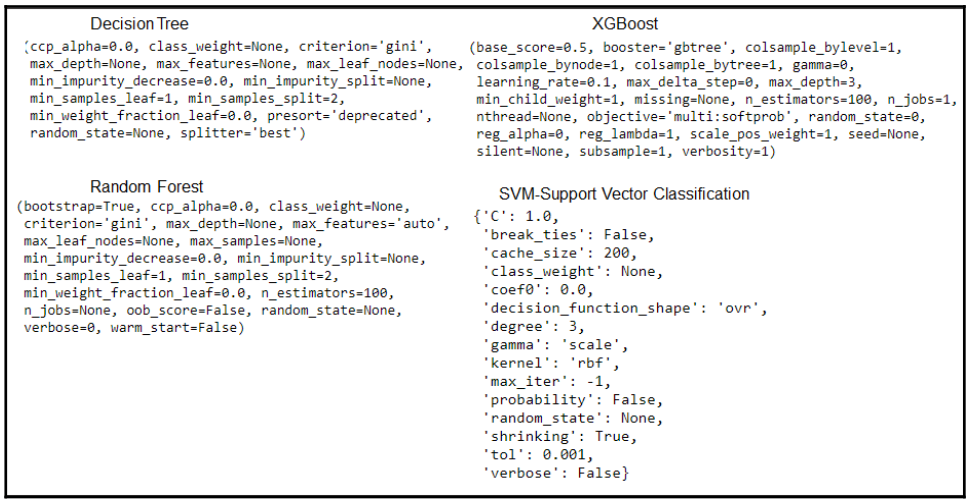
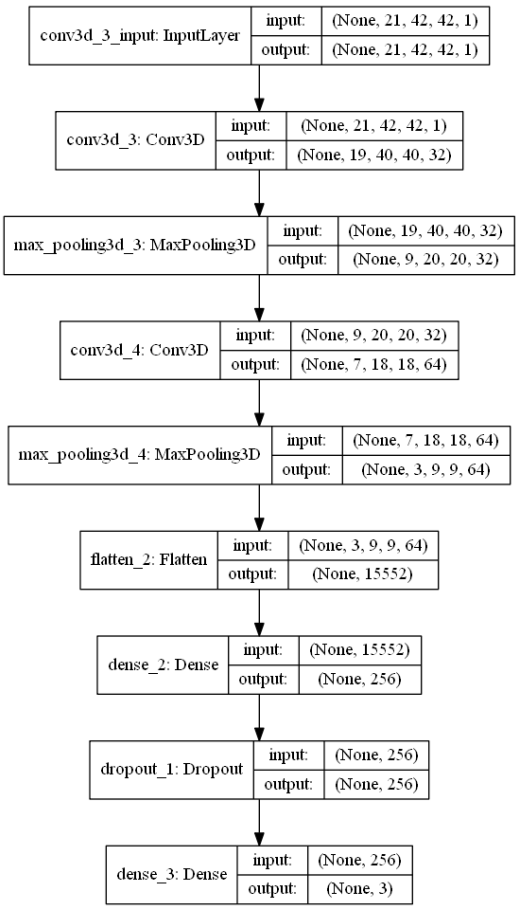



*Fig 27. Paramètres utilisés pour les quatre modèles de baseline en utilisant les librairies scikit-learn et xgboost.*

Du deep learning avec un simple réseau neuronal convolutif 3D a été ensuite appliqué aux images afin de garder l'information de la tridimensionnalité (Fig 28). Ce modèle a été sélectionné comme meilleur réseau neuronal convolutif dans le reporting du *MSPI(Mise en situation professionnelle 1)*. D'abord les fonctions d'activation pour les couches cachés et pour l'output ont été choisies: *ReLU* et *softmax* respectivement. L'optimiseur *adam* a été sélectionné sur la base de sa prévalence dans la littérature et la fonction loss *categorical_crossentropy* a été utilisée. Ensuite, 2 couches de convolution 3D et de max pooling 3D alternées ont été ajoutées séquentiellement. Une taille de *kernel* 3 x 3 x 3 a été sélectionnée pour l'extraction maximale des informations locales. Enfin la manipulation du *dropou*t (10%, 30%, 50%) a été explorée et l'essai utilisant le 30% de *dropout* a été choisi. Le nombre total de paramètres traités par cette architecture est de 4,038,595.




```
batch_size = 100
no_epochs = 100
learning_rate = 0.001
no_classes = 3
verbosity = 1
sample_shape = (21, 42, 42, 1)
```

```
model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adam(lr=learning_rate),
              metrics=['accuracy'])
```

*Fig 28. Architecture réseau neuronal convolutif 3D et paramètres utilisés avec les librairies Keras et Tensorflow.*

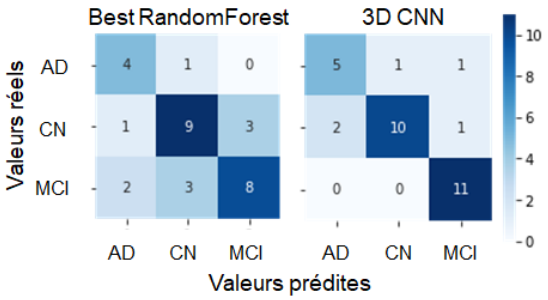
## Résultats

Comme prévu, la différence de performance entre les modèles baseline et le réseau neuronal convolutif 3D est flagrante (Table 3). Pour ce qui concerne les modèles de baseline, le Random Forest avec hyperparamètre à la meilleure performance.

| Modèle | Train Acc | Test ACC | Recall | Précision | f1-score |
|---|---|---|---|---|---|
| Arbre de décision | 98 % | 45 % | 0.45 | 0.45 | 0.45 |
| RandomForest | 99 % | 62 % | 0.61 | 0.64 | 0.61 |
| RF + StratifiedKFold + RandomizedSearch | 98 % | 63 % | 0.61 | 0.71 | 0.63 |
| RF + StratifiedKFold + GridSearch | 99 % | 68 % | 0.66 | 0.72 | 0.68 |
| XGBoost | 99 % | 62 % | 0.61 | 0.64 | 0.61 |
| SVM.SVC | 61 % | 42 % | 0.42 | 0.57 | 0.47 |
| 3D CNN | 99 % | 84 % | 0.84 | 0.84 | 0.84 |

*Table 3. Performances des modèles avec les métriques utilisées.*

Les modèles baselines sont caractérisés par un overfitting très important, en confirmant le fait qu'ils ne sont pas adaptés à une analyse des images.
Comme déjà traité pour les données démographiques, le nombre de faux négatifs doit être le plus bas possible et c'est le cas pour le réseau neuronal convolutif en 3D (Fig 29).




| Modèle | AD / CN | MCI / CN | AD / MCI | Tôt FN | Tôt VP | Tôt VN | Tôt FP |
|---|---|---|---|---|---|---|---|
| Best RandomForest | 1 | 3 | 0 | 4 | 12 | 9 | 6 |
| 3D CNN | 1 | 0 | 1 | 2 | 16 | 10 | 3 |

*Fig 29. Représentation des taux de FN, VP, VN, FP. et Matrice de confusion des modèles Random Forest et 3DCNN.*

Même si le dataset de test est déséquilibré, comme mentionné précédemment, le réseau neuronal convolutif arrive à bien prédire toutes les classes (Fig 30)

| | pre | rec | spe | f1 | geo | iba | sup |
|---|---|---|---|---|---|---|---|
| 0 | 0.71 | 0.71 | 0.92 | 0.71 | 0.81 | 0.64 | 7 |
| 1 | 0.91 | 0.77 | 0.94 | 0.83 | 0.85 | 0.71 | 13 |
| 2 | 0.85 | 1.00 | 0.90 | 0.92 | 0.95 | 0.91 | 11 |

*Fig 30. Rapport des métriques pour les trois classes avec le 3D CNN avec la librarie imbelearn.*

En conclusion, le réseau neuronal convolutif en 3D donne une très bonne performance dans la prédiction de stades de la maladie d'Alzheimer. La création d'un CNN plus complexe pourrait améliorer encore la performance. Puisque à présent, un vrai modèle de transfer learning sur des images tridimensionnelles n'apparaît pas dans la littérature, le réseau convolutif 3D reste le meilleur choix afin de préserver l'information de la tridimensionnalité. Cette section deep learning sera traitée plus en détail dans le rapport *Mise en situation professionnelle*.

## Améliorations IA

Pour le moment, deux modèles séparés ont été appliqués selon le type de données, réduisant probablement la performance. La fonction *concatenate* de *keras* permet d'accepter plusieurs input différents et donner un seul output en sortie afin de contribuer tous à la prédiction finale. Appliquer cette fonction dans le projet, pourrait incrément de manière significative la performance.

# Application

L'application cerebro est une aide au diagnostic du stade de la maladie d'Alzheimer à partir de l'analyse des images cérébrales. Elle est composée de plusieurs parties comme montré dans la figure 31. Docker est utilisé pour un déploiement plus facile. A ce moment, seulement les containers pour la base de données (MySQL et Adminer) ont été créés. L'ajout du module Flask est prévu, pour l'optimisation et le déploiement de l'application.

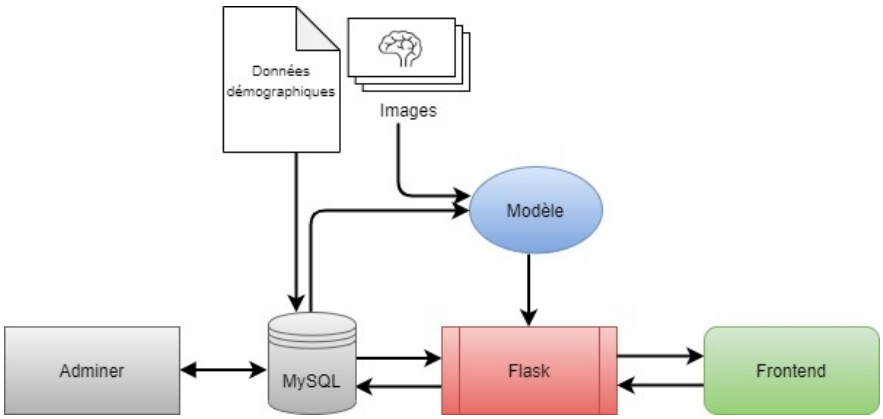


*Fig 31. Schéma fonctionnel de différents composants de l'application cerebro.*

Dans le détail, les données démographiques ainsi que le chemin vers les images sont stockées dans une base de données MySQL qui utilise l'interface Adminer pour les gérer. Les données, importées depuis la base de données, et les images, stockées en local, sont utilisées pour concevoir plusieurs modèles. Comme déjà mentionné, pour le moment, le seul modèle exporté et utilisé pour la prédiction dans l'API est celui provenant de l'entraînement des images. Ensuite l'API Flask requiert la base de données et la table "Docteurs" dans le spécifique, afin d'afficher la liste de docteurs dans sa première page. Une fois que l'utilisateur rentre toutes les données du patient, y compris l'image, une deuxième page avec la prédiction sur le stade de la maladie apparaît. De plus, une table avec le résultat de la prédiction et la date du scanner est affichée. Si le patient a déjà effectué un examen dans le passé, l'historique de ses scanners sera visualisé dans le tableau. Si c'est un nouveau patient, il sera inséré dans la table "Patients". Dans les deux cas, la table "Scanner_cérébral" sera toujours alimentée.

# Base de donnée

## Conception

Dans le cas de notre application cerebro, le choix d'une base de données de type relationnelle a été effectuée pour mettre en avant les relations entre les données. Afin de modéliser la composition des données (MCD), nous avons utilisé l'outil *DBDesigner* en respectant les principes de la méthode MERISE. Cet outil permet de générer rapidement des scripts SQL.
La base de données a été organisée en deux sous parties: la partie développement et la partie application.

### Développement

Comme déjà anticipé dans l'élaboration du dataset, les données de train et de test, ainsi qu' une historique de meilleurs modèles entraînés ont été stockées dans des tables de la base de données (Fig 32).

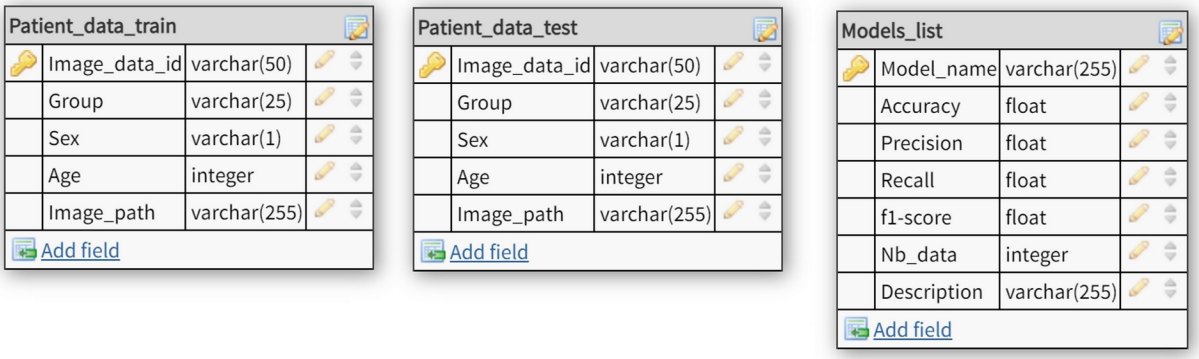



*Fig 32. Rapresentation MCD avec dictionnaire de données pour les tables Patient_data_train, Patient_data_test, Models_list.*

## Application

Le but de ces tables est principalement d'améliorer la prise en charge des patients et le diagnostic de leur maladie, de faire de la veille sanitaire et d'aider la recherche.
Dans le spécifique, une historique pour chaque patient a été considérée essentielle afin d'évaluer la progression de la maladie.
La réalisation du premier MCD pour la table "Historique_patient" (Fig 33) a relevé des formes non normales, car elle contient des redondances. En effet pour un specific "Id_patient", on aura toujours les mêmes valeurs de "Sex" et de "Date_de_naissance".


*Fig 33. Première MCD de la table pour le suivi du patient.*

La solution a été de décomposer cette table en plusieurs relations plus petites afin d'en éliminer les redondances et sans perdre d'information.
Deux entités ont été établies, le "Patient" et le "Scanner_cérébral" liées par la classe d'association "Examiner" (Fig 34).

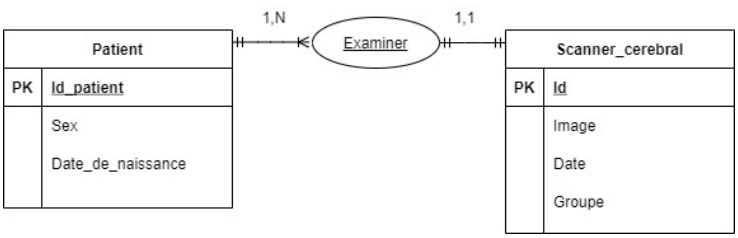



*Fig 34. Représentation des relations entre les entités Patient et Scanner cérébral.*

Une occurrence de l'entité PATIENT est en relation par la relation EXAMINER avec une ou plusieurs occurrences de SCANNER_CEREBRAL (1,N = un à plusieurs): un patient peut être diagnostiqué et soumis à plusieurs scanners. Une occurrence de

SCANNER_CEREBRAL est en relation avec la relation EXAMINER avec une et une seule occurrence de PATIENT (1,1 = un à un): un specific scanner est relié à un seul patient. De plus, pour faciliter le suivi d'un patient par le médecin, une table "Docteur" a été générée.
L'entité DOCTEURS est donc liée à l'entité PATIENTS par la classe d'association VISITER (Fig 35)

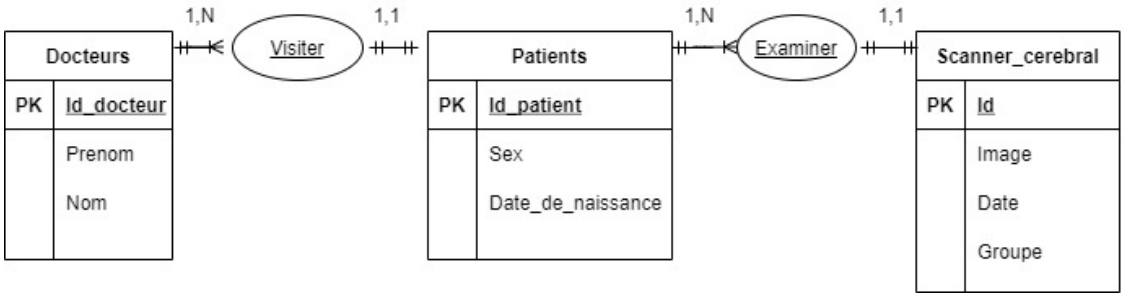



*Fig 35. Représentation finale des relations entre les entités Docteurs, Patients, Scanner_cerebral.*

Un médecin suit plusieurs patients (1,N) et normalement un patient est suivi par un seul médecin spécialiste (1,1).
A partir de ce diagramme de classe, un modèle physique de données a été créé (Fig 36)

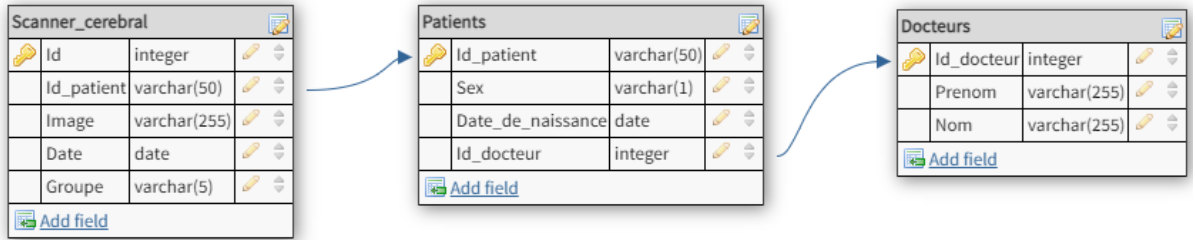



*Fig 36. Schéma avec les relations entre les tables.*

La relation entre les tables est définie par la création des clés étrangères: les tables Scanner_cerebral et Patients contiennent des colonnes qui correspondent à la clé principale des tables Patients et Docteurs respectivement.

## Création

Le système MYSQL a été choisi pour la gestion de cette base de données et Adminer a été l'interface utilisée. Adminer a été choisie pour sa simplicité d'installation et utilisation.
Lors de la première connexion, les tables ont été générées (Fig 37).

```
query2 = """CREATE TABLE `Patients` (
`Id_patient` varchar(50) NOT NULL,
`Sex` varchar(1) NOT NULL,
`Date_de_naissance` DATE NOT NULL,
`Id_docteur` INT NOT NULL,
PRIMARY KEY (`Id_patient`),
FOREIGN KEY (`Id_docteur`) REFERENCES
`Docteurs`(`Id_docteur`)
);"""
```

```
engine.execute(query2)
```

*Fig 37. Extrait du script qui crée la table Patient avec la librairie SQLAlchemy.*

## Utilisation

Puisque les effectifs dans un Hôpital ou clinique sont connus et sont assez stables dans le temps, la table “Docteurs” a été remplie directement (Fig 38).

```
INSERT INTO Docteurs
(Prenom,Nom)
VALUES ('John','Carter'),
('Doug', 'Ross'),
('Mark', 'Green'),
('Peter', 'Benton'),
('Susan', 'Lewis'),
('Kerry', 'Weaver');
```

*Fig 38. Exemple d'insertion de données dans la table Docteurs.*

Les tables “Patient_data_train” et “_test” ont été générées lors de la manipulation du dataset comme vu dans la chapitre dédié (Fig 39).

```
train.to_sql('Patient_data_train', con = engine, if_exists = 'append', chunksize
= 1000)
test.to_sql('Patient_data_test', con = engine, if_exists = 'append', chunksize =
1000)
```

*Fig 39. Exemple d'insertion de données sous forme de dataframe dans les tables train et test.*

L’insertion des données dans les tables “Patients” et “Scanner_cerebral” sera traitée plus en détail dans le chapitre dédiée à l’API Flask.
Pour le moment l’application sauvegarde les images uploadées par les clients dans un dossier /uploads/ et stockant les chemins d’accès vers ces images et leurs prédictions dans la table “Scanner_cerebral”. Par conséquent, un mécanisme de récupération automatique des images pourrait être mis en place. Si par exemple les images stockées dans la table “Scanner_cerebral” sont un quantité suffisante, un script pourrait être créé afin de récupérer ces informations pour enrichir les tables de train et test et éventuellement entraîner un modèle d’apprentissage automatique. Cependant, il faudrait avoir une confirmation de la prédiction de la part de l'utilisateur, tache pas encore réalisée.

## Sauvegarde

Une fois que les données sont insérées, une sauvegarde de la base de données pourrait être utile afin d'éviter certains risques comme le piratage de données sensibles (Fig 40).

```
docker exec -t database mysqldump -u cerebro -pcerebro cerebro > 20200317_backup.sql
more 20200317_backup.sql
docker exec -t database mysqldump -u cerebro -pcerebro cerebro | gzip >
20200317_backup.sql.gz
docker exec -t database mysqldump -u cerebro -pcerebro cerebro Scanner_cerebral >
```

```
Scanner_backup.sql
docker exec -t database mysqldump -u cerebro -pcerebro cerebro < 20200317_backup.sql
```

*Fig 40.Scripts pour la sauvegarde et la restauration d'une base de données. 1. Sauvegarde base de données; 2.Vérifier la sauvegarde; 3.Sauvegarde et compression; 4.Sauvegarde d'une table de la base de données; 5. Restaurer la base de données.*

De plus, un mécanisme de sauvegarde automatique a été ajouté directement dans le fichier docker-compose. Une sauvegarde de la base de données est réalisée chaque jour à minuit et enregistrée dans le dossier /backups/ (Fig.41).

```
container_name: db-backup
    image: fradelg/mysql-cron-backup
    depends_on:
      - database
    volumes:
      - ./backup:/backup
   environment:
      - MYSQL_HOST=
      - MYSQL_USER=
      - MYSQL_PASSWORD=
      - MYSQL_DATABASE=
      - MAX_BACKUPS=15
      - INIT_BACKUP=0
      - CRON_TIME=0 0 * * *
      - GZIP_LEVEL=9
```

*Fig 41. Extrait du code du fichier yaml avec la création du volume pour la sauvegarde automatique*

## Performance

Pour le moment le dataset est assez réduit, donc les requêtes sont très rapides (Fig. 42).

```
mysql> select count(*) from Patient_data_train WHERE `Group`='MCI';
+----------+
| count(*) |
+----------+
|      122 |
+----------+
1 row in set (0.00 sec)
```

*Fig 42. Exemple de requête MySQL pour afficher le nombre de patients affectés par un déficit cognitif léger (MCI).*

Si le temps d'exécution de la requête était trop long, on pourrait utiliser la fonction *MySQL EXPLAIN ANALYZE*. Cela permet de savoir où MySQL passe du temps sur la requête et pourquoi(Fig 43).

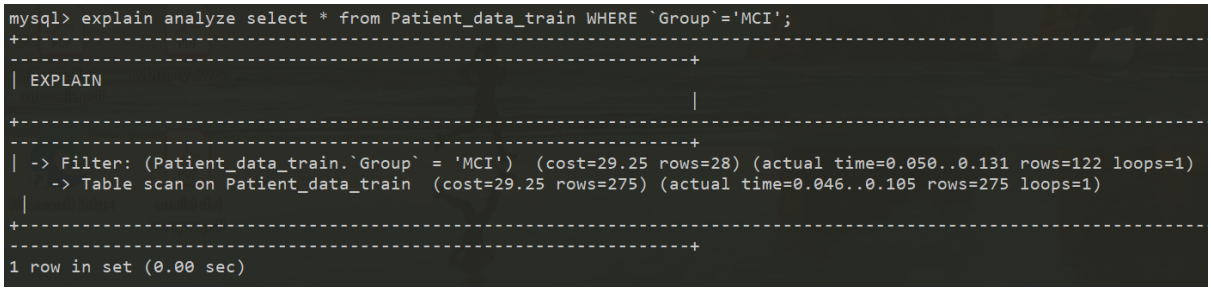



*Fig 43. Exemple de requête avec EXPLAIN ANALYZE.*

Par exemple pour filtrer les lignes correspondantes au group MCI, les informations qu' on obtient sont les suivantes:

| | |
|---|---|
| Temps estimé | cost=29.25ms |
| Nombre estimé de lignes lues | rows=28 |
| Temps réel pour obtenir la première ligne | actual time=0.046 ms |
| Temps réel pour obtenir toutes les lignes | actual time=0.132 ms |
| Nombre réel de lignes lues | rows=122 |
| Nombre réel de boucles | loops=1 |

*Fig 44. Informations générées par la fonction EXPLAIN ANALYZE. Les plus intéressantes sont celles qui concernent le temps réel de la requête.*

Comme déjà mentionné, pour le moment une optimisation n'est pas nécessaire car les requêtes sont très rapides. Si prochainement le dataset s'amplifie et les requêtes deviennent plus complexes et donc plus lentes, il sera intéressant de créer des index sur la base de données (Fig 45).

```
ALTER TABLE Patient_data_train ADD INDEX(`Patient-id`);
```

*Fig 45. Exemple requête pour ajouter un index*

En effet, si la table dispose d'un index, MySQL peut alors trouver rapidement les positions des lignes dans le fichier de données sans avoir à parcourir toute la table.

## Backend

Le micro-framework Python Flask a été utilisé pour concevoir l’application web. Pour rappel, la finalité de l’application est de donner une prédiction du stade de la maladie d’Alzheimer après chargement d’une image d' IRM cérébrale.
On pourrait diviser la conception de l’application en plusieurs partie parties:

| | |
|---|---|
| (app.py)<br><br>@app.route('/')<br>def index():<br>cursor = mysql.connection.cursor()<br>cursor.execute('SELECT Id_docteur, Nom FROM Docteurs')<br>doclist = cursor.fetchall()<br>return render_template('index.html', doclist=doclist) | Une requête HTTP avec l'URL "/" est effectuée. Le serveur génère une page web à partir du fichier index.html, ainsi que la liste de docteurs, récupérée depuis la base données, comme paramètre. |
| (index.html)<br><br><form action="/result" method="POST" enctype="multipart/form-data"><br><br> <input type="submit" value="Envoyer" /> | Cette page, qui contient un formulaire, est envoyée vers le client. Les attributs action et méthode indiquent que le client devra effectuer une requête de type POST dès que l'utilisateur appuiera sur le bouton "Envoyer" . |
| (app.py)<br><br>@app.route('/result', methods=['POST'])<br>def predict():<br>……..<br>return render_template('result.html', Text = show, Prediction = prediction_final, data=data ) | Cette requête POST sera envoyée à l'URL "/result" (voir l'attribut "action"). Les données saisies dans le formulaire seront envoyées au serveur par l'intermédiaire de cette requête. |

- Cette deuxième vue va traiter la requête POST. Elle récupère les données insérées par l'utilisateur: l'image, les données démographiques et la date d'examen.
- L'image est pré-traitée selon le même prétraitement utilisé lors de la modélisation (Fig 46).
- Le modèle de prédiction est ensuite chargé et utilisé pour la prédiction (Fig 46).
- En réponse à la requête POST, le serveur renvoie une deuxième page HTML créée à partir du template "*resultat.html*" et les paramètres "Text" et "Prediction", avec le résultat de la prédiction, sont transmis au navigateur. De plus, le paramètre "data" aussi est envoyé; il s'agit d'une table qui affiche l'historique des examens du patient analysé. Si le patient a déjà effectué un examen dans le passé, l'historique de ses scanners sera visualisé dans le tableau. Si c'est un nouveau patient, il sera inséré dans la table Patients et le résultat de l'examen sera affiché comme pour le cas précédent. Dans les deux cas, la table Scanner cérébral sera toujours alimentée avec les résultats de la prédiction de l'examen ainsi que les autres données requises.

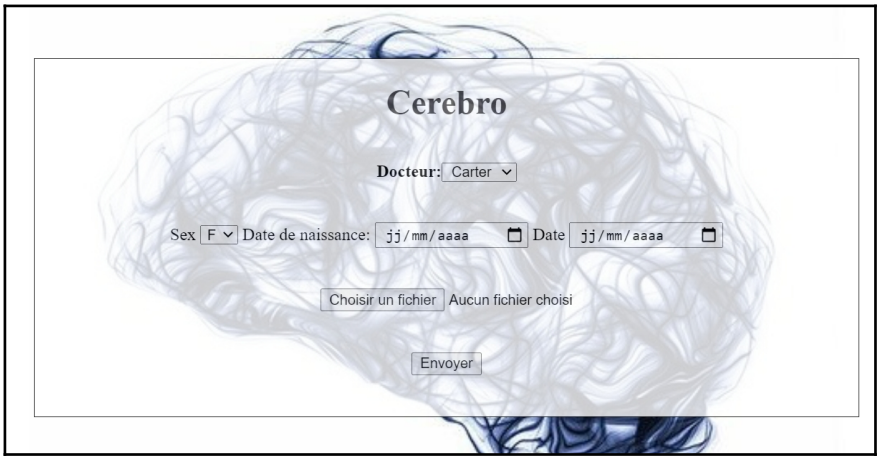
```
[2021-03-29 14:18:32,522] INFO in app:
<tensorflow.python.keras.engine.sequential.Sequential object at
0x000001CFE2FE4C08>
[2021-03-29 14:18:33,077] INFO in app: [[[[[-0.02658008]
    [-0.0265981 ]
    [-0.0266143 ]
    ...
    [-0.02661507]
    [-0.02660648]
    [-0.02659936]]
[2021-03-29 14:18:33,413] INFO in app: [[3.6175967e-05 8.3802027e-01 1.6194354e-
```

```
01]]
```

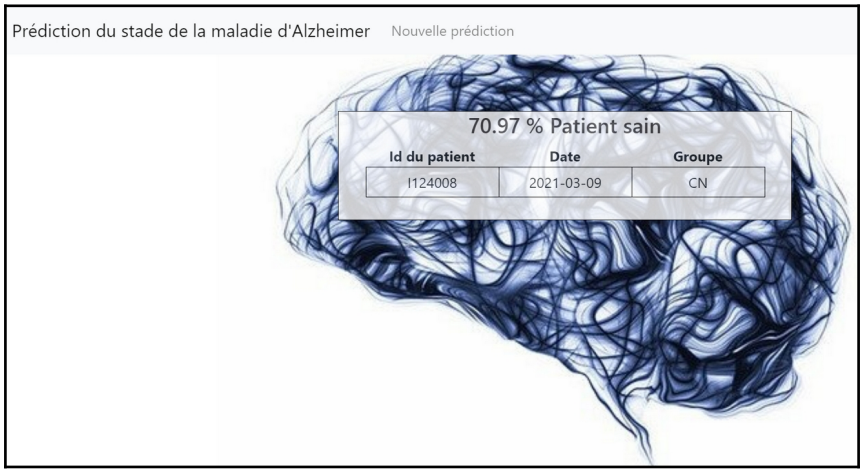
*Fig 46. Exemple de logs de l'intégration de l'IA: upload du modèle, prétraitement des images et entraînement du modèle.*

# Frontend

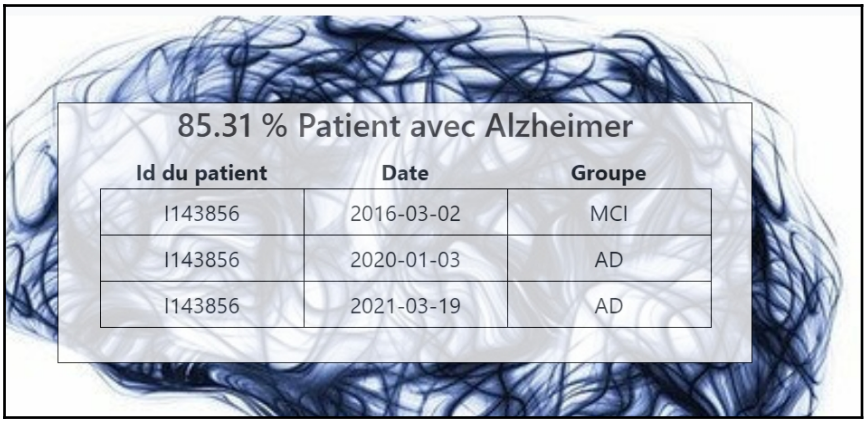
Le frontend a été réalisé avec Jinja, un système de template pour HTML et du CSS a été utilisé afin d'améliorer et optimiser l'esthétique de l'application.
L'application web est caractérisée par deux pages html:




*index.html avec le choix du docteur qui a effectué le diagnostic, les données du patients, ainsi que la date de l'examen. Après chargement de l'image cerebral en format nifti, on peut cliquer sur "Envoyer" afin de transmettre les données au serveur.*




La page *result.html* affiche le résultat de la prédiction du stade de la maladie (en % de probabilité) ainsi qu' une table avec certaines données du patient collectées depuis la table "Scanner_cerebral". Le retour à la page d'accueil est possible en *cliquant sur "Nouvelle prédiction".*




*Autre possible écran de la page result.html avec l'historique des examens, qui permet de visualiser l'évolution de la maladie.*

# Axes d'amélioration

Les points techniques majeurs d'améliorations ont été déjà traités dans les sections spécifiques.
Des autres optimisations sont à réalisées concernant le parcour utilisateur et l'application:

- L'interface web pourrait être amélioré avec un parcour utilisateur plus simple et rapide:
    - une première page où l'utilisateur s'identifie;
    - une deuxième page avec l'insertion de la date de l'examen et de l'image à analyser et cliquer sur "Envoyer";
    - en cliquant sur "Envoyer", deux scénarios seront possibles:
        - si le patient existe déjà dans la table "Patient", afficher la prédiction avec l'historique de ses scanners;
        - si c'est un nouveau patient, demander de mettre à jour les autres informations (comme genre et âge)

      Dans les deux cas, les informations seront toujours insérées dans la base de données.
- Comme déjà mentionné, le docteur n'a pas la possibilité de confirmer la prédiction affichée. Un possible axe d'amélioration pourrait être de montrer certaines sections du cerveau des images 3D qui peuvent aider la confirmation de la prédiction de la part de l'utilisateur. Donner les moyens de comprendre la décision prise par l'algorithme est un aspect crucial surtout dans le milieu de la santé. L'objectif est d' aider le médecin et non de le remplacer. C'est pour cela qu'il doit être en mesure de comprendre le pourquoi et le comment des décisions affichées, et de les contourner si besoin.
- Les données traitées sont des données sensibles et leur utilisation doit respecter le règlement général sur les données personnelles (RGPD). Dans ce cas, les données sont anonymisées, mais il faudrait réfléchir à comment identifier le patient dans le respect des lois.

# Conclusions

Ce projet a été l'occasion de combiner mes expertises dans le domaine des neurosciences avec celles acquises au cours de la formation. L'objectif a été de répondre à un besoin réel et concret des acteurs du milieu de la santé. Sensible à cette thématique et à l'impact que l'IA pourrait avoir dans le milieu de la santé, ce projet pourrait être un début pour ma suite professionnelle.

Il reste encore de nombreux points à améliorées, toutefois je suis satisfaite du projet et de ce premier résultat atteint.