ROYAUME DU MAROC
UNIVERSITE MOHAMMED PREMIER
Ecole Nationale des Sciences Appliquées (ENSA)
Oujda - Maroc

Mémoire de Fin d'Année

Filière : Ingénierie du Data Science et Cloud Computing

Modélisation des effets psychologiques du Covid-19 au milieu académique en utilisant des techniques de Machine Learning

Réalisé par :

ZERROUKI Chaima
BELAOUCHI Lamyae

Jury d'examen

Mr. KERKRI
Abdelmounaim

Mr. MADANI
Mohamed Amine

Encadrant

Mr. KERKRI
Abdelmounaim

Remerciements

Au terme de ce projet, nous tenons à exprimer nos vifs remerciements à toute personne ayant contribué de de près ou de loin à l'élaboration et la réussite de cette étude académique. De façon plus particulière, nous souhaitons exprimer notre profonde gratitude à Monsieur **KERKRI Abdelmounaim** pour son soutien qu'il n'a cessé de nous prodiguer tout au le long de la période de réalisation du projet, pour sa générosité en termes de formation et de partage de connaissances, ainsi que pour cette expérience prometteuse en tant que futures ingénieures.

TABLE DE MATIERES

1. Introduction :

En décembre 2019, l'Organisation mondiale de la santé (OMS) a identifié le nouveau coronavirus (COVID-19) comme la cause de la pneumonie à Wuhan, en Chine, et le 11 mars 2020 l'OMS a déclaré que COVID-19 était une pandémie. Entre le 31 décembre 2019 et le 4 mai en 2020, plus de 184 pays (parmi eux le Maroc) ont adopté des mesures strictes pour limiter la propagation du COVID-19, telles que les restrictions de confinement et la période de quarantaine, qui ont conduit à des facteurs socio-économiques, environnementaux, et des problèmes de santé mentale. Parmi ces restrictions le travail à domicile, l'éducation en ligne (e-learning), les restrictions sociales et la fermeture des frontières. Même si les politiques de confinement ont contribué au contrôle et à la diminution de la propagation du COVID-19, ils ont également entraîné la détérioration de la santé mentale de la population mondiale.

Pendant la période de confinement, le Maroc, et à partir du 16 mars 2020, a opté pour l'enseignement en ligne comme mesure pour atténuer les pertes en temps scolaire, étant donné ce changement inattendu, les étudiants ont été face à s'adapter à ce nouveau mode d'enseignement, ce qui avait sans doute un impact sur leur état psychologique.

Pour modéliser les effets psychologiques du Covid-19 sur les étudiants de l'enseignement supérieur au Maroc, nous avons effectué cette étude qui vise à analyser l'état d'anxiété des étudiants en fonction de cinq facteurs : le sexe, l'établissement, l'année d'étude, l'âge et le type de résidence (seul ou bien avec famille).

Dans cette étude, nous avons travaillé avec des techniques de Machine Learning pour classifier les étudiants afin de distinguer ceux qui souffrent d'anxiété, pour cela, nous avons utilisé l'algorithme de **XGBoost** pour construire un modèle de classification binaire. D'autre part, nous avons modélisé les soucis et les inquiétudes des étudiants à travers l'algorithme de **LDA** utilisé dans le traitement de texte.

2. La collecte des données :

2.1. Description du questionnaire :

Notre questionnaire a été destiné seulement aux étudiants de l'enseignement supérieur au Maroc, pour cette raison nous avons créé un formulaire à l'aide de Google Forms, et nous l'avons divisé en deux sections :

- La première section est composée d'une seule question « **Est-ce que vous êtes étudiant(e) ?** » à partir de laquelle le participant sera redirigé vers la deuxième section s'il répond par **« oui »**, sinon, il arrivera vers la fin du questionnaire avec le message « **Désolé, cette étude est destinée aux étudiants, merci pour votre participation** ».
- La deuxième section est composée du test de **Zung « Self-Rating Anxiety Scale (SAS) »** et des informations démographiques à savoir l'âge, le sexe, l'année d'études, l'établissement actuel et le type de résidence pendant la pandémie, et à la fin, une question ouverte pour savoir les soucis et les inquiétudes des participants en tant qu'étudiants.

2.2. Résultats du questionnaire :

Cette enquête a été réalisée entre le 28 mars et le 31 mai 2022, à travers la distribution du questionnaire au niveau des réseaux sociaux **(WhatsApp, Instagram, Facebook, LinkedIn).** Cette collecte a abouti à un échantillon de 721 participants dont 710 sont des étudiants provenant de différents établissements supérieurs au Maroc, avec un total de 522 femmes et 188 hommes.

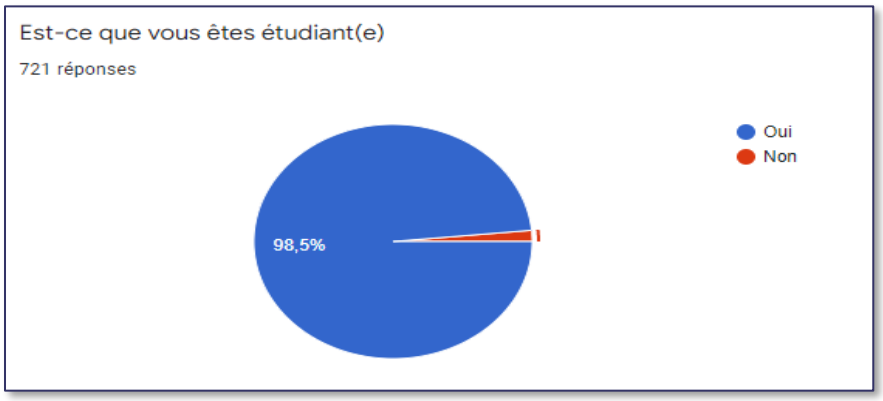

Fig.1 : Pourcentage des étudiants participants à l'enquête

Dans cette étude, nous avons évalué l'état d'anxiété à l'aide du **(SAS),** un test d'auto-évaluation réalisé par le psychiatre américain **William W.K Zung,** ce test est composé de 20 questions conçues pour mesurer les niveaux d'anxiété des patients ayant éprouvé des symptômes liés à l'anxiété, l'évaluation se fait par une échelle de 4 points à chaque réponse allant de 1 pour « rarement » vers 4 pour « très souvent ».
Les questions de 1 à 5 caractérisent les indicateurs émotionnels de l'anxiété, alors que les questions de 6 à 20 sont liées aux symptômes physiques de l'anxiété. (Voir l'annexe)

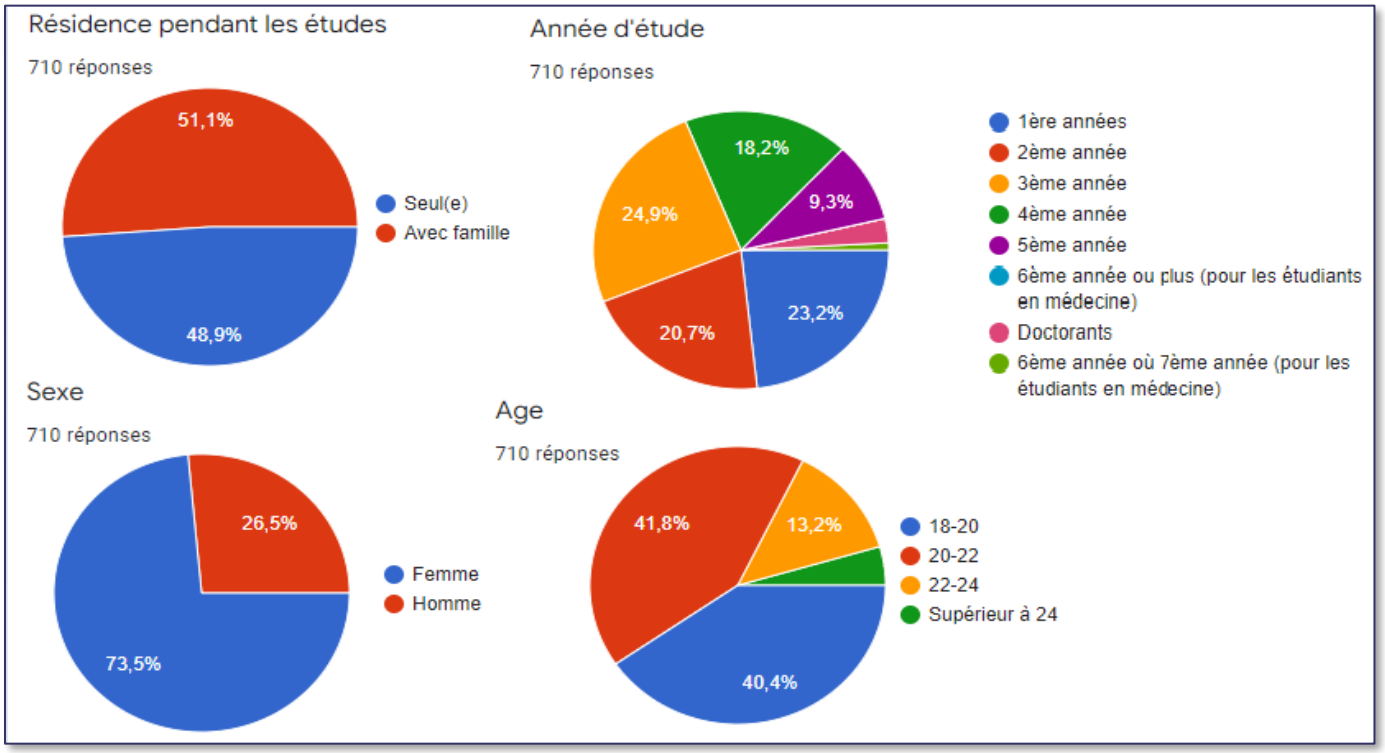

Fig.2 : Distribution des participants en fonction des informations démographiques

Pour mesurer la cohérence interne des questions du test, nous avons calculé le coefficient alpha de Cronbach qui nous a donné une valeur de 0.8 ce qui montre une forte cohérence des 20 items.

```
(0.8000947557277867, array([0.778, 0.821]))
```

Fig.3: Coefficient α de Cronbach

3. Préparation des données :

3.1. Data Cleaning:

Le Data Cleaning a été réalisé en 5 étapes :

Etape 1 :

Nous avons commencé par la suppression des réponses « **Non** » de la question de départ et puis après la suppression des colonnes inutiles.

Etape 2 :

Dans cette étape nous avons renommé l'entête de notre Dataset afin de faciliter la manipulation des variables.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	...	Q18	Q19	Q20	sexe	etablissement	annee	residence	age	score	anxiete
0	2	3	2	4	1	2	4	4	2	4	...	3	1	1	Femme	Ecole d'ingénieur	2ème année	Seul(e)	20-22	61.25	existence d'anxiété
1	4	4	4	4	1	4	4	4	4	4	...	2	1	4	Femme	Ecole d'ingénieur	4ème année	Avec famille	20-22	70.0	existence d'anxiété
2	2	1	2	2	4	1	2	2	1	1	...	1	2	1	Homme	Ecole d'ingénieur	3ème année	Seul(e)	20-22	42.5	état normal
3	1	1	1	1	2	1	1	1	2	1	...	1	4	1	Femme	Ecole d'ingénieur	3ème année	Seul(e)	20-22	35.0	état normal
4	1	2	4	1	2	3	2	1	4	3	...	1	2	2	Femme	Ecole d'ingénieur	3ème année	Seul(e)	20-22	47.5	état normal
5	2	1	1	1	3	2	3	2	1	2	...	2	2	1	Femme	Ecole d'ingénieur	3ème année	Seul(e)	20-22	45.0	état normal
6	2	1	1	2	2	1	1	1	1	1	...	2	2	2	Homme	Ecole d'ingénieur	4ème année	Seul(e)	20-22	38.75	état normal

Fig.4 : Tableau de données

Etape 3 :

Cette étape est consacrée au calcul du score d'anxiété obtenu par de la multiplication de la somme des notes données aux 20 questions par 1.25, à partir de la valeur obtenue nous avons classifié les étudiants en deux catégories :

- Etat normal pour ceux avec un score d'anxiété entre 25 et 49
- Existence d'anxiété pour ceux avec un score d'anxiété supérieur à 50

```
anxiete
état normal              354
existence d'anxiété      318
dtype: int64
```

Fig.5 : Effectif des classes (état normal et existence d'anxiété)

Etape 4 :

Le formulaire a été rempli également par des étudiants du lycée ce qui nous a posé un problème au niveau de la colonne de l'année d'étude parce que nous n'avons mentionné que les années d'études correspondantes à l'enseignement supérieur, pour cette raison et pour ne pas faire confondre le modèle de classification, nous avons supprimé les observations des lycéens.

Etape 5 :

Dans la dernière étape nous avons regroupé les établissements en 12 catégories suivantes :

```
ECOLE D'INGENIEUR                                          284
FACULTE DES SCIENCES                                        73
FACULTE DES SCIENCES ECONOMIQUES JURIDIQUES ET SOCIALES     71
ECOLE DE COMMERCE ET DE GESTION                             58
ECOLE SUPERIEURE DE TECHNOLOGIE                             44
FACULTE DE MEDECINE                                         43
CLASSES PREPARATOIRES AUX GRANDES ECOLES                    25
FACULTE DES LETTRES ET DES SCIENCES HUMAINES                25
INSTITUT TECHNICIENS SPECIALISES                            22
FACULTE DES SCIENCES ET TECHNIQUES                          14
ETABLISSEMENT DE PROFESSIONS INFIRMIERES                     8
ETABLISSEMENT DES METIERS D'EDUCATION                        5
Name: etablissement, dtype: int64
```

Fig.6 : Effectif des étudiants en fonction des établissements

3-2. Prétraitement des données :

Afin de pouvoir travailler avec les données collectées et nettoyées, nous devons effectuer un prétraitement pour les rendre exploitables par le modèle. Tout d'abord, nous avons séparé les colonnes du jeu de données en variable prédictive **X** contenant les 5 attribues (l'âge, le sexe, l'année d'études, l'établissement et le type de résidence) et la variable prédite **Y** représentant l'existence ou non d'anxiété.

```
array([['Femme', "ECOLE D'INGENIEUR", '2ème année', 'Seul(e)', '20-22'],
       ['Femme', "ECOLE D'INGENIEUR", '4ème année', 'Avec famille',
        '20-22'],
       ['Homme', "ECOLE D'INGENIEUR", '3ème année', 'Seul(e)', '20-22'],
       ...,
       ['Femme', "ECOLE D'INGENIEUR", '3ème année', 'Avec famille',
        '22-24'],
       ['Homme',
        'FACULTE DES SCIENCES ECONOMIQUES JURIDIQUES ET SOCIALES',
        '2ème année', 'Avec famille', '18-20'],
       ['Femme', "ECOLE D'INGENIEUR", '1ère années', 'Avec famille',
        '18-20']], dtype='<U57')
```

Fig.7 : Variable prédictive X

```
array(["existence d'anxiété", "existence d'anxiété", 'état normal',
       'état normal', 'état normal', 'état normal', 'état normal',
       'état normal', "existence d'anxiété", "existence d'anxiété",
       'état normal', 'état normal', "existence d'anxiété", 'état normal',
       "existence d'anxiété", 'état normal', 'état normal', 'état normal',
       "existence d'anxiété", 'état normal', "existence d'anxiété",
       'état normal', 'état normal', 'état normal', "existence d'anxiété",
       'état normal', 'état normal', "existence d'anxiété",
       "existence d'anxiété", 'état normal', "existence d'anxiété",
       'état normal', 'état normal', 'état normal', 'état normal',
       "existence d'anxiété", 'état normal', 'état normal', 'état normal',
       'état normal', 'état normal', 'état normal', 'état normal',
       'état normal', 'état normal', "existence d'anxiété", 'état normal',
       'état normal', "existence d'anxiété", 'état normal',
       "existence d'anxiété", 'état normal', "existence d'anxiété",
       "existence d'anxiété", 'état normal', 'état normal',
       "existence d'anxiété", "existence d'anxiété",
       "existence d'anxiété", 'état normal', 'état normal', 'état normal',
       "existence d'anxiété", 'état normal', "existence d'anxiété",
```

Fig.8 : Variable prédite Y

Nous constatons que les variables (prédictive et prédite) sont de format catégoriel, cependant, **XGBoost** ne supporte que les variables de format numérique. Pour cette raison nous avons converti les valeurs catégorielles en valeurs numériques à l'aide du **LabelEncoder** permettant d'encoder les valeurs de la variable **Y** avec des nombres entiers allant de 0 au n_classes-1. D'autre part, nous avons utilisé le **OneHotEncoder** basé sur les dummies pour transformer les valeurs de la variable **X.**

```
array([0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1,
       1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1,
       1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0,
       1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1,
       1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0,
       1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0,
       1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1,
       1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0,
       1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1,
       1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0,
       1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1,
       0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0,
       1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1,
       0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0,
       1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0,
       1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0,
       1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0,
       1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1,
       0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0,
       0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1,
       0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0,
       1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1])
```

Fig.9 : La variable Y encodée

	0	1	2	3	4	5	6	7	8	9	...	17	18	19	20	21	22	23	24	25	26
0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0
1	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0
2	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0
3	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0
4	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
667	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0
668	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0
669	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0
670	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0
671	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0

Fig.10 : La variable X encodée

A la fin, nous avons réparti nos données en un ensemble d'apprentissage et un ensemble de test à l'aide de la fonction **train_test_split** fournie par la bibliothèque **Scikit-learn** avec un **test_size** = 0.2 (la proportion de l'ensemble de données à inclure dans la répartition de test).

4. Théorie :

4.1. Gradient Boosting :

Le **Boosting** est une technique séquentielle qui fonctionne sur le principe d'un ensemble. Elle combine un ensemble d'apprenants faibles et offre une précision de prédiction améliorée. À tout instant t, les résultats du modèle sont pesés en fonction des résultats de l'instant t-1 précédent. Les résultats prédits correctement ont un poids inférieur et ceux qui ne sont pas classés sont pondérés plus haut. Notez qu'un apprenant faible est un apprenant légèrement meilleur que des suppositions aléatoires. Par exemple, un arbre de décision dont les prédictions sont légèrement supérieures à 50%. L'illustration suivante montre de manière générale le fonctionnement de cette technique :

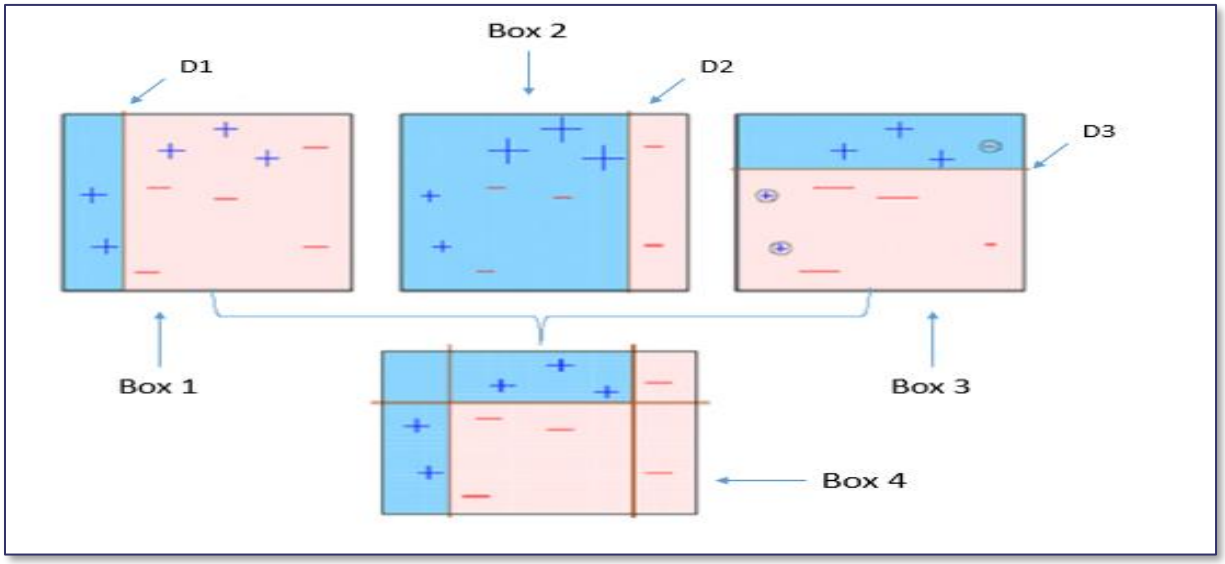

Fig.11 : Fonctionnement du Boosting

Quatre classificateurs (en 4 cases), illustrés ci-dessus, tentent de classer les classes (+) et (-) de la manière la plus homogène possible.

1. Box 1 : Le premier classificateur (généralement un stump de décision) crée une ligne verticale (split) à D1. Il classifie tout ce qui se trouve à gauche de D1 comme (+) et tout ce qui se trouve à droite de D1 comme (-). Cependant, ce classificateur nous donne trois points (+) mal classés.
2. Box 2 : Le deuxième classificateur donne plus de poids aux trois points (+) mal classés (voir la plus grande taille de +) et crée une ligne verticale à D2. Encore une fois, il considère que tout ce qui se trouve à droite de D2 comme (-) et à gauche comme (+). Pourtant, il fait des erreurs en classant incorrectement trois points (-).
3. Box 3 : Encore une fois, le troisième classificateur donne plus de poids aux trois points mal classés et crée une ligne horizontale à D3. Pourtant, ce classificateur ne parvient pas à classer correctement les points (dans les cercles).

4. Box 4 : Il s'agit d'une combinaison pondérée des classificateurs faibles (encadrés 1, 2 et 3). Comme vous pouvez le voir, il fait un bon travail pour classer correctement tous les points.

Le **Gradient Boosting** est une approche où de nouveaux modèles sont créés qui prédisent les résidus ou les erreurs des modèles précédents, puis additionnés pour faire la prédiction finale. Il est appelé Gradient Boosting car il utilise un algorithme de descente de gradient pour minimiser la perte lors de l'ajout de nouveaux modèles. Cette approche prend en charge à la fois les problèmes de modélisation prédictive de régression et de classification.

4.1.1. Fonctionnement du Gradient Boosting :

Le **Gradient Boosting** implique les trois éléments suivants :

1. Une fonction coût à minimiser.
2. Un « weak learner » pour faire des prédictions.
3. Un modèle additif pour ajouter des « weak learners » afin de minimiser la fonction coût.

Le **Gradient Boosting** est un algorithme qui peut surajuster rapidement un ensemble de données d'entraînement. Il peut bénéficier de méthodes de régularisation qui pénalisent diverses parties de l'algorithme et améliorent généralement la performance de l'algorithme en réduisant le surajustement.

4.2. XGBoost (eXtreme Gradient Boosting) :

XGBoost est une bibliothèque distribuée optimisée de Gradient Boosting conçue pour être très efficace, flexible et portable. Il implémente des algorithmes Machine Learning sous le framework de Gradient Boosting. XGBoost fournit un Boosting d'arbre parallèle (également connu sous le nom de GBDT, GBM) qui résout de nombreux problèmes de Data Science de manière rapide et précise. Le même code s'exécute sur un environnement distribué majeur (Hadoop, SGE, MPI) et peut résoudre des problèmes au-delà de milliards d'exemples.

Il est particulièrement performant :

- Dans sa capacité à généraliser car intégrant dans sa construction des mécanismes de **régularisation** assez puissants et astucieux
- Sa rapidité de calcul sur des gros volumes en faisant des approximations élégantes lors de la construction des arbres de décision.

Dans la construction des arbres, **XGBoost** intègre des mécanismes de régularisation permettant d'avoir des arbres relativement simples « weak learners » pour éviter l'overfitting et permettre une généralisation sur des données inconnues du modèle avec un bon niveau de performance. Pour une observation, chaque arbre donne un résultat, et la prédiction finale est obtenue en additionnant chacune des valeurs obtenues données par les arbres.

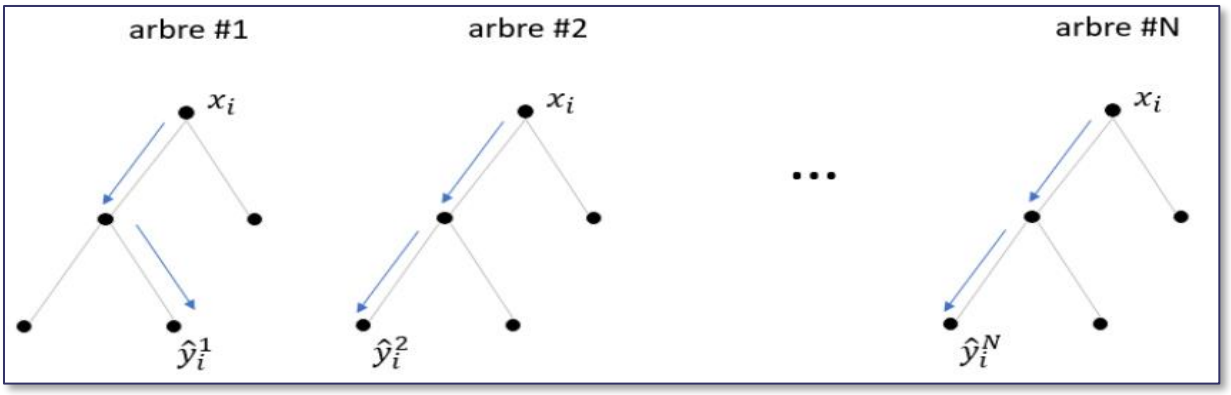

Fig.12 : un schéma avec N arbres avec leur prédiction

A chaque arbre #t : x_i ➡ $\hat{y}_i^t$

Au final, la prédiction sera égale à l'addition de toutes les prédictions :

$$\hat{y}_i=\phi(x_i) = \sum_{t=1}^{N} \varphi_t\,(x_i) = \sum_{t=1}^{N} \hat{y}_i^{(t)}$$

Φ : La fonction globale pour la prédiction (régression ou classification)
φ_t : La fonction de prédiction de l'arbre #t

Phase d'apprentissage et la fonction objective :

Lors de la phase d'apprentissage, **XGBoost** force la fonction globale objective à contenir ces deux caractéristiques :

- Une Loss fonction, mesurant l'écart entre valeurs prédites et cibles. Cette fonction doit être différentiable et convexe (afin de pouvoir être optimisée facilement et atteindre ainsi son minimum)
- Une fonction pour pénaliser la complexité du modèle, afin d'éviter l'overfitting (sur-apprentissage)

$$\mathcal{L} = \underbrace{\sum_{i}^{\#\text{instances}} l(\hat{y}_i, y_i)}_{\text{Loss}} + \underbrace{\sum_{k}^{\#\text{arbres}} (\gamma . T_k + \frac{1}{2} . \lambda . \| w_k \|^2)}_{\text{Modèle de pénalisation}}$$

$\hat{y}_i$: $Prédiction\ globale.$
y_i:$Valeur\ cible.$
$\#instances$:$nombre\ de\ données\ utilisées\ lors\ de\ la\ phase\ d'apprentissage.$
T: $Nombre\ de\ feuilles\ de\ l'arbre.$
$\|w\|^2$: $Norme\ L2\ des\ valeurs\ de\ chaque\ feuille\ des\ arbres.$
$\gamma\ et\ \lambda$: Sont des hyper-paramètres du modèle global.

La fonction de pénalisation a pour but de limiter les arbres qui ont un grand nombre de feuilles ou des valeurs de prédiction importantes

4.2.1. Les hyperparamètres XGBoost :

XGBoost requiert un certain nombre paramètres de réglage qui peuvent être distinguées selon :

- Les hyperparamètres liés au calcul numérique, dont les exécutions asynchrones ;
- Les hyperparamètres des arbres, tels que la profondeur maximale ou le nombre d'observations minimal dans un nœud ;
- Les hyperparamètres propres à l'optimisation, comme le taux d'apprentissage ou la fonction objectif.

Parmi les hyperparamètres les plus utilisés :

- learning_rate : utilisé pour éviter le surajustement, il prend une valeur entre 0 et 1.
- max_depth : détermine la profondeur de chaque arbre.
- subsample : pourcentage d'échantillons utilisés par arbre.
- colsample_bytree : pourcentage des variables utilisées par arbre.
- n_estimators : nombre d'arbres à construire.
- objective : détermine la fonction de perte à utiliser comme 'reg:linear' pour les problèmes de régression, 'reg:logistic' pour les problèmes de classification avec décision seulement, 'binary:logistic' pour les problèmes de classification avec probabilité.

XGBoost prend également en charge les paramètres de régularisation pour pénaliser les modèles à mesure qu'ils deviennent plus complexes et les réduire à des modèles simples :

- gamma : contrôle si un nœud donné se divise en fonction de la réduction attendue de la perte après le split. Une valeur plus élevée conduit à moins de fractionnements. Pris en charge uniquement pour les tree-based learners.
- alpha : régularisation L1 sur le poids des feuilles. Une valeur importante conduit à plus de régularisation.
- lambda : régularisation L2 sur le poids des feuilles, c'est plus lisse que la régularisation L1.

4.3. Modèle de classification :

Scikit-learn, également connue sous le nom de sklearn, est une bibliothèque open source pour le Machine Learning basée sur Python qui prend en charge quatre algorithmes de Machine Learning : classification, régression, réduction et clustering. Nous avons utilisé le classificateur de XGBoost '**XGBClassifier**' offert par sklearn.

Nous avons ajusté les hyperparamètres de XGBClassifier pour construire le meilleur modèle. Nous avons réglé XGBoost pour faire une classification binaire à l'aide de l'objectif **'binary:logistic'** et nous avons, également réglé '**n_estimators', 'max_depth', 'learning_rate', 'subsample', 'colsample_bynode' et 'max_delta_step'** comme le montre le tableau ci-dessous, tous les autres paramètres qui ne figurent pas dans le tableau sont les valeurs par défaut.

Hyperparamètre	Signification	Valeur
max_depth	Profondeur de chaque arbre	6
learning_rate	Taux d'apprentissage	0.02
subsample	Pourcentage d'échantillons utilisés par arbre	0.1
colsample_bynode	Rapport sous-échantillon par nœud	0.7
max_delta_step	Pas maximum de delta autorisé à chaque sortie de feuille	0.8
objective	La fonction de perte à utiliser	binary:logistic

Tableau.1 : Hyperparamètres ajustés

Enfin, et pour éviter le surajustement, nous avons le **'early_stopping_rounds'** permettant d'arrêter l'entrainement du modèle une fois que les performances sur le jeu de données de test ne se sont pas améliorées après un nombre fixe d'itérations d'entraînement, dans notre cas nous avons fixé ce nombre à 10.

4.4. Traitement de texte :

4.4.1. Nuage de mots :

Word Cloud est une technique de visualisation de données utilisée pour représenter des données textuelles dans lesquelles la taille de chaque mot indique sa fréquence ou son importance. Des points de données textuelles significatifs peuvent être mis en évidence à l'aide d'un nuage de mots.

Pour générer un nuage de mots en Python, nous avons utilisé le package **WordCloud** qui nous permet d'illustrer les mots les plus répétés par les étudiants en répondant à la question ouverte.

4.4.2. LDA (Latent Dirichlet Allocation) :

La modélisation de sujet est un type de modélisation statistique permettant de découvrir les « **sujets** » abstraits qui se produisent dans une collection de documents. **LDA** est un modèle probabiliste génératif qui suppose que chaque sujet est un mélange basé sur un ensemble sous-jacent de mots, et chaque document est un mélange de plus d'un ensemble de probabilités de sujet.

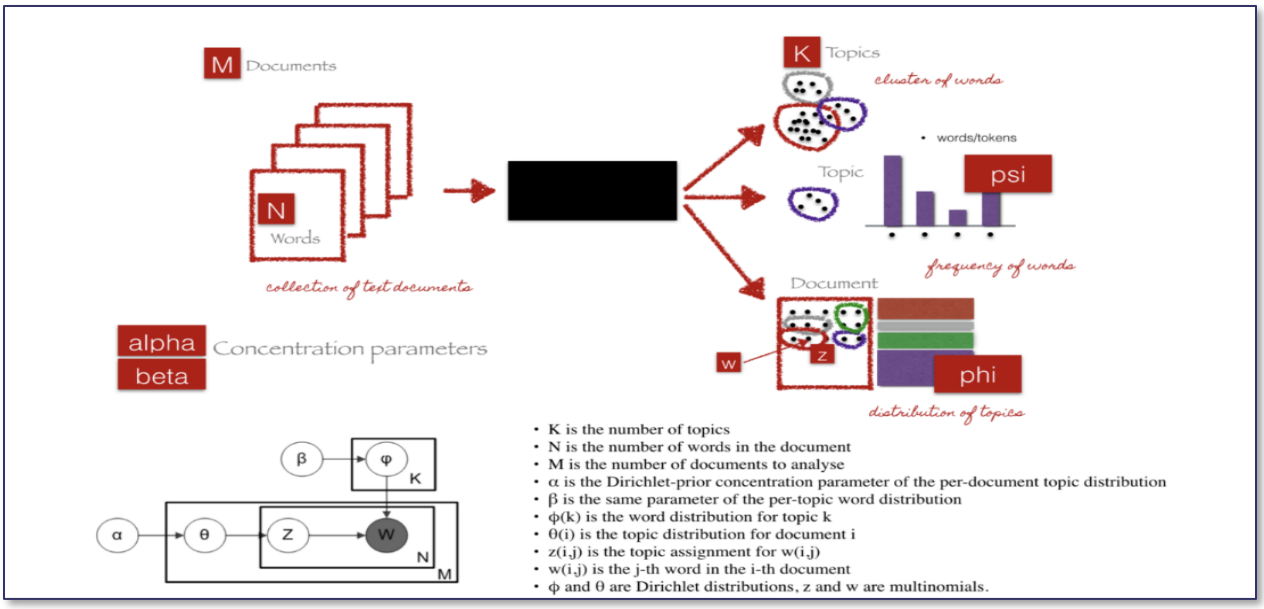

Fig.13 : Fonctionnement de LDA

Nous pouvons décrire le processus génératif de **LDA** comme, compte tenu du nombre M de documents, du nombre N de mots et du nombre K de sujets antérieurs, le modèle s'entraîne à produire :

- **psi** : la distribution des mots pour chaque sujet K
- **phi** : la distribution des sujets pour chaque document i

Paramètres de LDA :

- Le paramètre **alpha** est le paramètre de concentration antérieur de Dirichlet qui représente la densité document-sujet. Avec un alpha plus élevé, les documents sont supposés être composés de plus de sujets et entraîner une distribution plus spécifique des sujets par document.

- Le paramètre **bêta** est le même paramètre de concentration antérieur qui représente la densité sujet-mot. Avec un bêta élevé, les sujets sont supposés être constitués de la plupart des mots et entraîner une distribution de mots plus spécifique par sujet.

Après avoir entrainé le modèle, nous avons visualisé les sujets pour l'interprétabilité. Pour ce faire, nous avons utilisé un package de visualisation appelé **pyLDAvis,** conçu pour nous aider de manière interactive à:

- Mieux comprendre et interpréter les sujets individuels, en sélectionnant manuellement chaque sujet pour afficher ses termes les plus fréquents et/ou les plus « **pertinents** », en utilisant différentes valeurs du paramètre λ. Cela peut être utile lorsque nous essayons d'attribuer un nom interprétable par l'homme ou une « **signification** » à chaque sujet.

- Mieux comprendre les relations entre les sujets par l'exploration du graphe **Intertopic Distance** qui peut nous aider à en apprendre davantage sur la façon dont les sujets sont liés les uns aux autres.

5. Résultats :

5.1. Analyse descriptive :

5.1.1. Distribution du score selon les facteurs démographiques :

Afin d'effectué l'analyse descriptive du score, nous avons utilisé en premier lieu la boite à moustaches :

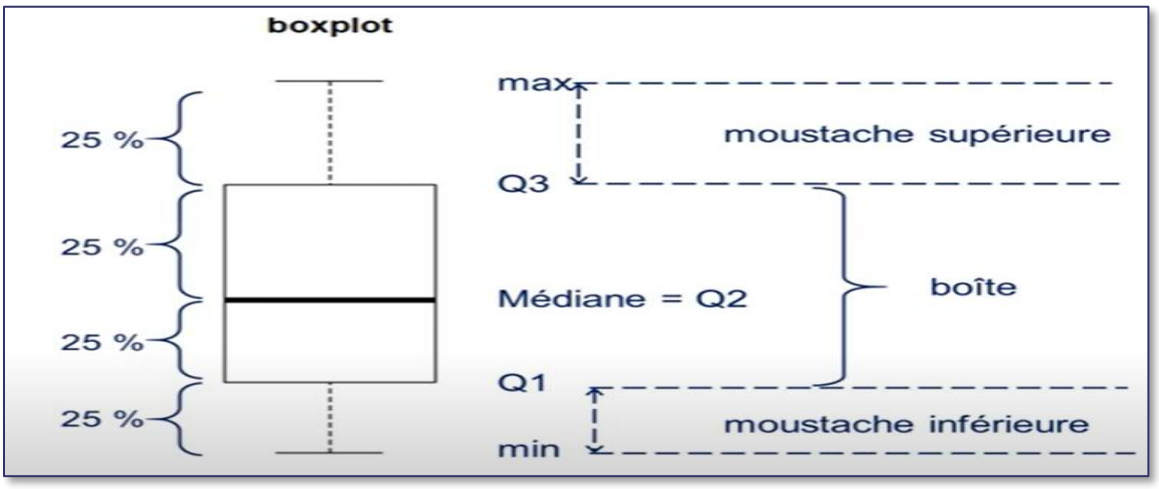

Fig.14 : la boite à moustaches

La boîte à moustaches est une méthode efficace de présentation graphique de données numériques, elle nous donne une synthèse des données en cinq informations cruciales identifiables en un coup d'œil : la mesure de position, la dispersion, l'asymétrie et la longueur de la moustache (le minimum, le quartile 1 (25%), la médiane (50%), le quartile 3 (75%) et le maximum).

<u>Interprétation des résultats :</u>

- **Boite à moustaches du sexe :**

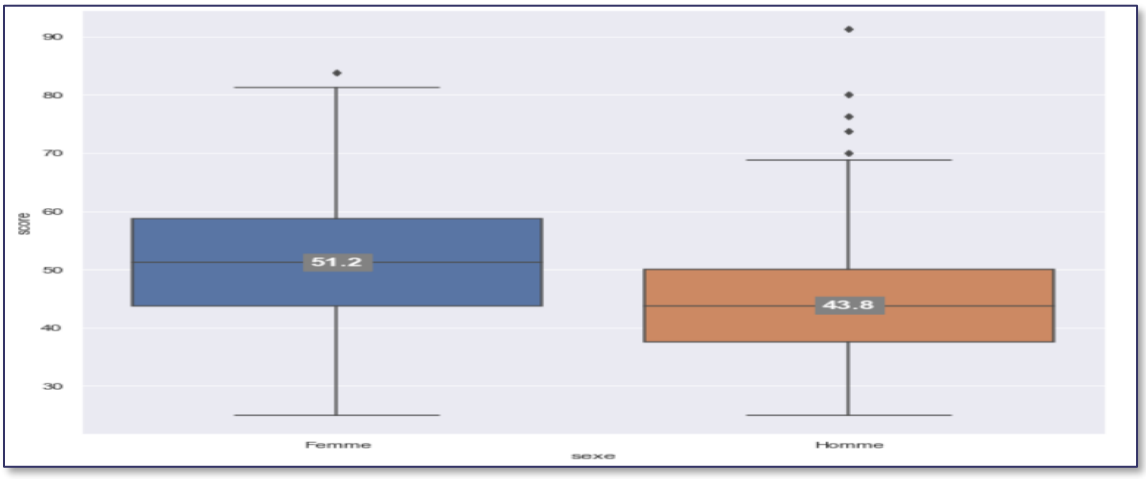

Fig.15 : Comparaison de la distribution du score selon le sexe

D’après la figue 12 : la distribution des scores des étudiants selon le sexe nous montre une différence de score d’anxiété entre Femme et Homme (une médiane de 51.2 pour le groupe Féminin et 43.8 pour le groupe Masculin) il est clair que le taux d’anxiété est plus élevé chez les femmes avec un maximum égale 82. Le nombre des points atypiques pour les hommes est plus élevé que les femmes. De plus, nous constatons une symétrie dans les deux cas.

- **La boite à moustaches du le type de résidence :**

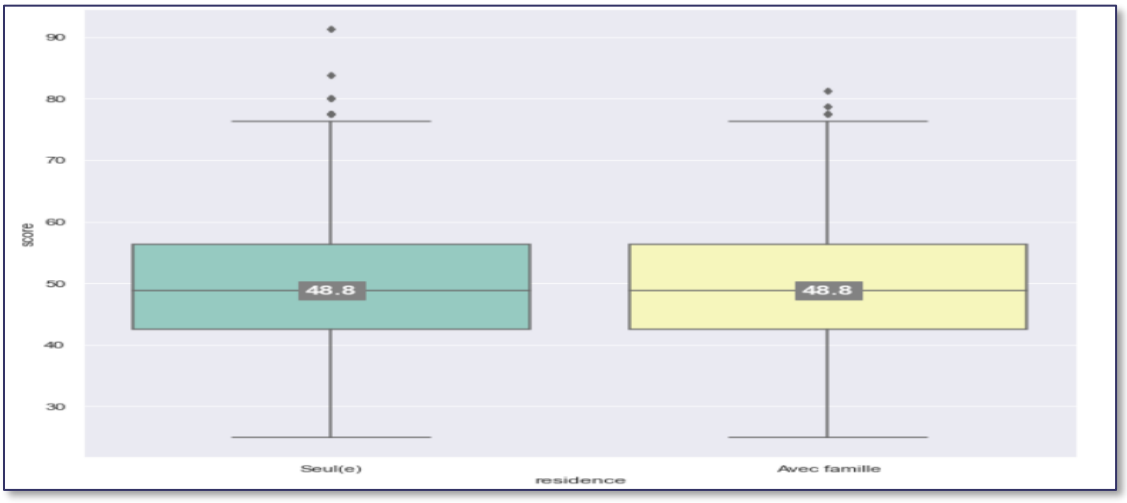

Fig.16 : Comparaison de la distribution du score selon le type de résidence

D’après la figure 13 : la distribution des scores des étudiants selon le type de résidence nous montre une égalité de score d’anxiété entre la résidence seul(e) et avec famille avec quelques valeurs atypique pour les deux groupes. De même, il y a une symétrie dans les deux cas.

- **La boite à moustaches d’année d’étude :**

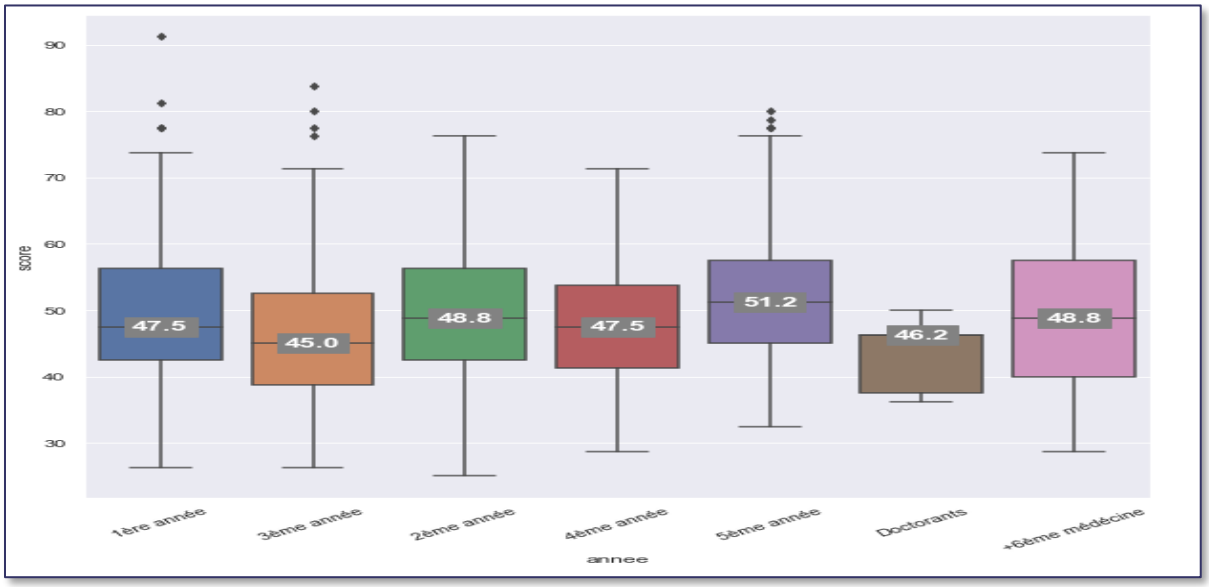

Fig.17 : Comparaison de la distribution du score selon l’année d’étude

D’après la figue 14 : la distribution des scores des étudiants selon l’année d’étude nous montre une différence de score d’anxiété entre les années d’étude. La valeur de la médiane est élevée pour chacune des 5éme année, 2éme année et 6éme année. Nous pouvons interpréter cette élévation par exemple pour la cinquième année par la pression causée par le stage de fin d’étude. Nous constatons aussi l’existence de quelques points atypiques dans la première, la deuxième et la cinquième année.

- **La boite à moustaches d’établissement :**

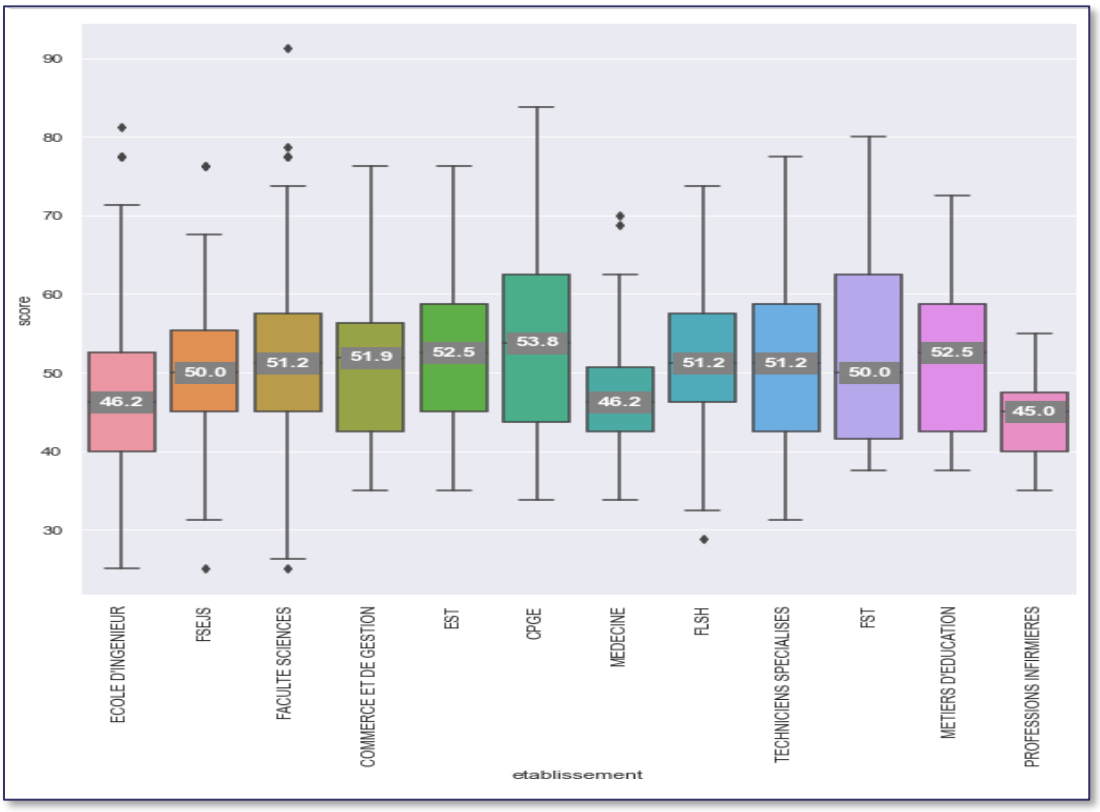

Fig.18 : Comparaison de la distribution du score selon l’établissement

D’après la figue 15 : la distribution des scores des étudiants selon l’établissement nous montre une différence de score d’anxiété entre les 12 établissements. Il est clair que les étudiants des classes préparatoire ont une valeur élevée de la médiane 53.8 avec un taux d’anxiété maximum égale à 84 et un écart interquartile plus étalé, ces résultats correspondent parfaitement à la nature des études au niveau des classes préparatoires et le stresse quotidien. Par contre, au niveau des établissements des professions infermières, les étudiants ont la plus petite valeur de médiane 45 avec un taux d’anxiété faible.

5.1.2 Distribution des classes d’anxiété selon les facteurs démographiques :

D’autre part, nous avons effectué quelques visualisations pour analyser la distribution des deux états d’anxiété en fonction des facteurs démographiques (l'âge, le sexe, l'année d'études, l’établissement et le type de résidence).

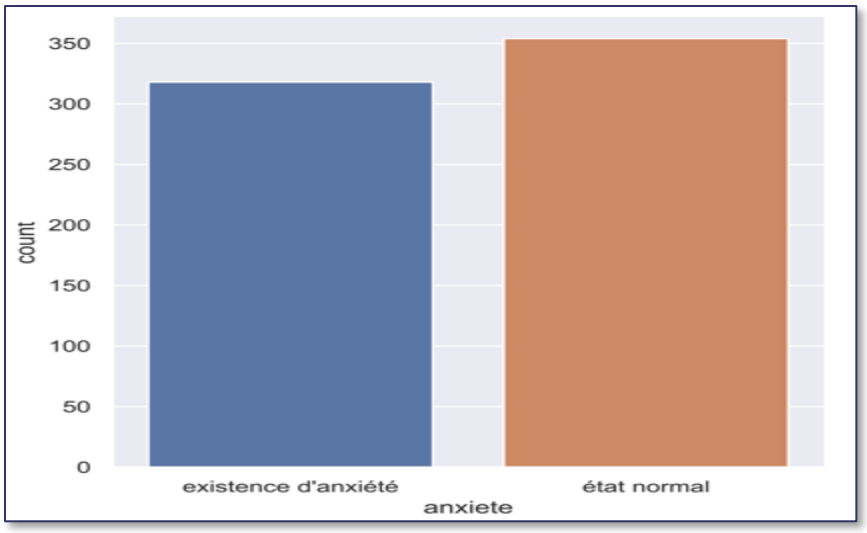

Fig.19 : Effectif au niveau de chaque classe d'anxiété

La figue 16 montre que l'effectif de la classe « état normale » est plus élevé que l'effectif de la classe « existence d'anxiété » avec une différence de 36 étudiants.

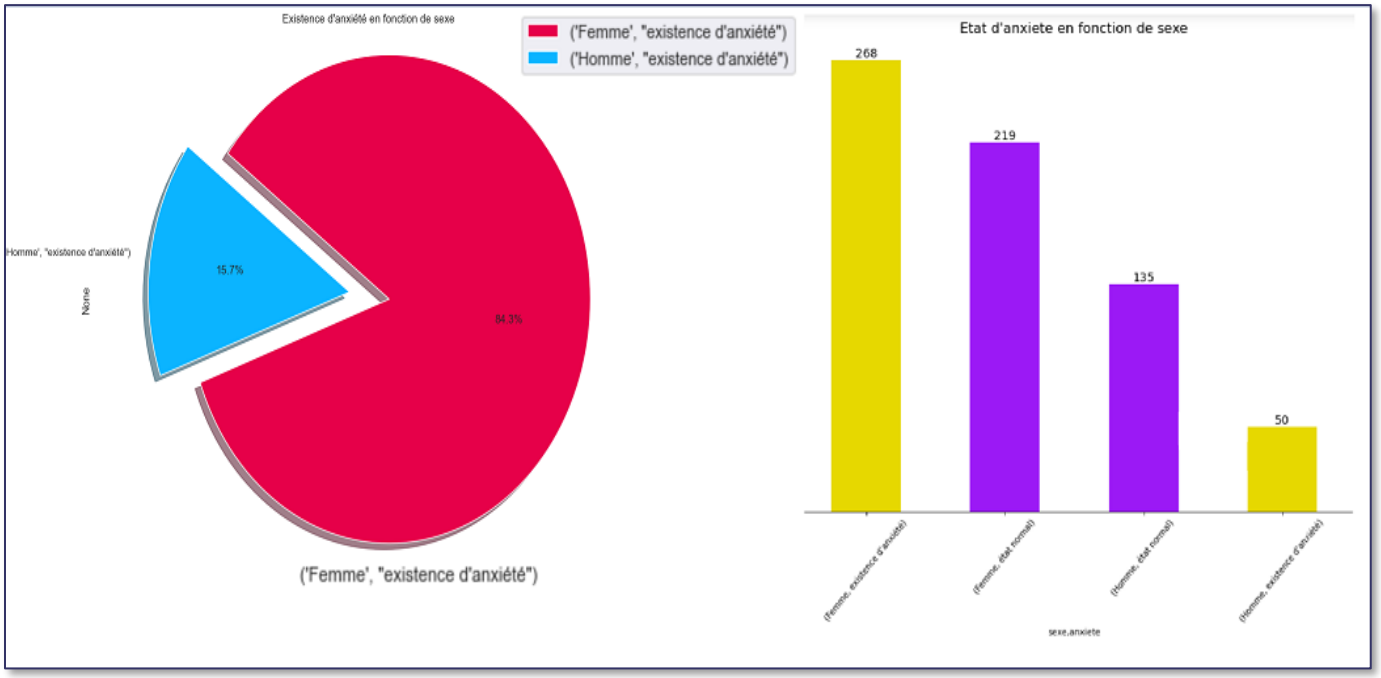

Fig.20 : Etat d'anxiété en fonction de sexe

Il est clair que l'état d'anxiété est plus élevé chez les femmes avec un pourcentage de 84.3%. En comparant le nombre total des cas « existence d'anxiété » nous constatons un nombre important chez les femmes avec un effectif de 268 en contrepartie un totale des cas « état normal » est élevé chez les hommes avec un effectif de 135.

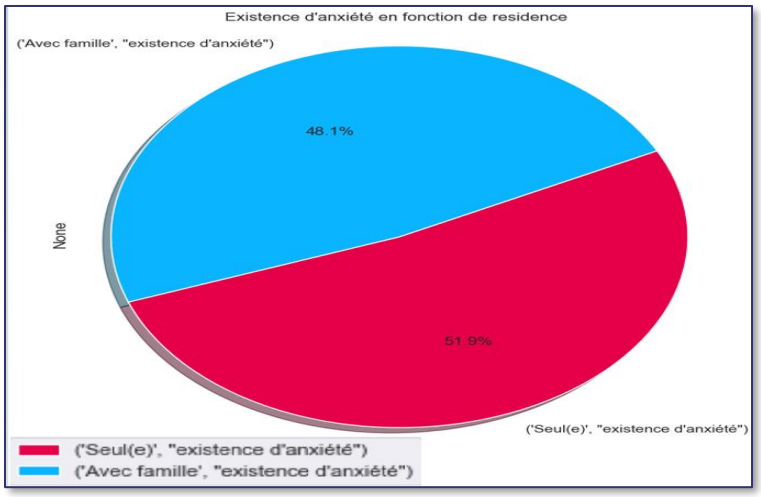

Fig.21 : L'état d'anxiété en fonction du type de résidence

D'après le graphe ci-dessus nous constatons que les valeurs de l'effectif d'existence d'anxiété dans les deux classes sont presque égales avec des pourcentages de 51.9% et 48.1% respectivement.

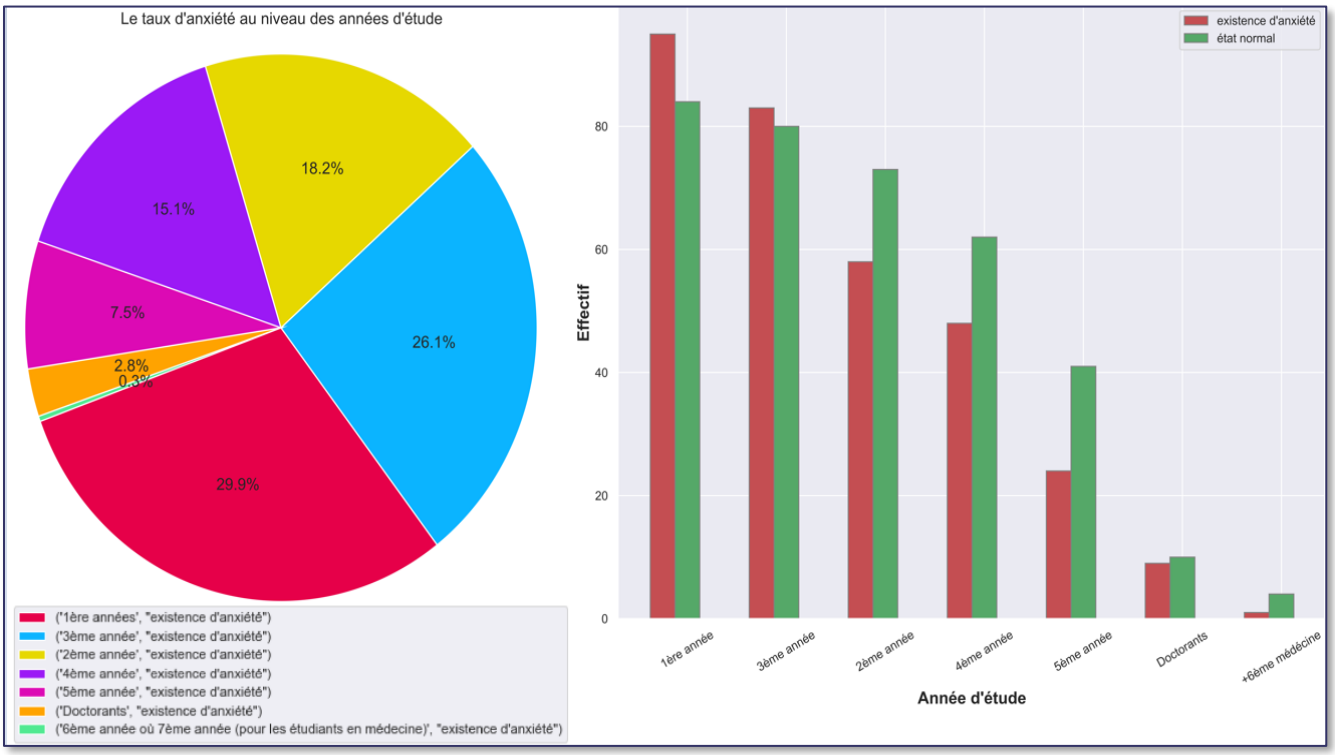

Fig.22 : L'état d'anxiété en fonction d'année d'étude

Nous constatons que l'effectif d'existence d'anxiété est supérieur à celui de l'état normal pour la première et la troisième année seulement. Cependant, pour les autres années d'étude, c'est celui de l'état normal qui est supérieur. D'autre part le taux d'anxiété est maximum en première année avec un pourcentage de 29.9% suivi par la troisième année avec un pourcentage de 26.1%, or, il est minimum en 6éme et 7éme année avec un pourcentage de 0.3%.

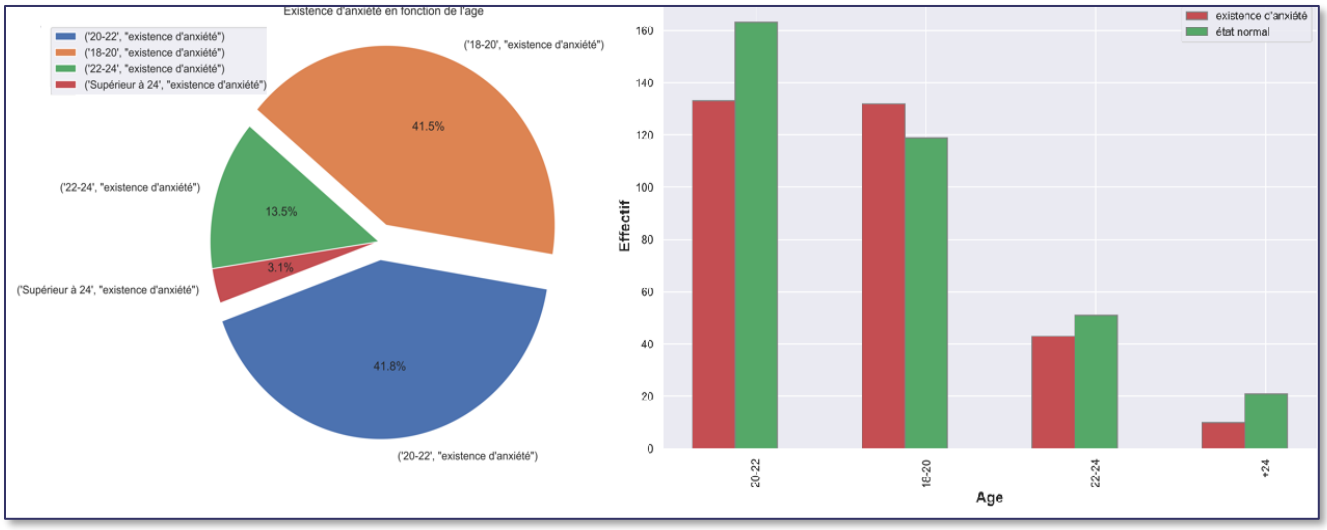

Fig.23: L'état d'anxiété en fonction d'âge

La tranche d'âge ayant un effectif des étudiants avec existence d'anxiété plus élevé à celui de ceux avec un état normal est de 18 à 20 ans avec un pourcentage 41.8%, et la tranche ayant un effectif des étudiants avec un état normal plus élevé à celui de ceux avec existence d'anxiété est de plus de 24 ans avec un pourcentage de 3.1%.

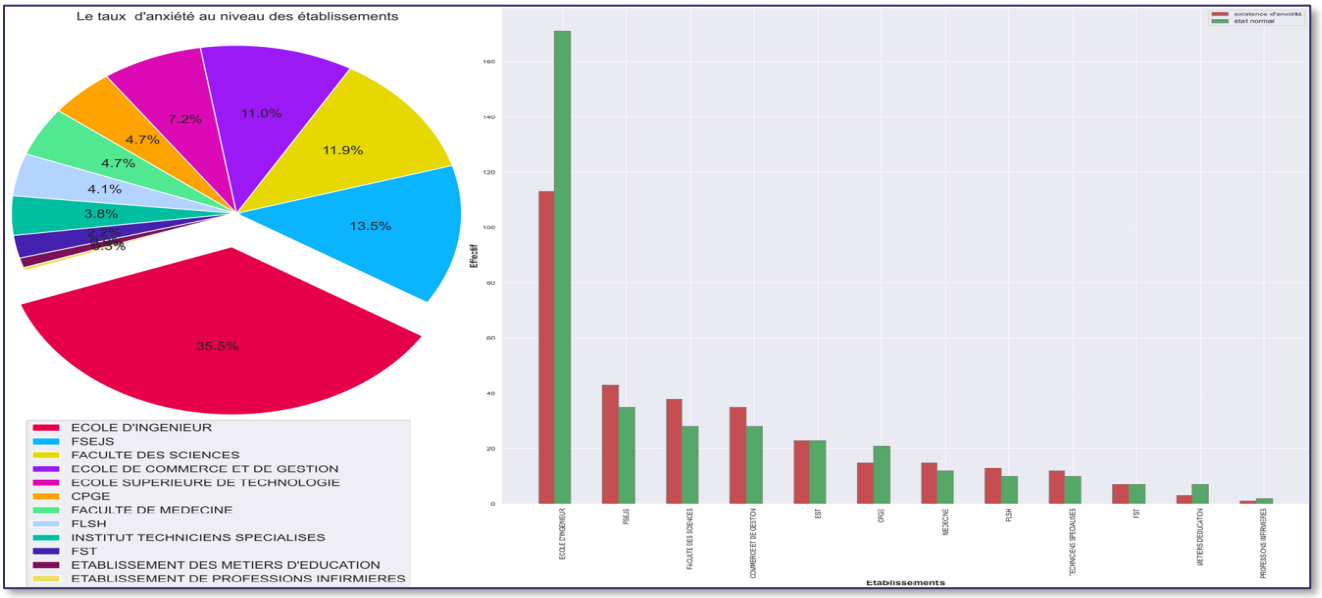

Fig.24 : L'état d'anxiété en fonction d'établissent

Nous observons que le taux d'anxiété dans les écoles d'ingénieurs est très important par rapport aux autres établissements. Par ailleurs, les établissements ayant un effectif des étudiants avec une existence d'anxiété plus élevé par à celui de ceux avec un état normal sont : la faculté des sciences économiques juridiques et sociales, la faculté des sciences, la faculté des lettres et sciences humaines, les établissements des techniciens spécialisés, les classes préparatoires aux grandes écoles, la faculté de médecine et les écoles de commerce et de gestion.

5.2. Corrélation entre les variables prédictives :

La matrice de corrélation indique les valeurs de corrélation, qui mesurent le degré de relation linéaire entre chaque paire de variables. Les valeurs de corrélation peuvent être comprises entre -1 et +1. Si les deux variables ont tendance à augmenter et à diminuer en même temps, la valeur de corrélation est positive. Si nous avons une forte corrélation entre deux variables cela influence notre étude.

La bibliothèque PyCorr de Python nous offre la fonction '**corr_matrix()'** et **'plot_corr()'** permettant de calculer et visualiser la matrice de corrélation de nos variables prédictives :

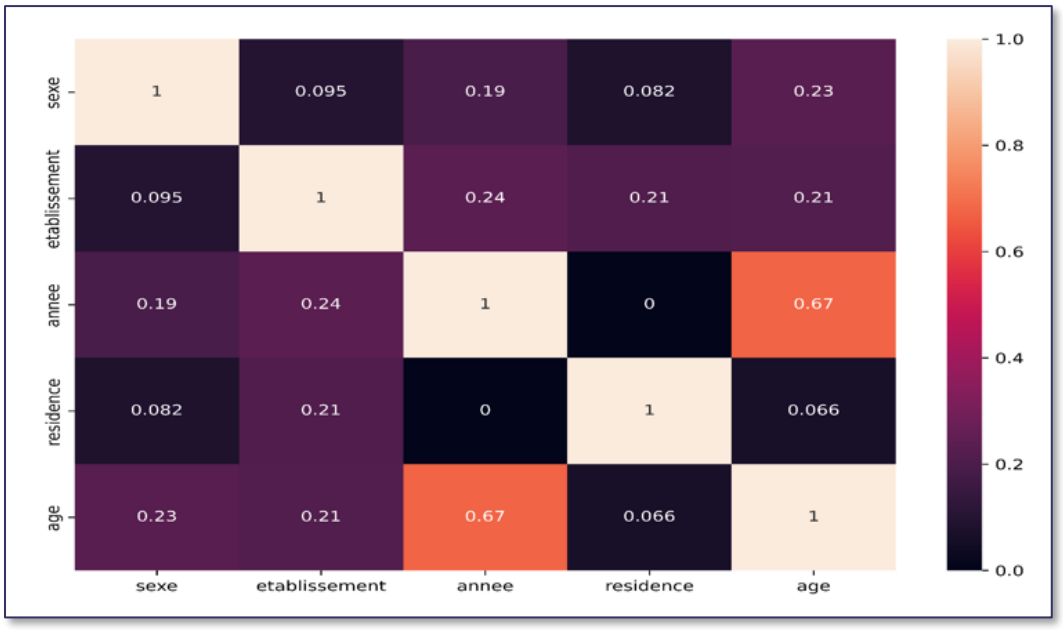

Fig.25: Matrice de corrélation des variables prédictives

Dans notre cas la matrice montre qu'il existe une corrélation positive entre l'année et l'âge avec une valeur de 0.67 mais cette valeur est faible il ne peut pas influencer sur notre étude.

5.3. Modèle de classification :

Afin d'évaluer la performance de notre modèle, nous avons utilisé un certain nombre d'outils que nous vous présenterons dans cette partie :

- **Précision :**

  La bibliothèque sklearn de Python offre une fonction pour le calcul de la précision du modèle appelée 'accuracy_score' qui prend comme arguments l'ensemble de test y_test et l'ensemble des valeurs prédites par le modèle de XGBoost.

  Dans notre cas, nous avons obtenu un taux de précision de **72.59%** avec un nombre maximal des arbres construites égal à 119 arbres.

➢ **Matrice de confusion :**

La matrice de confusion est en quelque sorte un résumé des résultats de prédiction pour un problème particulier de classification. Elle compare les données réelles pour une variable cible à celles prédites par un modèle.

Les résultats d'une matrice de confusion sont classés en quatre grandes catégories : **les vrais positifs, les vrais négatifs, les faux positifs et les faux négatifs**.

- Les vrais positifs ou TP (true positive) indiquent les cas où les prédictions et les valeurs réelles sont effectivement positives.
- Les vrais négatifs ou TN (true negative) indiquent par contre les cas où les prédictions et les valeurs réelles sont toutes les deux négatives.
- Les faux positifs ou FP (false positive) indiquent quant à eux une prédiction positive contraire à la valeur réelle qui est négative. Ils sont également considérés comme des erreurs de type 1.
- Les faux négatifs font référence aux cas où les prédictions sont négatives alors que les valeurs réelles sont positives. Ils sont également considérés comme des erreurs de type 2.

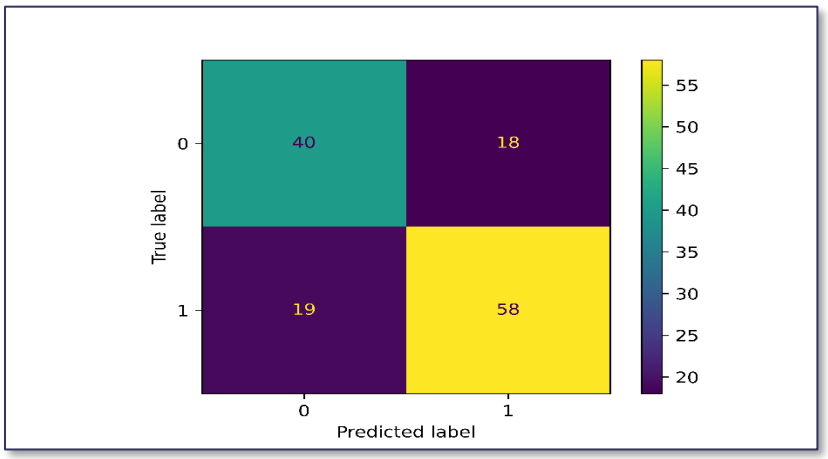

Fig.26 : Matrice de confusion

Notre matrice de confusion montre que 40 étudiants parmi 58 appartenant à la classe 0 ‘existence d’anxiété’ et 58 étudiants parmi 77 appartenant à la classe 1 ‘état normal’ sont bien classés. Cela montre que nous pouvons bien distinguer les cas d’anxiété en se basant sur les facteurs démographiques.

➢ **Rapport de classification :**

En plus des informations tirées par la matrice de confusion, le rapport de classification offert par sklearn indique les valeurs suivantes :

	precision	recall	f1-score	support
0	0.68	0.69	0.68	58
1	0.76	0.75	0.76	77
accuracy			0.73	135
macro avg	0.72	0.72	0.72	135
weighted avg	0.73	0.73	0.73	135

Fig.27 : Rapport de classification

- **La précision** est la capacité d'un classificateur à ne pas étiqueter une instance positive qui est réellement négative. Pour chaque classe, il est défini comme le rapport entre les vrais positifs et la somme des vrais et des faux positifs**.** Dans notre cas, 68% des étudiants avec existence d'anxiétéé, et 76% des étudiants avec un état normal sont bien classifiés.

- **Le recall** est la capacité d'un classificateur à trouver toutes les instances positives. Pour chaque classe, il est défini comme le rapport entre les vrais positifs et la somme des vrais positifs et des faux négatifs. La valeur de recall concernant notre modèle est de 69% pour la classe d'existence d'anxiété et de 75% pour la classe d'état normal, donc le modèle a une bonne capacité de trouver les instances positives de chaque classe.

- Le score F1 est une **moyenne harmonique de la précision et du recall**. Il équivaut au double du produit de ces deux paramètres sur leur somme. Sa valeur est maximale lorsque le rappel et la précision sont équivalents.

➢ **Test chi-square d'indépendance :**

Le test du khi-deux d'indépendance vérifie si deux variables sont susceptibles d'être liées ou pas. Nous avons un dénombrement pour deux variables catégorielles ou nominales. Nous avons également l'idée que les deux variables ne sont pas liées. Le test nous donne le moyen de décider si notre idée est plausible ou pas.

- **Les hypotheses du test:**

  H_0 : Il n'y a pas de relation entre les deux variables.

  H_1 : Il y a une relation de dépendance entre les deux variables.

- **Implémentation sur Python** :

  Afin de tester l'indépendance entre chaque attribut de la variable prédictive et la variable prédite nous avons utilisé la fonction **'stats.chi2_contingency'** offerte par la bibliothèque SciPy de Python, qui prend comme argument les valeurs observées.

anxiete sexe	existence d'anxiété	état normal	All
Femme	268	219	487
Homme	50	135	185
All	318	354	672

Fig.28 : Valeurs observées sexe-anxiété

Le test de chi-2 sur l'indépendance entre le sexe et l'état d'anxiété nous donne une p-value = $1.47.10^{-10}$ avec un degré de liberté égal à 1, nous pouvons déduire qu'il existe une relation de dépendance entre le sexe et l'existence d'anxiété.

anxiete etablissement	existence d'anxiété	état normal	All
CLASSES PREPARATOIRES AUX GRANDES ECOLES	15	10	25
ECOLE D'INGENIEUR	113	171	284
ECOLE DE COMMERCE ET DE GESTION	35	23	58
ECOLE SUPERIEURE DE TECHNOLOGIE	23	21	44
ETABLISSEMENT DE PROFESSIONS INFIRMIERES	1	7	8
ETABLISSEMENT DES METIERS D'EDUCATION	3	2	5
FACULTE DE MEDECINE	15	28	43
FACULTE DES LETTRES ET DES SCIENCES HUMAINES	13	12	25
FACULTE DES SCIENCES	38	35	73
FACULTE DES SCIENCES ECONOMIQUES JURIDIQUES ET SOCIALES	43	28	71
FACULTE DES SCIENCES ET TECHNIQUES	7	7	14
INSTITUT TECHNICIENS SPECIALISES	12	10	22
All	318	354	672

Fig.29 : Valeurs observées établissement-anxiété

Le test de chi-2 sur l'indépendance entre l'établissement et l'état d'anxiété nous donne une p-value = 0.007 avec un degré de liberté égal à 11, nous pouvons déduire qu'il existe relation de dépendance entre l'établissement et l'existence d'anxiété.

anxiete annee	existence d'anxiété	état normal	All
1ère années	95	62	157
2ème année	58	73	131
3ème année	83	84	167
4ème année	48	80	128
5ème année	24	41	65
6ème année où 7ème année (pour les étudiants en médecine)	1	4	5
Doctorants	9	10	19
All	318	354	672

Fig.30 : Valeurs observées année-anxiété

Le test de chi-2 sur l'indépendance entre l'année d'étude et l'état d'anxiété nous donne une p-value = 0.008 avec un degré de liberté égal à 1, nous pouvons déduire qu'il existe relation de dépendance entre l'année et l'existence d'anxiété.

anxiete age	existence d'anxiété	état normal	All
18-20	132	119	251
20-22	133	163	296
22-24	43	51	94
Supérieur à 24	10	21	31
All	318	354	672

Fig.31 : Valeurs observées âge-anxiété

Le test de chi-2 sur l'indépendance entre l'âge et l'état d'anxiété nous donne une p-value = 0.08 avec un degré de liberté égal à 1, nous pouvons déduire qu'il existe relation de dépendance entre l'âge d'étude et l'existence d'anxiété avec un seuil $\alpha = 0.1$.

anxiete residence	existence d'anxiété	état normal	All
Avec famille	153	176	329
Seul(e)	165	178	343
All	318	354	672

Fig.32 : Valeurs observées résidence-anxiété

Le test de chi-2 sur l'indépendance entre l'âge et l'état d'anxiété nous donne une p-value = 0.73 avec un degré de liberté égal à 1, nous pouvons déduire qu'il n'existe pas de relation de dépendance entre la résidence et l'existence d'anxiété.

5.4. Traitement de texte :

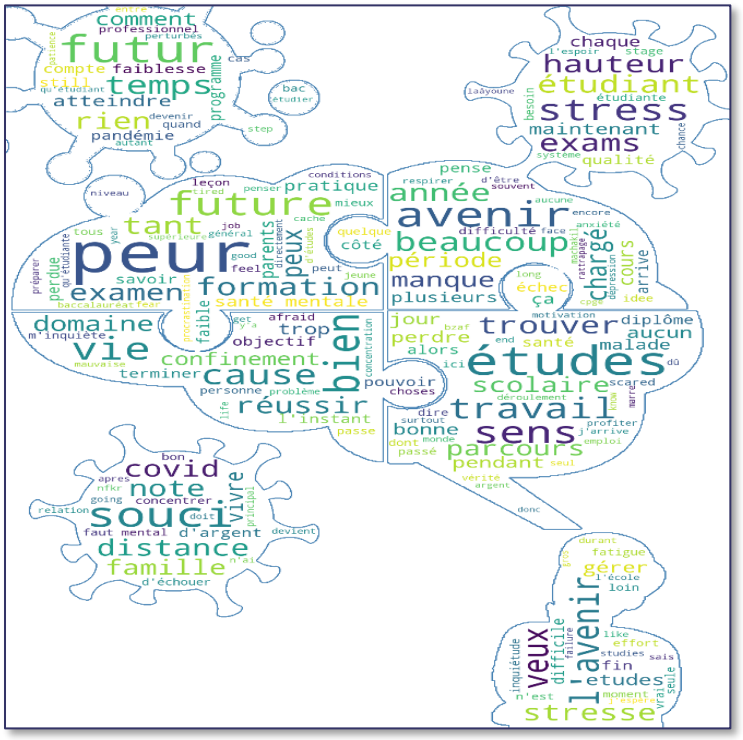

Fig.33 : Word Cloud

A partir de la figure 32, nous constatons que les mots les plus répétés par les étudiants en exprimant leurs soucis et inquiétudes ont une relation avec le futur, l'avenir, les études, la peur et le stress.

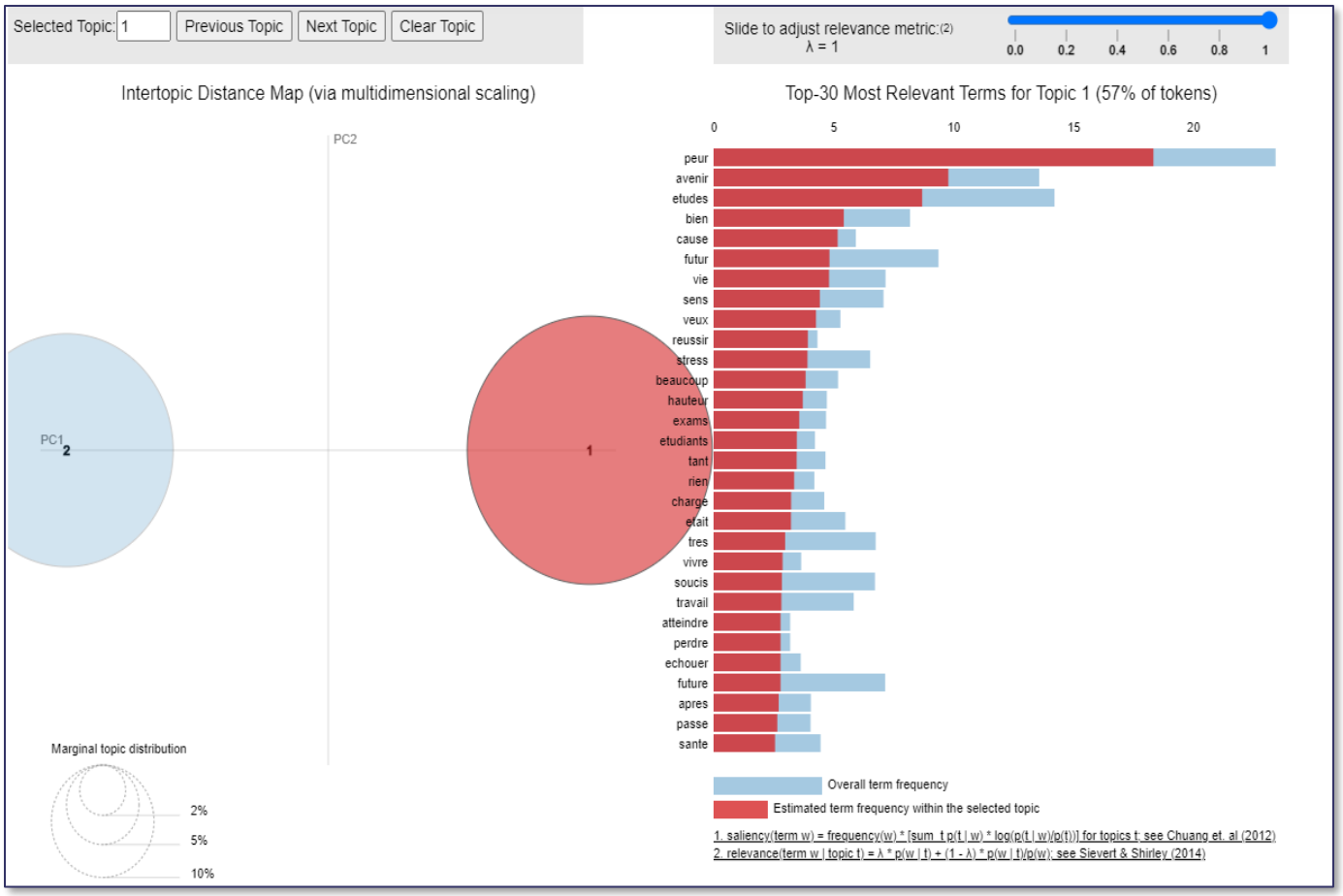

Fig.34 : Distribution des mots du premier sujet

Nous constatons que les mots les plus pertinents du premier sujet donné par le modèle de LDA ont une relation avec la peur, l'avenir, les études, et le stress …

Donc, nous pouvons conclure à partir du premier sujet que les étudiants sont trop inquiets sur leur avenir et ont peur de ne pas réussir dans leurs vies.

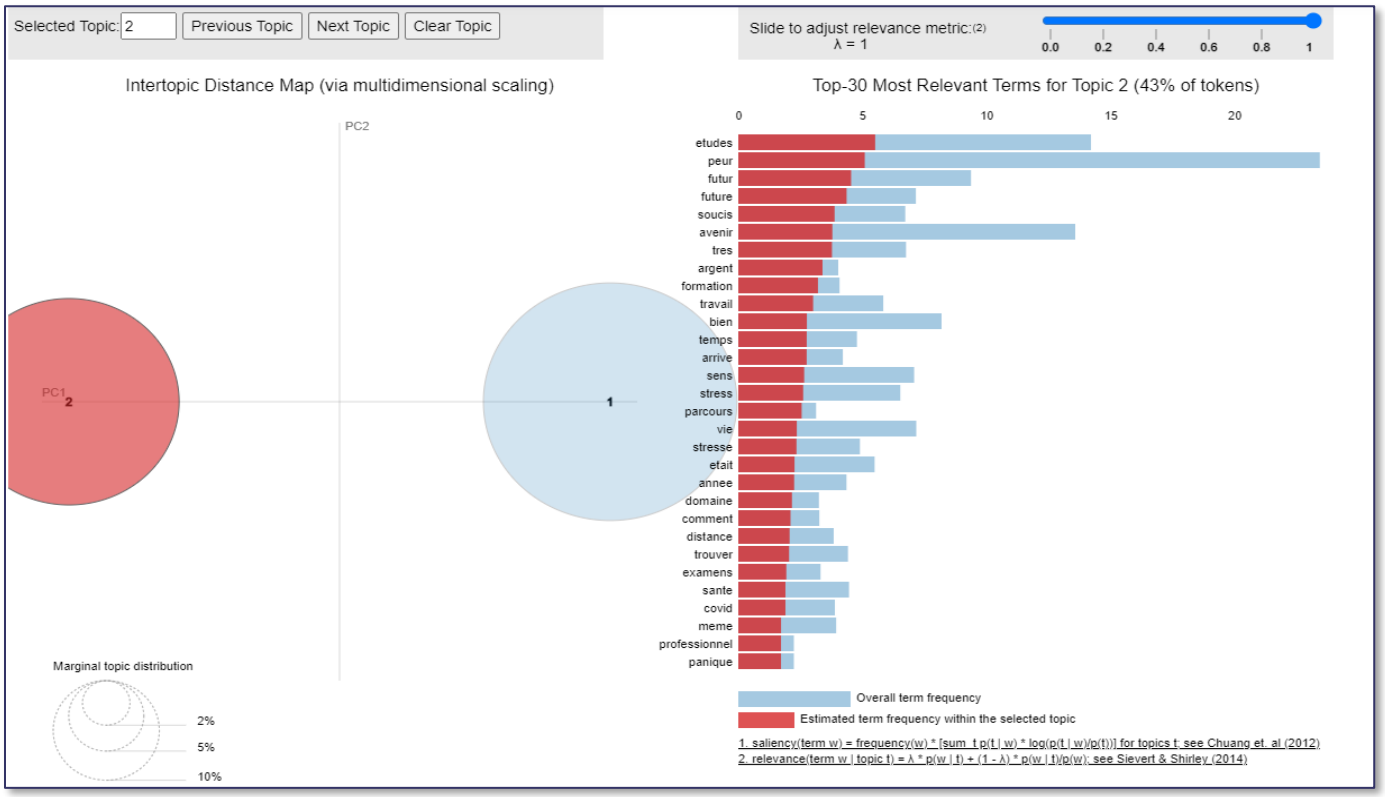

Fig.35 : Distribution des mots du deuxième sujet

Nous constatons que les mots les plus pertinents du deuxième sujet donné par le modèle de LDA ont une relation avec les études, la peur, le futur, l'argent et la formation …

D'après ce deuxième sujet, nous pouvons déduire que les étudiant ont des soucis avec leurs études et examens, et sont inquiets sur la qualité de leurs formations.

6. Conclusion:

Au terme de cette étude, nous avons pu modéliser les effets psychologiques de la période du Covid-19 sur les étudiants de l'enseignement supérieur au Maroc, et cela à travers quelques techniques de Machine Learning, à savoir les visualisations qui nous ont montré que les femmes ont été plus infectées par l'anxiété, cela peut être justifié, en plus des effets du Covid-19, par le fait qu'elles ont plus tendance à s'inquiéter, au niveau des établissements, les graphes nous ont montré que les étudiants des quatre facultés (sciences, économie, lettres, et de médecine) en plus des écoles de commerce et gestion et les établissements des techniciens spécialisés ont été les plus infectés par l'anxiété, cela peut être justifié par la difficulté d'adaptation au mode d'enseignement en ligne qui n'ont en avait aucune expérience précédente.
D'autre part, le modèle de classification de **XGBoost** qui nous a permis de prédire avec succès l'existence d'anxiété en se basant sur les facteurs démographiques des étudiants, ce qui peut être utile pour détecter les étudiants anxieux et leur donner plus d'attention.
Enfin, le traitement de texte qui nous a donné une idée sur les soucis et les inquiétudes des étudiants.

Cette étude démontre le potentiel de Machine Learning pour l'indentification des facteurs prédictifs de l'existence d'anxiété au niveau des étudiants de l'enseignement supérieur au Maroc, ainsi que la détermination de leurs soucis et inquiétudes. Cela veut dire que nous pouvons tirer profit des techniques de la science des données pour résoudre des problèmes du monde réel.

Annexe

	rarement	parfois	souvent	Très souvent
1 Je me sens plus nerveux et anxieux que d'habitude.				
2 J'ai peur sans aucune raison.				
3 Je m'énerve facilement ou je panique.				
4 J'ai l'impression de m'effondrer				
5 Je sens que tout va bien et qu'il ne se passera rien de mal.				
6 Mes bras et mes jambes tremblent				
7 Je suis gêné par un mal de tête et de dos.				
8 Je me sens faible et je me fatigue facilement.				
9 Je me sens calme et je peux rester assis sans bouger.				
10 Je peux sentir mon cœur battre plus vite.				
11 Je suis gêné par des étourdissements.				
12 J'ai des évanouissements ou j'en ai envie.				
13 Je peux inspirer et expirer facilement.				
14 J'ai des sensations d'engourdissement et de picotements dans les doigts et les orteils.				
15 Je suis gêné par des maux d'estomac ou une indigestion.				
16 Je dois souvent vider ma vessie. (aller à la toilette pour uriner)				
17 Mes mains sont généralement sèches et chaudes.				
18 Mon visage devient chaud et rougit.				
19 Je m'endors facilement et je passe une bonne nuit.				
20 Je fais des cauchemars.				

Code source des visualisations

```
#boxplot age
ax=sns.boxplot(y="score",x="age",data=data )
sns.set(rc = {'figure.figsize':(10,10)})
sns.set(font_scale=1)
lines = ax.get_lines()
categories = ax.get_xticks()
for cat in categories:
        y = round(lines[4+cat*6].get_ydata()[0],1)
        ax.text(
            cat,
            y,
            f'{y}',
            ha='center',
            va='center',
            fontweight='semibold',
            size=12,
            color='white',
            bbox=dict(facecolor='#828282', edgecolor='#828282')
    )
plt.show()
```

```
#boxplot sexe
ax=sns.boxplot(y='score',x='sexe', data=data)
lines = ax.get_lines()
categories = ax.get_xticks()
for cat in categories:
        y = round(lines[4+cat*6].get_ydata()[0],1)
        ax.text(
            cat,
            y,
            f'{y}',
            ha='center',
            va='center',
            fontweight='semibold',
            size=15,
            color='white',
            bbox=dict(facecolor='#828282', edgecolor='#828282')
    )
```

```
#boxplot residence
ax=sns.boxplot(y='score',x='residence', data=data, palette="Set3")
#plt.savefig('boxplotResid.png',facecolor = 'white',dpi = 1000)
lines = ax.get_lines()
categories = ax.get_xticks()
for cat in categories:
        y = round(lines[4+cat*6].get_ydata()[0],1)
        ax.text(
            cat,
            y,
            f'{y}',
            ha='center',
            va='center',
            fontweight='semibold',
            size=15,
            color='white',
            bbox=dict(facecolor='#828282', edgecolor='#828282')
    )
plt.show()
```

```
#boxplot etablissement
ax=sns.boxplot(y='score',x='etablissement', data=data)

sns.set(rc = {'figure.figsize':(10,20)})
sns.set(font_scale=1)
plt.xticks([r for r in range(len(data.etablissement.value_counts()))],
        ["ECOLE D'INGENIEUR", "FSEJS", "FACULTE SCIENCES","COMMERCE ET DE GESTION", "EST",
         "CPGE","MEDECINE","FLSH","TECHNICIENS SPECIALISES","FST",
        "METIERS D'EDUCATION","PROFESSIONS INFIRMIERES"],
          rotation=90)

lines = ax.get_lines()
categories = ax.get_xticks()
for cat in categories:
         y = round(lines[4+cat*6].get_ydata()[0],1)
         ax.text(
             cat,
             y,
             f'{y}',
             ha='center',
             va='center',
              fontweight='semibold',
             size=13,
             color='white',
             bbox=dict(facecolor='#828282', edgecolor='#828282')
      )
plt.show()
plt.savefig('boxplotEtab.png',facecolor = 'white',dpi = 1000)

plt.show()
plt.close()
```

```
#boxplot annee
ax=sns.boxplot(y='score',x='annee', data=data)
sns.set(rc = {'figure.figsize':(10,10)})
sns.set(font_scale=1)
plt.xticks([r for r in range(len(data.annee.value_counts()))],
        ["1ère année","3ème année","2ème année","4ème année","5ème année","Doctorants","+6ème médécine"],
          rotation=30)
lines = ax.get_lines()
categories = ax.get_xticks()
for cat in categories:
         y = round(lines[4+cat*6].get_ydata()[0],1)
         ax.text(
             cat,
             y,
             f'{y}',
             ha='center',
             va='center',
            fontweight='semibold',
             size=15,
             color='white',
             bbox=dict(facecolor='#828282', edgecolor='#828282')
      )
plt.show()
#plt.savefig('boxplotAnnee.png',facecolor = 'white',dpi = 1000)

plt.show()
plt.close()
```

```
#visualisation d'anxiete
sns.catplot('anxiete', data=data, kind='count')
plt.savefig('EtatAnx.png',facecolor = 'white',dpi = 1000)
```

```
#sexe pie chart data (nombre des cas d'anxiete)
lis = data.groupby(['sexe','anxiete']).size().sort_values(ascending=False)
lis
```

```
#create the plot
lfm = lis.plot(kind='bar',
           figsize=(15,18),

           width=0.5,
           color=["#e6d800", "#9b19f5", "#9b19f5","#e6d800"],
          )

lfm.set_title("Etat d'anxiete en fonction de sexe", fontsize=16)
lfm.spines['left'].set_visible(False)
lfm.spines['top'].set_visible(False)
lfm.spines['right'].set_visible(False)
lfm.axes.get_yaxis().set_visible(False)
lfm.tick_params(labelsize=10)
#lfm.legend(,fontsize=14)
for p in lfm.patches:
    lfm.annotate(np.round(p.get_height(),decimals=2),
                 (p.get_x()+p.get_width()/2., p.get_height()),
                 ha='center',
                 va='center',
                 xytext=(0, 8),
                 textcoords='offset points',
                 fontsize = 14
                )
plt.xticks(
         rotation=30)
plt.savefig('EtatSexe2.png',facecolor = 'white',dpi = 1000)

plt.show()
plt.close()
```

```
#etablissement pie chart data (nombre des cas d'anxiete)
pie1 = data.mask(data['anxiete'].ne("existence d'anxiété")).groupby(['etablissement','anxiete']).size().sort_values(ascending=Fa
pie1_norm = data.mask(data['anxiete'].ne("état normal")).groupby(['etablissement','anxiete']).size().sort_values(ascending=False
pie1
```

```
#etablissement plot pie chart

colors_list =["#e60049", "#0bb4ff", "#e6d800", "#9b19f5", "#dc0ab4",  "#ffa300","#50e991", "#b3d4ff",
              "#00bfa0","#4421af","#7c1158","#ede15b"]
pie1.plot(kind='pie',
         figsize=(20, 8),
         autopct='%1.1f%%',
         startangle=200,
          explode = (0.2, 0, 0, 0, 0, 0,0,0,0,0,0,0),

         shadow=False,
         colors=colors_list,  # add custom colors
         labels=None)
plt.title("Le taux  d'anxiété au niveau des établissements")
plt.axis('equal')
plt.legend(labels=["ECOLE D'INGENIEUR", "FSEJS", "FACULTE DES SCIENCES","ECOLE DE COMMERCE ET DE GESTION",
                   "ECOLE SUPERIEURE DE TECHNOLOGIE",
         "CPGE","FACULTE DE MEDECINE","FLSH","INSTITUT TECHNICIENS SPECIALISES","FST",
        "ETABLISSEMENT DES METIERS D'EDUCATION","ETABLISSEMENT DE PROFESSIONS INFIRMIERES"] ,bbox_to_anchor=(0, 1)
          , loc='upper left')
plt.savefig('pieEtab.png',facecolor = 'white',dpi = 1000)

plt.show()
plt.close()
```

```
barWidth = 0.25
fig = plt.subplots(figsize =(12, 18))
# Set position of bar on X axis
br1 = np.arange(len(data.etablissement.value_counts()))
br2 = [x + barWidth for x in br1]
# Make the plot
plt.bar(br1, pie1, color ='r', width = barWidth,
        edgecolor ='grey', label ="existence d'anxiété")
plt.bar(br2, pie1_norm, color ='g', width = barWidth,
        edgecolor ='grey', label ="état normal")
# Adding Xticks
plt.xlabel('Etablissements', fontweight ='bold', fontsize = 15)
plt.ylabel('Effectif', fontweight ='bold', fontsize = 15)
plt.xticks([r + barWidth for r in range(len(data.etablissement.value_counts()))],
        ["ECOLE D'INGENIEUR", "FSEJS", "FACULTE DES SCIENCES","COMMERCE ET DE GESTION", "EST",
         "CPGE","MEDECINE","FLSH","TECHNICIENS SPECIALISES","FST",
        "METIERS D'EDUCATION","PROFESSIONS INFIRMIERES"],
          rotation=90)

plt.legend()
plt.savefig('multi bar etab.png',facecolor = 'white',dpi = 1000)
plt.show()
plt.close()
```

```
#annee pie chart data
pie2 = data.mask(data['anxiete'].ne("existence d'anxiété")).groupby(['annee','anxiete']).size().sort_values(ascending=False)
pie2_norm = data.mask(data['anxiete'].ne("état normal")).groupby(['annee','anxiete']).size().sort_values(ascending=False)
pie2
```

```
#annee plot pie chart
pie2.plot(kind='pie',
      figsize=(8,8),
      autopct='%1.1f%%',
      startangle=200,
      colors=colors_list,
      labels=None)
plt.title("Le taux d'anxiété au niveau des années d'étude")

plt.axis('equal')
plt.legend(labels=pie2.index ,bbox_to_anchor=(1, 1), loc='upper left')
plt.savefig('annne.png',facecolor = 'white',dpi = 1000)

plt.show()
plt.close()
```

```
barWidth = 0.25
fig = plt.subplots(figsize =(12, 10))
# Set position of bar on X axis
br1 = np.arange(len(data.annee.value_counts()))
br2 = [x + barWidth for x in br1]
# Make the plot
plt.bar(br1, pie2, color ='r', width = barWidth,
        edgecolor ='grey', label ="existence d'anxiété")
plt.bar(br2, pie2_norm, color ='g', width = barWidth,
        edgecolor ='grey', label ="état normal")
# Adding Xticks
plt.xlabel("Année d'étude", fontweight ='bold', fontsize = 15)
plt.ylabel('Effectif', fontweight ='bold', fontsize = 15)
plt.xticks([r + barWidth for r in range(len(data.annee.value_counts()))],
        ["1ère année","3ème année","2ème année","4ème année","5ème année","Doctorants","+6ème médécine"],
          rotation=30)

plt.legend()

plt.savefig('multi bar annee.png',facecolor = 'white',dpi = 1000)
plt.show()
plt.close()
```

```
barWidth = 0.25
fig = plt.subplots(figsize =(12, 8))
# Set position of bar on X axis
br1 = np.arange(len(data.age.value_counts()))
br2 = [x + barWidth for x in br1]
# Make the plot
plt.bar(br1, pie3, color ='r', width = barWidth,
        edgecolor ='grey', label ="existence d'anxiété")
plt.bar(br2, pie3_norm, color ='g', width = barWidth,
        edgecolor ='grey', label ="état normal")
# Adding Xticks
plt.xlabel('Age', fontweight ='bold', fontsize = 15)
plt.ylabel('Effectif', fontweight ='bold', fontsize = 15)
plt.xticks([r + barWidth for r in range(len(data.age.value_counts()))],
        ["20-22","18-20","22-24","+24"],
          rotation=90)

plt.legend()
#plt.savefig('multi bar age.png',facecolor = 'white',dpi = 1000)
plt.show()
plt.close()
```

```
#sexe pie chart data
pie4 = data.mask(data['anxiete'].ne("existence d'anxiété")).groupby(['sexe','anxiete']).size().sort_values(ascending=False)
pie4
```

```
#sexe plot pie chart

pie4.plot(kind='pie',
      figsize=(12, 10),
      autopct='%1.1f%%',
      startangle=200,
      colors=colors_list,
       explode = (0.2, 0),

      shadow=True,
      labels=pie4.index)
plt.title("Existence d'anxiété en fonction de sexe")

plt.axis('equal')
plt.legend(labels=pie4.index ,bbox_to_anchor=(1, 1), loc='upper left')
plt.savefig('mul.png',facecolor = 'white',dpi = 500)

plt.show()
plt.close()
```

```
#residence pie chart data
pie5 = data.mask(data['anxiete'].ne("existence d'anxiété")).groupby(['residence','anxiete']).size().sort_values(ascending=False)
pie5
```

```
#sexe plot pie chart

pie5.plot(kind='pie',
      figsize=(8, 8),
      autopct='%1.1f%%',
      startangle=200,
      colors=colors_list,
      shadow=True,
      labels=pie5.index)
plt.title("Existence d'anxiété en fonction de residence")

plt.axis('equal')
plt.legend(labels=pie5.index ,bbox_to_anchor=(1, 1), loc='upper left')
plt.savefig('resid.png',facecolor = 'white',dpi = 100)

plt.show()
plt.close()
```

```
#confusion matrix
plot_confusion_matrix(model, X_test, y_test, values_format = 'd')
```

```
from pycorrcat.pycorrcat import plot_corr, corr_matrix
features = pd.DataFrame(data=X)
features.columns = ['sexe','etablissement','annee','residence','age']
correlation_matrix = corr_matrix(features,['sexe','etablissement','annee','residence','age'])
plot_corr(features,['sexe','etablissement','annee','residence','age'])
```

Netographie :

- https://www.datacamp.com/tutorial/xgboost-in-python
- https://xgboost.readthedocs.io/en/stable/
- https://blent.ai/xgboost-tout-comprendre/
- https://www.mdpi.com/2227-9032/10/1/149/html
- https://www.jedha.co/formation-ia/matrice-confusion
- https://www.jmp.com/fr_ca/statistics-knowledge-portal/chi-square-test/chi-square-test-of-independence.html
- https://fr.acervolima.com/generer-un-nuage-de-mots-en-python/
- https://towardsdatascience.com/end-to-end-topic-modeling-in-python-latent-dirichlet-allocation-lda-35ce4ed6b3e0
- https://www.journaldemontreal.com/2016/02/23/les-femmes-sont-plus-anxieuses

Bibliographie :

- "XGBoost With Python", Jason Brownlee, 2018
- "Anxiety Level of University Students During COVID-19 in Saudi Arabia" Khoshaim HB, Al-Sukayt A, Chinna K, Nurunnabi M, Sundarasen S, Kamaludin K, Baloch GM and Hossain SFA, 2020
- "Chinese College Students Have Higher Anxiety in New Semester of Online Learning During COVID-19: A Machine Learning Approach", Wang C, Zhao H and Zhang H, 2020

Résumé :

À cause de la propagation du Covid-19, et avec le renfermement durant le confinement, une grande partie de la population marocaine s'est infectée par une dégradation au niveau mental. Pour cela, et dans le cadre de notre projet de fin d'année qui a pour objectif de modéliser les effets psychologiques du Covid-19 lors du premier confinement sur les étudiants de l'enseignement supérieur au Maroc en fonction de divers indicateurs, nous avons utilisé quelques techniques de la science des données, à savoir des visualisations qui nous ont montré la distribution de l'anxiété au niveau des étudiants en fonction des facteurs démographiques. D'autre part, un modèle de classification de **XGBoost** qui nous a permis de prédire avec succès l'existence d'anxiété en se basant sur les facteurs démographiques des étudiants, ce qui peut être utile pour détecter les étudiants anxieux et leur donner plus d'attention. Enfin, un traitement de texte qui nous a donné une idée sur les soucis et les inquiétudes des étudiants.

A travers les résultats de cette étude, nous constatons le potentiel de Machine Learning pour l'indentification des facteurs prédictifs de l'existence d'anxiété au niveau des étudiants de l'enseignement supérieur au Maroc, ainsi que la modélisation de leurs soucis et inquiétudes. Cela veut dire que nous pouvons tirer profit des techniques de la science des données pour résoudre des problèmes du monde réel.