

# RENDU FINAL DU PROJET DE FIN D'ANNEE ING1 GM

**Groupe : ING1 GMI 4**

**Le sujet du projet : Orientation**

**Fait par :**

**Equipe 7**

**Encadré Par :**

**Pr. ZOGHLAMI Naïm**

**ATIG Tasnim**

**BEN MANSOUR Youssef**

**BOWE Romain**

**POCHIC Arthur**

***Cheffe de l'équipe : ATIG Tasnim***

***Interlocuteur de l'équipe : BOWE Romain***

# Table des matières

I-	Etat de l'art: Synthèse de recherche bibliographique	3
	1- Introduction et problématique	3
	2- Synthèse bibliographique	4
	a- Contextualisation	4
	b- Orientation vers le lycée et l'université en France (Affelnet et Parcoursup) et problème de mariages stables	4
	c- Autres algorithmes d'affectation utilisés	5
	d- Algorithmes de classification utilisés (Data Mining et Machine Learning)	6
	e- Evaluation des différents algorithmes	7
	3- Conclusion	8
II-	Cahier de charge de notre solution	9
	1- Présentation du projet	9
	a- Description du projet	9
	b- Contexte et Mise en situation	9
	c- Analyse de l'existant	9
	d- Les objectifs qualitatifs	9
	e- Les objectifs quantitatifs	10
	2- Différents aspects du projet	10
	a- Aspects fonctionnels	10
	b- Aspects techniques	10
	3- Périmètre du sujet	10
	4- Ressources	11
	5- Délais	11
III-	Notre solution conceptuelle : Plan et première vue	12
	1- Plan détaillé	12
	2- Organigramme explicatif	13
IV-	Notre solution conceptuelle : explication exhaustive	14
	1- Plan détaillé	14
	2- Choix du langage de programmation	14
	3- Création des classes	14
	4- Implémentation des algorithmes	16
	5- Préparation des jeux de données et test	17
	6- Statistiques	20
	7- Interfaces utilisateur	21
	8- Conclusion	21
V-	Répartition des tâches	22
VI-	Regard critique	23
VII-	Références bibliographiques	24

# Etat de l'art : Synthèse de recherche bibliographique

## 1) Introduction et problématique :

Dans le contexte académique contemporain, l'orientation des étudiants vers des domaines spécifiques revêt une importance capitale voire cruciale, étant fondamentale pour leur préparation à la vie professionnelle. Notamment, les filières en sciences techniques et informatiques sont aujourd'hui hautement valorisées et largement prisées en raison de leur pertinence envers les exigences du marché actuel se basant sur les avancées technologiques et scientifiques.

En raison de cette ampleur, la question de l'orientation demeure un défi significatif pour les étudiants, étant donné les limitations que peuvent imposer leurs résultats scolaires antérieurs quant à leurs possibilités de choix de filière. Ainsi, certains étudiants pourraient se trouver mal adaptés à leur programme d'études en raison de leurs aptitudes cognitives limitées, ce qui peut induire des difficultés psychologiques. Dans un contexte contemporain, les choix d'orientation inappropriés sont fréquemment associés à des échecs académiques, des abandons de cursus, une incapacité à s'intégrer professionnellement après l'obtention du diplôme, et une inadéquation avec les exigences du monde du travail. Ainsi, les étudiants sont souvent demandés de faire preuve de discernement et de ténacité dans leurs choix d'orientation, ceux-ci étant les fondations sur lesquelles reposera leur parcours futur. Conformément à la vision de l'interaction entre les systèmes éducatifs et les paramètres socio-culturels des étudiants, ce processus englobe la formulation des aspirations académiques, incluant le choix du niveau d'études et de la filière, ainsi que la sélection d'une discipline, d'un programme ou d'une carrière.

De ce fait se pose le problème de l'orientation des étudiants vers leurs filières au sein d'une école, qui fait partie du grand processus de l'orientation scolaire et universitaire. En effet, cette étape revêt une importance primordiale dans leur parcours académique et professionnel vu qu'elle constitue souvent le point de départ pour leur immersion dans le monde universitaire et la construction de leur future carrière. En d'autres termes, choisir une filière adaptée à leurs intérêts, leurs compétences et leurs aspirations est essentiel pour favoriser leur engagement, leur motivation et leur réussite académique. [3]

Pareillement au problème de l'orientation scolaire ou celle universitaire et postbac, les étudiants ainsi que l'administration d'une école d'ingénieurs comme CY TECH peuvent envisager quelques problèmes contraignants à cette opération qui se développent de plus en plus avec la croissance annuelle marquée du nombre des étudiants essentiellement d'une promo à une autre et la diversité des spécialités et des filières selon les besoins du marché professionnel. D'autres défis sont liés à l'orientation des étudiants vers leurs filières au sein d'une école : ceux-ci comprennent la diversité des intérêts et des compétences des étudiants, le manque d'information sur les filières disponibles, la pression sociale et familiale, l'accès limité aux ressources d'orientation, le besoin de conseil personnalisé, l'adaptation des programmes aux besoins du marché du travail en constante évolution, la stigmatisation de certaines filières, ainsi que l'incertitude quant aux débouchés professionnels. Ces contraintes rendent essentiel le développement de stratégies d'orientation plus efficaces afin d'assister les étudiants dans leurs choix de filière, tout en favorisant leur succès académique et professionnel. C'est ainsi que notre projet et notre solution prend tout son sens. En effet, nous cherchons au biais de ce projet de fin d'année et dont le sujet est « l'orientation » de concevoir et développer une application pour l'orientation des élèves ING2 CYTech admis en ING3 vers plusieurs options offertes par l'école telles que : l'IA, le Data science, la cybersécurité, ... tout en

prenant en compte le nombre de places pédagogiques déterminé pour chaque filière, les vœux et les préférences de chaque étudiant ainsi qu'au taux d'aspirations pour chaque filière et ceci afin de d'aboutir finalement à une orientation équitable des élèves, basée sur leurs résultats et leurs vœux. En d'autres mots, nous sommes amenés à trouver une bonne solution à la problématique suivante :

*« Comment concevoir et développer une application d'orientation pour les élèves de deuxième année à Cy-Tech, afin de répartir équitablement les étudiants en troisième année parmi les différentes options proposées, tout en tenant compte de la popularité inégale des options, des places pédagogiques limitées et en se basant sur les résultats académiques et les vœux des étudiants ? »*

## **2) Synthèse bibliographique :**

### **a- Contextualisation :**

N'ayant pas trouvé des documents ni des ressources illustrant les méthodes d'une orientation de filière au sein d'une école, cette synthèse établie de notre recherche bibliographique est issue des thèses et des recherches archivées qui le sujet principal et l'orientation au sein du système éducatif français au sens général, c'est à dire d'une phase scolaire à une autre (de l'école primaire au collège, du collège au lycée, du lycée à l'université). En faisant l'analogie entre notre problématique et le contenu de notre recherche, nous vous présentons par la présente partie une synthèse globale et concise concernant les différentes techniques et méthodes utilisées récemment lors des étapes d'orientation.

Depuis les années 2000, le système d'orientation en établissements scolaires et universitaires français témoignait d'une automatisation des processus. En effet, les professionnels de l'orientation ont commencé à employer diverses méthodes et techniques et d'outils spécifiques comme les tests psychométriques, les logiciels d'aide à l'orientation, les techniques d'entretien et les méthodes d'animation de groupe qui sont tous détaillés par plusieurs articles publiés. D'ailleurs, au-delà de la dimension technique et numérique du problème de l'orientation, les élèves et les étudiants recevaient toujours du soutien professionnel au cours des étapes du parcours d'orientation postbac, depuis l'exploration des possibles jusqu'à la préparation de l'entrée dans le supérieur. Ce soutien inclut l'utilisation de questionnaires d'auto-évaluation, des MOOC d'aide à l'orientation, et des plateformes numériques comme Inspire, qui permettent aux étudiants d'interagir avec des mentors et d'autres étudiants pour une meilleure projection dans leurs choix éducatifs et professionnels. [5] [6]

### **b- Orientation vers le lycée et l'université en France (Affelnet et Parcoursup) et problème de mariages stables :**

Quant aux techniques et logiciels utilisés, des procédures centralisées et automatisées, telles que Affelnet pour le lycée et Parcoursup pour l'enseignement supérieur, ont été mises en place pour gérer les transitions entre le collège et le lycée, ainsi que sur l'entrée dans l'enseignement supérieur. Ces algorithmes, inspirés par la théorie de l'appariement, visent à optimiser la distribution des élèves dans les établissements scolaires tout en respectant divers critères comme les résultats scolaires et les critères sociaux. Cependant, malgré leur utilité, certains systèmes ont souvent été critiqués pour leur manque de transparence tels que le programme Affelnet qui a été réformé pour répondre aux critiques en introduisant une sectorisation progressive et des bonus sociaux pour favoriser la mixité sociale. L'analyse de ces mécanismes révèle les défis et les ajustements nécessaires pour améliorer leur fonctionnement et leur acceptabilité sociale. [2]

Parcoursup, en revanche, est une plateforme utilisée pour la préinscription dans les formations de l'enseignement supérieur en France. Elle permet aux lycéens, apprentis, étudiants en réorientation et autres candidats éligibles de formuler leurs vœux et de recevoir des propositions d'admission, conformément à la loi du 8 mars 2018 sur l'orientation et la réussite des étudiants. Un comité éthique

et scientifique veille au respect des principes juridiques et éthiques de la procédure, et le code source de l'algorithme de Parcoursup est public pour garantir la transparence. [1]

La plateforme de Parcoursup utilise une variante du problème des mariages stables pour l'affectation des étudiants dans les établissements d'enseignement supérieur. En mathématiques, informatique et économie, le problème des mariages stables consiste à trouver, étant donné  $n$  hommes et  $n$  femmes, et leurs listes de préférences, une façon stable de les mettre en couple.

Par ailleurs, le fonctionnement de cette plateforme suit la démarche suivante :

- **Liste de Vœux :** Les étudiants soumettent des listes de vœux classées par ordre de préférence pour les formations qu'ils souhaitent intégrer.
- **Classements des Établissements :** Les établissements classent les candidats selon divers critères (notes, projets personnels, etc.).
- **Algorithme d'Affectation :** L'algorithme compare les listes de préférences des étudiants et les classements des établissements pour réaliser les affectations de manière stable, minimisant les cas où un étudiant et un établissement préfèrent une autre affectation à celle proposée par l'algorithme. Parcoursup applique ainsi les principes des mariages stables pour garantir une affectation où ni les étudiants ni les établissements ne trouvent de meilleures alternatives mutuelles non satisfaites par l'algorithme, ce qui se traduirait par une paire bloquante dans le contexte de Gale-Shapley.
  - **Algorithme de Gale-Shapley :** c'est une quinzaine de lignes de code qui affecte les étudiants en fonction de deux classements :
    - celui des établissements par les étudiants
    - celui des étudiants par les établissements

Par conséquent, l'algorithme ne détermine pas ces classements mais uniquement l'affectation finale. Ce n'est pas l'algorithme qui est une boîte noire, mais la façon dont les établissements classent les dossiers. L'algorithme produit une affectation efficace dans la mesure où il est impossible de trouver une paire étudiant-établissement telle que : *L'étudiant aurait préféré cet établissement à celui auquel il a été affecté, et l'établissement aurait préféré cet étudiant à un des étudiants qui lui ont été affectés.*

Ainsi, toute cette approche permet de s'assurer que les affectations sont faites de manière à minimiser les cas d'insatisfaction majeure (paires bloquantes). [7]

Il est à noter par la suite que le problème d'orientation des étudiants vers leurs filières au sein d'une école peut être abordé également sous l'angle de l'affectation ou de la classification, selon la manière dont il est formulé et traité. Les méthodes d'affectation comme l'algorithme de Gale-Shapley (utilisé par exemple dans les mariages stables) ou l'algorithme de l'assignation optimale peuvent être adaptées pour résoudre ce type de problème.

### c- Autres algorithmes d'affectation utilisés :

Lorsque l'objectif est de placer chaque étudiant dans une filière spécifique parmi un ensemble de choix disponibles, le problème peut être considéré comme un problème d'affectation. Dans ce contexte, un algorithme d'affectation peut être utilisé pour attribuer de manière optimale les étudiants aux filières en tenant compte de leurs préférences, de leurs compétences et des capacités disponibles dans chaque filière.

- **Algorithme de l'enchère (Auction Algorithm) :**

Cet algorithme fonctionne en attribuant des valeurs aux préférences des étudiants et en optimisant la répartition des étudiants aux filières en fonction de ces valeurs.

Étapes principales :

- Chaque filière "enchère" pour les étudiants en fonction de critères prédéfinis.
- Les étudiants sont assignés aux filières en fonction des offres et de leurs préférences.

#### ▪ *Algorithme de Flot Maximal (Max Flow Algorithm)*

Ce type d'algorithme est utilisé pour maximiser l'assignation des étudiants aux filières tout en respectant les contraintes de capacité.

#### Étapes principales :

- Représenter le problème comme un réseau de flux avec des étudiants et des filières comme nœuds.
- Utiliser des techniques comme Ford-Fulkerson ou Edmonds-Karp pour trouver le flot maximal dans le réseau, qui représente la répartition optimale.

#### ▪ *Algorithmes basés sur les Scores (Score-Based Algorithms)*

Ces algorithmes utilisent des scores basés sur des critères académiques, des tests d'aptitude, ou d'autres indicateurs pour assigner les étudiants à des filières.

#### Étapes principales :

- Calculer un score pour chaque étudiant en fonction des critères définis.
- Trier les étudiants et les assigner aux filières disponibles en fonction de leurs scores et des capacités des programmes.

#### ▪ *Algorithme de Gale-Shapley modifié pour l'Admission École*

Une version adaptée de l'algorithme de Gale-Shapley peut être utilisée pour l'affectation des étudiants dans les écoles ou universités. Cet algorithme prend en compte non seulement les préférences des étudiants mais aussi celles des filières.

#### ▪ *Algorithmes Génétiques*

Ces algorithmes de type évolutionniste utilisent des techniques inspirées de la sélection naturelle pour trouver une solution optimale à l'affectation des étudiants.

#### Étapes principales :

- Générer une population initiale de solutions possibles.
- Évaluer et sélectionner les meilleures solutions.
- Appliquer des opérations génétiques comme le croisement et la mutation pour créer de nouvelles solutions.
- Répéter le processus jusqu'à atteindre une solution satisfaisante. [4]

### **d- Algorithmes de classification utilisés (Data Mining et Machine Learning) :**

Avec l'essor des données et de l'apprentissage automatique, certains systèmes utilisent des techniques de Machine Learning et de Data Mining pour prédire l'orientation idéale des étudiants en se basant sur des données historiques et des profils d'étudiants.

#### Étapes principales :

- Collecter des données historiques sur les étudiants et les filières.

- Entraîner un modèle de Machine Learning pour prédire la filière idéale pour chaque nouvel étudiant.
- Utiliser le modèle pour effectuer l'assignation en temps réel.

Chaque algorithme a ses propres avantages et inconvénients, et le choix de l'algorithme dépend des critères spécifiques de l'école, des capacités des programmes, des préférences des étudiants, et des objectifs globaux du processus d'orientation.

Plusieurs algorithmes de Machine Learning sont donc utilisés dans le contexte de l'orientation éducative, en particulier l'algorithme des SVM et des réseaux de neurones et des méthodes. L'utilisation de Data Mining pour l'extraction de connaissances à partir de grandes bases de données éducatives est aussi discutée en profondeur et pleinement utilisée. L'objectif est d'identifier des tendances et des corrélations qui ne sont pas immédiatement apparentes, afin d'améliorer les recommandations d'orientation. Les algorithmes décrits incluent les méthodes de classification et de régression, cruciales pour une analyse prédictive efficace.

Voici la liste des algorithmes les plus connus dans ce cadre :

- Arbres de décision : Un modèle qui prédit la filière en fonction d'une série de décisions basées sur les caractéristiques des étudiants.
- Forêts aléatoires : Une collection d'arbres de décision utilisés pour améliorer la précision de la classification.
- Réseaux de neurones : Modèles inspirés du cerveau humain, capables de reconnaître des patterns complexes dans les données pour prédire la filière.
- Machines à vecteurs de support (SVM) : Algorithmes qui trouvent l'hyperplan optimal séparant les différentes catégories d'étudiants en fonction de leurs caractéristiques. [3] [4]

### e- Evaluation des différents algorithmes :

En plus de souligner les algorithmes mentionnés ci-dessus et de les présenter précisément, les documents trouvés lors de notre recherche les évaluent selon des critères tels que la précision (Accuracy), le rappel (Recall), la précision (Precision) et le F-score. Voici les résultats de cette évaluation :

	Accuracy	Recall	Precision	F-score
SVM	70%	0.7	0.64	0.66
Naïve Bayes	65%	1	1	1
Réseau de Neurones	64%	0.41	0.35	0.37
Arbre de Décision	52%	0.52	0.45	0.46

On déduit de ce tableau le classement suivant :

- **Meilleur Algorithme Global** : Le SVM est le plus performant en termes de précision et de F-score, ce qui en fait le choix préféré pour les applications d'orientation éducative basées sur ces critères.
- **Algorithme à Haut Risque de Surapprentissage** : Naïve Bayes montre des performances théoriques excellentes, mais le risque de surapprentissage le rend moins fiable en pratique.
- **Algorithme pour la Reconnaissance de Motifs Complexes** : Les réseaux de neurones sont efficaces mais nécessitent un réglage précis et des données de haute qualité.
- **Algorithme le Plus Interprétable** : Arbre de décision est le plus facile à comprendre et à interpréter, bien que sa performance soit la plus faible selon les critères analysés.

Ces évaluations aident à choisir l'algorithme le plus approprié en fonction des besoins spécifiques et des contraintes de l'application d'orientation éducative.

### **3) Conclusion :**

Les limites actuelles de l'orientation des filières au sein des écoles mettent en évidence la complexité et les défis associés à cette tâche. Ces difficultés soulignent la pertinence et l'intérêt de notre projet. En effet, l'absence de transparence dans les critères de classement des dossiers par les établissements, le manque de ressources et d'information sur les filières disponibles, et la pression sociale et familiale sont autant d'obstacles qui compliquent le processus d'orientation. De plus, l'adéquation entre les compétences des étudiants et les exigences des filières, ainsi que l'évolution constante des besoins du marché du travail, nécessitent des solutions innovantes et personnalisées.

En s'inspirant des algorithmes d'affectation comme celui de Gale-Shapley et d'autres techniques de Machine Learning et de Data Mining, notre solution vise à améliorer l'équité et l'efficacité du processus d'orientation, offrant ainsi aux étudiants de notre école une meilleure correspondance entre leurs aspirations et leurs futurs parcours académiques et professionnels. Ce projet, en répondant à une problématique concrète et actuelle, se révèle non seulement raisonnable mais aussi indispensable pour une orientation de filières plus juste et réussie.



# Cahier de Charge de notre solution

## 1) Présentation du projet :

### a- Description du projet :

Notre projet consiste à la résolution mathématique du problème de l'orientation des étudiants de ING2 vers leurs futures filières de la 3<sup>ème</sup> année en établissant une plateforme dynamique appelée « CY Tech Oriente » se basant sur des méthodes algorithmiques pour rendre ce processus plus automatique et mettre fin à l'iniquité lors de l'orientation des étudiants.



### b- Contexte et Mise en situation :

Les étudiants de CYTech en ING2 doivent choisir leur option ING3, ainsi il leur est demandé de formuler un panier de vœux contenant les options qui les intéressent, classées par ordre de préférence. Un tri sera fait à l'aide de notre solution algorithmique pour privilégier les demandes des étudiants aux meilleurs résultats dans leur choix, graduellement jusqu'à ce que toute la population d'étudiants soit couverte et qu'une option ait été attribuée à tous les étudiants. La somme du nombre de places dans chaque option coïncide exactement avec le nombre d'étudiants faisant leur choix.

### c- Analyse de l'existant :

Selon les informations données par notre professeur encadrant, notre école ne prend en compte que les moyennes triées par ordre décroissant du premier semestre des étudiants de ING2 ainsi que leurs listes de vœux pour les orienter vers leurs filières de ING3. Cette méthode nous semble être à la fois coûteuse et injuste vu qu'elle ne considère pas les différences des compétences techniques acquises entre les étudiants en lien avec la pédagogie de chaque filière. Ainsi, nous comptons donner une solution alternative à cette stratégie qui fait intervenir les notes de certaines matières.

### d- Les objectifs qualitatifs :

L'objectif principal du projet est de se renseigner sur les algorithmes d'orientation actuels, donc de classement, et de choisir une ou plusieurs méthodes de classement à implémenter afin de classer au mieux les étudiants, et donc d'attribuer de façon équitable une option à chaque étudiant.

Classer les étudiants demandant l'option 1 peut s'avérer challengeant, étant donné que certaines options sont ouvertes à plusieurs filières, par exemple fintech qui est ouverte à GMI et GMF. Il faudra donc trouver un moyen de peser les résultats de chaque étudiant proportionnellement à sa filière, tout en sachant que les étudiants en GMI n'ont pas les mêmes matières et attendus que les étudiants en GMF.

Implémenter une méthode juste et sans biais est aussi un des points clés de notre projet, car nous jugeons que l'attribution d'une option basée uniquement sur les résultats d'un seul semestre, et ce, en session principale est loin d'être juste et équitable.

Finalement, nous voudrions également par le présent projet implémenter une plateforme simple et accessible aux étudiants pour leur rendre la tâche de l'orientation plus aisée et claire et permettre à l'administration de mieux la gérer.

#### **e- Objectifs quantitatifs :**

- Répondre à un afflux massif d'étudiants au même temps
- Bien gérer la croissance des effectifs des promotions en vue de nombre de groupes et filières
- Présenter des statistiques et des diagrammes clairs et récents de l'orientation de chaque promotion à l'administration

### **2) Différents aspects du projet :**

#### **a- Aspects fonctionnels :**

Une personne est censée en premier lieu, après avoir ouvert l'application, se connecter. Soit elle est étudiante, soit la personne est une admin.

Pour un profil admin, l'admin doit pouvoir dans un premier temps ouvrir l'accès au panier aux étudiants afin qu'ils puissent inscrire leurs vœux dans le panier de vœux. Les étudiants doivent ensuite pouvoir se connecter et avoir accès à leur panier de vœux pour le remplir avec leurs 3 vœux principaux, rentrer leurs résultats et confirmer leur profil.

Après une certaine période, les admins doivent pouvoir fermer l'accès au panier à vœux, si des étudiants n'ont pas rentrés leurs vœux après cette période, ils doivent communiquer leurs vœux à un admin qui se chargera de les rentrer pour eux dans leur profil manuellement, cette opération doit être consignée comme opération spéciale dans l'application et sa date, son auteur ainsi que la modification faite doivent être accessibles par l'étudiant et les admins.

Une fois tous les vœux rentrés, les admins doivent pouvoir lancer la procédure de classement et d'attribution des options.

Une fois l'algorithme terminé, les admins ouvrent l'accès des étudiants aux résultats, qui vont pouvoir se connecter et regarder quelle option leur a été attribuée.

#### **b- Aspects Techniques :**

- Faire un cahier de charges détaillé
- Faire un diagramme des classes (UML) de l'application en entier
- Apercevoir le résultat avec un prototype Figma sous la forme d'une animation
- Construire et publier un état de l'art des algorithmes de classement et d'orientation détaillé
- Construire un jeu de données cohérent et fourni pour les tests
- Trouver et implémenter les algorithmes d'orientation en python, orienté objet
- Construire une interface homme/machine à l'aide de TKinter sur python

### **3) Périmètre du projet :**

Pas de limite, le projet peut être repris par n'importe qui extérieur à l'équipe de départ ou n'importe qui de l'équipe en vue d'une amélioration du modèle, le projet peut être utilisé par l'école ou n'importe quel organisme en ayant besoin, et ce gratuitement en mentionnant tout de même les noms des contributeurs et en les contactant au préalable.

Version 1.0 implémentée par :

- ATIG Tasnim: [atigtasnim@cy-tech.fr](mailto:atigtasnim@cy-tech.fr)
- BEN MANSOUR Youssef: [benmanso@cy-tech.fr](mailto:benmanso@cy-tech.fr)
- BOWE Romain: [boweromain@cy-tech.fr](mailto:boweromain@cy-tech.fr)
- POCHIC Arthur: [pochicarh@cy-tech.fr](mailto:pochicarh@cy-tech.fr)

#### **4) Ressources:**

- Etat de l'art détaillé
- Jeu de données détaillé, complet [Nom, Prénom, e-mail, résultats (moyenne générale, score détaillé et calculé), liste ordonnée de vœux] pour les étudiants et pour les admins
- Interface homme/machine sommaire, l'esthétique est secondaire : il faut que ça marche et que la navigation soit simple

#### **5) Délais :**

- 19 mai 2024 : Rendu 1 (état de l'art, cahier des charges, première conception de la solution et organnigramme)
- 26 mai 2024 : Rendu 2 (diagramme UML et prototype figma, rapport final, présentation de soutenance, application avec le jeu de données)

# Notre solution conceptuelle: *Plan et première vue*

## 1) Plan détaillé :

Le sujet d'orientation des filières de 2<sup>ème</sup> année au sein de notre école est une combinaison de problèmes de classification et d'orientation, voire de mariages stables étant donné les places pédagogiques limités et fixes pour chaque filière ainsi que la prise en compte de différentes préférences des étudiants, leurs résultats et leurs mérites afin de bien remplir ces places en toute justesse.

Par conséquent, nous avons choisi une approche combinée basée sur des algorithmes de classification tels que les arbres de décision et la méthode de Naive Bayes puis de l'utilisation de l'algorithme Gale-Shapley pour l'affectation finale en tenant compte des préférences des étudiants et des options. Cette méthode assure une répartition précise et équitable des étudiants selon leurs préférences et leurs performances académiques.

- **Étapes du processus combiné :**

- I. Prédiction avec un Algorithme de classification :**

- *Collecte et préparation des données :*

Nous rassemblons des données historiques sur les étudiants, telles que leur filière, un score pour chaque option, issu de la moyenne générale et des notes obtenues dans certaines matières jugées importantes pour l'option en question ainsi que les critères de sélection des options.

- *Entraînement du modèle de classification :*

Les données collectées sont utilisées pour entraîner notre modèle de classification. Nous utiliserons des arbres de décision et la méthode de Naive Bayes pour prédire l'option la plus appropriée pour chaque étudiant.

- *Prédiction :*

Le modèle prédit pour chaque nouvel étudiant l'option correspondant le mieux à son profil.

- II. Affinage et orientation finale avec un algorithme d'affectation**

- *Regroupement des prédictions :*

En fonction des prédictions effectuées, on peut construire pour chaque option une liste des étudiants appropriés en la triant en fonction des scores des étudiants.

- *Application de l'algorithme d'affectation :*

L'algorithme de Gale-Shapley est utilisé pour assigner les étudiants aux options en tenant compte des prédictions, des préférences des étudiants et des contraintes de capacité des options.

L'algorithme de Gale-Shapley est également connu sous le nom d'algorithme des mariages stables. Nous l'utilisons ici pour trouver une correspondance stable entre les deux ensembles d'éléments que sont les préférences des étudiants et les préférences des options. Cet algorithme garantit que les éléments seront appariés de manière stable.

- *Étapes de l'algorithme :*

1. **Initialisation :** Aucun étudiant n'est affecté à une option
2. **Propositions :** Chaque option propose à l'étudiant en tête de sa liste de préférence auquel elle n'a pas encore proposé.

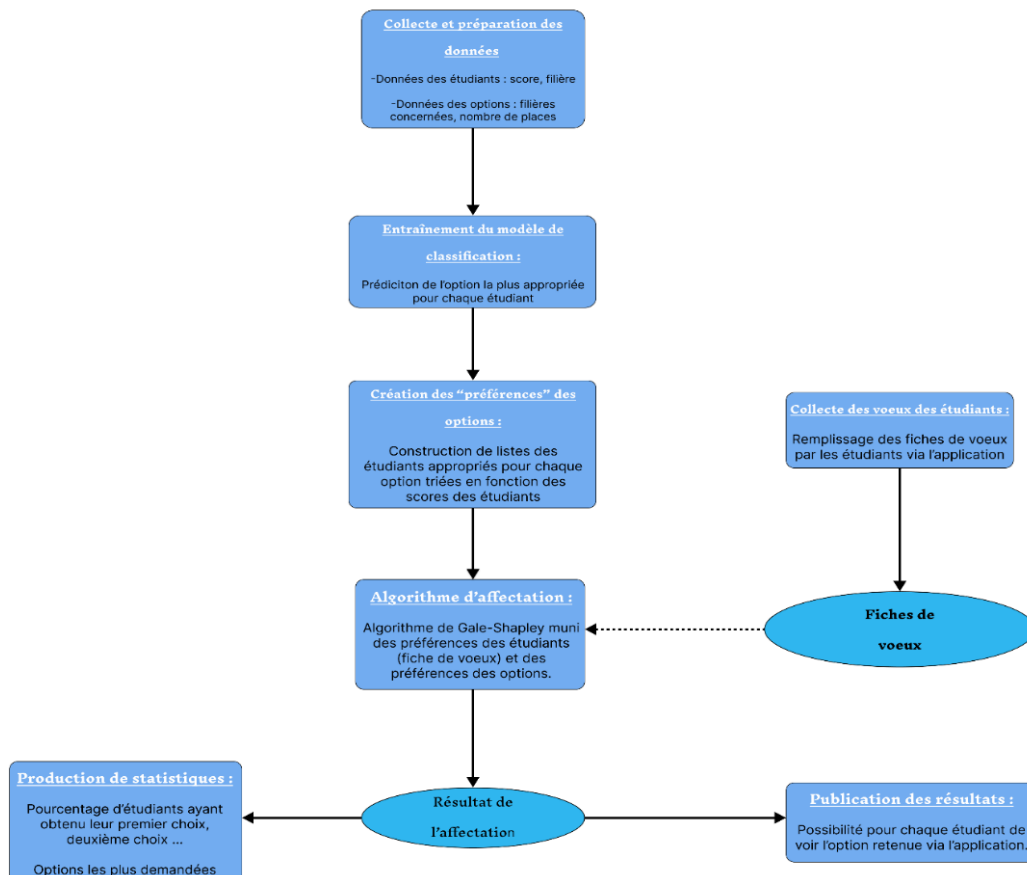
3. **Réponses des étudiants** : On considère les propositions reçues par un étudiant et on accepte temporairement celle de l'option la plus souhaitée par l'étudiant d'après sa fiche de vœux. On rejette les autres propositions.
4. **Rejets et propositions suivantes** : Les options rejetées proposent alors au prochain étudiant sur leur liste de préférence.
5. **Répétition** jusqu'à ce que l'option n'ait plus de place disponible.

• *Avantages de cette approche combinée :*

- **Précision** : La prédiction initiale avec des algorithmes de classification améliore la pertinence des recommandations en se basant sur des données historiques et des caractéristiques individuelles.
- **Optimisation** : L'utilisation d'algorithmes d'affectation garantit que les contraintes de capacité et les préférences globales sont respectées, permettant une répartition optimale et stable.
- **Personnalisation et équité** : Cette approche assure que les préférences des étudiants sont prises en compte de manière personnalisée tout en respectant les critères des options, offrant une solution équitable pour tous.

## 2) Organigramme explicatif :

L'organigramme suivant sert à mettre en exergue toutes les étapes ordonnées et les relations successives entre elles pour le bon déroulement de l'algorithme de notre solution.



# Notre solution conceptuelle: *explication exhaustive*

## 1) Plan détaillé :

En nous basant sur le plan et la première vue de notre solution conceptuelle, nous avons donc décidé d'utiliser des algorithmes de classification et d'affectation. Notre solution se scinde donc en deux parties distinctes.

La première partie constitue la classification des étudiants selon une prédiction de l'option la plus adaptée à leurs compétences et leurs résultats grâce à un algorithme d'arbre de décision. Puis, la seconde partie utilise les résultats fournis par l'arbre de décision ainsi que les fiches des vœux des étudiants pour une affectation finale grâce à l'algorithme de Gale-Shapley.

## 2) Choix du langage de programmation :

Nous avons fait le choix d'utiliser python comme langage de programmation pour notre solution conceptuelle. En effet :

- Python est un excellent choix pour la programmation orientée objet en raison de sa simplicité, de sa flexibilité, de son support robuste pour les concepts de la programmation orientée objet ainsi que son riche écosystème de bibliothèques. En effet, Python offre un support natif robuste pour la POO avec des fonctionnalités comme l'héritage et l'abstraction que nous avons utilisés. De plus, les classes peuvent être facilement définies et instanciées.
- Python dispose de nombreuses bibliothèques dédiées au Machine Learning, telles que **TensorFlow**, **Keras**, **PyTorch**, **Scikit-learn**, et bien d'autres. Ces outils facilitent le développement et le déploiement de modèles de Machine Learning. En outre, des bibliothèques comme **Pandas** et **NumPy** sont excellentes pour la manipulation et l'analyse de données.

## 3) Création des classes :

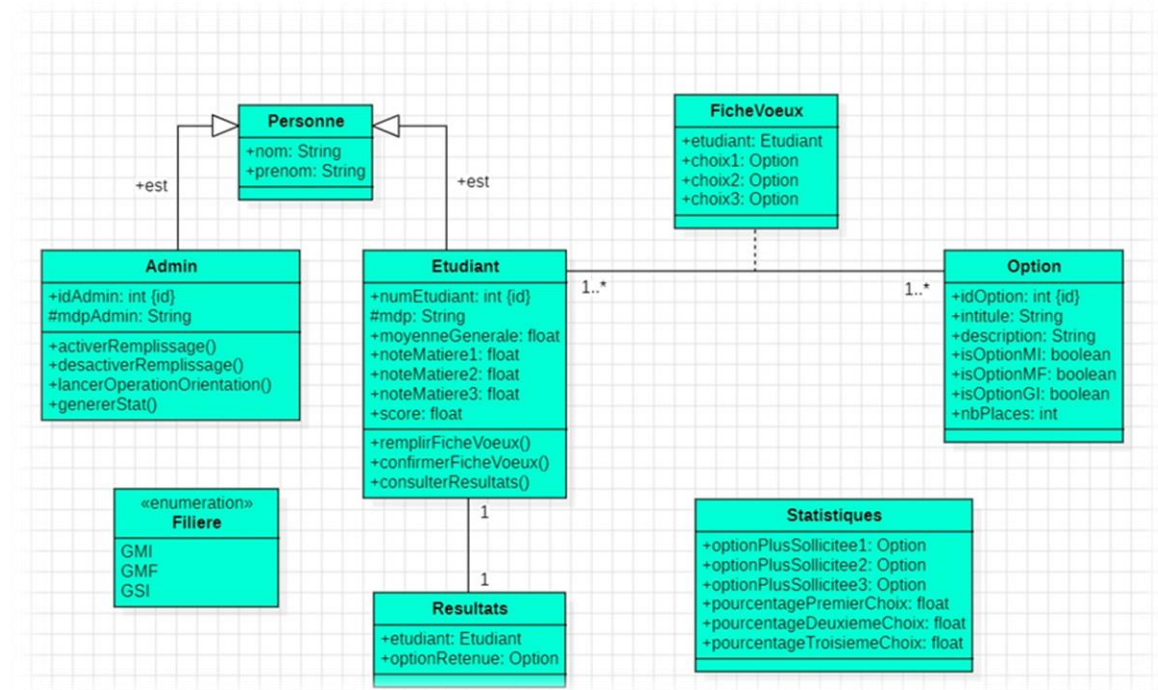
Pour implémenter notre solution, nous avons commencé par définir les classes et les énumérations nécessaires pour représenter les différentes entités de notre système. Voici les classes principales que nous avons développées :

1. **Filière** : une énumération pour représenter les trois différentes filières (MI, MF, GI)
2. **Personne** : Une classe abstraite représentant une personne avec des attributs de base comme le nom et le prénom.
3. **Admin** : Une classe dérivée de Personne représentant un administrateur, avec des méthodes pour gérer le processus de remplissage des vœux et l'orientation.
4. **Etudiant** : Une classe dérivée de Personne représentant un étudiant, avec des méthodes pour remplir et confirmer la fiche de vœux, et consulter les résultats.
5. **Option** : Une classe pour représenter les différentes options disponibles, avec des attributs tels que l'intitulé, la description et le nombre de places.
6. **FicheVoeux** : Une classe pour représenter la fiche de vœux de l'étudiant, contenant les trois choix de l'étudiant.
7. **Résultats** : Une classe pour représenter les résultats de l'orientation de l'étudiant.
8. **Statistiques** : Une classe pour générer et afficher des statistiques sur les choix des étudiants et les options les plus sollicitées.

Pour chaque classe, nous avons défini des attributs étant utiles pour la mise en oeuvre de la solution dans son ensemble.

Afin de représenter les différentes opérations faites par les classes, nous avons également défini des méthodes qui peuvent correspondre à des calculs ou des créations de nouvelles variables ou bien à la réponse attendue par les boutons de l'interface utilisateur.

Pour une meilleure visualisation de ces classes, de leur rôle et de leurs liens entre elles, nous avons réalisé ce diagramme de classe UML :



Notre objectif lors de la création de ces classes était de fournir des données plus importantes et plus précise sur les résultats académiques des étudiants. C'est pourquoi nous avons décidé dans un premier temps de prendre en compte leurs notes dans différentes matières. L'idée était ensuite de créer des scores différents pour chaque option qui prendrait davantage en considération les matières que nous jugeons les plus importantes pour cette option. Par exemple, un étudiant de GMI pourrait avoir un score élevé pour l'option Data Science grâce à une note élevée en Data Mining 2 et un score plus faible pour l'option IA à cause d'une mauvaise note en IA : application.

Face à la difficulté de mise en œuvre, nous avons finalement opté pour un seul score par étudiant. Ce score se base principalement sur les matières de mathématiques et d'informatique jugées importantes pour l'ensemble des options proposées par la filière de l'étudiant en question. Nous gardons tout de même également la moyenne générale comme variable utilisée pour l'arbre de décision.

*Voici les formules adaptées pour le calcul du score :*

## Score de la MI:

**Score** = **5** \* « IA: applications » + **4.5** \* « DataMining 2 » + **3** \* « Décidabilité & complexité » + **3** \* « Architecture et prog parallèle » + **2.5** \* « Compressive Sensing » + **2** \* « Intro aux Séries temporelles »



## Score de la MF:

Score = 4.5\* « Contingent Claims Valuation » + 3.5\* « Proba » + 3\* « Introduction à la finance » + 3\* « Méthodes Num Av pour EDP » + 2.5\* « EQ aux dérivées partielles » + 2\* « Portfolio Management & Financial Risks » + 1.5\* « Introduction à l'assurance »

## Score de la GSI:

Score = 3\* « IA: applications » + 4.5\* « IA : théories et algorithmes » + 3\* « Cybersécurité opérations » + 3\* « Architecture réseau » + 2.5\* « Développement distribué Java EE » + 2.5\* « Architecture et prog parallèle » + 1.5\* « Programmation fonctionnelle »

### 4) Implémentation des algorithmes :

#### a- Arbre de décision :

L'arbre de décision a été utilisé pour prédire les affectations des étudiants en fonction de leur moyenne et de leur score. Pour cet algorithme, nous avons eu besoin de la bibliothèque Scikit-learn afin de créer l'arbre de décision et de l'entraîner sur notre base de données.

Cet arbre de décision a permis de faire une première classification en affectant à chaque étudiant une prédiction de l'option la plus adaptée à ses compétences et ses résultats.

#### b- Algorithme de Gale-Shapley :

Nous avons donc utilisé l'algorithme de Gale-Shapley pour l'affectation finale des étudiants aux options. Pour cela, nous avons la fiche de vœux de chaque étudiant qui constitue la liste des "préférences" des étudiants. Pour obtenir la liste des "préférences" des options, nous avons utilisé les résultats obtenus par l'utilisation de l'arbre de décision. En effet, l'arbre de décision nous donne une prédiction de l'option la plus adaptée pour chaque étudiant. Nous pouvons alors former pour chaque option une liste d'étudiant adaptés que nous trions par moyenne générale décroissante.

L'algorithme de Gale-Shapley est utilisé pour résoudre le problème des mariages stables. C'est un problème classique en théorie des jeux et en mathématiques combinatoires. Il concerne deux ensembles égaux (traditionnellement hommes et femmes) où chaque individu a une liste de préférences ordonnée pour l'ensemble opposé. L'objectif est de trouver un appariement tel qu'aucun homme et aucune femme ne préfèrent être ensemble plutôt qu'avec leurs partenaires actuels, ce qui rend l'appariement stable.

- **Définition du Problème :**

- **Ensembles :**

- Un ensemble de  $n$  hommes.



- Un ensemble de  $n$  femmes.
- **Préférences :**
  - Chaque homme a une liste ordonnée de toutes les femmes, selon ses préférences.
  - Chaque femme a une liste ordonnée de tous les hommes, selon ses préférences.
- **Appariement stable :**
  - Aucun homme et aucune femme ne préfèrent être ensemble plutôt qu'avec leurs partenaires actuels.

L'algorithme de Gale-Shapley (ou algorithme des mariages stables) garantit toujours un appariement stable. Les hommes proposent aux femmes suivant l'ordre de leur liste de préférence puis les femmes acceptent ou non les propositions selon l'ordre de leur liste de préférence.

L'analogie du problème des mariages stables peut être directement appliquée à notre problème des étudiants et des options dans un contexte éducatif :

- **Étudiants et Options :**
  - Un ensemble d'étudiants qui ont des préférences pour différentes options.
  - Un ensemble d'options qui ont des préférences pour différents étudiants.
- **Préférences :**
  - Chaque étudiant a une liste ordonnée des options qu'il préfère.
  - Chaque option a une liste ordonnée des étudiants qu'elle préfère.
- **Capacités des options :**
  - Chaque option a une capacité limitée, c'est-à-dire un nombre maximum d'étudiants qu'elle peut accepter.

Dans notre cas, les étudiants jouent le rôle des hommes et les options jouent le rôle des femmes. Les étudiants proposent aux options de leur liste de préférences dans l'ordre de préférence et les options acceptent ou non les propositions des étudiants en fonction de leur liste de préférence. On pourrait choisir d'inverser les rôles mais le fait que les étudiants proposent donne plus d'importance aux choix qu'ils ont fait dans leur liste de vœux.

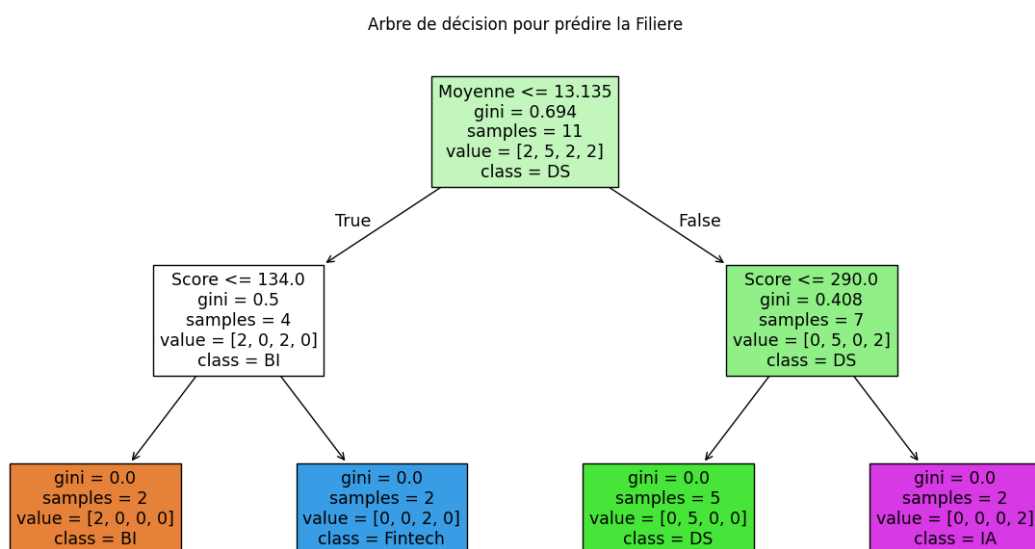
## **5) Préparation des jeux de données et tests :**

Nous avons utilisé des fichiers Excel et PDF pour stocker les données des étudiants et les lire dans notre programme. Pour cela, nous avons utilisé les bibliothèques pandas et tabula pour lire ces fichiers et convertir les données en DataFrame. Pour tester et valider notre solution, nous avons utilisé un jeu de données cohérent et fourni pour les tests. Les données ont été lues à partir des fichiers Excel et PDF, et nous avons utilisé ces données pour entraîner et tester notre modèle d'arbre de décision ainsi que pour valider l'algorithme de Gale-Shapley.

*Voici un exemple de jeu de données que nous avons utilisé pour tester nos algorithmes :*

Identifiant	Nom	Prenom	Moyenne	Score	Filiere	Affectation
10000000	Jade	Martin	18.79	375	GMI	IA
10000001	Louise	Bernard	17.2	350	GMI	IA
10000002	Ambre	Thomas	16.2	300	GMI	IA
10000007	Romy	Moreau	15.2	280	GMI	DS
10000008	Anna	Laurent	14.32	260	GMI	DS
10000009	Lina	Simon	14.25	263	GMI	DS
10000010	Gabriel	Michel	14.2	258	GMI	DS
10000013	Maelle	Roux	14	255	GMI	DS
10000014	Louise	David	12.27	170	GMI	Fintech
10000019	Lucas	Bonnet	12.25	174	GMI	Fintech
10000020	Aziz	Ahmed	10.78	98	GMI	BI
10000021	Rouge	Mourad	10.58	95	GMI	BI

- L'arbre de décision obtenu par l'algorithme avec ce jeu de données est le suivant :



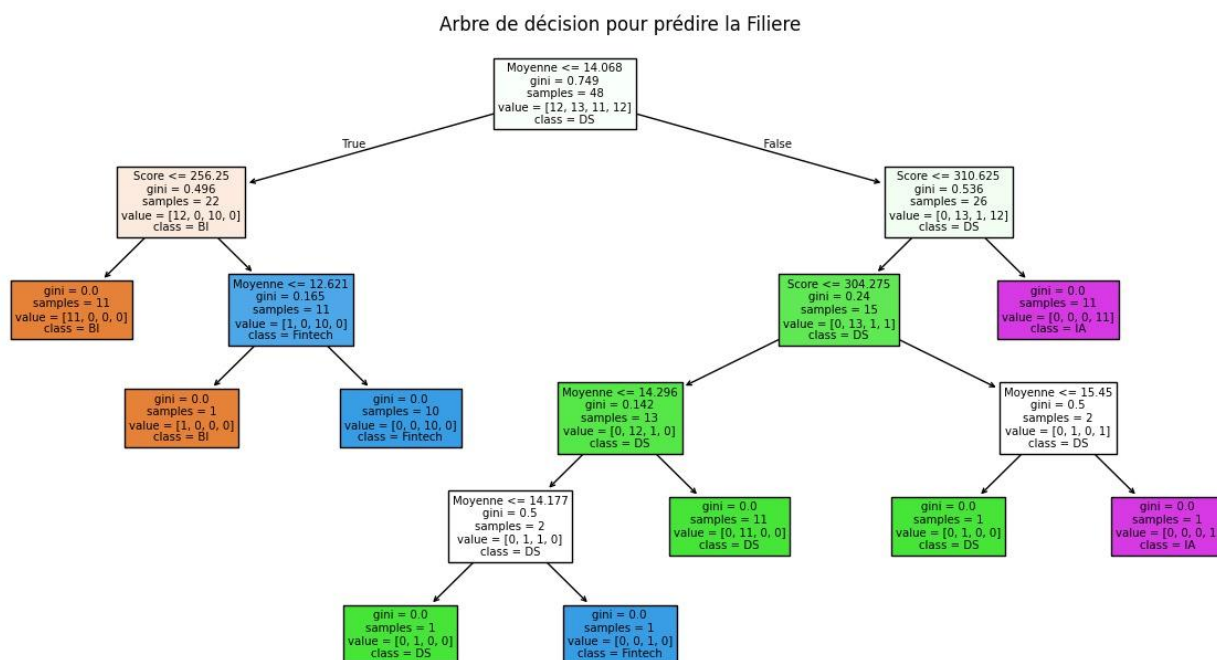
Dans un premier temps nous avons comme objectif de rentrer les notes des étudiants dans différentes matières à l'aide d'un dictionnaire de clés et de valeurs. Nous avons donc créé une fonction **ajouterNote** pour la classe **Etudiant**. Nous avons également créé une fonction permettant de calculer le score en prenant en paramètres un étudiant et une option. Cette fonction utilise les notes de différentes matières et leur affecte des coefficients afin de calculer le score attendu :

```
def calculer_score(etudiant, option):
    score=0
    if etudiant.filiere == "MI":
        score += 4.5*etudiant.notes["Data Mining 2"] + 5*etudiant.notes["IA:application"] + \
        2.5*etudiant.notes["Compressive Sensing"] + 2*etudiant.notes["Introduction Séries Temporelles"] + \
        3*etudiant.notes["Décidabilité et complexité"] + \
        3*etudiant.notes["Architecture et Programmation Parallèle"]
    if etudiant.filiere == "MF":
        score += 4.5*etudiant.notes["Contingent Claims Valuation"] + 3.5*etudiant.notes["Proba"] + \
        3*etudiant.notes["Introduction à la finance"] + 3*etudiant.notes["Méthodes Num Av pour EDP"] + \
        2.5*etudiant.notes["Equations aux dérivées partielles"] + \
        2*etudiant.notes["Portfolio Management & financial risks"] + 1.5*etudiant.notes["Intro à l'assurance"]
    if etudiant.filiere == "GI":
        score += 4.5*etudiant.notes["IA:théories et algorithmes"] + 3*etudiant.notes["IA:applications"] + \
        3*etudiant.notes["Cybersécurité opérations"] + 3*etudiant.notes["Architecture Réseaux"] + \
        1.5*etudiant.notes["Programmation fonctionnelle"] + 2.5*etudiant.notes["Architecture et Programmation parallèle"] \
        + 2.5*etudiant.notes["Développement distribué Java EE"]
    return score
```

Pour un traitement plus simple des données, nous avons entré directement le score des étudiants dans la base de données sans passer par un calcul.

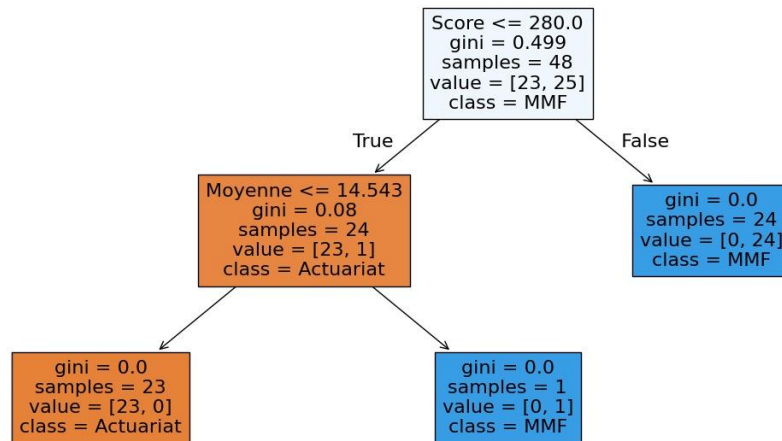
Nous avons également scindé le jeu de données en fonction des filières pour faciliter le traitement des données.

L'arbre de décision a ensuite été testé pour les différentes filières. **Pour le jeu de données correspondant à la filière GMI, nous avons obtenu l'arbre de décision ci-dessus :**



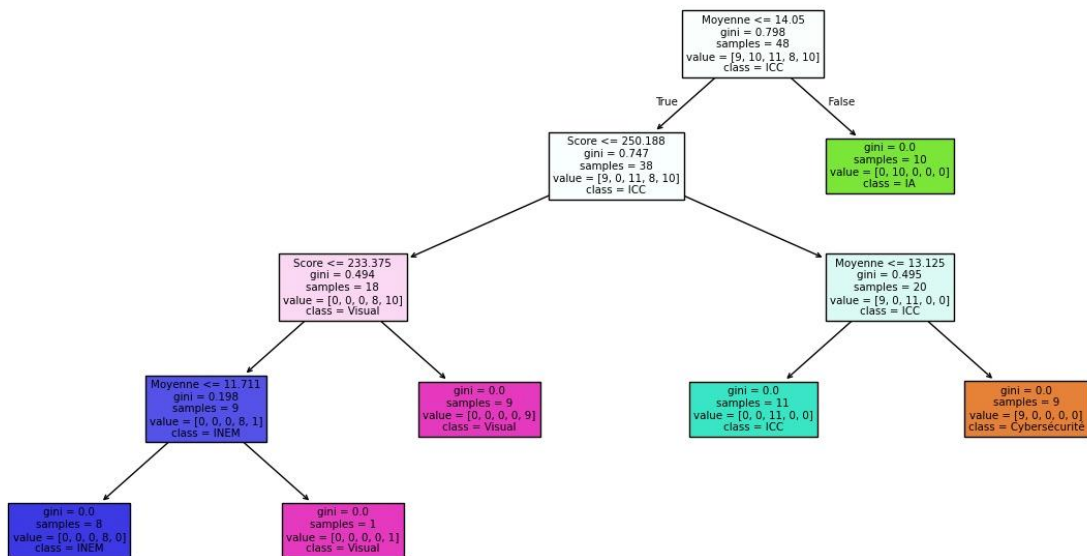
Pour le jeu de données de la filière GMF, nous avons obtenu l'arbre de décision suivant :

Arbre de décision pour prédire la Filière



Pour le jeu de données de la filière GSI, nous avons obtenu l'arbre de décision ci-dessous :

Arbre de décision pour prédire la Filière

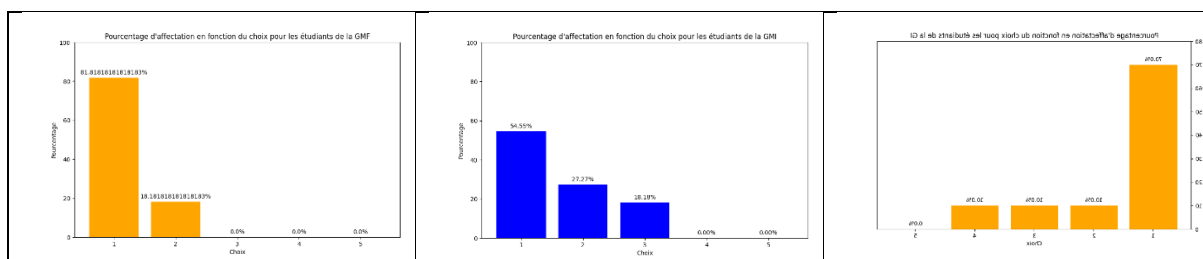


## 6) Statistiques sur les résultats

Après obtention des affectations finales par l'algorithme de Gale-Shapley, nous avons décidé d'implémenter des fonctions en python pour calculer des statistiques sur ces résultats. Le but de cette démarche est de vérifier si notre solution nous fournit des résultats satisfaisants et justes mais également de nous fournir des informations sur la popularité des options en vue d'éventuelles modifications de notre solution. Avec la bibliothèque matplotlib, nous avons pu afficher une représentation graphique de ces statistiques sous forme d'histogrammes.

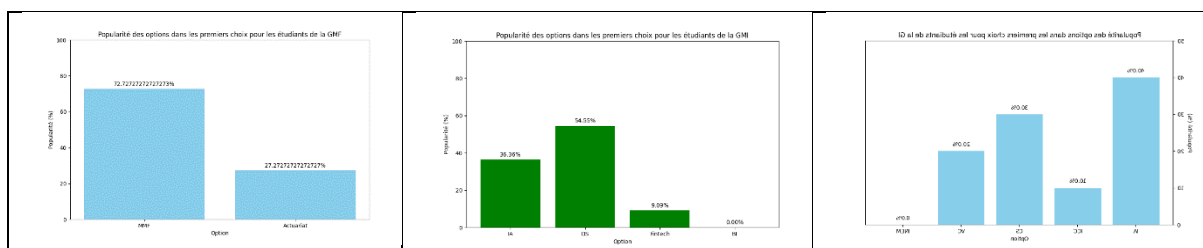
Nous avons donc implémenté une fonction `calculate_choice_statistics` qui prend en paramètre les fiches de vœux des étudiants ainsi que l'affectation finale par Gale-Shapley et retourne les pourcentages d'étudiants ayant été affectés à leur premier choix, leur deuxième choix, etc.

Voici le résultat obtenu pour notre base test :



Nous avons également implémenté une fonction `calculate_first_choice_popularity` qui prend en paramètre les fiche de vœux des étudiants et retourne la fréquence d'apparition de chaque option en premier choix.

Voici le résultat obtenu pour notre base test :



## 7) Interface utilisateur :

Nous avons également implémenté une interface utilisateur simple à l'aide des deux bibliothèques dédiées à la création des interfaces graphiques « Tkinter » et « customtkinter » en Python. Cette interface permet aux étudiants de se connecter, remplir leur fiche de vœux et consulter leurs résultats, et aux administrateurs de gérer le processus de remplissage des vœux et de lancer l'orientation.

Dans un premier temps, nous avons créé un prototype pour cette interface à l'aide de l'application **Figma**. Nous avons ainsi pu présenter une vue de chaque page et les liens entre les différentes pages.

## 8) Conclusion :

La mise en œuvre de notre projet « CY Tech Oriente » a permis de développer une solution algorithmique pour l'orientation des étudiants en fonction de leurs préférences et de leurs résultats académiques. En utilisant des méthodes avancées telles que les arbres de décision et l'algorithme de Gale-Shapley, nous avons pu créer un processus d'orientation plus équitable et plus efficace. L'interface utilisateur développée permet également de rendre ce processus plus transparent et accessible pour les étudiants et les administrateurs.

# Répartition des tâches

## 1) Première partie :

- **Etat de l'art, recherches bibliographiques :** Atig Tasnim, Ben Mansour Youssef
- **Cahier des charges :** Bowé Romain
- **Organigramme, diagramme UML :** Pochic Arthur

## 2) Deuxième partie :

- **Création des jeux de données :** Ben Mansour Youssef
- **Réflexion et choix de la solution théorique, choix des algorithmes utilisés, choix du calcul des scores :**

Atig Tasnim, Ben Mansour Youssef, Bowé Romain, Pochic Arthur

- **Interface utilisateur :** Bowé Romain (*recherche et choix des bibliothèques*), Atig Tasnim (*conception et implémentation*)
- **Implémentation Python :** (*classes, arbre de décision, Gale-Shapley*)

Atig Tasnim, Pochic Arthur

- **Création de la présentation pour la soutenance :** Bowé Romain, Atig Tasnim
- **Rédaction du rapport :** Atig Tasnim, Bowé Romain, Pochic Arthur, Ben Mansour Youssef

## Synthèse et Regard critique

Nous avons essayé de proposer deux solutions pour l'affectation des étudiants. La première solution consiste à créer une matrice où sont stockés les identifiants de tous les élèves ainsi que leurs préférences. Une deuxième matrice est également utilisée pour stocker les préférences pour les options disponibles. Ensuite, nous appliquons l'algorithme de Gale-Shapley pour tenter d'affecter les étudiants de manière à obtenir un couplage stable. Cependant, cette solution s'avère très coûteuse en termes de ressources et de temps de calcul.

Pour optimiser cette solution, nous avons envisagé une approche basée sur un jeu de données contenant les scores de chaque filière. En utilisant des techniques de datamining, comme les arbres de décision, combinées avec l'algorithme de Gale-Shapley, nous pouvons affecter les étudiants de manière plus efficace et plus simple. De plus, nous avons essayé de développer un score spécifique pour chaque filière afin de garantir une égalité entre tous les étudiants. Malheureusement, nous n'avons pas pu aboutir à cette solution car nous souhaitions maintenir le jeu de données aussi simple que possible.

**Complexité et ressources :** La première solution, bien que conceptuellement solide, présente une complexité algorithmique élevée. Le stockage et le traitement de grandes matrices demandent des ressources informatiques importantes. De plus, l'algorithme de Gale-Shapley, malgré son efficacité pour trouver un couplage stable, devient moins pratique à mesure que le nombre d'étudiants et d'options augmente. Cette complexité peut rendre la solution difficile à déployer dans des environnements avec des contraintes de ressources.

**Optimisation et simplification :** La deuxième solution, qui combine datamining et l'algorithme de Gale-Shapley, offre une amélioration notable en termes d'efficacité. Cependant, l'approche de maintenir la simplicité du jeu de données pour éviter la surcharge de complexité a ses propres limites. En essayant de garantir une égalité entre les étudiants avec des scores spécifiques pour chaque filière, nous avons été confrontés à des défis méthodologiques et techniques. La difficulté de développer un score spécifique pour chaque filière tout en conservant la simplicité a empêché l'implémentation complète de cette solution.

**Équité et efficacité :** L'algorithme de Gale-Shapley, bien qu'efficace pour créer des appariements stables, ne prend pas en compte les scores spécifiques des filières, ce qui pourrait entraîner des déséquilibres dans l'affectation. La tentative d'introduire des techniques de datamining visait à pallier ce problème, mais la simplification excessive du jeu de données a limité notre capacité à mettre en œuvre une solution parfaitement équitable et efficace.

**Perspectives d'amélioration :** Pour aller au-delà des limitations actuelles, une approche hybride pourrait être envisagée. Par exemple, combiner l'algorithme de Gale-Shapley avec des méthodes de machine Learning plus avancées pour prédire les préférences et optimiser l'affectation de manière dynamique pourrait offrir une solution plus équilibrée. De plus, une évaluation approfondie des ressources nécessaires et des compromis entre complexité et simplicité serait essentielle pour affiner ces solutions.

En adoptant une approche plus nuancée et multidimensionnelle, utilisant plusieurs scores, il est possible de créer un système d'affectation plus juste et équitable. Cela permet de reconnaître et de valoriser la diversité des talents, des compétences et des parcours des étudiants, garantissant ainsi que chaque élève a les meilleures chances de réussir dans la filière qui correspond le mieux à ses capacités et à ses aspirations.



# Références bibliographiques

- [1] Ministère de l'enseignement supérieur et de la recherche : *Document de présentation des algorithmes de Parcoursup 2023* : [https://gitlab.mim-libre.fr/parcoursup/algorithmes-de-parcoursup/-/blob/1a2d806c3fb9b583fbef68970a1a9980626bfafo/doc/presentation\\_algorithmes\\_parcoursup\\_2023.pdf](https://gitlab.mim-libre.fr/parcoursup/algorithmes-de-parcoursup/-/blob/1a2d806c3fb9b583fbef68970a1a9980626bfafo/doc/presentation_algorithmes_parcoursup_2023.pdf)
- [2] Jules GRENET : *Les algorithmes d'affectation dans le système éducatif français* : <https://hal.science/hal-03686962/document>
- [3] KANDUKI KIVUYIRWA MYSTERE, HERITIER NSENGE MPIA, and MUTEGHEKI BARAKA VINGI: *Prédiction de l'orientation des étudiants dans des filières d'études appropriées en utilisant les techniques de Data Mining* : [https://www.researchgate.net/profile/Heritier-Mpia/publication/368653622\\_Prediction\\_de\\_l'orientation\\_des\\_etudiants\\_dans\\_des\\_filiere\\_d'etudes\\_appropriees\\_en\\_utilisant\\_les\\_techniques\\_de\\_Data\\_Mining/links/63f312c119130a1a4a92baao/Prediction-de-lorientation-des-etudiants-dans-des-filiere-detudes-appropriees-en-utilisant-les-techniques-de-Data-Mining.pdf](https://www.researchgate.net/profile/Heritier-Mpia/publication/368653622_Prediction_de_l'orientation_des_etudiants_dans_des_filiere_d'etudes_appropriees_en_utilisant_les_techniques_de_Data_Mining/links/63f312c119130a1a4a92baao/Prediction-de-lorientation-des-etudiants-dans-des-filiere-detudes-appropriees-en-utilisant-les-techniques-de-Data-Mining.pdf)
- [4] OUATIK FAROUK : *Contribution à la prédiction de la réussite et l'orientation des étudiants à base de E-learning et du Big Data*
- [5] Leïla Frouillou, Clément Pin, Agnès van Zanten : *Le rôle des instruments dans la sélection des bacheliers dans l'enseignement supérieur. La nouvelle gouvernance des affectations par les algorithmes* | Cairn.info
- [6] *Orientation et numérique - 2* Numéro thématique sous la coordination de Claude Houssement et Even Loarer
- [7]: Vincent Bebie : *Graphe et éthique : une contribution à Parcoursup* : <https://dumas.ccsd.cnrs.fr/dumas-03843917v1/document>