

Rapport : Arbre Généalogique

ING1 GM - Groupe 8



TECH

Salim OUESLATI

Bacarie TCHABO

Mehdi TORJMEN

Roshanth RUBAN

Iyad BEN MOSBAH

Kathirvele RANGANADANE

Remerciements

Nous tenons tout d'abord à remercier et à faire preuve de reconnaissance envers ces personnes, qui nous ont non seulement permis de prendre connaissance de ce projet, mais qui nous ont aussi encadré et aidé pour sa réalisation :

- o Mme. Zaouche Djaouida, encadrante du projet sur l'arbre généalogique en première année du cycle pré-ingénieur (ING 1) en filière Génie Mathématiques et Informatique (GMI) à CY Tech en 2023-2024. Ses conseils précis, sa sympathie et sa patience nous ont été d'une grande aide quant à la résolution de plusieurs difficultés conceptuelles et techniques.

- o CY Tech et la totalité des enseignants et travailleurs au sein de l'école qui par leur coopération, rigueur et assiduité, nous ont permis tout au long de la réalisation de ce projet d'avoir accès à un environnement académique riche, plein de savoir-faire et d'expertise.

Sommaire

I - Introduction :	4
A - Description des variables.....	4
B - Manipulation du fichier familles.csv.....	5
C - Consultation de l'arbre généalogique global.....	5
D - Prédiction des risques de survenue d'un diabète.....	5
E - Interface graphique.....	5
II - Création de la base de données familles.csv :	6
A. Structure de la base de données.....	6
B. Contraintes de la base de données.....	7
III. Création de l'interface pour visualiser l'arbre (Java).....	8
A. Programmation orientée objet.....	8
B. Fonctionnalités de l'interface.....	9
IV. Intégration des fonctionnalités liées au diabète.....	13
A. Enrichissement et analyse de la base de données.....	13
B. Visualisation des prédictions de risque.....	14
V. Conclusion.....	17
Annexes.....	18

I - Introduction :

Dans le cadre de ce projet que nous sommes amenés à réaliser lors du second semestre de notre ING 1, l'objectif est de concevoir et d'implémenter une application permettant la création et la gestion d'arbres généalogiques.

Plus précisément, l'application doit permettre de construire un arbre généalogique global à partir d'un fichier CSV, nommé *familles.csv*. Il s'agit d'un jeu de données/dataset dont les lignes sont des individus (on dira aussi des personnes) et dont les colonnes sont des variables.

A - Description des variables

Parmi ces variables, certaines sont quantitatives et d'autres sont qualitatives. Parmi les variables quantitatives, certaines sont discrètes et d'autres sont continues. Les variables quantitatives discrètes sont :

- L'identifiant unique
- L'identifiant du père
- L'identifiant de la mère

La seule variable quantitative continue est le poids.

Parmi les variables qualitatives, il y en a une qui est ordinale (la date de naissance), certaines sont cardinales et d'autres dichotomiques (booléennes). Les variables qualitatives cardinales sont :

- Le nom
- Le prénom
- La nationalité

Les variables qualitatives dichotomiques sont :

- Présence du diabète (Outcome)
- Présence d'antécédents de diabète

Les données doivent rester cohérentes lorsque le fichier *familles.csv* est modifié. Pour ce faire, des règles de cohérence doivent être définies.

B - Manipulation du fichier *familles.csv*

Afin de manipuler le fichier *familles.csv*, il faut rendre compte des opérations suivantes :

- Vérification de la cohérence du fichier
- Nettoyage du fichier
- Suppression une ligne du fichier
- Ajout d'une ligne au fichier

C - Consultation de l'arbre généalogique global

À partir du fichier *familles.csv*, il faut pouvoir visualiser :

- L'arbre généalogique global
- Une partie de l'arbre généalogique relatif à une famille donnée
- La descendance d'une personne donnée
- L'ascendance d'une personne donnée
- Les frères, sœurs ou autres proches d'une personne donnée
- La liste des personnes sans ascendance

D - Prédiction des risques de survenue d'un diabète

Cette partie du projet consiste à mettre en place un système de prédiction des risques de survenue d'un diabète. Ces prédictions doivent se baser sur des facteurs comme les antécédents de diabète, les types de diabète ou encore les âges de diagnostic. Afin de diviser les membres d'une même famille en clusters homogènes en fonction de leurs caractéristiques, il faut utiliser une méthode de clustering comme K-means. Les clusters ainsi formés représentent des groupes de membres de la famille partageant des similitudes dans leurs antécédents de diabète, leur type de diabète ou encore leur âge.

E - Interface graphique

L'application devra fournir une interface graphique pour visualiser et modifier facilement l'arbre généalogique global à partir des données du fichier *feuilles.csv*.

II - Création de la base de données *familles.csv* :

A. Structure de la base de données

Table Personne :

Attribut	Type	Description
ID Personne	int	Clé primaire, identifiant unique pour chaque individu
ID Père	int	Identifiant unique pour le père de chaque individu

ID Mère	int	Identifiant unique pour la mère de chaque individu
ID Conjoint	int	Identifiant unique pour le conjoint de chaque individu
Nom	String	Nom de famille de la personne
Prénom	String	Prénom de la personne
Date Naissance	Date	Date de naissance de la personne
Nationalité	String	Nationalité (champ optionnel) de la personne
Enfants	Liste de Personne	La liste des enfants de type Personne de la personne concernée
Sexe	boolean	Masculin ou féminin
IMC	float	Indice de masse corporelle de la personne

Taux d'insuline	int	Taux d'insuline de la personne
Taux de glucose	int	Taux de glucose dans le sang de la personne
Outcome	boole an	Présence ou non du diabète chez la personne

B. Contraintes de la base de données

La base de données doit respecter des règles de validation strictes pour s'assurer de l'intégrité et de la fiabilité des données enregistrées dans le fichier *familles.csv*. Ces règles sont conçues pour préserver l'uniformité et l'exactitude des renseignements concernant des personnes spécifiques et leurs relations familiales. Les critères suivants doivent être respectés :

1. L'unicité des identifiants : Chaque individu a un identifiant unique. La valeur de cet identifiant ne doit jamais être modifiée. L'identifiant permettra d'identifier de manière unique chaque personne de l'arbre généalogique.
2. Le format de la date de naissance : Chaque individu a une date de naissance qui doit être valide, et de plus, le format de la date doit être correct (AAAA-MM-JJ). On doit vérifier la syntaxe de la date en veillant à bien respecter le format donné et à vérifier que les valeurs des jours, des mois et des années sont dans des plages raisonnables et cohérentes.
3. L'existence de parents : Les identifiants des parents indiqués pour chaque individu doivent figurer dans la base de données. On doit vérifier que les identifiants des parents correspondent à des individus réels présents dans le fichier. Cette vérification permet de veiller à ce que chaque individu soit bien lié à plusieurs parents dans l'arbre. Seules la première génération et les personnes extérieures à la famille directe (épouse) sont exemptées de cette règle.
4. Vérifiez la cohérence des dates de naissance : La date de naissance d'un enfant devrait toujours être ultérieure (écart de 18 ans minimum) à celle de ses parents. On doit vérifier que la date de naissance d'un enfant est postérieure aux dates de naissance de son père et de sa mère. Cette règle sert à empêcher toute incohérence chronologique et assure la validité des liens de parenté.

5. L'absence de cycles dans les relations : On ne peut pas créer de cycle dans l'arbre généalogique, c'est-à-dire que toute personne ne peut pas être son propre parent, plusieurs fois grand-parent, etc. Cela donnerait des incohérences dans les relations, on doit donc faire attention à cela lors de la création de la base de données.

De cette manière, ces règles strictes peuvent aider à maintenir la qualité et la fiabilité des données de la base de données pour l'arbre généalogique et donc éviter les erreurs lors de la programmation orientée objet.

III. Création de l'interface pour visualiser l'arbre (Java)

A. Programmation orientée objet

Afin de déterminer les arbres généalogiques issus du fichier *familles.csv*, nous avons procédé comme ceci :

Nous avons créé une classe **Personne** contenant les attributs susmentionnés, notamment l'attribut « Enfants » dont le type est une liste d'éléments de type *Personne*.

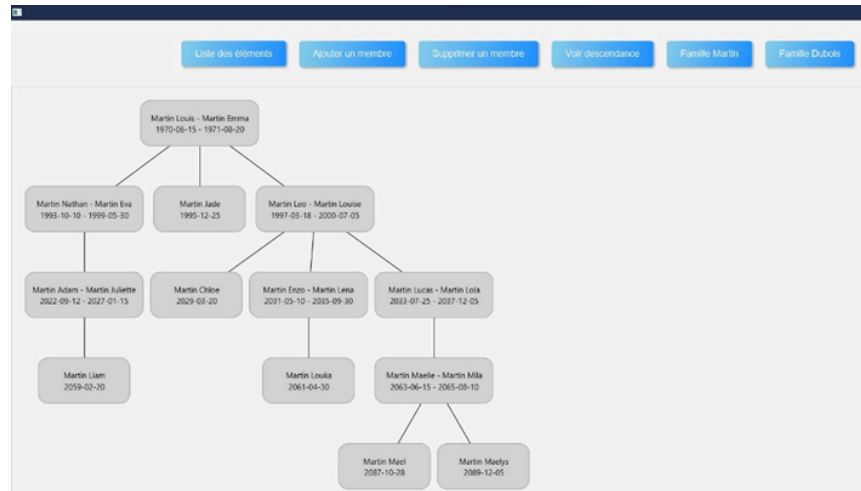
Nous avons ensuite créé une classe **Arbre**. Cette dernière contient toutes les méthodes nous permettant de créer l'arbre généalogique

- Nous disposons d'une méthode utilisant la bibliothèque *OpenCSV*. Cette méthode nous permet de créer un flux de données entre le fichier *familles.csv* et le code. Grâce à ce flux, nous lisons itérativement chaque ligne du fichier pour créer une instance de la classe **Personne**, chaque nouvel objet de type *Personne* est stocké dans une *HashMap* dont la clé est l'identifiant de ladite personne et la valeur est l'instance de la classe *Personne* associée.
- Afin de créer l'arbre, nous disposons de plusieurs méthodes : La première nous permet de retourner une liste d'objets de type *Personne* représentant les enfants d'un individu spécifique. Cette dernière prend en paramètre l'identifiant de la personne dont nous recherchons les enfants. Nous itérons la *HashMap* afin de chercher les individus dont l'attribut « ID père » est égal à l'identifiant passé en paramètre. Si c'est le cas, la personne est ajoutée à la liste.
- La seconde construit l'arbre de hiérarchie généalogique en ajoutant récursivement les enfants à chaque personne à partir d'une racine donnée. Dans cette méthode, nous commençons par déterminer les enfants de la racine : nous obtenons alors une liste d'objets de type « *Personne* » qui sont les enfants de la racine. Nous parcourons cette liste et pour chaque élément de la liste, nous déterminons ses enfants.

Ensuite, nous utilisons la bibliothèque *Abego TreeLayout Core* afin de représenter graphiquement les arbres généalogiques.

B. Fonctionnalités de l'interface

Sur la page principale de l'interface, nous affichons l'ensemble des arbres généalogiques. Nous affichons également plusieurs boutons relatifs aux fonctionnalités de l'application.



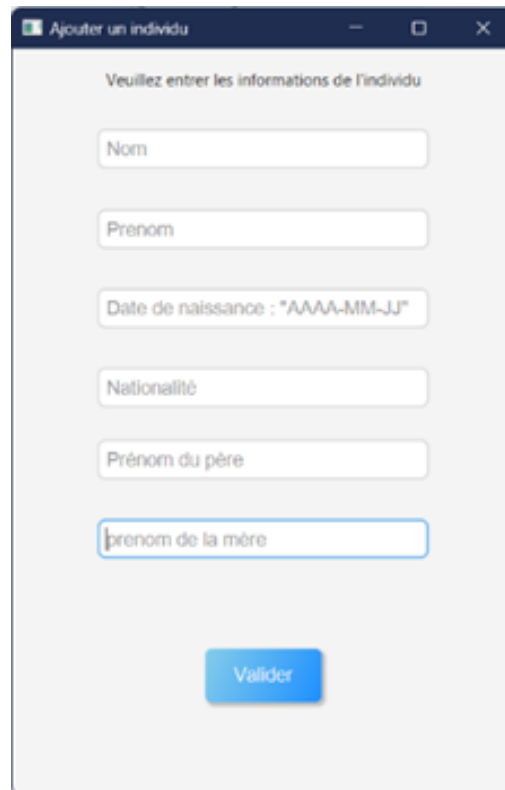
Affichage d'un exemple d'arbre généalogique

Le premier bouton permet à l'utilisateur d'afficher la liste des personnes présentes dans le fichier *familles.csv*. On affiche chaque attribut ainsi que le nom et prénom du père et de la mère.

Membres des familles					
Nom	Prénom	Date de naissance	Nationalité	Père	Mère
Dubois	Alice	1970-05-15	FR	Inconnu	Inconnu
Dubois	Louis	1971-07-20	FR	Inconnu	Inconnu
Dubois	Charlotte	1993-09-10	FR	Inconnu	Inconnu
Dubois	Hugo	1995-11-25	FR	Dubois Louis	Dubois Alice
Dubois	Emma	1997-02-18	FR	Inconnu	Inconnu
Dubois	Gabriel	1999-04-30	FR	Dubois Louis	Dubois Alice
Dubois	Chloe	2000-06-05	FR	Dubois Louis	Dubois Alice
Dubois	Lucas	2022-08-12	FR	Dubois Hugo	Dubois Charlotte
Dubois	Eva	2024-10-28	FR	Dubois Hugo	Dubois Charlotte
Dubois	Mathis	2026-12-15	FR	Dubois Hugo	Dubois Charlotte
Dubois	Lea	2028-02-20	FR	Inconnu	Inconnu
Dubois	Nathan	2025-04-10	FR	Dubois Gabriel	Dubois Emma
Dubois	Manon	2028-06-25	FR	Inconnu	Inconnu
Dubois	Raphael	2054-08-30	FR	Dubois Mathis	Dubois Lea
Dubois	Zoe	2056-10-05	FR	Dubois Mathis	Dubois Lea
Dubois	Louise	2058-12-20	FR	Inconnu	Inconnu
Dubois	Noah	2050-02-25	FR	Dubois Nathan	Dubois Manon
Dubois	Maeva	2052-04-15	FR	Dubois Nathan	Dubois Manon
Dubois	Arthur	2084-06-01	FR	Dubois Raphael	Dubois Louise

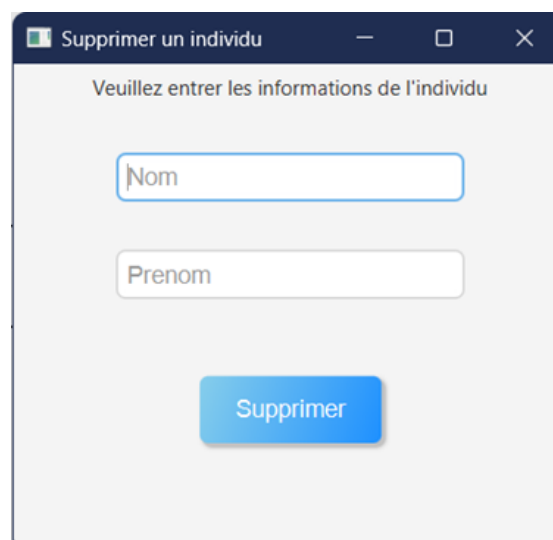
Affichage du fichier *familles.csv*

Le deuxième bouton est un formulaire nous permettant d'ajouter un individu : si les contraintes susmentionnées sont respectées par l'utilisateur, l'individu est ajouté. Le cas échéant, on affiche un message d'erreur.



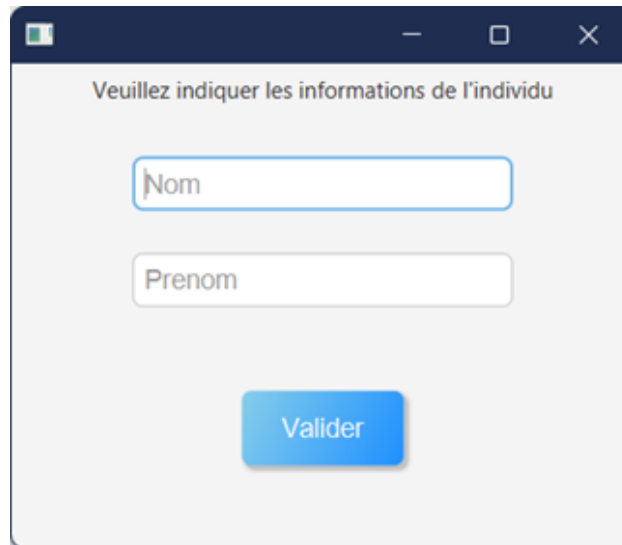
Formulaire d'ajout d'un individu

Similairement, l'utilisateur dispose d'un formulaire lui permettant de supprimer un individu. Pour ce faire, il faut indiquer le nom et prénom de la personne à supprimer.



Formulaire de suppression d'un individu

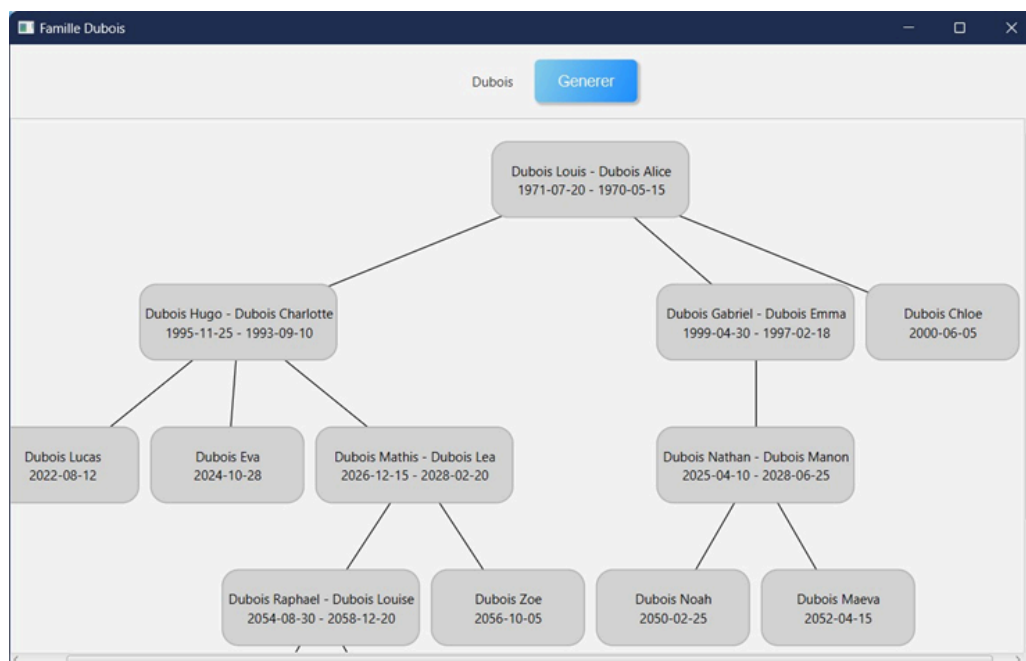
Identiquement, le bouton suivant permet d'afficher dans une fenêtre la descendance d'un individu en indiquant son nom et son prénom.



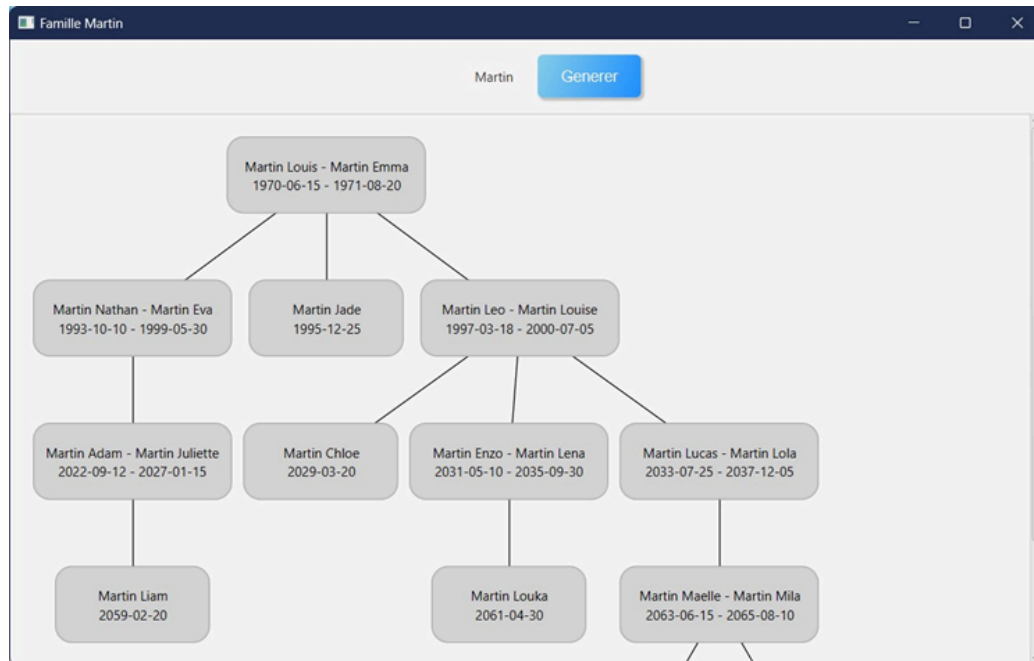
Formulaire de recherche d'un descendant. Le formulaire est intitulé "Veuillez indiquer les informations de l'individu". Il contient deux champs de saisie : "Nom" et "Prenom", suivis d'un bouton "Valider".

Formulaire de recherche d'un descendant

Enfin, les autres boutons sont générés selon le nombre de familles présentes dans le fichier *familles.csv*. Chaque bouton permet d'afficher l'arbre généalogique relatif à une famille.



Affichage de l'arbre généalogique de la famille Dubois



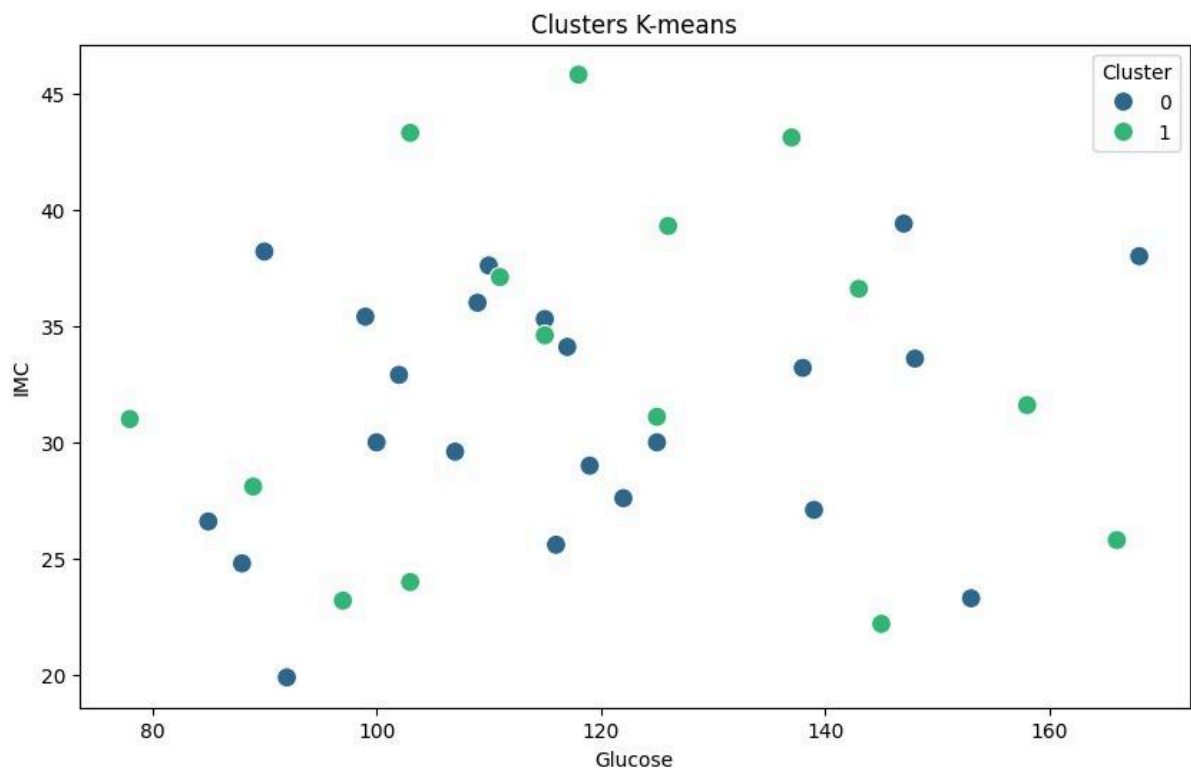
Affichage de l'arbre généalogique de la famille Martin

IV. Intégration des fonctionnalités liées au diabète

A. Enrichissement et analyse de la base de données

Pour notre étude du diabète, nous allons prendre en compte trois paramètres, l'IMC (indice de masse corporelle), le taux d'insuline et le taux de glucose dans le sang. Nous réduisons les paramètres pour simplifier car n'étant pas biologistes, nous ne pouvons faire qu'une modeste étude. Nous utilisons la même base de données à la différence que nous avons ajouté ces nouvelles données dans notre base. Nous avons utilisé du Python, car étant plus approprié pour une étude de ce genre. L'idée de notre programme est de traduire l'algorithme du K-means en Python et de pouvoir représenter un nuage de points qui montre l'appartenance d'un point à un cluster et un graphique qui calcule l'inertie. L'inertie dépend de la distance entre chaque point et le centre de son cluster. Autrement dit. plus les clusters sont démarqués et hétérogènes, plus l'inertie baissera montrant alors l'efficacité des paramètres choisies.

B. Visualisation des prédictions de risque

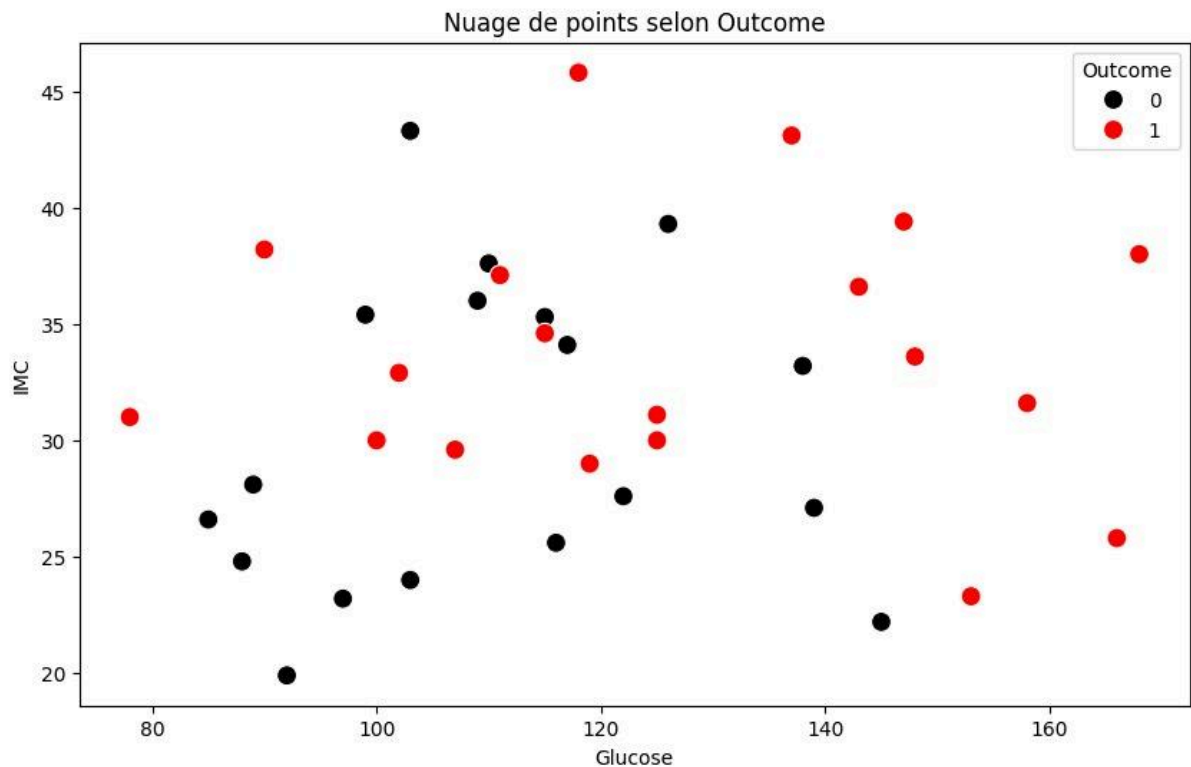


Ce graphique montre une visualisation des clusters obtenus avec l'algorithme du K-means. Les points de données sont projetés sur deux composantes principales (IMC et Glucose), ce qui permet de visualiser les clusters dans un espace bidimensionnel.

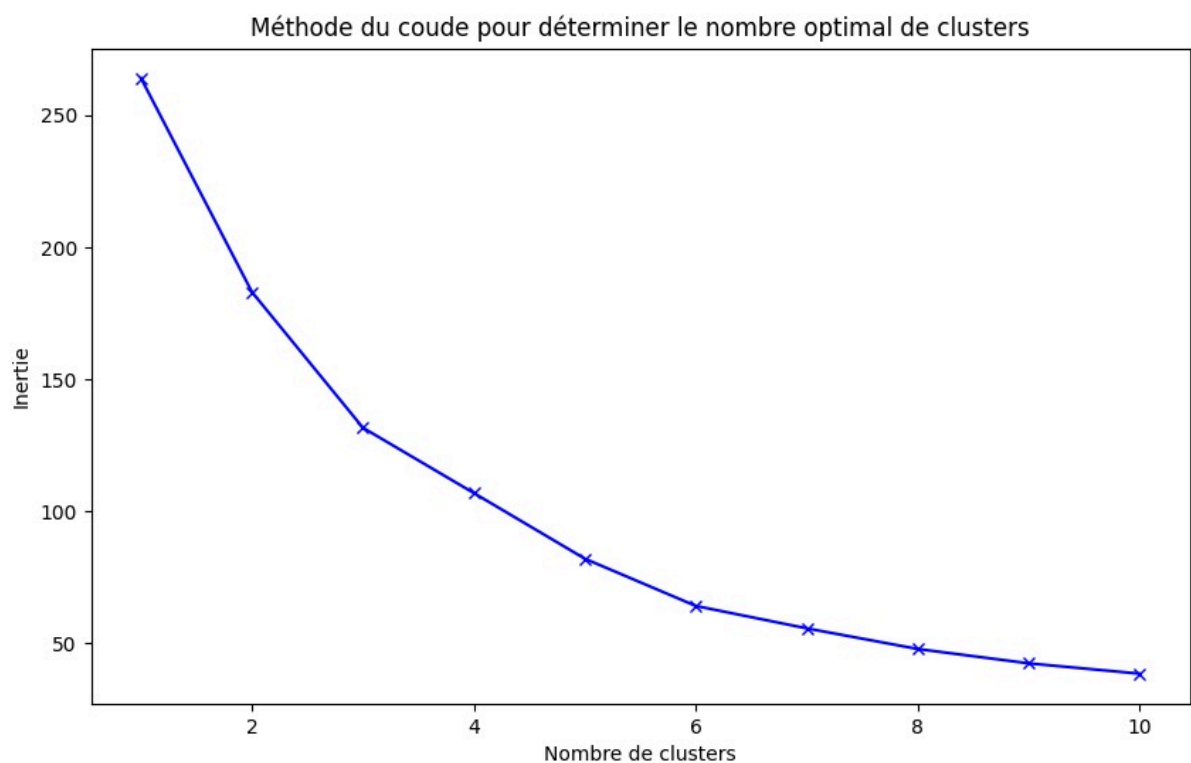
Cluster 0 : Représenté en bleu

Cluster 1 : Représenté en turquoise

Les points de données sont bien séparés en deux clusters distincts, avec des zones de densité bien définies. Ces clusters représentent des groupes de membres partageant des similitudes dans leurs antécédents de diabète et d'autres facteurs de risque. Ainsi, chaque cluster permet d'identifier les membres de la famille ayant des caractéristiques similaires et d'évaluer leur risque de développer un diabète.



Ce graphique est le même que le précédent à la différence que celui-ci montre uniquement qui a le diabète. (0 : la personne n'a pas le diabète, 1 : la personne a le diabète).

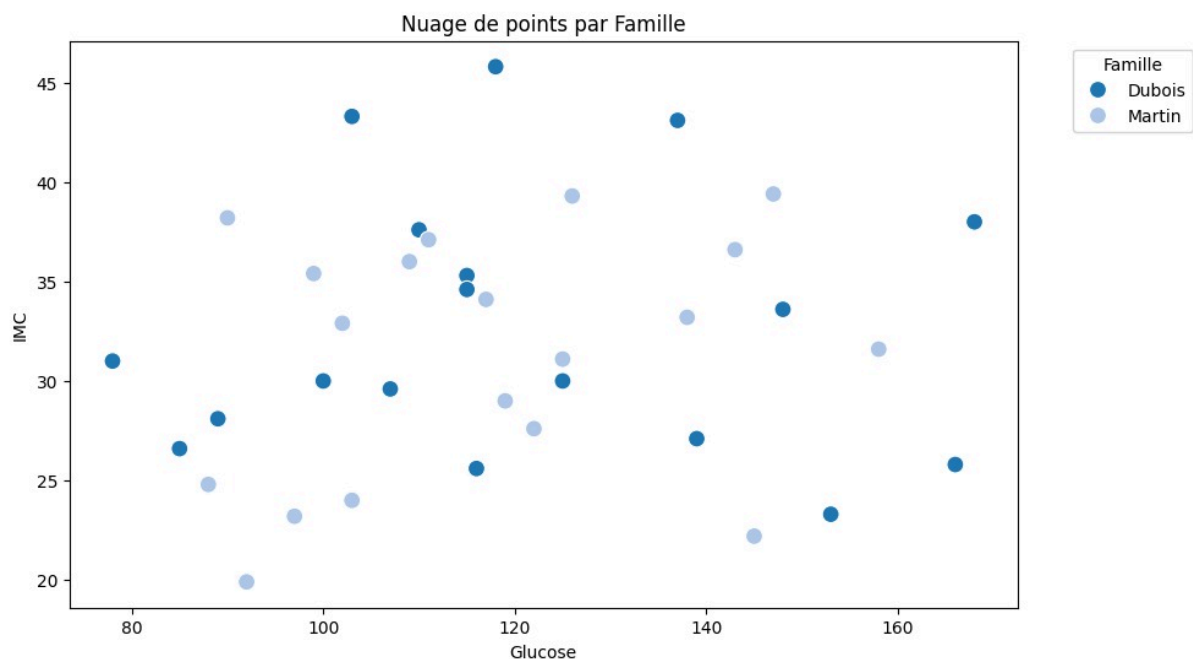


Cette image montre l'inertie totale en fonction du nombre de clusters. La méthode du coude est utilisée pour identifier le nombre optimal de clusters.

Axe des ordonnées (Inertie) : Représente la somme des distances au carré des points de données par rapport à leur centre de cluster.

Axe des abscisses (Nombre de clusters) : Représente le nombre de clusters K

L'inertie diminue à mesure que le nombre de clusters augmente, mais la diminution est plus prononcée au début. Le "coude" dans la courbe se situe généralement à l'endroit où l'ajout de clusters supplémentaires n'entraîne plus de réduction significative de l'inertie. Dans ce cas, le coude semble se situer autour de $K=2$, ce qui suggère que trois clusters pourraient être un bon choix.



Ce graphique montre la répartition des individus en fonction de leur famille.

Interprétation :

Cluster 1 : représente un groupe à risque élevé de diabète , probablement influencé par des mauvais modes de vie . Les membres de ce cluster devraient surveiller leur alimentation afin que le taux de glucose et d'insuline soient bien gérés et l'IMC plus proche de la moyenne.

Cluster 0 : représente un groupe à risque moins élevé de diabète. Pour ce cluster, les personnes ont un meilleur mode de vie, un IMC proche de la moyenne et un taux de glucose et d'insuline dans la norme.

V. Conclusion

En conclusion, ce rapport expose la démarche méthodique adoptée par notre groupe pour aborder la création d'une application permettant de consulter, construire et modifier des arbres généalogiques. À partir de la récolte de références bibliographiques, la rédaction d'un cahier des charges, la construction de la solution conceptuelle, l'exploration des diverses méthodes et applications existantes à ce sujet et l'écriture de l'application en langage Java, chaque étape a été réalisée avec sérieux.

Tout au long de ce projet, nous avons pris conscience de la complexité de la réalisation d'un arbre généalogique, notamment des défis liés à la cohérence des liens entre chaque génération, à l'adaptation de la solution conceptuelle sous forme d'interface et à l'exploitation de la base de données. Notre réunion avec notre enseignante référente a été particulièrement bénéfique, étant donné que ses conseils nous ont permis d'aborder le projet plus clairement.

Références

Family Echo : <https://www.familyecho.com/>

Geneanet : <https://www.geneanet.org/creer-votre-arbre/>

Article Futura décrivant des applications avec des arbres généalogiques :

<https://www.futura-sciences.com/sciences/questions-reponses/logiciels-arbre-genealogique-sont-meilleurs-logiciels-11197>

Définition du diabète :

<https://www.ameli.fr/seine-saint-denis/assure/sante/themes/diabete/diabete-comprendre/definition>

Bibliothèque Abego TreeLayout : <https://treelayout.sourceforge.net/>

Stack Overflow : <https://stackoverflow.com/>

Annexes

Voir le lien GitHub contenant l'ensemble des documents annexes (code Python et Java) :

<https://github.com/RKathirvele/Projet-GMI-Groupe-8>