Quantitative
&
Financial Modelling
(QFM)

Rapport de l'Atélier 1 : Classification

Machine and Deep Learning

TINA Djara Olivier
DJOSSOU Djidjoho Isidore Borel

Professor :
Ali IDRI

Table des matières

1 Introduction

Le diabète est une maladie chronique grave qui se déclare lorsque le pancréas ne produit pas suffisamment d'insuline, ou lorsque l'organisme n'est pas capable d'utiliser efficacement l'insuline qu'il produit.

En effet :

- Le nombre de personnes atteintes de diabète est passé de 108 millions en 1980 à 422 millions en 2014. La prévalence du diabète a augmenté plus rapidement dans les pays à revenu faible ou intermédiaire que dans les pays à revenu élevé.

- Le diabète est une cause importante de cécité, d'insuffisance rénale, d'infarctus du myocarde, d'accidents vasculaires cérébraux et d'amputation des membres inférieurs.

- Entre 2000 et 2019, les taux de mortalité due au diabète selon l'âge ont augmenté de 3 %.

De ces constats, on réalise à quel point le problème est préocupant.

La mise en place d'un algorithme qui permettrait de classifier si une personne est diabétique ou non à partir de certaines caractéristiques peut présenter plusieurs avantages, notamment :

1. Détection précoce du diabète : L'algorithme peut identifier les personnes à risque de développer le diabète en se basant sur des facteurs de risque spécifiques. Cela permet une prise en charge précoce, avant que la maladie ne progresse et ne cause des complications graves.

2. Diagnostic rapide et précis : Les algorithmes de classification peuvent être très précis dans la détection du diabète, ce qui permet un diagnostic rapide et précis. Cela permet également de réduire les erreurs de diagnostic et d'assurer une meilleure prise en charge.

3. Réduction des coûts : La détection précoce et le diagnostic rapide du diabète permettent de réduire les coûts liés aux soins de santé en réduisant les complications de la maladie.

4. Personnalisation des soins : Les algorithmes de classification peuvent aider à personnaliser les soins en identifiant les caractéristiques spécifiques des patients qui nécessitent une attention particulière. Cela permet de fournir des soins de santé plus adaptés aux besoins individuels de chaque patient.

C'est justement ce que nous allons faire dans la suite de ce travail suivant le plan décrit comme suit :

- Materiels et Méthodes : Ici, nous allons décrire le dataset, les critères de performance ainsi que quelques techniques de classifications que nous utiliserons.

- Experimental Design : Cette section sera consacré à la présentation d'un diagramme qui présente notre méthodologie de travail.

- Résultats et discussions : Nous allons présenter, discuter, interpreter et comparer les résultats obtenus à partir de quelques modèles tel que : KNN, Decision Trees, SVM.

- Conclusion et Perspectives

2 Materiels et Méthodes

2.1 Description du dataset

La dataset Pima Indians est une base de données médicales comprenant des données sur des patients indiens Pima. Il contient des informations sur 8 variables numériques pour chacun des 768 patients :

- Nombre de grossesses : Nombre de fois où une femme a été enceinte
- Glucose : Taux de glucose plasmatique à 2 heures dans un test de tolérance au glucose
- Pression artérielle : pression artérielle diastolique (mm Hg)
- Épaisseur du pli cutané : épaisseur du pli cutané du triceps (mm)
- Insuline : niveau d'insuline sérique à 2 heures (mu U/ml)
- Indice de masse corporelle : indice de masse corporelle (poids en kg/(taille en m)^2)
- Diabète de parent proche de fonction : Histoire familiale de diabète
- Âge : âge en années

Dans ce document on cherche à analyser le dataset PIMA Indian Diabetes afin de prédire si selon certains critères un patient est à même de devenir diabétique ou non, représenté par un 1 ou un 0 respectivement.

Il faut rester prudent dans l'interprétation comme ces données sont basées sur un groupe de population, on ne peut avec certitude l'appliquer sur d'autres populations que les PIMA. Les Pima sont un groupe de population de natifs américains qui vivent dans la zone actuelle du Sud de l'Arizona.

2.2 Critères de perfomance

On commence par définir la matrice de confusion. Par exemple, dans le cadre d'une classification binaire pour detecter si un mail est spam ou pas on a :

Confusion matrix

	Predicted: Spam Email	Predicted: Real Email
Actual: Spam Email	True Positive	False Negative
Actual: Real Email	False Positive	True Negative

2.2.1 Accuracy

L'accuracy mesure la proportion de prédictions correctes parmi toutes les prédictions effectuées. Elle est calculée en divisant le nombre de prédictions correctes par le nombre total de prédictions. L'accuracy est une mesure utile pour évaluer la capacité de l'algorithme à classifier correctement les données.

Pour utiliser l'accuracy pour évaluer la performance de l'algorithme, on peut diviser les données en un ensemble d'entraînement et un ensemble de test. On entraîne l'algorithme sur l'ensemble d'entraînement et on évalue sa performance sur l'ensemble de test. L'accuracy de l'algorithme est alors calculée en comparant les prédictions effectuées par l'algorithme avec les étiquettes réelles des données de test.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2.2.2 Precision

la precision mesure la proportion de prédictions correctes parmi les prédictions positives. Autrement dit, la precision mesure la capacité d'un modèle à identifier correctement les vrais positifs par rapport aux faux positifs.

$$precision = \frac{TP}{TP + FP}$$

2.2.3 Sensitivity

Le rappel mesure la proportion de vrais positifs qui ont été correctement identifiés par rapport à tous les vrais positifs qui existent réellement.

Pour utiliser la sensibilité pour évaluer la performance de l'algorithme, on peut utiliser une matrice de confusion. La matrice de confusion est une table qui montre le nombre de vrais positifs, de faux positifs, de vrais négatifs et de

$$recall = \frac{TP}{TP + FN}$$

faux négatifs. La sensibilité est alors calculée en divisant le nombre de vrais positifs par la somme des vrais positifs et des faux négatifs.

$$recall = \frac{TP}{TP + FN}$$

2.2.4 Spécificité (Specificity)

La spécificité mesure la proportion de vrais négatifs (observations négatives correctement identifiées) parmi toutes les observations négatives. Elle est calculée en divisant le nombre de vrais négatifs par le nombre total d'observations négatives. La spécificité est une mesure utile pour évaluer la capacité de l'algorithme à identifier les cas négatifs.

Pour utiliser la spécificité pour évaluer la performance de l'algorithme, on peut également utiliser une matrice de confusion. La spécificité est alors calculée en divisant le nombre de vrais négatifs par la somme des vrais négatifs et des faux positifs.

$$specificity = \frac{TN}{TN + FP}$$

2.2.5 F1-score

Le F1-score est un autre critère de performance important pour évaluer un algorithme de classification. Le F1-score est la moyenne harmonique de la précision et de la sensibilité. Il est particulièrement utile lorsque les classes ne sont pas équilibrées, c'est-à-dire lorsque le nombre d'exemples positifs et négatifs n'est pas le même.

Le F1-score est la moyenne harmonique de la précision et de la sensibilité, calculée selon la formule suivante :

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall}$$

Cette formule permet de donner plus de poids à la valeur la plus faible entre la précision et la sensibilité. Ainsi, si la précision est faible mais la sensibilité est élevée, le F1-score sera également faible.

2.3 Techniques de preprocessing

2.3.1 Normalisation des données

La normalisation est une technique de preprocessing en machine learning qui vise à mettre les données à la même échelle, afin de faciliter leur traitement et leur comparaison. Cette technique est souvent utilisée dans les modèles de machine learning qui utilisent des distances ou des mesures de similarité pour comparer les observations.

Il existe plusieurs méthodes de normalisation, mais l'une des plus courantes est la normalisation Min-Max, également appelée mise à l'échelle. Cette méthode consiste à transformer les données en une échelle comprise entre 0 et 1, en utilisant la formule suivante :

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

où X est la valeur d'une observation, X_{min} est la valeur minimale de la variable et X_{max} est la valeur maximale de la variable.

La normalisation $Min - Max$ peut être utilisée pour les variables continues, les variables discrètes ou les variables binaires. Cette méthode présente plusieurs avantages :

- Elle ne modifie pas la distribution des données.
- Elle ne modifie pas l'ordre des données.
- Elle est facile à mettre en œuvre.

Toutefois, la normalisation Min-Max peut présenter quelques inconvénients :

- Elle peut être sensible aux valeurs aberrantes (outliers), car elle utilise les valeurs minimales et maximales de la variable.
- Elle peut ne pas être appropriée pour les variables qui suivent une distribution asymétrique ou pour les variables avec des valeurs manquantes.

2.3.2 Imputation des données manquantes

L'imputation des données manquantes est une technique de preprocessing en machine learning qui vise à remplacer les valeurs manquantes dans un ensemble de données par des valeurs estimées. Cette technique est couramment utilisée dans les modèles de machine learning, car les données manquantes peuvent avoir un impact significatif sur les résultats de modélisation.

Il existe plusieurs méthodes d'imputation des données manquantes, telles que :

Imputation par la moyenne : cette méthode consiste à remplacer les valeurs manquantes d'une variable par la moyenne des valeurs de cette variable. Cette méthode est simple et rapide, mais elle peut ne pas être appropriée pour les variables avec une distribution asymétrique ou pour les variables avec des valeurs aberrantes (outliers).

Imputation par la médiane : cette méthode consiste à remplacer les valeurs manquantes d'une variable par la médiane des valeurs de cette variable. Cette méthode est plus robuste que l'imputation par la moyenne aux valeurs aberrantes, mais elle peut ne pas être appropriée pour les variables avec une distribution asymétrique.

Imputation par le mode : cette méthode consiste à remplacer les valeurs manquantes d'une variable par le mode (la valeur la plus fréquente) des valeurs de cette variable. Cette méthode est appropriée pour les variables catégorielles ou binaires, mais elle peut ne pas être appropriée pour les variables continues.

Imputation par les valeurs les plus proches : cette méthode consiste à remplacer les valeurs manquantes d'une variable par les valeurs les plus proches dans l'ensemble de données. Cette méthode peut être utilisée pour les variables continues et catégorielles, mais elle peut être sensible à la dimensionnalité de l'ensemble de données.

Imputation par les modèles de machine learning : cette méthode consiste à utiliser des modèles de machine learning pour imputer les valeurs manquantes. Cette méthode peut fournir des estimations plus précises et plus fiables que les méthodes d'imputation simples, mais elle est plus complexe et plus coûteuse en termes de temps de calcul.

Il est important de noter que l'imputation des données manquantes peut avoir un impact significatif sur les résultats de modélisation. Une imputation inappropriée peut introduire un biais dans les données et entraîner des erreurs de modélisation. Par conséquent, il est recommandé d'utiliser une méthode appropriée pour imputer les données manquantes, en fonction de la nature des données et du modèle de machine learning utilisé. Il est également recommandé de tester plusieurs méthodes d'imputation pour trouver celle qui convient le mieux aux données et au modèle de machine learning.

2.3.3 Sélection de caractéristiques (features selection)

La sélection de caractéristiques (ou features selection) est une technique de preprocessing de données en machine learning qui vise à réduire la dimensionna-

lité d'un ensemble de données en sélectionnant un sous-ensemble de caractéristiques (ou variables) pertinentes et discriminantes pour la tâche de classification ou de régression. L'objectif de la sélection de caractéristiques est d'améliorer la performance de modélisation, d'accélérer le temps de formation du modèle et de faciliter l'interprétation des résultats.

Les filtres

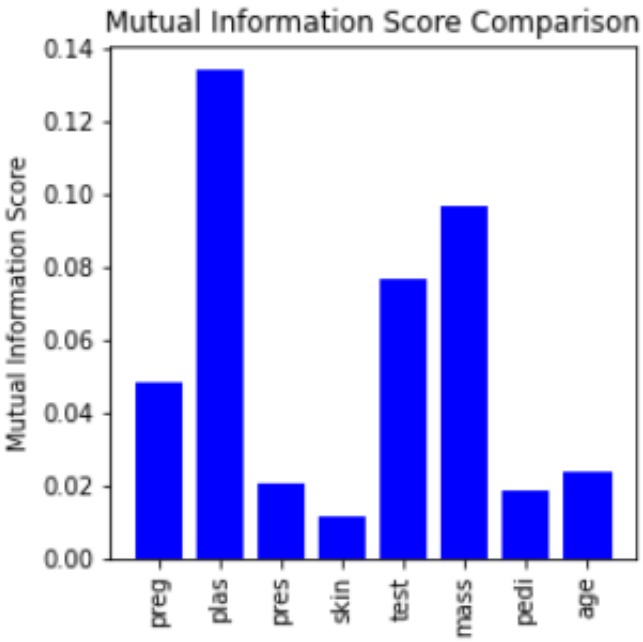
Les méthodes de filtrage prennent en compte les propriétés intrinsèques des caractéristiques mesurées par des statistiques univariées ou multivariées. Ces méthodes sont plus rapides et moins coûteuses en termes de calcul que les méthodes d'encapsulation (wrappers). Lorsque l'on traite des données à haute dimension, il est plus économique d'utiliser des méthodes de filtrage.

Les méthodes de filtrage sont souvent utilisées en premier lieu car elles sont simples, rapides et efficaces pour éliminer les caractéristiques non pertinentes. Voici quelques-unes des méthodes de filtrage les plus couramment utilisées :

- Variance theshold : Le seuil de variance est une approche de base simple pour la sélection des caractéristiques. Il élimine toutes les caractéristiques dont la variance n'atteint pas un certain seuil. Par défaut, il élimine toutes les caractéristiques à variance nulle, c'est-à-dire les caractéristiques qui ont la même valeur dans tous les échantillons. Nous supposons que les caractéristiques ayant une variance plus élevée peuvent contenir plus d'informations utiles.

- Chi-square Test : Utilisée pour les caractéristiques catégorielles d'un ensemble de données. Nous calculons le Chi- square entre chaque caractéristique et la cible et sélectionnons le nombre souhaité de caractéristiques ayant les meilleurs scores. Pour appliquer correctement le chi-deux afin de tester la relation entre diverses caractéristiques de l'ensemble de données et la variable cible, les conditions suivantes doivent être remplies : les variables doivent être catégoriques, échantillonnées indépendamment et les valeurs doivent avoir une fréquence attendue supérieure à 5.

- Information Gain : Consiste à calculer la réduction de l'entropie résultant de la transformation d'un ensemble de données. Il peut être utilisé pour la sélection des caractéristiques en évaluant le gain d'information de chaque variable dans le contexte de la variable cible.



Les wrappers

Les wrappers nécessitent une méthode d'encapsulation pour rechercher l'espace de tous les sous- ensembles possibles de caractéristiques, en évaluant leur qualité par l'apprentissage et l'évaluation d'un classificateur avec ce sous-ensemble de caractéristiques. Le processus de sélection des caractéristiques est basé sur un algorithme d'apprentissage automatique spécifique que nous essayons d'adapter à un ensemble de données donné. Il suit une approche de recherche gloutonne en évaluant toutes les combinaisons possibles de caractéristiques par rapport au critère d'évaluation. Les méthodes wrapper donnent généralement une meilleure précision prédictive que les méthodes de filtrage.

- Forward feature selection : Il s'agit d'une méthode itérative dans laquelle nous commençons par la variable la plus performante par rapport à l'objectif. Ensuite, nous sélectionnons une autre variable qui donne la meilleure performance en combinaison avec la première variable sélectionnée. Ce processus se poursuit jusqu'à ce que le critère prédéfini soit atteint.

- Exhaustive feature selection : Il s'agit de la méthode de sélection des caractéristiques la plus robuste couverte jusqu'à présent. Il s'agit d'une évaluation par force brute de chaque sous-ensemble de caractéristiques. Cela signifie qu'elle essaie toutes les combinaisons possibles de variables et renvoie le sous-ensemble le plus performant.

2.3.4 Équilibrage des classes

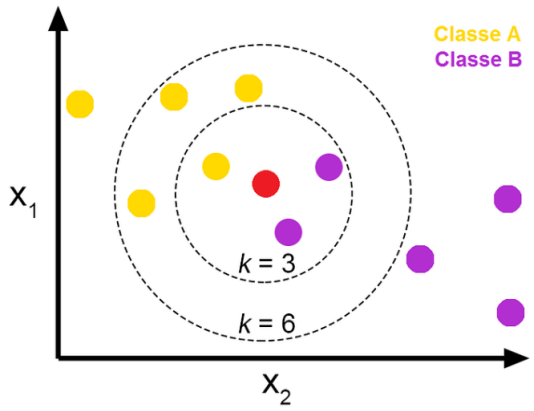
Under Sampling ou sous-échantillonnnage
Cette méthode consiste à supprimer des échantillons de la classe majoritaire pour obtenir un ensemble de données équilibré. Cette méthode peut être risquée car elle peut entraîner une perte d'informations importantes dans le jeu de données.

Over Sampling ou sur-échantillonnnage
Cette méthode consiste à dupliquer des échantillons de la classe minoritaire pour obtenir un ensemble de données équilibré. Il y a plusieurs manières de sur-échantillonner les données, comme la méthode de réplication, la méthode SMOTE (Synthetic Minority Over-sampling Technique), etc.

2.4 Techniques de classification

2.4.1 Les k-plus proches voisins (k-NN)

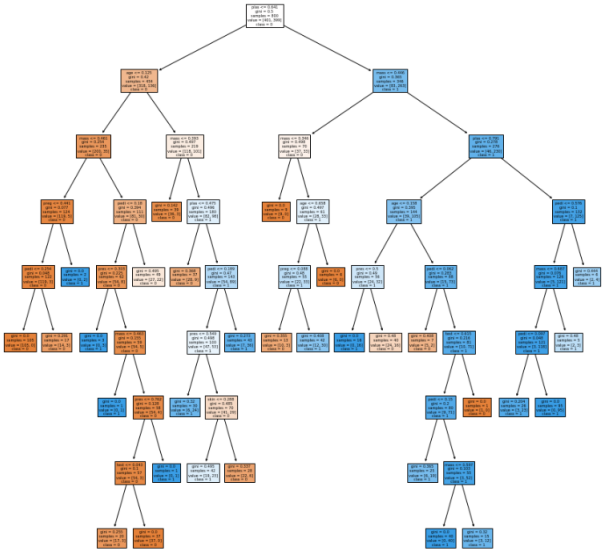
Le k-NN est une méthode de classification simple et populaire basée sur la similarité entre les échantillons. Pour classer un nouvel échantillon, l'algorithme calcule d'abord la distance entre cet échantillon et tous les autres échantillons du jeu de données d'apprentissage. Ensuite, les k échantillons les plus proches (les "k-plus proches voisins") sont sélectionnés, et la classe majoritaire parmi ces k échantillons est attribuée au nouvel échantillon. La valeur de k est un paramètre de l'algorithme et doit être spécifiée par l'utilisateur.



2.4.2 Les arbres de decision (Decision trees)

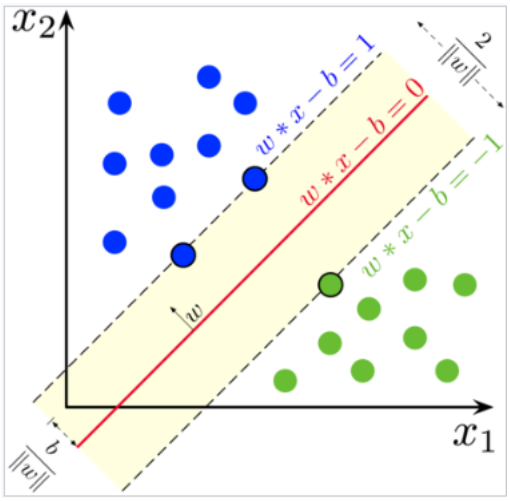
Les arbres de décision sont des modèles de classification qui permettent de représenter graphiquement toutes les combinaisons de caractéristiques possibles pour un ensemble de données. L'algorithme divise progressivement l'ensemble de données en sous-ensembles plus petits, jusqu'à ce que chaque sous-ensemble

ne contienne que des échantillons d'une seule classe. Pour chaque division, l'algorithme sélectionne la caractéristique qui maximise la réduction d'entropie, c'est-à-dire qui divise le mieux les données en classes distinctes. Les arbres de décision sont populaires car ils sont faciles à interpréter et peuvent gérer des données de différents types (numériques, catégorielles, binaires, etc.).



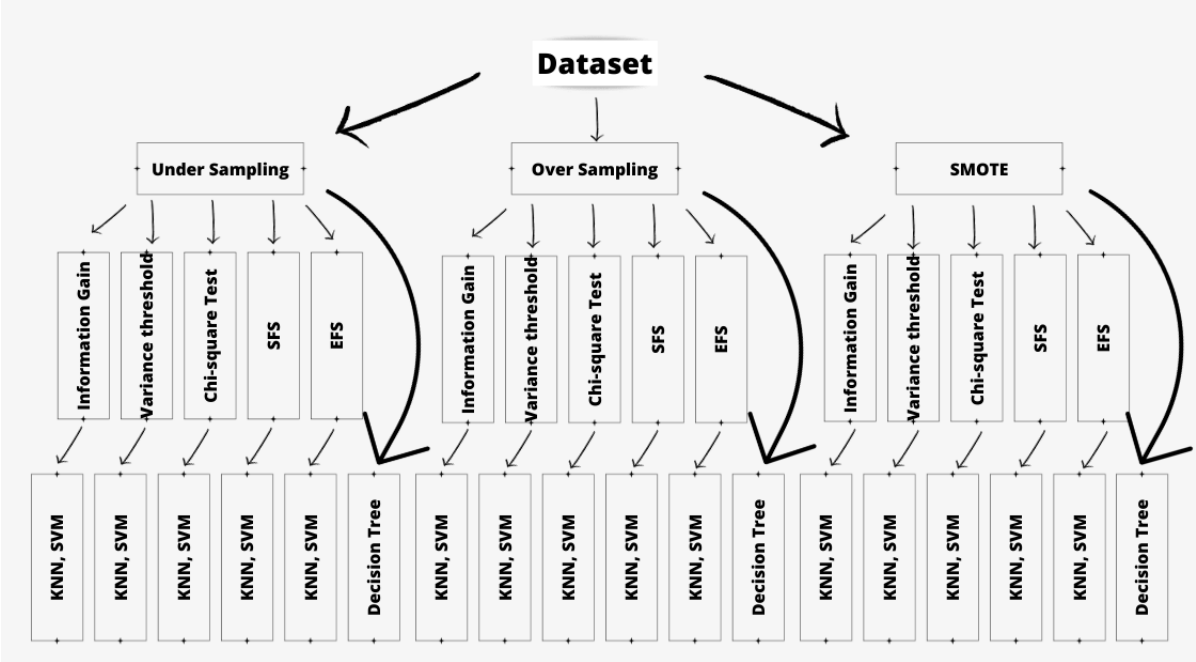
2.4.3 Les machines à vecteurs de support (SVM)

Les SVM sont des algorithmes de classification qui cherchent à trouver la meilleure séparation entre les différentes classes de données en créant une frontière de décision qui maximise la marge entre les deux classes les plus proches. La marge est la distance entre la frontière de décision et les échantillons les plus proches de chaque classe. Les SVM sont populaires car ils peuvent gérer des données de grande dimension et sont efficaces pour les ensembles de données à deux classes ou plus. Les SVM peuvent également utiliser des "fonctions noyau" pour projeter les données dans des espaces de grande dimension, ce qui peut améliorer la précision de la classification pour les ensembles de données non linéaires.



3 Experimental Design

Le diagramme ci-dessous représente le plan de notre experimentation.



4 Résultats et discussion

4.1 Les k-plus proches voisins (k-NN)

4.1.1 Random Under Sampling

Suite à un sous-échantillonnage comme décrit dans la sous section 2.3.4 nous avons obtenu un dataset avec 536 lignes sur lequel nous avons appliquées les filtres et wrappers présentés dans la sous section 2.3.3. Donc, nous avons finalement 5 datasets à partir du sous-echantillonnage. Le tableau ci-dessous contient les resultats obtenu par application du KNN sur ces différents jeux de données.

Criteres / Datasets	Classe	Precision	Recall	F1-score	Support	Accuracy	k vakues
X_new_MIC_1	0	0.8	0.77	0.78	99	0.79	4
	1	0.78	0.81	0.8	101		
X_new_Cor_1	0	0.81	0.73	0.77	59	0.759	8
	1	0.71	0.8	0.75	49		
X_new_SKB_1	0	0.81	0.73	0.77	59	0.759	8
	1	0.71	0.8	0.75	49		
X_new_SFS_1	0	0.91	0.69	0.79	59	0.796	6
	1	0.71	0.92	0.8	49		
X_new_EFS_1	0	0.81	0.85	0.83	59	0.805	34
	1	0.8	0.76	0.78	49		

4.1.2 Random Over Sampling

Suite à un sur-échantillonnage comme décrit dans la sous section 2.3.4 nous avons obtenu un dataset avec 1000 lignes sur lequel nous avons appliquées les filtres et wrappers présentés dans la sous section 2.3.3. Donc, nous avons finalement 5 datasets à partir du sur-echantillonnage. Le tableau ci-dessous contient les resultats obtenu par application du KNN sur ces différents jeux de données.

Critères / Datasets	Classe	Precision	Recall	F1-score	Support	Accuracy	k vakues
X_new_MIC_2	0	0.74	0.74	0.74	99	0.85	47
	1	0.75	0.75	0.75	101		
X_new_Cor_2	0	0.76	0.74	0.75	99	0.848	32
	1	0.75	0.77	0.76	101		
X_new_SKB_2	0	0.76	0.74	0.75	99	0.848	32
	1	0.75	0.77	0.76	101		
X_new_SFS_2	0	0.73	0.73	0.73	99	0.85	41
	1	0.73	0.73	0.73	101		
X_new_EFS_2	0	0.73	0.77	0.75	99	0.84	32
	1	0.76	0.72	0.74	101		

4.1.3 SMOTE

Ici, une méthode particulière est utilisée pour le sur-echantillonnage, il s'agit de SMOTE. En utilisant le même raisonnement que pour le sous-echantillonnage, on a 5 datasets. Le tableau ci-dessous contient les resultats obtenu par application du KNN sur ces différents jeux de données.

Critères / Datasets	Classe	Precision	Recall	F1-score	Support	Accuracy	k vakues
X_new_MIC_3	0	0.76	0.7	0.73	149	0.799	29
	1	0.72	0.78	0.75	151		
X_new_Cor_3	0	0.85	0.76	0.8	99	0.821	5
	1	0.79	0.87	0.83	101		
X_new_SKB_3	0	0.85	0.78	0.81	99	0.82	3
	1	0.8	0.86	0.83	101		
X_new_SFS_3	0	0.77	0.81	0.79	99	0.821	4
	1	0.8	0.76	0.78	101		
X_new_EFS_3	0	0.8	0.83	0.82	99	0.815	4
	1	0.83	0.8	0.81	101		

4.1.4 Comparaison k-NN

En comparant les performances du modèle KNN pour les 15 datasets des tableaux ci-dessus, on en deduit que l'option optimale est d'appliquer le Random Over Sampling sur le dataset initial puis utiliser le wrapper SFS. Le tableau suivant montre les paramètres optimaux.

```
0.8560000000000001
{'metric': 'euclidean', 'n_neighbors': 49, 'weights': 'distance'}
KNeighborsClassifier(metric='euclidean', n_neighbors=49, weights='distance')
Classification report
              precision    recall  f1-score   support

           0       0.71      0.72      0.71        99
           1       0.72      0.71      0.72       101

    accuracy                           0.71       200
   macro avg       0.71      0.72      0.71       200
weighted avg       0.72      0.71      0.72       200
```

4.2 Les arbres de decision (Decision trees)

Les arbres de decision sont des modèles qui gèrent eux même la selection de variables. Ce qui fait que nous les donnons directement notre dataset àpres nettoyage. Nous avons ainsi 3 datasets correspondants respectivement aux resultat de Random Under Sampling, Random Over Sampling et SMOTE.

Notons que dans notre tableau ci-dessous, les abréviations suivantes : m_d, m_s_l et m_s_s sont nommées respectivements comme suit : max_depth, min_samples_leaf et min_samples_split

Critères / Datasets	Classe	Precision	Recall	F1-score	Support	Accuracy	Criterion	m_d	m_s_l	m_s_s
X1	0	0.76	0.76	0.76	99	0.76	entropy	4	2	10
	1	0.76	0.76	0.76	101					
X2	0	0.94	0.95	0.94	99	0.945	entropy	10	1	2
	1	0.95	0.94	0.95	101					
X3	0	0.93	0.81	0.86	149	0.873	gini	6	4	10
	1	0.83	0.94	0.88	151					

4.2.1 Comparaison Decision Tree

En comparant les performances du modèle Decision Tree pour les 3 datasets du tableau ci-dessus, on en deduit que l'option optimale est d'appliquer le Random Over Sampling sur le dataset initial. Le tableau suivant montre les paramètres optimaux.

```
{'criterion': 'entropy', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2}
DecisionTreeClassifier(criterion='entropy', max_depth=10, random_state=12)
Acc: 0.945
Classification report
              precision    recall  f1-score   support

           0       0.94      0.95      0.94        99
           1       0.95      0.94      0.95       101

    accuracy                           0.94       200
   macro avg       0.94      0.95      0.94       200
weighted avg       0.95      0.94      0.95       200
```

4.3 Les machines à vecteurs de support (SVM)

On suit la même logique que nous avons suivi pour le KNN.

Dans les tableaux ci-dessous, la valeur "Default" représente la valeur par défaut du paramètre concerné.

4.3.1 Random Under Sampling

Le tableau suivant contient les résultats obtenus suite à l'application du SVM sur les datasets issus de Random Under Sampling.

Critères / Datasets	Classe	Precision	Recall	F1-score	Support	Accuracy	C	Kernel
X_new_MIC_1	0	0.81	0.78	0.79	59	0.77	Default	Default
	1	0.75	0.78	0.76	49			
X_new_Cor_1	0	0.83	0.73	0.77	59	0.768	Default	Default
	1	0.71	0.82	0.76	49			
X_new_SKB_1	0	0.83	0.73	0.77	59	0.7685	Default	Default
	1	0.71	0.82	0.76	49			
X_new_SFS_1	0	0.8	0.76	0.78	59	0.768	Default	Default
	1	0.73	0.78	0.75	49			
X_new_EFS_1	0	0.83	0.75	0.79	59	0.77	Default	Default
	1	0.73	0.82	0.77	49			

4.3.2 Random Over Sampling

Le tableau suivant contient les résultats obtenus suite à l'application du SVM sur les datasets issus de Random Over Sampling.

Critères / Datasets	Classe	Precision	Recall	F1-score	Support	Accuracy	C	Kernel
X_new_MIC_2	0	0.71	0.76	0.74	99	0.7625	1	rbf
	1	0.75	0.7	0.72	101			
X_new_Cor_2	0	0.77	0.72	0.74	99	0.755	Default	Default
	1	0.74	0.79	0.77	101			
X_new_SKB_2	0	0.77	0.72	0.74	99	0.755	Default	Default
	1	0.74	0.79	0.77	101			
X_new_SFS_2	0	0.72	0.79	0.75	99	0.763	2	poly
	1	0.77	0.69	0.73	101			
X_new_EFS_2	0	0.73	0.74	0.73	99	0.761	2	rbf
	1	0.74	0.73	0.74	101			

4.3.3 SMOTE

Le tableau suivant contient les résultats obtenus suite à l'application du SVM sur les datasets issus de SMOTE.

Critères / Datasets	Classe	Precision	Recall	F1-score	Support	Accuracy	C	Kernel
X_new_MIC_3	0	0.78	0.77	0.78	99	0.78	Default	Default
	1	0.78	0.79	0.78	101			
X_new_Cor_3	0	0.82	0.76	0.8	99	0.8	Default	Default
	1	0.78	0.84	0.79	101			
X_new_SKB_3	0	0.81	0.74	0.77	99	0.785	Default	Default
	1	0.76	0.83	0.8	101			
X_new_SFS_3	0	0.82	0.75	0.78	99	0.795	Default	Default
	1	0.77	0.84	0.81	101			
X_new_EFS_3	0	0.81	0.72	0.76	99	0.775	Default	Default
	1	0.75	0.83	0.79	101			

4.3.4 Comparaison svm

En comparant les performances du modèle KNN pour les 15 datasets des tableaux ci-dessus, on en deduit que l'option optimale est d'appliquer le SMOTE sur le dataset initial puis utiliser le filtre correlation coefficient. Le tableau suivant montre les paramètres optimaux.

```
Accuracy: 0.8
Classification report
              precision    recall  f1-score   support

           0       0.82      0.76      0.79        99
           1       0.78      0.84      0.81       101

    accuracy                           0.80       200
   macro avg       0.80      0.80      0.80       200
weighted avg       0.80      0.80      0.80       200
```

4.4 Comparaison entre k-NN, Decision Tree et SVM

En analysant les indicateurs de perfomances de nos 3 meilleurs modèles (kNN, DT et SVM), on realise que le Decision Tree est le meilleur classificateur pour notre cas.

Meilleur Classificateur

```
{'criterion': 'entropy', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2}
DecisionTreeClassifier(criterion='entropy', max_depth=10, random_state=12)
Acc: 0.945
Classification report
              precision    recall  f1-score   support

           0       0.94      0.95      0.94        99
           1       0.95      0.94      0.95       101

    accuracy                           0.94       200
   macro avg       0.94      0.95      0.94       200
weighted avg       0.95      0.94      0.95       200
```

Avec ce modèle, nous pouvons determiner l'état diabétique ou pas d'une personne de la tribu des PIMA dès lors qu'on connais les valeurs des variables explicatives utilisées dans notre dataset.

5 Conclusion et Perspectives

Après avoir utilisé les algorithmes de classification KNN, SVM et arbre de décision pour classer les données du dataset Pima Indian Diabetes, nous proposons quelques éléments de bilan :

- Les trois algorithmes ont réussi à classifier les données avec une précision raisonnable, mais leur performance varie en fonction des critères de performance considérés (par exemple, l'exactitude, la sensibilité, la spécificité, etc.).

- Après l'application des algorithmes de classification, on a constaté que l'accuracy de Decision Tree est plus élevé que celui des deux autres modèles. De plus, l'arbre de décision a tendance à être plus facile à interpréter que les autres algorithmes, car il peut produire des règles de décision simples et facilement compréhensibles.

- En général, le KNN et le SVM ont tendance à être plus précis que l'arbre de décision, mais ils peuvent être plus difficiles à interpréter.

- Nous avons constaté que la performance des trois algorithmes varient en fonction de la taille de l'ensemble de données, de la qualité des données et des paramètres choisis.

Ce travail peut avoir comme perspectives :

- L'exploration d'autres algorithmes de classification pour comparer leur performance avec les algorithmes existants. Par exemple, le réseau de neurones, le gradient boosting, le random forest, etc.

- Une étude de l'impact des facteurs sociodémographiques sur la prévalence du diabète chez les femmes Pima indiennes, en utilisant des techniques d'analyse de données supplémentaires telles que l'analyse de régression et l'analyse de survie.

Références

[1] Pr. Ali Idri et Fatima Zahrae Nakach, Ph.D. student, UM6P *Atelier 1 : Classification*

[2] Medium https ://towardsdatascience.com/knn-k-nearest-neighbors-1-a4707b24bd1d *KNN (K-Nearest Neighbors) #1*

[3] OMS https ://www.who.int/fr/news-room/fact-sheets/detail/diabetes *Diabète*

[4] OpenClassroom https ://openclassrooms.com/fr/courses/4297211-evaluez-les-performances-dun-modele-de-machine-learning *Evaluez les performances d'un modele de machine learning*

[5] Kaggle https ://www.kaggle.com/code/laurakonig/pima-indians-diabetes-database/notebook *Pima Indians Diabetes Database*

[6] Wikipedia https ://en.wikipedia.org/wiki/Support$_v ector_m achine$*Support vector machine*