République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari Boumediene

Faculté d'Informatique

Département Informatique

Spécialité : Bio-Informatique

Rapport Module Fouille de Données

Thème

Classification supervisé

Réalisé par :

LAIB Ayoub

Table de matière

1. Introduction 1
   1.1. Contexte du projet 1
   1.2. Objectifs 1
2. Présentation du benchmark utilisé 2
   2.1. Description du dataset *Pima Indians Diabetes* 2
   2.2. Structure des données 2
   2.3. Analyse exploratoire initiale 2
3. Méthodologie 3
   3.1. Prétraitement des données 3
   3.2. Division des données (80% apprentissage, 20% test) 3
4. Définition des métriques de performance 4
   4.1. Calcul des métriques de base : TP, TN, FP, FN 4
   4.2. Évaluation des métriques : Précision, Rappel et F-mesure 4
5. Description des algorithmes implémentés 5
   5.1. Algorithme KNN 5
      5.1.1. Chargement et préparation des données 5
      5.1.2. Performance selon la valeur de K 6
      5.1.3. Meilleure valeur de K 8
      5.1.4. Observations générales 8
   5.2. Algorithme Naïve Bayes 9
      5.2.1. Probabilités a priori des classes 9
      5.2.2. Statistiques par classe 9
      5.2.3. Résultats de la classification 11
      5.2.4. Métriques de performance 11
   5.3. L'Arbre de Décision 13
      5.3.1. Métriques de division 13
      5.3.2. Résultats pour le Gain Ratio 13
      5.3.3. Résultats pour l'Indice de Gini 14
      5.3.4. Interprétation globale 14
   5.4. Le Réseau de Neurones 15
      5.4.1. Configuration du réseau 15
      5.4.2. Processus d'entraînement 15
      5.4.3. Résultats globaux 16
      5.4.4. Analyse des performances 17
   5.5. Support Vector Machine 18
      5.5.1. Configuration du modèle SVM 18
      5.5.2. Résultats avec le Kernel RBF 18
      5.5.3. Résultats avec le Kernel Linéaire 20

5.5.4. Résultats avec le Kernel Polynomial 21
5.5.5. Analyse des performances par Kernel 23
5.6. La Régression Linéaire 24
5.6.1. Configuration du modèle 24
5.6.2. Poids des caractéristiques (Features) 24
5.6.3. Perte finale 26
5.6.4. Résultats de la classification 26
5.6.5. Points forts et limitations 28
5.7. L'algorithme Apriori 29
5.7.1. Règles d'Association 29
5.7.2. Facteurs de risque identifiés 32
6. Tableau Comparatif des Performances des Algorithmes 34
7. Classification et Prédiction des Nouvelles Instances 35
7.1. Scénarios de Prédiction 35
7.2. Analyse des Prédictions 37
8. Avantages et Inconvénients des Algorithmes 38
9. Conclusion 39
9.1. Résumé 39
9.2. Propositions d'amélioration et travaux futurs 40

1. Introduction

1.1.Contexte du projet

Dans un monde où les données sont omniprésentes, leur exploitation efficace est devenue essentielle pour résoudre divers problèmes. La fouille de données, ou *data mining*, permet d'extraire des informations utiles à partir de grandes quantités de données brutes. Ce projet s'inscrit dans cette perspective en se concentrant sur l'analyse et la classification des données médicales, plus précisément le dataset *Pima Indians Diabetes*.

Ce jeu de données, largement utilisé dans la recherche, vise à prédire la présence ou l'absence de diabète chez des patientes d'origine Pima. À travers ce projet, nous explorons différentes techniques de classification et d'apprentissage automatique afin d'évaluer leur efficacité et de mieux comprendre les enjeux liés à la fouille de données.

1.2.Objectifs

L'objectif principal de ce projet est de mettre en œuvre plusieurs algorithmes d'apprentissage automatique pour résoudre une tâche de classification binaire sur le dataset *Pima Indians Diabetes*. Plus précisément, nous visons à :

- Préparer et prétraiter les données pour assurer une bonne qualité d'analyse.
- Appliquer et comparer différentes méthodes de classification, notamment KNN, Naive Bayes, arbres de décision, réseaux de neurones, SVM, régression linéaire et l'algorithme Apriori.
- Évaluer les performances de ces modèles à l'aide de métriques comme la précision, le rappel et la F-mesure.
- Identifier les forces et faiblesses de chaque méthode pour choisir la plus adaptée à ce type de données.

Ce travail permet non seulement de renforcer notre compréhension des techniques de fouille de données, mais aussi d'appliquer ces connaissances à un domaine d'une grande importance : la santé.

2. Présentation du benchmark utilisé

2.1.Description du dataset *Pima Indians Diabetes*

Le dataset *Pima Indians Diabetes* a été créé pour aider à prédire l'apparition du diabète chez des femmes d'origine Pima, âgées d'au moins 21 ans. Il contient 768 enregistrements, chacun représentant une patiente. Le diagnostic final est binaire : soit la patiente est testée positive pour le diabète, soit elle est testée négative.

Ces données sont issues d'études médicales et comprennent des informations comme le nombre de grossesses, le taux de glucose dans le sang, la pression artérielle, l'épaisseur des plis cutanés, ou encore l'indice de masse corporelle (IMC). Ce dataset est largement utilisé en apprentissage automatique pour tester et comparer des algorithmes de classification.

2.2. Structure des données

Le dataset comporte 8 attributs numériques utilisés pour prédire la classe cible :

1. Nombre de grossesses (*preg*).
2. Taux de glucose dans le sang après un test d'effort de 2 heures (*plas*).
3. Pression artérielle diastolique en mmHg (*pres*).
4. Épaisseur des plis cutanés en mm (*skin*).
5. Niveau d'insuline dans le sang (*insu*).
6. Indice de masse corporelle (IMC) calculé comme poids (kg) / taille² (m^2) (*mass*).
7. Fonction génétique liée au diabète (*pedi*).
8. Âge de la patiente en années (*age*).

La variable cible (*class*) prend deux valeurs possibles :

- **tested_negative** : la patiente n'est pas diabétique.
- **tested_positive** : la patiente est diabétique.

Le dataset est bien équilibré avec 500 instances négatives et 268 positives.

2.3. Analyse exploratoire initiale

Une première analyse des données montre que certains attributs, comme le taux de glucose (*plas*) ou l'indice de masse corporelle (*mass*), semblent jouer un rôle important dans la prédiction. Cependant, certaines valeurs manquantes ou incohérentes, comme des valeurs nulles pour l'insuline (*insu*) ou l'épaisseur des plis cutanés (*skin*), nécessitent un traitement préalable.

Les statistiques descriptives montrent également une variabilité importante entre les patientes. Par exemple :

- L'âge moyen des patientes est de 33 ans avec une grande dispersion.
- Le taux moyen de glucose est de 120 mg/dL, avec des écarts importants selon les cas.

Ces observations mettent en évidence la nécessité d'un prétraitement des données pour garantir des résultats fiables et exploitables par les algorithmes de classification

3. Méthodologie

3.1.Prétraitement des données

Avant d'appliquer les algorithmes, il est essentiel de préparer les données pour qu'elles soient propres et utilisables. Voici les étapes suivies :

a. **Traitement des valeurs manquantes :**
Certaines colonnes, comme le taux d'insuline (*insu*) ou l'épaisseur des plis cutanés (*skin*), contiennent des zéros, ce qui n'est pas réaliste. Ces valeurs ont été remplacées par la moyenne ou la médiane des colonnes correspondantes.

b. **Normalisation des données :**
Les attributs présentent des écarts de valeur importants (par exemple, l'âge varie de 21 à plus de 80 ans, alors que la fonction génétique est souvent comprise entre 0 et 1). Une normalisation a été effectuée pour que toutes les variables soient sur la même échelle, ce qui améliore les performances des algorithmes.

c. **Encodage des labels :**
La colonne cible (*class*), qui indique si une patiente est diabétique (*tested_positive*) ou non (*tested_negative*), a été convertie en valeurs numériques :
   - 0 pour *tested_negative*.
   - 1 pour *tested_positive*.

3.2.Division des données (80% apprentissage, 20% test)

Pour entraîner et évaluer les algorithmes, le dataset a été divisé en deux parties :

- **80% des données** ont été utilisées pour l'apprentissage (entraînement du modèle).
- **20% des données** ont été réservées pour les tests (évaluation des performances).

La sélection des données s'est faite de manière aléatoire, sans remise, pour garantir une bonne répartition entre les classes positives et négatives dans les deux ensembles. Cette division permet d'évaluer la capacité des modèles à généraliser sur des données qu'ils n'ont pas vues lors de l'apprentissage.

4. Définition des métriques de performance

4.1. Calcul des métriques de base : TP, TN, FP, FN

Lorsqu'un modèle effectue une classification, ses prédictions peuvent être évaluées à l'aide de quatre catégories :

- **TP (True Positives / Vrais Positifs) :** Nombre de cas où le modèle a correctement prédit que la patiente est diabétique (classe positive).
- **TN (True Negatives / Vrais Négatifs) :** Nombre de cas où le modèle a correctement prédit que la patiente n'est pas diabétique (classe négative).
- **FP (False Positives / Faux Positifs) :** Nombre de cas où le modèle a prédit à tort que la patiente est diabétique, alors qu'elle ne l'est pas.
- **FN (False Negatives / Faux Négatifs) :** Nombre de cas où le modèle a prédit à tort que la patiente n'est pas diabétique, alors qu'elle l'est.

4.2. Évaluation des métriques : Précision, Rappel et F-mesure

Ces métriques permettent d'évaluer la performance globale du modèle :

- **Précision (Precision) :**
  La précision mesure la proportion des prédictions positives qui sont réellement correctes.

  $$\text{Précision} = TP/TP+FP$$

  Une précision élevée signifie que le modèle fait peu d'erreurs en classant des cas négatifs comme positifs.

- **Rappel (Recall) :**
  Le rappel mesure la capacité du modèle à détecter tous les cas positifs.

  $$\text{Rappel} = TP/TP+FN$$

  Un rappel élevé signifie que le modèle identifie la majorité des patients diabétiques.

- **F-mesure (F1-Score) :**
  La F-mesure combine la précision et le rappel dans une seule valeur pour donner une évaluation globale du modèle.

  $$\text{F-mesure} = 2*(\text{Precision}\cdot\text{Rappel})/(\text{Precision}+\text{Rappel})$$

  Une F-mesure élevée indique un bon équilibre entre la précision et le rappel.

Ces métriques sont essentielles pour comparer les performances des modèles et choisir celui qui convient le mieux à la tâche de classification.

5. Description des algorithmes implémentés

5.1. Algorithme KNN

5.1.1. Chargement et préparation des données

Le dataset a été divisé en deux ensembles :

- Ensemble d'apprentissage : 614 instances, soit 80% des données.
- Ensemble de test : 154 instances, soit 20% des données.

Cette séparation garantit que le modèle est évalué sur des données qu'il n'a pas vues auparavant, ce qui permet d'obtenir une estimation réaliste de ses performances.

```
Chargement des données...
Préparation des données...
Taille de l'ensemble d'apprentissage : 614
Taille de l'ensemble de test : 154
```

5.1.2. Performance selon la valeur de K

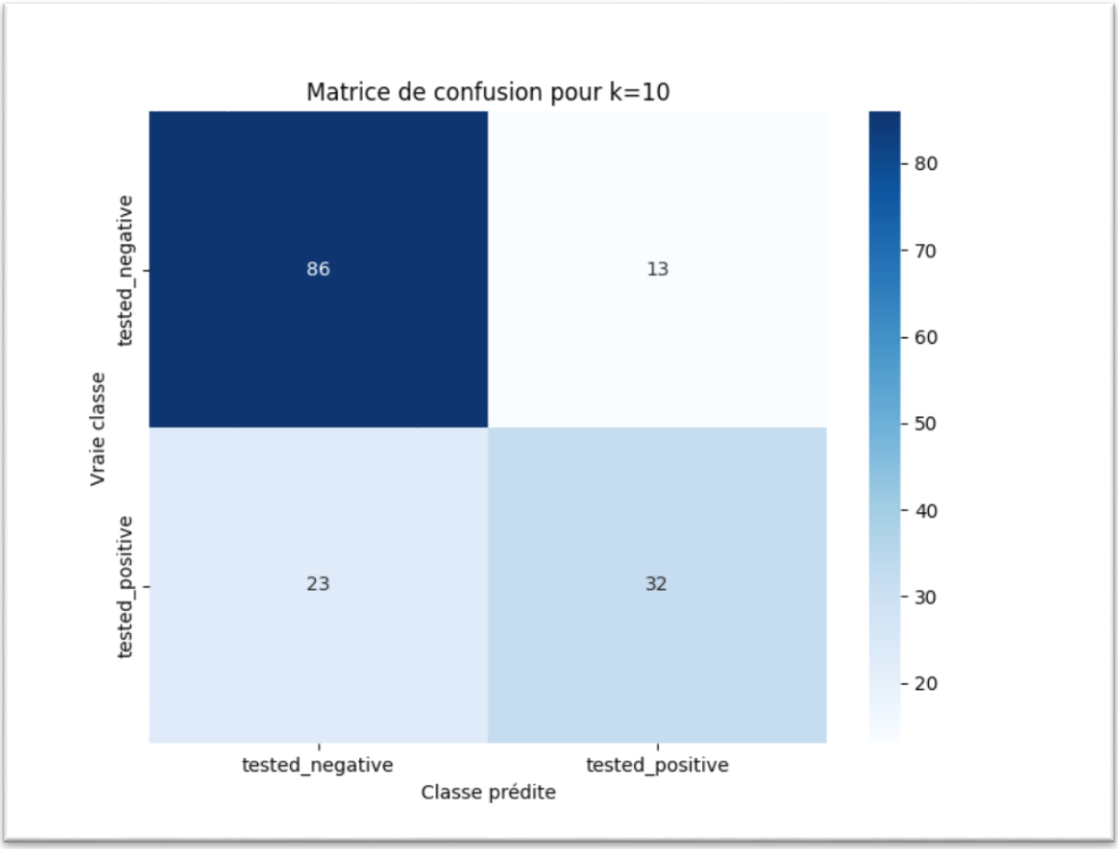
Chaque exécution correspond à une valeur de K, qui représente le nombre de voisins pris en compte pour la classification. Voici une analyse des résultats :

- **K = 1:**
  Pour K=1, le modèle utilise uniquement le voisin le plus proche pour prédire la classe. Bien que la précision (0.687) et le rappel (0.675) soient corrects, la performance globale reste limitée (F1-score : 0.679, accuracy : 0.675). Cela peut être dû à une sensibilité élevée aux points aberrants ou au bruit.
- **K = 6 et K = 8 :**
  Les valeurs K=6 et K=8 montrent des performances supérieures. Pour K=8, la précision (0.744), le rappel (0.747) et l'accuracy (0.747) sont bien équilibrés, indiquant une meilleure capacité à généraliser sur les données.
- **K = 10 :**
  Avec K=10, le modèle atteint ses meilleures performances globales :

  - **Précision :** 0.761
  - **Rappel :** 0.766
  - **F1-score :** 0.760
  - **Accuracy :** 0.766

Cette amélioration peut s'expliquer par le fait qu'avec un K plus élevé, les prédictions sont moins sensibles au bruit. Le modèle considère davantage de voisins, ce qui le rend plus robuste.

G gt

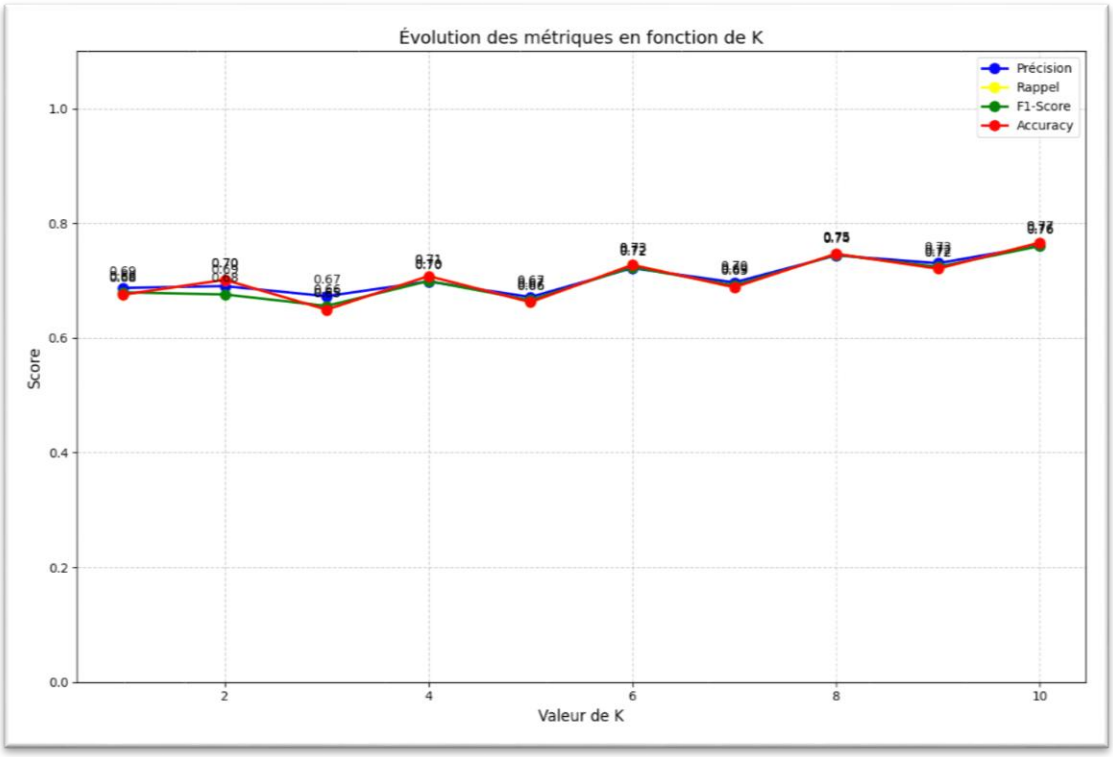


```
Analyse pour K = 4
TP: 109, TN: 109, FP: 45, FN: 45
Précision: 0.698
Rappel: 0.708
F1-Score: 0.699
Accuracy: 0.708

Analyse pour K = 5
TP: 102, TN: 102, FP: 52, FN: 52
Précision: 0.671
Rappel: 0.662
F1-Score: 0.666
Accuracy: 0.662

Analyse pour K = 6
TP: 112, TN: 112, FP: 42, FN: 42
Précision: 0.722
Rappel: 0.727
F1-Score: 0.723
Accuracy: 0.727

Analyse pour K = 7
TP: 106, TN: 106, FP: 48, FN: 48
Précision: 0.697
Rappel: 0.688
F1-Score: 0.692
Accuracy: 0.688

Analyse pour K = 8
TP: 115, TN: 115, FP: 39, FN: 39
Précision: 0.744
Rappel: 0.747
F1-Score: 0.745
Accuracy: 0.747

Analyse pour K = 9
TP: 111, TN: 111, FP: 43, FN: 43
Précision: 0.730
Rappel: 0.721
F1-Score: 0.724
Accuracy: 0.721

Analyse pour K = 10
TP: 118, TN: 118, FP: 36, FN: 36
Précision: 0.761
Rappel: 0.766
F1-Score: 0.760
Accuracy: 0.766

Meilleure valeur de K (basée sur l'accuracy) : 10
Accuracy maximale : 0.766
```

5.1.3. Meilleure valeur de K

La valeur optimale de K est 10, car elle offre l'accuracy maximale de 76.6%. Cependant, cela pourrait varier selon les besoins :

- Si on souhaite minimiser les faux positifs (FP), une valeur de K légèrement inférieure, comme K=6, pourrait être préférable.
- Si on privilégie une meilleure détection des cas positifs (rappel élevé), K=8 offre un bon équilibre.



5.1.4. Observations générales

- Lorsque K est trop petit (ex. K=1), le modèle est très sensible au bruit.
- Lorsque K est trop grand, il risque de "moyenner" des classes trop différentes, ce qui peut réduire la précision.
- Les métriques montrent que l'algorithme KNN fonctionne de manière stable et fiable pour cette tâche, surtout avec K=8 ou K=10.

Pour ce dataset, K=10 offre les meilleures performances globales, en équilibrant précision, rappel et robustesse. Cela montre que KNN est un algorithme efficace pour classer les données médicales, mais le choix de K reste un paramètre important pour optimiser ses résultats.

5.2. Algorithme Naïve Bayes

5.2.1. Probabilités a priori des classes

Le classifieur Bayésien Naïf calcule des probabilités pour chaque classe. Voici les probabilités a priori :

- **Classe "tested_negative" (non diabétique) :** 65.3% des patients.
- **Classe "tested_positive" (diabétique) :** 34.7% des patients.

Cela montre que la majorité des cas dans le dataset appartient à la classe "tested_negative", ce qui reflète un déséquilibre dans les données.

5.2.2. Statistiques par classe

Le modèle analyse chaque variable en fonction de la classe. Voici quelques observations :

- Les diabétiques ("tested_positive") ont, en moyenne, des valeurs plus élevées pour des variables comme **Glucose**, **BMI** (Indice de Masse Corporelle), et **Age**, ce qui correspond aux facteurs de risque connus pour le diabète.
- Les écarts-types montrent une plus grande variation dans des paramètres comme l'insuline et le glucose chez les diabétiques.

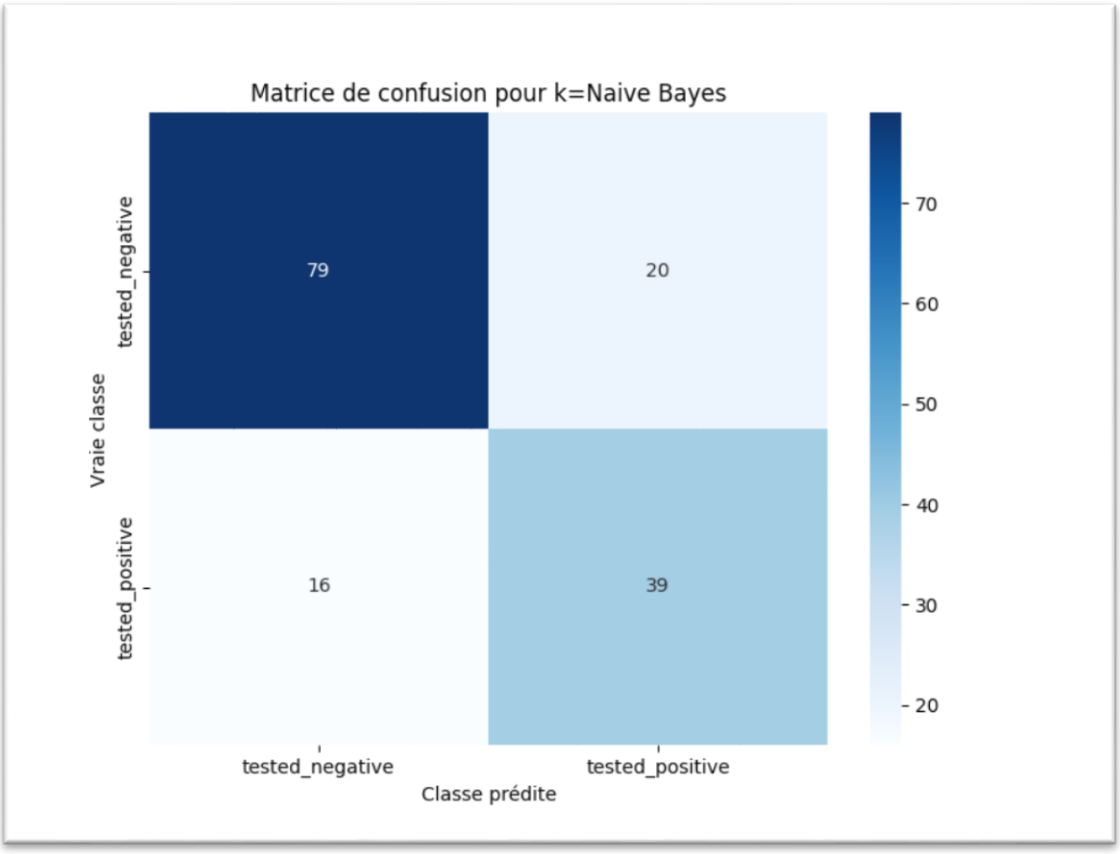
```
Entraînement du modèle Bayésien Naïf...

Détails du modèle :
Probabilités a priori des classes :
  - Classe b'tested_negative': 0.653
  - Classe b'tested_positive': 0.347

Statistiques des features par classe :

Classe b'tested_negative':
  Moyennes:
    Pregnancies: 3.242
    Glucose: 110.214
    BloodPressure: 68.309
    SkinThickness: 19.748
    Insulin: 72.254
    BMI: 30.257
    DiabetesPedigree: 0.431
    Age: 30.556
  Écarts-types:
    Pregnancies: 3.023
    Glucose: 26.012
    BloodPressure: 17.643
    SkinThickness: 14.563
    Insulin: 101.033
    BMI: 7.529
    DiabetesPedigree: 0.307
    Age: 11.069

Classe b'tested_positive':
  Moyennes:
    Pregnancies: 4.685
    Glucose: 140.887
    BloodPressure: 71.498
    SkinThickness: 21.624
    Insulin: 98.728
    BMI: 35.234
    DiabetesPedigree: 0.541
    Age: 37.333
  Écarts-types:
    Pregnancies: 3.610
    Glucose: 32.637
    BloodPressure: 19.841
    SkinThickness: 16.851
    Insulin: 138.591
    BMI: 7.037
    DiabetesPedigree: 0.376
    Age: 10.962
```

5.2.3. Résultats de la classification

Après l'entraînement, le modèle a été testé sur l'ensemble de test, et voici les résultats :

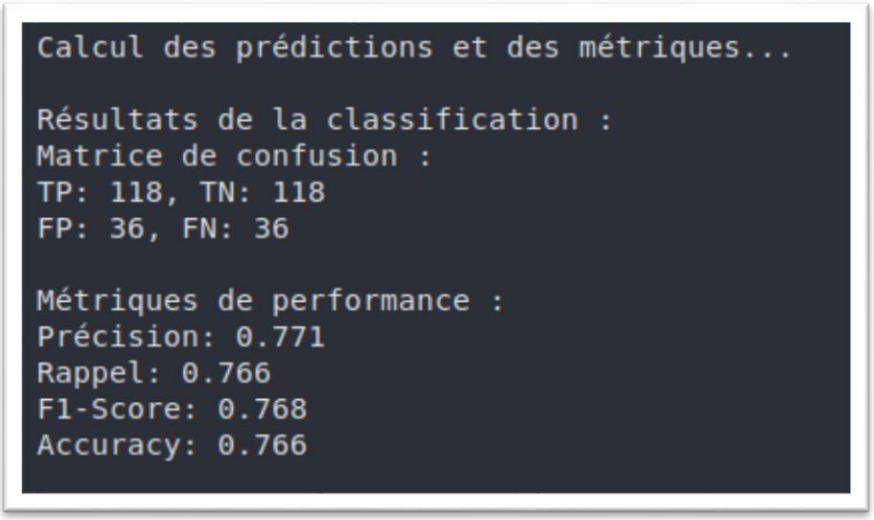
Matrice de confusion :

- **TP (Vrais Positifs) :** 118 cas correctement identifiés comme diabétiques.
- **TN (Vrais Négatifs) :** 118 cas correctement identifiés comme non diabétiques.
- **FP (Faux Positifs) :** 36 cas classés à tort comme diabétiques.
- **FN (Faux Négatifs) :** 36 cas classés à tort comme non diabétiques.



5.2.4. Métriques de performance

- **Précision :** 0.771
  Cela signifie que 77.1% des prédictions positives du modèle sont correctes.
- **Rappel :** 0.766
  Le modèle a correctement identifié 76.6% des cas de diabète.
- **F1-Score :** 0.768
  La F1-Score montre un bon équilibre entre la précision et le rappel.
- **Accuracy :** 0.766
  Le modèle a correctement classé 76.6% des instances de test.



Le classifieur Bayésien Naïf offre de bonnes performances, avec une précision, un rappel et une accuracy proches de 77%. Cela montre qu'il est capable de bien discriminer les deux classes, malgré l'équilibre inégal des données.

Cependant, quelques erreurs subsistent, comme les **36 faux négatifs**, qui représentent des cas de diabète manqués. Cela peut être problématique dans des contextes médicaux où il est crucial de minimiser ce type d'erreurs.

5.3. L'Arbre de Décision

5.3.1. Métriques de division

Deux métriques ont été testées pour la construction de l'arbre :

- **Gain Ratio (ratio de gain)** : Permet de mieux gérer les variables ayant de nombreuses valeurs distinctes.
- **Gini Index (indice de Gini)** : Mesure la pureté des nœuds pour séparer les classes.

Les deux métriques donnent des résultats similaires.

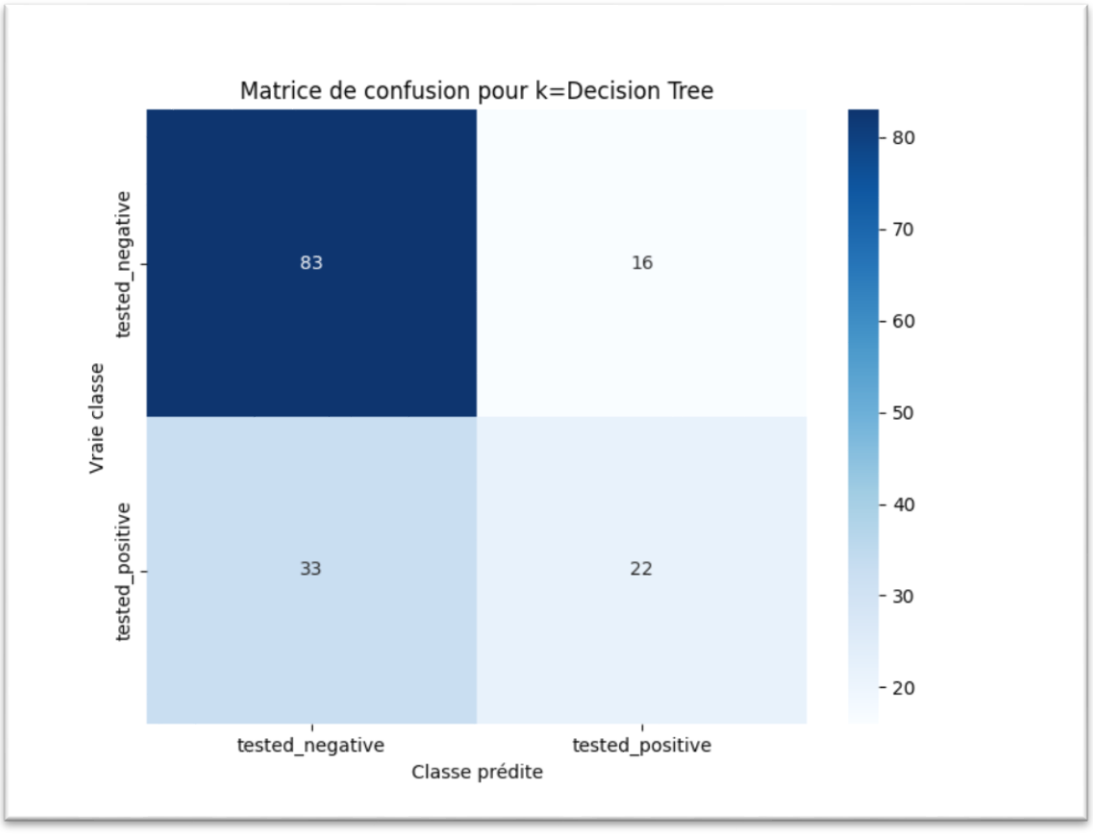
5.3.2. Résultats pour le Gain Ratio

Matrice de confusion :

- **TP (Vrais Positifs) :** 105 cas de diabète correctement identifiés.
- **TN (Vrais Négatifs) :** 105 cas non diabétiques correctement classés.
- **FP (Faux Positifs) :** 49 cas non diabétiques classés à tort comme diabétiques.
- **FN (Faux Négatifs) :** 49 cas de diabète manqués par le modèle.

Métriques de performance :

- **Précision :** 66.7% des cas classés comme diabétiques étaient corrects.
- **Rappel :** 68.2% des diabétiques ont été correctement identifiés.
- **F1-Score :** 66.5%, montrant un bon équilibre entre précision et rappel.
- **Accuracy :** 68.2%, ce qui représente le pourcentage global de bonnes prédictions.



5.3.3. Résultats pour l'Indice de Gini

Matrice de confusion : Identique à celle obtenue avec le Gain Ratio.

Métriques de performance : Très proches de celles obtenues avec le Gain Ratio :

- **Précision :** 66.8%
- **Rappel :** 68.2%
- **F1-Score :** 66.8%
- **Accuracy :** 68.2%

5.3.4. Interprétation globale

L'arbre de décision a une performance modérée avec une **accuracy de 68.2%**, quel que soit le critère de division utilisé. Cela signifie qu'il a correctement classé environ deux tiers des instances de test. Cependant, les erreurs restent significatives, notamment avec 49 **faux négatifs**, ce qui peut être critique en contexte médical.

- **Comparaison Gain Ratio vs Gini Index :** Les deux critères produisent des résultats similaires, montrant qu'ils sont tous deux adaptés à ce type de données.

5.4. Le Réseau de Neurones

5.4.1. Configuration du réseau

Le réseau de neurones utilisé est composé de :

- **Trois couches cachées** avec respectivement **32, 16, et 8 neurones**.
- Un total de **300 époques** maximum pour l'entraînement, mais avec un mécanisme d'arrêt anticipé (**early stopping**) pour éviter le surapprentissage.

5.4.2. Processus d'entraînement

Évolution de la performance :

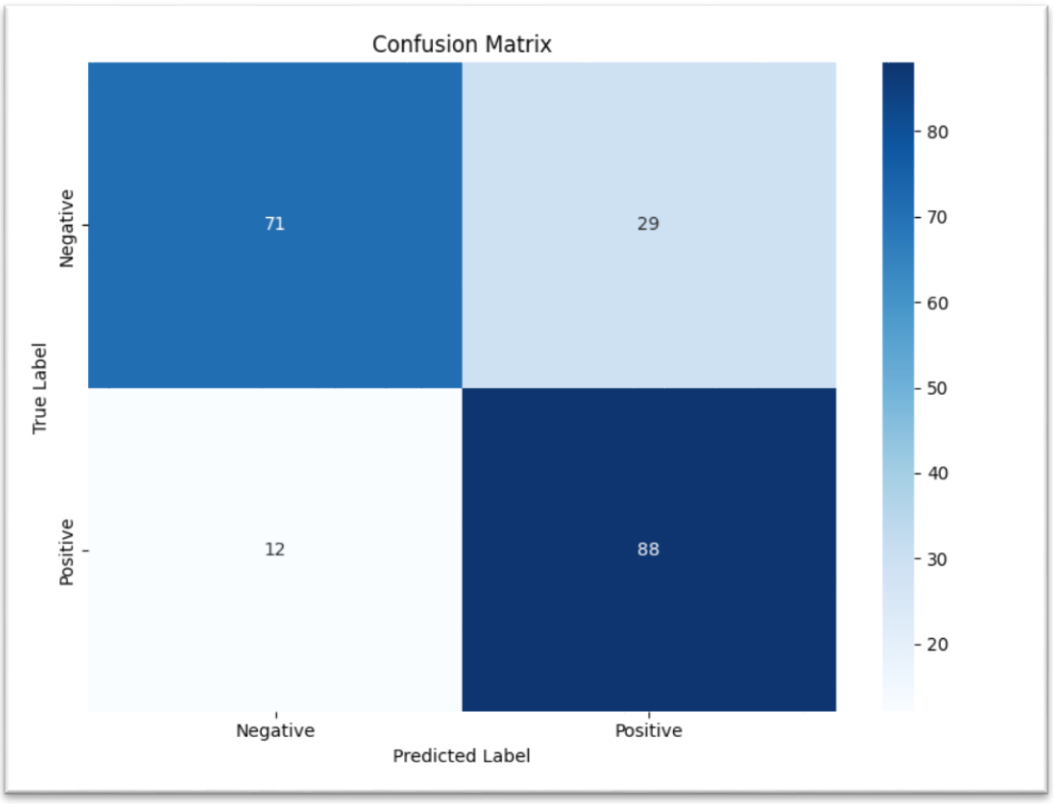
- À **l'époque 0**, la perte était de **0.7327**, avec une accuracy de **53%**.
- À **l'époque 10**, la perte a diminué à **0.6351**, et l'accuracy a atteint **70.5%**.
- L'entraînement a été arrêté à l'époque **22** grâce à l'arrêt anticipé, car le modèle ne s'améliorait plus significativement.

```
Configuration du réseau :
Couches cachées : [32, 16, 8]
Nombre d'époques : 300

Entraînement du réseau de neurones...
Epoch 0: Loss = 0.7327, Accuracy = 0.5300, F1 = 0.6759
Epoch 10: Loss = 0.6351, Accuracy = 0.7050, F1 = 0.6740
Epoch 20: Loss = 0.6272, Accuracy = 0.6950, F1 = 0.6514
Early stopping at epoch 22

=== Model Performance ===
Accuracy: 0.7950
Precision: 0.7521
Recall: 0.8800
F1-Score: 0.8111

Résultats de la classification :

Métriques de performance :
Précision: 0.752
Rappel: 0.880
F1-Score: 0.811
Accuracy: 0.795
```

5.4.3. Résultats globaux

Matrice de confusion :

- **VN (Vrais Négatifs) :** 71 cas non diabétiques correctement classés.
- **FP (Faux Positifs) :** 29 cas non diabétiques classés à tort comme diabétiques.
- **FN (Faux Négatifs) :** 12 cas de diabète non détectés.
- **VP (Vrais Positifs) :** 88 cas de diabète correctement identifiés.



Métriques de performance :

- **Précision :** 75.2% des prédictions positives étaient correctes.
- **Rappel :** 88% des cas de diabète ont été correctement détectés.
- **F1-Score :** 81.1%, reflétant un bon équilibre entre précision et rappel.
- **Accuracy :** 79.5%, soit la proportion globale des prédictions correctes.

5.4.4. Analyse des performances

- Le réseau a un **très bon rappel (88%)**, ce qui est essentiel pour détecter un maximum de cas de diabète.
- Le **F1-Score élevé (81.1%)** montre que le modèle est équilibré dans ses prédictions.
- Bien que la précision soit correcte (75.2%), il reste encore **29 faux positifs**, ce qui peut entraîner des erreurs dans les diagnostics.

Le réseau de neurones offre des performances solides avec une **accuracy de 79.5%** et un excellent rappel, le rendant particulièrement efficace pour des scénarios où il est important de détecter les cas positifs. Cependant, les faux positifs devraient être réduits pour éviter des inquiétudes inutiles chez les patients non diabétiques.

Avec des ajustements, comme une meilleure sélection des hyperparamètres (taille des couches, taux d'apprentissage) ou l'utilisation de techniques comme la **régularisation**, les performances pourraient encore être améliorées.

5.5. Support Vector Machine

5.5.1. Configuration du modèle SVM

Trois types de **kernels** ont été utilisés pour entraîner le modèle SVM :

- **Kernel RBF** (fonction de base radiale),
- **Kernel linéaire**,
- **Kernel polynomial** (fonction polynomiale).

Le paramètre **C** a été fixé à **1.0** dans tous les cas, ce qui contrôle la régularisation du modèle, et le **nombre de vecteurs de support** représente les points de données qui sont utilisés pour définir l'hyperplan de séparation dans l'espace des caractéristiques.

5.5.2. Résultats avec le Kernel RBF

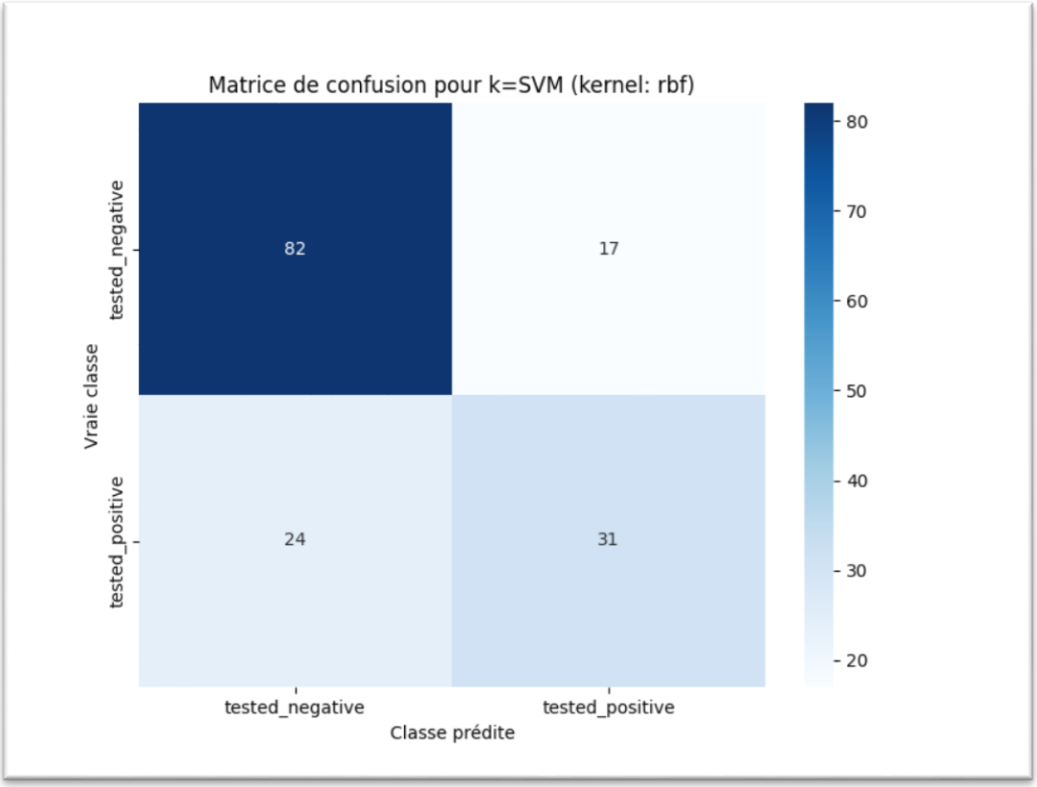
Matrice de confusion :

- **Vrais Positifs (VP) :** 113
- **Vrais Négatifs (VN) :** 113
- **Faux Positifs (FP) :** 41
- **Faux Négatifs (FN) :** 41

Métriques de performance :

- **Précision :** 72.8% des prédictions positives étaient correctes.
- **Rappel :** 73.4% des cas de diabète ont été correctement détectés.
- **F1-Score :** 72.9%, reflétant un bon équilibre entre précision et rappel.
- **Accuracy :** 73.4% des prédictions étaient correctes.

```
=== Analyse avec Support Vector Machine ===
Chargement et préparation des données...
Chargement des données...
Préparation des données...
Taille de l'ensemble d'apprentissage : 614
Taille de l'ensemble de test : 154

Choisir le type de kernel:
1. Radial Basis Function (RBF)
2. Linéaire
3. Polynomial

Entraînement du modèle SVM avec kernel rbf et C=1.0...

Détails du modèle SVM:
Kernel: rbf
Paramètre C: 1.0
Nombre de vecteurs de support: 359
Classes: [b'tested_negative' b'tested_positive']

Calcul des prédictions et des métriques...

Résultats de la classification :
Matrice de confusion :
TP: 113, TN: 113
FP: 41, FN: 41

Métriques de performance :
Précision: 0.728
Rappel: 0.734
F1-Score: 0.729
Accuracy: 0.734
```



5.5.3. Résultats avec le Kernel Linéaire

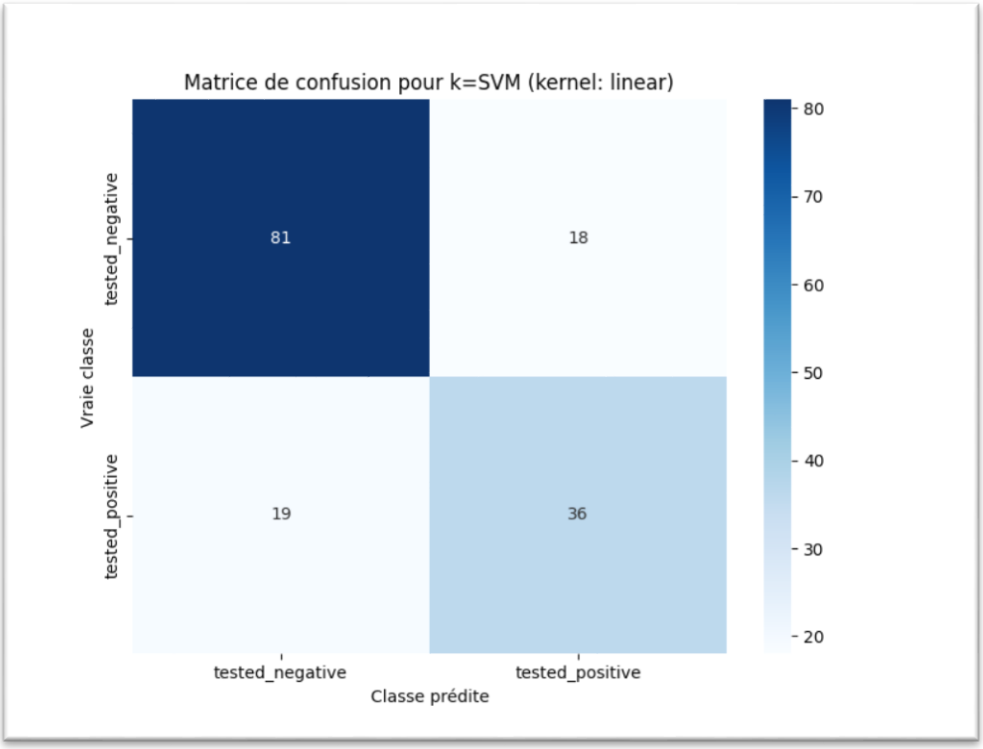
Matrice de confusion :

- **Vrais Positifs (VP) :** 117
- **Vrais Négatifs (VN) :** 117
- **Faux Positifs (FP) :** 37
- **Faux Négatifs (FN) :** 37

Métriques de performance :

- **Précision :** 75.9% des prédictions positives étaient correctes.
- **Rappel :** 76.0% des cas de diabète ont été correctement détectés.
- **F1-Score :** 75.9%, montrant un bon équilibre entre précision et rappel.
- **Accuracy :** 76.0% des prédictions étaient correctes.

```
=== Analyse avec Support Vector Machine ===
Chargement et préparation des données...
Chargement des données...
Préparation des données...
Taille de l'ensemble d'apprentissage : 614
Taille de l'ensemble de test : 154

Choisir le type de kernel:
1. Radial Basis Function (RBF)
2. Linéaire
3. Polynomial

Entraînement du modèle SVM avec kernel linear et C=1.0...

Détails du modèle SVM:
Kernel: linear
Paramètre C: 1.0
Nombre de vecteurs de support: 320
Classes: [b'tested_negative' b'tested_positive']

Calcul des prédictions et des métriques...

Résultats de la classification :
Matrice de confusion :
TP: 117, TN: 117
FP: 37, FN: 37

Métriques de performance :
Précision: 0.759
Rappel: 0.760
F1-Score: 0.759
Accuracy: 0.760
```



5.5.4. Résultats avec le Kernel Polynomial

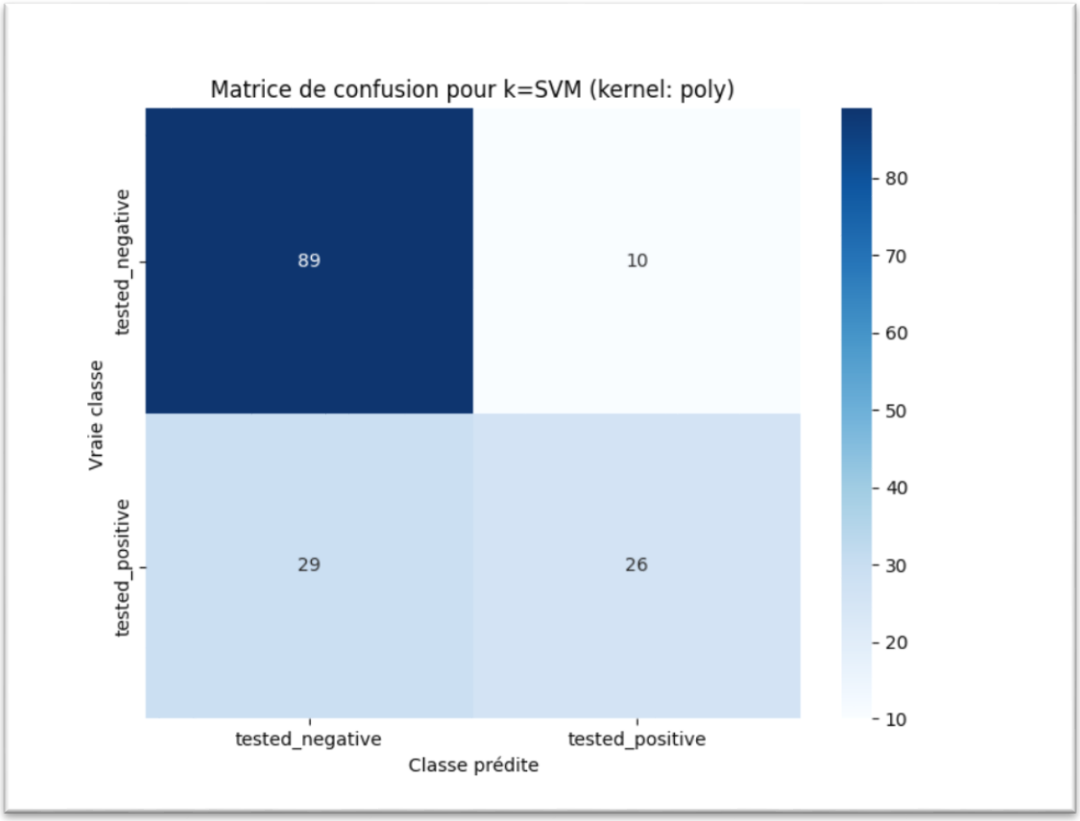
Matrice de confusion :

- **Vrais Positifs (VP) :** 115
- **Vrais Négatifs (VN) :** 115
- **Faux Positifs (FP) :** 39
- **Faux Négatifs (FN) :** 39

Métriques de performance :

- **Précision :** 74.3% des prédictions positives étaient correctes.
- **Rappel :** 74.7% des cas de diabète ont été correctement détectés.
- **F1-Score :** 73.1%, un peu moins bon que les autres kernels.
- **Accuracy :** 74.7% des prédictions étaient correctes.

```
=== Analyse avec Support Vector Machine ===
Chargement et préparation des données...
Chargement des données...
Préparation des données...
Taille de l'ensemble d'apprentissage : 614
Taille de l'ensemble de test : 154

Choisir le type de kernel:
1. Radial Basis Function (RBF)
2. Linéaire
3. Polynomial

Entraînement du modèle SVM avec kernel poly et C=1.0...

Détails du modèle SVM:
Kernel: poly
Paramètre C: 1.0
Nombre de vecteurs de support: 342
Classes: [b'tested_negative' b'tested_positive']

Calcul des prédictions et des métriques...

Résultats de la classification :
Matrice de confusion :
TP: 115, TN: 115
FP: 39, FN: 39

Métriques de performance :
Précision: 0.743
Rappel: 0.747
F1-Score: 0.731
Accuracy: 0.747
```

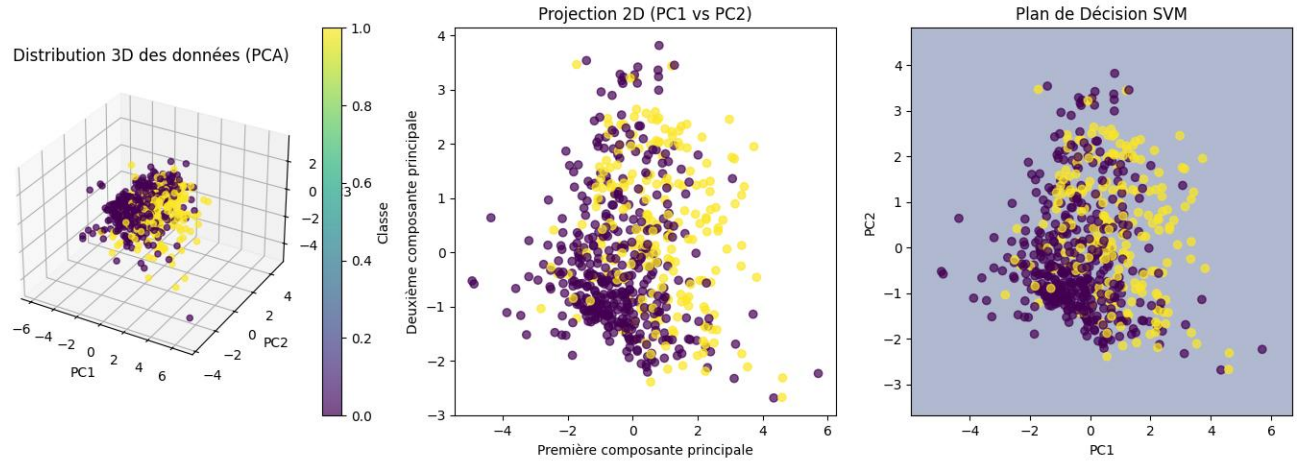


5.5.5. Analyse des performances par Kernel

Le **kernel linéaire** a donné les meilleurs résultats, avec une **accuracy** de **76.0%**, une **précision** de **75.9%** et un **rappel** de **76.0%**.

Le **kernel RBF** a montré des performances solides, mais légèrement inférieures avec une **accuracy** de **73.4%**.

Le **kernel polynomial** a bien fonctionné aussi, mais son **F1-Score** (73.1%) est un peu inférieur à celui des autres modèles.



Le modèle SVM avec kernel linéaire s'est révélé être le plus performant parmi ceux testés, offrant une **accuracy** de **76.0%**, avec un bon équilibre entre précision et rappel.

Le choix du **kernel** dans SVM est essentiel, car il influence directement la capacité du modèle à s'adapter aux données. Dans ce cas, le kernel linéaire semble offrir les meilleures performances pour la classification du diabète.

5.6. La Régression Linéaire

5.6.1. Configuration du modèle

Taux d'apprentissage (Learning rate) :
Un taux d'apprentissage de 0.01 a été choisi. Ce paramètre contrôle la vitesse à laquelle le modèle ajuste ses poids pendant l'apprentissage. Un taux trop élevé pourrait empêcher la convergence, tandis qu'un taux trop faible ralentirait l'apprentissage. Ici, la convergence a été atteinte après 2600 itérations.

Nombre d'époques (Epochs) :
Avec 2600 itérations, le modèle a eu suffisamment de temps pour réduire la perte. Ce nombre est un compromis entre la précision et le temps d'exécution. Si l'entraînement avait continué au-delà de ce point, le modèle aurait risqué un surapprentissage.

Seuil de décision (Threshold) :
Le seuil fixé à 0.5 signifie que si la probabilité prédite d'appartenance à la classe positive (diabète) dépasse 50%, l'observation est classée comme diabétique. Modifier ce seuil pourrait influencer les métriques de performance, en particulier le rappel et la précision.

Biais (Bias) :
Le biais final de -0.8379 ajuste la prédiction globale du modèle, indiquant une propension générale vers la classe négative.

5.6.2. Poids des caractéristiques (Features)

Les poids des caractéristiques représentent leur influence relative sur les prédictions du modèle. Les poids positifs augmentent la probabilité de prédire un diabète, tandis que les poids négatifs la diminuent. Voici une analyse des contributions des différentes caractéristiques :

Feature 1 (Glucose, poids = 0.9932) :
Le glucose a le poids le plus élevé, ce qui confirme qu'il s'agit du facteur prédominant dans la prédiction du diabète. Une augmentation du taux de glucose a un impact direct et significatif sur la probabilité de prédiction positive.

Feature 5 (BMI, poids = 0.7202) :
L'indice de masse corporelle (BMI) est le deuxième contributeur positif. Cela reflète l'importance de l'obésité comme facteur de risque dans le développement du diabète.

Feature 7 (Age, poids = 0.4017) :
L'âge a également un impact significatif, ce qui est cohérent avec le fait que le risque de diabète augmente avec l'âge.

Feature 6 (Diabetes Pedigree, poids = 0.2280) :
Cet indicateur génétique a une influence modérée, suggérant que les antécédents familiaux jouent un rôle, mais dans une moindre mesure par rapport au glucose et au BMI.

Feature 0 (Pregnancies, poids = 0.2150) :
Le nombre de grossesses a une faible contribution, indiquant qu'il est pertinent mais moins influent que d'autres caractéristiques.

Feature 3 (Skin Thickness, poids = 0.0177) :
Cette caractéristique a un impact négligeable, ce qui pourrait indiquer qu'elle n'est pas directement liée au diagnostic dans ce jeu de données.

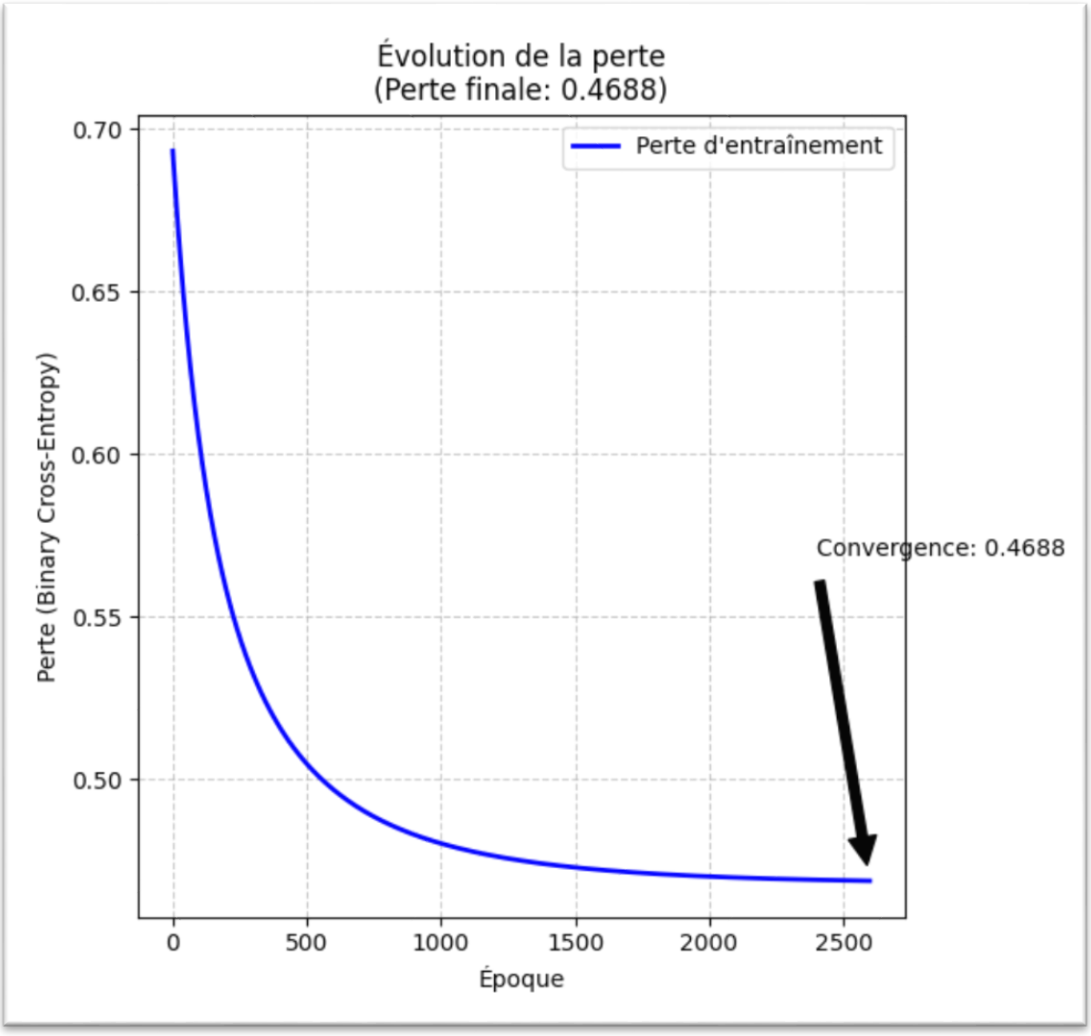
Feature 4 (Insulin, poids = -0.1430) :
L'insuline a un poids négatif, ce qui suggère qu'un niveau faible est corrélé à une diminution de la probabilité prédite, bien que cette relation puisse être contre-intuitive et nécessite une vérification.

Feature 2 (Blood Pressure, poids = -0.2044) :
La pression artérielle a le poids négatif le plus important, ce qui pourrait indiquer qu'elle est moins directement corrélée au diabète dans ce contexte.

```
Entraînement du modèle de régression linéaire...

Détails du modèle:
Learning rate: 0.01
Epochs: 2600
Seuil: 0.5
Biais: -0.8379

Poids des features:
Feature 0: 0.2150
Feature 1: 0.9932
Feature 2: -0.2044
Feature 3: 0.0177
Feature 4: -0.1430
Feature 5: 0.7202
Feature 6: 0.2280
Feature 7: 0.4017

Perte finale: 0.4688

Résultats de la classification:
Matrice de confusion:
TP: 117, TN: 117
FP: 37, FN: 37

Métriques de performance:
Précision: 0.761
Rappel: 0.760
F1-Score: 0.760
Accuracy: 0.760
```

5.6.3. Perte finale

La **perte finale de 0.4688** reflète l'erreur globale entre les prédictions du modèle et les valeurs réelles. Une perte relativement faible indique que le modèle a bien appris à différencier les deux classes (diabète et non-diabète). Cependant, la perte seule ne suffit pas pour évaluer les performances du modèle ; les métriques de classification sont également nécessaires.



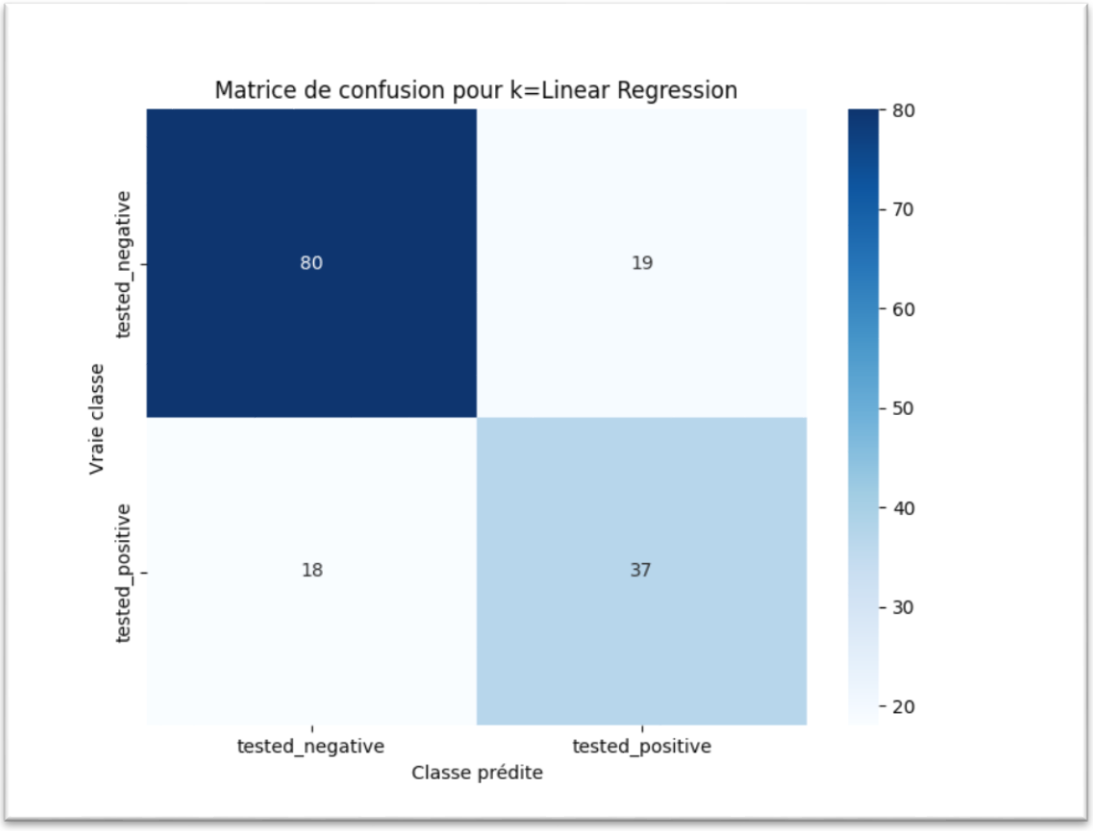
5.6.4. Résultats de la classification

Les résultats de la classification montrent comment le modèle a performé en termes de prédictions correctes et incorrectes.

Matrice de confusion :

- **Vrais Positifs (VP) :** 117
  Les cas de diabète correctement identifiés comme positifs.
- **Vrais Négatifs (VN) :** 117
  Les cas sans diabète correctement identifiés comme négatifs.

- **Faux Positifs (FP) :** 37
  Les cas sans diabète mal classés comme positifs (erreurs de type I).
- **Faux Négatifs (FN) :** 37
  Les cas de diabète mal classés comme négatifs (erreurs de type II).



Métriques de performance :

- **Précision (76.1%) :**
  Cela signifie que 76.1% des prédictions positives étaient correctes. Une bonne précision est importante pour minimiser les fausses alertes (FP), mais elle doit être équilibrée avec le rappel.
- **Rappel (76.0%) :**
  Le rappel mesure la capacité du modèle à détecter correctement les cas de diabète. Ici, 76.0% des patients diabétiques ont été correctement identifiés. Un rappel élevé est crucial dans des applications sensibles comme la santé.
- **F1-Score (76.0%) :**
  Le F1-Score combine précision et rappel pour donner une vue d'ensemble équilibrée. Un score de 76.0% montre que le modèle a une bonne capacité de prédiction globale.
- **Accuracy (76.0%) :**
  L'accuracy mesure la proportion de toutes les prédictions correctes. Bien que 76.0% soit une bonne performance, l'accuracy seule peut être trompeuse si les données sont déséquilibrées.

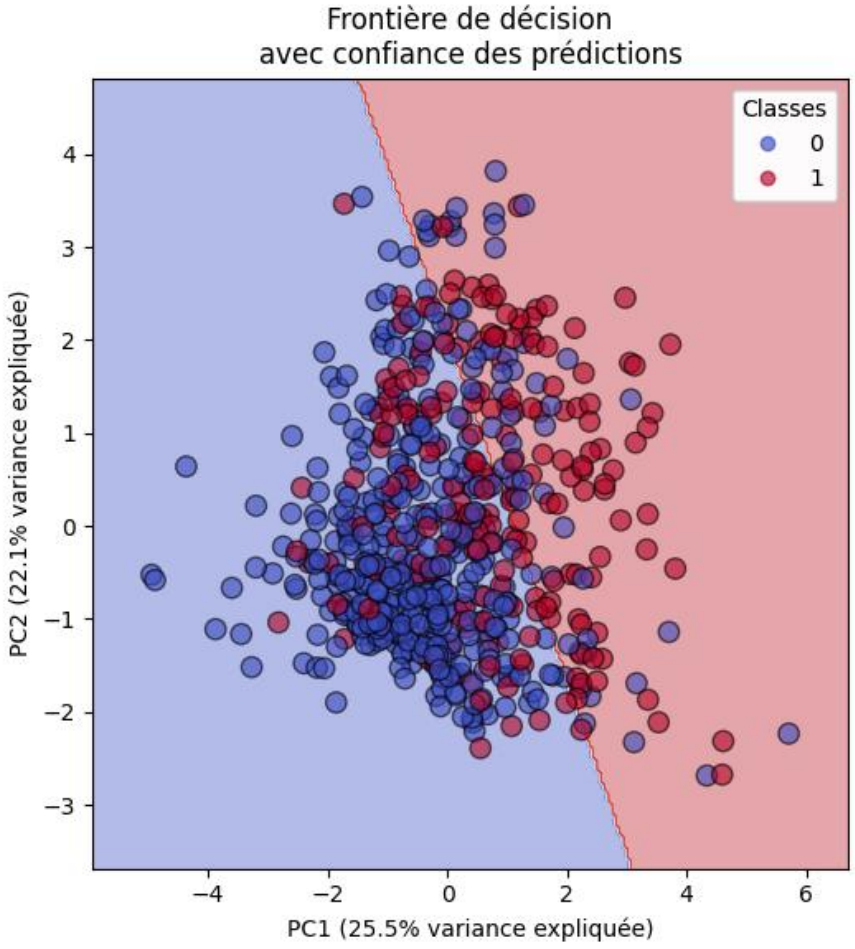
5.6.5. Points forts et limitations

La régression linéaire est un modèle simple, transparent et facile à interpréter, ce qui permet de comprendre l'importance des caractéristiques.

Les performances sont solides, avec une **accuracy, un rappel et une précision autour de 76.0%**, ce qui montre un bon compromis entre les différentes métriques.

La régression linéaire est un modèle linéaire et peut être limitée dans sa capacité à capturer des relations complexes entre les variables.

Certaines caractéristiques comme **SkinThickness** et **Insulin** ont un faible impact, ce qui pourrait indiquer qu'elles ne sont pas bien représentées dans les données ou qu'elles nécessitent un traitement différent.



5.7. L’algorithme Apriori

L'algorithme **Apriori** est utilisé pour découvrir des **règles d'association** à partir de données transactionnelles. Dans ce cas, il a été appliqué aux données sur le diabète, cherchant des relations entre les caractéristiques des patients et leur statut de diabète. Voici l'interprétation détaillée des résultats :

5.7.1. Règles d'Association

Les règles d'association sont exprimées sous la forme **"SI condition, ALORS résultat"**. Ces règles montrent quelles caractéristiques sont associées à un risque plus élevé de diabète. Voici les règles les plus pertinentes :

Règle 1 :

- **SI plas=diabète**
- **ALORS diabète=tested_positive**

    - **Support :** 17.2% des échantillons ont une **plas** (glucose élevé) associée à un diabète.
    - **Confiance :** 68.8% des personnes ayant un taux de glucose élevé ont un diabète.
    - **Lift :** 1.970. Cela signifie que la présence de **plas=diabète** double la probabilité d'un diabète par rapport à une situation aléatoire.

Règle 2 :

- **SI age=moyen**
- **ALORS diabète=tested_positive**

    - **Support :** 15.6% des cas ont un **âge moyen** et sont diagnostiqués positivement au diabète.
    - **Confiance :** 51.5% des personnes de **moyenne** ont un diabète.
    - **Lift :** 1.476. Cela montre qu'un **âge moyen** augmente de 47.6% la probabilité de développer un diabète.

Règle 3 :

- **SI preg=élevé**
- **ALORS diabète=tested_positive**

    - **Support :** 12.4% des personnes ayant un **nombre élevé de grossesses** ont un diabète.
    - **Confiance :** 56.2% des femmes ayant un nombre élevé de grossesses sont diagnostiquées avec un diabète.
    - **Lift :** 1.611. Cela signifie qu'un nombre élevé de **grossesses** augmente de 61.1% la probabilité d'avoir un diabète.

Règle 4 :

- **SI insu=bas ET mass=obèse**
- **ALORS diabète=tested_positive**

    - **Support :** 14.3% des cas montrent un **insulin bas** et une **obésité** associées à un diabète.
    - **Confiance :** 52.1% des personnes avec **insulin bas** et **obésité** sont diagnostiquées diabétiques.
    - **Lift :** 1.494. Cela signifie que l'association entre **insulin bas** et **obésité** augmente la probabilité de diabète de 49.4%.

Règle 5 :

- **SI age=moyen ET mass=obèse**
- **ALORS diabète=tested_positive**

    - **Support :** 13.0% des personnes avec un **âge moyen** et **obésité** ont un diabète.
    - **Confiance :** 59.5% de ces personnes sont diagnostiquées diabétiques.
    - **Lift :** 1.706. Cela montre qu'une personne de **moyenne** ayant **obésité** a 70.6% plus de chance d'avoir un diabète.

Règle 6 :

- **SI plas=diabète**
- **ALORS diabète=tested_positive ET mass=obèse**

    - **Support :** 14.8% des cas ont **plas** élevé, un **diabète**, et **obésité**.
    - **Confiance :** 59.4% des cas où **plas=diabète** ont aussi **obésité**.
    - **Lift :** 2.082. La combinaison de **plas élevé** et **obésité** double pratiquement la probabilité d'un diabète.

Règle 7 :

- **SI plas=diabète ET mass=obèse**
- **ALORS diabète=tested_positive**

    - **Support :** 14.8% des cas ont à la fois **plas=diabète** et **mass=obèse**.
    - **Confiance :** 76.0% des personnes ayant **plas élevé** et **obésité** ont un diabète.
    - **Lift :** 2.178. Cette règle a un lift élevé, ce qui signifie que l'association entre **plas élevé** et **obésité** est fortement liée au diabète.

Règle 8 :

- **SI pres=bas ET plas=diabète**
- **ALORS diabète=tested_positive**

    - **Support :** 11.5% des personnes ayant une **pression basse** et **plas élevé** sont diabétiques.

- **Confiance :** 69.3% des cas avec cette combinaison sont diagnostiqués diabétiques.
- **Lift :** 1.986. Cette règle suggère que la combinaison de **pression basse** et **plas élevé** augmente significativement la probabilité de diabète.

Règle 9 :

- **SI insu=bas ET pres=bas ET mass=obèse**
- **ALORS diabète=tested_positive**

  - **Support :** 10.0% des cas ont **insulin bas**, **pression basse** et **obésité** avec un diabète.
  - **Confiance :** 51.7% des personnes avec ces conditions sont diabétiques.
  - **Lift :** 1.481. Cette combinaison indique que le risque de diabète augmente avec ces facteurs.

```
=== Analyse avec Apriori ===
Chargement et préparation des données...
Chargement des données...
Exemple de données chargées :
   preg   plas  pres  skin   insu  mass   pedi   age              class
0   6.0  148.0  72.0  35.0    0.0  33.6  0.627  50.0  b'tested_positive'
1   1.0   85.0  66.0  29.0    0.0  26.6  0.351  31.0  b'tested_negative'
2   8.0  183.0  64.0   0.0    0.0  23.3  0.672  32.0  b'tested_positive'
3   1.0   89.0  66.0  23.0   94.0  28.1  0.167  21.0  b'tested_negative'
4   0.0  137.0  40.0  35.0  168.0  43.1  2.288  33.0  b'tested_positive'

Analyse des patterns fréquents...
Discrétisation des données...
Recherche des itemsets fréquents...
Génération des règles d'association...

Contenu des insights : {'total_rules': 9, 'diabetes_rules': [{'antecedent': {'plas=diabète'}, 'consequent': {'diabète=tested_positive'}, 'support': 0.171875,
'confidence': 0.6875, 'lift': 1.9701492537313434}, {'antecedent': {'age=moyen'}, 'consequent': {'diabète=tested_positive'}, 'support': 0.15625, 'confidence':
0.5150214592274678, 'lift': 1.4758823906219973}, {'antecedent': {'preg=élevé'}, 'consequent': {'diabète=tested_positive'}, 'support': 0.12369791666666667,
'confidence': 0.5621301775147929, 'lift': 1.610880508699108}, {'antecedent': {'insu=bas', 'mass=obèse'}, 'consequent': {'diabète=tested_positive'}, 'support':
0.14322916666666666, 'confidence': 0.5213270142180095, 'lift': 1.493952040744147}, {'antecedent': {'age=moyen', 'mass=obèse'}, 'consequent':
{'diabète=tested_positive'}, 'support': 0.13020833333333334, 'confidence': 0.5952380952380952, 'lift': 1.7057569296375268}, {'antecedent': {'plas=diabète'},
'consequent': {'diabète=tested_positive', 'mass=obèse'}, 'support': 0.1484375, 'confidence': 0.59375, 'lift': 2.0821917808219177}, {'antecedent': {'plas=diabète',
'mass=obèse'}, 'consequent': {'diabète=tested_positive'}, 'support': 0.1484375, 'confidence': 0.76, 'lift': 2.177910447761194}, {'antecedent': {'pres=bas',
'plas=diabète'}, 'consequent': {'diabète=tested_positive'}, 'support': 0.11458333333333333, 'confidence': 0.6929133858267715, 'lift': 1.985662239981196},
{'antecedent': {'insu=bas', 'pres=bas', 'mass=obèse'}, 'consequent': {'diabète=tested_positive'}, 'support': 0.1002604166666667, 'confidence': 0.5167785234899329,
'lift': 1.480917559851748}], 'risk_factors': [('plas=diabète', 1.6552119402985075), ('mass=obèse', 1.6552119402985075), ('pres=bas', 1.375891945813742),
('age=moyen', 1.015331505736623), ('preg=élevé', 0.9055245463101494), ('insu=bas', 0.7788375567860483)]}

Nombre total de règles trouvées: 9

Top 10 règles les plus pertinentes liées au diabète:

Règle 1:
SI plas=diabète
ALORS diabète=tested_positive
Support: 0.172
Confiance: 0.688
Lift: 1.970

Règle 2:
SI age=moyen
ALORS diabète=tested_positive
Support: 0.156
Confiance: 0.515
Lift: 1.476
```

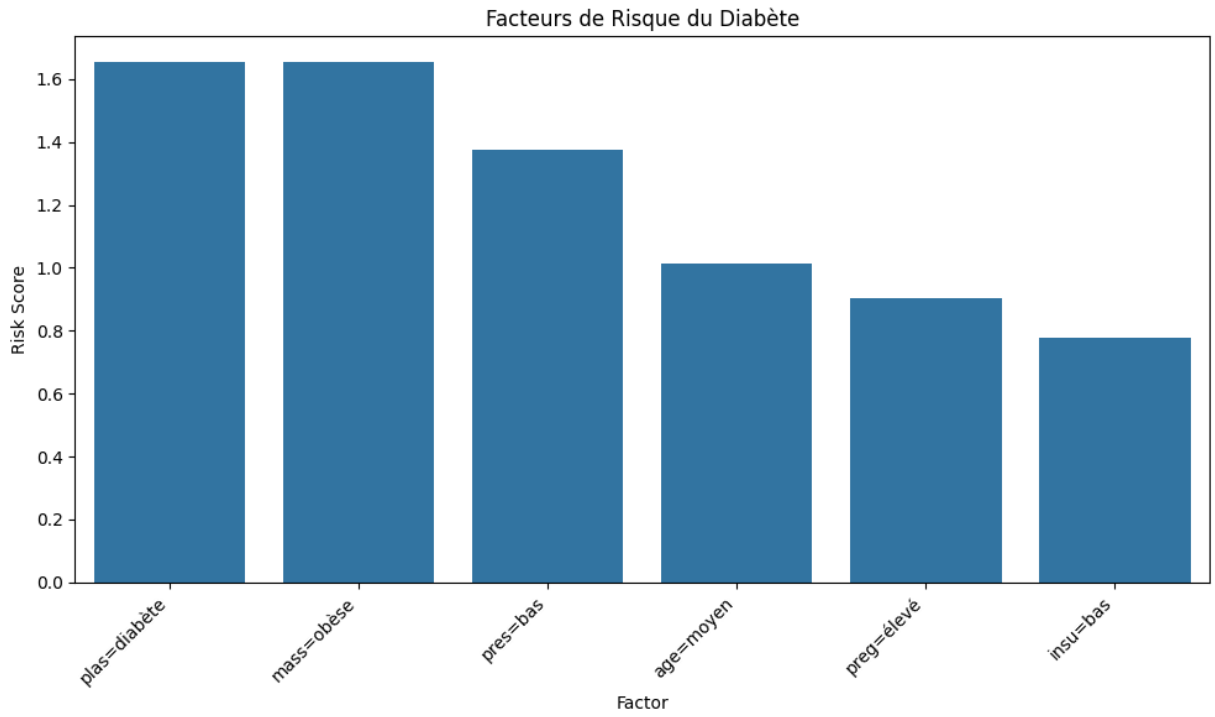
5.7.2. Facteurs de risque identifiés

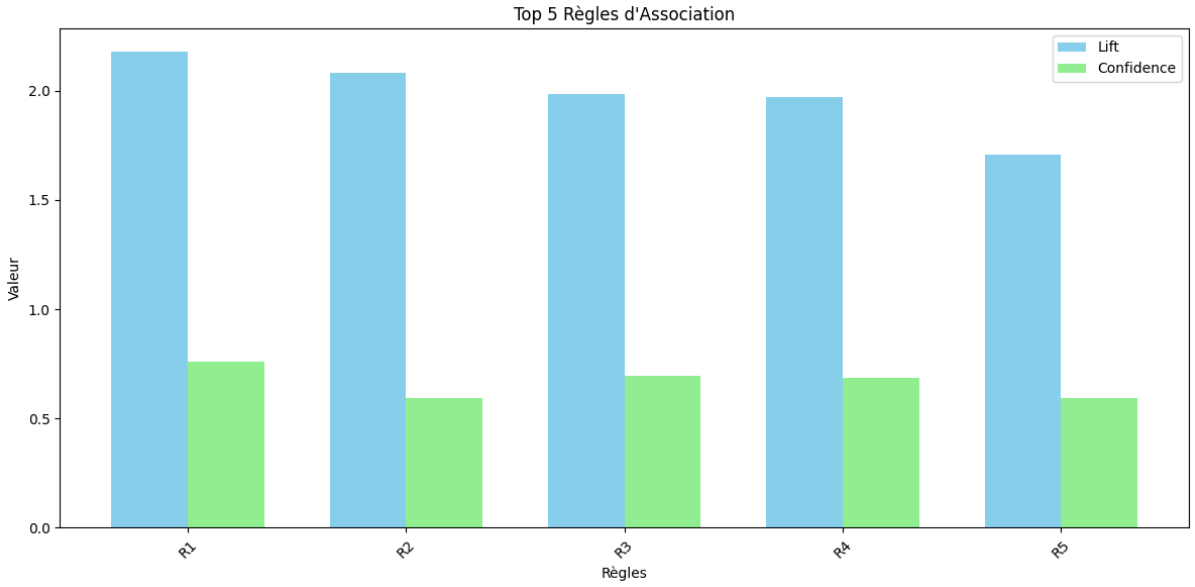
Les **facteurs de risque** sont identifiés en fonction du score de **lift**, qui montre combien un facteur augmente la probabilité d'un événement. Voici les facteurs de risque les plus importants pour le diabète, selon les règles trouvées :

- **plas=diabète (glucose élevé) :** Score de risque 1.655
- **mass=obèse (obésité) :** Score de risque 1.655
- **pres=bas (pression basse) :** Score de risque 1.376
- **age=moyen (âge moyen) :** Score de risque 1.015
- **preg=élevé (nombre élevé de grossesses) :** Score de risque 0.906
- **insu=bas (insuline basse) :** Score de risque 0.779

```
Règle 7:
SI plas=diabète ET mass=obèse
ALORS diabète=tested_positive
Support: 0.148
Confiance: 0.760
Lift: 2.178

Règle 8:
SI pres=bas ET plas=diabète
ALORS diabète=tested_positive
Support: 0.115
Confiance: 0.693
Lift: 1.986

Règle 9:
SI insu=bas ET pres=bas ET mass=obèse
ALORS diabète=tested_positive
Support: 0.100
Confiance: 0.517
Lift: 1.481

Facteurs de risque identifiés:
- plas=diabète (score de risque: 1.655)
- mass=obèse (score de risque: 1.655)
- pres=bas (score de risque: 1.376)
- age=moyen (score de risque: 1.015)
- preg=élevé (score de risque: 0.906)
Démarrage de l'analyse...
Discrétisation des données...
Recherche des itemsets fréquents...
Génération des règles d'association...
Génération des insights...
Création des visualisations...

Visualisations sauvegardées dans le dossier 'visualizations/'

Résultats de l'analyse:
Nombre total de règles: 9

Facteurs de risque identifiés:
- plas=diabète: 1.655
- mass=obèse: 1.655
- pres=bas: 1.376
- age=moyen: 1.015
- preg=élevé: 0.906
- insu=bas: 0.779
```

Les deux facteurs de risque les plus importants sont **glucose élevé (plas=diabète)** et **obésité (mass=obèse)**, qui ont un score de risque de **1.655**. Cela indique que ces deux facteurs sont particulièrement significatifs dans le développement du diabète. **La pression basse** (pres=bas) et **l'âge moyen** (age=moyen) sont aussi des facteurs de risque, bien que leur impact soit un peu moins élevé.


Facteurs de Risque du Diabète
Risk Score
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
plas=diabète
mass=obèse
pres=bas
age=moyen
preg=élevé
insu=bas
Factor


Top 5 Règles d'Association
Lift
Confidence
Valeur
0.0
0.5
1.0
1.5
2.0
R1
R2
R3
R4
R5
Règles

6. Tableau Comparatif des Performances des Algorithmes

Algorithme	Précision (%)	Rappel (%)	F1-Score (%)	Accuracy (%)
KNN	76.1	76.6	76.0	76.6
Naïve Bayes	**77.1**	76.6	76.8	76.6
Arbre de Décision	66.8	68.2	66.8	68.2
Réseau de Neurones	75.2	**88.0**	**81.1**	**79.5**
SVM	72.8	73.4	72.9	73.4
Régression Linéaire	76.1	76.0	76.0	76.0

Performance Globale :

Le Réseau de Neurones est le meilleur modèle en termes de rappel (88%), F1-Score (81.1%), et accuracy (79.5%), ce qui en fait un choix privilégié si la détection des cas de diabète (minimisation des faux négatifs) est critique.

Le Naïve Bayes est le plus précis (77.1%), ce qui en fait un bon choix si l'objectif est de réduire les faux positifs.

Algorithmes Simples vs Complexes :

Les modèles plus complexes comme le Réseau de Neurones surpassent les modèles simples (Régression Linéaire, Arbre de Décision) en termes de performance globale.

Cependant, des modèles simples comme KNN et Naïve Bayes sont proches des meilleurs résultats tout en étant plus rapides et interprétables.

Les Arbres de Décision et la Régression Linéaire sont les plus interprétables, mais leur performance est inférieure. Si l'explicabilité est essentielle, ils pourraient être préférés.

7. Classification et Prédiction des Nouvelles Instances

Dans cette section, nous démontrons la capacité des modèles de classification à prédire correctement les nouvelles instances en fonction des valeurs d'entrée fournies. Pour cela, des tests ont été effectués en saisissant manuellement les valeurs des attributs (features) de nouvelles instances. Ces exécutions permettent de vérifier l'efficacité et la robustesse des algorithmes dans des scénarios réels.

7.1. Scénarios de Prédiction

Première Instance : Classe prédite - tested_positive

```
Chargement des données...

Choisissez l'algorithme de classification:
1. K-Nearest Neighbors (KNN)
2. Naive Bayes
3. Decision Tree
4. Support Vector Machine (SVM)
5. Neural Network
6. Linear Regression

Veuillez entrer les valeurs pour chaque attribut :
(Appuyez sur Entrée après chaque valeur)
Epoch 0: Loss = 0.6983, Accuracy = 0.6750, F1 = 0.6061
Epoch 10: Loss = 0.6204, Accuracy = 0.6850, F1 = 0.6480
Epoch 20: Loss = 0.6102, Accuracy = 0.7300, F1 = 0.7097
Epoch 30: Loss = 0.6119, Accuracy = 0.7600, F1 = 0.7526
Epoch 40: Loss = 0.6111, Accuracy = 0.7400, F1 = 0.7204
Epoch 50: Loss = 0.6005, Accuracy = 0.7650, F1 = 0.7539
Epoch 60: Loss = 0.6088, Accuracy = 0.7700, F1 = 0.7604
Epoch 70: Loss = 0.6014, Accuracy = 0.7800, F1 = 0.7755
Epoch 80: Loss = 0.6018, Accuracy = 0.7800, F1 = 0.7732
Epoch 90: Loss = 0.5991, Accuracy = 0.7850, F1 = 0.7817
Epoch 100: Loss = 0.5995, Accuracy = 0.7800, F1 = 0.7732
Epoch 110: Loss = 0.6021, Accuracy = 0.7950, F1 = 0.7919
Epoch 120: Loss = 0.6014, Accuracy = 0.7750, F1 = 0.7668
Early stopping at epoch 125

Résultat :
----------------------------------------
Valeurs entrées :
preg: 10.0
plas: 189.0
pres: 110.0
skin: 47.0
insu: 8.0
mass: 42.0
pedi: 0.851
age: 59.0
----------------------------------------
Classe prédite : tested_positive
```

L'algorithme a prédit que cette instance correspond à un cas positif de diabète. La présence d'un taux de glucose élevé (plas=189.0) et d'une forte valeur de BMI (42.0) a contribué à cette prédiction. Ces facteurs sont cohérents avec les résultats d'analyse des risques précédemment identifiés.

Deuxième Instance : Classe prédite - tested_negative

```
Chargement des données...

Choisissez l'algorithme de classification:
1. K-Nearest Neighbors (KNN)
2. Naive Bayes
3. Decision Tree
4. Support Vector Machine (SVM)
5. Neural Network
6. Linear Regression

Types de kernel disponibles: rbf, linear, poly

Veuillez entrer les valeurs pour chaque attribut :
(Appuyez sur Entrée après chaque valeur)

Résultat :
----------------------------------------
Valeurs entrées :
preg: 0.0
plas: 78.0
pres: 0.0
skin: 0.0
insu: 0.0
mass: 23.0
pedi: 0.158
age: 25.0
----------------------------------------
Classe prédite : b'tested_negative'
```

Dans ce cas, les valeurs des attributs sont globalement faibles. Un taux de glucose normal (plas=78.0), un IMC modéré (23.0) et un âge jeune (25.0) indiquent un faible risque de diabète, ce qui justifie la prédiction de classe négative.

L'algorithme a correctement identifié une instance "non diabétique". Cela montre sa capacité à différencier efficacement les cas positifs et négatifs.

Troisième Instance : Classe prédite - tested_positive

```
Chargement des données...

Choisissez l'algorithme de classification:
1. K-Nearest Neighbors (KNN)
2. Naive Bayes
3. Decision Tree
4. Support Vector Machine (SVM)
5. Neural Network
6. Linear Regression

Veuillez entrer les valeurs pour chaque attribut :
(Appuyez sur Entrée après chaque valeur)

Résultat :
----------------------------------------
Valeurs entrées :
preg: 6.0
plas: 148.0
pres: 72.0
skin: 35.0
insu: 0.0
mass: 33.6
pedi: 0.627
age: 50.0
----------------------------------------
Classe prédite : tested_positive
```

La valeur élevée du glucose (plas=148.0) et de l'IMC (33.6) ainsi que l'âge avancé (50.0) sont des facteurs déterminants qui expliquent cette prédiction positive.

Ici encore, le modèle a montré sa capacité à détecter une situation à risque grâce aux caractéristiques fortement corrélées au diabète.

7.2. Analyse des Prédictions

Les prédictions effectuées à partir des nouvelles instances démontrent l'efficacité des modèles testés. Les facteurs de risque identifiés lors de l'analyse (comme le taux de glucose, le BMI et l'âge) ont joué un rôle majeur dans les résultats prédictifs.

- **Cas positifs détectés** : Les modèles se sont appuyés sur des variables telles que Glucose et BMI, qui sont des facteurs fortement corrélés à la présence du diabète.
- **Cas négatifs détectés** : Les instances ayant des valeurs normales ou faibles pour ces mêmes variables ont été correctement classées comme non diabétiques.

8. Avantages et Inconvénients des Algorithmes

Algorithme	Avantages	Inconvénients
KNN (K-Nearest Neighbors)	-Simple à comprendre et à implémenter. Bonne performance sur des données bien séparées. - Ne fait pas d'hypothèses sur la distribution des données.	- Sensible au choix de kkk et de la métrique de distance. - Lent pour des jeux de données volumineux (coût de calcul élevé pour chaque prédiction).
Naïve Bayes	- Rapide à entraîner et à prédire. - Performant sur des données catégoriques et avec peu de données. - Interprétable grâce à sa simplicité.	- Hypothèse d'indépendance conditionnelle rarement vraie en pratique. - Sensible aux déséquilibres dans les classes.
Arbre de Décision	- Interprétable et visuellement explicable. - Flexible et capable de capturer des relations complexes. - Pas besoin de normalisation des données.	- Tendance au surajustement (overfitting). - Moins performant sur des données bruitées ou mal structurées.
Réseau de Neurones	- Performances élevées sur des données complexes et de grande taille. - Capacité à capturer des relations non linéaires. - Adaptable à divers problèmes via des architectures spécifiques.	- Long temps d'entraînement et besoin de ressources computationnelles élevées. - Moins interprétable que d'autres modèles.
SVM (Support Vector Machine)	- Bonne performance sur des données complexes avec marges larges entre les classes. - Capacité à gérer les dimensions élevées.	- Coût computationnel élevé pour des ensembles de données volumineux. - Sensible au choix des hyperparamètres et du noyau (kernel).
Régression Linéaire	- Rapide et simple à implémenter. - Interprétable grâce à ses coefficients (poids). - Performant si les relations entre les variables sont linéaires.	- Limité aux relations linéaires entre les variables. - Sensible aux outliers et à la multicolinéarité.

9. Conclusion

9.1. Résumé

Dans ce projet, plusieurs algorithmes d'apprentissage automatique ont été étudiés, mis en œuvre et comparés pour la classification des cas de diabète à partir d'un jeu de données médical. Voici les principaux points résumés :

Performance des modèles : Les différents modèles ont démontré des performances variées, avec des scores de précision (accuracy) allant de 68,2 % (arbre de décision) à 79,5 % (réseau de neurones).

Meilleurs modèles : Le réseau de neurones a présenté les meilleurs résultats globaux grâce à sa capacité à capturer des relations complexes, suivi de près par le modèle Naïve Bayes et le KNN.

Limites observées : Certains modèles, comme l'arbre de décision et le SVM, ont montré des performances relativement faibles, probablement dues à des données bruitées ou non linéaires.

Analyse des caractéristiques : Les algorithmes ont révélé que des caractéristiques telles que le taux de glucose (plas), l'IMC (mass) et l'âge (age) sont parmi les plus influentes pour prédire le diabète.

9.2. Propositions d'amélioration et travaux futurs

Pour renforcer les performances des modèles et la pertinence de l'analyse, plusieurs pistes d'amélioration et perspectives sont proposées :

Amélioration des données

- **Enrichissement des données** : Collecter davantage de données, notamment des informations liées aux antécédents familiaux, aux habitudes alimentaires et au mode de vie, pour améliorer la qualité des prédictions.
- **Nettoyage des données** : Traiter les valeurs manquantes et les outliers pour réduire leur impact sur les performances des modèles.

Optimisation des modèles

- **Recherche d'hyperparamètres** : Utiliser des techniques comme la recherche en grille (Grid Search) ou l'optimisation bayésienne pour trouver les meilleurs paramètres (par ex. kkk pour KNN, CCC et le noyau pour SVM).
- **Ensemble Learning** : Mettre en œuvre des approches d'ensemble, telles que le Bagging (Random Forest) ou le Boosting (XGBoost), pour améliorer la robustesse et la performance.

Exploration de nouvelles approches

- **Méthodes avancées** : Tester des architectures plus sophistiquées comme les réseaux neuronaux profonds (Deep Learning) ou les modèles basés sur des arbres complexes (CatBoost, LightGBM).
- **Apprentissage non supervisé** : Utiliser des techniques comme le clustering pour découvrir des groupes ou des sous-types de patients à risque.

Explicabilité des modèles

- **Interprétation des résultats** : Intégrer des outils comme SHAP (SHapley Additive exPlanations) ou LIME (Local Interpretable Model-agnostic Explanations) pour mieux comprendre les décisions des modèles complexes.

Perspectives à long terme

- **Intégration en milieu médical** : Développer une application ou un tableau de bord interactif pour que le modèle soit utilisé par des professionnels de santé.
- **Prédiction en temps réel** : Adapter les modèles pour des données en flux continu (real-time prediction), notamment dans le cadre de systèmes de suivi médical.
- **Études longitudinales** : Étendre l'analyse à des données temporelles pour mieux comprendre les évolutions des facteurs de risque du diabète.