

École Supérieure de Technologie - DAKHLA

Module : **Machine Learning**

Professeur : **Pr. DIB**



TP1 Apprentissage Supervisé Machine Learning

Un modèle de Machine Learning Supervisé

Gaussian Naïve Bayes

Réalisé par :

Lucien YAOGO

Filière : Cybersécurité & IA

Année universitaire : 2025-2026

Table des matières

| | |
|--|----|
| Table des Figure | 3 |
| Objectif de ce TP..... | 4 |
| Introduction..... | 5 |
| QUESTION 1..... | 6 |
| a) <i>Quel est le type d'apprentissage Utilisé ?</i> | 6 |
| b) <i>S'agit-il d'un problème de régression ou de classification ?</i> | 6 |
| c) <i>Justifiez votre réponse</i> | 6 |
| QUESTION 2..... | 6 |
| a) <i>Quel type de Naïve Baye est adapté à ce problème ?</i> | 6 |
| b) <i>Justifiez votre choix</i> | 6 |
| QUESTION 3..... | 6 |
| a) <i>Importez pandas et Chargez le data</i> | 6 |
| b) <i>Analysez le data</i> | 7 |
| QUESTION 4..... | 7 |
| QUESTION 5..... | 8 |
| a) <i>Séparez les Variables explicatives et la variable cible.</i> | 8 |
| b) <i>Importez le module nécessaire et Divisez les données en ensemble d'entraînement et de test</i> 8 | |
| QUESTION 6..... | 9 |
| a) <i>Importez le module nécessaire et Initialisez un modèle Gaussian Naive Bayes.</i> | 9 |
| b) <i>Entraînez le modèle</i> | 9 |
| QUESTION 7 | 9 |
| a) <i>Effectuez les prédictions</i> | 9 |
| b) <i>Importez les Modules nécessaires pour Calculer :</i> | 9 |
| QUESTION 8..... | 10 |
| a) <i>Effectuez les prédictions</i> | 10 |
| b) <i>Interprétez le Résultat.</i> | 10 |
| Conclusion | 11 |

Table des Figures

| | |
|--|----|
| Figure 1 : Chargement de data | 7 |
| Figure 2 : Résultat du chargement de data | 7 |
| Figure 3 : Analyse de data | 7 |
| Figure 4 : Séparation des Variables | 8 |
| Figure 5 : Les résultats de la séparation des Variables | 8 |
| Figure 6 : Entraînement des données | 8 |
| Figure 7 : Résultats des entraînements | 8 |
| Figure 8 : Initiation du modèle | 9 |
| Figure 9 : Entraînement du modèle | 9 |
| Figure 10 : Teste de Performance | 9 |
| Figure 11 : Résultat des testes | 10 |
| Figure 12 : La prédiction de l'état d'un nouveau patient | 10 |
| Figure 13 : Résultat du Patient | 10 |

Objectif de ce TP

Le but de ce TP est d'implémenter un modèle de classification Gaussian Naive Bayes afin de prédire l'état d'un patient à partir de paramètres cliniques, tout en comprenant les principes fondamentaux de la classification supervisée.

Introduction

Avec l'essor du numérique et de l'intelligence artificielle, le Machine Learning joue un rôle clé dans l'analyse des données, notamment dans le domaine médical. Les techniques d'apprentissage automatique permettent de concevoir des modèles capables d'analyser des données cliniques et de produire des prédictions utiles à la prise de décision.

Ce travail pratique, réalisé dans le cadre du module Machine Learning, porte sur l'apprentissage supervisé à travers l'algorithme Naïve Bayes. Il s'appuie sur un jeu de données médicales comprenant l'âge, la pression artérielle, le taux de glucose et l'IMC, afin de classer les patients selon leur état de santé.

L'objectif de ce TP est de mettre en œuvre un modèle de classification Naive Bayes et d'évaluer ses performances à l'aide de métriques telles que l'accuracy, la matrice de confusion et le rapport de classification, tout en analysant les résultats obtenus et leurs limites.

QUESTION 1

a) *Quel est le type d'apprentissage Utilisé ?*

==> Le type d'Apprentissage utilisé est l'Apprentissage Supervisé

b) *S'agit-il d'un problème de régression ou de classification ?*

==> Il s'agit d'un problème de Classification

c) *Justifiez votre réponse*

==> Les Variables "Classe" prennent des valeurs 'Discrètes (0 / 1)'. L'objectif est de prédire une catégorie (Non Malade / Risque de Maladie). Les données sont étiquetées.

QUESTION 2

a) *Quel type de Naïve Baye est adapté à ce problème ?*

==> Le type de Naïve Baye adapté est le Gaussian Naive Bayes

b) *Justifiez votre choix*

==> Ce type de Naive Bayes est adapté car les variables explicatives sont continues (age, pression, glucose, imc)

- Les variables explicatives sont numériques continues.
- Elles peuvent être modélisées par une loi normale.
- La variable cible est catégorielle.
- Le jeu de données est petit.

QUESTION 3

a) *Importez pandas et Chargez le data*

```
data = {  
"age": [25, 45, 50, 35, 60, 40, 55, 30, 48, 52],  
"pression": [120, 140, 150, 130, 160, 135, 155, 125, 145, 150],  
"glucose": [90, 160, 170, 110, 180, 120, 175, 95, 165, 168],  
"imc": [22, 28, 30, 24, 32, 26, 31, 23, 29, 30],  
"classe": [0, 1, 1, 0, 1, 0, 1, 0, 1, 1]}
```

```
[38]: # ==> Importation de pandas
import pandas as pd

# ==> Chargement du Data
df = pd.DataFrame(data)
df
```

Figure 1 : Chargement de data

```
[38]:
```

| | age | pression | glucose | imc | classe |
|---|-----|----------|---------|-----|--------|
| 0 | 25 | 120 | 90 | 22 | 0 |
| 1 | 45 | 140 | 160 | 28 | 1 |
| 2 | 50 | 150 | 170 | 30 | 1 |
| 3 | 35 | 130 | 110 | 24 | 0 |
| 4 | 60 | 160 | 180 | 32 | 1 |
| 5 | 40 | 135 | 120 | 26 | 0 |
| 6 | 55 | 155 | 175 | 31 | 1 |
| 7 | 30 | 125 | 95 | 23 | 0 |
| 8 | 48 | 145 | 165 | 29 | 1 |
| 9 | 52 | 150 | 168 | 30 | 1 |

Figure 2 : Résultat du chargement de data

b) Analysez le data

```
[ ]: # ==> Afficher Les 5 premières Lignes de notre Data
df.head(5) # Ou df.head()

# ==> Afficher Les 5 dernières Lignes de notre Data
df.tail(5) # Ou df.tail()

# ==> Sommer Les éléments de chaque colonne de notre Data
df.sum()

# ==> Compter Les nombre de Lignes de notre Data
df.count()

# ==> Afficher Les informations de notre Data
df.info()

# ==> Verifier ou comptes s'il y avait des parties vides dans notre Data
df.isnull() # ou df.isnull().sum()

# ==> Supprimer Les parties vides ou null de notre Data
df.dropna().sum()
```

Figure 3 : Analyse de data

QUESTION 4

Créez un jeu de données médical fictif contenant :

==> Jeu de données médical fictif (déjà fourni et utilisé ci-dessus) :

- 4 variables explicatives : âge, pression, glucose, IMC
- 1 variable cible : classe (0 / 1)

QUESTION 5

a) Séparez les Variables explicatives et la variable cible.

=> Séparation des variables :

- 4 Variables explicatives (X)
- 1 Variable cible (y)

```
[40]: X = df[["age", "pression", "glucose", "imc"]] # ==> Variables explicatives
      y = df["classe"] # ==> Variables cibles
      |
      print("Les variables explicatives (X) : \n", X)
      print("La variable cible (y) : \n", y)
```

Figure 4 : Séparation des Variables

```
Les variables explicatives (X) :
age  pression  glucose  imc
0    25      120      90    22
1    45      140      160   28
2    50      150      170   30
3    35      130      110   24
4    60      160      180   32
5    40      135      120   26
6    55      155      175   31
7    30      125      95    23
8    48      145      165   29
9    52      150      168   30

La variable cible (y) :
0    0
1    1
2    1
3    0
4    1
5    0
6    1
7    0
8    1
9    1
Name: classe, dtype: int64
```

Figure 5 : Les résultats de la séparation des Variables

b) Importez le module nécessaire et Divisez les données en ensemble d'entraînement et de test

```
[41]: # ==> Importation du module nécessaire
      from sklearn.model_selection import train_test_split

      # ==> Division des données
      X_train, X_test, y_train, y_test = train_test_split(
          X, y, test_size=0.3, random_state=43)

      print(X_train)
```

Figure 6 : Division des données en ensemble d'entraînement et de test

```
age  pression  glucose  imc
8    48      145      165   29
2    50      150      170   30
5    40      135      120   26
1    45      140      160   28
7    30      125      95    23
0    25      120      90    22
4    60      160      180   32
```

Figure 7 : Résultats des entraînements

QUESTION 6

a) *Importez le module nécessaire et Initialisez un modèle Gaussian Naive Bayes.*

```
[42]: # ==> Importez Le module nécessaire
      from sklearn.naive_bayes import GaussianNB

      # ==> Initialisez un modèle Gaussian Naive Bayes.
      model = GaussianNB()
```

Figure 8 : Initiation du modèle

b) *Entraînez le modèle*

==> Entraînement du model

```
[33]: model.fit(X_train, y_train)
```

[33]:

- GaussianNB ⓘ ?
- Parameters

Figure 9 : Entraînement du modèle

QUESTION 7

a) *Effectuez les prédictions*

==> Les prédictions de l'entraînement

```
[50]: y_pred = model.predict(X_test)
      print("Les predictions de l'entraînement : \n", y_pred)
```

Les predictions de l'entraînement :
[0 1 1]

b) *Importez les Modules nécessaires pour Calculer :*

==> Importons respectivement les modules nécessaires pour les calculs :

- L'accuracy
- La matrice de Confusion
- Le rapport de classification

```
[43]: # ==> Importez Le Module (respectivement: accuracy, La matrice de Confusion, Le rapport de classification)
      from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

      accuracy = accuracy_score(y_test, y_pred)
      matrice_confusion = confusion_matrix(y_test, y_pred)
      rapport_classification = classification_report(y_test, y_pred)

      print("Accuracy : \n", accuracy)
      print("La matrice de Confusion : \n", matrice_confusion)
      print("Le rapport de classification : \n", rapport_classification)
```

Figure 10 : Teste de Performance

```

Accuracy :
1.00
La matrice de Confusion :
[[1 0]
 [0 2]]
Le rapport de classification :
      precision    recall  f1-score   support

     0       1.00      1.00      1.00         1
     1       1.00      1.00      1.00         2

 accuracy          1.00         3
 macro avg          1.00         3
 weighted avg          1.00         3

```

Figure 11 : Résultat des testes

QUESTION 8

Un patient a les caractéristiques suivantes :

- âge = 50, pression = 145, glucose = 160, IMC = 28

a) Effectuez les prédictions

```

[53]: new_patient = pd.DataFrame(
      [{"age", "145", "160", "26"}],
      columns=["age", "pression", "glucose", "imc"])

prediction = model.predict(new_patient)

print("La prediction de l'état du patient : ", prediction)

if (prediction == 0):
    print(" Non malade ")
else:
    print(" Malade ")

```

Figure 12 : La prédiction de l'état d'un nouveau patient

```

La prediction de l'état du patient : [1]
Malade

```

Figure 13 : Résultat du Patient

b) Interprétez le Résultat.

==> Interprétation :

La prediction de l'état du patient : [1]

- Le modèle estime que le patient présente un risque médical

Conclusion

Ce travail pratique a permis de mettre en œuvre un modèle de classification supervisée Gaussian Naive Bayes pour prédire l'état de santé d'un patient à partir de données biomédicales. Il a facilité la compréhension des principes de l'apprentissage supervisé ainsi que des étapes clés d'un projet de Machine Learning, de l'entraînement du modèle à l'évaluation des performances.

Les résultats obtenus sont satisfaisants, tout en mettant en évidence les limites liées à la taille réduite du jeu de données. Ce TP constitue ainsi une base solide pour l'apprentissage des méthodes de classification et leur application dans le domaine médical.