

**Présentation de TP du Groupe 1**  
**Niveau : Master Cybersécurité**  
**Thématique :**

**Diagnostic medical**

Présenté par:

**ADOUM AHMAT GRENE**  
**BOKHIT ABDOULAYE DIGUI**  
**LAGRE GABBA BERTRAND**  
**MADJINGUESSOUM BRICE**  
Professeur : **MUSA BOUKAR**

# Plan du travail

- ① Introduction
- ② Visualisation des données
- ③ Présentation du modèle
- ④ Évaluation du modèle
- ⑤ Exportation du modèle
- ⑥ Déploiement de la solution
- ⑦ Conclusion

## Introduction

La **classification** est un processus de catégorisation d'un ensemble donné de données en classes.

Un **arbre de décision** ou **decision tree** est un algorithme de Machine Learning qui utilise une structure arborescente pour prendre des décisions ou faire des prédictions.

## Visualisation des données

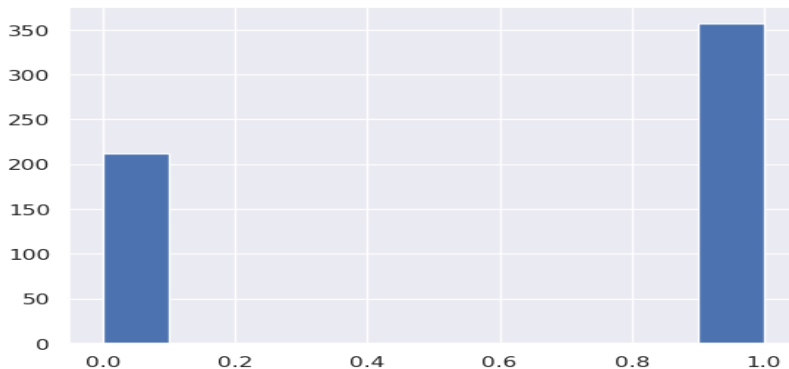


Figure – Proportion de chaque attribut du target

## Visualisation des données

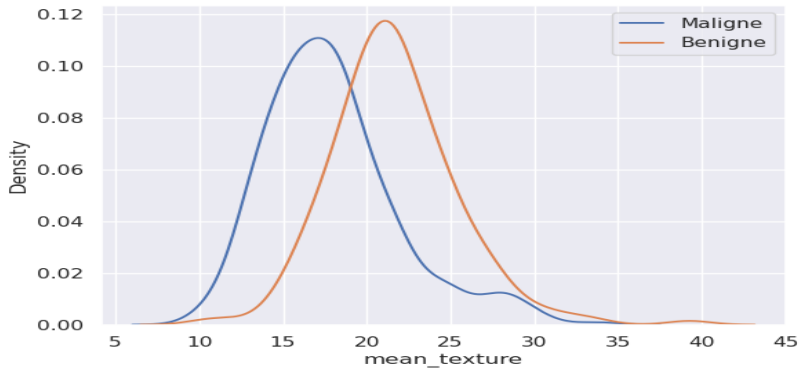


Figure – Impact du featire meantexture sur le target

## Présentation du modèle

### Decision Tree Algorithm

```
: classifier_DT = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
: classifier_DT.fit(X_train, Y_train)
```

```
: DecisionTreeClassifier
: DecisionTreeClassifier(criterion='entropy', random_state=0)
```

```
: print("Le score est :")
: round(classifier_DT.score(X_test, Y_test),2)
```

Le score est :

```
: 0.9
```

Figure – Arbre de décision avec score obtenu

## Évaluation du modèle

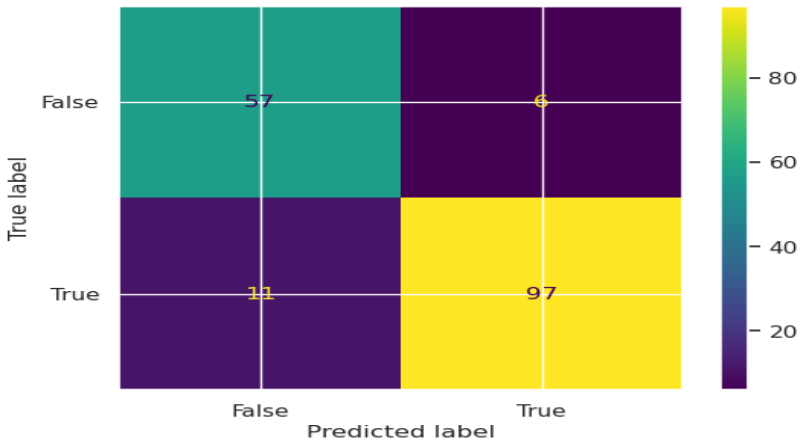


Figure – Matrice de confusion

## Évaluation du modèle

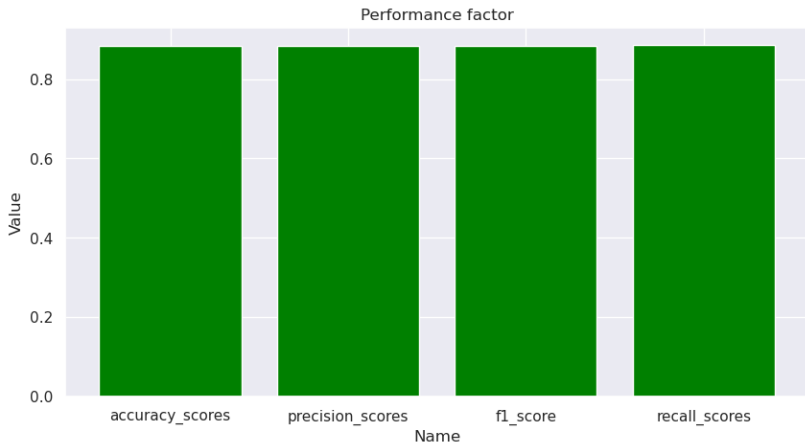


Figure – Evaluation du modèle



## Évaluation du modèle

Vrai Positive (VP): 97

Vrai Negative (VN): 57

Faux Positive (FP): 6

Faux Negative (FN): 11

Formule de calcul:

Accuracy:  $(VP + VN) / (VP + VN + FP + FN)$

Precision:  $VP / (VP + FP)$

Recall:  $VP / (VP + FN)$

F1-score:  $2 * (Precision * Recall) / (Precision + Recall)$

Résultats:

Accuracy: 0.9

Precision: 0.94

Recall: 0.9

F1-score: 0.92

Figure – Evaluation du modèle

## Exportation du modèle

```
|: joblib.dump(classifier_DT, "Breast_cancer_data.sav")  
|: ['Breast_cancer_data.sav']
```

Figure – Exportation du modèle

## Déploiement de la solution

Nous avons utilisé streamlit en important les bibliothèques suivantes :

```
import joblib
import pandas as pd
import numpy as np
import streamlit as st
import seaborn as sns
import matplotlib.pyplot as plt
from streamlit_option_menu import option_menu
```

Figure – Déploiement de la solution

## Conclusion

Le but ultime de cette réalisation consiste à développer une solution automatique qui servira à diagnostiquer des affections médicales basées sur les symptômes du patient et son historique médical. Nous avons obtenue un Mean Square Error (MSE) de 0.1. Ce resultat peut être amélioré et nous sommes ouverts à toutes critiques, remarques et suggestions.