CERGY PARIS UNIVERSITÉ

DATA MINING

Professeur :

Jen Tao YUAN

Binôme :

Mohamed II BAYO

Benoit FAGOT

Sommaire

- ★ Présentation du jeu de données
- ★ Analyse de données
- ★ Choix Algorithme
  - ○ Arbre de décision
  - ○ KNN
- ★ Conclusion

Présentation des données - Wine Dataset

- Plateforme : UC Irvine Machine Learning Repository
- Source : The Institute of Pharmaceutical and Food Analysis and Technologies
- Objets : vins provenant de 3 cultivars différents, 13 attributs (valeurs continues)
- Problème posé : **classification**
- Attributs déterministes ?

Distribution des classes

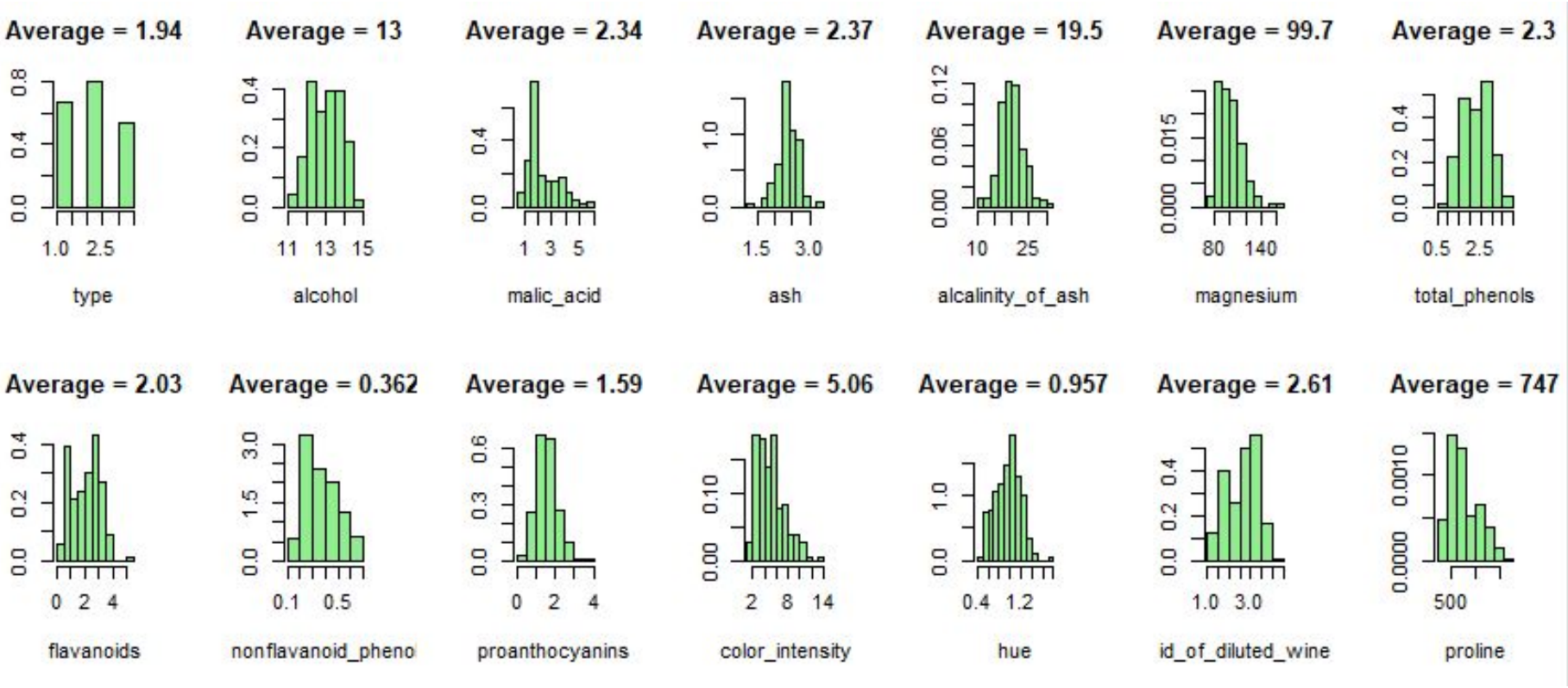
Pour un total de 178 objets :

- **classe 1 :** 59 instances soit 33%
- **classe 2 :** 71 instances soit 40%
- **classe 3 :** 48 instances soit 27%

Les attributs :

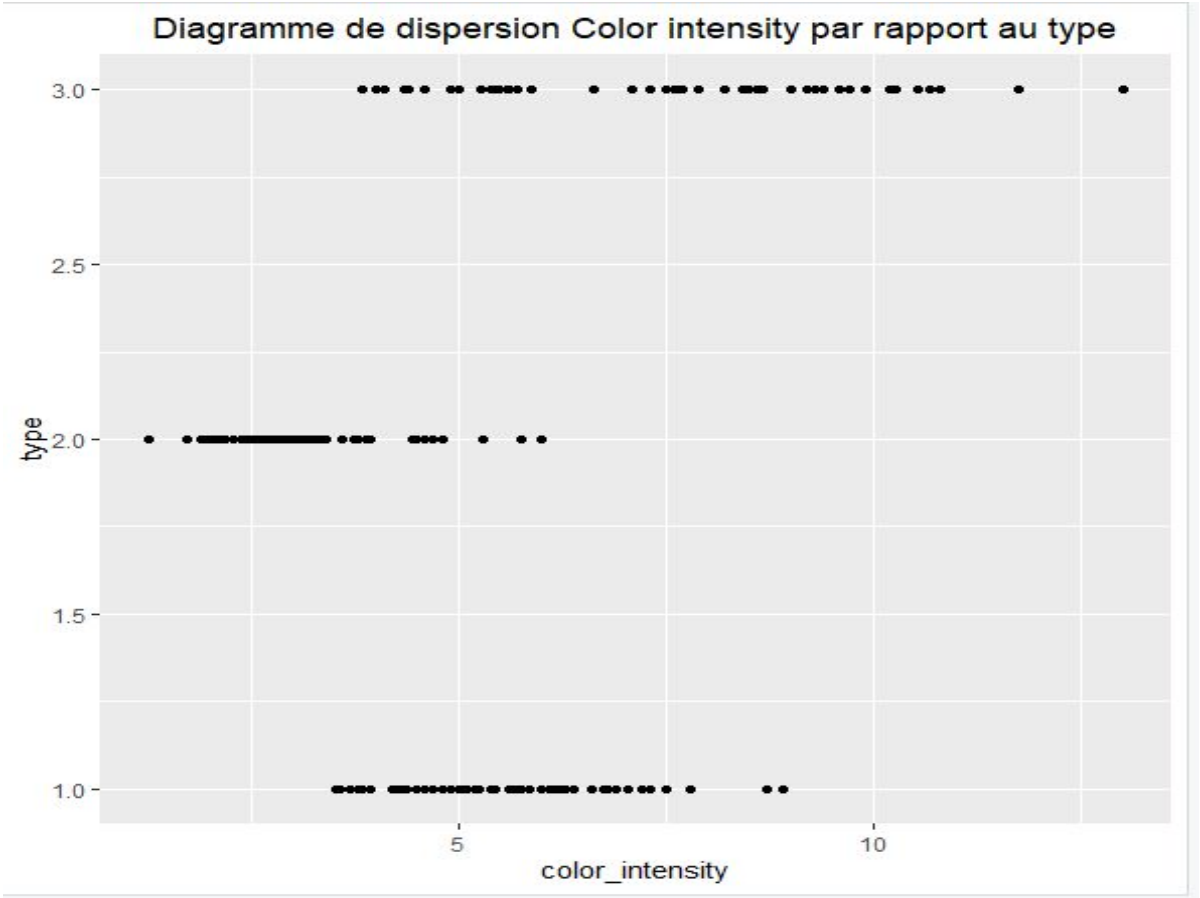
Attributs	Valeurs Distinctes	Valeurs Manquantes
type (classe)	3	0
Alcohol	126	0
Malic acid	133	0
Ash	79	0
Alcalinity of ash	63	0
Magnesium	53	0
Total phenols	97	0
Flavanoids	132	0
Nonflavanoid phenols	39	0
Proanthocyanins	101	0
Color intensity	132	0
Hue	78	0
id of diluted wines	122	0
proline	121	0

Les attributs

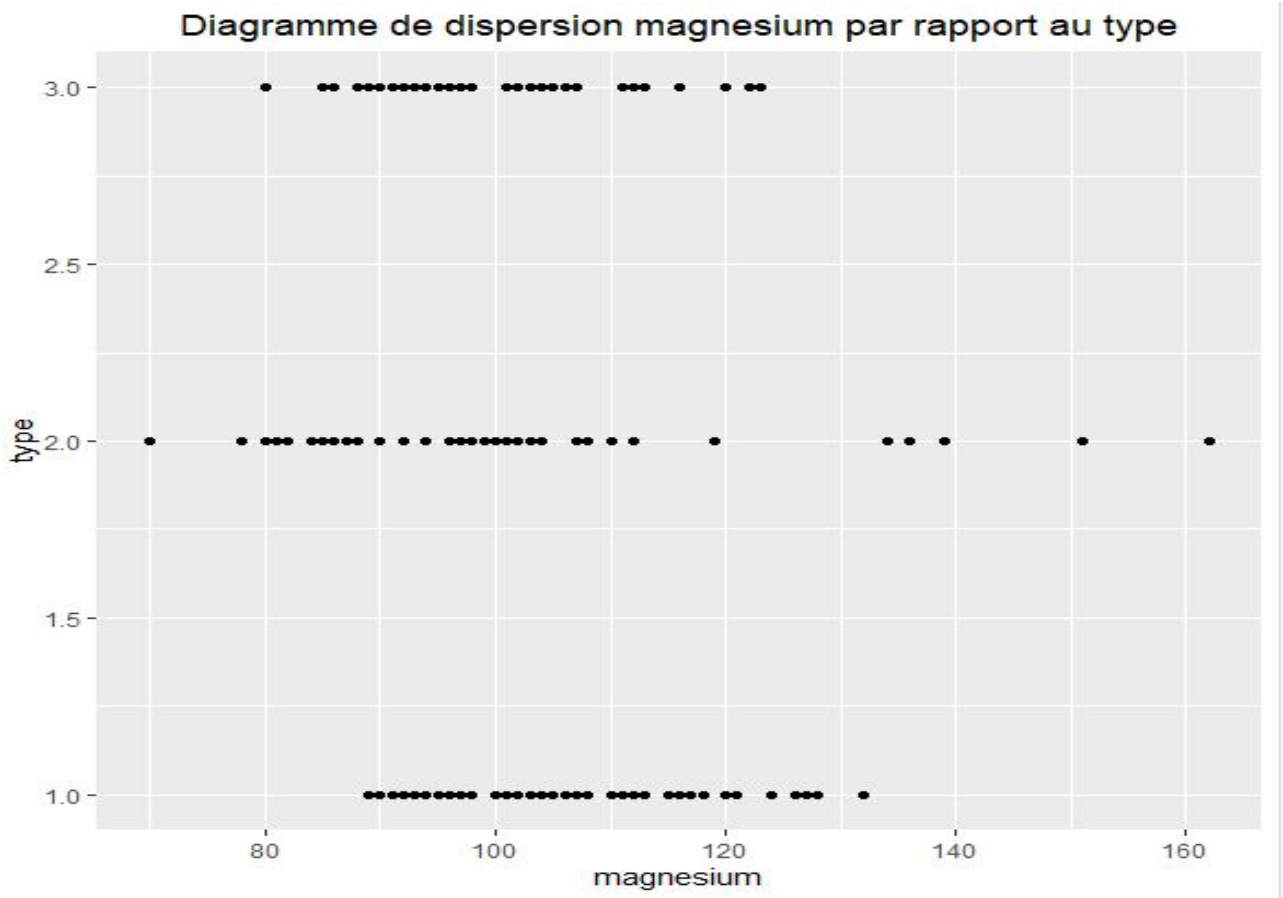


Analyse des données

Analyse des données - attribut déterministe



Analyse des données - attribut non déterministe



Attributs retenus

- Color intensity
- Flavonoids
- ID_of_diluted_wine
- Proline

4 / 13 attributs utilisés pour les classificateurs

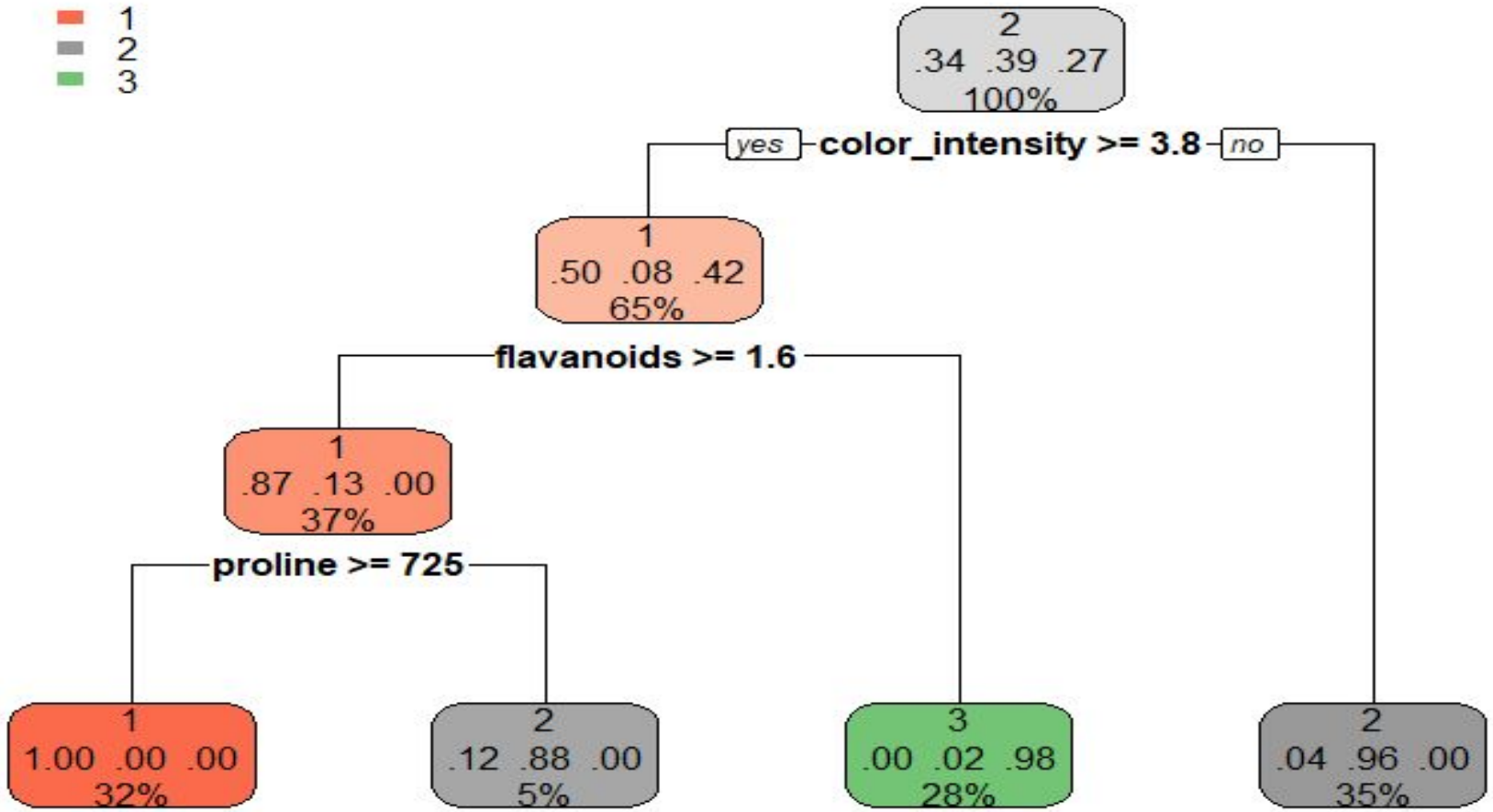
CHOIX DES ALGORITHMES

- ★ Arbre de décision
- ★ KNN (K-Nearest Neighbors)

Arbre de Décision

Arbre De Décision

- Règles de décisions : attributs déterministes
- Librairie **rpart** :
  - découpe récursivement le jeu de données
  - plus grande réduction possible de l'hétérogénéité des classes
  - décision optimale locale ➜ arbre non globalement optimal
- Training set / Testing set ratio : **0.7/0.3**
- Seuil de confiance minimale : **80%**


1
2
3
2
.34 .39 .27
100%
yes
color_intensity >= 3.8
no
1
.50 .08 .42
65%
flavanoids >= 1.6
1
.87 .13 .00
37%
proline >= 725
1
1.00 .00 .00
32%
2
.12 .88 .00
5%
3
.00 .02 .98
28%
2
.04 .96 .00
35%

Précision du classificateur Arbre de décision

On obtient une précision globale de 95%.

- Type 1 : Précision 100%, Rappel 100%
- Type 2 : Précision 94%, Rappel 94%
- Type 3 : Précision 93%, Rappel 93%

KNN (K-Nearest Neighbors)

KNN(K-Nearest Neighbors)

Pourquoi KNN ?

- Identifie le nombre k d'observations les plus proches de l'échantillon de test
- À partir de cet ensemble de k- voisins, la règle de la majorité est utilisée pour prédire la classe. Par exemple :
  - Si K = 5
  - Type des vins les plus proches voisins = {2, 2, 1, 1, 1} => échantillon de type 1

KNN(K-Nearest Neighbors)

Pour entraîner notre modèle nous avons utilisé :

- La Bibliothèque **caret**
- Méthode **trainControl() :**

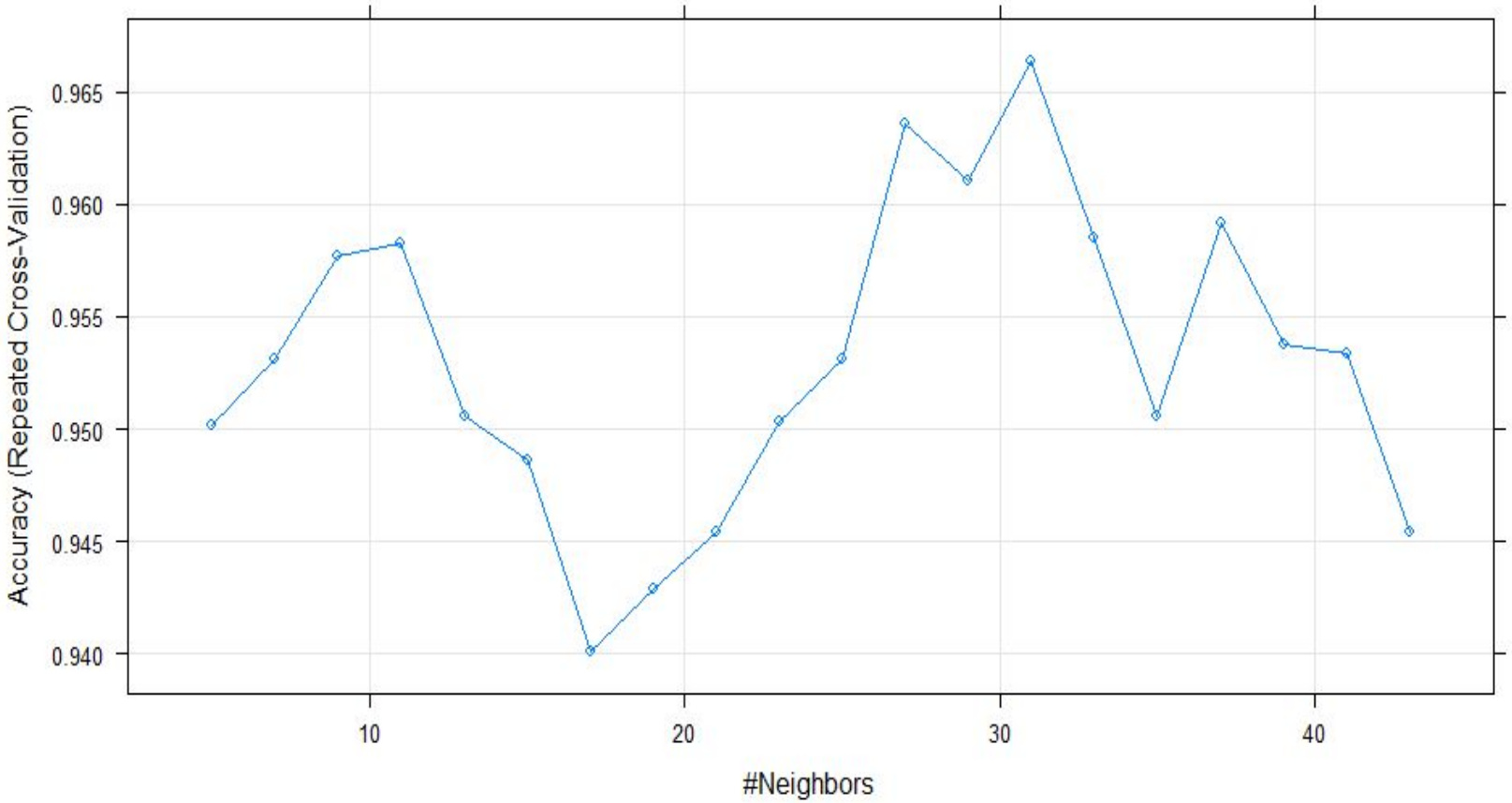
```
trainControl(
    method=”repeatedcv”,
    number = 10,
    repeats = 3
)
```

- Méthode **train() :**

```
train(
    type ~.,
    data = df2_train, method = "knn",
    trControl = trainCtrl,
    preProcess = c("center", "scale"),
    tuneLength = tunel
)
```

Notre modèle de formation choisit k = 31 comme valeur finale.

Choix du K optimal



KNN(K-Nearest Neighbors) - Prédiction

Nous avons utiliser la méthode Predict() pour faire de la prédiction.

- **predict(model_knn, newdata = test_df2)**

On a obtenu une précision de **94.23%** pour l'ensemble de test.

Conclusion

Conclusion

- **Jeu de données :** attributs chimique de vins issus de cultivars différents
- **Problème posé :** vecteurs de caractéristiques ➜ classification des vins
- **Observation :** 4 attributs potentiellement déterministes
- **Modèles de classification testés :** Arbre de décision, K-nn
- **Résultats :**
  - ➢ Arbre de décision **95%**
  - ➢ K-nn **94%**