MATHÉMATIQUES ET INFORMATIQUE
Sciences
Université de Paris

Master 1 Informatique
Rapport TP DATA1

TP2-classification supervisé

Régression linéaire-KNN-bayésien naïf

Houcine FORLOUL
Ridha TIGOULMAMINE
Sara BAICHE
Rachda BOUGDOUR

Année universitaire : 2020 – 2021

PARTIE I : PROSTATE

1. Description :

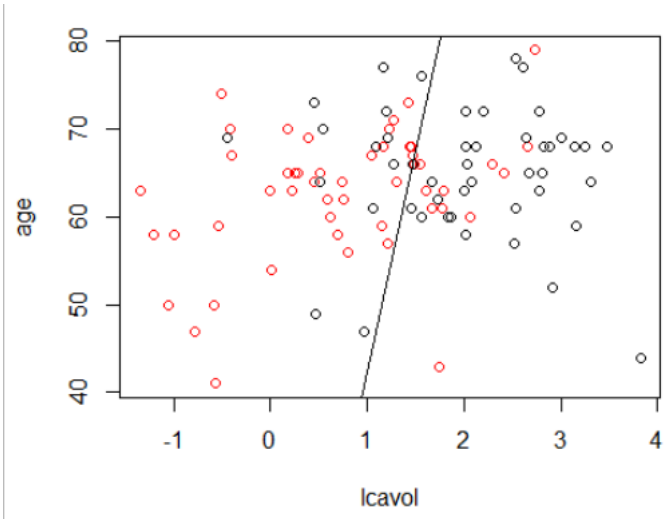
Données pour examiner la corrélation entre le niveau d'antigène prostatique spécifique et un certain nombre de mesures cliniques chez les hommes qui étaient sur le point de subir une prostatectomie radicale.

Notre Dataset avec 97 observations et 10 variables.

- lcavol : log volume de cancer
- lweight : log poids de la prostate
- age
- lbph : log de la quantité d'hyperplasie bénigne de la prostate
- svi : invasion des vésicules séminales
- LCP : log de pénétration capsulaire
- gleason : un vecteur numérique
- pgg45 : pourcentage du score de Gleason 4 ou 5
- lpsa : réponse
- train : un vecteur logique

2. Régression linéaire

- **Pourquoi la régression linéaire n'est pas adaptée ?**




```
Residual standard error: 0.3963 on 93 degrees of freedom
Multiple R-squared:  0.397,     Adjusted R-squared:  0.3776
F-statistic: 20.41 on 3 and 93 DF,  p-value: 3.014e-10
```

-La régression linéaire n'est pas adaptée car les hypothèses de départ pour l'utilisation de la régression linéaire ne sont pas vérifiées :

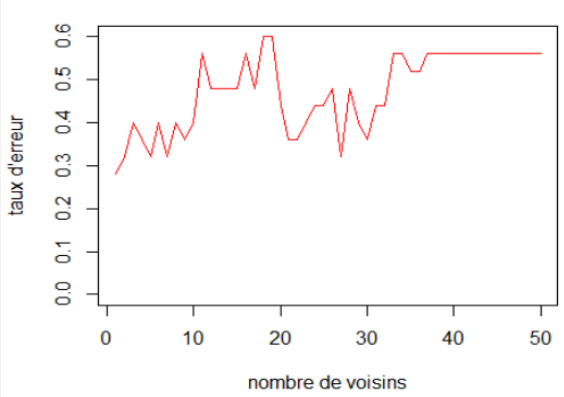
- D'après le graphe, le nuage de points est dispersé et a une forme sans véritables lignes directrices donc l'absence d'homoscédasticité.
- D'après le résumé de notre model « **summary(lm.fit)** » on remarque que R-squared=0.39 (**pour une linéarité forte il faut R-sqaured > 0.70**).

« Le R^2, ou R-carré est appelé coefficient de détermination. C'est un indicateur utilisé en statistiques pour juger de la qualité d'une régression linéaire »

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

3. KNN :

(1) Pour 75%(apprentissage)-25%(test) :

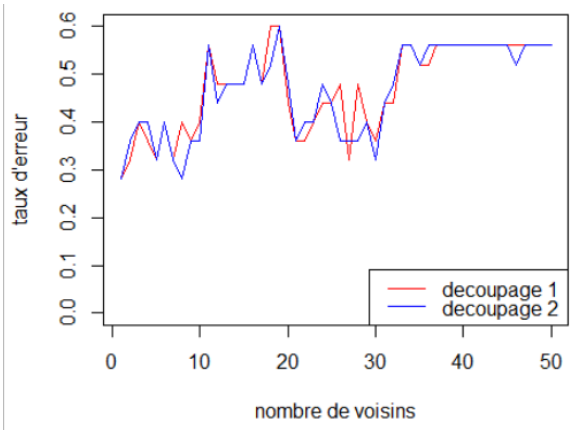


Le meilleur choix du K :

K=1 avec Erreur =0.28 (à éviter)

K=2,5,7 avec Erreur =0.32

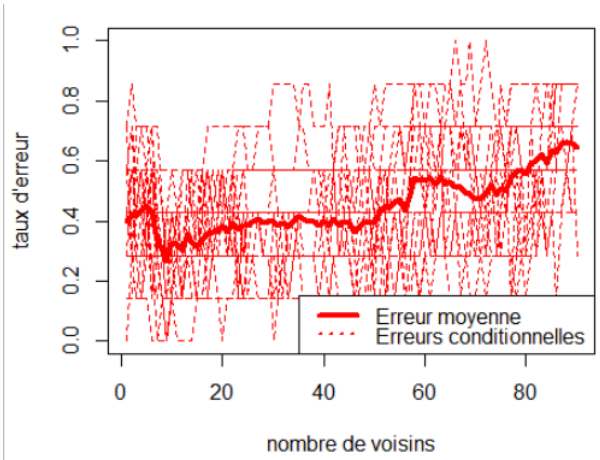
(2) Nouveau découpage apprentissage/test



Le meilleur choix du K :

K=1,8 avec Erreur =0.28

(3) KNN avec 20 découpage aléatoire :

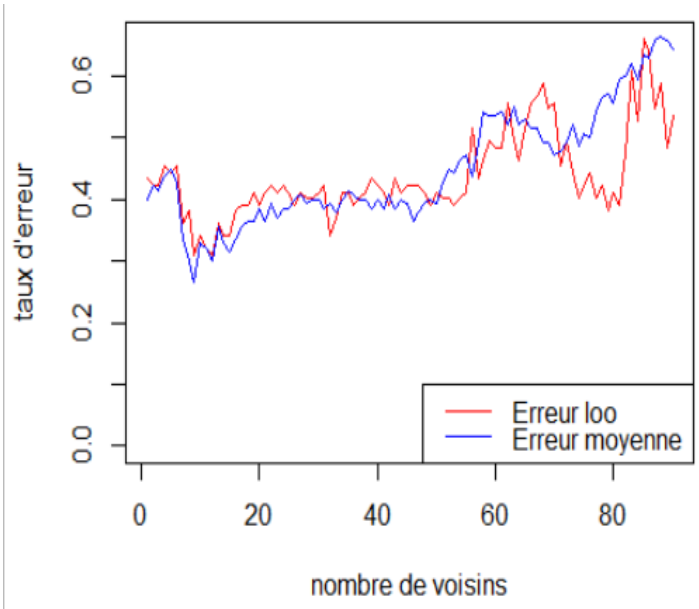


Le meilleur choix du K :

```
> which.min(mean_err_test)
[1] 9
> mean_err_test[which.min(mean_err_test)]
[1] 0.2642857
```

K=9 avec Erreur =0.26428

(4) KNN avec cross-validation :



Le meilleur choix du K :

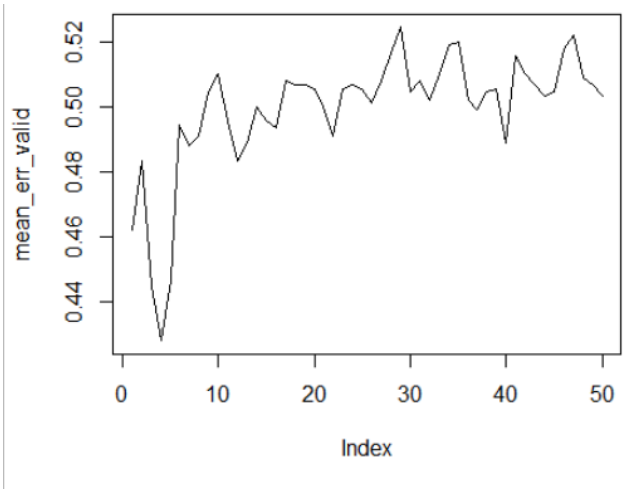
```
> which.min(err_test)
[1] 9
> err_test[9]
[1] 0.3092784
```

K=9 avec Erreur =0.30

(5) Bilan méthodologique pour le choix du paramètre K :

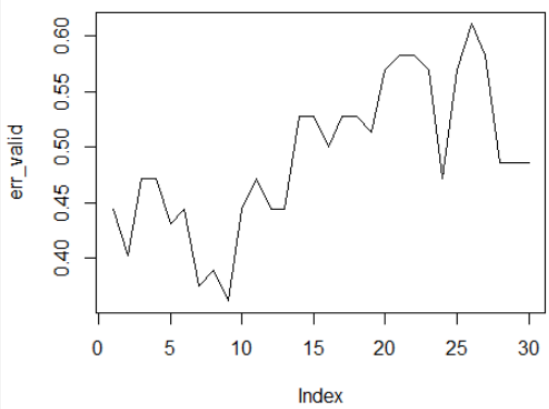
-Il n'y a pas de méthodes statistiques prédéfinies pour trouver la valeur la plus favorable de K.
-Initialisez une valeur K aléatoire et commencez le calcul.
-Le choix d'une petite valeur de K conduit à des frontières de décision instables.
-Tracer un graphe entre le taux d'erreur et K (indiquant les valeurs de K dans un intervalle définie). Ensuite, choisissez la valeur K comme ayant un taux d'erreur minimum.

(6) Pour 75%(apprentissage-validation) et 25%(test) :



Le meilleur choix du K :
K=4 avec Erreur =0.36

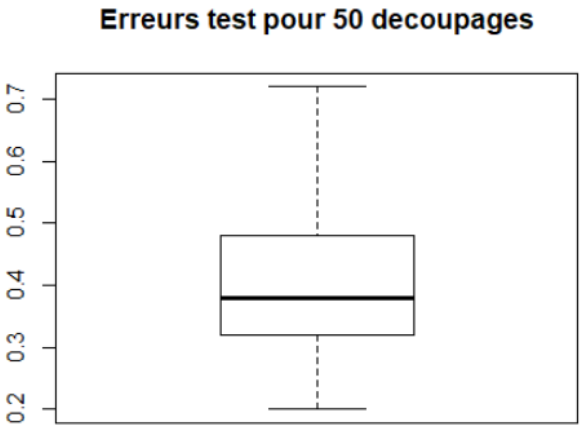
(7) 75%(apprentissage-validation) et 25%(test) avec cross validation:



Le meilleur choix du K :

K=9 avec Erreur =0.36

(8) Pour les courageux knn.cv avec 50 découpages :




```
> which.min(err_valid)
[1] 7
> summary(err_test)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.2000  0.3200  0.3800  0.3952  0.4800  0.7200
```

4. Bayésien naïf :

(1) Avec prostate [, -c (9,10)] :

	high	low
high	31	6
low	16	44

Erreur= (6+16) / 97 = 0.22

(2) Avec X :

	high	low
high	47	0
low	0	50

Erreur=0

```
prostate.d<-X
prostate.d
library(e1071)
m <- naiveBayes(g ~ ., data = prostate.d)
predict(m, prostate.d)
table(predict(m, prostate.d), g)
```

PARTIE II : SPAM

1. Description :

Un ensemble de données collecté chez Hewlett-Packard Labs, qui classe 4601 e-mails comme spam ou non-spam.

En plus de cette étiquette de classe, 57 variables indiquent la fréquence de certains mots et caractères dans l'e-mail.

Les 48 premières variables contiennent la fréquence du nom de la variable (par exemple, entreprise) dans l'e-mail.

Si le nom de la variable commence par num (par exemple, num650), il indique la fréquence du nombre correspondant (par exemple, 650).

Les variables 49-54 indiquent la fréquence des caractères ';', '(', '[', '!', '\ $' Et '\ #'.

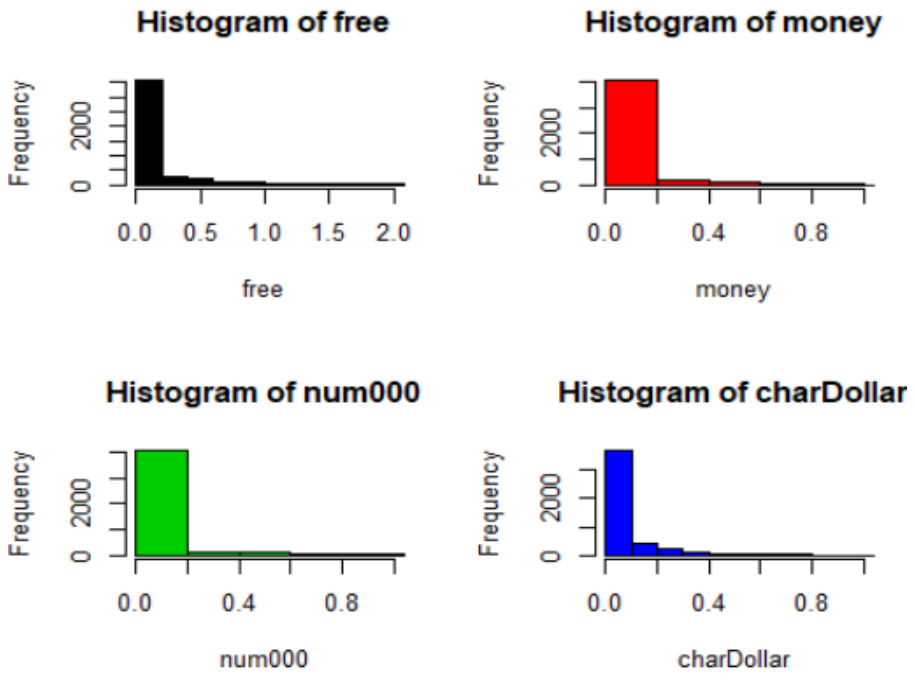
Les variables 55-57 contiennent la moyenne, la plus longue et le total longueur d'exécution des majuscules.

La variable 58 indique le type de courrier. Ce dernier est soit "non-spam" soit "spam".

2. Les analyses statistiques :

1. Analyse statistique univarié :

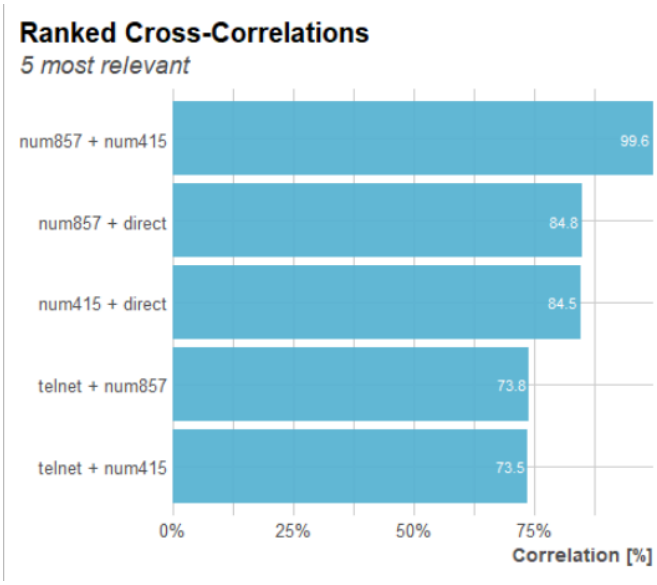
On va choisir quatre variables et faire leurs résumé « Summary » en plus l'histogramme de chacun.



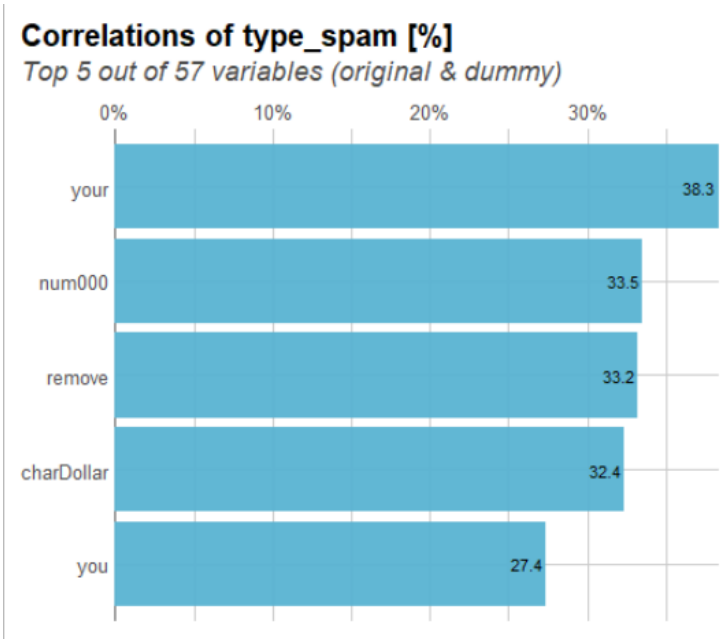
2. Analyse statistique bivariés :

On va travailler avec la bibliothèque « Lares » pour trouver les deux meilleures variables corrélées entre elles et les meilleures variables corrélées avec notre class « type ».

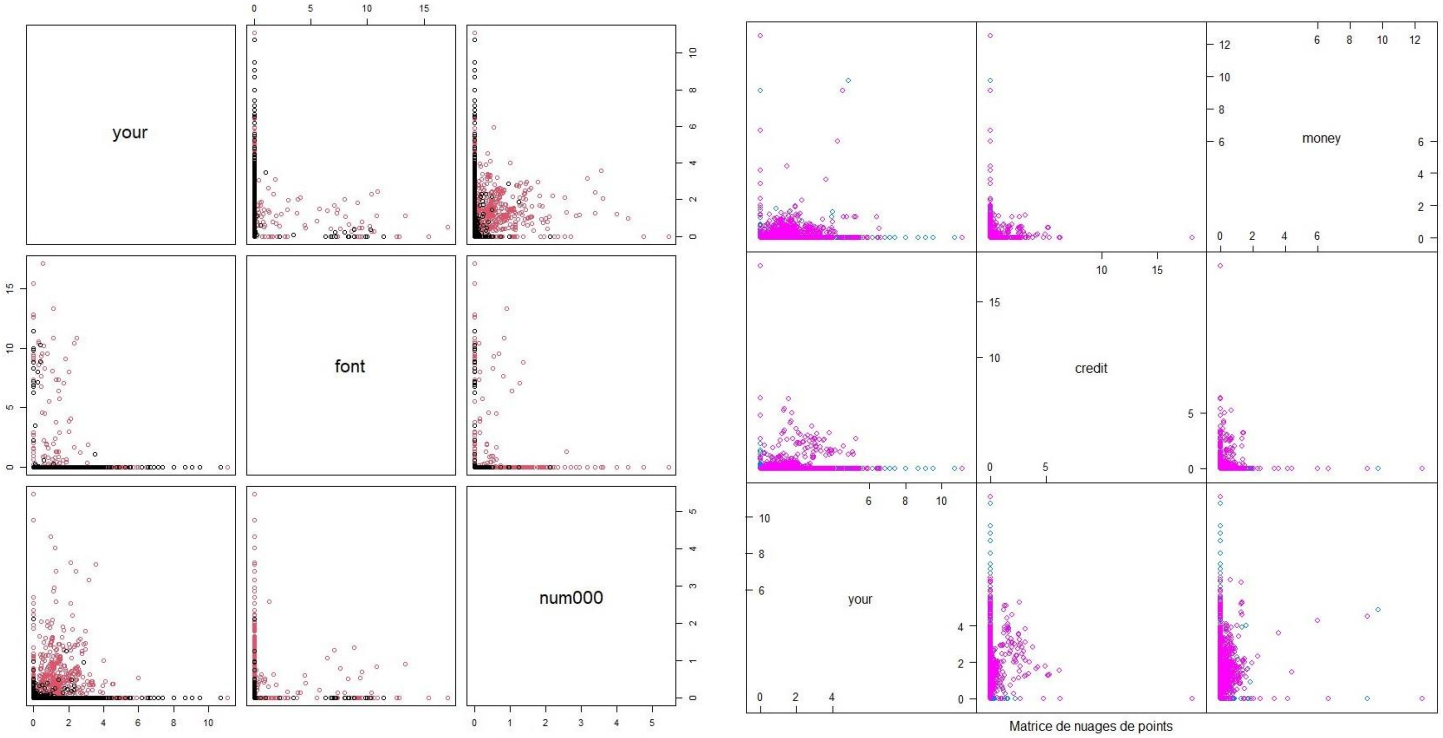
```
library(lares)
corr_cross(spam,top=5) #Meilleures variables corrélées entre eux
```




```
corr_var(spam,type,top =5) #Meilleures Variables corrélées avec "type"
```



-L’utilisation des différentes distributions entre deux variables :



Matrice de nuages de points

3. L’étude comparative des méthodes de classification :

- Nous avons fait une étude comparative des trois méthodes de classification : régression linéaire, k plus proches voisins et le classifieur bayésien naïf, et nous avons obtenu les résultats représentés dans la table suivante :

La méthode	Erreur
régression linéaire	0.1119322
knn avec 75% train 25 % test	0.1998262
knn avec cross validation	0.1798436
le classifieur bayésien naïf	0.2864595

4. L’étude sur le jeu de donnée spam après normalisation :

-Durant notre étude, nous avons utilisé la méthode de normalisation qui consiste à diviser chaque case nij de la table spam par la racine carrée du produit des sommes marginale ni. Et n.j.

```
nj=colSums(X1)
ni=rowSums(X1)
for (i in 1:nrow(X1))
{
  for (j in 1:ncol(X1))
  {
    spam_normalize[i,j] = (X1[i,j]/sqrt(ni[i]*nj[j]))
  }
}
```

Nous avons obtenu les résultats représentés dans la table suivante :

La méthode	Erreur
régression linéaire	0.1388829
knn avec 75% train 25 % test	0.08601216
knn avec cross validation	0.105126
le classifieur bayésien naïf	0.3040643

5. Nous avons également effectué d'autres études :

-Faire des études avec un choisi des variables sur les modèles précédents.

-Utiliser une autre méthode pour la normalisation des données.

-Effectuer de différents tests pour le choix de K dans KNN.

Voici la table de résultats obtenu :

Régression linéaire :

-Les variables choisies pour la régression linéaire sont des variables avec les meilleures corrélations avec notre class « type » (Your , num000, remove) pour construire notre model.

Les cas étudiés	Erreur de régression linéaire
Avec tous les variables	0.1119322
Avec les variables choisi	0.2147359
Tous les variables avec la normalisation Scale()	0.1119322
Avec choix des variables avec la normalisation Scale()	0.2147359

KNN :

Les cas étudiés	Erreur de KNN
knn avec 75% train 25 % test avec tous les variables	0.1998262
knn avec cross validation avec tous les variables	0.1798436
knn avec 75% train 25 % test avec choix des variables	0.1468288
knn avec cross validation avec choix des variables	0.1668115
knn avec 75% train 25 % test avec choix des variables avec la normalisation Scale()	0.1190269
knn avec cross validation avec choix des variables avec la normalisation Scale ()	0.1303215

Le classifieur bayésien naïf :

Les cas étudiés	Erreur de classifieur bayésien naïf
Avec choix des variables	0.04064334
avec choix des variables + scale	0.009345794
tous les variables avec la normalisation demander	0.3040643
tous les variables	0.3040643

6. Bilan concernant les méthodes utilisées et l'impact de la normalisation sur les performances de ces méthodes :

-D'après notre étude, l'estimation du modèle KNN devient moins performante quand le nombre de variables explicatives est grand.

-k plus proches voisins (KNN) est une méthode non paramétrique seul k doit être fixé, le modèle se base sur la mémorisation des observations de l'ensemble d'entraînement pour faire la classification des données de l'ensemble de test.

-La méthode KNN fait le calcul des distances entre la variable et les classes disponibles et ensuite attribue la classe correspondante par vote majoritaire.

-La méthode de régression linéaire est adaptée aux données quantitatives, son objectif est de trouver la relation qui lie une variable d'intérêt, donc la régression linéaire n'est pas dédiée aux problèmes de classification, et la distribution des données ne vérifie pas les hypothèses de départ pour l'appliquer, malgré le fait que dans notre étude elle donne un résultat acceptable dans certains cas.

- Le classifieur bayésien naïf est une méthode facile à mettre en œuvre et ce modèle ne présente pas de risque de surapprentissage.

- Le principe du classifieur naïf de Bayes est de supposer que l'existence d'une caractéristique d'une classe est indépendante de l'existence d'autres caractéristiques, raison pour laquelle nous utilisons l'adjectif naïf.

-La normalisation est une technique efficace pour changer les valeurs des colonnes numériques de l'ensemble de données à une échelle commune, sans déformer les différences dans les plages de valeurs.

-D'après notre étude, nous avons remarqué l'effet de normalisation les résultats choisis surtout KNN qui a donné un bon résultat après normalisation.

-L'algorithme k-plus proche est basé sur la distance, donc la normalisation peut améliorer sa précision et sa performance.

-Certains modèles n'ont pas besoin de normalisation pour donner une bonne performance, d'après notre étude le classifieur naïf de Bayes est l'un de ces modèles.

-Selon notre étude, la normalisation n'a pas amélioré le modèle régression linéaire comme il n'est pas adapté pour les classifications.