Rapport du TP de Machine Learning 2019

Handwritten digit recognition: Reconnaissance automatique des chiffres 7 , 9

I. Contexte:

< Handwritten digit recognition> ou la reconnaissance des chiffres manuscrits(en français) apparait comme un sujet de recherche toujours vivace et suffisamment vaste pour que très peu d'articles envisagent de décrire un tel système dans sa globalité. De l'image au résultat, deux grandes étapes se succèdent dont la première est une transcription de l'image dans une forme faisant intervenir des espaces vectoriels. Celle-ci fait intervenir de nombreux traitements d'images selon la qualité du document initial vient ensuite une modélisation plus formelle qui constitue la reconnaissance proprement dite, on y croise le terme apprentissage cher à l'intelligence artificielle qui équivaut souvent à une optimisation de fonctions aux nombreux coefficients. Et de nombreux domaines.

II. **Objectif**:

L'objectif de ce TP est d'entrainer plusieurs modèles de classification à la reconnaissance des chiffres 7 et 9.

Notamment les types de machines que je vais utiliser dans ce TP sont, dans l'ordre : **la régression logistique**, La **classification naïve bayésienne, les arbres de décision, aussi avec les hypers paramètres**

Dans cette étude j'essaierai d'effectuer une comparaison entre chaque modèle.

III. Préparation des Données :

Avant tout, je dois commencer par explorer mes données, On constate alors que:

- j'ai deux jeux de données **train.zip(taille=7291x257)** et **test.zip(taille=2007x257)** qui sont tous chargé de la Library **ElemStatLearn.**

Les 257 sont les 16x16(dimension de la représentation de l'image) +1 colonne (qui contient le chiffre représenté sur la ligne). Ainsi les colonnes sont notées de V1 à V257.

- Il n'y a aucune valeur manquantes dans les deux jeux de données (train et test)
- J'ai choisi les lignes qui représentent les chiffrent 7 et 9 :
  - Pour les données "train", il y reste au total 1289 observations sur les 7291, soit 17,6%
  - Pour les données "test", il y reste au total 324 observations sur les 2007 soit 16,14%

-J'ai converti la première colonne en valeur catégorielle vu que c'est elle qui contient le chiffre à representer sur la ligne.

IV. Entrainement des Modèles:

1. Regression Logistique:

Après entrainement et prédictions, j'ai obtenu les performances suivantes :

```
Confusion Matrix and Statistics

          Reference
Prediction   7   9
         7 131   4
         9  16 173

               Accuracy : 0.9383
                 95% CI : (0.9063, 0.9619)
    No Information Rate : 0.5463
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.8746

 Mcnemar's Test P-Value : 0.01391

            Sensitivity : 0.8912
            Specificity : 0.9774
         Pos Pred Value : 0.9704
         Neg Pred Value : 0.9153
             Prevalence : 0.4537
         Detection Rate : 0.4043
   Detection Prevalence : 0.4167
      Balanced Accuracy : 0.9343

       'Positive' Class : 7
```

Sur cette capture d'ecran, On voit clairement une accuracy de 93,8%. Ce qui est un peu très bien.

2. Classification naïve bayésienne :

Après entrainement du modèle, on obtient les performances suivantes :

```
Confusion Matrix and Statistics

          Reference
Prediction   7   9
         7 129  10
         9  18 167

               Accuracy : 0.9136
                 95% CI : (0.8775, 0.9418)
    No Information Rate : 0.5463
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8249

 Mcnemar's Test P-Value : 0.1859

            Sensitivity : 0.8776
            Specificity : 0.9435
         Pos Pred Value : 0.9281
         Neg Pred Value : 0.9027
             Prevalence : 0.4537
         Detection Rate : 0.3981
   Detection Prevalence : 0.4290
      Balanced Accuracy : 0.9105

       'Positive' Class : 7
```

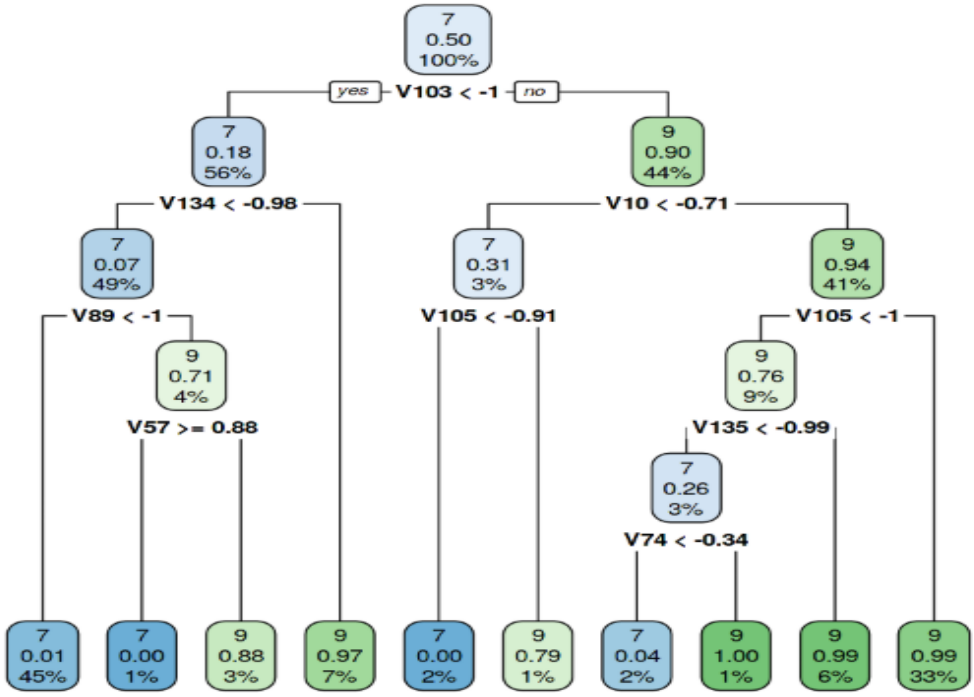
On voit que l'accuracy a diminué de 2%, ce qui veut dire que le modèle logistique est mieux que la naïve bayésienne pour cette étude.

3. Decison Tree:

```
Confusion Matrix and Statistics

          Reference
Prediction   7   9
         7 138   4
         9   9 173

               Accuracy : 0.9599
                 95% CI : (0.9324, 0.9785)
    No Information Rate : 0.5463
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9188

 Mcnemar's Test P-Value : 0.2673

            Sensitivity : 0.9388
            Specificity : 0.9774
         Pos Pred Value : 0.9718
         Neg Pred Value : 0.9505
             Prevalence : 0.4537
         Detection Rate : 0.4259
   Detection Prevalence : 0.4383
      Balanced Accuracy : 0.9581

       'Positive' Class : 7
```

Ici par contre, j'ai eu une très bonne performance, notre accuracy est de 95%.

Notre arbre de décision ayant permis une telle performance est donné par la figure suivante :



Ajoutons les Hyper paramètres :

On obtient alors une accuracy de 0.966049382716049 soit 96,6% ce qui semble être le meilleur modèle dans mon étude de cas avec les hypers paramètres:

minsplit = 4, minbucket = round(5 / 3),

maxdepth = 20, cp = 0

V. Conclusion:

Pour la reconnaissance automatique des chiffres 7 et 9, le modèle de classification qui m'a permis d'avoir une meilleure performance est l'arbre avec les hypers paramètres :

minsplit = 4, minbucket = round(5 / 3),

maxdepth = 20, cp = 0