Prévision de durées d'hospitalisation aux Hospices Civils de Lyon (HCL)

Grégoire MASSOT

21/01/2016

Plan

Plan

Déroulement du projet

- **Entreprise** : Hospices civils de Lyon
- **Tuteur entreprise** : Antoine DUCLOS - Médecin à Lyon
- **Tuteur école** : Thierry GARAIX

Déroulement du projet

4 venues à Lyon : 20 Octobre, 6 Novembre, 11 Décembre, 15 Janvier.

Contexte du projet

Les HCL doivent optimiser la gestion de leurs ressources pour accueillir le plus de patients avec les ressources disponibles.

Contexte du projet

Prévoir la durée d'hospitalisation des patients permet de maximiser l'occupation des lits

Contexte du projet

Les HCL souhaitent utiliser leur base de données sur les patients précédents pour prédire les durées d'hospitalisation des nouveaux patients.

Position du problème

Objectif du PI : déterminer la durée d'hospitalisation d'un patient à partir des données recueillies lors de son arrivée à l'hôpital et des statistiques sur les patients précédents.

Position du problème

Schéma de la Base de données anonymisée des HCL

Duree du séjour	Âge du patient	Diagnostic principal	Service hospitalier	...
duree1	age1	dp1	service1	...
duree2	age2	dp2	service2	...
duree3	age3	dp3	service3	...
...	...	...	...	...

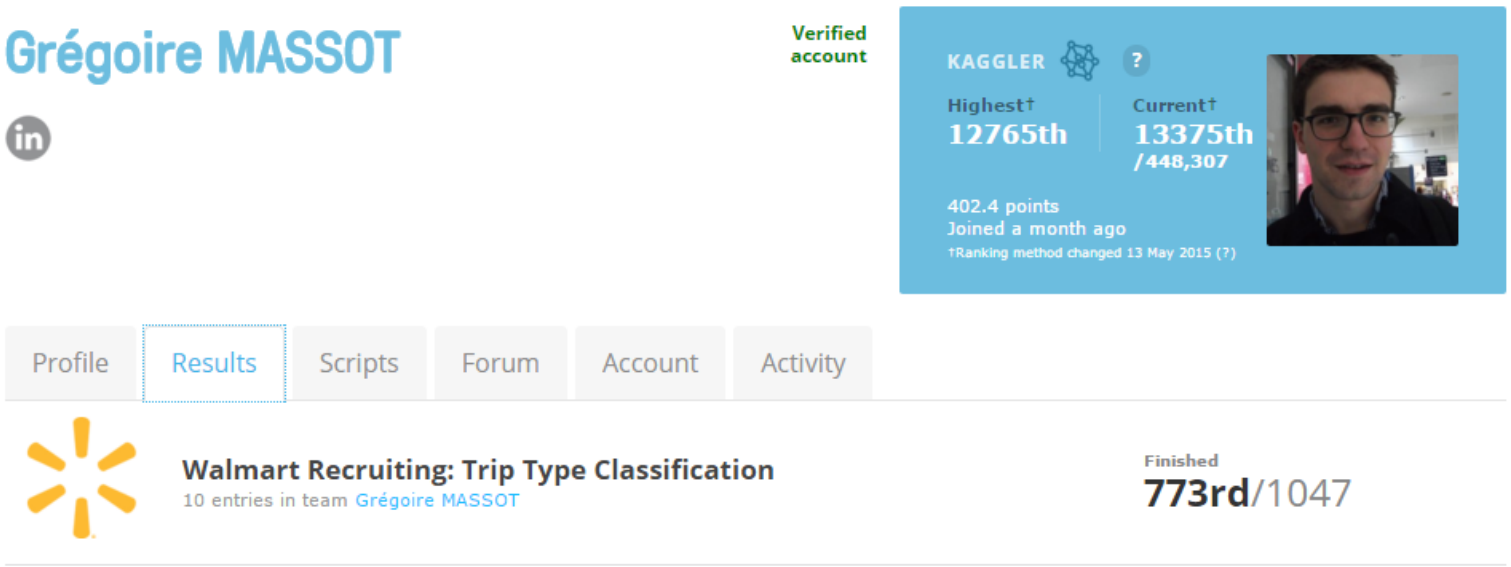
Plan

Démarche

Reprise du code R du challenge "Éolienne" de la majeure Data Science

Démarche

Inscription au challenge Walmart sur kaggle.com



Plan

Code R - Chargement et sélection des données

```
1  # Chargement de la BDD
2  donnees_hcl <- read.csv2(file = "base_ano.txt",header =
       TRUE, sep="\t")
3  # Sélection des séjours de durée supérieure ou égale á 1
       jour
4  donnees_hcl <- donnees_hcl[donnees_hcl$duree >= 1,]
5  #Selection des prédicteurs effectivement disponibles
       lors de l'entrée du patient
6  donnees_hcl <- donnees_hcl[,-seq(8,57)]
7  donnees_hcl <- subset(donnees_hcl, select = -c(moissor))
8  donnees_hcl <- subset(donnees_hcl, select = -c(sortie))
9  donnees_hcl <- subset(donnees_hcl, select = -c(ID))
10
11 # Transformation du facteur "age" en variable continue
12 donnees_hcl$age <- as.character(donnees_hcl$age)
13 donnees_hcl$age <- as.numeric(donnees_hcl$age)
```

Code R - Split de la BDD

```r
# Création des jeux de Train et de Test
TrainData <- donnees_hcl[donnees_hcl$Selected == 1,]
TestData <- donnees_hcl[donnees_hcl$Selected == 0,]

predicteurs_Train <- subset(TrainData, select = -c(duree))
duree_Train <- subset(TrainData, select = c(duree))

predicteurs_Test <- subset(TestData, select = -c(duree))
duree_Test <- subset(TestData, select = c(duree))
```

Code R - Métrique de performance RMSE

```r
# Déclaration de la fonction RMSE

RMSE <- function(y, pred)
{
  sqrt(mean((y - pred)^2))
}
```

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i^{pred})^2}$$

```r
# Tuning des paramètres de xgboost avec caret
library(caret)
library(foreach)
library(doParallel)
cl <- makeCluster(8)

tuneGrid <- expand.grid(max_depth = c(1,2,3,4,5,6,7,8,9)
    ,
                        nrounds = 100,
                        eta = c
                            (0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,
                            )

fitControl <- trainControl(method = "repeatedcv",
                           number = 3,
                           repeats = 1)

xvalXGB <- train(TrainData$duree ~ .,
                 data = TrainData,
                 method = "xgbTree",
                 tuneGrid = tuneGrid,
                 trControl = fitControl)
xvalXGB
```

Code R - Construction du modèle

```r
# Construction du modèle
library(xgboost)
param <- list("objective" = "reg:linear",
              "eta"=0.5,
              "max.depth"=5,
              "nthread" = 8)

modelXgboost <- xgboost(data = as.matrix(predicteurs_Train)
                        , label = as.matrix(duree_Train)
                        , params=param
                        , nrounds = 100)
predictionXGBoost <- predict(modelXgboost,
                             as.matrix(predicteurs_Test)
                                 )

RMSE(duree_Test, predictionXGBoost)
```

Code R - Visualisation des résultats

```r
# feature importance

names <- dimnames(predicteurs_Train)[[2]]
importance_matrix <- xgb.importance(names, model =
    modelXgboost)
xgb.plot.importance(importance_matrix)
xgb.plot.tree(feature_names = names, model =
    modelXgboost, n_first_tree = 2)
```

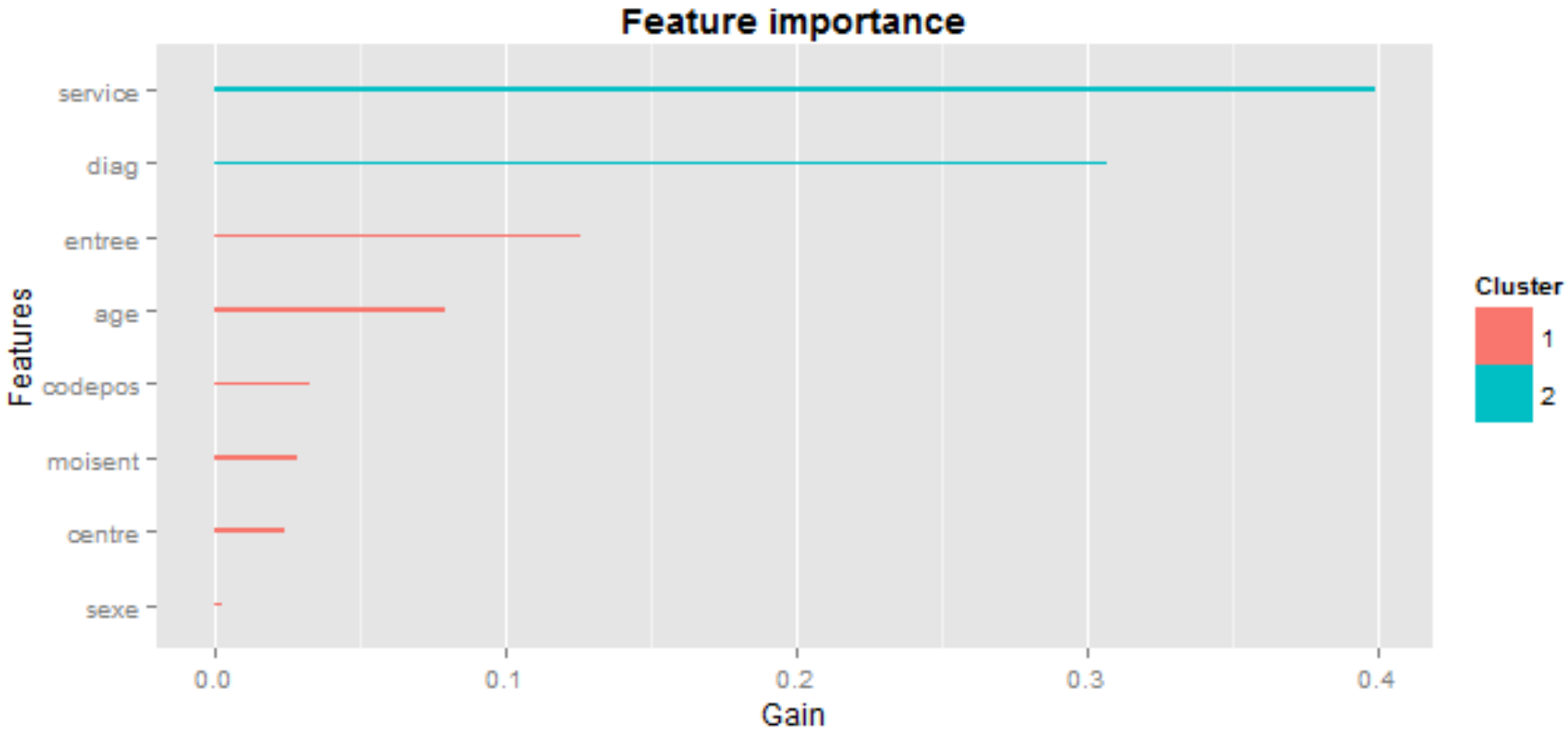
Plan

Analyse des résultats

Méthode	RMSE
Régression linéaire (étude précédente menée par les HCL)	6.81
Machine learning - sans la variable ID	6.86
Machine learning - avec la variable ID	5.73

Analyse des résultats

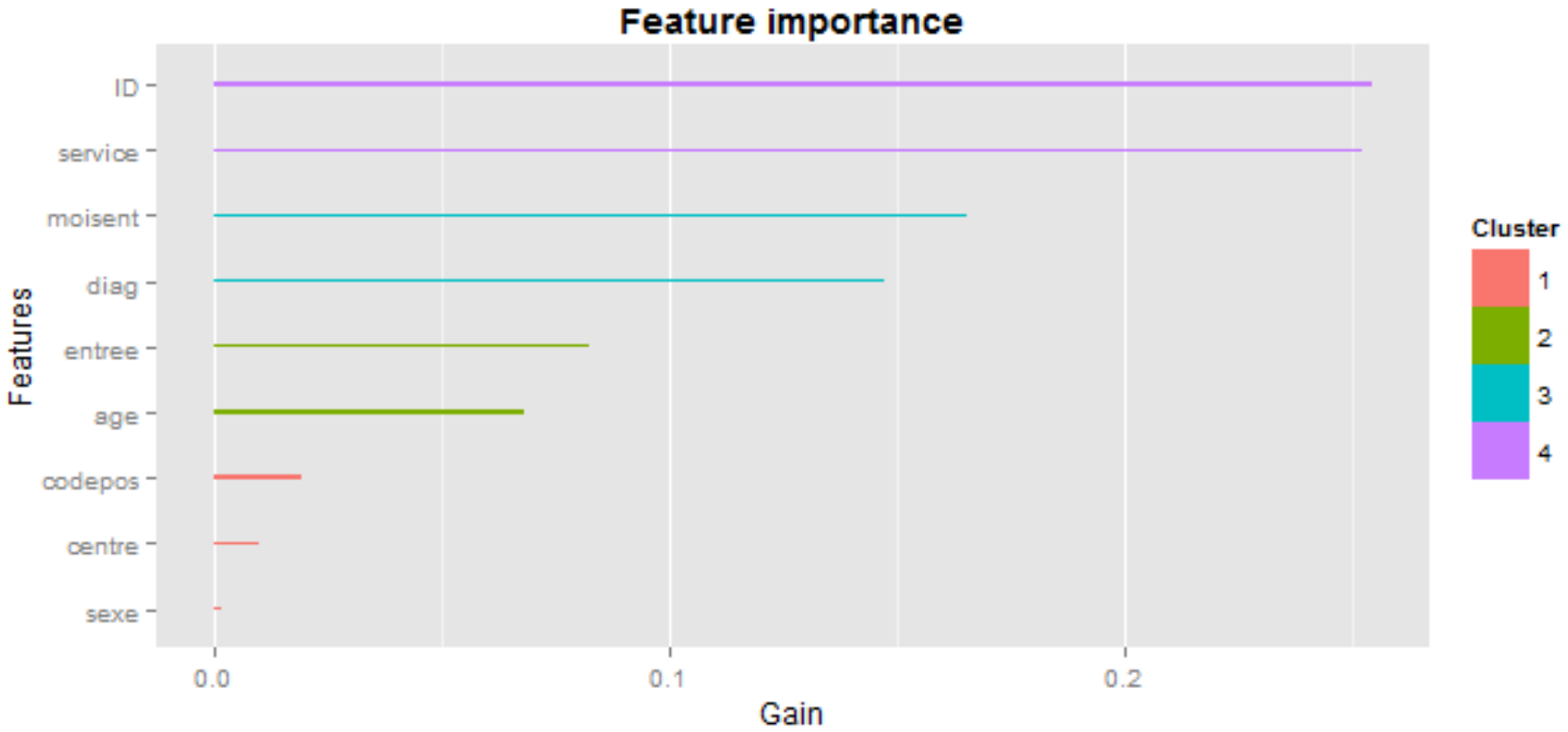
Analyse de l'importance des prédicteurs - Sans ID



MINES Saint-Étienne

Analyse des résultats

Analyse de l'importance des prédicteurs - Avec ID



Fin