**Rapport du projet : Régression Linéaire : Modèles et régularisation**

**Language :R**

**Dataset : Home\_data.csv**

**Description :**

La Dataset comporte 21631 Observations et 21 caractères . Elle révèle les paramètres après lesquels un prix d’un logement peut être estimer selon l’étude faite . Notre problème et celui de régression sur une variable continue (le prix est un réel (float)).

Etudiant : Aymen Tlili

Groupe : 2éme IDSD 1

Matière : Machine Learning

Sommaire :

1. Exploration et visualisation des données

1-exploration du fichier csv (le séparateur des colonnes, la première ligne d’entête)

2-importation et exploration des types des donnés (numériques , dates , ordinal)

3-étude le matrice de corrélation

4-description de la dimensionnalité de la DataSet

(5)-dans un temps ultérieur : interrogation à propos du Skew et Kurtosis

6-Le dendrogramme des features

1. Data pre-processing

[0]-choix des variables à étudier (se débarrasser de l’ID et date )

1-recherche des valeurs manquantes et remplacement avec moyenne(plupart des caractéres)

2-définition des quantiles pertinents

3-normalisation et standardisation des valeurs des features et de la cible (prix)

4-visualisation d’un boxplot et essai à l’élimination des quantiles

5-traçage d’ histogramme ,scatterplot,bar chart

1. Modeling

1-le regresseur lineaire basique (Vanilla)

2-les modéles regularisés : (Hyper Parameter search)

(recherche des paramètres optimaux)

-Ridge

-Lasso

-Elastic Net

1. Evaluation

1-a R2 score ,RMSE,MSE,RSE

1-b repeated Cross validation

2-Définition d’une fonction qui sera utilisée

Observations et remarques :

A :

-Il faut se placer dans l’environnement de travail convenable pour importer la Dataset

-Read\_table et read\_csv ont des comportements diffénrent à l’égard des’’

-On a observé la forme des Dates fournies et la partie heure /minute/secondes ne fournit pas d’informations comme c’est toujours identique.

-En effet L’ID et la date n’avait qu’un effet presque aléatoire sur le , L’ID se corrélait avec 40% .

C’est purement de la chance et on peut se débarrasser de ses paramétres sans y penser profondément.

Cependant le code postale peut désigner une région spécifique avec un certain niveau de vie et un environnement qui peut influer le prix .Quand on avance dans l’exploration on voit clairement que cette information peut être introduite dans la variable ‘View’ et ‘Lat’

-La dataset comporte plusieurs cas particuliers qui peuvent nuire à notre modélisation . ceci est observable dans le boxplot ainsi dans la Skewness des variables(un travail de normalisation est mise à l’echelle est nécissaire)

B :

-si on tenait compte du coefficient de corrélation linéaire de Pearson avec le niveau 0.87 comme critère de choix : le modèle se limitera à 2 variables : nombre de chambres et la surface de l’appartement

-Cependant d’autres variables semble intéressantes et englobe l’information dont on a peut-être avoir eu s’en débarrasser et qui sont proches de ce seuil .

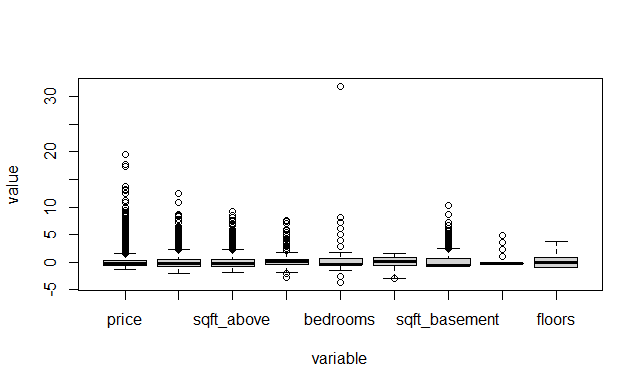
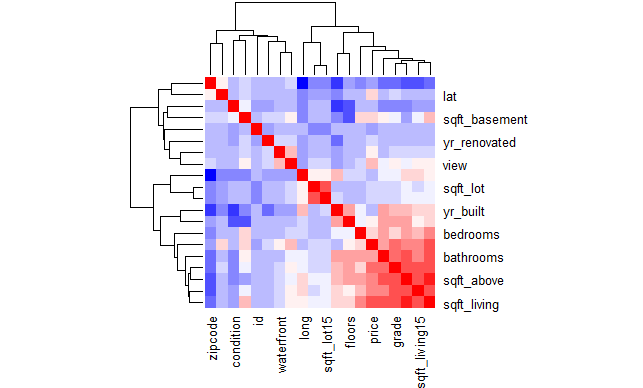
Remarque : LA CORRELATION NE FAIT PAS LA CAUSALITE

-mon échec à se débarrasser des outliers se traduira par un mauvais modèle .

-On remplace par une moyenne les valeurs manquantes sauf dans le cas des floors qui ne peuvent etre que des entiers

-Je démontre ainsi ma compréhension du principe du graphique dans R (la grammaire= les couches)

gg.plot(data, aes())+geompoint +facet grid+systéme de coordonnée +stats +metriques geo



**C- le résultat du sommaire qui indique le bon choix des varibales**

Residuals:

Min 1Q Median 3Q Max

-4.3797 -0.3083 -0.0440 0.2148 11.6355

Coefficients: (1 not defined because of singularities)

Estimate Std. Error t value Pr(>|t|)

(Intercept) 0.0001746 0.0048300 0.036 0.97116

sqft\_living 0.5790987 0.0133654 43.328 < 2e-16 \*\*\*

sqft\_above 0.0849318 0.0118490 7.168 7.95e-13 \*\*\*

bathrooms 0.0391449 0.0082127 4.766 1.89e-06 \*\*\*

bedrooms -0.0957512 0.0060918 -15.718 < 2e-16 \*\*\*

lat 0.2713092 0.0048933 55.445 < 2e-16 \*\*\*

sqft\_basement NA NA NA NA

view 0.2093666 0.0051211 40.883 < 2e-16 \*\*\*

floors -0.0191745 0.0064638 -2.966 0.00302 \*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6149 on 16201 degrees of freedom

Multiple R-squared: 0.6172, Adjusted R-squared: 0.617

F-statistic: 3731 on 7 and 16201 DF, p-value: < 2.2e-16

**RMSE R2**

**7.8899246 0.6171866**

La fonction d’évaluation pour les modèles Régularisés qui comporte les métriques d’évaluation demandés :

#Step 1 - create the evaluation metrics function

3eval\_metrics = function(model, df, predictions, target){

4 resids = df[,target] - predictions

5 resids2 = resids\*\*2

6 N = length(predictions)

7 r2 = as.character(round(summary(model)$r.squared, 2))

8 adj\_r2 = as.character(round(summary(model)$adj.r.squared, 2))

9 print(adj\_r2) #Adjusted R-squared

10 print(as.character(round(sqrt(sum(resids2)/N), 2))) #RMSE

11}

12

13# Step 2 - predicting and evaluating the model on train data

14predictions = predict(lr, newdata = train)

15eval\_metrics(lr, train, predictions, target = 'unemploy')

16

17# Step 3 - predicting and evaluating the model on test data

18predictions = predict(lr, newdata = test)

19eval\_metrics(lr, test, predictions, target = 'unemploy')