PROJET MLG

L'objectif de ce projet est d'expliquer le prix d'un logement à partir de ses caractéristiques et de sa localisation.

La base de données

Nous avons à notre disposition une base de données « house_data.csv » avec 21 613 observations et 21 variables.
Dans ces variables se trouve 20 potentielles variables explicatives et une variable à expliquer « price », le prix de vente du logement.
Nous allons ici expliquer « price » avec un modèle linéaire de la forme Price=XB, où X représente notre jeu de données et B le vecteur des estimateurs associés au jeu de données.

Le Choix des variables explicatives

Plusieurs variables présentent dans la base de données ne peuvent pas être significatives dans un modèle linéaire pour expliquer le prix d'un logement.
La variable ID est un identifiant unique pour chaque logement vendu, il n'a donc aucune influence sur notre variable à expliquer.
La date de vente ne donne pas vraiment d'informations sur le prix de vente d'un logement, d'autant plus que dans notre jeu de données est présent l'année de construction et l'année de la dernière rénovation.
Les variables lat (lattitude), long (longitude), zipcode (code postal) et date (date de vente), ne peuvent être utilisées dans notre modèle car il s'agit de variables non ordonnées.
On pourrait donc introduire ces variables sous forme d'indicatrices mais cela surchargerait notre modèle.

Nous décidons alors de retirer de notre modèle les cinq variables ID, lat, long, zipcode et date.
Il nous reste alors 15 variables pour expliquer le prix du logement.

Transformation des variables

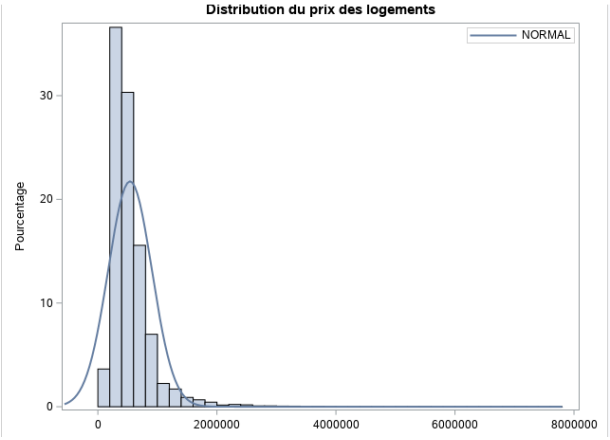
La procédure Freq sur SAS nous a permis d'observer les 15 variables une à une et de décider ensuite si de premières transformations sont nécessaires.

Variable	Type	Commentaires
bedrooms	Numérique	Regrouper certaines modalités
bathrooms	Numérique	Regrouper certaines modalités
condition	Numérique	Regroupement 1 et 2
floors	Texte	Regrouper 3 et 3.5
grade	Numérique	Regrouper 1 à 5 et 11 à 13
sqft_above	Numérique	Regrouper par milliers
sqft_living	Numérique	Regrouper par milliers
sqft_lot	Numérique	Regrouper par milliers
sqft_basement	Numérique	Regrouper par 500
sqft_living15	Numérique	Regrouper par milliers
sqft_lot15	Numérique	Regrouper par milliers
yr-renovated	Numérique	Regroupement années
yr_built	Numérique	Regroupement années

view	Numérique	Rien à changer
waterfront	Numérique	Rien à changer

A travers ces observations, nous allons opérer de nombreux regroupements sur SAS afin que les différentes variables soient plus équilibrées.
Pour cela, nous allons implémenter les nouvelles variables dans un nouveau jeu de données, que nous nommerons house_final.

Pour une meilleure précision, il est préférable que nos variables s'apparentent à des lois normales.
Graphiquement, le prix des logements est proche d'une loi normale.
Via PROC SGPLOT, nous observons la distribution de nos variables, et la comparons à celle d'une loi normale.
A l'aide des graphiques obtenus des variables explicatives, on décide de ne pas modifier car cela ne leur permettrait pas de plus s'apparenter à une loi normale.



Scinder la base de données, apprentissage et validation

Une fois les transformations réalisées sur les variables explicatives, nous allons pouvoir réaliser le modèle.
Mais tout d'abord, nous allons scinder notre base de données en deux, une partie apprentissage et une partie validation.
La base de données apprentissage nous permettra de poser le modèle alors que la base de données validation nous permettra de vérifier notre prédiction.

Pour cela, nous utilisons PROC SURVEYSELECT après avoir ordonné nos données apprentissage en fonction du prix des logements
PROC SURVEYSELECT nous permet de scinder de façon aléatoire notre base de données en conservant une même répartition du prix des logements dans les deux jeux de données.
Ainsi, nous avons « APP» avec 17501 observations et « VAL » avec 4112 observations.

Le modèle

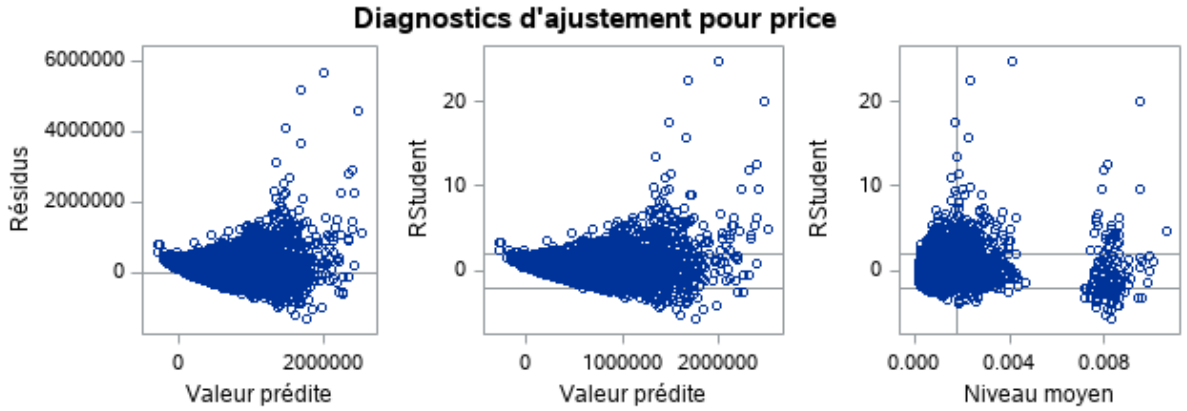
Une fois les deux matrices obtenues, nous utilisons PROC REG sur la base de données « APP » (apprentissage).
On obtient le tableau suivant :

Variable	Valeur estimée des paramètres	Erreur type	SC Type II	Valeur F	Pr > F
Intercept	6116846	164875	7.526099E13	1376.40	<.0001
bedrooms_rec	-24468	2549.81502	5.035064E12	92.08	<.0001
bathrooms_rec	66607	3759.62956	1.716217E13	313.87	<.0001
condition_rec	22048	3037.59751	2.880833E12	52.69	<.0001
floors_rec	10814	5003.75514	2.553718E11	4.67	0.0307
grade_rec	140283	2646.30299	1.536597E14	2810.18	<.0001
sqft_above_rec	36.70772	6.69556	1.643493E12	30.06	<.0001
sqft_living_rec	79.42985	6.55312	8.033368E12	146.92	<.0001
sqft_lot_rec	-4.06210	1.20325	6.231805E11	11.40	0.0007
sqft_basement_rec	43.23293	6.02541	2.815026E12	51.48	<.0001
sqft_living15_rec	46.58733	3.75907	8.398485E12	153.59	<.0001
sqft_lot15_rec	-2.73838	1.26525	2.561301E11	4.68	0.0305
yr_renovated_rec	15.91827	4.64563	6.419913E11	11.74	0.0006
yr_built_rec	-3629.96922	84.57926	1.007173E14	1841.95	<.0001
view_rec	54195	2692.59254	2.215156E13	405.11	<.0001
waterfront_rec	649092	21593	4.940862E13	903.60	<.0001

L'option « selection = stepwise » est la méthode pas à pas, d'ajout et de retrait des variables, afin d'observer quel modèle est le plus significatif.
Toutes les variables étant significatives les unes après les autres (au niveau 0.15), on conserve notre modèle.

Précision du modèle

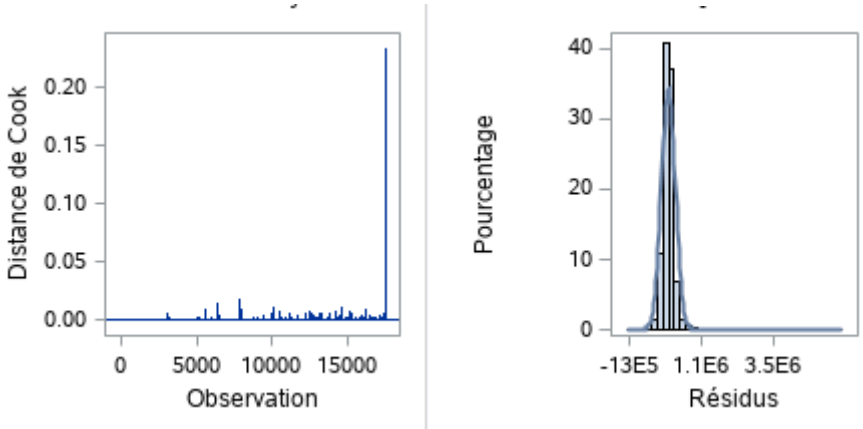
Le modèle possède un R de 0.6309 et la somme des erreurs quadratiques est égale à 16.



On peut observer les graphiques de valeurs prédites sur la base apprentissage.

La distance de cook est toujours inférieure à 1, il n'y a donc pas de point trop influent.

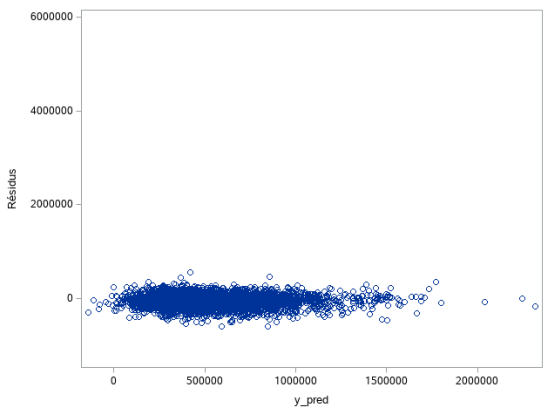
De même, les résidus suivent graphiquement une loi normale.



Nous pouvons désormais appliquer ce modèle à notre base de données de validation.

Les résidus semblent respecter l'hypothèse d'homoscédasticité (espérance nulle), ce qui est positif pour notre modèle.

Notre modèle est donc un plutôt bon prédicteur du prix des logements.



AUBRY ANTONIN
TEBOUL SAMUEL
EL KASMI ADNANE