House prices

Vanessa Kenniche, Thomas Fernandes, Yassine Ouerghi

2024-05-03

1. Analyse exploratoire et traitement des données

1.1. Les données

Id	MSSubClass	LotFrontage	LotArea	Street	...	MiscFeature	MoSold	YrSold	SaleCondition	SalePrice
0	60	65.0	8450	Pave	...	NaN	2	2008	Normal	208500
1	20	80.0	9600	Pave	...	NaN	5	2007	Normal	181500
2	60	68.0	11250	Pave	...	NaN	9	2008	Normal	223500
3	70	60.0	9550	Pave	...	NaN	2	2006	Abnorml	140000
4	60	84.0	14260	Pave	...	NaN	12	2008	Normal	250000

Les deux jeux de données sont composés de 80 variables explicatives. On remarque que la seule différence entre les deux est la présence de la variable cible SalePrice dans le jeu de données d'entraînement. Certaines colonnes sont catégorielles, d'autres numériques, et il semblerait que certaines colonnes contiennent des valeurs manquantes, que nous analyserons par la suite.

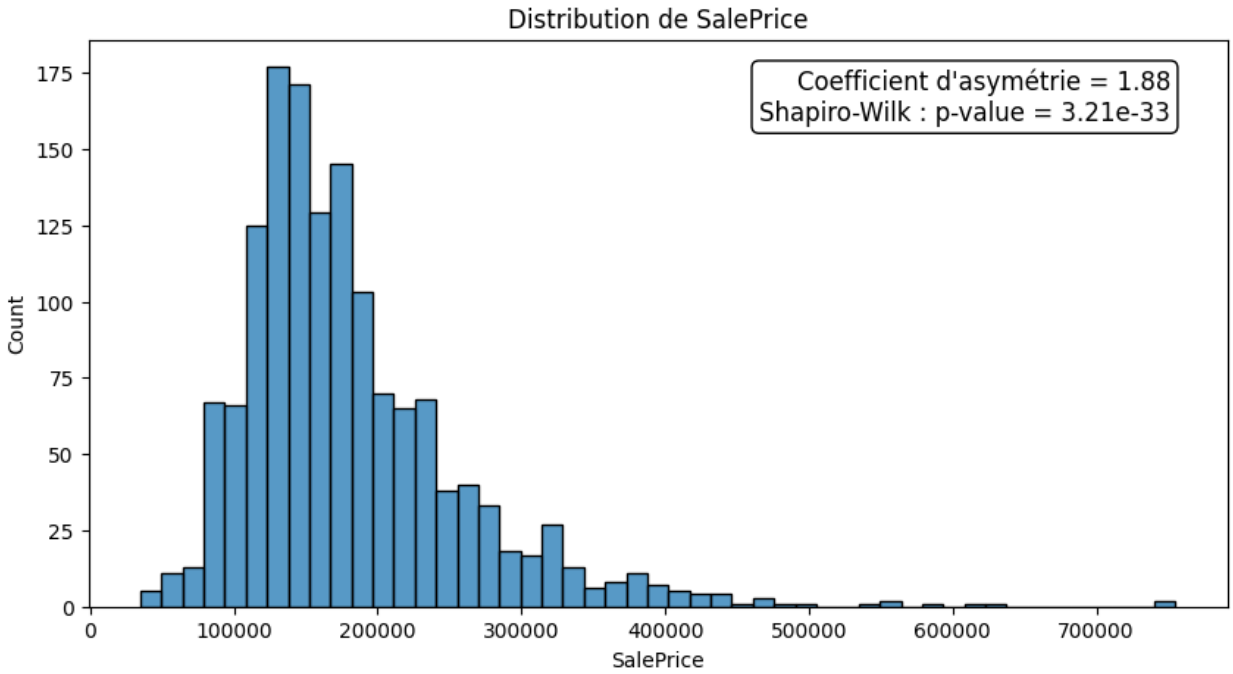
LotArea	YearBuilt	YearRemodAdd	YrSold	SalePrice
1460.0000	1460.0000	1460.0000	1460.0	1460.0000
10516.828	1971.2678	1984.8657	2007.8	180921.19
9381.2649	30.2029	20.6454	1.3281	79442.502
1300.0000	1872.0000	1950.0000	2006.0	34900.000
...	...	...	...	...
215245.00	2010.0000	2010.0000	2010.0	755000.00

La fonction describe() permet de voir les statistiques descriptives des données. Notre jeu de données comprend une liste de 1460 maisons. On remarque que pour notre variable cible SalePrice, la moyenne est de 180 921,2$ pour un minimum de 34 900$ et un maximum de 755 000$. La superficie du terrain LotArea varie considérablement, avec une moyenne d'environ 10 516 pieds carrés. Cela montre la diversité des propriétés en termes de taille de terrain. La qualité générale des matériaux et de la finition de la maison est évaluée sur une échelle de 1 à 10, avec une moyenne d'environ 6, ce qui suggère que la plupart des maisons sont de qualité au-dessus de la moyenne. YearBuilt et YearRemodAdd : Ces colonnes indiquent respectivement l'année de construction et l'année de la dernière rénovation des maisons. La moyenne de l'année de construction est de 1971, ce qui implique que le jeu de données inclut principalement des maisons construites dans la seconde moitié du 20ème siècle.

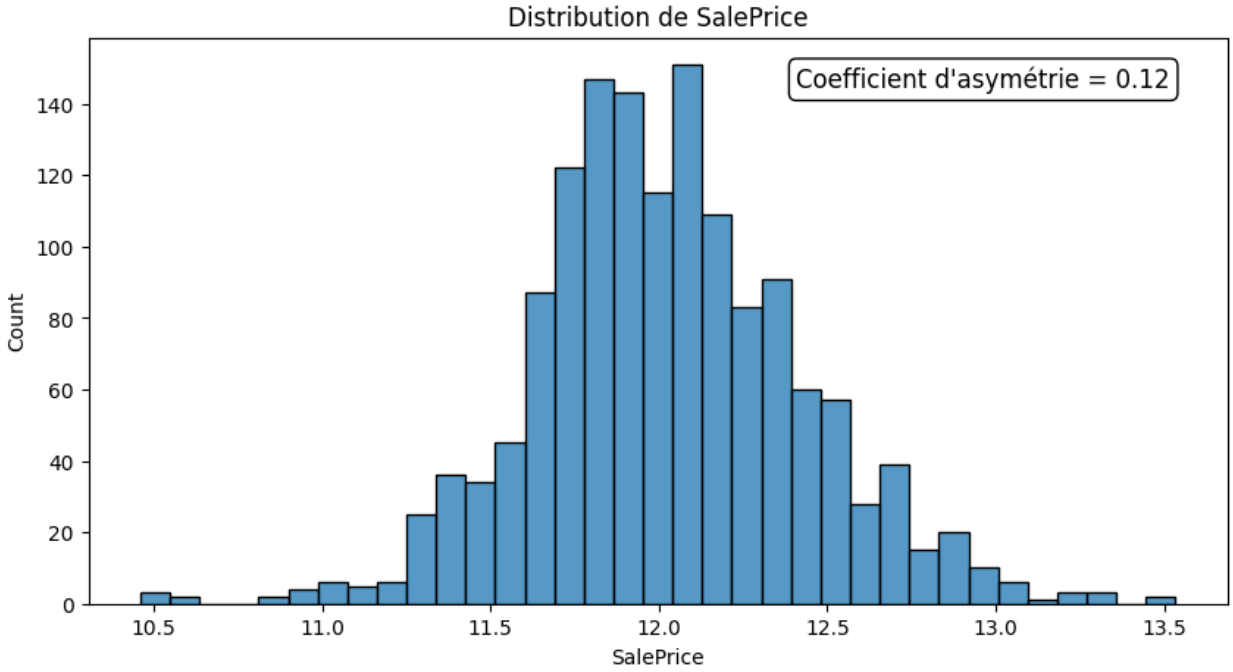
1.2. Traitement des données

1.2.1. Test de normalité sur SalePrice

Dans l'optique d'avoir les régressions les plus performantes possibles, surtout si elles sont paramétriques, vérifier la normalité de la variable cible est une étape crucial du traitement des données.



On remarque graphiquement que la distribution est étalée à droite, le coefficient d'asymétrie de 1.88 vient confirmer cette hypothèse. De plus, la p.value associée au test de Shapiro étant inférieure à 0.05, on rejette l'hypothèse nulle de normalité. On peut donc conclure que la variable cible n'est pas normalement distribuée. Pour réduire cette étalement on va lui appliquer une transformation logarithmique.

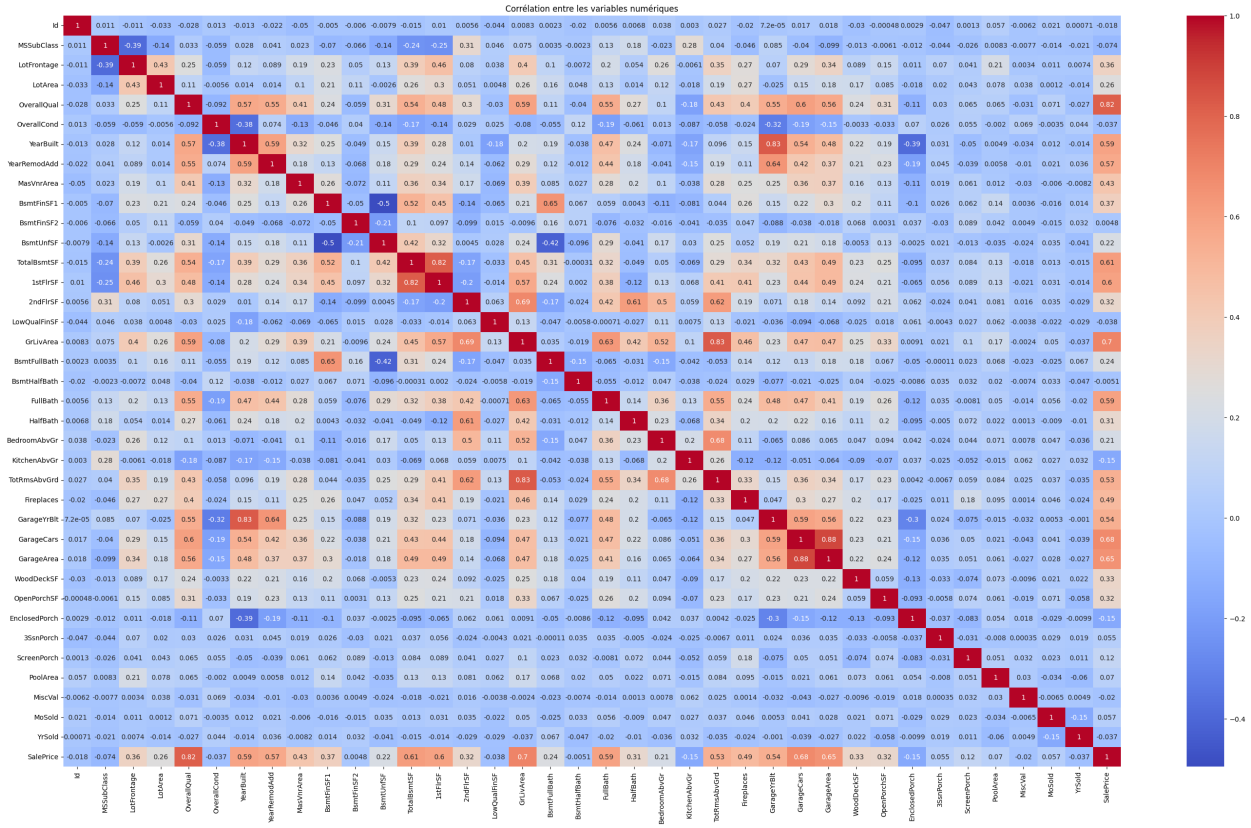


Après la transformation logarithmique, la distribution de SalePrice semble beaucoup plus proche d'une distribution normale, comme en témoigne le coefficient d'asymétrie réduit à 0.12. Cela suggère que l'asymétrie de la distribution a été significativement diminuée et que la distribution est désormais plus proche d'une gaussienne.

Nous appliquons également cette transformation à toutes les variables explicatives qui sont étalées.

1.2.3. Variable importantes et corrélation

Afin d'extraire les variables les plus importantes de notre jeu de données, on va utiliser la méthode de corrélation. Les variables qui auront un coefficient de corrélation avec SalePrice inférieur à 0.3 en valeur absolue seront supprimées.



On remarque qu'il y a des variables explicatives qui sont fortement corrélées entre elles. On va donc les éliminer pour éviter le problème de multicolinéarité.

La multicolinéarité entre les variables explicatives peut conduire à des estimations des coefficients instables et à une diminution de la performance. GarageCars et GarageArea sont fortement corrélés, car tous les deux décrivent la taille du garage, ce qui rend l'un d'eux redondant. De même, YearBuilt et GarageYrBlt sont susceptibles d'être similaires dans de nombreux cas, car le garage est souvent construit en même temps que la maison. La superficie habitable (GrLivArea) et le nombre total de pièces au-dessus du sol (TotRmsAbvGrd) sont également liés car plus l'espace est grand, plus on peut y aménager de pièces. Enfin, TotalBsmtSF et 1stFlrSF sont souvent identiques ou très proches car le premier étage est généralement construit directement au-dessus du sous-sol.

Variable 1	Variable 2	Corrélation
GarageCars	GarageArea	0.882475
YearBuilt	GarageYrBlt	0.825667
GrLivArea	TotRmsAbvGrd	0.825489
TotalBsmtSF	1stFlrSF	0.819530
OverallQual	SalePrice	0.817185

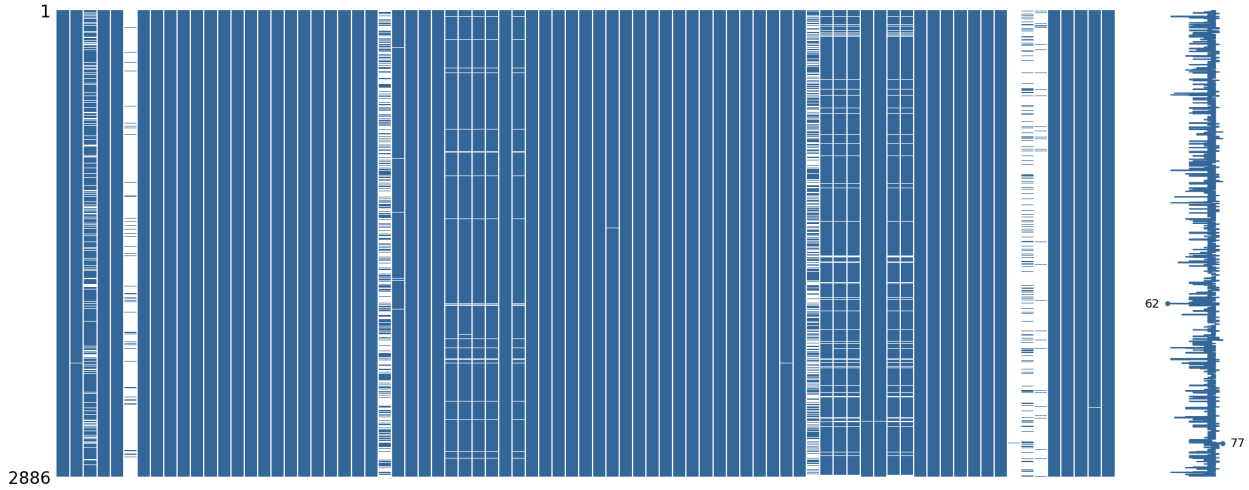
1.2.3. Supression des outliers

D'après la documentation de la base de données, il y a des valeurs aberrantes dans le jeu de données. On va donc les supprimer pour éviter qu'elles n'affectent la performance de notre modèle. On se basera sur le score IQR pour identifier les outliers. L'IQR mesure la dispersion statistique et est calculé comme la différence entre le troisième quartile (Q3) et le premier quartile (Q1) des données. Les valeurs situées en dehors des limites définies par [Q1 - 1.5 * IQR, Q3 + 1.5 * IQR] sont considérées comme des valeurs aberrantes. En appliquant cette méthode, nous avons identifié et éliminé 33 valeurs aberrantes.

1.2.4. Gestion des valeurs manquantes

PoolQC	Nombre total de données manquantes
MiscFeature	2786
Alley	2690
Fence	2322
MasVnrType	1750
FireplaceQu	1415
LotFrontage	476
GarageYrBlt	155
GarageFinish	155
GarageQual	155
GarageCond	155
GarageType	153
BsmtExposure	81
BsmtCond	81
BsmtQual	80
BsmtFinType2	79
BsmtFinType1	78
MasVnrArea	22
MSZoning	4
BsmtHalfBath	2

Variable	Pourcentage de données manquantes
PoolQC	99.792100
MiscFeature	96.534997
...	...
BsmtFinType1	2.702703
MasVnrArea	0.762301
MSZoning	0.138600
BsmtHalfBath	0.069930



On remarque qu'il y a pour certaines colonnes plus que d'autres, énormément de valeurs manquantes. Initialement, nous avions envisagé de conserver ces variables, supposant qu'elles pourraient contribuer, même de façon limitée, à expliquer la variabilité de SalePrice. Cependant, les analyses ultérieures ont démontré que l'exclusion de ces variables aux données incomplètes améliorait la performance de notre modèle. En conséquence, nous avons adopté une approche plus rigoureuse en éliminant les colonnes présentant plus de 100 valeurs manquantes. Pour les données manquantes restantes, nous avons opté pour une stratégie d'imputation adaptée au type de variable : la médiane pour les variables quantitatives et la classe mode pour les variables qualitatives. Cette démarche méthodique vise à maintenir l'intégrité de notre modèle sans compromettre sa capacité prédictive.

Variable	Nombre de valeurs manquantes avant imputation
BsmtCond	81
BsmtExposure	81
BsmtQual	80
BsmtFinType2	79
BsmtFinType1	78
Foundation	..
Heating	Ø
HeatingQC	Ø
CentralAir	Ø
SaleCondition	Ø

Des variables comme BsmtCond, BsmtExposure, BsmtQual, et d'autres, bien qu'elles contiennent des valeurs manquantes, n'atteignent pas le seuil critique de 100. Ces variables seront donc traitées avec notre stratégie d'imputation pour combler les NA.

2. Modélisation

2.1. Validation croisée et métriques

Nous allons tester plusieurs modèles de régression pour prédire le prix de vente des maisons. Nous utiliserons la validation croisée sur 10 plis pour mélanger les données et assurer une meilleure performance. Concernant les métriques, nous utiliserons l'erreur quadratique moyenne (RMSE) et l'erreur logarithmique quadratique moyenne (RMSLE), qui est plus adapté à notre variable transformée. Nous utiliserons également la part des prédictions correctes à plus ou moins 5% ainsi que le score obtenu sur Kaggle après avoir soumis les résultats par modèle. Ce "Score Kaggle" pourra être interprêter comme le score de généralisation, à d'autres données que celle d'entraînement.

2.2. Modèles de regression et paramètres

Concernant les modèles de régression utilisés, nous avons sélectionné une poignet de modèles pour tenter d'obtenir les meilleurs résultats. Pour tous les modèles, nous avons fait varier les paramètres de sorte à obtenir les meilleurs scores sur les données d'entraînement, tout en faisant en sorte qu'ils se généralisent à toutes les données.

- XGBRegressor : Algorithme d'arbre de décision optimisé par un gradient.
- LGBMRegressor : Algorithme d'arbre de décision optimisé par un gradient, avec des arbres de décision plus légers et plus efficaces pour les variables catégorielles.
- Ridge : Algorithme de régression basé sur la pénalisation des coefficients élevés.
- Lasso : Variante de la régression linéaire multiple, qui réduit le nombre de variables en appliquant 0 à leurs coefficients.
- GradientBoostingRegressor : Algorithme de construction d'arbres peu profonds, de manière séquentielle, en améliorant les arbres t avec les erreurs des arbres t - 1.
- RandomForestRegressor : Méthode ensembliste qui construit un grand nombre d'arbres de décision en parallèle et améliore la précision par la moyenne de leurs prédictions.
- SVR : Version pour la régression de la machine à vecteurs de support qui cherche à trouver la meilleure marge pour englober les points de données dans l'espace de caractéristiques.
- ElasticNet : Mix de la régression Ridge et Lasso, qui applique à la fois des pénalités à la Ridge et une réduction de variable à la Lasso.
- DecisionTreeRegressor : Algorithme d'arbre de décision classique.

2.3. Résultats

Modèle	Moyenne CV RMSE	RMSLE Ensemble Données	% Prédictions Correctes ±5%	Score Kaggle
XGBRegressor	0.128751	0.093918	45.550105	0.14181
LGBMRegressor	0.110928	0.069386	68.044849	0.12269
Ridge	0.110566	0.087891	50.735809	0.12254
Lasso	0.111788	0.087409	50.805886	0.12398
GradientBoostingRegressor	0.109919	0.073125	84.723196	0.12595
RandomForestRegressor	0.136842	0.086273	56.131745	0.15013
SVR	0.107887	0.088619	57.603364	0.12405
ElasticNet	0.111487	0.087443	51.016118	0.12364
DecisionTreeRegressor	0.176293	0.089538	50.105116	0.18888

Le SVR (Support Vector Regressor) se révèle être le modèle le plus performant sur le jeu d'entraînement, avec une Moyenne CV RMSE de 0.107887, la plus basse observée, indiquant une très grande précision dans les prédictions sur diverses subdivisions de l'ensemble de données. Cette valeur de 0.107887 signifie que, en moyenne, les prédictions du modèle s'écartent de seulement 10.7887% des vraies valeurs de vente des maisons, sur une échelle logarithmique. De plus, un RMSLE de 0.088619 souligne la compétitivité du modèle dans la prédiction des prix sur l'échelle logarithmique avec précision. Néanmoins, lorsqu'appliqué au jeu de test, le SVR montre une performance légèrement réduite avec un score Kaggle de 0.12405, ce qui révèle une diminution de la précision par rapport à l'entraînement et suggère un besoin de réajustements pour optimiser sa généralisation sur des données non vues.

Le DecisionTreeRegressor montre les performances les moins satisfaisantes sur l'ensemble d'entraînement, avec une Moyenne CV RMSE particulièrement élevée de 0.176293, ce qui indique que l'erreur moyenne de ses prédictions est considérable. En outre, il présente le RMSLE le plus élevé parmi les modèles testés, à 0.089538, soulignant sa difficulté à prédire avec précision les prix de vente sur une échelle logarithmique et à modéliser correctement les tendances des données. Ces faiblesses se traduisent par un score Kaggle de 0.18888, le plus élevé et donc le moins performant parmi les modèles évalués, confirmant ses limites dans la généralisation à des ensembles de données inédits. Cette valeur de 0.176293 pour la Moyenne CV RMSE révèle un écart

significatif entre les prédictions et les valeurs réelles, reflétant une capacité limitée du DecisionTreeRegressor à capturer la complexité du jeu de données et à fournir de bonnes prédictions.

Le GradientBoostingRegressor se démarque avec le pourcentage le plus élevé de prédictions correctes (84.723196%), indiquant qu'il prédit les prix des maisons avec une marge d'erreur très serrée.

En examinant les scores Kaggle, le LGBMRegressor a le meilleur score de test (0.12269), le rendant potentiellement le meilleur modèle en termes de généralisation sur des données non entrainées, malgré des performances légèrement inférieures en termes de RMSE et de RMSLE sur l'ensemble d'entraînement par rapport au SVR. Cela souligne l'importance de tester les modèles sur un ensemble de données distinct pour évaluer leur capacité à généraliser.

3. Conclusion

Le choix du "meilleur" modèle dépend de l'équilibre souhaité entre les différentes métriques d'évaluation. En considerant le score Kaggle comme la mesure finale de la performance, le LGBMRegressor serait le modèle de choix et permet d'obtenir la **323e place** sur la compétition. Il est le compromis parfait entre minimisation des erreurs et généralisation.