

Prédire le coût de logement de la Californie



2024

de Gunst

Une introduction au Machine Learning

Pour beaucoup d'experts de la donnée, la prédiction des prix de logement fut un point d'entrée dans le vaste monde du machine learning. Que ce soit sur le site kaggle.com, qui rassemble la plus grande communauté de data scientists au monde, ou le livre emblématique sur le machine learning d'Aurélien Géron, l'estimation de la valeur immobilière en Californie aux États-Unis est proposé comme premier projet à réaliser pour les novices.

Même dans la formation : BUT Science des Données, ce projet est notre première introduction au machine learning et à l'intelligence artificielle qui prend de plus en plus d'ampleur dans les entreprises ainsi que dans le monde académique. Ce rapport se présente alors comme étant un guide pour connaître et comprendre les étapes nécessaires à la constitution d'un modèle de machine learning. Mais avant tout voici une définition de ce dernier terme :

Le machine learning est le domaine d'étude qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmés. (Arthur Samuel 1959)

Cette définition est une façon élégante pour dire que l'ordinateur définit par lui-même des règles pour essayer de prédire ou de définir une certaine caractéristique telle que la valeur foncière d'un bien. Un algorithme prédit cette caractéristique sans biais à l'inverse de l'humain, même si les biais humains hantent encore les algorithmes qui sont derrière nos écrans.

Sommaire

01

Connaître et comprendre le marché du logement de la Californie

02

Analyse de chacune des variables

03

Test d'ajustement à une loi normale

04

Croisement des variables

05

Analyse en composantes principales

06

Préparation des données pour les modèles de machine learning

07

Les modèles de machine learning

08

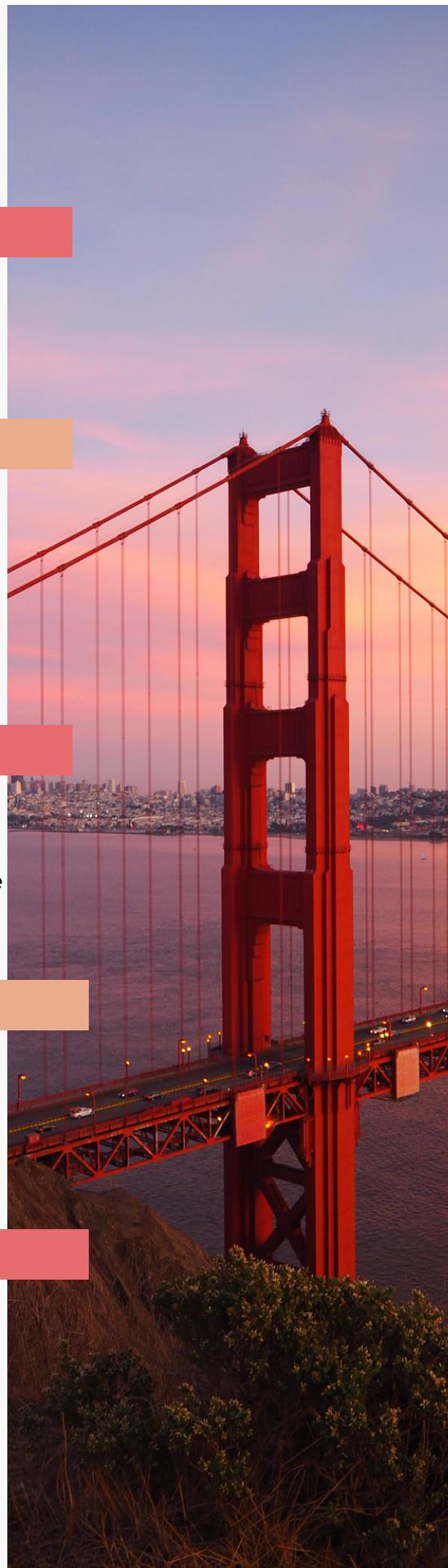
les réseaux de neurones

09

Optimisation des meilleurs modèles

10

Validation des modèles finales

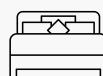


I. Connaitre et comprendre le marché du logement de la Californie

	longitude	latitude	age_median	pieces	chambres	population	menages	revenu_median	valeur_mediante
0	-119.60	36.57	33	1923	403	1205	389	1.8333	23211
1	-119.80	36.75	52	1788	449	1156	418	1.7298	61629
2	-118.22	34.10	33	1903	386	1187	340	4.0469	114033
3	-117.04	32.98	16	1332	196	640	193	6.0226	180869
4	-118.06	33.72	14	2665	331	964	319	15.0001	416539
5	-122.28	37.53	34	1980	385	970	391	5.1207	163798
6	-117.26	33.20	13	3163	725	1675	629	2.8214	95937
7	-122.45	37.70	16	6457	1336	4375	1231	5.1788	136992
8	-117.74	33.46	9	6564	1316	1720	904	4.89	177192
9	-122.81	38.36	18	2399	389	1131	391	5.2769	195755

La première étape pour créer un bon modèle prédictif est la connaissance des données. Il faut repérer les données qui nous ont été fournies. Plus précisément, il faut comprendre les types de données, leur dispersion et traiter les données manquantes. Les données qui nous ont été fournies contiennent huit différentes caractéristiques sur les logements regroupés en îlots de maisons. De plus, toutes les variables sont numériques. Voici les informations dont on disposait pour les 20 milles îlots en Californie :

- des données géographiques :
 - la longitude
 - la latitude
- l'âge médian
- le nombre de pièces total
- le nombre de chambres total
- le nombre d'habitants



- le nombre de ménages dans l'îlot,
- le revenu médian
- la valeur foncière médiane



	longitude	latitude	age_median	pieces	
count	20300.000000	20300.000000	20300.000000	20300.000000	20300.000000
mean	-119.557069	35.612233	28.651872	2636.252463	2636.252463
std	2.002971	2.127369	12.601434	2184.214228	2184.214228
min	-124.350000	32.540000	1.000000	2.000000	2.000000
25%	-121.800000	33.930000	18.000000	1451.000000	1451.000000
50%	-118.480000	34.250000	29.000000	2127.000000	2127.000000
75%	-118.000000	37.700000	37.000000	3143.250000	3143.250000
max	-114.310000	41.950000	52.000000	39320.000000	60000.000000

Lorsqu'on s'intéresse aux **statistiques clé** de nos variables, on voit qu'en moyenne les logements ont été construits il y a **29 ans** et ont une valeur de **130 milles dollars**. On voit également que les données sont très **homogènes** au niveau de la taille des îlots. Tandis qu'en moyenne, un îlot contient 500 ménages, l'écart-type est de 380, indiquant que notre jeu contient à la fois des îlots de grande et de petite taille.

Le dernier élément clé pour comprendre notre jeu de données est la gestion des **valeurs manquantes**. En effet, dans notre fichier de données comme tout autre, certains valeurs sont manquantes et c'est à la personne qui effectue l'analyse de décider quant à la gestion de ces données, les options les plus communes sont :

- remplacer les valeurs par la moyenne
- supprimer les lignes contenant une valeur manquante
- les garder comme étant des valeurs manquantes

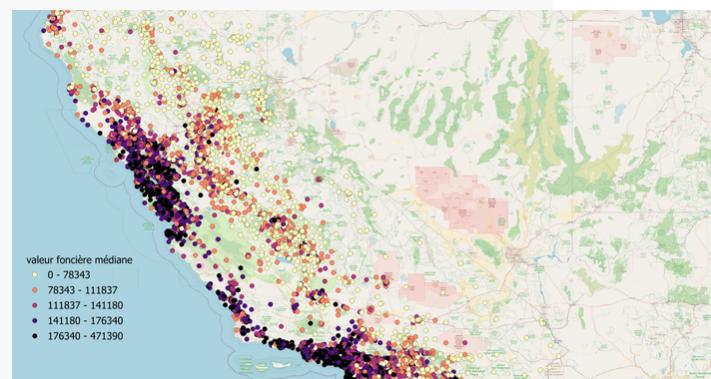


Dans notre cas, nous avons remplacé les informations absentes par la valeur foncière moyenne.

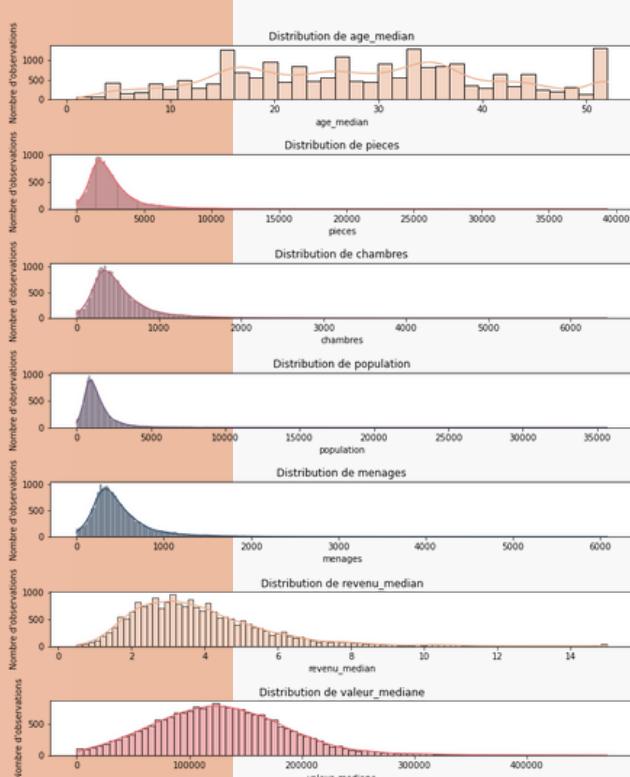
II. Analyse des variables

Comprendre les données avec lesquelles on travaille est extrêmement important. Pourtant, si on ne visualise pas graphiquement nos données et on les voit uniquement sous forme de tableau, on n'arrivera jamais à cerner les informations qui se cachent derrière les valeurs chiffrées. Pour illustrer cela, nous connaissons la géolocalisation des logements grâce à la longitude et la latitude.

Pour autant, très peu de personnes pourront affirmer avec certitude qu'il s'agit de la Californie. Ce n'est qu'en montrant les informations sur une carte qu'on voit qu'il s'agit de la Californie. De plus, on voit que, plus on est proche de la côte, plus on paie pour un logement.



Distribution des autres variables



Des logements de tout âge

Les logements de la Californie peuvent à la fois être neuves et plutôt agées, mais il n'y a pas d'îlots avec un âge au-dessus de 52 ans. La plupart des logements datent des années 2000.



Infesté par des valeurs extrêmes

Notre jeu de données est infesté par des valeurs extrêmes, notamment quant aux variables qui décrivent la taille des îlots et des logements. Le nombre de ménages pour chaque îlot est concentré au tour des 500, mais il y a quelques îlots regroupant plus de 6 000 logements.



La valeur médiane : ça cloche

Quant à la valeur médiane des îlots, on observe que sa distribution ressemble à une cloche ce qui est propre à la distribution d'une loi normale. Il serait donc intéressant de s'intéresser à la normalité de la valeur médiane.

III. Test d'ajustement à une loi normale

Nous allons donc tester si la valeur médiane suit bien une loi normale. Pour comprendre pourquoi ce test d'ajustement est primordial à la conception d'un modèle. Voici donc une courte description de l'utilité de la loi normale :

La loi normale est particulièrement utile pour modéliser des données où les valeurs tendent à être réparties de manière symétrique autour d'une moyenne centrale. Il en résulte que l'identification des valeurs aberrantes devient plus facile.

Pour vérifier la normalité de la valeur médiane, on peut effectuer un test de Shapiro-Wilk. On suppose alors que notre phénomène suit une loi normale. En effectuant ce test sur l'ordinateur, nous pouvons obtenir la probabilité (p-valeur) que la valeur foncière suit une loi normale. Il faut donc fixer un seuil d'acceptation de la normalité qui sera ici de 95 %. Nous obtenons les résultats suivants :

loi normale :

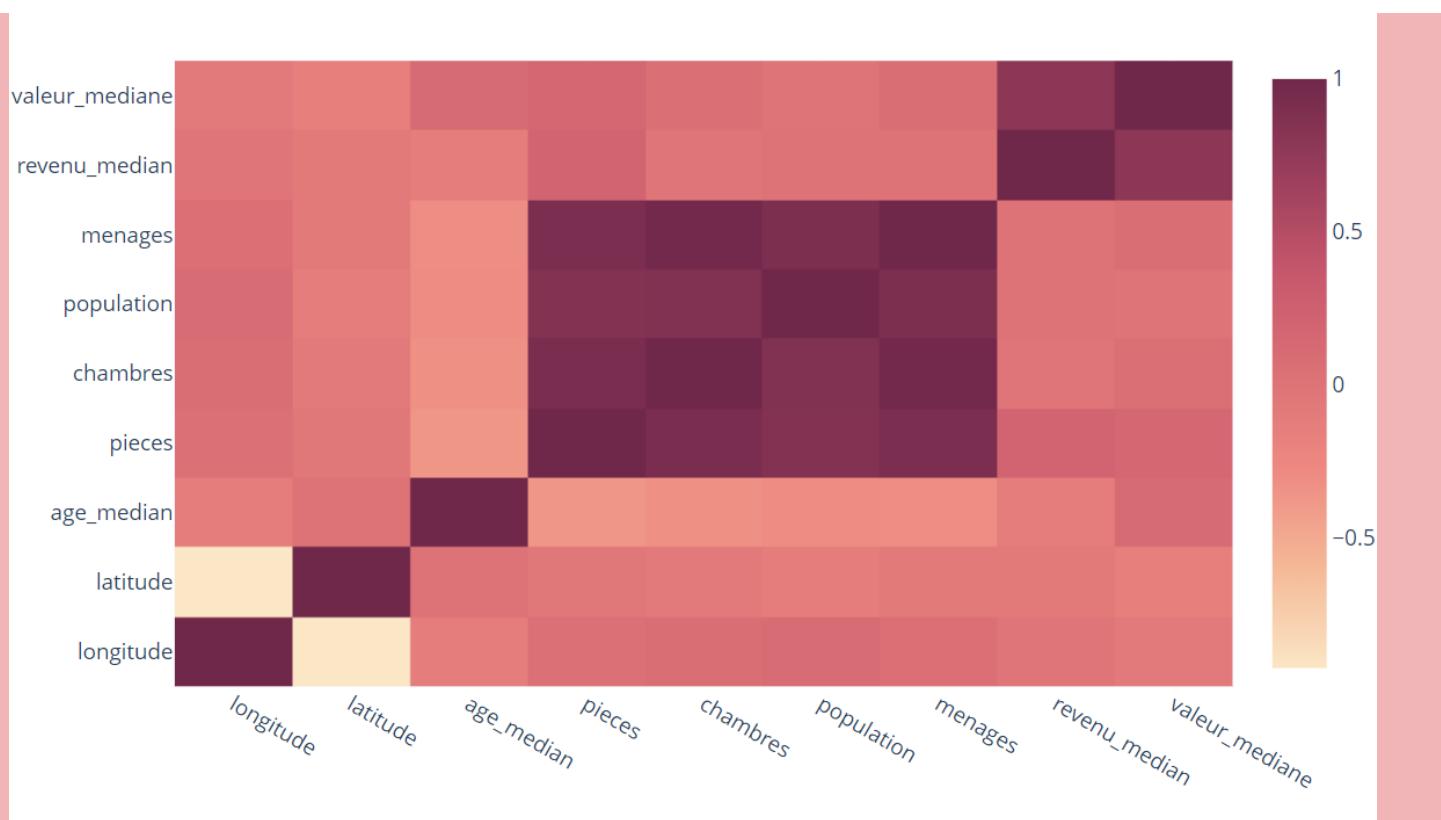
```
La statistique est de : 0.9948374032974243 et la p-valeur de :1.0502487074459665e-10
```

La probabilité est extrêmement faible malgré le fait que le graphique ressemble à une cloche. Même en tirant un échantillon ou en éliminant les valeurs extrêmes, le test échoue.

Cet échec nous indique que nos préjugés sur les données sont parfois faux et montre l'utilité du machine learning qui n'a aucun jugement vis-à-vis des données.

IV Croisement des variables

Pour rappel, notre but est de prédire la valeur foncière à l'aide des autres variables dont nous disposons. Il faut donc un lien entre ces variables et la valeur foncière. Comme toutes les données sont qualitatives, nous pouvons facilement mesurer l'intensité du lien entre deux variables grâce au coefficient de corrélation. Ce coefficient est compris entre -1 et 1, si le coefficient s'approche de 0, nos variables n'ont aucune liaison. Cela serait catastrophique pour la prédition. À l'inverse, si le coefficient s'approche de 1, alors les deux variables varient dans le même sens (si x augmente, alors y augmente). L'inverse se produit lorsque le coefficient est proche de -1 (si x augmente, y diminue). Heureusement, nous n'avons pas besoin de calculer ces coefficients à la main et nous pouvons les visualiser avec une carte de chaleur :

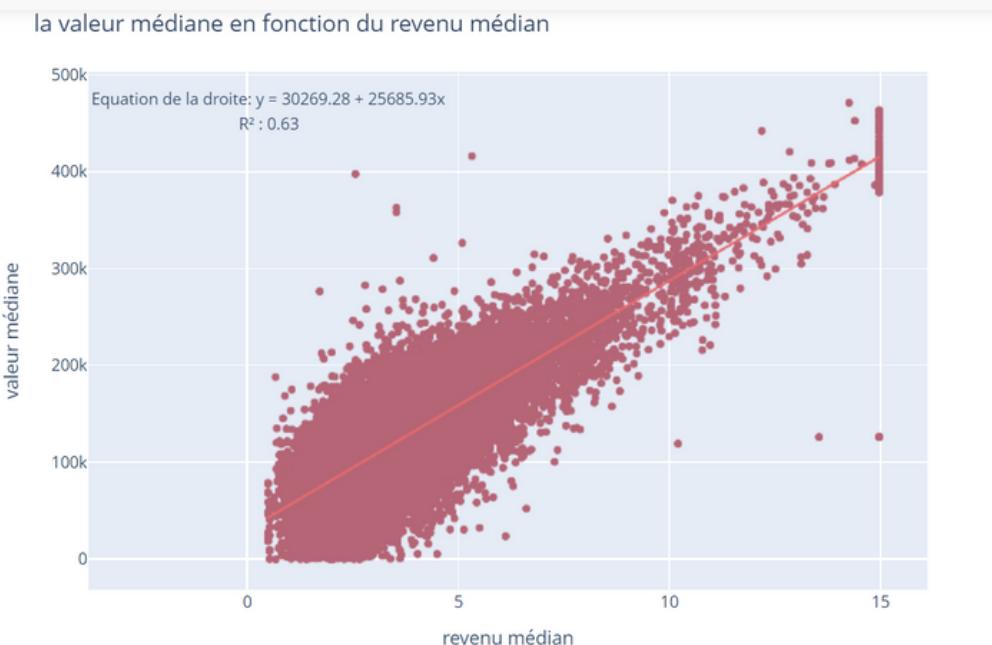


[Visualiser le graphique interactif](#)



Ca explique !

Les variables semblent être plutôt liées à la valeur médiane. Notamment, le revenu médian a un coefficient de corrélation de 0.8. Cela indique donc un lien fort entre les deux variables. On peut donc essayer d'appliquer une régression linéaire simple :



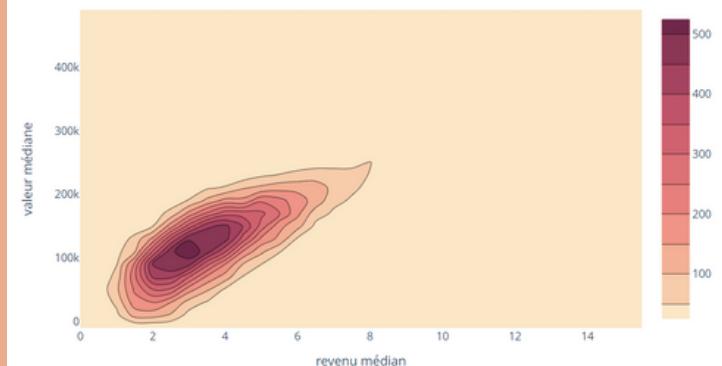
Un modèle simple est déjà plutôt adapté pour prédire la valeur médiane. On pourrait donc se contenter de ce modèle. Or, nous allons essayer d'obtenir des meilleurs modèles qui pourraient possiblement expliquer pourquoi certains points sur le nuage de points ci-dessus se trouvent au dessus ou en dessous de la droite de régression.

Concentration des différents valeurs médianes en fonction du revenu médian



les maisons sont comme le jus d'orange : concentrées

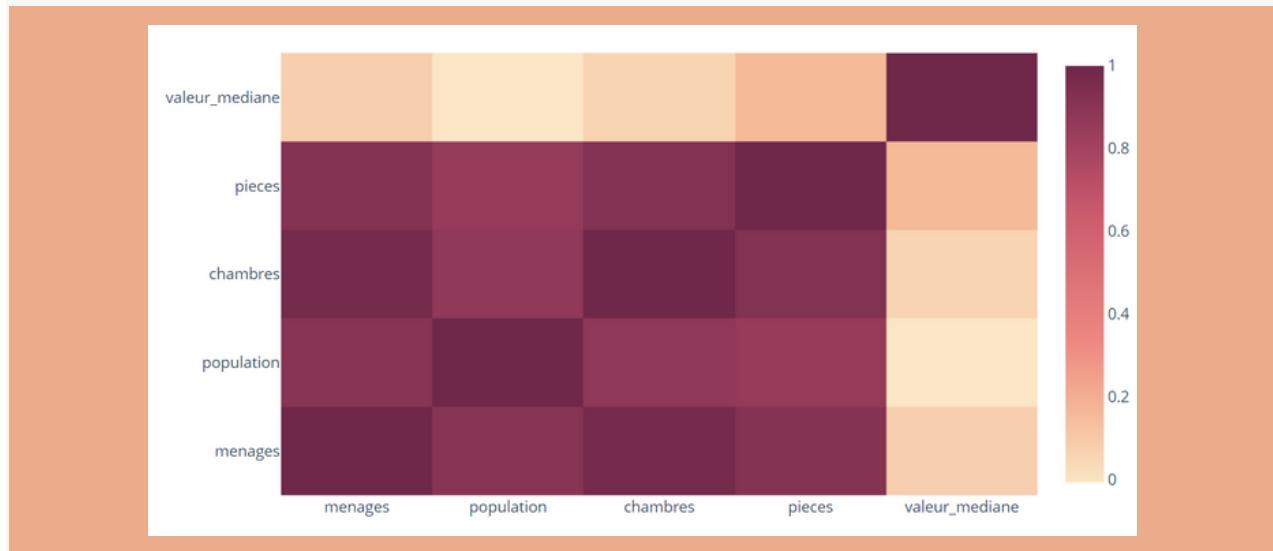
En effet, même si on arrive à bien expliquer la dispersion de la valeur médiane à l'aide du revenu, des variations plus fines autour de la droite de régression ne peuvent être pris en compte par une unique variable. Il faut donc faire appel aux autres caractéristiques dont nous disposons.



Plus c'est grand, plus c'est lié

Sur la carte de chaleur présentée sur la page d'avant, on peut voir que des variables indiquant la taille des îlots et des maisons : menages, population, chambres et pièces sont très fortement corrélées (coefficients au-dessus de 0.8) et risquent donc de saturer nos futurs modèles avec plus de dimensions qui n'apportent pas forcément des nouveautés à notre modèle. On peut donc réduire le nombre de variables, favorisant ainsi la vitesse de calcul.

V. Analyse en composantes principales



Les variables : pièces, chambres, population et ménages ont des liaisons fortes entre elles, mais ne sont pas fortement liées à notre variable cible qui est la valeur médiane. Que faire ?

Comme ces informations sont très proches, proposer les quatre variables tels quels à notre modèle prédictif apporte peu de ressources pour prédire la variable cible. Pour autant, il existe une méthode pour réduire le nombre de variables sans pour autant perdre trop d'informations : l'Analyse en Composantes Principales (ACP).

En effet, cette technique permet d'utiliser l'unité de mesure de lien entre deux variables qui est la covariance pour en tirer les informations principales, comme nous avons quatre variables, nous pouvons constituer une matrice de covariance.

Cette matrice est ensuite décomposée en vecteurs propres et en valeurs propres. Les vecteurs propres représentent les directions dans lesquelles les données varient le plus, tandis que les valeurs propres quantifient l'importance de cette variation dans chaque direction. L'ensemble est appelé composantes principales. En choisissant le nombre de vecteurs et valeurs propres à garder, nous pouvons restituer la plus grande quantité d'information possible en peu de variables.

Dans le cas de l'immobilier en Californie, nous garderons les deux premières composantes pour restituer 97,3 % de l'information en deux variables et non quatre, favorisant ainsi la vitesse de calcul de nos futurs modèles.

VI. Préparation des données pour les modèles de machine learning

Malheureusement, le fait d'avoir mis en œuvre un ACP complique également les choses pour les modèles. En effet, maintenant, que nous connaissons bien notre jeu de données, il est, enfin, temps de préparer les données pour construire des modèles et à la fin, on aura douze tableaux pour quatres jeux différents. En machine learning, il faut deux, voir trois différents tableaux (dataframes) fait à partir d'un jeu de données :

- Le tableau train, le plus volumineux en général, pour entraîner le modèle et faire en sorte que le modèle trouve les bonnes relations qui se généralisent aux autres données au sein de ce tableau.
- Un tableau test. Ce tableau va permettre de tester notre modèle et ainsi voir si les règles définissent par notre modèle sont adaptés aux autres données. Si le modèle a une mauvaise performance sur ce tableau, il faut essayer d'optimiser le modèle en changeant des paramètres pour s'assurer que le modèle est bon.
- Le tableau de validation sur lequel on valide le modèle. On teste, en général, uniquement notre modèle après être sûr que le modèle ait une très bonne performance sur le tableau test. Dans notre cas, ce tableau a été fourni quelques jours après le début du projet.



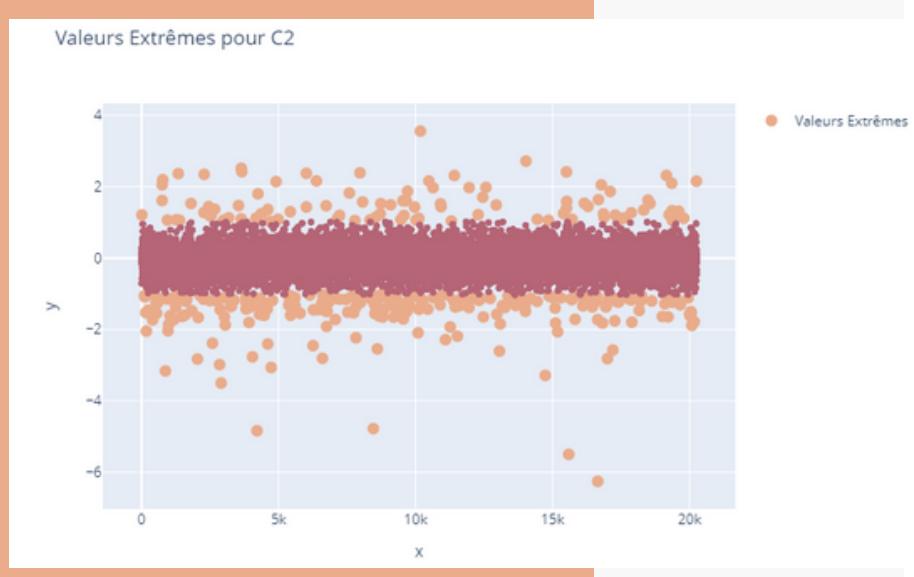
En plus de cette séparation, nous allons construire nos modèles sur trois groupes de données différents car le fichier de base est infesté par des valeurs extrêmes. Étant conscientes de ce dernier, nous allons séparer notre fichier en trois différents groupes :

- Avec les valeurs aberrantes. Ce groupe de données va permettre à nos modèles prédictifs de voir si nos modèles ne sont pas facilement influencés par des îlots de maisons pour lesquels les prix semblent anormaux vis-à-vis des caractéristiques connues.
- Sans valeurs inhabituelles. On supprime des îlots pour lesquels les données dépassent un certain seuil. Au-delà de ce seuil, on les considère hors norme et on les supprime.

- un dernier groupe pour lesquels les valeurs aberrantes sont “corrigées”. C'est à dire que nous allons conservées ces valeurs mais nous les ramenons plus proche de la normalité et des autres valeurs. Cette technique permet de garder le plus de données possibles sans pour autant trop affecter notre modèle avec les valeurs extrêmes. On appelle ce dernier phénomène le sous-ajustement.

Comment identifie-t-on alors ces anomalies? Il n'y a, bien sûr, aucune recette magique permettant de dire qu'un îlot est bizarre. Pour les identifier, nous utilisons le z-score. Cette technique attribue à chaque îlot un score en fonction du nombre d'écart-type que l'îlot est éloigné de la moyenne. Si une variable se trouve exactement trois écarts-types de la moyenne, alors son z-score sera de trois.

Dans le cas de l'immobilier en Californie, nous estimerons qu'un z-score au-dessus de trois est hors norme. On peut donc facilement identifier nos outliers :



Après l'identification de ces valeurs, nous pouvons créer sans problèmes le groupe sans anomalies. En revanche, pour le groupe corrigé, nous allons utiliser le Robust scaler de SKLearn qui permet de soustraire aux valeurs aberrantes la moyenne puis de les diviser par l'écart entre le premier quartile et le troisième quartile soit la différence entre les bornes qui contiennent la moitié des valeurs.

Nous appliquons donc ces techniques à toutes les variables et presto ! Nous avons nos données pour réaliser les modèles. Nous avons donc les données sans, ou avec ACP, trois groupes et des tableaux train et test pour un total de 12 tableaux. Chacune des variables dans les tableaux est ensuite normalisée, c'est-à-dire qu'on soustrait la moyenne puis on divise par l'écart-type pour que toutes les variables soient du même ordre de grandeur, facilitant ainsi le calcul pour nos modèles.

VII. Les modèles de machine learning

Dans cette partie, le but est de tester différents algorithmes ou modèles afin d'identifier lequel ou lesquels donnent les meilleurs résultats. Pour mesurer la précision des résultats, il faut mettre en place des indicateurs qui vont permettre de comparer les modèles. Dans notre cas, nous allons utiliser deux indicateurs :

- Le R² ou coefficient de détermination : cet indicateur nous indique la part de la dispersion ou de la variance qui est mesurée par notre modèle. Nous pouvons donc savoir si le modèle s'adapte bien au marché de l'immobilier en Californie. Nous cherchons donc à maximiser ce R² le plus possible.
- Le s² ou le carré de la moyenne des erreurs : mesure la distance ou la différence qui sépare la valeur prédictive de la valeur réelle et le met au carré afin d'éliminer les signes. Cet indicateur permet de mesurer la précision et nous cherchons à rendre cette valeur la plus faible possible.

Nous allons donc entraîner les modèles avec les tableaux train et mesurer les performances des modèles sur le jeu de test. Le data scientist doit ensuite, avec son expertise et ses connaissances sur le jeu de données, identifier les différents algorithmes à tester. Voici lesquels nous allons tester sur l'immobilier en Californie :

- Une régression linéaire multiple  : Ce modèle est probablement le plus basique et ne fait rien d'autre que tracer une droite qui s'ajuste au mieux aux points des données d'entraînement.
- Un arbre de décision  : Il faut imaginer cet algorithme comme étant un arbre constitué de branches, chaque branche représente une décision à prendre en fonction de l'îlot et de ses caractéristiques. Par exemple : Si l'îlot a un revenu médian élevé, est-ce que la valeur de base (qui est la moyenne) augmente ? Si c'est le cas, de combien ? Cet ensemble de branches permet d'obtenir une prédiction de valeur foncière unique à chaque ensemble de caractéristiques.

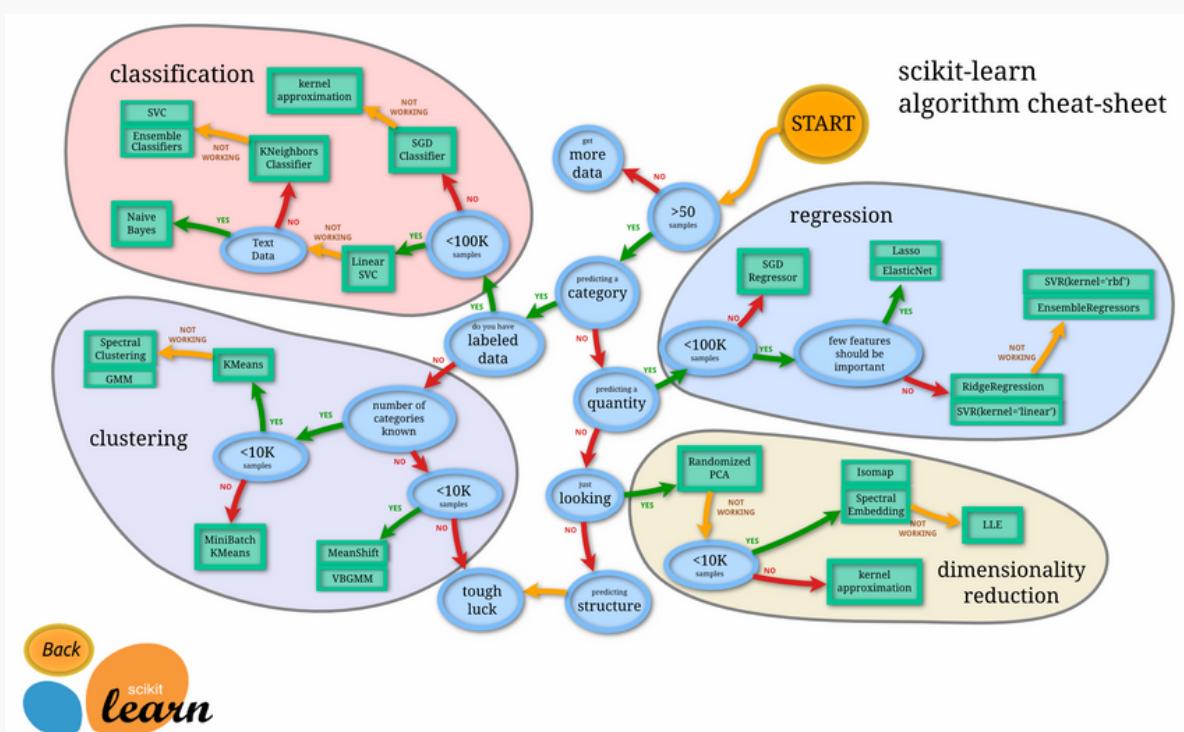
- Les forêts aléatoires  : Cet algorithme n'est rien d'autre qu'un **ensemble d'arbres de décisions** réunis. Chaque arbre étant différent, la moyenne des résultats de chaque arbre va permettre d'obtenir une valeur foncière qui sera probablement plus proche de la réalité, car nous disposons de **plusieurs estimateurs**.
- Le GBM ou Gradient boosting machine  : Un modèle de GBM est **itératif**. En effet, ce modèle part d'un autre modèle basique tel que la régression linéaire et entraîne un **premier modèle sur les données d'entraînement**, puis un **deuxième modèle** non pas sur les données, mais **sur les erreurs du premier modèle** puis un troisième modèle sur les erreurs du deuxième modèle, etc... En ajoutant les résultats de ces modèles, les prédictions sont **plus fiables**, mais pourraient trop s'ajuster aux données d'entraînement. Ce qu'on appelle le **surajustement**.
- Le SVM ou support vector machine  : Cet algorithme n'est pas tellement différent de la régression linéaire multiple. Or, en plus de tracer une **droite de régression**, le support vector machine va définir une **marge de tolérance** qui est une zone autour de la droite. Le modèle essaie alors de minimiser le nombre d'instances en dehors de cette zone.
- La régression ridge  : Encore une fois, un modèle de régression ridge est comme un modèle de régression linéaire multiple, mais avec des **pénalités pour certaines variables** et leur **importance dans le modèle** qui va permettre à l'algorithme de ne pas trop s'adapter au tableau "train" pour que le modèle soit plus généralisable sur les autres tableaux.
- La régression lasso  : A l'inverse de la régression ridge, la régression LASSO (Least Absolute Shrinkage and Selection Operator) utilise la pénalité pour rendre **l'importance de certaines variables nulles**. Ainsi, le modèle sélectionne automatiquement les variables les plus importantes et intéressantes pour le modèle.
- Le XGBoost  : Le XGboost est un algorithme proche du GBM mais il utilise des **techniques de régression ridge et lasso** pour ne pas trop s'ajuster aux données d'entraînement.

Les meilleurs algorithmes pour prédire la valeur foncière semblent alors être :

- la RLM
- le SVM
- le GBM
- le ridge regression

Ces algorithmes sont les plus puissants pour la prédiction de la valeur foncière, car ils expliquent à la fois bien le phénomène (R^2 de 0.8 environ) et ils ont un bon pouvoir prédictif (environ 0.2 d'erreur au carré).

De plus le jeu de données avec les outliers après avoir “corrigé” les valeurs semble être le plus puissant, nous allons utiliser ce jeu de données pour optimiser les modèles. L'ACP a été assez intéressant, car le R^2 baisse très peu, nous allons donc essayer d'entraîner le meilleur modèle sur les données avec l'ACP. Or, avant cela, nous nous intéresserons à des techniques plus avancées de l'intelligence artificielle qui sont les réseaux de neurones.



Formulaire de SciKit Learn pour le choix du meilleur modèle

VIII les réseaux de neurones

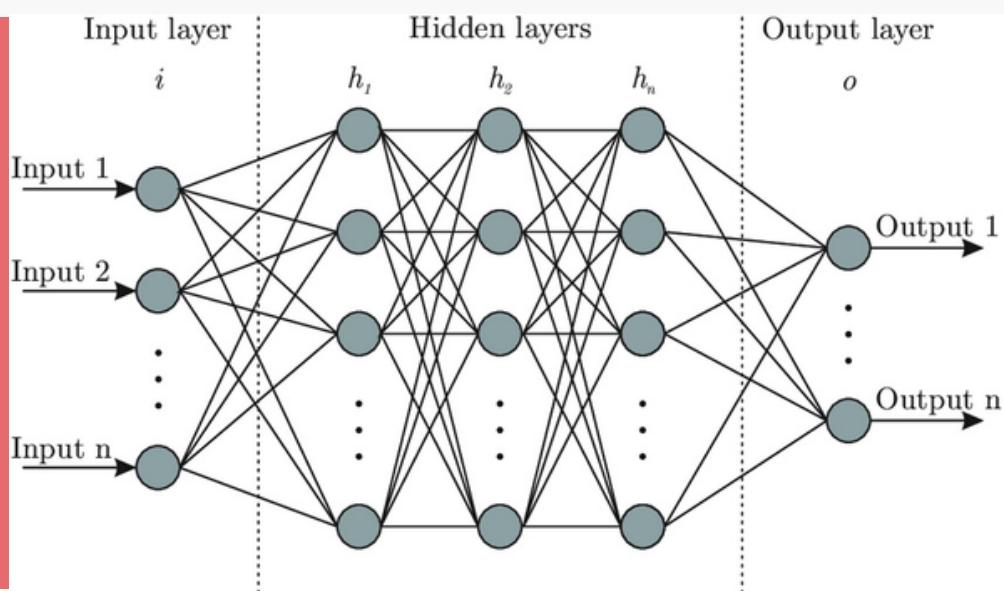
Depuis le début de l'humanité, les humains ont essayé de copier les merveilles de la nature afin de pouvoir en bénéficier nous-même. Les humains ont vu les oiseaux et ont passé des siècles à faire des avions capables de nous transporter. La biologie s'intéresse également au cerveau et les milliards de neurones qui font que nous sommes capables de réfléchir. Tout à fait comme les statisticiens qui copient depuis les années quarante les neurones pour en faire des modèles.

En effet, les réseaux de neurones font partie des modèles les plus puissants qu'on a à notre disposition. Même si, auparavant, il n'y avait pas assez de données pour alimenter les modèles, l'arrivée des grandes quantités de donnée grâce au monde connecté ont fait que les réseaux de neurones sont devenus puissants et omniprésents dans l'intelligence artificielle.

Un réseau de neurones est bien sûr, constitué de **neurones ou unités** qui sont en réalité des **formules mathématiques** simples ou complexes. Ces neurones sont connectés entre eux formant ainsi un réseau puissant. Les **réseaux** sont en général constitués de trois couches :

- Une couche d'entrée : c'est la couche qui reçoit les données en entrée. Chaque neurone dans cette couche représente une caractéristique ou une dimension des données d'entrée. Les neurones peuvent ensuite transformer la donnée avant de la transmettre à la couche suivante.
- Le ou les couches cachées : Ce sont les neurones qui font le plus de traitements mathématiques. Dans notre cas, ce sont ces couches qui vont décider de la valeur **foncière** en fonction des caractéristiques. Chaque neurone dans une couche cachée est connecté à tous les neurones de la couche précédente et de la couche suivante.
- La couche de sortie: Dans cette couche, chaque neurone produit une valeur. Dans notre cas, nous essayons de prédire qu'une seule valeur qui est le prix médian d'une maison. Il en résulte que nous n'aurons qu'un seul neurone en sortie, mais elle est clé à la validation de notre modèle.

En effet, la puissance d'un réseau est mesurée par la sortie du réseau en minimisant le résultat de la fonction de perte tel que l'erreur quadratique (s^2). Ainsi, pendant l'entraînement, les neurones ou formules mathématiques sont mis à jour.



Visualisation d'un réseau de neurones

Il existe plein de différents types de réseaux. Pour prédire le prix immobilier, nous allons en tester quatre :

- Le feed-forward : C'est le plus simple des quatre, les données rentrent dans le réseau, passent par la couche cachée puis un certain montant est prédit et donné par la couche de sortie, et cela, sans boucler dans certains couches.
- Le convolutionnel : les réseaux de neurones convolutionnels (CNN) sont souvent utilisés pour traiter des données de forme matricielle telles que des images, des séquences temporelles ou des données spectrographiques. Les CNN sont particulièrement efficaces pour extraire des caractéristiques spatiales et temporelles des données. Malgré cet optimisation pour des données visuelles, nous allons essayés de l'appliquer à l'immobilier de la Californie.
- Le récurrent : Les réseaux de neurones récurrents (RNN) sont optimisés pour les données séquentielles ou temporelles. Chaque neurone se souvient alors des résultats antérieurs proches pour prédire le suivant.
- Les LSTM (Long Short-Term Memory) sont une variante spécifique de réseaux de neurones récurrents, conçue pour mieux capturer et conserver les dépendances à long terme dans les séquences de données. En effet, ce type de réseau cherche à se souvenir des prédictions fait dans le passé lointain.

Dans notre cas, c'est ce dernier type de réseau le plus efficace avec un R^2 de 0.81. Nous pouvons donc passer à l'optimisation des modèles.

IX Optimisation des meilleurs modèles

L'optimisation ou **fine-tuning** d'un modèle consiste à trouver les paramètres pour lesquels le modèle a les meilleurs résultats pour une tache spécifique. Cette optimisation est très coûteuse en termes de calculs. C'est pour cela que nous allons restreindre l'optimisation à trois modèles.

- L'optimisation de la **régression linéaire** est assez simple. Comme le modèle est très basique et ne trace qu'une seule droite, nous pouvons uniquement éléver les caractéristiques à une puissance pour que le modèle capture des relations possiblement non-linéaires. Dans notre cas, cela semble être efficace, car en élevant nos **variables au carré**, notre coefficient de détermination monte à 0.825 et l'erreur quadratique est de 0.17.
- Le fine-tuning de la **SVM** a été beaucoup plus coûteux en termes de calculs. Mon ordinateur a passé près d'une heure pour trouver le meilleur paramétrage. En créant une matrice de paramètres à tester, la fonction grid search va essayer toutes les combinaisons de paramètres pour trouver le meilleur modèle. Pour l'immobilier, le coefficient de détermination s'élève à 0.83 et une erreur de 0.17. Ce sont donc des résultats assez décevants pour autant de calculs.
- L'optimisation du réseau de neurones est quasiment identique à celui d'un SVM, mais il n'y a pas de fonctions telles que le grid search de SKLearn pour trouver le meilleur modèle. Il faut donc l'implémenter à la main. Les résultats sont encore plutôt décevants avec un R^2 de 0.83 et un s^2 de 0.16.



X. Validation des modèles

Après tout ce travail, nous pouvons enfin valider notre modèle sur le jeu de validation auquel nous avons à présent pas touché. Il est généralement recommandé de ne pas interagir avec le jeu de validation pour ne pas biaiser notre modèle avec des préjugés humains. Malheureusement, étant humain, notre cerveau capte très facilement des liens qui se généralisent mal notamment sur les données. En regardant le jeu de validation, nous pouvons inconsciemment essayer d'optimiser les prédictions uniquement sur ce jeu de données.

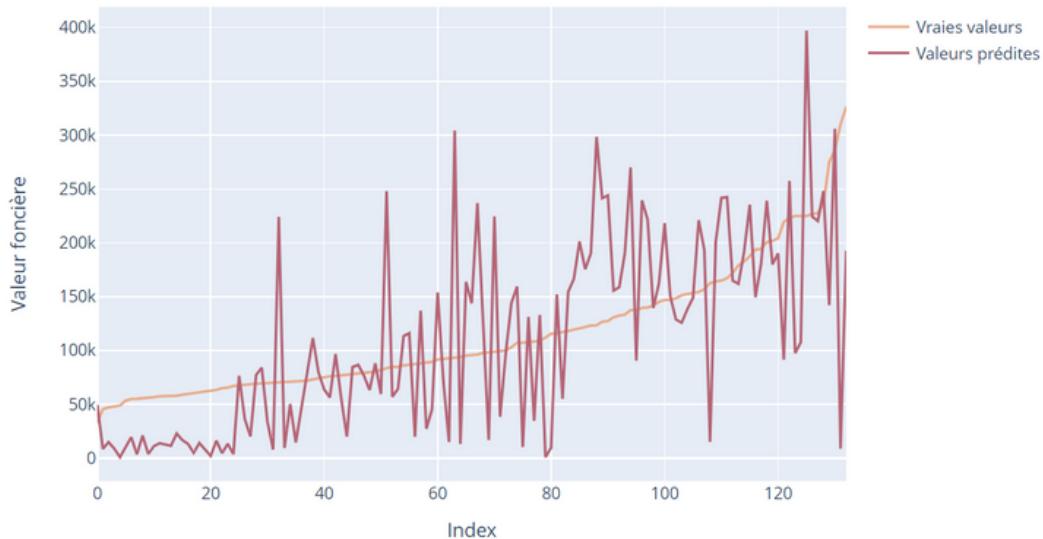
Après l'import du fichier, nous corrigions les données pour ensuite séparer les variables explicatives de la variable cible et nous testons ensuite nos trois modèles sur les données de validation. Or, on s'aperçoit que les données sont remplies de valeurs aberrantes qui ne représentent pas du tout la réalité. Voici les résultats de chacun des trois modèles :

Modèle	Coefficient de détermination R ²	Erreur moyenne s ²	Coefficient de correlation
Modèle linéaire	0.33	1.12	0.59
SVM	0.3	1.18	0.58
Réseau LSTM	0.39	0.9	0.66

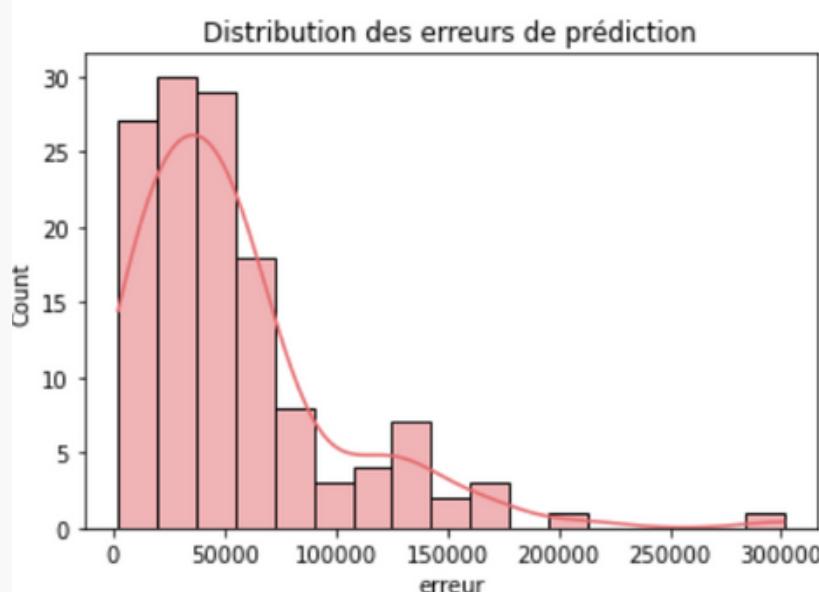
Les résultats sont très mauvais. Seul le réseau de neurones semble prédire assez correctement les valeurs des îlots de validation même si on n'arrive qu'à expliquer 39 % de la dispersion totale. Or, ces valeurs sont encore corrigées par le robust scaler et ne sont pas très parlant.

Après avoir inversé la transformation du robust scaler, on observe que les erreurs de notre modèle sont catastrophiques. Notre modèle prédit des valeurs foncières négatives, ce qui est impossible. Nous allons donc mettre les prédictions en valeur absolue.

Comparaison des valeurs médianes réelles et prédictées



Nos valeurs prédites semblent s'éloigner de la réalité plutôt que de s'en approcher. En moyenne, notre modèle prédit une valeur foncière à 54 000 dollars du vrai prix de vente. On peut donc se demander si le modèle n'est pas surajusté au jeu d'entraînement.



Or, lorsqu'on s'intéresse à la distribution des erreurs de prédictions, on voit que les erreurs restent relativement faibles avec la plus grande erreur étant de 300 mille dollars, mais la plupart des erreurs restent en dessous de 75 milles dollars.

Même si le réseau de neurones LSTM , la SVM et la régression linéaire ont du mal à prédire les prix des maisons hors norme, ce sont des modèles qui semblent être plutôt généralisables sur le marché de l'immobilier en Californie.

Conclusion

Pour conclure, dans ce rapport, nous avons vu une multitude de techniques de machine learning qui sont utilisées partout dans le domaine du machine learning ainsi que les difficultés et pièges à éviter. C'est ce qui fait du marché de l'immobilier de la Californie, un excellent point d'entrée pour les débutants.

Or, il existe plein d'autres utilités quant au machine learning. On peut par exemple faire de la classification ou du clustering qui sont des techniques très intéressantes que nous allons étudier par la suite au sein du BUT Science des Données.

