**Casablanca House Price modeling**

**Projet statistique**



Réalisé par :

**Walid Choukri**

Table des matières

[I. Vue générale & Problématique 2](#_Toc100958626)

[1. Contexte général du projet 2](#_Toc100958627)

[II. Data sourcing 2](#_Toc100958628)

[1. Data scrapping 2](#_Toc100958629)

[2. Process du scrapping 2](#_Toc100958630)

[III. Visualisation première du dataset 3](#_Toc100958631)

[1. Features description 3](#_Toc100958632)

[2. Défauts déduits de la visualisation 4](#_Toc100958633)

[Box plots : 4](#_Toc100958634)

[Bivariate analysis : 5](#_Toc100958635)

[Bilan : 6](#_Toc100958636)

[IV. Data cleaning 6](#_Toc100958637)

[V. Data processing 7](#_Toc100958638)

[1. Rating des caractéristiques : 7](#_Toc100958639)

[2. Data clustering : 8](#_Toc100958640)

[3. Missing values : 9](#_Toc100958641)

[1. Traitement des valeurs aberrantes (outliers) : 9](#_Toc100958642)

[VI. Feature selection : 11](#_Toc100958643)

[VII. Tests statistiques : 13](#_Toc100958644)

[1. Test d’Anova à un facteur : 13](#_Toc100958645)

[2. Test d’ajustement de Kolmogorov-Smirnov : 14](#_Toc100958646)

[VIII. Data modeling : 15](#_Toc100958647)

[1. La régression linéaire : 15](#_Toc100958648)

[2. Régression Ridge : 16](#_Toc100958649)

[3. Régression Lasso : 17](#_Toc100958650)

[4. Elastic Net Regression : 18](#_Toc100958651)

[5. Random forest : 18](#_Toc100958652)

[6. Régression multi linéaire avec deux paramètres : 18](#_Toc100958653)

[Multi colinéarité : 21](#_Toc100958654)

[Autocorrélation : 21](#_Toc100958655)

[IX. Interface graphique 21](#_Toc100958656)

[X. Conclusion 24](#_Toc100958657)

# **Vue générale & Problématique**

## Contexte général du projet

Notre projet consiste sur le développement d’un modèle de régression multilinéaire visant la prédiction du prix des appartements/maisons situés à Casablanca, et ce en se basant sur les diverses caractéristiques du bien.

Pour ce faire, nous procéderons comme suit :

Nous commencerons dans un premier temps par le **data sourcing,** ensuite nous visualiserons notre dataset pour éliminer les données biaisées afin d’assurer une bonne qualité de notre base de données : **étape du** **cleaning et du processing**. Une fois la base de données prête, nous y appliquerons des modèles de prédiction basés sur la régression multilinéaire que nous évaluerons en nous basant sur des indicateurs de performance. Il est à noter qu’une base de données bien traitée permettra au modèle d’avoir une meilleure performance.

Par ailleurs, nous envisagerons de développer une interface graphique à l’aide de l’outil StreamLit pour faciliter à toute personne intéressée par l’achat d’une maison à Casablanca, ayant un certain nombre de caractéristiques qu’il aura à saisir, d’avoir une idée sur son prix. Pour s’assurer que ce dernier soit fiable, nous le comparerons avec les prix présents sur le marché de l’immobilier à Casablanca.

# **Data sourcing**

## Data scrapping

Le data scrapping repose sur l’extraction de données à partir de sites web, Mubawab en l’occurrence. Nous avons choisi de scrapper de nouveau la data dans le but de collecter plus de données relatives à notre sujet.

## Process du scrapping

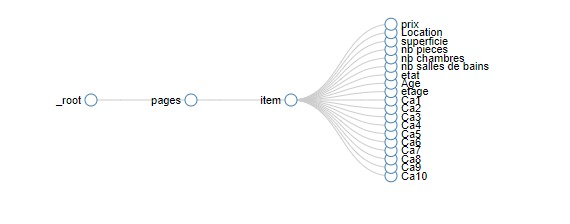
Accéder au site duquel on souhaite scrappée la data.

Préciser les features qu’on souhaite extraire.

Automatiser l’opération de la collecte des données.

* Enregistrer la data sous forme d’un Csv.

Figure 1: Structure de la data scrappée



# **Visualisation première du dataset**

Après avoir complété l’étape relative au sourcing, nous allons maintenant explorer notre dataset.

Une image contenant texte, capture d’écran, rempli, groupe

Description générée automatiquementNotre dataset est de taille (2067,19). Ci-après un extrait du dataset :

Figure 2: Version non traitée du Dataset scrappé

Définissons tout d’abord l’ensemble des colonnes (feaures) dont nous disposons :

## Features description

**Prix :** Le prix du bien, le target.

**Localisation :** le lieu d’implantation du bien.

**Superficie :** la surface du bien (m2).

**Nb pièces :** nombre de pièces du bien.

**Nb chambres :** nombre de chambres du bien.

**Nb salles de bain :** nombre de salles de bain du bien.

**Etat :** l’état du bien (nouveau, bon état, à rénover).

**Age :** l’ancienneté du bien.

**Etage :** l’étage s’il s’agit d’un appartement.

**Cai (1≤i≤10) :** les caractéristiques qui distinguent le bien immobilier.

## Défauts déduits de la visualisation

* Comme il est affiché, le premier défaut remarqué est la présence de **défauts textuels** comme ‘\n\t\t\t\t\tm²’, ‘à\n\t\t\t\t\n\t\t\t\tCasablanca’ et autres, que l’on envisage traiter dans l’étape du processing. Par ailleurs, nous remarquons que certaines **colonnes** sont **décalées**, parmi lesquelles figurent celles relatives à l’état et à l’âge du bien immobilier.
* Certaines colonnes comportent un nombre considérable de missing values, nous allons dans la partie suivante voir comment les traiter. Le type de données sera à son tour être traité de manière à n’avoir que des float pour pouvoir appliquer des modèles de régression linéaire. La data présente 426 de lignes dupliquées (se référer au code ci-après) ;
* Il existe aussi des valeurs aberrantes voire des données erronées en raison de la présence de fausses annonces. Celles-ci pourraient être visibles au niveau des box plots **(univariate analysis)** ci-dessous ou encore au niveau des scatter plots (**bivariate analysis).**

### Box plots :

Ci-après une version première des box plots avant le traitement de la base de données. Chacun de ceux-ci donne une idée sur la distribution de chaque feature.

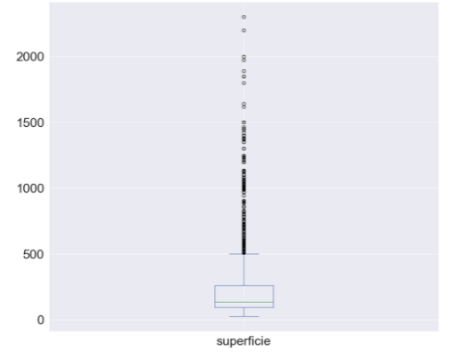
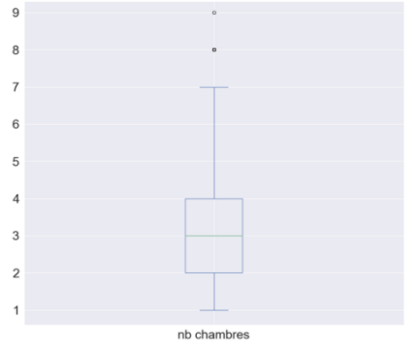
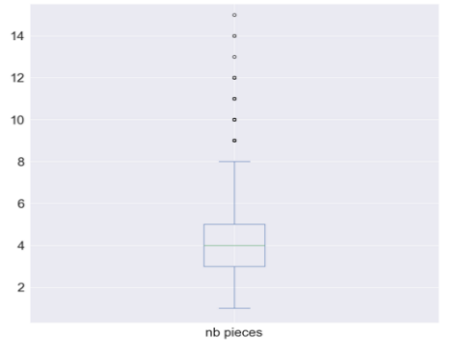
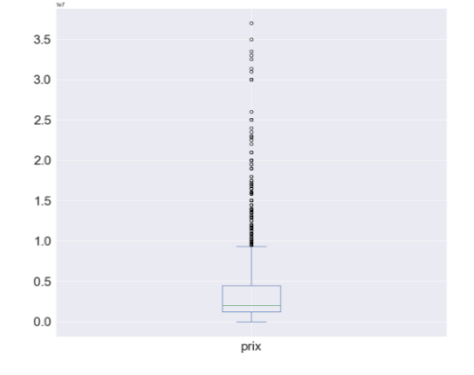


Figure 3: Les box plots

### Bivariate analysis :

Il s’agit ici d’un exemple de scatter plots. Ceux-ci mettent en évidence la présence de valeurs aberrantes comme le montre les deux figures ci-dessous.

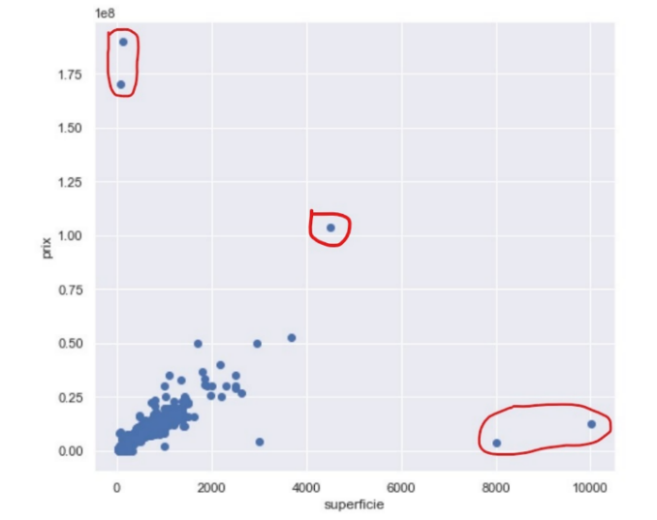
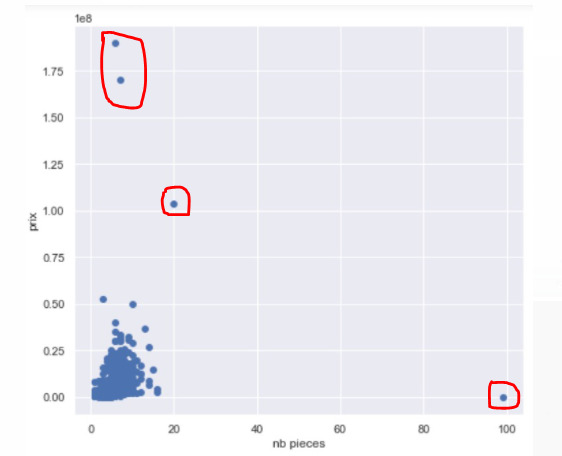


Figure 4.1 : Bivariate Analysis de la superficie

Figure4.2 : Bivariate Analysis du nb de piéces

L’ensemble des scatter plots sont résumés dans la figure ci-après :

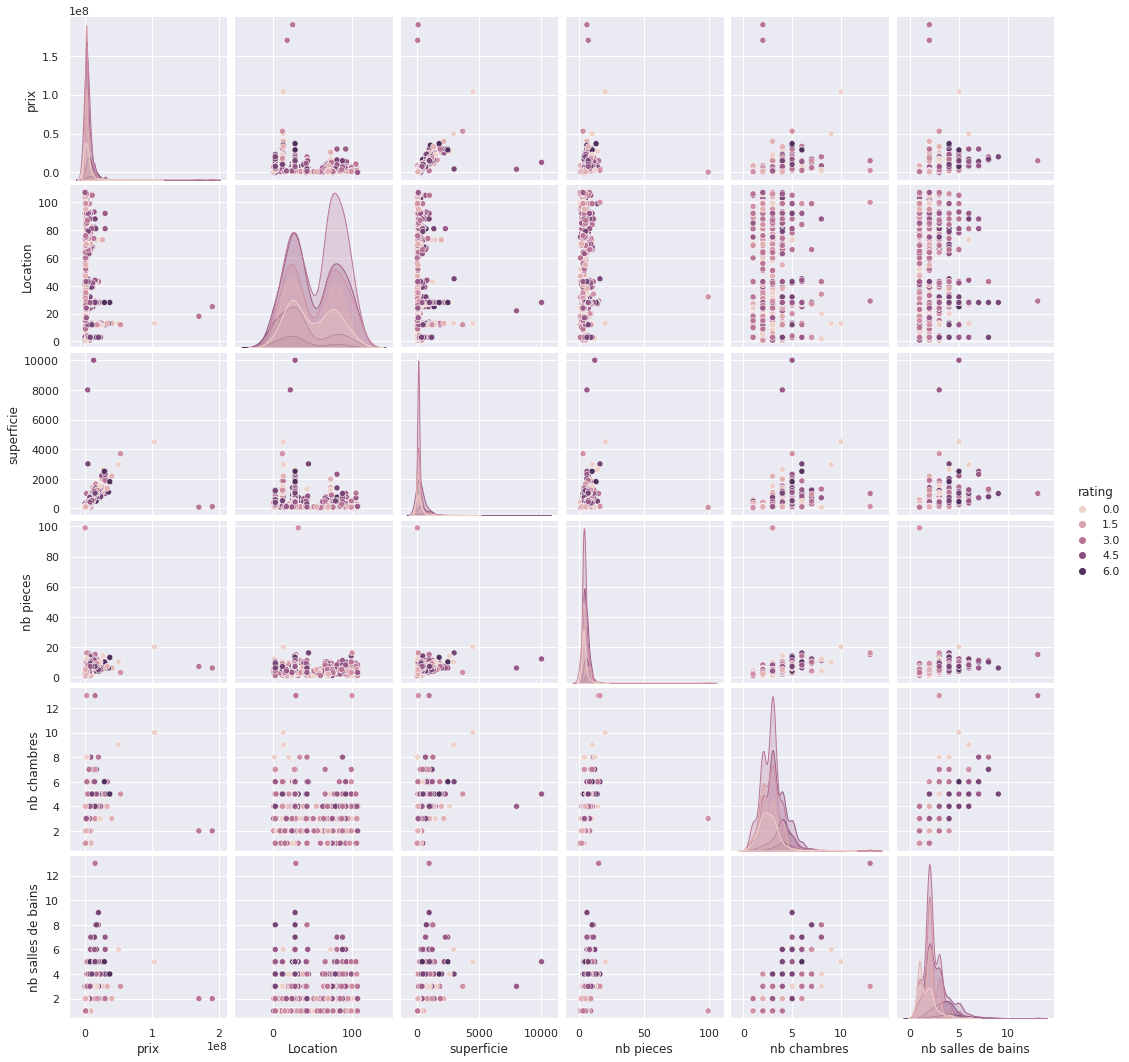


Figure 5 : Scatter plots (avant le traitement des valeurs aberrantes)

Ces différents plots (box plots et scatter plots) permettent d’avoir une idée sur les valeurs aberrantes existantes dans notre dataset. Les boxplots ne permettent pas à eux seuls de conclure qu’il s’agit de fausses annonces vues qu’elles nous ne fournissent que la distribution d’une variable indépendamment des autres. Dans ce sens, pour juger qu’il s’agit bien de valeurs aberrantes, nous avons adopté en plus une analyse bivariée qui met en évidence la variation d’une variable en fonction d’une autre. A titre d’exemple, les deux figures 4.1 et 4.2 présentent la variation du prix en fonction des deux features nombre de pièces et superficie ce qui permet de fournir des résultats plus parlants.

### Bilan :

En somme, cette data nécessite d’être nettoyée en raison des maints défauts qu’elle présente : **colonnes décalées, défauts textuels, présence de valeurs aberrantes et de missing values**, Nous allons dans la partie suivante nous intéresser au traitement de chacun des défauts présentés.

# **Data cleaning**

Pour l’étape du cleaning, nous avons commencé par supprimer les colonnes qui ne participent pas à la prédiction de notre target « prix » comme la page de l’annonce, son lien et autres. Ensuite, nous avons supprimé les lignes dupliquées (426 lignes) et traité les défauts textuels et ceux relatifs aux chevauchements de colonnes pour pouvoir en fin de compte obtenir une première version d’un dataset exploitable.

Nous allons maintenant essayer d’étudier chaque colonne pour voir de près les informations qu’elle présente et donc parvenir à corriger l’ensemble des erreurs qui y sont contenues.

* La Localisation :

La localisation, étant un feature qui a un impact que l’on juge non négligeable sur le prix d’un bien immobilier, nous avons choisi de standardiser les noms des quartiers dans un premier temps pour éviter tout bruit dans la data et donc ne pas entraver l’apprentissage du modèle lors de son implémentation.

* Etat et Age :

Quant aux colonnes relatives à l’âge et l’état de la maison, il y’a eu un chevauchement des deux au niveau de certaines lignes ; nous avons essayé de traiter ce décalage pour éviter de perdre l’information et donc fournir aux modèles de régression un large training set. Ceci fait, analysons de près les informations fournies par chacune de ces deux colonnes.

* Etat :

Pour la colonne relative à l’état du bien (nouveau, bon état, à rénover), nous remarquons que près de 86% de la data fournit une information positive du bien ; en effet, les propriétaires ont dans la plupart du temps tendance à évoluer positivement leur maison, parce qu’en fin de compte leur but n’est autre que maximiser leur prix de vente, ce qui pousse à se fier de la crédibilité de l’information donnée. Dans ce sens, nous avons choisi de l’éliminer.

* Age :

Les informations fournies par les propriétaires des maisons présentent un certain niveau d’incertitude par rapport à l’âge exact du bien : Pour cela, ils apportent une information sous forme d’intervalle (moins d’un an, 5-10 ans …etc.). Pour remédier à cela, nous avons opté pour le remplacement de la donnée par la moyenne de l’intervalle. Cependant, près de 38% de la data est manquante. Pour cela, et pour ne pas biaiser notre base de données avec près de 38% de data estimée, nous avons choisi de supprimer cette colonne.

* Nombre de salles de bain, de chambres et de pièces :

Ces trois colonnes ont nécessité un traitement de texte. Sinon, en termes de missing values : Nombre de pièces : 1,64% - Nombre de chambres : 0% - Nombre de salles de bains : 0,05%. Ce qui est assez soft en comparaison avec les 38% de la colonne « Age ».

Il ne s’agit à ce stade que d’une première version du dataset. Ce dernier sera traité au fur et à mesure lors de l’étape du processing.

# **Data processing**

Tout le travail restant sur la data se rapporte à l’encoding des features du type String (localisation, Ça(i) ; 1≤i≤10), la sélection des caractéristiques qui donnent plus de valeurs au bien, puis le développement d’un système de rating du bien en fonction du nombre de caractéristiques qu’il possède. Par ailleurs, nous allons développer un système de clustering qui regroupe les quartiers qui ont le même prix de vente par et enfin nous allons nous intéresser au traitement des missing values et des valeurs aberrantes.

## Rating des caractéristiques :

En collectant notre data à l’aide des outils de web scrapping des sites d’annonces d’immobilier, nous avons constaté qu’il y a des annonces qui contiennent certaines caractéristiques supplémentaires pour solidifier leurs offres et attirer l’attention de l’acheteur. Les postulants des annonces n’inscrivent pas forcément toutes les caractéristiques se rapportant à leur maison ; pour ce, nous avons choisi d’adopter comme démarche de mettre un “Nan” sur les caractéristiques non mentionnées et un 1 sinon. Par ailleurs, nous avons gardé seulement les caractéristiques qui nous ont semblé pertinentes et éliminer les plus communes. En somme, nous avons gardé : 'Terrasse', 'Garage', 'Ascenseur', 'Piscine', 'Jardin', 'Cheminée', 'Chauffage central’.

En ce qui concerne le système de rating, nous avons ajouté une colonne qui fait la sommation des caractéristiques du bien ce qui permettra d’attribuer un score pour chaque maison/appartement. Nous avons enfin supprimé toutes les colonnes Ca(i) pour les remplacer par la colonne qui contient le score de chaque annonce.

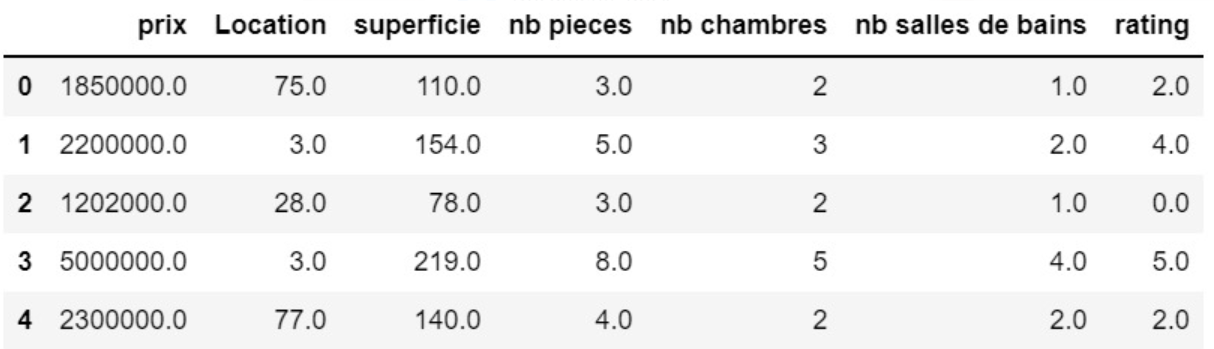


Figure 6 : L’ajout de la colonne du rating

## Data clustering :

La localisation est un string. Pour pouvoir l’exploiter, nous l’avons dans un premier temps encodé par le biais du Label Encoder. Les nombres attribués à chaque localisation ne sont en effet pas parlantes et ont donné une corrélation très faible (-0,1%) ce qui est assez contradictoire à la réalité vu l’importance du quartier dans la détermination du prix d’un bien.

En parlant du prix du housing à Casablanca, nous parlons nécessairement de gammes différentes selon la localisation de la propriété. Pour ce, nous avons choisi d’adopter un système de clustering qui regroupe les quartiers qui ont un prix par comparable par le biais du K-means.

Pour trouver le K le plus optimal dans notre cas, nous avons opté pour l’Elbow Method :

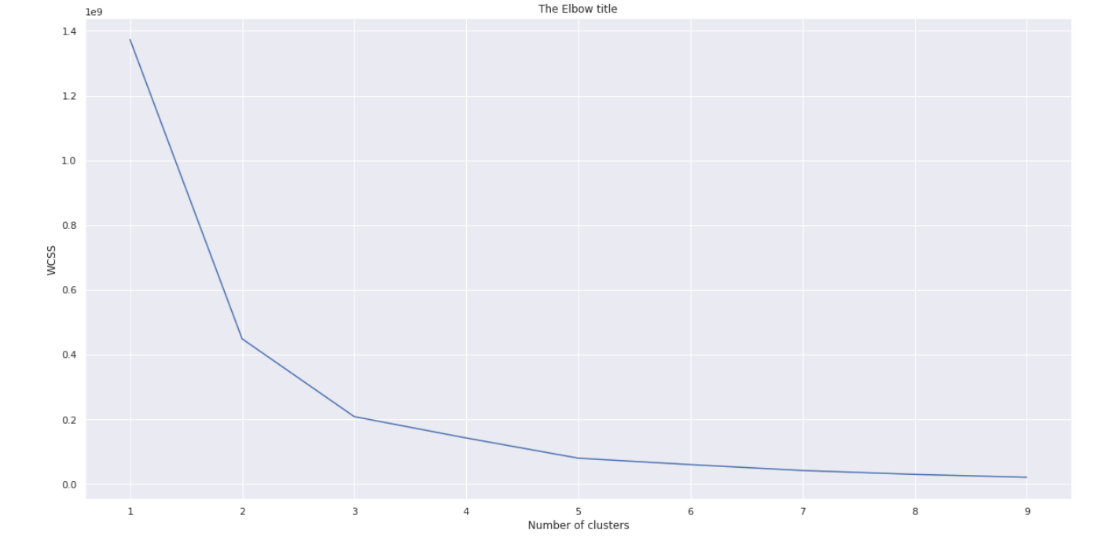


Figure 7 : Elbow method pour trouver le K optimal

Cette courbe nous conduit à choisir K=2 comme étant le K le plus optimal. Nous aurons donc 2 clusters différenciés par un code couleur comme le montre la figure ci-dessous :



Figure 8:les différents clusters trouvés

Les deux clusters font allusion à la moyenne et haute gamme. Pour pouvoir exploiter le clustering lors du modeling, nous avons ajouté une colonne que l’on a nommé « Clusters » qui attribue soit 0 soit 1 à chaque annonce pour indiquer à quelle cluster cette dernière appartient.

## Missing values :

Ci-après le graphe qui met en évidence les missing values dont on dispose actuellement :

Figure 9: représentation des différents missing values dans notre Dataset

Chaque barre blanche modélise une valeur manquante. Le pourcentage de missing values est faible. Pour ne pas biaiser notre data, nous avons décidé de supprimer les lignes comportant les valeurs manquantes.

## Traitement des valeurs aberrantes (outliers) :

Nous avons remarqué lors de la visualisation de notre base de données la présence de valeurs aberrantes dans cette dernière et ceci est dû principalement à la présence de fausses annonces ou encore les annonces qui ne concernent pas les maisons mais plutôt les terrains ou encore les fermes mais aussi d’autres annonces qui se rapportent à la location du bien et non à sa vente. Ces outliers étaient décelables en affichant les box plots mais aussi les scatter plots.

Pour le traitement des outliers, nous avons choisi de les supprimer au lieu de les remplacer pour ne pas falsifier notre base de données.

Nous avons affiché dans un premier les outliers par la méthode de l’IQR (interquartile range method). Pour ce faire, nous avons défini deux variables le « Ceiling » et le « Floor », ils se définissent ainsi :

Q1 et Q3 étant le premier et le troisième quartile et l’IQR la différence entre les deux.

Tout point inférieur au Floor ou supérieur au Ceiling sera considéré comme outlier et sera par la suite supprimé de notre base de données.

Afin de nous assurer que notre base de données est crédible, nous avons en plus de la méthode IQR, opter pour la visualisation du prix/m² de chaque bien immobilier en ajoutant une colonne qui le calcule, et on a veillé à ne laisser que ceux compatibles avec ce qui est présent dans le marché des biens immobiliers de Casablanca.

Ci-après les nouveaux scatter plots obtenus après le traitement des valeurs aberrantes :

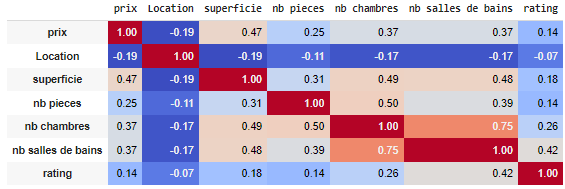
Une image contenant table

Description générée automatiquement

Figure 10 : Scatter plots

* **Corrélation des features :**

Nous remarquons, après le traitement de notre base de données que la matrice de corrélation s’est fortement améliorée et fournit des résultats assez cohérents.



**Version préliminaire de la matrice de corrélation**

**Version améliorée de la matrice de corrélation**

Une image contenant texte, armoire, capture d’écran

Description générée automatiquement

Nous remarquons que la superficie, le nombre de pièces, le nombre de chambres, le nombre de salles de bain et le rating sont corrélées positivement aux prix du bien, ce qui est très parlant. Plus l’un des features que l’on vient de citer augmente plus le prix augmente.

La prochaine étape sera de sélectionner les features qui participent le plus à la prédiction du bien. Nous incluons cette étape pour rendre le modèle moins complexe et ce en diminuant la dimension de notre dataset.

# **Feature selection :**

Cette étape s’avère cruciale car elle nous permet d’avoir une idée sur les variables qui contribuent le plus à la prédiction de notre variable cible « le prix ».

* **PPScore :**

Il existe sur python une bibliothèque quickda qui fournit les colonnes participent à la prédiction du prix ainsi que leur degré de participation en calculant leur pp-score:

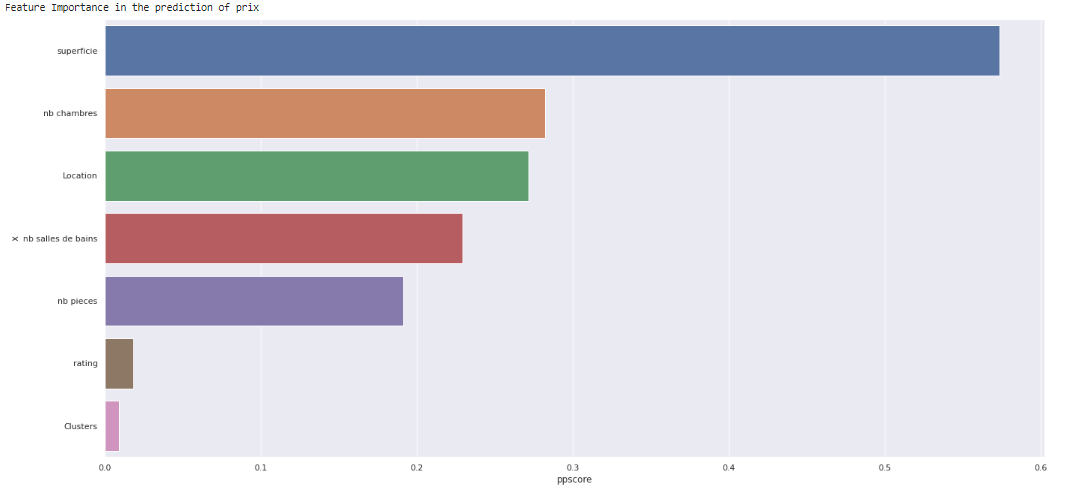


Figure 11 : degré de participation des features dans la prédiction du prix

PPScore ( Predictive Power Score) est un coefficient qui donne une idée sur la capacité d’une variable à prédire la variable cible, le prix en l’occurrence.

Plus ce score est proche de 1 plus la variable a une aptitude à prédire le prix.

* **La régression Stepwise :**

La régression Stepwise est une méthode qui permet de faire des régressions par phase afin de ne laisser que les variables qui contribuent le plus à la prédiction de notre variable dépendante prix. Pour ce faire, elle commence par appliquer une régression simple entre le target et chacune des variables indépendantes, elle garde celle qui a la plus petite p-value (doit être <0,05). Lors de la phase suivante, elle garde la target et la variable sélectionnée précédemment et applique la régression classique. Idem, elle sélectionne celle qui dispose de la plus petite p-value<0,05. La même opération se poursuit jusqu’à ce que p-value>0,05.

Dans notre cas, les variables jugées pertinentes par le modèle sont : la superficie, les clusters et le nombre de salles de bain.

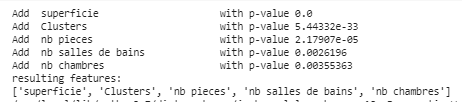


Figure 12 : Features sélectionnés lors de la régression stepwise

# **Tests statistiques :**

Nous avons envisagé d’effectuer deux tests statistiques : le test d’Anova à un facteur et un test d’ajustement.

## Test d’Anova à un facteur :

Nous allons par le biais de ce test essayer d’expliquer la cause de la diversité des informations relatives au prix/m² par l’analyse de sa variance. Nous aurons à traiter une Anova à un facteur tel que :

* **La variable quantitative : le prix/m².**
* **La variable qualitative (le facteur) : Le quartier.**

Notre objectif sera de déterminer si la dépendance de la variable prix/m² est significative ou non au facteur considéré.

Pour ce faire, nous allons essayer de calculer la moyenne des prix/m² de 5 maisons dans différents quartiers.

Tableau: Moyenne des prix/m² des maisons par quartier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Maarif | Ain Diab | Californie | Roches Noires |
| **Prix/m² des maisons** | 13000 | 19166,66667 | 16298,85057 | 8076,923077 |
| 11678,83212 | 14655,17241 | 14923,80952 | 8804,347826 |
| 16170,21277 | 22727,27273 | 17451,32743 | 10000 |
| 18500 | 17580,64516 | 14000 | 13200 |
| 17393,61702 | 17500 | 13269,23077 | 8333,333333 |
| Moyenne | 15348,53238 | 18325,95139 | 15188,64366 | 9682,920847 |

Pour tester l’homogénéité des moyennes, on effectue un test de variance ; le test F réalisé est un test de Fisher (ce test permet de comparer la variance inter-échantillon à la variance intra-échantillon). En effet, la statistique F calculée dans ce test s’exprime comme suit :

**F= (Variance inter-échantillon) / (Variance intra-échantillon)**

Nous obtenons : **F = 10,63** et **.**

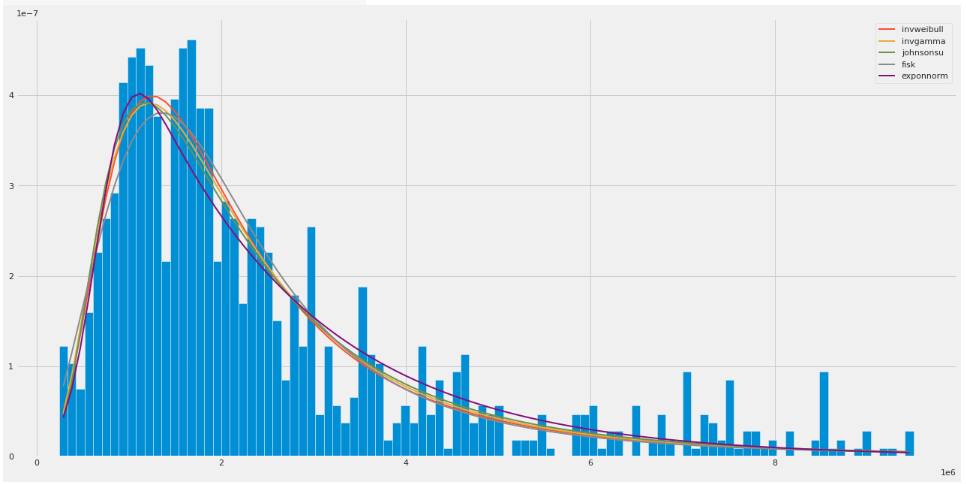
<<0,05 : On rejette l’hypothèse H0 d’égalité des moyennes.

Par conséquent, la variation du prix/m² d’une maison s’explique par la variation du quartier, ce qui est assez logique car sur le terrain ; le prix par mètre carré d’un bien immobilier dépend fortement de la localisation du bien.

## Test d’ajustement de Kolmogorov-Smirnov :

Notre objectif dans cette partie est de comparer la distribution des prix des maisons à une distribution théorique. Pour savoir la distribution qui ressemble le plus à celle de notre échantillon, nous avions recourt à la bibliothèque **fitter** de python. Cette bibliothèque génère les distributions les plus proches de la nôtre.

Figure : Invweibull distribution



On remarque graphiquement que invweibull (inverse weibull distribution) est la plus proche de notre distribution avec la valeur la plus basse du sum squarred errors comme le montre le tableau ci-dessous.

* Inverse Weibull Distribution :

La distribusion inverse de Weibull est une distribution de probabilité continue. Elle est l’inverse de la distribution e Weibull de densité :

Tel que :

k>0 : le paramètre de forme de la distribution.

λ : le paramètre d’échelle de la distribution.

* **Test de Kolmogorov-Smirnov :**

Le test d’ajustement de kolmogorov-Smirnov compare la distribution observée à une distribution théorique, InvWeibull en l’occurrence.

Nous posons comme hypothèse nulle H0 : « Les fonctions de répartition du prix et de la distribution inverse de Weibull sont égales avec un risque d’erreur α=0,05 »

Nous obtenons les résultats suivants :

Résultats du K-S test



, on garde donc l’hypothèse H0.

Par conséquent, on conclut que la distribution du prix suit bien une loi de weibull inverse.

# **Data modeling :**

# 

Après avoir nettoyé, traité et visualisé notre dataset, ce dernier semble prêt pour lui appliquer des modèles de régression linéaire/multi linéaire, surtout que les résultats étaient assez bons en termes de corrélation comme nous venons de le voir dans la partie qui précède.

Pour la division de notre data set, le training set est de 70%, et le testing set 30%.

La variable dépendante est : **prix**

Les variables indépendantes sont : **la location, la superficie, nb pièces, nb chambres, nb de salles de bains et le rating**

## La régression linéaire :

L’idée derrière la régression linéaire est de pouvoir trouver la droite qui minimise la distance entre la variable observée et celle prédite. Étant la variable à prédire, l’**équation \*** de la régression s’exprime comme suit :

Avec :

 : les coefficients à prédire ; : la variable indépendante.

Dans notre cas ; celui de la régression multilinéaire, l’équation \* devient :

Avec :

: les coefficients à prédire ; : les variables indépendantes.

Après avoir appliqué ce modèle, nous avons obtenus les résultats résumés dans le tableau ci-après :

Une image contenant table

Description générée automatiquement

Figure 14 : Résultats de la régression linéaire multiple

En scrutant la colonne coef, nous pouvons très bien observer que les variables indépendantes influent fortement le prix, de plus les p-value associées sont nulles, ce qui montre qu’il s’agit bien de variables significatives dans la prédiction du prix.

Une image contenant texte

Description générée automatiquementLe R-squarred et le MSE (mean sqarred error) qui sont des indicateurs de performance du modèle donnent de bons résultats sur le testing set.

Indicateurs de performance du modèle

Nous craignons qu’il y’ait un overfitting, car le modèle performe mieux dans le training set que dans le testing set. Nous avons, pour cette raison, opté pour la régression ridge et lasso qui s’intéressent à remédier ce type de problème.

## Régression Ridge :

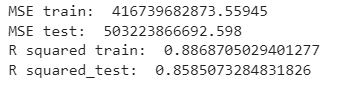
La régression ridge est une version améliorée de la régression linéaire. Ce type de régression pallie l’overfitting en essayant de trouver un compromis entre le **biais** et la **variance** :

* **Le biais:** « The term bias is not the y-intercept but the extent to which the model fails to produce a plot that approximates the samples. » Cory Maklin, data engineer.
* **Variance:** « In contrast to the statistical definition, it measures how the accuracy of a model changes when presented with a different dataset. » Cory Maklin, data engineer.

Le but de la régression ridge est donc de trouver un compromis entre le biais et la variance, en d’autres termes, trouver le grand biais pour la variance la plus faible. En termes quantitatif, ceci se traduit par l’ajout d’un terme de régularisation au RMSE :

Dans notre cas, nous avons choisi d’introduire, après l’application d’une cross validation, un α=40.

Figure 15 : résultats obtenus de la régression Lasso



Certes, le R squared a diminué, cependant, la différence entre le R-squared du train et du test est plus proche donc nous avons pu remédier à l’overfitting à un certain niveau.

Appliquons maintenant la régression Lasso et voyons si les résultats vont s’améliorer.

## Régression Lasso :

La régression Lasso est une version améliorée de la régression Ridge. La régression Lasso pallie aux limitations de la régression Ridge qui se rapportent à la réduction des features. En effet, « Indeed as α increases Lasso reduces more and more feature weights to zero however for Ridge even with high α number of zeros remain the same (weights will be decreasing without ever reaching zero) », Cory Maklin, data engineer.

En outre, il est à noter qu’il y’a une différence dans le terme de régularisation :

Une image contenant texte

Description générée automatiquementLes résultats obtenus de cette régression sont les suivants :

Figure 16 : résulats de la régression Lassà

Les résultats ont bien été amélioré la différence entre les deux R squared est de plus en plus proche et donc le modèle s’avère plus généralisable.

La régression Lasso et Ridge ont des limitations :

Limitations de la régression Ridge : La régression Ridge ne peut pas diminuer le nombre de variables vu qu’elle ne fait que diminuer les coefficients sans pour autant les annuler. Et donc ce type de régression ne contribue pas à la diminution des variables indépendantes.

Limitations de la régression Lasso :

- Si deux variables sont fortement corrélées, la régression Lasso sélectionne l’une d’elle aléatoirement ce qui pourrait entraver l’interprétation du modèle.

- Si le nombre de variables indépendantes est supérieure au nombre d’observations, Lasso limitera le nombre de variables indépendantes à n variables et annulera les autres.

## Elastic Net Regression :

La régression Elastic net vient pallier les limitations des deux régressions Lasso et Rodge et ce en combinant le terme de régularisation des deux.

Une image contenant texte

Description générée automatiquementLes coefficients obtenus après la cv sont indiques ci-dessous

Une image contenant texte

Description générée automatiquementLes R-squared obtenus sont :

R-squared du training set et du testing set sont proches. Ce qui nous assure qu'il n'y a pas de surajustement

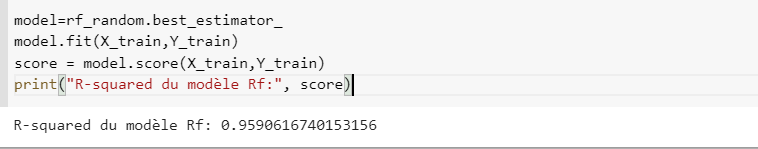
## Random forest :

Ce modèle ne s’appuie pas sur la régression multi-linéaire, mais nous avons choisi de l’appliquer pour visualiser sa performance sur notre dataset.

Avant d’appliquer le random forest, nous avons essayé d’avoir les meilleurs paramètres. Pour ce, nous avons opté pour le hyperparameter tuning comme approche.

Après avoir obtenu les meilleurs paramètres, nous avons utilisé ces derniers pour entrainer notre modèle :

Figure 17 : R-squared Random forest



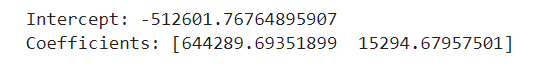
Le modèle de Random Forest est légèrement meilleur que celui de la régression linéaire, ceci se traduit par le fait que la précision de chaque modèle utilisé dans la Random Forest est supérieure à 50%, et puisque le résultat final est une combinaison des autres, alors automatiquement la précision de tout le modèle va s’améliorer.

## Régression multi linéaire avec deux paramètres :

Le but de cette partie est de visualiser comment les paramètres sont linéairement liées, pour ce, nous n’avons utilisé que deux variables pour pouvoir visualiser la linéarité des paramètres en schéma 3D.

En premier lieu nous avons appliqué un modèle de régression linéaire aux deux variables, puis nous avons dessiné un plan passant par les deux lignes de corrélation trouvés par ce modèle, afin de mieux voir la distribution des prédictions de certaines valeurs de ce modèle.

Coefficients de la régression bilinéaire



Ci-après, une visualisation 3D qui représente les graphes que nous obtenons grâce à cette modélisation :

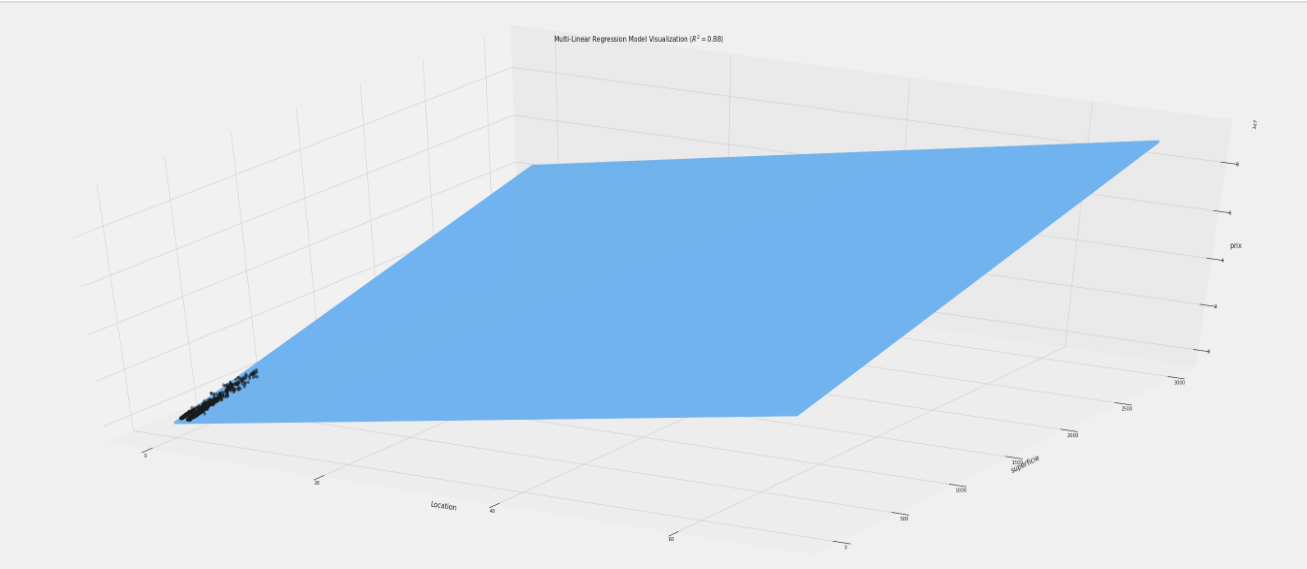
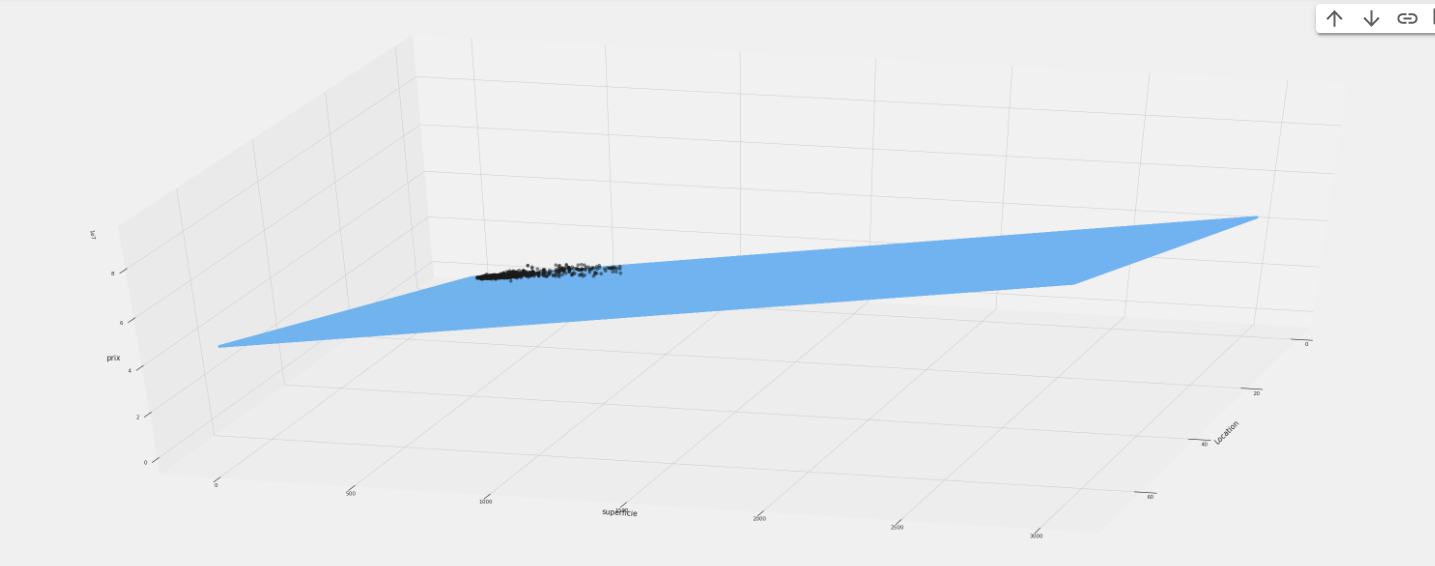


Figure 18 : Représentation 3D (Localisation-Superficie-Prix)

Superficie

Localisation

Prix



Superficie

Prix

Localisation

Figure 19 : Représentation 3D (Localisation-Superficie-Prix)

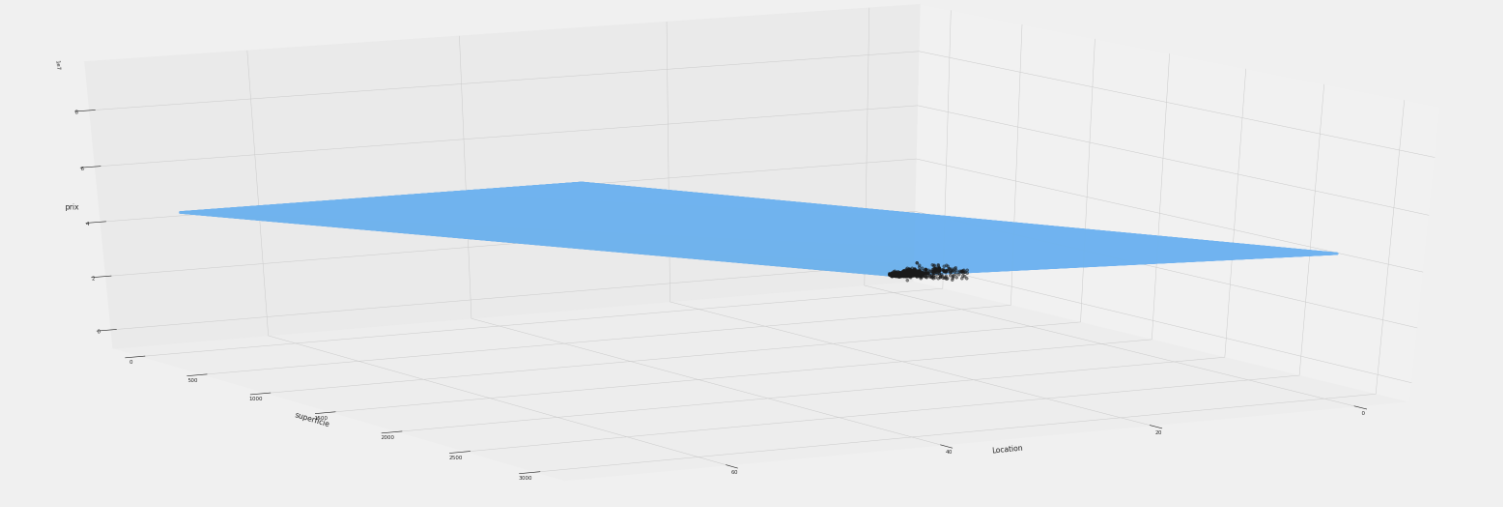


Figure 20 : Représentation 3D (Localisation-Superficie-Prix)

Superficie

Localisation

Prix

Ce modèle a permis d’obtenir les résultats suivants :

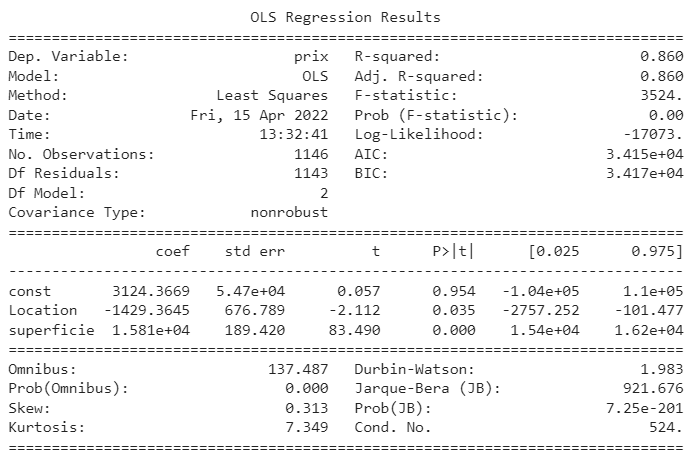


Tableau : Résultats issus de la régression bilinéaire

Le R-squared trouvé est proche de celui que nous avons trouvé précédemment, ceci se justifie par le fait que nous avons utilisé les deux variables les plus corrélées avec notre target.

Après avoir appliqué cette régression linéaire, nous avons fait des tests en occurrence le test de Fischer, test de Student, et nous avons testé quelques propriétés y parmi : la linéarité, la normalité, multi colinéarité, autocorrélation.

### Multi colinéarité :

Une image contenant carré

Description générée automatiquementCela suppose que les prédicteurs utilisés dans la régression ne sont pas corrélés les uns aux autres. Pour identifier s'il existe une corrélation entre nos prédicteurs, nous pouvons calculer le coefficient de corrélation de Pearson entre chaque colonne de nos données.

Figure 21 : Matrice de corrélation

### Autocorrélation :

Nous pouvons détecter l'autocorrélation en effectuant un test de Durbin-Watson pour déterminer si une corrélation positive ou négative est présente.

Le graphe suivant détecte l'homoscédasticité :

Les résidus semblent avoir une variance constante et uniforme, nous pouvons donc dire qu’elles sont homoscédastique.

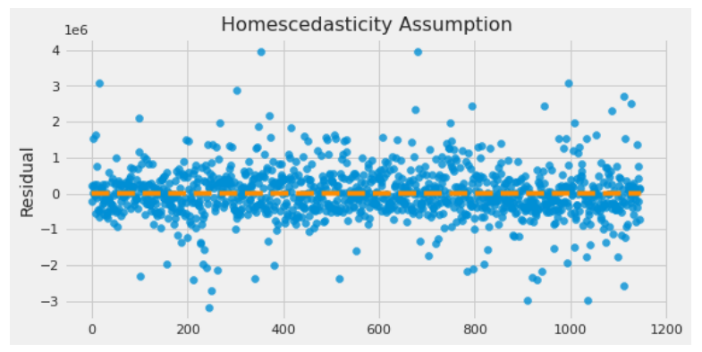
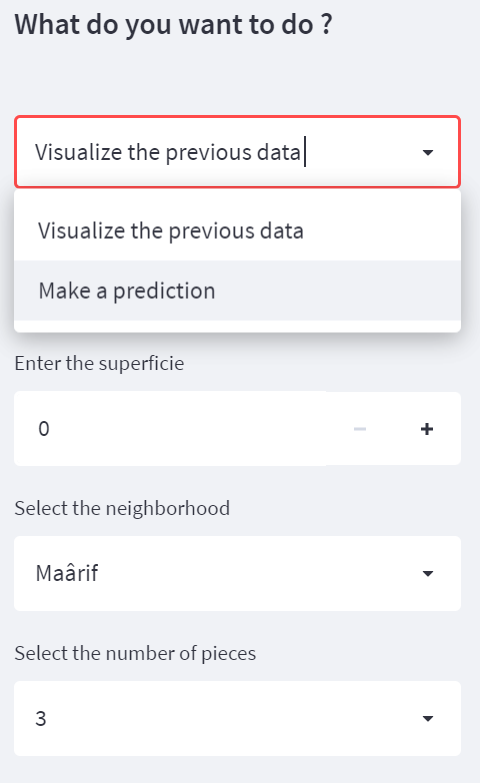
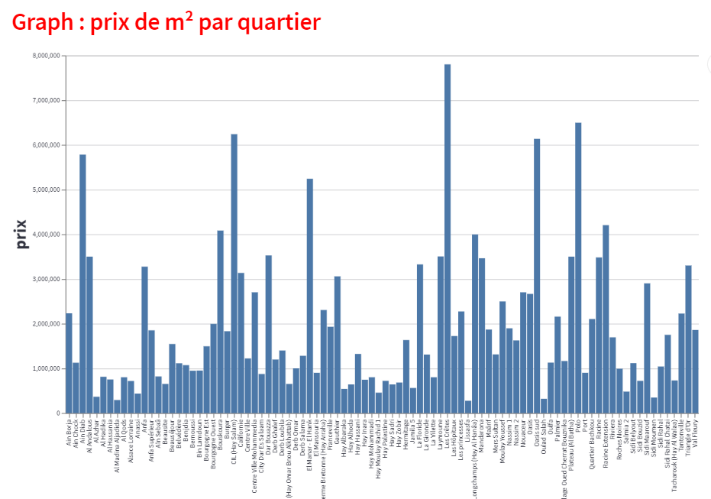
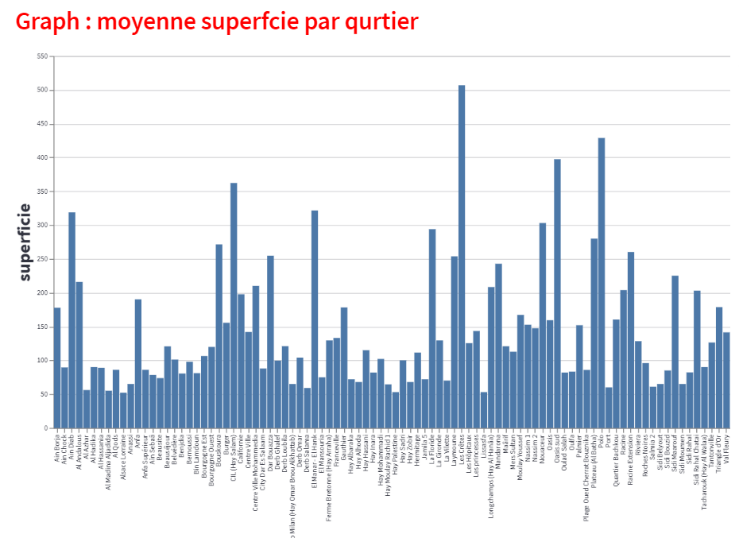


Figure 22 : Homoscédasticité

# **Interface graphique**

Pour rendre le travail fait dans ce projet plus interactif nous avons décidé de réaliser une interface graphique qui facilite aux acheteurs d’avoir une idée sur le prix d’une maison selon des critères bien précis (la superficie, le nombre de pièces et la localisation).

L’utilisateur doit choisir entre les 2 options « Make a prediction » et « visualize the previous data »

Grâce à cette interface nous pouvons aussi visualiser la variation du prix par rapport au quartier, nombre de pièces et la moyenne de la superficie dans chaque quartier.

Une image contenant texte

Description générée automatiquementPour faire la prédiction, en Backend nous nous sommes basés sur le Modèle « Random forest » vu sa bonne performance. Nous avons pris l’exemple d’une maison que l’on vient d’être enregistré sur le site. Notre programme a généré le prix ci-dessous.

Le prix généré est bien crédible car il est près de celui affiché sur l’annonce.

Une image contenant texte

Description générée automatiquementPar conséquent, nous jugeons que cette prédiction est correcte

# **Conclusion**

L’objectif de cette étude était de développer des modèles basés sur la régression multilinéaire qui permettent la prédiction des prix des maisons situées à Casablanca. Pour ce faire, nous avons commencé par scrapper une base de données d’un site d’annonces des biens immobiliers situés à Casablanca. Ensuite, nous nous sommes intéressés au cleaning de la donnée. Il s’agit d’une étape assez cruciale et qui demande le plus de temps pour être bien assurée, vu que les défauts que la data présente pourrait ne pas être facilement décelable. Ensuite, le traitement de la donnée. En effet, nous avions veillé à ce que notre base de données soit crédible et conforme à ce qui existe dans le marché de l’immobilier à Casablanca. Pour diminuer la complexité de notre modèle, nous avons veillé à sélectionner les variables qui participent le plus à la prédiction du prix. Pour finir, nous avons appliqué des modèles qui se base sur la régression multilinéaire pour la prédiction du prix des maisons. Dans le but de rendre cette étude exploitable, nous avons mis à disposition de l’utilisateur une interface graphique qui lui donnera une idée sur le prix des maisons situés à Casablanca en fonction de la superficie, du nombre de pièces et de la localisation.