CESI
ÉCOLE D'INGÉNIEURS

A5 FISA INFO

Bloc Intelligence artificielle

PROSIT 1 : On sépare les appartements

Pierre Carteron

11 février 2026

Table des matières

Table des figures

1 Introduction

1.1 Contexte

Dans un but d'optimisation des propositions de bien immobilier et de réduction des visites. L'agence LoftCraft envisage de recouvrir à l'IA en utilisant un jeu de données.

1.2 Problématique

Comment préparer et analyser un jeu de données ?

1.3 Généralisation

Préparation de dataset

1.4 Contraintes

- Jeu de données incomplet et incohérent

1.5 Hypothèses

- Les données doivent être anonymisées
- Utilisation de l'algorythme K-mean
- Normaliser les données

1.6 Livrables

- Un jeu de données propre et préparé pour l'analyse et le machine learning.
- Un notebook Jupyter documenté comprenant l'analyse exploratoire des données, les visualisations et les traitements appliqués.
- Une interprétation des clusters obtenus mettant en évidence les zones à forte et faible valeur immobilière.
- Une réflexion synthétique sur les enjeux éthiques liés à l'utilisation de l'IA dans le secteur immobilier

2 Mots Clés

- **IA :** Ensemble de techniques permettant à une machine de simuler des capacités humaines comme raisonner, apprendre ou prendre des décisions.

- **K-means** : Algorithme de clustering qui regroupe automatiquement des données en K groupes selon leur similarité.

- **Clustering** : Méthode d'apprentissage non supervisé qui consiste à regrouper des données similaires en clusters sans connaître les catégories à l'avance.

- **Jeu de données** : Ensemble structuré d'informations utilisées pour entraîner, tester ou évaluer un modèle.

- **Apprentissage automatisée** : Processus par lequel une machine améliore ses performances à partir de données, sans être explicitement programmée.

- **Machine learning** : Domaine de l'IA qui conçoit des algorithmes capables d'apprendre des modèles à partir de données.

- **Dataset** : Terme anglais désignant un jeu de données, souvent utilisé dans le contexte du machine learning.

3 Recherches

3.1 Préparation et compréhension d'un jeu de données

Rôle des données en intelligence artificielle

En intelligence artificielle, la donnée constitue la base de tout système d'apprentissage. Un algorithme, aussi performant soit-il, ne peut produire de résultats pertinents que si les données sur lesquelles il s'appuie sont fiables, cohérentes et représentatives du problème étudié. Un jeu de données se présente généralement sous forme tabulaire, où chaque ligne correspond à une observation et chaque colonne à une variable descriptive.

Dans le contexte immobilier de Loft's Craft, chaque observation représente un secteur de recensement en Californie. Les variables associées décrivent à la fois la localisation géographique, les caractéristiques des logements, les aspects démographiques et les éléments économiques. Cette diversité permet une analyse riche du territoire, mais implique également un travail de compréhension approfondi avant toute exploitation algorithmique.

Structure et nature du jeu de données immobilier

Le jeu de données utilisé est composé de plusieurs milliers d'entrées et de variables majoritairement numériques, complétées par une variable catégorielle indiquant la proximité à l'océan. Les données géographiques, telles que la latitude et la longitude, permettent de situer précisément chaque secteur, tandis que les variables économiques comme le revenu médian ou la valeur médiane des logements donnent une indication directe du niveau de vie et du marché immobilier local.

L'analyse de la structure du jeu de données met en évidence des ordres de grandeur très différents entre les variables, ainsi que la présence de valeurs manquantes dans certaines colonnes. Cette observation est essentielle, car les algorithmes d'apprentissage automatique sont sensibles à ces problématiques et nécessitent des données homogènes et complètes pour fonctionner correctement.

Exploration et compréhension statistique des données

La phase d'exploration vise à produire une première lecture globale du jeu de données. À l'aide d'outils statistiques simples, il est possible d'observer la distribution des variables, d'identifier les valeurs extrêmes et de repérer les déséquilibres potentiels. Cette étape permet également de détecter les variables les plus informatives pour la suite de l'analyse.

Dans le cas étudié, l'exploration révèle notamment une forte variation des valeurs immobilières en fonction de la localisation géographique. Ces premières observations orientent naturellement le choix des méthodes d'analyse ultérieures, en particulier vers des techniques capables de prendre en compte la similarité entre zones.

3.2 Environnement de travail : Jupyter Notebook et Pandas

Jupyter Notebook comme support d'analyse

Le Jupyter Notebook est un environnement interactif largement utilisé en data science. Il permet d'intégrer dans un même document du code, des visualisations et des commentaires explicatifs. Cette approche favorise une compréhension progressive du raisonnement et facilite la reproductibilité des analyses.

Dans un cadre pédagogique, le Jupyter Notebook joue également le rôle de support de cours pratique. Chaque cellule de code correspond à une étape précise du traitement des données, tandis que les cellules de texte permettent d'expliquer les choix méthodologiques et les résultats obtenus.

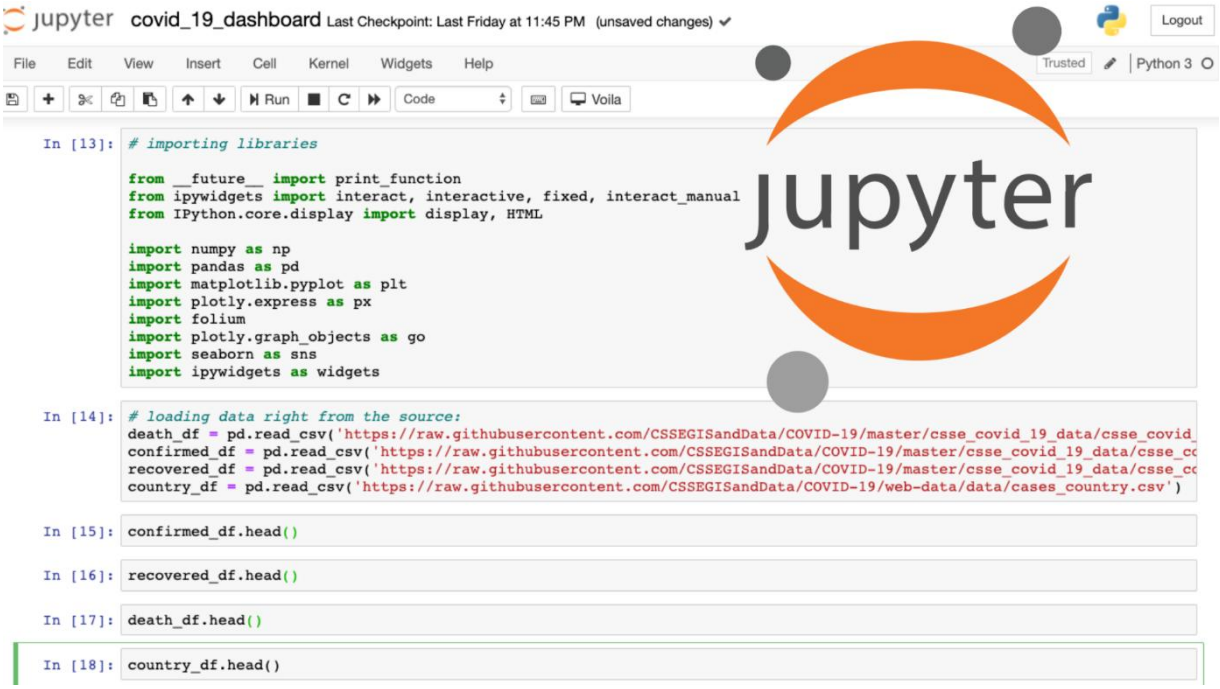

Figure 1 : Exemple d'une interface Jupyter

Manipulation des données avec Pandas

Pandas est une bibliothèque Python spécialisée dans la manipulation de données structurées. Elle introduit la notion de DataFrame, qui permet de représenter un jeu de données sous forme de tableau, proche d'une feuille de calcul mais beaucoup plus puissant.

Dans ce projet, Pandas est utilisé pour charger le jeu de données, afficher ses premières lignes, examiner les types de variables et résumer les informations disponibles. Ces opérations sont fondamentales, car elles permettent de vérifier la cohérence des données et de préparer leur transformation pour les algorithmes d'apprentissage automatique.

Gestion des valeurs manquantes et transformation des variables

Les valeurs manquantes constituent un problème fréquent dans les jeux de données réels. Dans ce cours, une stratégie d'imputation simple est adoptée afin de conserver l'intégralité des observations. Les variables numériques sont complétées à l'aide de la médiane, tandis que les variables catégorielles sont remplacées par leur valeur la plus fréquente.

Cette approche permet de limiter la perte d'information tout en garantissant que le jeu de données est exploitable par les modèles. Les transformations appliquées à cette étape sont déterminantes pour la qualité des résultats obtenus par la suite.

3.3 Apprentissage non supervisé et clustering

Principe de l'apprentissage non supervisé

L'apprentissage non supervisé regroupe des méthodes permettant d'analyser des données sans disposer de résultats attendus ou de labels prédéfinis. L'objectif n'est pas de prédire une valeur connue, mais de découvrir des structures internes au jeu de données.

Dans un contexte immobilier, cette approche est particulièrement pertinente, car elle permet d'identifier des profils de zones ou de biens sans imposer de critères arbitraires. L'algorithme se base uniquement sur les similarités entre les données.

Notion de clustering

Le clustering consiste à regrouper des observations similaires en ensembles appelés clusters. Deux observations appartenant à un même cluster sont plus proches l'une de l'autre que de celles appartenant à un autre cluster. Cette notion de proximité repose généralement sur des mesures de distance calculées à partir des variables numériques.

Appliqué au marché immobilier, le clustering permet de segmenter le territoire en zones homogènes, facilitant ainsi l'analyse et l'aide à la décision.

Visualisation des regroupements

La visualisation des données joue un rôle clé dans l'interprétation des clusters. En projetant les données sur une carte géographique, il devient possible d'observer la répartition spatiale des groupes identifiés. Dans le cas étudié, ces visualisations mettent en évidence des zones côtières et urbaines où les valeurs immobilières sont plus élevées.

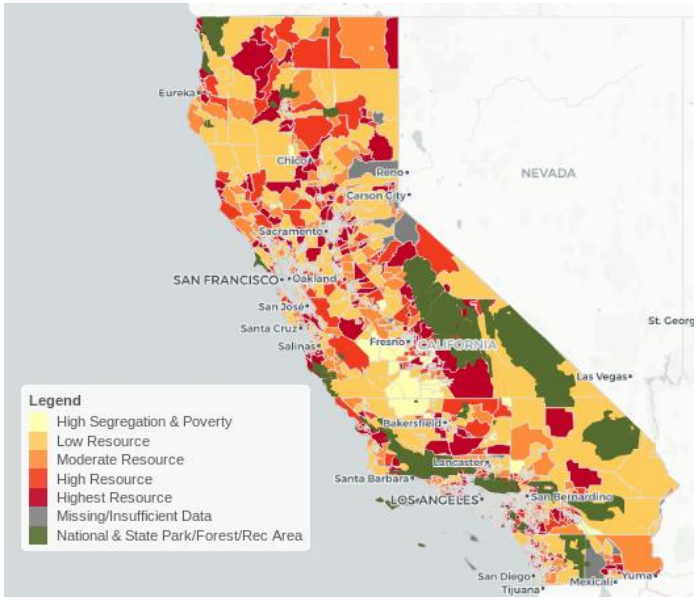

Figure 2 : Schéma de représentation les différences entre les localités de la Californie aux Etats-Unis

3.4 Algorithme K-means

Principe général de K-means

L'algorithme K-means est une méthode de clustering qui vise à partitionner un jeu de données en un nombre fixé de groupes. Chaque groupe est représenté par un centroïde, correspondant à la moyenne des points qui lui sont associés. L'algorithme cherche à minimiser la distance entre les points et le centroïde de leur cluster.

Ce fonctionnement itératif permet à K-means de converger vers une partition stable, révélant des structures naturelles dans les données.

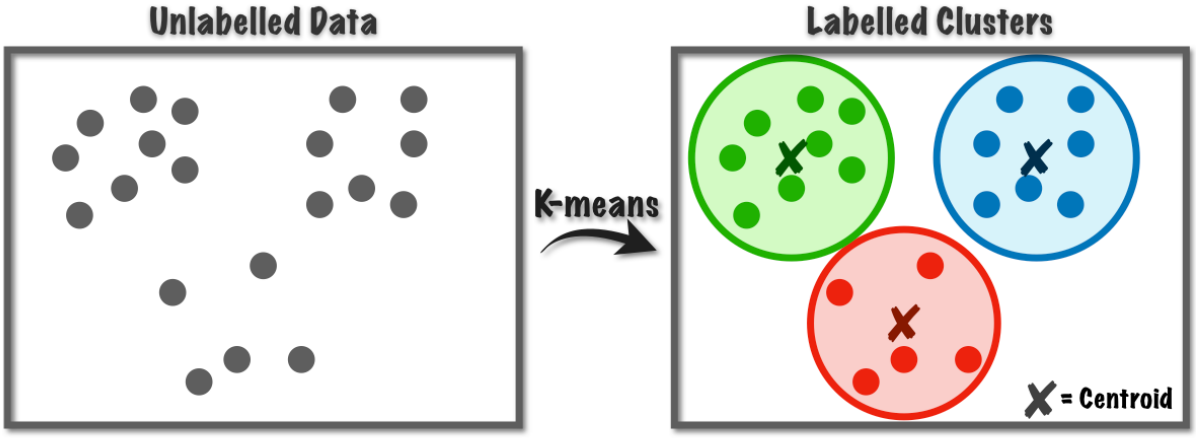

Figure 3 : Schéma représentant l'intérêt et le principe de l'algorithme K-means

Application de K-means au jeu de données immobilier

Dans le cadre de Loft's Craft, l'utilisation de K-means avec deux clusters permet de distinguer deux grands types de zones immobilières. Cette séparation est obtenue sans paramétrage métier préalable, uniquement à partir des caractéristiques des secteurs de recensement.

Les résultats montrent que l'algorithme parvient à isoler des zones à forte valeur immobilière, notamment autour des grandes métropoles et des régions côtières, confirmant les tendances observées lors de l'exploration des données.

Intérêt pédagogique et métier de K-means

D'un point de vue pédagogique, K-means illustre parfaitement le fonctionnement de l'apprentissage non supervisé. D'un point de vue métier, il constitue un outil d'aide à la décision permettant de mieux comprendre la structure du marché immobilier. Il offre ainsi à l'agence une base solide pour optimiser ses recommandations et améliorer la qualité de ses services.

Figure 4 : Graphique de l'utilisation d'un algorithme K-means dans un jeu de données

3.5 Dash

Dash est un framework Python permettant de créer des applications web interactives dédiées à la visualisation et à l'analyse de données. Il est principalement utilisé en data science pour transformer des résultats d'analyse en tableaux de bord dynamiques, accessibles depuis un navigateur web, sans avoir besoin de maîtriser le développement web classique (HTML, CSS, JavaScript).

Dash s'appuie sur l'écosystème Python et s'intègre naturellement avec des bibliothèques comme Pandas, NumPy et Plotly. Il constitue ainsi un outil idéal pour valoriser des données issues de projets d'intelligence artificielle ou de machine learning.

Principe de fonctionnement de Dash

Une application Dash repose sur trois éléments fondamentaux : les données, l'interface utilisateur et les interactions. Les données sont généralement manipulées avec Pandas, tandis que l'interface est décrite à l'aide de composants Python représentant des éléments HTML et graphiques. Les interactions permettent à l'utilisateur de modifier dynamiquement les visualisations, par exemple en changeant un paramètre ou en sélectionnant une zone.

Dash fonctionne selon un modèle réactif. Cela signifie que lorsqu'un utilisateur interagit avec un composant (un bouton, un onglet, un graphique), l'application met automatiquement à jour les éléments concernés sans recharger la page.

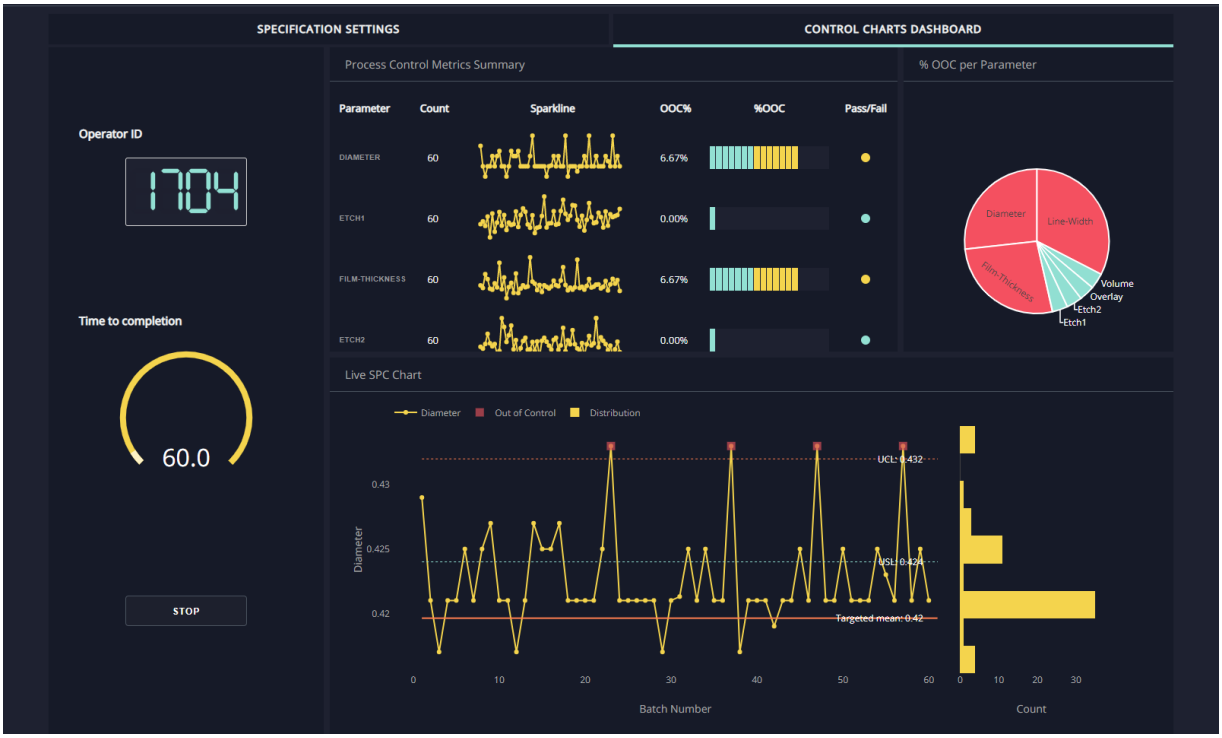

Figure 5 : Exemple d'un dashboard avec Dash

Structure d'une application Dash

Une application Dash commence par l'initialisation de l'application, qui crée le lien entre Python et le navigateur web. Ensuite, on définit le layout, c'est-à-dire la structure de la page. Le layout décrit l'organisation des titres, graphiques, onglets et autres composants visibles par l'utilisateur.

Les graphiques intégrés dans Dash sont généralement construits avec Plotly, ce qui permet d'obtenir des visualisations interactives comme des histogrammes, des nuages de points ou des cartes géographiques. Chaque graphique est encapsulé dans un composant Dash, ce qui facilite son intégration dans l'interface.

Interactivité et callbacks

L'un des points clés de Dash est la notion de callback. Un callback est une fonction Python qui relie une interaction utilisateur à une mise à jour de l'interface. Par exemple, lorsqu'un utilisateur sélectionne une variable dans une liste déroulante, un callback peut recalculer un graphique et l'afficher instantanément.

Ce mécanisme permet de créer des tableaux de bord dynamiques et personnalisables, adaptés à l'exploration des données. Dans un contexte d'intelligence artificielle, cela permet à un utilisateur non technique d'interagir avec les résultats d'un modèle sans avoir à manipuler le code.

Intérêt de Dash en data science et IA

Dash joue un rôle essentiel dans la valorisation des données. Il permet de transformer des analyses complexes en outils visuels compréhensibles par des décideurs ou des clients. Dans un projet immobilier comme celui de Loft's Craft, Dash peut servir à explorer les prix, les zones géographiques ou les clusters identifiés par un algorithme comme K-means.

Ainsi, Dash ne remplace pas les algorithmes d'intelligence artificielle, mais agit comme une interface entre les données, les modèles et l'utilisateur final, facilitant la prise de décision.

4 Réalisation

4.1 Initialisation de l'environnement et importation des bibliothèques

La réalisation débute par l'importation des bibliothèques Python nécessaires au projet. Cette étape permet de définir l'ensemble des outils qui seront utilisés tout au long du notebook. Les bibliothèques de manipulation de données servent à charger et transformer le jeu de données, tandis que les bibliothèques de visualisation permettent de produire des graphiques statistiques, des cartes et des représentations interactives.

Cette phase d'initialisation est essentielle car elle conditionne toutes les opérations suivantes. Elle garantit que l'environnement de travail est prêt et que les différentes briques du pipeline d'analyse peuvent fonctionner ensemble.

4.2 Extraction et mise à disposition du jeu de données

Le jeu de données est initialement fourni sous la forme d'une archive compressée. Le code commence donc par gérer cette contrainte technique en automatisant l'extraction de l'archive dans un répertoire de travail dédié.

Cette étape a pour objectif de rendre accessible le fichier de données brut sans intervention manuelle. Elle permet également de structurer proprement le projet en séparant les fichiers sources des scripts d'analyse. Une fois l'archive extraite, le fichier de données devient exploitable par les outils de data science.

4.3 Chargement des données dans Pandas

Après l'extraction, le fichier CSV contenant les données immobilières est chargé en mémoire à l'aide de Pandas. Cette opération transforme le fichier brut en un DataFrame, c'est-à-dire une structure de données tabulaire permettant des manipulations efficaces.

À ce stade, les données sont prêtes à être explorées. Le chargement constitue un point clé du notebook, car il marque le passage d'un simple fichier à un objet manipulable, sur lequel des analyses statistiques et des visualisations peuvent être appliquées.

4.4 Exploration initiale du jeu de données

Une première exploration est réalisée afin de vérifier la structure du DataFrame. Le code permet d'observer le nombre de variables, leur type et la présence éventuelle de valeurs manquantes. Cette phase joue un rôle de contrôle qualité, en s'assurant que les données ont été correctement importées et qu'elles correspondent bien aux attentes du projet.

Cette étape permet également de se familiariser avec le contenu du jeu de données avant d'engager des transformations plus complexes.

Résultat de la fonction info() :

```
<class 'pandas.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
```

```
---  ------              --------------  -----
 0   longitude           20640 non-null  float64
 1   latitude            20640 non-null  float64
 2   housing_median_age  20640 non-null  float64
 3   total_rooms         20640 non-null  float64
 4   total_bedrooms      20433 non-null  float64
 5   population          20640 non-null  float64
 6   households          20640 non-null  float64
 7   median_income       20640 non-null  float64
 8   median_house_value  20640 non-null  float64
 9   ocean_proximity     20640 non-null  str
dtypes: float64(9), str(1)
memory usage: 1.7 MB
```

Résultat de la fonction describe() :

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
count	20640.00	20640.00	20640.00	20640.00	20433.00	20640.00	20640.00	20640.00	20640.00
mean	-119.57	35.63	28.64	2635.76	537.87	1425.48	499.54	3.87	206855.82
std	2.00	2.14	12.59	2181.62	421.39	1132.46	382.33	1.90	115395.62
min	-124.35	32.54	1.00	2.00	1.00	3.00	1.00	0.50	14999.00
25%	-121.80	33.93	18.00	1447.75	296.00	787.00	280.00	2.56	119600.00
50%	-118.49	34.26	29.00	2127.00	435.00	1166.00	409.00	3.53	179700.00
75%	-118.01	37.71	37.00	3148.00	647.00	1725.00	605.00	4.74	264725.00
max	-114.31	41.95	52.00	39320.00	6445.00	35682.00	6082.00	15.00	500001.00

Figure 6 : Tableau résultant de la fonction describe()

4.5 Analyse descriptive des variables numériques

Le notebook poursuit avec une analyse descriptive des variables numériques. Le code calcule des statistiques globales telles que les moyennes, médianes, minimums et maximums. Cette analyse fournit une première compréhension des ordres de grandeur et de la dispersion des données.

Cette phase est importante car elle permet d'identifier des caractéristiques propres au marché immobilier, comme des distributions asymétriques ou la présence de valeurs élevées concentrées sur certaines variables.

4.6 Visualisation et identification des valeurs manquantes

Une visualisation spécifique est ensuite générée pour analyser les valeurs manquantes dans le jeu de données. Le code produit une représentation graphique permettant d'identifier rapidement quelles colonnes sont concernées et dans quelle proportion.

Cette étape apporte une vision synthétique de la qualité des données et sert de base pour justifier les choix de nettoyage appliqués par la suite.

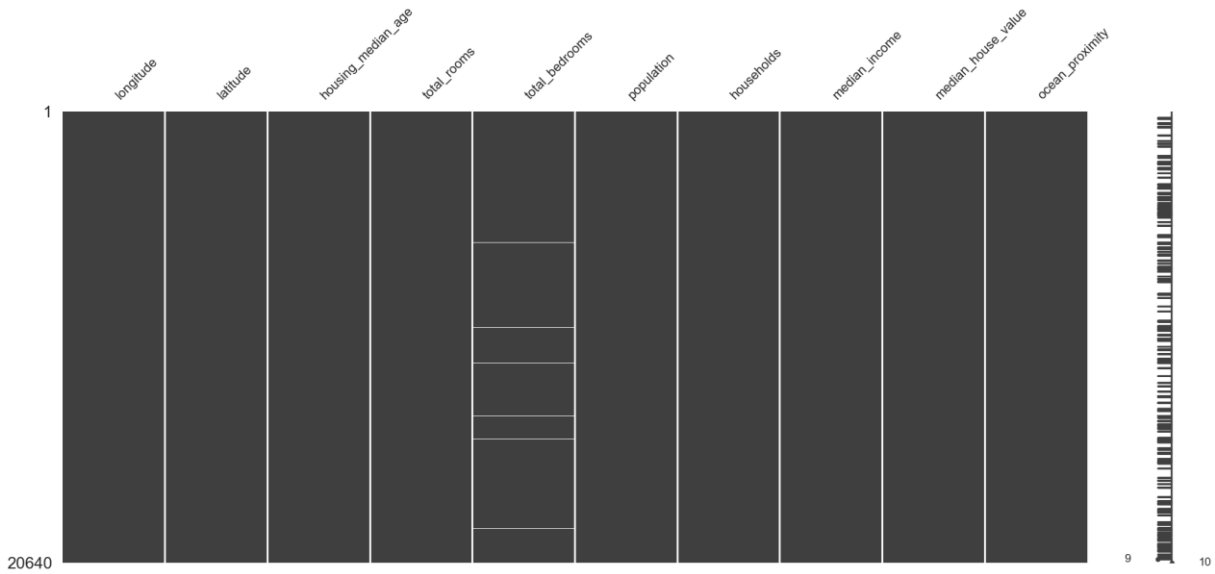

Figure 7 : Graphique affichant les données manquantes dans chaque colonne

4.7 Traitement et imputation des valeurs manquantes

Une fois les valeurs manquantes identifiées, le code met en place une stratégie de nettoyage des données. Les colonnes numériques sont traitées séparément des colonnes catégorielles, afin d'appliquer une méthode d'imputation adaptée à chaque type de variable.

Cette opération permet de compléter le jeu de données sans supprimer d'observations, ce qui est essentiel pour conserver un maximum d'information. À l'issue de cette étape, le DataFrame est complet et prêt pour les analyses exploratoires approfondies.

4.8 Analyse univariée par visualisation

Le notebook propose ensuite une analyse univariée à l'aide de visualisations adaptées. Le code génère des graphiques permettant d'étudier la distribution de certaines variables clés, notamment celles liées aux prix de l'immobilier.

Cette étape permet d'identifier des tendances globales, des concentrations de valeurs et des limites structurelles du jeu de données, comme des plafonds de prix.

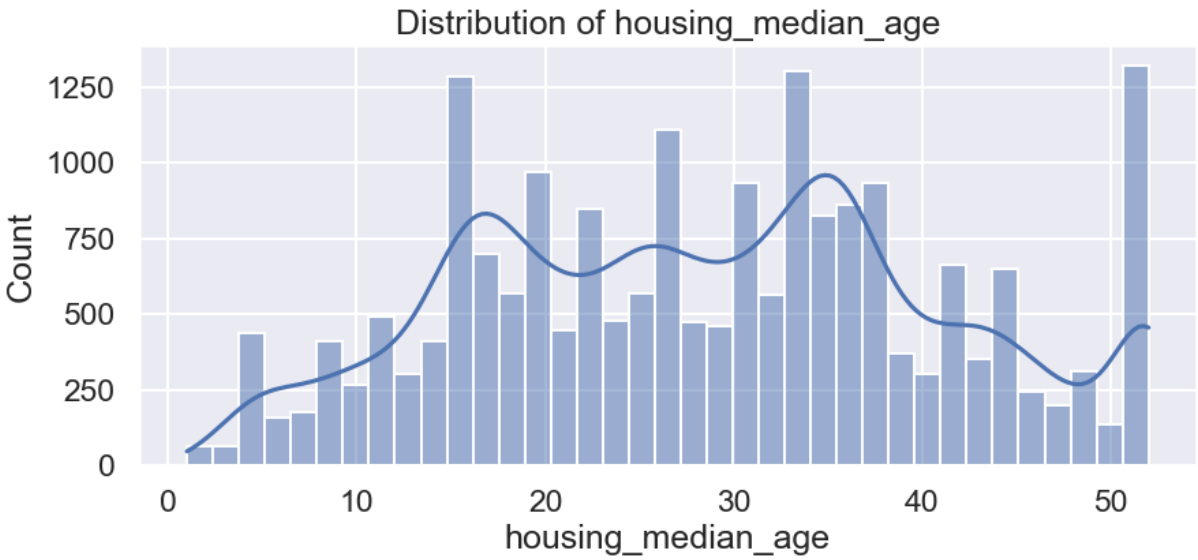

Figure 8 : Graphique représentant le nombre de bien vendu selon l'âge médian

Figure 9 : Graphique du nombre de bien vendu selon le salaire médian

4.9 Analyse bivariée et relations entre variables

Le code enchaîne avec une analyse bivariée visant à explorer les relations entre deux variables. Des visualisations permettent notamment de mettre en relation les revenus médians et les valeurs immobilières.

Cette phase apporte une première compréhension des liens entre les caractéristiques socio-économiques et le marché immobilier, et permet de confirmer ou d'infirmer des hypothèses intuitives.

Nous pouvons observer ci-dessous les graphiques représentant la relation des tendances linéaires en fonction des paires de chaque colonne :

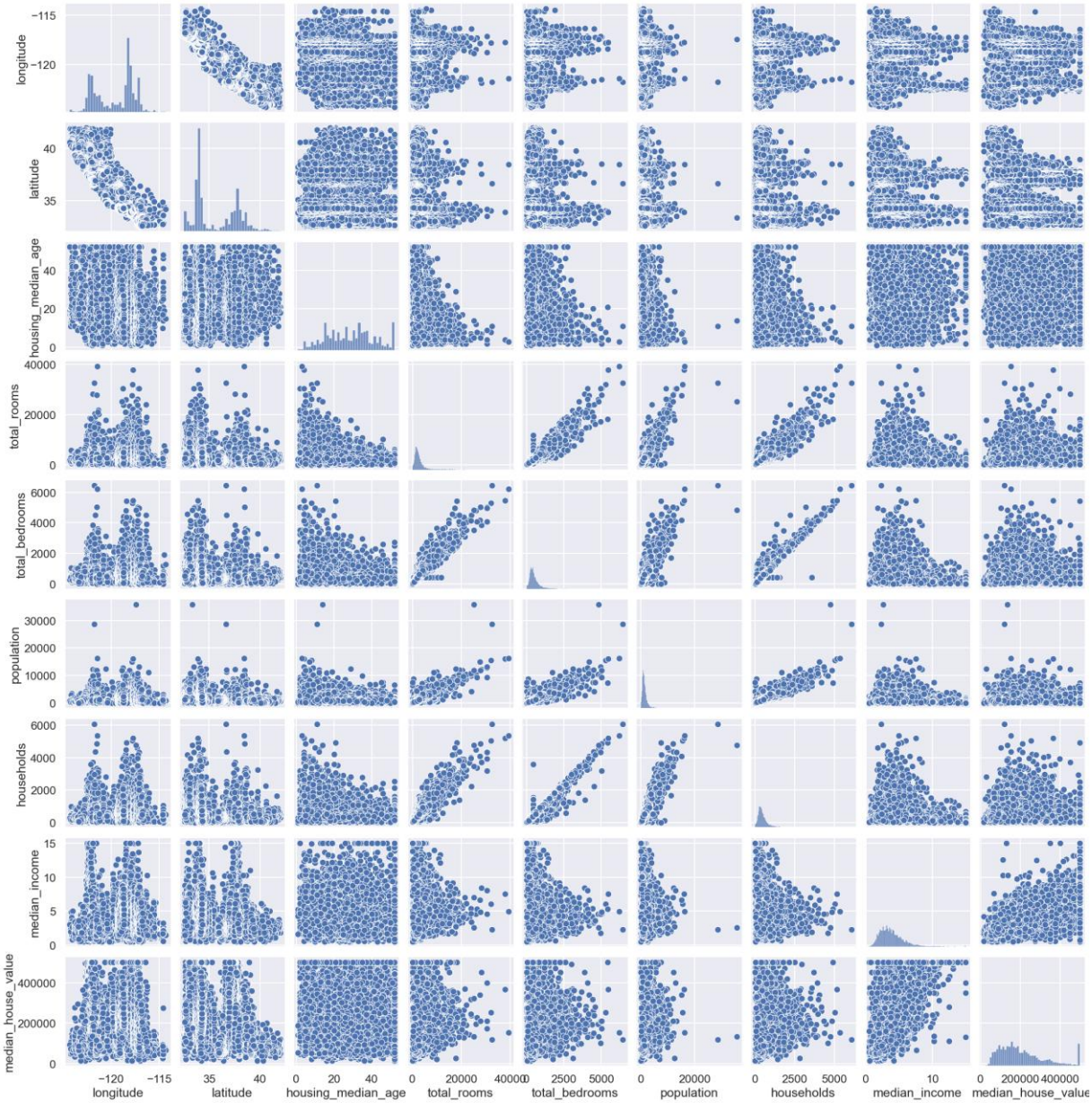

Figure 10 : Graphique de la tendance linéaire entre chaque colonne de données

4.10 Analyse des corrélations globales

Une analyse plus globale est réalisée à l'aide d'une matrice de corrélation. Le code calcule les coefficients de corrélation entre les variables numériques et les représente sous forme de carte de chaleur.

Cette visualisation synthétique permet d'identifier rapidement les variables fortement liées entre elles et de mieux comprendre la structure globale du jeu de données.

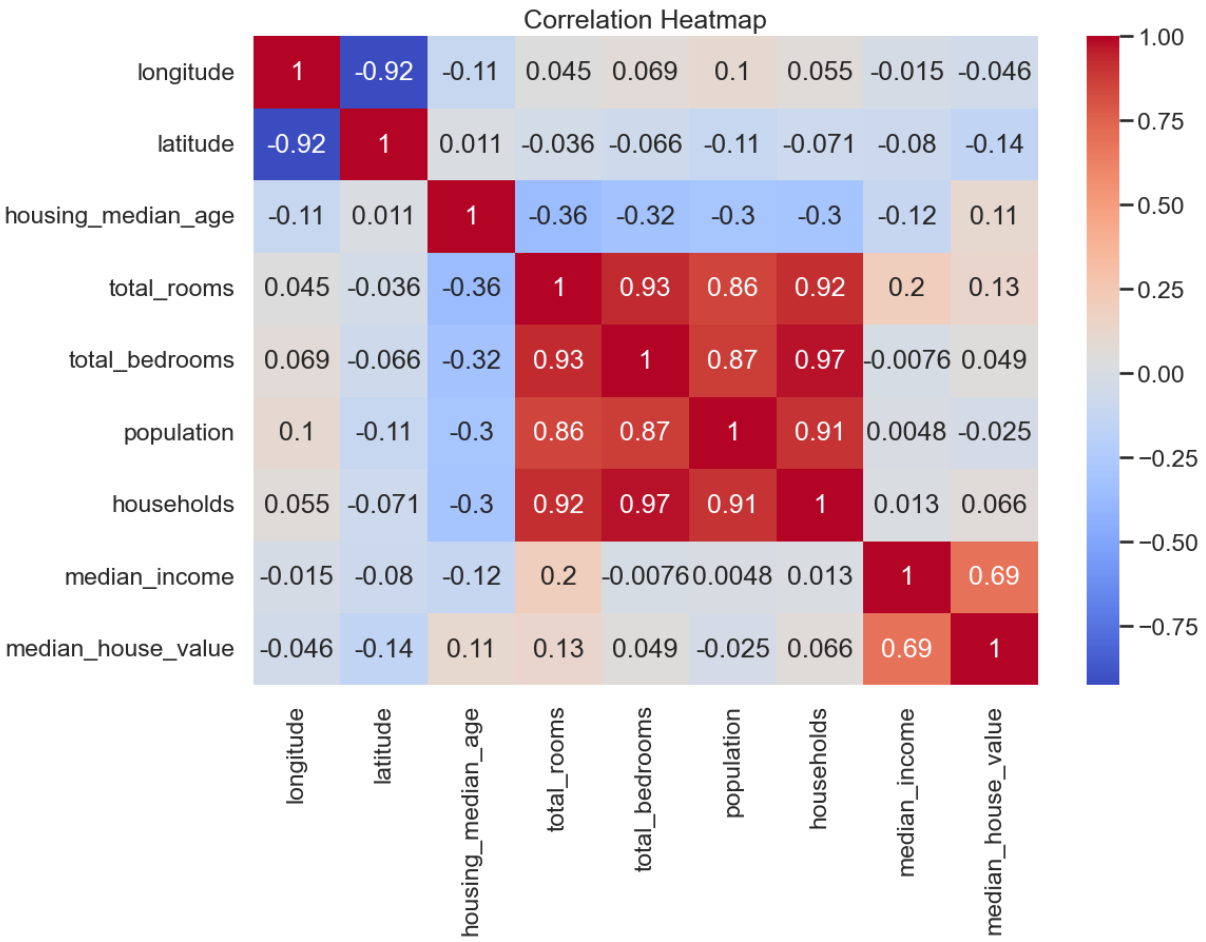

Figure 11 : Graphique mettant en avant la corrélation de chaque colonne avec la température de chaque bien immobilier

4.11 Analyse géographique des données

Le notebook exploite ensuite les variables géographiques pour produire des visualisations spatiales. Le code transforme les coordonnées de latitude et de longitude en cartes permettant de situer les observations dans l'espace.

Cette analyse replace les données dans leur contexte géographique réel et met en évidence des zones présentant des caractéristiques immobilières similaires.

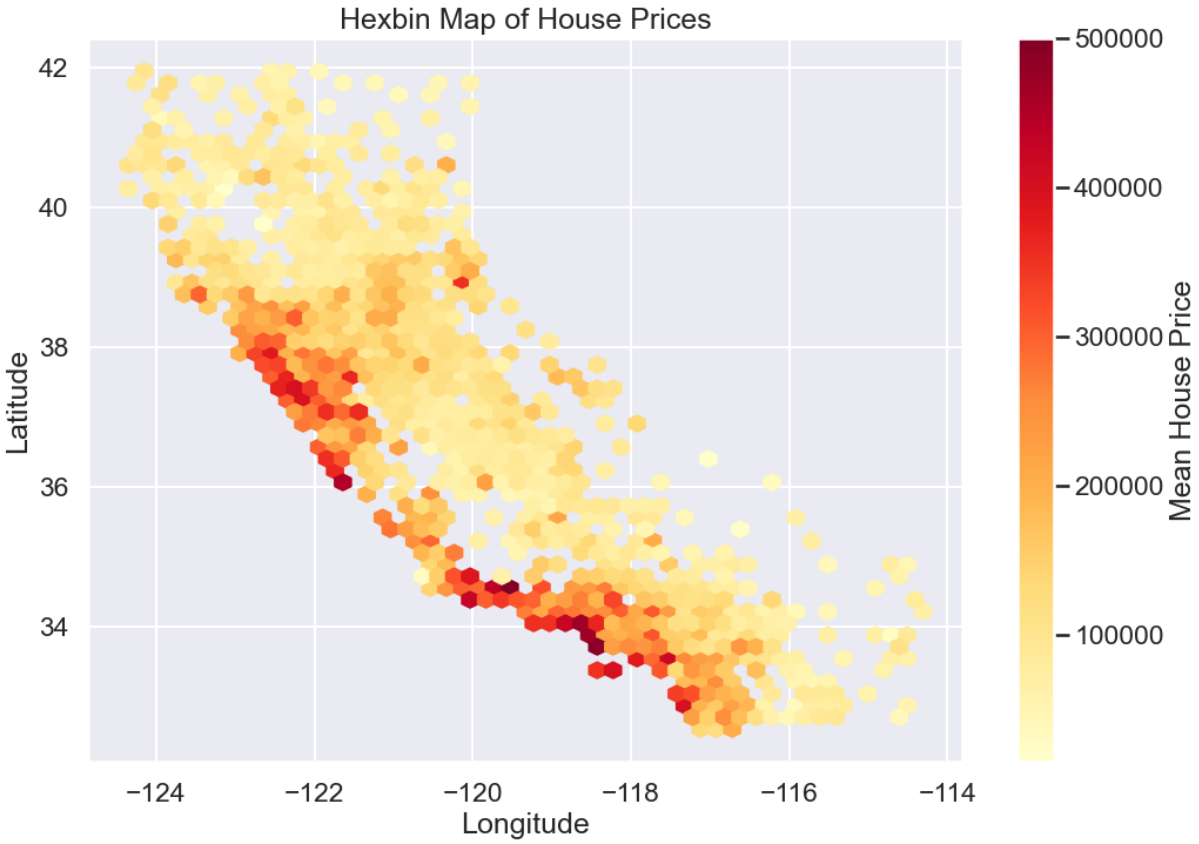

Figure 12 : Graphique permettant de visualiser la position géographique des biens et de leur prix

4.12 Cartes interactives et cartes de densité

Des visualisations géographiques plus avancées sont ensuite mises en place. Le code génère des cartes de densité et des cartes interactives permettant d'observer la concentration des prix immobiliers sur le territoire étudié.

Ces cartes offrent une lecture intuitive des données et constituent un support visuel particulièrement pertinent pour une interprétation métier.

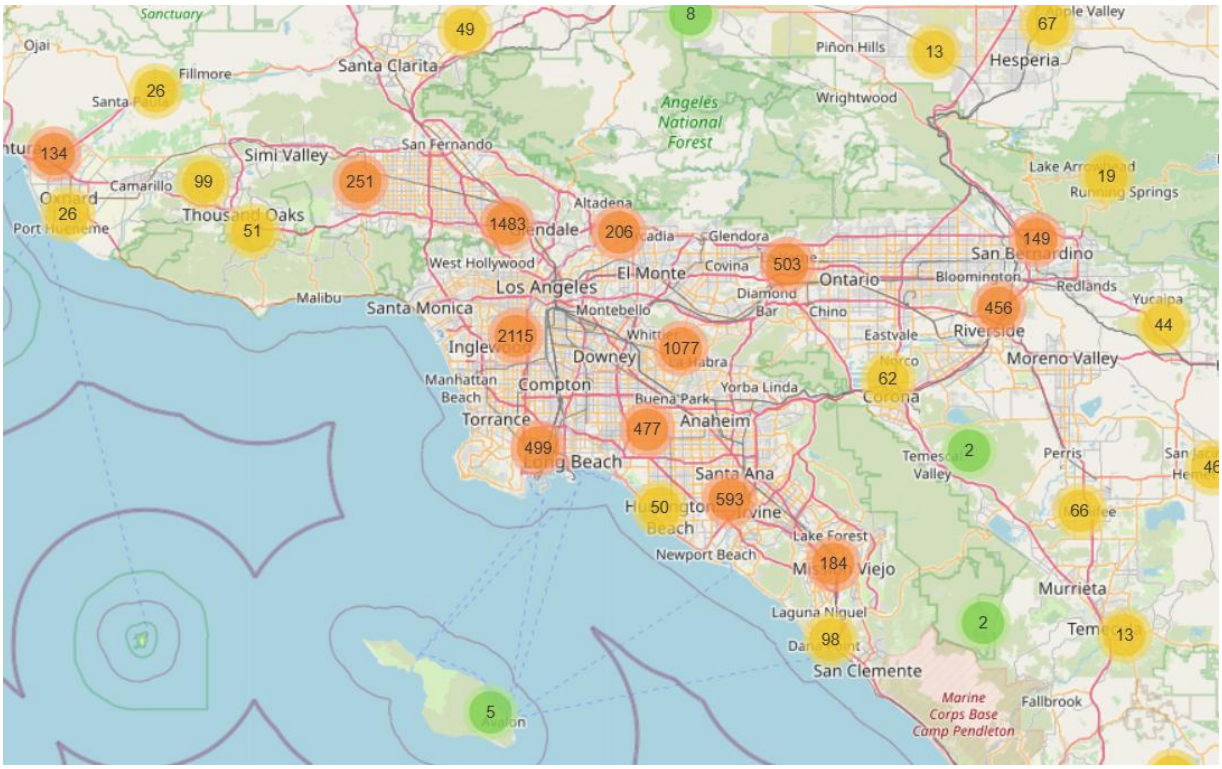

Figure 13 : Extrait de la carte interactives des biens immobiliers et précisant leur prix

4.13 Création et structuration d'un tableau de bord interactif

La dernière partie du notebook est consacrée à la création d'un tableau de bord interactif. Le code définit une interface web structurée en plusieurs sections, chacune regroupant un type d'analyse spécifique.

Cette application permet de centraliser l'ensemble des résultats produits au cours du workshop et de les rendre accessibles de manière interactive, facilitant ainsi l'exploration des données.

5 Validation des hypothèses

Hypothèse	Choix retenu	Justification
Les données doivent être anonymisées	Non	Pas besoin pour cette exercice
Utilisation de l'algorythme K-mean	Oui	Utile pour cet exercice de comparaison et de corrélation de paramètres
Normaliser les données	Oui	On a appliqué un filtre de remplacement des données par les médians en cas de données vides ou nulles.

6 Conclusion

Après avoir étudier les différentes notions de l'intelligence artificielle et de préparation de jeu de données, nous avons voir et comprendre les différentes étapes de la préparation d'un jeu de données pour ensuite les visualiser et les interpréter.