Les Plus Beaux Logis de Paris
Partie 1

Les Plus Beaux Logis de Paris

Analyse de l’évolution des prix de l’immobilier à Paris

Samuel
OSENAT
01/07/2025

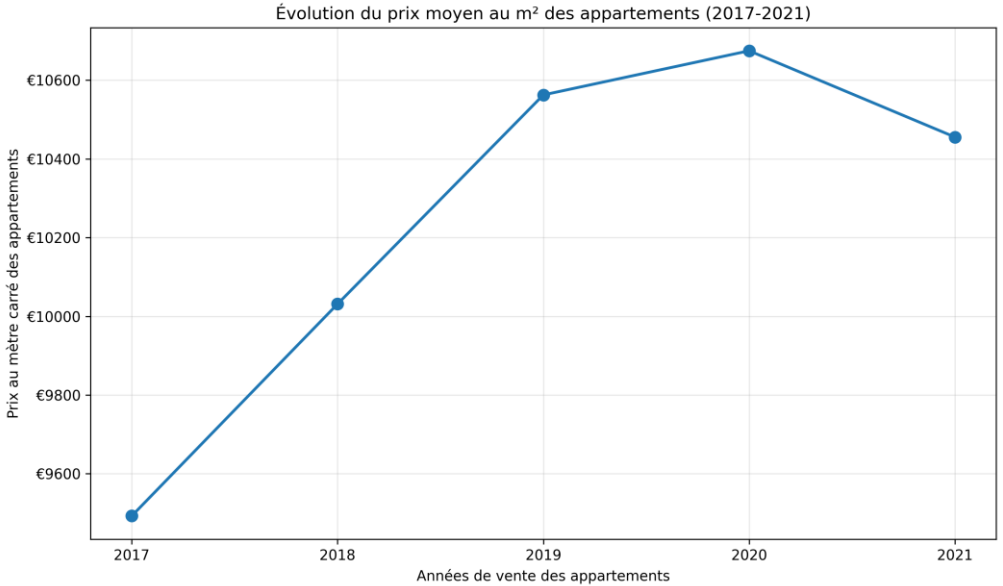
Optez toujours pour des slides allégées : 6 éléments par page maximum.

I. Analyse du marché de l'immobilier

- *Contexte : Les Plus Beaux Logis de Paris doivent vendre des actifs sur un des segments (corporate ou particulier) pour assurer la trésorerie*
- *Besoin : analyse des données pour aider à la prise de décision*
- *Analyse de 26196 transactions*
- *2 types de biens: Local industriel/commercial et les appartements*
- *Informations sur la date de vente, valeur foncière, informations géographiques (code postal)*

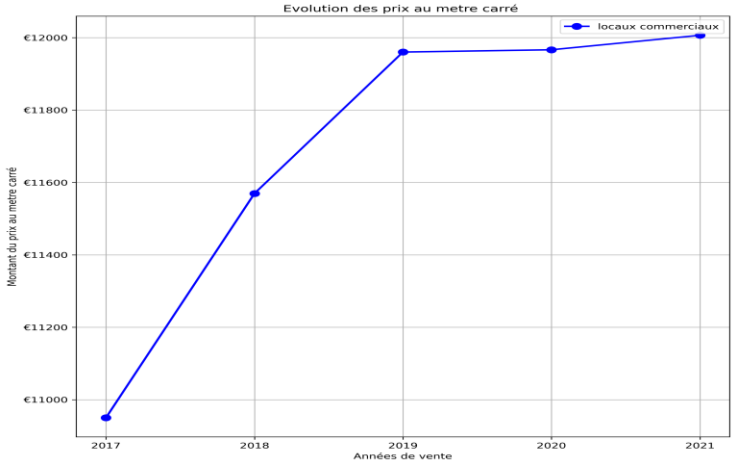
I. Analyse du marché de l'immobilier

- *Entre 2017 et 2021 on constate que le prix des appartements a augmenté. Ce constat est valable dans tous les arrondissements de Paris*



I. Analyse du marché de l'immobilier

- *Légère baisse des prix entre 2020 et 2021 -> crise économique liée à la pandémie du COVID19*
- *Locaux commerciaux et industriels : constat légèrement différent -> Augmentation globale des prix entre 2017 et 2021.*
- *Mais pas d'effets négatifs de la crise économique.*



II. Méthodologie suivie

- Identifications des facteurs qui influencent les prix dans l'immobilier
- Nettoyage et transformation des données
- Séparation des données en un jeu de données d'entrainement et un jeu de données de test.
- Entrainement de l'algorithme
- Utilisation de l'algorithme sur de nouvelles données (qui ont été nettoyés au préalable).

II. Méthodologie suivie

- Identifications des facteurs qui influencent les prix dans l'immobilier

```python
#On calcule la corrélation de Spearman
from scipy import stats
import numpy as np
from scipy.stats import pearsonr
# Données du 6ème arrondissement
df_6eme = df_appartements[df_appartements['arrondissement'] == 6].copy()

# Convertir les dates en valeurs numériques (nombre de jours depuis une date de référence)
date_reference = df_6eme['date_mutation'].min()
df_6eme['jours_depuis_ref'] = (df_6eme['date_mutation'] - date_reference).dt.days

# Calculer le coefficient de corrélation de Pearson
correlation, p_value = pearsonr(df_6eme['jours_depuis_ref'], df_6eme['prix_m2'])

print("=== COEFFICIENT DE CORRÉLATION DE PEARSON ===")
print(f"Coefficient de corrélation (r) : {correlation:.4f}")
print(f"P-value : {p_value:.2e}")
print(f"R² (coefficient de détermination) : {correlation**2:.4f}")
```

```
=== COEFFICIENT DE CORRÉLATION DE PEARSON ===
Coefficient de corrélation (r) : 0.9038
P-value : 7.11e-263
R² (coefficient de détermination) : 0.8169
```

Le coefficient de corrélation est de 0.9038 avec une pvalue de 7.11e-263 donc nous pouvons confirmer la corrélation.

```python
# Données du 6ème arrondissement
df_6eme = df_appartements[df_appartements['arrondissement'] == 6].copy()

# Calculer le coefficient de corrélation de Pearson avec p-value
correlation, p_value = pearsonr(df_6eme['surface_reelle'], df_6eme['valeur_fonciere'])

print("=== CORRÉLATION VALEUR FONCIÈRE - SURFACE ===")
print(f"Coefficient de corrélation (r) : {correlation:.4f}")
print(f"P-value : {p_value:.2e}")
print(f"R² (coefficient de détermination) : {correlation**2:.4f}")
```

```
=== CORRÉLATION VALEUR FONCIÈRE - SURFACE ===
Coefficient de corrélation (r) : 0.9978
P-value : 0.00e+00
```

II. Méthodologie suivie

- Nettoyage et transformation des données

- Transformation des variables textuelles en nombres

```
df['année'] = df['date_mutation'].dt.year
df['code_postal'] = df['code_postal'].astype(str)

print(df.dtypes)
# Sélection des colonnes utiles
colonnes_features = ['code_postal', 'type_local', 'surface_reelle', "année"]
df_ml = df[colonnes_features + ['valeur_fonciere']].copy()

# Transformation avec get_dummies()


# Appliquer get_dummies sur les colonnes catégoriques
df_encoded = pd.get_dummies(df_ml,
                            columns=['code_postal', 'type_local'],
                            prefix=['CP', 'TYPE'],
                            drop_first=True)
```

II. Méthodologie suivie

- Entrainement de l'algorithme

```
# On sépare le jeu de données entre echantillons d'apprentissage et de test
X = df_encoded.drop('valeur_fonciere', axis=1)  # Toutes les colonnes sauf la cible
y = df_encoded['valeur_fonciere']  # Variable à prédire
# La valeur y à trouver est la valeur foncière
X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.33,  # 33% pour le test
    random_state=42  # Pour la reproductibilité
)
```

```
from sklearn.linear_model import LinearRegression
#On entraîne l'algorithme ci-dessous et on effectue la prédiction
model = LinearRegression()
model.fit(X_train, y_train)

print(f"Nombre de coefficients : {len(model.coef_)}")
print(f"Intercept : {model.intercept_:,.2f}€")
```

```
Nombre de coefficients : 22
```

[33]:

```
y_pred = model.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
r2 = r2_score(y_test, y_pred)

# Erreur en pourcentage
erreur_pct = np.mean(np.abs((y_test - y_pred) / y_test)) * 100

print(f" MÉTRIQUES DE PERFORMANCE :")
print(f"- R² (qualité du modèle) : {r2:.3f}")
print(f"- Erreur absolue moyenne : {mae:,.0f}€")
print(f"- RMSE : {rmse:,.0f}€")
print(f"- Erreur moyenne en % : {erreur_pct:.2f}%")
```

```
 MÉTRIQUES DE PERFORMANCE :
- R² (qualité du modèle) : 0.986
- Erreur absolue moyenne : 34,424€
- RMSE : 78,279€
- Erreur moyenne en % : 9.49%
```

Vérification des performances de l'algorithme

III. Résultat des prédictions

```
#On effectue la prédiction
X_2022 = df_encoded_2022
y_pred=model.predict(X_2022)
#On vérifie les 10 premières valeurs

print(y_pred[:10])
```

```
[424705.4768635  362460.90572081 820619.14229955 872575.77089695
 318523.94991801 457036.90516274 383770.59114308 636092.51711023
 133279.30884188 135692.66461253]
```

```
df_ml_2022['valeur foncière prédite'] = y_pred
print(df_ml_2022.head())
```

	code_postal	type_local	surface_reelle	année	valeur foncière prédite
0	75001	Appartement	25	2022	424705.476863
1	75003	Appartement	22	2022	362460.905721
2	75003	Appartement	65	2022	820619.142300
3	75008	Appartement	74	2022	872575.770897
4	75008	Appartement	22	2022	318523.949918

Maintenant nous allons comparer la valorisation prédite pour les deux segments.

- *Valorisation prédite au 31 décembre 2022 :*
- *de 70 millions d'euros pour les appartements*
- *97 millions pour les locaux commerciaux et industriels.*

- *Attention : certains facteurs influençant le prix de l'immobilier ne sont peut etre pas pris en compte(nombre de pièces principales par exemple). L'analyse temporelle n'est pas assez poussée et certaines variables explicatives sont peut être corrélées (type de local et surface réelle).*

Les Plus Beaux Logis de Paris
Partie 2

I. Méthodologie suivie

- Choix de l'algorithme des K-Means.
- Permet de classer des objets similaires entre eux.

Classification des biens immobiliers avec comme facteur différenciant leur prix.

```python
from sklearn.cluster import KMeans

# PRÉPARATION DES DONNÉES
# Utiliser uniquement le prix au m² pour le clustering
X = df_classification[['prix_m2']].values
print(f"Prix au m² - Min: {df_classification['prix_m2'].min():.0f}€, Max: {df_classification['prix_m2'].max():.0f}€")

# APPLICATION DE K-MEANS (k=2)
kmeans = KMeans(n_clusters=2, random_state=42)
clusters = kmeans.fit_predict(X)

# Récupérer les centroïdes
centroides = kmeans.cluster_centers_
print(f"\n CENTROÏDES TROUVÉS :")
print(f"Centroïde 1 : {centroides[0][0]:.0f}€/m²")
print(f"Centroïde 2 : {centroides[1][0]:.0f}€/m²")

# AJOUTER LES RÉSULTATS AU DATAFRAME
df_classification['cluster'] = clusters
print(f"\nRépartition des clusters :")
print(df_classification['cluster'].value_counts().sort_index())
```

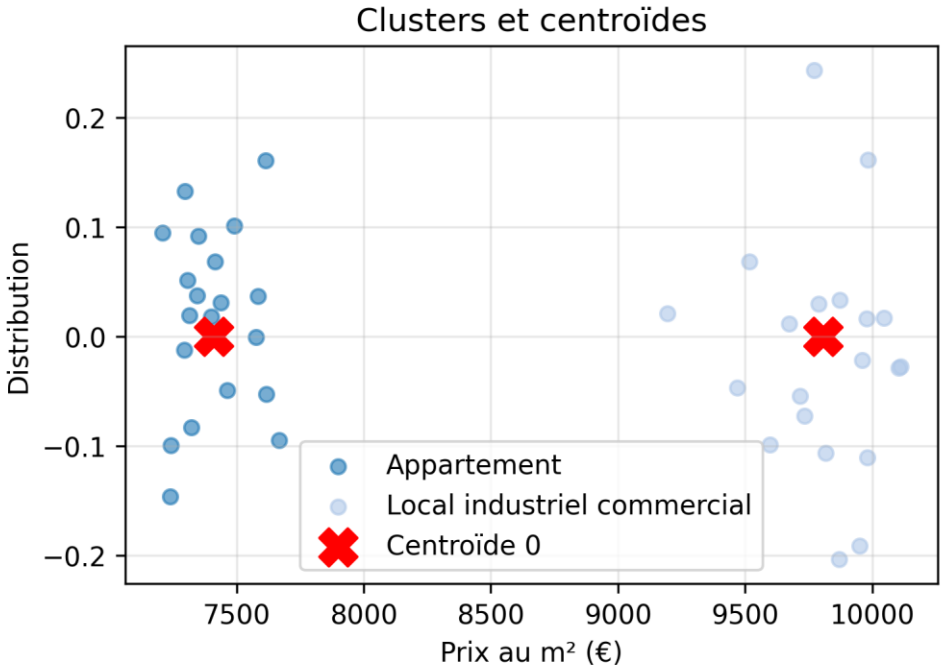
II. Résultat de la classification

- *On trouve deux clusters. 20 appartements et 20 locaux commerciaux.*
- *On considère que le cluster des locaux commerciaux est celui des biens avec le prix le plus élevé.*

Limites éventuelles de notre analyse :

- *Il n'y a que deux groupes -> non prise en compte des différents segments possibles.*
- *Des appartements peuvent être considérés par l'algorithme comme des locaux commerciaux en raison de leur prix élevé.*
- *Une seule variable est utilisée ce qui limite la fiabilité. La situation géographique des biens n'est pas prise en compte.*

II. Résultat de la classification



II. Résultat de la classification

- *Cet outil permet de détecter rapidement à quel type de bien l'équipe a affaire => Gain de temps*
- *Il peut permettre aussi de détecter les opportunités d'investissement : biens sous évalués.*

II. Conclusion

- *Utilisation d'un algorithme de régression linéaire pour prédire valorisation portefeuille actifs*
- *Outil qui aide à la prise de décision, mais il ne faut pas en faire une vérité absolue*
- *Entreprise Plus beaux logis => Focus Locaux Commerciaux ?*