الكلية متعددة التخصصات - ورزازات
ⵜⴰⵢⵜⵍⵍⴰⵢⵜ ⵜⴰⵅⵜⵙⵙⵉⵙⵜ – ⵡⵓⵔⵣⵣⴰⵣⵓⵜ
FACULTÉ POLYDISCIPLINAIRE DE OUARZAZATE

جامعة ابن زهر أكادير
UNIVERSITÉ IBN ZOHR - AGADIR

Faculté Polydisciplinaire d'Ouarzazate

House Price Prediction Using Machine Learning

Réalisé par : Amina El Helymy

Intelligence Artificielle et Ingénierie Logiciel

Année universitaire 2025-2026

Table des matières

Introduction

Le but de ce projet est de construire un modèle de Machine Learning capable de prédire le prix d'une maison en fonction de sa surface et du nombre de chambres. Ce projet permet de comprendre les concepts de features, target, entraînement et prédiction.

Objectif du projet

- ***Features* :** surface, rooms
- ***Target* :** price
- ***Objectif* :** entraîner le modèle pour apprendre la relation entre les features et le target et prédire les prix de nouvelles maisons.

Données utilisées (Data)

1	surface	rooms	price
2	193	7	276248
3	36	1	60499
4	100	4	145024
5	87	3	117972
6	218	8	313958
7	203	8	316870
8	169	6	238648
9	181	7	272025
10	38	1	39576
11	53	2	71764
12	89	3	132359
13	184	7	262669
14	173	6	247115
15	213	8	325895
16	209	8	317656
17	137	5	196623
18	144	5	217109
19	101	4	164723
20	31	1	55064
21	236	8	337431
22	208	8	312448

.....

190	230	8	350727
191	146	5	209547
192	138	5	213425
193	217	8	335069
194	172	6	261088
195	213	8	320547
196	69	2	90022
197	105	4	150133
198	44	1	64778
199	218	8	328366
200	45	1	71509
201	110	4	150873

Environnement de projet

Le projet a été développé en **Python**, en utilisant **Jupyter Notebook** pour exécuter et visualiser le code. Toutes les étapes, du chargement des données à l'entraînement et à la prédiction, ont été réalisées dans cet environnement.

L'environnement **Python + Jupyter Notebook** a permis d'exécuter les scripts de Machine Learning de manière interactive et de visualiser les résultats immédiatement. Les bibliothèques utilisées ont facilité la manipulation des données, l'entraînement du modèle et la visualisation des prédictions.

Élément	Version / Info
Langage	Python 3.x
IDE / Notebook	Jupyter Notebook / Anaconda
Bibliothèques utilisées	pandas, numpy, matplotlib, scikit-learn
OS	Windows
Excel	pour le fichier de données (`house_data_large.xlsx`)
Plotting / Visualisation	matplotlib
Random State	42 (pour train_test_split reproducible)

Préparation des données

- **X** et **Y** séparés
- Split Train/Test

```
X = data[['surface', 'rooms']]
y = data['price']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Nous avons séparé les features (surface et rooms) du target (price) et divisé les données en ensembles d'entraînement (train) et de test (test) pour pouvoir entraîner et évaluer le modèle correctement.

Entraînement (Training)

Création du modèle ***LinearRegression*** et training ***(model.fit())***

```
model = LinearRegression()
model.fit(X_train, y_train)
```

```
LinearRegression
LinearRegression()
```

Nous avons choisi un modèle de régression linéaire car la relation entre la surface, le nombre de chambres et le prix est linéaire. Le modèle apprend à partir des données d'entraînement.

Prédictions sur Test Set

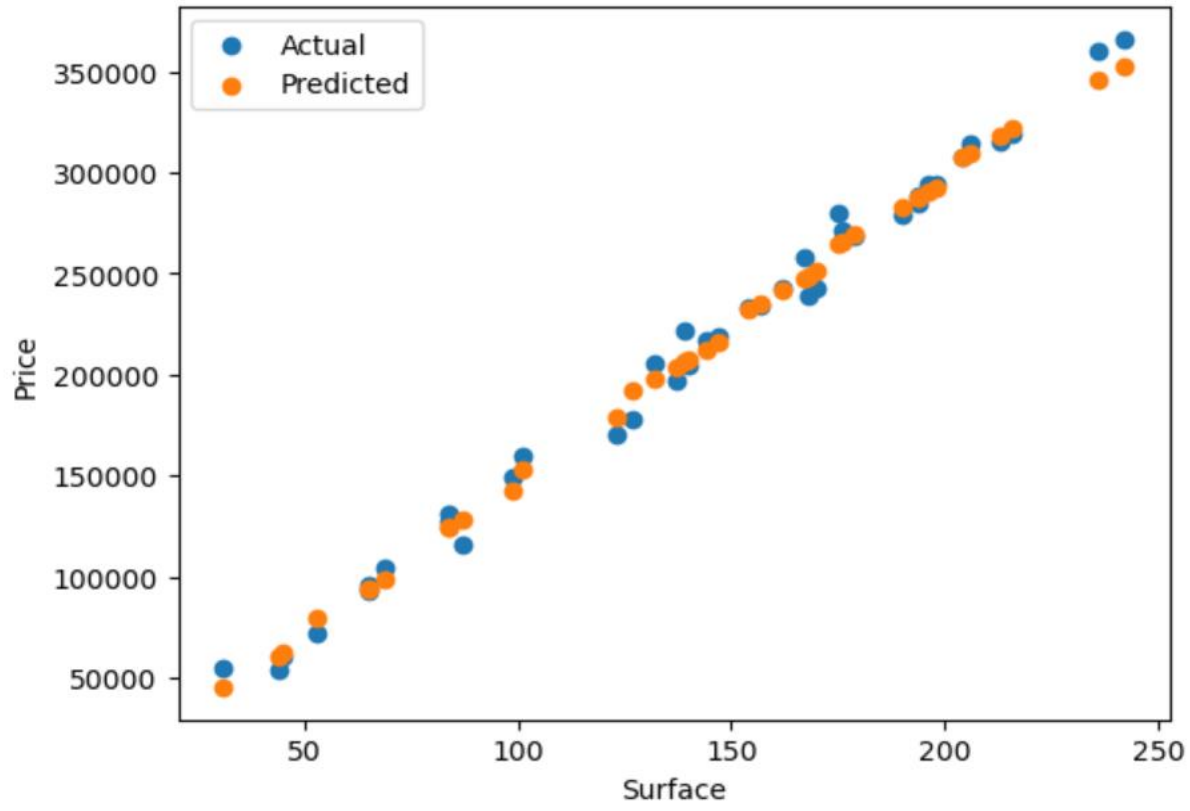
```
y_pred = model.predict(X_test)
```

Le modèle prédit les prix sur l'ensemble de test. Ces valeurs seront comparées aux prix réels pour évaluer la performance du modèle.

Visualisation Test Set

```
plt.scatter(X_test['surface'], y_test, label='Actual')
plt.scatter(X_test['surface'], y_pred, label='Predicted')
plt.xlabel('Surface')
plt.ylabel('Price')
plt.legend()
plt.show()
```

Le graphique montre les prix réels **(Actual)** et les prix prédits **(Predicted)** pour l'ensemble de test.



Prédictions sur nouvelles maisons

```
new_house = pd.DataFrame({
    'surface': [90, 110, 70],
    'rooms': [3, 4, 2]
})
predicted_prices = model.predict(new_house)
print("Predicted prices for new houses:\n", predicted_prices)
```

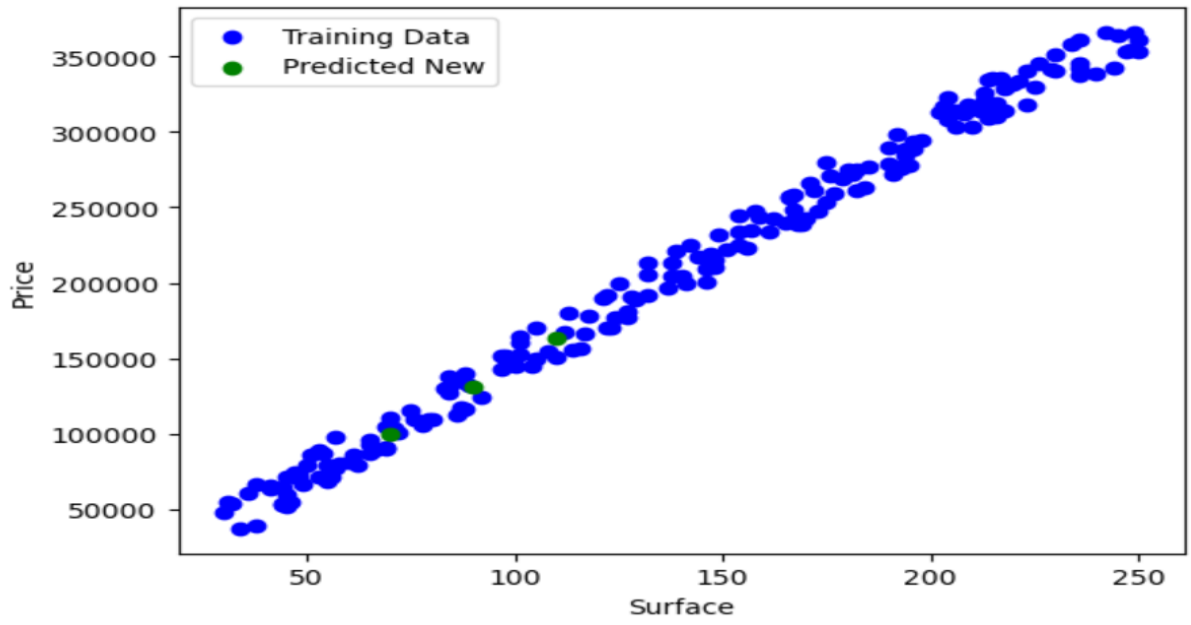
Le modèle prédit le prix des nouvelles maisons jamais vues auparavant. Les prédictions sont cohérentes avec la logique des données.

```
Predicted prices for new houses:
 [131570.42247723 163392.25746957  99748.58748488]
```

Visualisation avec nouvelles maisons

```
plt.scatter(X['surface'], y, color='blue', label='Training Data')
plt.scatter(new_house['surface'], predicted_prices, color='green', label='Predicted New')
plt.xlabel('Surface')
plt.ylabel('Price')
plt.legend()
plt.show()
```

Les points verts représentent les prix prédits pour les nouvelles maisons. Ils s'alignent avec la tendance observée dans les données d'entraînement.



Évaluation du modèle

```
from sklearn.metrics import mean_absolute_error, r2_score
r2 = r2_score(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
print("R²:", r2)
print("MAE:", mae)
```

R^2 proche de 1 indique que le modèle a très bien appris la relation entre les features et le target. MAE ≈ 6029 montre un écart moyen faible entre les valeurs réelles et prédites.

```
R²: 0.9925261546853733
MAE: 6029.503747607178
```

Conversion R^2: 0.9925 → 99.25/100 ou 9.9/10

Conclusion

Le projet a montré qu'un modèle de régression linéaire peut prédire avec précision le prix des maisons en fonction de leurs caractéristiques. Les métriques et les visualisations confirment que le modèle a appris correctement la relation entre les features et le target. Les prédictions pour de nouvelles maisons sont cohérentes et fiables.