



## **Rapport de rédaction du code**

Projet Web ML Housing California

Réalisé par :

**Abdelhamid SADOUDI**

## Sommaire

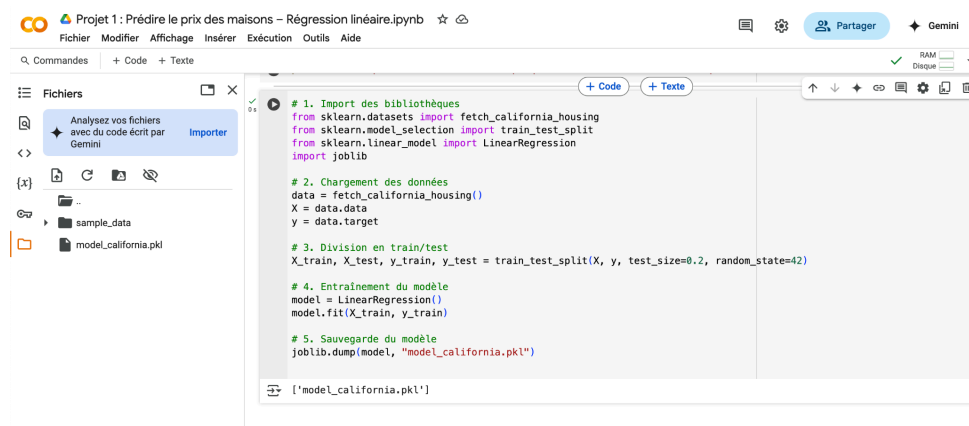
Introduction.....	3
1. Importation des bibliothèques.....	3
2. Chargement du modèle.....	4
3. Titre de l'application et description.....	4
4. Création du formulaire utilisateur.....	4
5. Prédiction et affichage des résultats.....	5
Conclusion.....	5

## Introduction

Le code que j'ai rédigé utilise la bibliothèque Streamlit pour créer une interface web interactive permettant aux utilisateurs de prédire le prix d'un bien immobilier en Californie à partir de plusieurs paramètres géographiques et démographiques.

Au départ, je voulais utiliser Flask, mais des erreurs d'installation m'ont empêché de poursuivre le projet efficacement, c'est pourquoi j'ai opté pour Streamlit.

Le modèle de prédiction a été préalablement entraîné sur un ensemble de données et sauvegardé sous le nom de fichier `model_california.pkl`



```
Projet 1 : Prédire le prix des maisons – Régression linéaire.ipynb
Fichier Modifier Affichage Insérer Exécution Outils Aide

Fichiers
- Analysez vos fichiers avec du code écrit par Gemini
- Importer
- sample_data
- model_california.pkl

# 1. Import des bibliothèques
from sklearn.datasets import fetch_california_housing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import joblib

# 2. Chargement des données
data = fetch_california_housing()
X = data.data
y = data.target

# 3. Division en train/test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 4. Entraînement du modèle
model = LinearRegression()
model.fit(X_train, y_train)

# 5. Sauvegarde du modèle
joblib.dump(model, "model_california.pkl")

['model_california.pkl']
```

le fichier de base est envoyé dans un autre devoir, nom du fichier : (Projet\_1\_Prédire\_le\_prix\_des\_maisons\_-\_Régression\_linéaire.ipynb).

Cette application permet d'interagir avec le modèle via des curseurs (sliders) pour entrer des valeurs et obtenir des résultats en temps réel.

### 1. Importation des bibliothèques

```
import streamlit as st
import numpy as np
import joblib
```

**Streamlit (st)** : la bibliothèque principale utilisée pour la création de l'application web.

**NumPy (np)** : une bibliothèque utilisée pour la gestion des données numériques, elle permet de transformer les données saisies en tableau NumPy, elle est donc compatible avec le modèle de prédiction.

**Joblib (joblib)** : on va l'utiliser pour charger le modèle préalablement entraîné et sauvegardé dans un fichier .pkl afin de pouvoir l'utiliser dans ce projet.

## 2. Chargement du modèle

```
model = joblib.load("model_california.pkl")
```

On va le charger en mémoire à l'aide de Joblib et l'utiliser pour faire des prédictions sur les données qu'on a.

## 3. Titre de l'application et description

```
st.title("🏠 Estimation du prix immobilier en Californie")
```

```
st.write("Remplis les champs ci-dessous pour obtenir une estimation du prix moyen d'un bien immobilier dans ce quartier (en $100k).")
```

**st.title** : Cette fonction permet d'afficher un titre visible sur l'application web. Dans ce cas, il s'agit de l'en-tête de l'application : "Estimation du prix immobilier en Californie".

**st.write** : Affiche un texte explicatif sous le titre, permettant à l'utilisateur de comprendre l'objectif de l'application. Cela le guide sur la manière d'utiliser l'application.

## 4. Création du formulaire utilisateur

```
medinc = st.slider("Revenu médian dans le quartier (MedInc)", 0.0, 20.0, 3.0)
house_age = st.slider("Âge moyen des maisons (HouseAge)", 1, 52, 20)
ave_rooms = st.slider("Nombre moyen de pièces (AveRooms)", 1.0, 50.0, 5.0)
ave_bedrms = st.slider("Nombre moyen de chambres (AveBedrms)", 0.5, 10.0, 1.0)
population = st.slider("Population du quartier", 100, 50000, 1500)
ave_occup = st.slider("Nombre moyen d'occupants par logement (AveOccup)", 1.0, 10.0, 3.0)
latitude = st.slider("Latitude", 32.0, 42.0, 36.0)
longitude = st.slider("Longitude", -125.0, -114.0, -120.0)
```

- Chaque ligne de code correspond à un **curseur** (ou slider) permettant d'entrer une valeur pour un paramètre spécifique.
- **Les paramètres sélectionnés** sont les suivants :
  - **MedInc** : revenu médian dans le quartier.
  - **HouseAge** : age moyen des maisons.
  - **AveRooms** : nombre moyen de pièces dans les maisons.
  - **AveBedrms** : nombre moyen de chambres dans les maisons.

- **Population** : taille de la population dans le quartier.
- **AveOccup** : nombre moyen d'occupants par logement.
- **Latitude et Longitude** : coordonnées géographiques du quartier.
- Ces curseurs permettent de spécifier des valeurs dans une plage définie, avec des valeurs par défaut pratiques (ex. 3.0 pour le revenu médian).

## 5. Prédiction et affichage des résultats

if st.button("Prédire") :

```

user_data = np.array([[medinc, house_age, ave_rooms, ave_bedrms, population,
ave_occup, latitude, longitude]])
prediction = model.predict(user_data)
st.success(f"🏠 Le prix estimé du logement est de **{prediction[0]:.2f} x $100k**, soit
environ **${prediction[0]*100_000:,.0f}**.")

```

Explication :

On va avoir besoin d'un bouton cliquable, on va le nommer "Prédire" pour lancer l'opération, donc de déclencher la prédiction. Une fois le bouton cliqué, l'application effectue l'action suivante.

Ensuite, les données saisies dans les curseurs sont stockées dans un tableau NumPy, qui est l'entrée requise par le modèle.

Ce modèle est utilisé pour prédire le prix de l'immobilier basé sur les données saisies. Cette prédiction est calculée et stockée dans `prediction`.

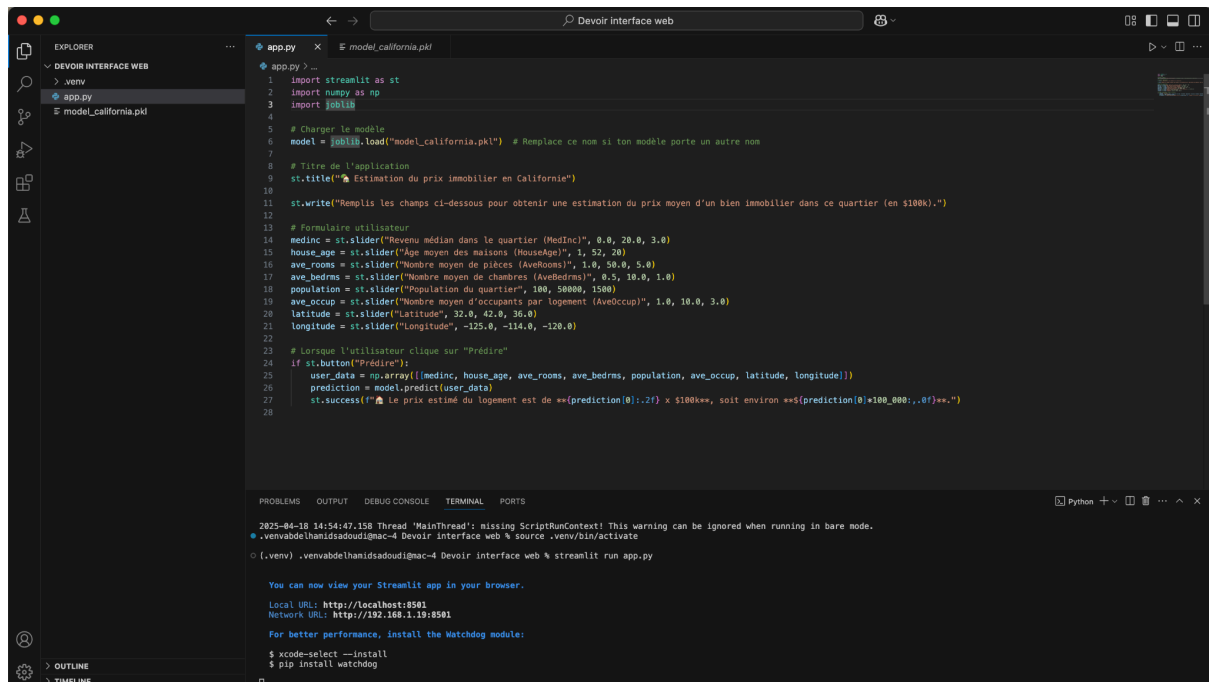
Une fois la prédiction effectuée, **Streamlit** affiche le résultat avec un message de succès, précisant le prix estimé sous forme de multiples de 100 000 \$ et en valeur absolue.

## Conclusion

Pour conclure, ce projet est très intéressant dans le cadre d'une implémentation d'une application de prédiction simple et intuitive (ici pour le prix de l'immobilier) en local host (voir la capture d'écran ci-dessous). Cette interface nous permet de manipuler des informations et d'obtenir instantanément une estimation de prix.

So it's done !

## Le code que j'ai lancé sur VSC (capture d'écran)



```
1 import streamlit as st
2 import numpy as np
3 import joblib
4
5 # Charger le modèle
6 model = joblib.load("model_california.pkl") # Remplace ce nom si ton modèle porte un autre nom
7
8 # Titre de l'application
9 st.title("🏠 Estimation du prix immobilier en Californie")
10
11 st.write("Remplis les champs ci-dessous pour obtenir une estimation du prix moyen d'un bien immobilier dans ce quartier (en $100k).")
12
13 # Formulaire utilisateur
14 medinc = st.slider("Revenu médian dans le quartier (MedInc)", 0.0, 20.0, 3.0)
15 house_age = st.slider("Âge moyen des maisons (HouseAge)", 1, 52, 20)
16 ave_rooms = st.slider("Nombre moyen de pièces (AveRooms)", 1.0, 50.0, 5.0)
17 ave_bedrms = st.slider("Nombre moyen de chambres (AveBedrms)", 0.5, 10.0, 1.0)
18 population = st.slider("Population du quartier", 100, 50000, 1500)
19 ave_occup = st.slider("Nombre moyen d'occupants par logement (AveOccup)", 1.0, 10.0, 3.0)
20 latitude = st.slider("Latitude", 32.0, 42.0, 36.0)
21 longitude = st.slider("Longitude", -125.0, -114.0, -120.0)
22
23 # Lorsque l'utilisateur clique sur "Prédire"
24 if st.button("Prédire"):
25     user_data = np.array([medinc, house_age, ave_rooms, ave_bedrms, population, ave_occup, latitude, longitude])
26     prediction = model.predict(user_data)
27     st.success(f"🏠 Le prix estimé du logement est de {(prediction[0]:.2f) * 100k}€, soit environ ${(prediction[0]*100_000):.0f}€.")
28
```

2025-04-18 14:54:47.158 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.  
• .venv\develhanidsadoudi@mac-4 Devoir interface web % source .venv/bin/activate  
• (.venv) .venv\develhanidsadoudi@mac-4 Devoir interface web % streamlit run app.py

You can now view your Streamlit app in your browser.

Local URL: <http://localhost:8501>  
Network URL: <http://192.168.1.19:8501>

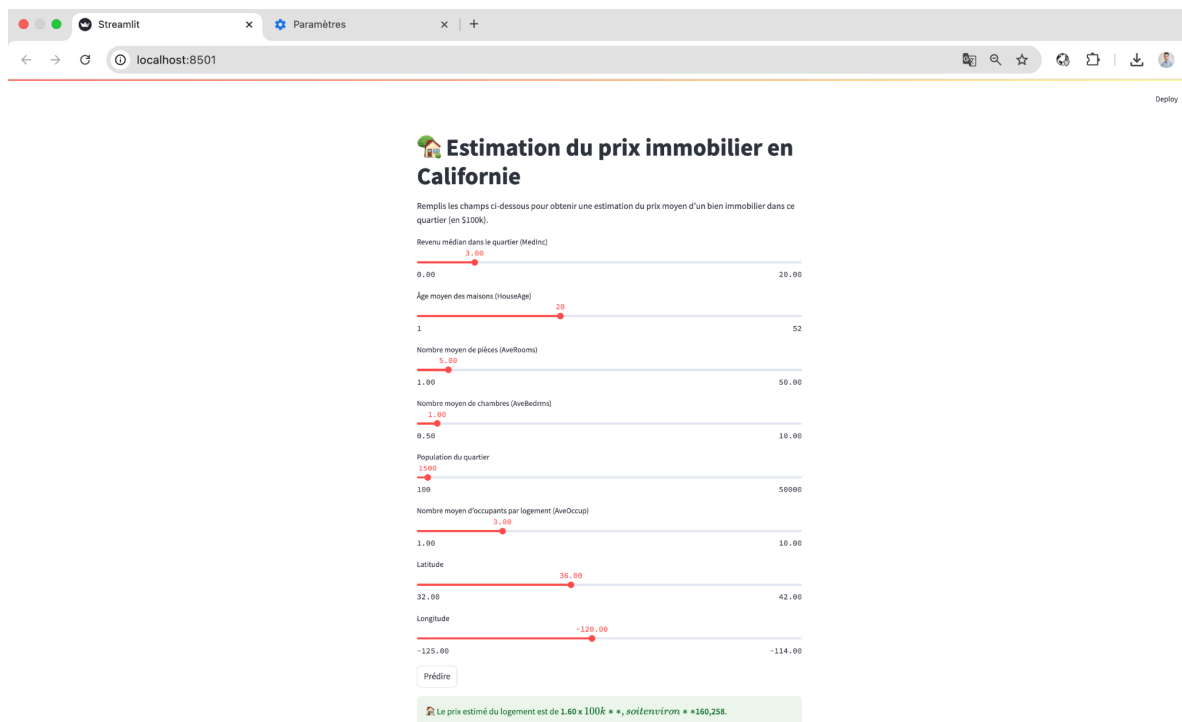
For better performance, install the Watchdog module:

```
$ xcode-select --install
$ pip install watchdog
```

## Streamlit (capture d'écran)

Adresse : localhost:8501

Première navigation sur l'interface :



**Estimation du prix immobilier en Californie**

Remplis les champs ci-dessous pour obtenir une estimation du prix moyen d'un bien immobilier dans ce quartier (en \$100k).

Revenu médian dans le quartier (MedInc): 3.00 (0.00 to 20.00)

Âge moyen des maisons (HouseAge): 20 (1 to 52)

Nombre moyen de pièces (AveRooms): 5.00 (1.00 to 50.00)

Nombre moyen de chambres (AveBedrms): 1.00 (0.50 to 10.00)

Population du quartier: 1500 (100 to 50000)

Nombre moyen d'occupants par logement (AveOccup): 3.00 (1.00 to 10.00)

Latitude: 36.00 (32.00 to 42.00)

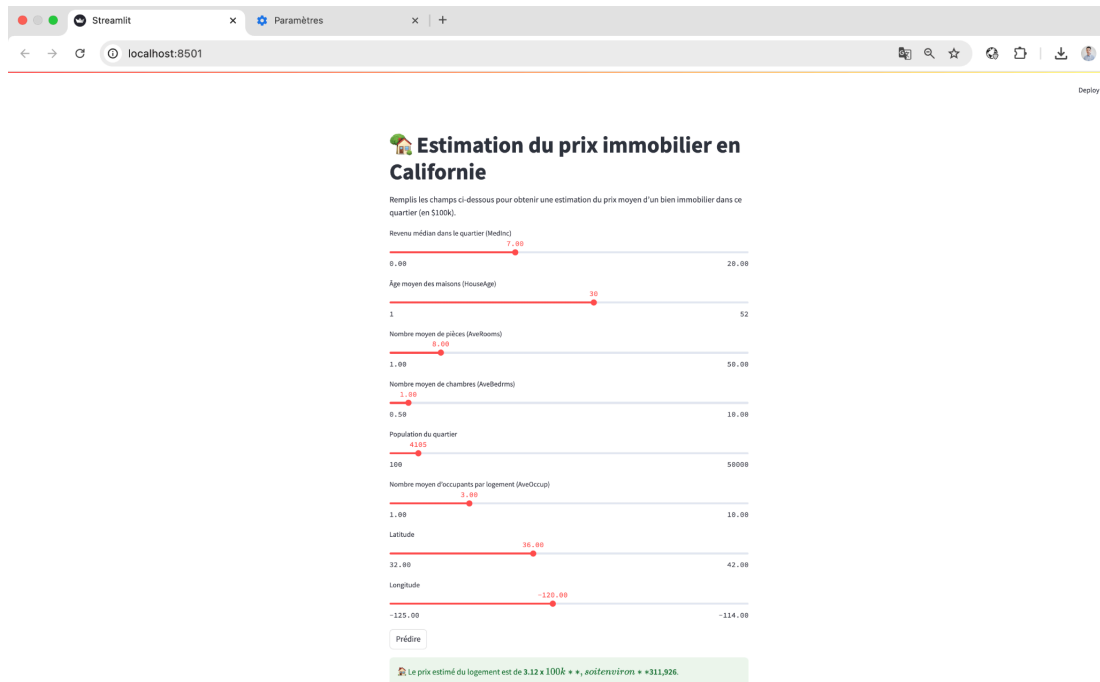
Longitude: -120.00 (-125.00 to -114.00)

Prédire

🏠 Le prix estimé du logement est de 1.60 x 100k €, soit environ \*160,258.

Avec les paramètres prédéfinis, le prix affiché est de **160,258 K\$ !**

Et si on modifie les données pour voir les changements ? : 🖱️



Les valeurs que j'ai modifiées :

- Revenu médian dans le quartier (MedInc)
- Âge moyen des maisons (HouseAge)
- Nombre moyen de pièces (AveRooms)
- Population du quartier

Avec ces paramètres, le prix passe à **311,926 K\$ !**