

Protocole de mise au point d'un modèle IA de prédiction des prix immobiliers

Cédric Tchack (44020455)

Anas Boutah (43014239)

Abdelkader Cherfaoui (43014163)

Université Paris Nanterre / L1 MIASHS – Semestre 2

23 mai 2025

Lien du projet GitHub

<https://github.com/offlinereal/prediction-bien-immobilier>

Résumé

Ce projet s'inscrit dans le cadre d'une initiation à la programmation assistée par intelligence artificielle. Il consiste à développer, en langage C, un outil capable d'estimer le prix d'un bien immobilier à partir de caractéristiques renseignées par l'utilisateur. Une attention particulière a été portée à l'intégration d'un modèle d'IA externe, utilisé non pas comme simple calculateur, mais comme assistant d'évaluation dans des situations où les données peuvent être incomplètes ou imprécises. Le travail ne s'est pas limité à la rédaction automatique de code par une IA : il a impliqué des choix de conception, des tests, des ajustements et une réflexion sur la pertinence des requêtes formulées à l'IA. Ce projet a ainsi permis d'explorer de manière concrète le dialogue entre un programme C et une intelligence artificielle via une API, tout en réfléchissant à la place que peut occuper l'IA dans des scénarios de programmation réels.

1 Introduction

L'estimation du prix d'un bien immobilier repose sur un grand nombre de paramètres : la surface, la localisation, le nombre de pièces, l'année de construction, l'état général du logement, etc.

Si certains de ces critères peuvent être facilement quantifiés, d'autres dépendent d'éléments moins objectifs ou simplement absents de l'information disponible.

Ce projet a pour but de concevoir un outil, écrit en langage C, qui assiste l'utilisateur dans cette tâche en combinant des saisies manuelles avec la puissance d'un modèle d'intelligence artificielle. L'enjeu est double : d'une part, permettre à un programme C de communiquer efficacement avec une API d'IA en générant dynamiquement un prompt pertinent, et d'autre part, explorer concrètement le rôle que peut jouer l'intelligence artificielle dans un processus de prise de décision technique.

Il s'agit ainsi d'un projet à la frontière entre développement algorithmique, conception de programme et utilisation raisonnée d'un outil d'IA générative. Cette interaction, bien qu'assistée, suppose une compréhension fine de la manière dont le programme structure les données avant de les transmettre à l'intelligence artificielle.

L'approche retenue ne prétend pas produire des estimations parfaites, mais vise à proposer un outil souple, capable de s'adapter à des données partielles tout en restant lisible et compréhensible, tant dans sa logique que dans son exécution.

Pour concevoir ce programme, on est parti d'un principe simple : un utilisateur qui cherche une estimation de prix ne dispose pas toujours de toutes les informations techniques sur le bien. Il fallait donc concevoir un programme souple, capable d'accepter des entrées partielles sans planter, et de les transmettre malgré tout à l'IA de manière structurée.

On a donc décidé d'utiliser le tiret " - " pour signaler qu'un champ est inconnu. Cette convention permet de construire dynamiquement un prompt adapté à la situation, sans forcer l'utilisateur à remplir toutes les données.

L'enjeu principal a ensuite été d'organiser clairement la communication entre le programme C et l'API d'intelligence artificielle. Le programme devait générer automatiquement un prompt textuel pertinent à partir des données, l'envoyer à l'IA, puis afficher le résultat à l'écran de manière lisible.

Enfin, on a mis un point d'honneur à produire un code accessible, à notre hauteur, avec des commentaires explicites et une organisation claire.

2 Méthodologie

Le travail s'est organisé selon une démarche rigoureuse en trois étapes principales : la conception du programme en langage C, la mise en place des échanges avec l'interface d'intelligence artificielle, et la validation par des tests d'interaction avec divers cas utilisateurs.

2.1 Présentation de l'IA utilisée

Nous avons opté pour le langage C car il constitue l'environnement de programmation que nous maîtrisons le mieux. Pour établir la communication avec le modèle d'intelligence artificielle, nous avons implémenté un appel système à la commande "curl". Cette solution technique permet d'effectuer des requêtes HTTP vers l'API OpenAI tout en transmettant une structure JSON contenant le prompt généré dynamiquement.

L'architecture du programme repose sur trois composants fonctionnels :

- Le module de collecte des données utilisateur,
- Le générateur de prompt,
- Le système d'envoi et de réception des requêtes.

Gestion des entrées utilisateur

Pour chaque paramètre descriptif du bien immobilier (comme la surface, la localisation géographique ou le nombre de pièces), l'utilisateur est invité à saisir une valeur. Une fonctionnalité importante permet de gérer les cas où l'utilisateur ne dispose pas d'une information particulière : la saisie du caractère "-" permet d'ignorer ce paramètre dans la construction ultérieure du prompt.

Construction dynamique du prompt

Les informations recueillies sont transformées en un texte structuré en langage naturel. Par exemple, pour un appartement de 60 mètres carrés comportant trois pièces et situé à Nanterre, le prompt généré serait : "Donne-moi une estimation du prix d'un bien immobilier situé à Nanterre. Il fait 60 m² et dispose de 3 pièces." Les paramètres non renseignés sont simplement omis, garantissant ainsi la cohérence de la requête.

Interaction avec l'IA

Le prompt final est incorporé dans une requête JSON conforme aux spécifications de l'API OpenAI. L'envoi s'effectue via la commande système "curl", et la réponse de l'intelligence artificielle, généralement sous forme de texte libre, est restituée dans l'interface du terminal.

3 Développement technique

3.1 Choix de l'approche

La solution technique retenue associe un programme écrit en langage C à l'API OpenAI, permettant de transformer les caractéristiques d'un bien immobilier en une requête naturelle. Cette approche évite le développement complexe d'un modèle de machine learning tout en offrant des estimations pertinentes grâce aux Large Language Models.

L'utilisation de la bibliothèque `libcurl` a été privilégiée pour sa robustesse dans les communications HTTP et sa compatibilité avec le langage C.

3.2 Structure générale du programme

L'architecture du programme suit un enchaînement logique :

- Saisie interactive des caractéristiques du bien (9 paramètres disponibles)
- Construction dynamique d'un prompt en langage naturel
- Communication directe avec l'API OpenAI via `libcurl`
- Récupération et affichage de la réponse formatée

3.3 Gestion flexible des entrées utilisateur

Le programme accepte des données partielles grâce à un système de validation intelligent. L'utilisateur peut saisir "-" pour les informations inconnues.

Listing 1 – Saisie des données utilisateur

```
char surface[MAX_INPUT_SIZE], nb_pieces[MAX_INPUT_SIZE];
char localisation[MAX_INPUT_SIZE], type_bien[MAX_INPUT_SIZE];
char annee[MAX_INPUT_SIZE], etat[MAX_INPUT_SIZE];
char exterieur[MAX_INPUT_SIZE], stationnement[MAX_INPUT_SIZE];
char ascenseur[MAX_INPUT_SIZE];
```

```
printf("Type_de_bien_(appartement/maison)_:_");
scanf("%[^\\n]", type_bien);
printf("Surface_(en_m2)_:_");
scanf("%[^\\n]", surface);
...
```

Explication :

- `%[^\\n]` permet de capturer des chaînes multi-mots, y compris avec espaces
- Chaque champ est vérifié pour ignorer les données absentes (marquées par "-")

3.4 Construction dynamique du prompt

Le prompt est progressivement enrichi selon les informations disponibles :

Listing 2 – Génération conditionnelle du prompt

```
char prompt[MAX_PROMPT_SIZE] = "Peux-tu_estimer_le_prix_d'un_bien_
    immobilier";

if (strcmp(type_bien, "-") != 0) {
    strcat(prompt, "_de_type_");
    strcat(prompt, type_bien);
}
if (strcmp(surface, "-") != 0) {
    strcat(prompt, "_de_");
    strcat(prompt, surface);
    strcat(prompt, "_m^2");
}
if (strcmp(nb_pieces, "-") != 0) {
    strcat(prompt, "_avec_");
    strcat(prompt, nb_pieces);
    strcat(prompt, "_pieces");
}
...
strcat(prompt, "_?");
```

Résultat typique :

"Peux-tu estimer le prix d'un bien immobilier de type maison de 110 m² avec 5 pièces, situé à Marseille, construit en 2005, en bon état, avec jardin, disposant d'un garage, avec ascenseur : non ?"

3.5 Communication avec l'API OpenAI

La requête API est construite dynamiquement au format JSON avec `snprintf` :

Listing 3 – Construction et envoi de la requête

```
char data[2048];
snprintf(data, sizeof(data),
    "{
    \"model\": \"gpt-3.5-turbo\",
    \"messages\": [
    { \"role\": \"system\", \"content\": \"Tu es un expert en estimation
        immobiliere, donne moi une fourchette de prix.\" },

    { \"role\": \"user\", \"content\": \"%s\" }
    ]
    }", prompt);

curl_easy_setopt(curl, CURLOPT_URL,
    "https://api.openai.com/v1/chat/completions");
curl_easy_setopt(curl, CURLOPT_POSTFIELDS, data);
...
```

Fonctionnalités clés :

- Format JSON conforme aux spécifications de l'API
- Authentification sécurisée avec header **Bearer**
- Prompt injecté directement dans la requête

3.6 Traitement de la réponse

Une structure dynamique permet d'accueillir la réponse de longueur variable :

Listing 4 – Gestion de la réponse OpenAI

```

struct string {
    char *ptr;
    size_t len;
};

void init_string(struct string *s) {
    s->len = 0;
    s->ptr = malloc(1);
    if (s->ptr == NULL) {
        fprintf(stderr, "Erreur_memoire\n");
        exit(EXIT_FAILURE);
    }
    s->ptr[0] = '\0';
}

size_t writefunc(void *ptr, size_t size,
    size_t nmemb, struct string *s) {
    size_t new_len = s->len + size * nmemb;
    s->ptr = realloc(s->ptr, new_len + 1);
    if (s->ptr == NULL) {
        fprintf(stderr, "Erreur_realloc\n");
        exit(EXIT_FAILURE);
    }
    memcpy(s->ptr + s->len, ptr, size * nmemb);
    s->ptr[new_len] = '\0';
    s->len = new_len;
    return size * nmemb;
}

```

Avantages :

- Allocation dynamique de la mémoire pour toute longueur de réponse
- Affichage final formaté dans le terminal avec surlignage
- Gestion sécurisée des erreurs mémoire

4 Résultats

Après avoir terminé le développement du programme en langage C, nous avons pu effectuer plusieurs tests d'exécution avec des cas concrets. À partir des données saisies par l'utilisateur, le programme construit dynamiquement un prompt en français, puis interroge l'API d'OpenAI pour obtenir une estimation immobilière basée sur des connaissances générales.

Voici un exemple d'exécution du programme :

Type de bien (appartement/maison) : appartement
Surface (en m²) : 45
Nombre de pièces : 2
Localisation (ville ou code postal) : Levallois-Perret
Année de construction : -
État du logement (neuf, bon état, à rénover) : neuf
Présence d'un extérieur (balcon, terrasse, jardin) : balcon
Type de stationnement (garage, parking, aucun) : parking
Présence d'un ascenseur (oui/non) : oui

Le prompt généré est :

Peux-tu estimer le prix d'un bien immobilier de type appartement de 45 m² avec 2 pièces, situé à Levallois-Perret, en neuf, avec balcon, disposant d'un parking, avec ascenseur : oui ?

La réponse obtenue par l'IA (OpenAI GPT-3.5-turbo) est la suivante :

En tenant compte des caractéristiques mentionnées (appartement neuf de 45 m² avec 2 pièces, balcon, parking et ascenseur à Levallois), je pourrais estimer le prix de ce bien immobilier entre 350 000€ et 450 000€. Cependant, il est important de souligner que cette estimation peut varier en fonction de plusieurs critères tels que l'emplacement exact, la qualité des finitions et des équipements du bien, ainsi que du marché immobilier local. Une expertise approfondie serait nécessaire pour obtenir une estimation plus précise.

Analyse des résultats

Ce résultat est cohérent avec le marché Levalloisien. Le modèle IA permet de produire une estimation rapide et contextualisée, même sans base de données locale. Le langage utilisé est clair, et l'intervalle proposé tient compte de l'incertitude. Cela illustre l'intérêt d'un assistant IA intégré dans un outil de pré-estimation immobilière.

Limites rencontrées

Les principales limites observées sont :

- Les résultats peuvent varier légèrement d'une exécution à l'autre (aléa génératif).
- L'IA n'intègre pas de données en temps réel (prix exacts du marché local).
- Une connexion Internet est nécessaire pour interroger l'API.

5 Discussion

Ce projet a été pour nous, étudiants de première année, une occasion concrète d’appliquer les notions que nous avons vues tout au long du semestre : la manipulation des chaînes de caractères, la gestion de la mémoire, les fonctions, les structures conditionnelles, et plus largement la construction logique d’un programme en langage C.

Ce qui nous a particulièrement motivés dans ce travail, c’est la possibilité de relier un langage que nous apprenons encore à maîtriser — le C — à une technologie d’actualité : l’intelligence artificielle. Ce lien donne une nouvelle dimension à notre apprentissage, en montrant qu’il est possible de faire dialoguer un programme C avec une IA à travers une API.

L’intérêt de l’IA dans notre approche

En utilisant une API d’intelligence artificielle (comme OpenAI), nous avons pu déléguer une partie du raisonnement — l’évaluation du prix d’un bien — à un modèle externe. Cela nous a permis de nous concentrer sur ce que nous avons appris en cours : demander proprement des informations à l’utilisateur, manipuler des chaînes de caractères, construire dynamiquement un message (prompt), et réfléchir à la structure du programme.

Nous avons conçu un programme capable de construire une question cohérente en fonction des données renseignées, et c’est cette question qui est ensuite transmise à l’IA. L’intelligence n’est pas dans notre code, mais notre code a été pensé pour bien exploiter cette intelligence.

Choix techniques et ce que nous avons appris

Nous avons décidé de permettre à l’utilisateur de taper « - » s’il ne connaît pas une information. Cette logique conditionnelle simple nous a permis de réutiliser les structures de contrôle étudiées en cours (comme les `if` et les `strcmp`), tout en rendant le programme plus souple.

Ce projet nous a aussi appris à faire attention à la taille des chaînes, à ne pas dépasser les limites fixées par les tableaux, et à organiser proprement notre code. Nous avons pris le temps de bien nommer les variables, de commenter certaines parties pour garder une trace de notre raisonnement, et de tester différentes entrées.

Un vrai projet de programmation

Enfin, ce projet nous a obligés à réfléchir comme des programmeurs. Même si nous avons été accompagnés par des outils d’intelligence artificielle pour certaines parties, la logique de base du programme, sa structure et ses conditions sont le fruit de ce que nous avons appris pendant le semestre. Et cela a été très formateur.

6 Conclusion

Ce projet nous a permis de donner du sens à nos apprentissages en programmation C. Il nous a confrontés à un vrai problème pratique, avec une logique à construire, des choix à faire, des limites techniques à surmonter. . . bref, une vraie expérience de développement logiciel, même à notre échelle.

En combinant les bases du C (structures conditionnelles, entrées utilisateur, chaînes de caractères, etc.) avec une ouverture vers le monde des API et de l'intelligence artificielle, nous avons réussi à produire un programme fonctionnel, qui interagit dynamiquement avec l'utilisateur pour construire une requête cohérente destinée à un agent IA. Et surtout : il la lui envoie réellement, ce qui boucle la chaîne du traitement.

Nous sommes fiers d'avoir mené ce projet jusqu'au bout. Nous avons dû réfléchir à la manière de guider l'utilisateur, de gérer des cas partiels, de composer une phrase naturelle, et d'utiliser une bibliothèque externe (`libcurl`) pour communiquer avec une API. Tout cela en respectant les contraintes de notre niveau et de notre formation.

Ce projet est donc autant une application de nos connaissances qu'une ouverture sur la suite : des projets plus ambitieux, plus structurés, avec potentiellement de vraies bases de données, des interfaces utilisateurs, ou des services en ligne.

Enfin, il nous a appris à collaborer efficacement, à tester, à corriger, à documenter, et à écrire un rapport dans une forme attendue. Autant de compétences qui nous serviront dans les années à venir.

A Annexe : Exemple d'exécution

Nous joignons ci-dessous deux captures d'écran issues de nos tests sur le terminal (cf. Figure 1 et Figure 2) :

```
PS C:\Users\Le Roi> cd "$HOME\Desktop"
PS C:\Users\Le Roi\Desktop> gcc estimations_immo.c -o estimations_immo.exe -lcurl
PS C:\Users\Le Roi\Desktop> .\estimations_immo.exe
Début du programme
Type de bien (appartement/maison) : appartement
Surface (en m²) : 45
Nombre de pieces : 2
Localisation (ville ou code postal) : Levallois
Annee de construction : -
Etat du logement (neuf, bon etat, a renover) : neuf
Presence d'un exterieur (balcon, terrasse, jardin) : balcon
Type de stationnement (garage, parking, aucun) : parking
Presence d'un ascenseur (oui/non) : oui

---
Prompt genere :
Peux-tu estimer le prix d'un bien immobilier de type appartement de 45 m² avec 2 pieces, situe a Levallois, en neuf, avec balcon, disposant d'un parking, avec ascenseur : oui ?
---
```

FIGURE 1 – Exécution d'un cas de test à Levallois

```
▲ ESTIMATION IMMOBILIÈRE POUR VOTRE BIEN :

{
  "id": "chatcmpl-82d4d6u2h7j3c7m61y4CnPjbcx0d22",
  "object": "chat.completion",
  "created": 1747984222,
  "model": "gpt-3.5-turbo-0125",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "En tenant compte des caractéristiques mentionnées (appartement neuf de 45 m² avec 2 pièces, balcon, parking et ascenseur à Levallois), je pourrais estimer le prix de ce bien immobilier entre 350 000€ et 450 000€. Cependant, il est important de souligner que cette estimation peut varier en fonction de plusieurs critères tels que l'emplacement exact, la qualité des finitions et des équipements du bien, ainsi que du marché immobilier local. Une expertise approfondie serait nécessaire pour obtenir une estimation plus précise.",
        "refusal": null,
        "annotations": []
      },
      "logprobs": null,
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 85,
    "completion_tokens": 133,
    "total_tokens": 218,
    "prompt_tokens_details": {
      "cached_tokens": 0,
      "audio_tokens": 0
    },
    "completion_tokens_details": {
      "reasoning_tokens": 0,
      "audio_tokens": 0,
      "accepted_prediction_tokens": 0,
      "rejected_prediction_tokens": 0
    }
  },
  "service_tier": "default",
  "system_fingerprint": null
}
```

PS C:\Users\Le Roi\Desktop>

FIGURE 2 – Réponse de l'IA pour un bien situé à Levallois