Prédiction Prix Logement Airbnb

Généré par Doxygen 1.8.17

Chapitre 1

Index des classes

1.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

Chapitre 2

Index des fichiers

2.1 Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

Chapitre 3

Documentation des classes

3.1 Référence de la structure defHousing

Attributs publics

- float accommodates
- float bedrooms
- float bathrooms
- float beds
- float price
- float min_nights
- float max_nights
- float number_of_reviews
- float * distance

3.1.1 Documentation des données membres

3.1.1.1 accommodates

```
float defHousing::accommodates
```

Nombre de places

3.1.1.2 bathrooms

```
float defHousing::bathrooms
```

Nombre de salles de bains

3.1.1.3 bedrooms

```
float defHousing::bedrooms
```

Nombre de chambres

3.1.1.4 beds

```
float defHousing::beds
```

Nombre de lits

3.1.1.5 distance

```
float* defHousing::distance
```

Tableau des distances

3.1.1.6 max_nights

```
float defHousing::max_nights
```

Nombre de nuits maximum

3.1.1.7 min_nights

```
float defHousing::min_nights
```

Nombre de nuits minimum

3.1.1.8 number_of_reviews

```
float defHousing::number_of_reviews
```

Nombre de reviews

3.1.1.9 price

```
float defHousing::price
```

Prix

La documentation de cette structure a été générée à partir du fichier suivant :
— src/structures.h

3.2 Référence de la structure defregisterMae

Attributs publics

— float mae
— int characteristic_num
— int k

3.2.1 Documentation des données membres

3.2.1.1 characteristic_num

```
int defregisterMae::characteristic_num
```

Numero de la caractèristique ayant donnée cette MAE

3.2.1.2 k

```
int defregisterMae::k
```

Valeur de k ayant donnée cette MAE

3.2.1.3 mae

```
float defregisterMae::mae
```

Valeur de la MAE

La documentation de cette structure a été générée à partir du fichier suivant :
— src/structures.h

3.3 Référence de la structure Housing

Structure d'un logement.

```
#include <structures.h>
```

3.3.1 Description détaillée

Structure d'un logement.

La documentation de cette structure a été générée à partir du fichier suivant :
— src/structures.h

3.4 Référence de la structure RegisterMae

Structure d'un enregistrement de mae.

```
#include <structures.h>
```

3.4.1 Description détaillée

Structure d'un enregistrement de mae.

La documentation de cette structure a été générée à partir du fichier suivant :
— src/structures.h

Chapitre 4

Documentation des fichiers

4.1 Référence du fichier src/defines.h

Fichier de définitions.

Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

4.2 Référence du fichier src/functions_algo_KNN.c

Fonctions principales de l'algorithme KNN.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <time.h>
#include "structures.h"
#include "defines.h"
#include "functions_algo_KNN.h"
#include "functions_fileToArray.h"
#include "functions_sort.h"
```

Graphe des dépendances par inclusion de functions_algo_KNN.c:

Fonctions

- void calculateDistance (Housing *housingArray, int housingArraylength, Housing housingX, char *characteristics[ ], int number_characteristics)
  Fonction calculant les distances entre tous les logements dont le prix est connu avec un logement dont on cherche à prédir le prix.
- Housing housingXDataFilling ()
  Permet d'entrer les valeurs des caractéristiques d'un logement.
- void shuffleHousingsArray (Housing *housingArray, size_t housingArrayLength)
  Fonction permettant de mélanger aléatoirement un tableau.
- float pricePrediction (Housing *housingArray, size_t housingArrayLength, int k)
  Fonction calculant le prix prédit.
- float calculateMAE (Housing *housingArray, size_t housingArrayLength, float *pricePredictions)
  Fonction calculant la MAE.
- int inputInt (int lowerLimit, int upperLimit)

Fonction d'entrée d'un entier dans des bornes.

— int inputNumCharac ()
Menu de choix de la caracteristique selon laquelle le prix va être prédit.

— void impementationKnnAlgorithm ()
Fonction principale permettant de prédire le prix d'un logement d'après ses caractéristique - Partie 1 du sujet.

— void evalPredictionModel ()
Fonction principale permettant de déterminer la caractéristique et le k permettant des prédictions plus précises - Partie 2(Bonus) du sujet.

4.2.1 Description détaillée

Fonctions principales de l'algorithme KNN.

Auteur

Lucas Velay

Date

10 décembre 2022

4.2.2 Documentation des fonctions

4.2.2.1 calculateDistance()

```
void calculateDistance (
            Housing * housingArray,
            int housingArraylength,
            Housing housingX,
            char * characteristics[],
            int number_characteristics )
```

Fonction calculant les distances entre tous les logements dont le prix est connu avec un logement dont on cherche à prédir le prix.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

housingArray	Tableau de logements d'on le prix est connu
housingArraylength	Taille du tableau de logements
housingX	Logement dont on cherche à déterminé le prix
characteristics	Tableau contenant le nom des caractéristiques d'un logement
number_characteristics	Nombre de caractéristiques

4.2.2.2 calculateMAE()

```
float calculateMAE (
            Housing * housingArray,
            size_t housingArrayLength,
            float * pricePredictions )
```

Fonction calculant la MAE.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

housingArray	Tableau de logements tests
housingArraylength	Taille du tableau de logements
pricePredictions	Tableau de prix prédit

Renvoie

MAE

4.2.2.3 evalPredictionModel()

```
void evalPredictionModel ( )
```

Fonction principale permettant de déterminer la caractéristique et le k permettant des prédictions plus précises - Partie 2(Bonus) du sujet.

Auteur

Lucas Velay

Date

10 décembre 2022

4.2.2.4 housingXDataFilling()

```
Housing housingXDataFilling ( )
```

Permet d'entrer les valeurs des caractéristiques d'un logement.

Auteur

Lucas Velay

Date

10 décembre 2022

Renvoie

Le logement

4.2.2.5 impementationKnnAlgorithm()

```
void impementationKnnAlgorithm ( )
```

Fonction principale permettant de prédire le prix d'un logement d'après ses caractéristique - Partie 1 du sujet.

Auteur

Lucas Velay

Date

10 décembre 2022

4.2.2.6 inputInt()

```
inputInt (
            int lowerLimit,
            int upperLimit )
```

Fonction d'entrée d'un entier dans des bornes.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

lowerLimit	Borne inférieur
upperLimit	Borne superieur

Renvoie

Valeur entrée

4.2.2.7 inputNumCharac()

```
int inputNumCharac ( )
```

Menu de choix de la caracteristique selon laquelle le prix va être prédit.

Auteur

Lucas Velay

Date

10 décembre 2022

Renvoie

Numero de la caracteristique

4.2.2.8 pricePrediction()

```
float pricePrediction (
            Housing * housingArray,
            size_t housingArrayLength,
            int k )
```

Fonction calculant le prix prédit.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

housingArray	Tableau de logements d'on le prix est connu
housingArraylength	Taille du tableau de logements
k	Paramètre k déterminant le nombre d'élément du tableau impliqués dans la moyenne déterminant le prix prédit

Renvoie

Prix prédit

4.2.2.9 shuffleHousingsArray()

```
void shuffleHousingsArray (
            Housing * housingArray,
            size_t housingArrayLength )
```

Fonction permettant de mélanger aléatoirement un tableau.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

housingArray	Tableau de logements d'on le prix est connu
housingArrayLength	Taille du tableau de logements

4.3 Référence du fichier src/functions_algo_KNN.h

Fichier d'en tête des fonctions de fonctions_algo_KNN.c.

Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Fonctions

- void **calculateDistance** (Housing housingArray[ ], int housingArraylength, Housing housingX, char ∗characteristics[ ], int number_characteristics)
- Housing housingXDataFilling ()
  Permet d'entrer les valeurs des caractéristiques d'un logement.
- void impementationKnnAlgorithm ()
  Fonction principale permettant de prédire le prix d'un logement d'après ses caractéristique - Partie 1 du sujet.
- void evalPredictionModel ()
  Fonction principale permettant de déterminer la caractéristique et le k permettant des prédictions plus précises - Partie 2(Bonus) du sujet.
- int inputInt (int lowerLimit, int upperLimit)
  Fonction d'entrée d'un entier dans des bornes.
- int inputNumCharac ()
  Menu de choix de la caracteristique selon laquelle le prix va être prédit.
- int **inputK** (int housingArrayLength)

4.3.1 Description détaillée

Fichier d'en tête des fonctions de fonctions_algo_KNN.c.

Auteur

Lucas Velay

Date

10 décembre 2022

4.3.2 Documentation des fonctions

4.3.2.1 evalPredictionModel()

```
void evalPredictionModel ( )
```

Fonction principale permettant de déterminer la caractéristique et le k permettant des prédictions plus précises - Partie 2(Bonus) du sujet.

Auteur

Lucas Velay

Date

10 décembre 2022

4.3.2.2 housingXDataFilling()

```
Housing housingXDataFilling ( )
```

Permet d'entrer les valeurs des caractéristiques d'un logement.

Auteur

Lucas Velay

Date

10 décembre 2022

Renvoie

Le logement

4.3.2.3 impementationKnnAlgorithm()

```
void impementationKnnAlgorithm ( )
```

Fonction principale permettant de prédire le prix d'un logement d'après ses caractéristique - Partie 1 du sujet.

Auteur

Lucas Velay

Date

10 décembre 2022

4.3.2.4 inputInt()

```
int inputInt (
            int lowerLimit,
            int upperLimit )
```

Fonction d'entrée d'un entier dans des bornes.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

lowerLimit	Borne inférieur
upperLimit	Borne superieur

Renvoie

Valeur entrée

4.3.2.5 inputNumCharac()

```
int inputNumCharac ( )
```

Menu de choix de la caracteristique selon laquelle le prix va être prédit.

Auteur

Lucas Velay

Date

10 décembre 2022

Renvoie

Numero de la caracteristique

4.4 Référence du fichier src/functions_sort.c

Fonctions traitant des fichiers et des tableaux.

```
#include <stdio.h>
#include <stdlib.h>
#include "structures.h"
#include "defines.h"
#include "functions_sort.h"
```

Graphe des dépendances par inclusion de functions_sort.c:

Fonctions

- int floatComparator0 (const void *first, const void *second)
  Fonction permettant de comparer le champ 0 du tableau de distance.
- int floatComparator1 (const void *first, const void *second)
  Fonction permettant de comparer le champ 1 du tableau de distance.
- int floatComparator2 (const void *first, const void *second)
  Fonction permettant de comparer le champ 2 du tableau de distance.
- int floatComparator3 (const void *first, const void *second)
  Fonction permettant de comparer le champ 3 du tableau de distance.
- int floatComparator4 (const void *first, const void *second)
  Fonction permettant de comparer le champ 4 du tableau de distance.
- int floatComparator5 (const void *first, const void *second)
  Fonction permettant de comparer le champ 5 du tableau de distance.
- int floatComparator6 (const void *first, const void *second)
  Fonction permettant de comparer le champ 6 du tableau de distance.
- void sortHousingArray (Housing *housingArray, size_t housingArrayLength, int num_charac_sort)
  Fonction de tri du tableau de logements utilisant un switch pour choisir le champ du tableau de ditance à utiliser pour le tri.
- int maeComparator (const void *first, const void *second)
  Fonction permettant de comparer le tableau de MAE.

4.4.1 Description détaillée

Fonctions traitant des fichiers et des tableaux.

Fonctions de tri.

Auteur

Lucas Velay

Date

10 décembre 2022

4.4.2 Documentation des fonctions

4.4.2.1 floatComparator0()

```
int floatComparator0 (
            const void * first,
            const void * second )
```

Fonction permettant de comparer le champ 0 du tableau de distance.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

first	Première valeur à comparer
second	Seconde valeur à comparer

Renvoie

l'état de la comparaison

4.4.2.2 floatComparator1()

```
int floatComparator1 (
            const void * first,
            const void * second )
```

Fonction permettant de comparer le champ 1 du tableau de distance.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

first	Première valeur à comparer
second	Seconde valeur à comparer

Renvoie

l'état de la comparaison

4.4.2.3 floatComparator2()

```
int floatComparator2 (
            const void * first,
            const void * second )
```

Fonction permettant de comparer le champ 2 du tableau de distance.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

first	Première valeur à comparer
second	Seconde valeur à comparer

Renvoie

l'état de la comparaison

4.4.2.4 floatComparator3()

```
int floatComparator3 (
            const void * first,
            const void * second )
```

Fonction permettant de comparer le champ 3 du tableau de distance.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

first	Première valeur à comparer
second	Seconde valeur à comparer

Renvoie

l'état de la comparaison

4.4.2.5 floatComparator4()

```
int floatComparator4 (
            const void * first,
            const void * second )
```

Fonction permettant de comparer le champ 4 du tableau de distance.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

first	Première valeur à comparer
second	Seconde valeur à comparer

Renvoie

l'état de la comparaison

4.4.2.6 floatComparator5()

```
int floatComparator5 (
            const void * first,
            const void * second )
```

Fonction permettant de comparer le champ 5 du tableau de distance.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

first	Première valeur à comparer
second	Seconde valeur à comparer

Renvoie

l'état de la comparaison

4.4.2.7 floatComparator6()

```
int floatComparator6 (
            const void * first,
            const void * second )
```

Fonction permettant de comparer le champ 6 du tableau de distance.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

first	Première valeur à comparer
second	Seconde valeur à comparer

Renvoie

l'état de la comparaison

4.4.2.8 maeComparator()

```
int maeComparator (
            const void * first,
            const void * second )
```

Fonction permettant de comparer le tableau de MAE.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

first	Première valeur à comparer
second	Seconde valeur à comparer

Renvoie

l'état de la comparaison

4.4.2.9 sortHousingArray()

```
void sortHousingArray (
            Housing * housingArray,
            size_t housingArrayLength,
            int num_charac_sort )
```

Fonction de tri du tableau de logements utilisant un switch pour choisir le champ du tableau de ditance à utiliser pour le tri.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

housingArray	Tableau de logements à trier
housingArrayLength	Taille du tableau de logements à trier
num_charac_sort	Le numero de la caractèristique selon laquelle le tableau est trié

Renvoie

l'état de la comparaison

4.5 Référence du fichier src/functions_sort.h

Fichier d'en tête des fonctions de fonctions_sort.c.

Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Fonctions

- int floatComparator0 (const void *first, const void *second)
  Fonction permettant de comparer le champ 0 du tableau de distance.
- int floatComparator1 (const void *first, const void *second)
  Fonction permettant de comparer le champ 1 du tableau de distance.
- int floatComparator2 (const void *first, const void *second)
  Fonction permettant de comparer le champ 2 du tableau de distance.
- int floatComparator3 (const void *first, const void *second)
  Fonction permettant de comparer le champ 3 du tableau de distance.
- int floatComparator4 (const void *first, const void *second)
  Fonction permettant de comparer le champ 4 du tableau de distance.

- int floatComparator5 (const void *first, const void *second)
  Fonction permettant de comparer le champ 5 du tableau de distance.
- int floatComparator6 (const void *first, const void *second)
  Fonction permettant de comparer le champ 6 du tableau de distance.
- void sortHousingArray (Housing *housingArray, size_t housingArrayLength, int num_charac_sort)
  Fonction de tri du tableau de logements utilisant un switch pour choisir le champ du tableau de ditance à utiliser pour le tri.
- int maeComparator (const void *first, const void *second)
  Fonction permettant de comparer le tableau de MAE.

4.5.1 Description détaillée

Fichier d'en tête des fonctions de fonctions_sort.c.

Auteur

Lucas Velay

Date

10 décembre 2022

4.5.2 Documentation des fonctions

4.5.2.1 floatComparator0()

```
int floatComparator0 (
            const void * first,
            const void * second )
```

Fonction permettant de comparer le champ 0 du tableau de distance.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

first	Première valeur à comparer
second	Seconde valeur à comparer

Renvoie

l'état de la comparaison

4.5.2.2 floatComparator1()

```
int floatComparator1 (
            const void * first,
            const void * second )
```

Fonction permettant de comparer le champ 1 du tableau de distance.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

first	Première valeur à comparer
second	Seconde valeur à comparer

Renvoie

l'état de la comparaison

4.5.2.3 floatComparator2()

```
int floatComparator2 (
            const void * first,
            const void * second )
```

Fonction permettant de comparer le champ 2 du tableau de distance.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

first	Première valeur à comparer
second	Seconde valeur à comparer

Renvoie

l'état de la comparaison

4.5.2.4 floatComparator3()

```
int floatComparator3 (
            const void * first,
            const void * second )
```

Fonction permettant de comparer le champ 3 du tableau de distance.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

first	Première valeur à comparer
second	Seconde valeur à comparer

Renvoie

l'état de la comparaison

4.5.2.5 floatComparator4()

```
int floatComparator4 (
            const void * first,
            const void * second )
```

Fonction permettant de comparer le champ 4 du tableau de distance.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

first	Première valeur à comparer
second	Seconde valeur à comparer

Renvoie

l'état de la comparaison

4.5.2.6 floatComparator5()

```
int floatComparator5 (
            const void * first,
            const void * second )
```

Fonction permettant de comparer le champ 5 du tableau de distance.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

first	Première valeur à comparer
second	Seconde valeur à comparer

Renvoie

l'état de la comparaison

4.5.2.7 floatComparator6()

```
int floatComparator6 (
            const void * first,
            const void * second )
```

Fonction permettant de comparer le champ 6 du tableau de distance.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

first	Première valeur à comparer
second	Seconde valeur à comparer

Renvoie

l'état de la comparaison

4.5.2.8 maeComparator()

```
int maeComparator (
            const void * first,
            const void * second )
```

Fonction permettant de comparer le tableau de MAE.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

first	Première valeur à comparer
second	Seconde valeur à comparer

Renvoie

l'état de la comparaison

4.5.2.9 sortHousingArray()

```
void sortHousingArray (
            Housing * housingArray,
            size_t housingArrayLength,
            int num_charac_sort )
```

Fonction de tri du tableau de logements utilisant un switch pour choisir le champ du tableau de ditance à utiliser pour le tri.

Auteur

Lucas Velay

Date

10 décembre 2022

Paramètres

housingArray	Tableau de logements à trier
housingArrayLength	Taille du tableau de logements à trier
num_charac_sort	Le numero de la caractèristique selon laquelle le tableau est trié

Renvoie

l'état de la comparaison

4.6 Référence du fichier src/main.c

Main program.

```
#include <stdio.h>
#include <stdlib.h>
#include "structures.h"
#include "defines.h"
#include "functions_algo_KNN.h"
#include "functions_sort.h"
```

Graphe des dépendances par inclusion de main.c:

Fonctions

— int **main** (int argc, char const ∗argv[ ])

4.6.1 Description détaillée

Main program.

Auteur

Lucas Velay

Date

10 décembre 2022

4.7 Référence du fichier src/structures.h

Fichier de structures.

Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

— struct defHousing
— struct defregisterMae

Définitions de type

— typedef struct defHousing **Housing**
— typedef struct defregisterMae **RegisterMae**

4.7.1 Description détaillée

Fichier de structures.

Auteur

Lucas Velay

Date

10 décembre 2022

Index