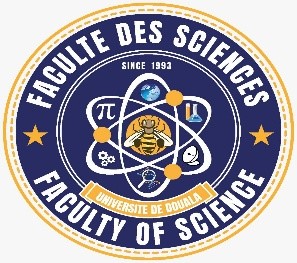
****

**RÉPUBLIQUE DU CAMEROUN**

**Paix-Travail-Patrie**

**\*\*\*\*\*\*\*\*\*\*\*\*\***

**MINISTERE DE L’ENSEIGNEMENT SUPERIEUR**

**\*\*\*\*\*\*\*\*\*\*\*\*\***

**UNIVERSITE DE DOUALA**

**\*\*\*\*\*\*\*\*\*\*\*\*\***

**DEPARTEMENT DE MATHEMATIQUE-INFOMATIQUE**

**INFORMATIQUE L2**

**REPUBLIC OF CAMEROON**

**Peace – Work – Fatherland**

**\*\*\*\*\*\*\*\*\*\*\*\*\***

**MINISTRY OF SECONDARY SCHOOL**

**\*\*\*\*\*\*\*\*\*\*\*\*\***

**UNIVERSITY OF DOUALA**

**\*\*\*\*\*\*\*\*\*\*\*\*\***

**DEPARTMENT OF MATHEMATICS-COMPUTER SCIENCE**

**COMPUTER SCIENCE L2**

***PROJET***

**Arbre de Décision sur la gestion des Données commerciales**

**LISTE DES MEMBRES DU GROUPE 3**

|  |  |  |
| --- | --- | --- |
| MATRICULE | NOMS | PRENOMS |
| 21S56800 | FOKA MAGHEN | YANN BRONDON |
| 21S56824 | FOTSO TEWE | GUSTAVE |
| 20S42750 | MENGOULOUNE | ROUKAYATOU |
| 20S43010 | NJIKE KOUDJA | CHRISTIAN |
| 21S57227 | NOUPOUE MBOURGA | DESVAN-KEROL |
| 20S47746 | TAPI KENFACK | STEPHANE |
| 21S57305 | TCHAKAM | DUPLEX CEDRIC |

Enseignant Chef de groupe  
**DR. JUSTIN MOSKOLAI FOKA MAGHEN YANN BRODON**

***NOVEMBRE 2023­­­‑‑‑‑***

Table des matières

[I. Introduction 2](#_Toc151451627)

[II. Arbre de decision 2](#_Toc151451628)

[A. Definition 2](#_Toc151451629)

[B. Mécanisme de fonctionnement 2](#_Toc151451630)

[C. Avantage et Inconient 3](#_Toc151451631)

[1. Avantage 3](#_Toc151451632)

[2. Inconvenients 3](#_Toc151451633)

[D. Rôle et importance dans la gestion des donnees commerciales 3](#_Toc151451634)

[III. Jeu de donnee (dataset) 4](#_Toc151451635)

[A. description 4](#_Toc151451636)

[B. Pré\_traitement des données 5](#_Toc151451637)

[C. Choix des variables d’entree et de sortie pour la construction de l’arbre de decision 7](#_Toc151451638)

[IV. Construction de L’arbre de decision 8](#_Toc151451639)

[A. Choix des criteres de division et des mesures d’evaluations 8](#_Toc151451640)

[B. Construction de l’arbre 9](#_Toc151451641)

[V. Evaluation et validation de l’arbre de decision 9](#_Toc151451642)

[A. Mesure de la performance de l’arbre 9](#_Toc151451643)

[B. Test pour evaluer la generalisation de l’arbre 10](#_Toc151451644)

[VI. Application Pratique et recommandation 11](#_Toc151451645)

[A. Prise de decision commercial A partir de l’arbre de decision 11](#_Toc151451646)

[B. Recommandation 12](#_Toc151451647)

[VII. Conclusion 12](#_Toc151451648)

# Introduction

De nos jours la gestion des données devient de plus en plus un aspect important avec le flux de données énormes dans le monde. A cet effet l’utilisation de ces données nécessite des traitements au préalable qui relève de l’analyse des données. Avec La naissance de l’intelligence artificielle notamment de l’apprentissage automatique nous pouvons faire facilement ces analyses en optimisant, prédisant les résultats et aussi faire des recommandations en fonction de ces résultats. Dans le cadre des données commerciales, qui sont les informations collectées dans le cadre d’activité commerciale tel que les données sur les ventes, les clients, les produits, les transactions, les préférences des utilisateurs…. Ces données servent à prendre des décisions commerciales pour éclairer et améliorer les performances de l’entreprise. Nous nous intéresserons sur la prédiction des prix de vente des maisons du domaine de l’immobilier. La réalisation de cette prédiction sera basée sur un modèle d’apprentissage automatique : l’arbre de décision et sera implémenté avec le langage python.

# Arbre de décision

## Définition

Un arbre de décision est un modèle d’apprentissage automatique qui utilise une structure en forme d’arbre pour prendre des décisions en se basant sur des données. Plus précisément, Un algorithme “arbre de décision” estime un concept cible par une représentation d’arbre, où chaque nœud interne correspond à un attribut, et chaque nœud terminal (ou feuille) correspond à une classe. Il est composé de nœuds, qui représentent des tests sur les caractéristiques des données, et de branches, qui relient les nœuds et représentent les différentes issues possibles. Chaque nœud interne de l’arbre représente un test sur une caractéristique, tandis que les feuilles représentent les résultats ou les décisions finales.

Pour construire un arbre de décision, on peut utiliser les modèles ID3(introduction of Decision Trees) Il se base sur le concept d’attributs et de classe de l’apprentissage automatique (sur classification discrète 3). Cet algorithme recherche l’attribut le plus pertinent à tester pour que l’arbre soit le plus court et optimisé possible, C4.5, souvent appelé classificateur statistique, **C4.5** est une extension de l’algorithme ID3, qui est également utilisé pour générer des arbres de décision ; CART(Classification and Regression Trees) ou CHAD(Chi-squared Automatic Interaction Detection), Cet algorithme est similaire à l’algorithme ID3, mais il utilise une méthode différente pour la sélection de l’attribut de partitionnement. CART utilise une mesure appelée **indice de Gini** pour mesurer l’impureté d’un nœud. L’indice de Gini mesure la probabilité qu’un élément choisi au hasard dans un nœud soit incorrectement classé s’il était étiqueté au hasard selon la distribution des étiquettes dans le nœud.

## Mécanisme de fonctionnement

Il y a différentes approches de construction d’un arbre de décision, approche descendante et approche ascendante. Nous nous focaliserons sur l’approche descendante(top-down).

Nous commencerons par déterminer quel variable a\* sera placé comme racine de l’arbre, ensuite diviser l’ensemble d’apprentissage D en sous ensemble {D1,D2,D3,…Dn} où chaque Di contient des exemples ayant la même valeur a\* et enfin récursivement appliquer le même procéder sur chaque nouveau sous ensemble jusqu’à ce que les exemples aient la même classe.

## Avantage et Inconvénient

### Avantage

La principale avantage est que les arbres de décision sont Ils sont faciles à comprendre et à interpréter, de plus avec les arbres de décision on peut faire la classification et la régression, on peut l’utilisés pour des données qualitatives et quantitatives, pour des données continues et discrètes, des données bruyantes, des manquantes, des données non linéaires, pour des données multi classe et des données multi-étiquettes.

### Inconvénients

Les arbres de décision ont de nombreux inconvénients :

Ils peuvent être instables, c’est-à-dire que de petites variations dans les données peuvent entraîner de grands changements dans l’arbre résultant.

Ils peuvent être biaisés, c’est-à-dire qu’ils peuvent privilégier les variables avec de nombreux niveaux ou catégories.

Ils peuvent être sur ajustés, c’est-à-dire qu’ils peuvent être trop complexes et ne pas généraliser bien aux données de test.

Ils peuvent être sensibles aux erreurs de mesure et aux données manquantes.

## Rôle et importance dans la gestion des données commerciales

Les arbres de décision sont des outils d’aide à la décision qui permettent de classer les différentes informations de façon logique et visuelle. Cette méthode permet non seulement de présenter les informations visuellement mais également de les hiérarchiser d’une manière claire. Cet outil facilite ainsi les décisions et limite grandement le risque de surcharge informationnelle.

Un arbre de décision vous permet d’analyser, par le biais d’une représentation visuelle, les résultats potentiels, les coûts et les conséquences d’une décision complexe. Ces outils s’avèrent particulièrement utiles pour analyser des données quantitatives et prendre une décision fondée sur des éléments chiffrés. Les arbres de décision sont largement utilisés dans divers domaines tels que la finance, la médecine, l’ingénierie, la recherche opérationnelle, etc. Ils sont également utilisés dans les systèmes d’information pour la classification et la prédiction. Dans notre cas il nous servira a analyser les données commerciales afin de prendre des décisions plus précisément de prédire le prix des maisons.

# Jeu de donnée (dataset)

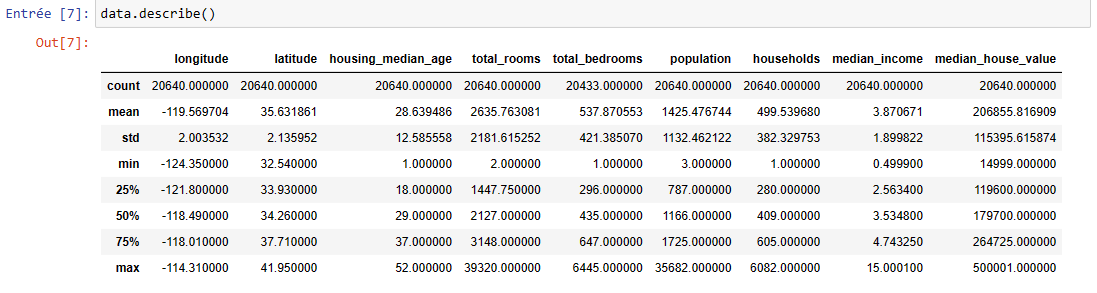
## Description

Le jeu de donnée utilisé est « california jousing » contient des informations sur les logements en californie aux états unies ainsi que leur valeur.

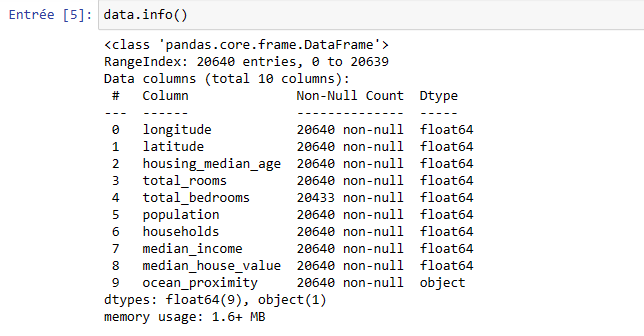
L’objectif de ce projet est de prédire la valeur des logements en fonction de leurs caractéristiques.

* Les variables présentes dans lq dataset sont :
* Longitude de la localisation de logement
* Latitude de la localisation de logement
* Housing\_median\_age pour l’age median des logements dans la région
* total\_room le nombre total d’appartement dans la région
* total\_bedroom pour le nombre total de chambre dans la région
* population pour la population totale dans la région-
* households pour le nombre total de ménage dans la région
* median\_income pour les revenues médiane de ménage dans la région
* median\_house\_value pour la valeur médiane des maisons dans la région
* ocean\_proximity pour la proximité de la maison par rapport à l’océan

Grace a la fonction describe() nous avons une description complète de notre dataset.



La fonction info() nous donne les informations sur la qualité des variables



Là on voit bien qu’on a 9 variables quantitative et 1 variable qualitative ‘ocean\_proximity’.

## Prétraitement des données

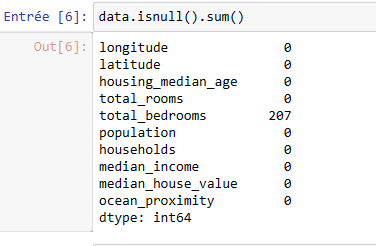
Dans cette phase, nous procédons à :

* Le nettoyage des données :

Cette phase consiste à enlever/remplacer les données manquantes à l’aide des méthodes prédéfinies tels que :

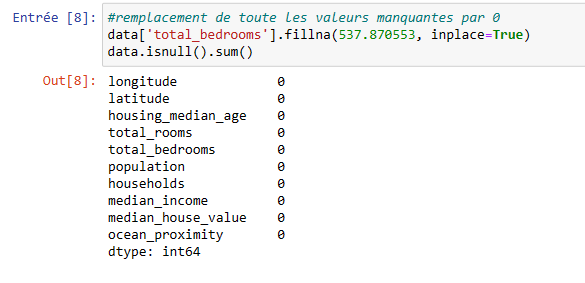
* DataFrame.isnull() : permet de d’afficher par valeur de vrai ou faux toutes les données du dataframe qui sont non définit.

Dans notre cas nous avons afficher le nombre de valeur nulle totale que contient chaque variable.



* DataFrame.fillna(valeur\_par\_defaut) : permet de remplacer les valeurs nulles de notre DataFrame par la valeur par défaut passer en paramètre.

Dans notre cas nous avons remplacé les valeurs manquantes par la moyenne des valeurs de la variable concerner.



* DataFrame.dropna(axis=0) : permet de supprimer les enregistrments null dans notre dataframe
* La transformation et la préparation des données :

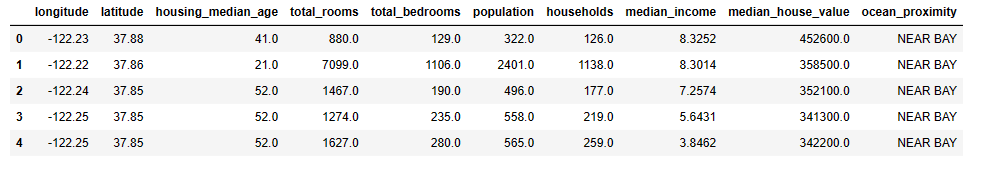
Dans cette phase, on met les données dans un format propice à notre développement et on optimise notre modèle d’apprentissage.

Lorsque les données ne sont pas au bon format,le format propice est le format uft-8 cet encodage ce fait par l’attribut encoding = ‘’ ISO- ‘’

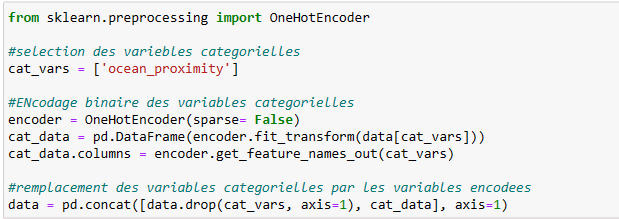
Pour les données qualitatives, une transformation en valeur numérique est impérative et cela se fait par le processus d’encodage. Dans notre cas nous avons une variable qualitative nous allons l’encoder grâce à la méthode encoder.get\_feature\_names\_out(variable).

Dans notre cas nous avons transformer notre variable qualitative ‘ocean\_proximity’ en valeur numérique. Puis fusionner avec notre jeu de donner.

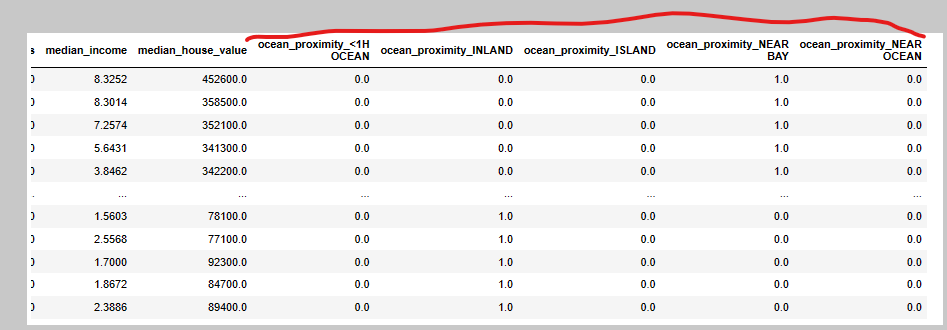
Avant



Code python



Apres encodage



## Choix des variables d’entrée et de sortie pour la construction de l’arbre de décision

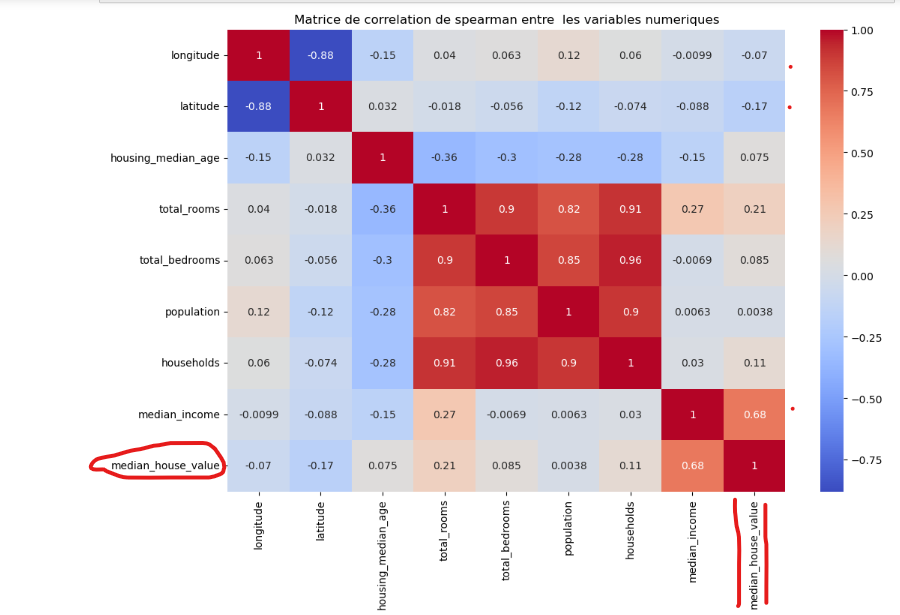
Pour construire un arbre de décision, il est important de choisir les variables d’entrée et de sortie appropriées. Les variables d’entrée sont les variables qui sont utilisées pour prédire la variable de sortie. Les variables de sortie sont les variables que l’on cherche à prédire. Il est important de noter que le choix des variables d’entrée et de sortie peut avoir un impact significatif sur la précision de l’arbre de décision. Il est donc important de choisir les variables avec soin et de les tester sur un ensemble de données de test pour évaluer leur précision.

Il existe plusieurs méthodes pour sélectionner les variables d’entrée et de sortie pour la construction d’un arbre de décision. Voici quelques-unes des méthodes les plus courantes:

* **Méthode de la corrélation**: Cette méthode consiste à sélectionner les variables qui ont une forte corrélation avec la variable de sortie.
* **Méthode de l’analyse en composantes principales (ACP)**: Cette méthode consiste à sélectionner les variables qui expliquent la plus grande partie de la variance dans les données.
* **Méthode de la régression**: Cette méthode consiste à utiliser une régression pour prédire la variable de sortie à partir des variables d’entrée et à sélectionner les variables qui ont une forte corrélation avec la variable de sortie.
* **Méthode de la validation croisée**: Cette méthode consiste à diviser les données en ensembles d’apprentissage et de test et à sélectionner les variables qui ont une forte corrélation avec la variable de sortie dans l’ensemble de test.

**Dans notre cas nous voulons prédire le prix ou la valeur d’une maison donc nous allons opter pour la méthode de corrélation pour pouvoir évaluer la corrélation qui existe entre nos variables et ‘median\_house\_value’ afin d’avoir nos variables d’entrer.**

**Pour ce fait nous avons tracer la matrice de corrélation de spearman grâce à la méthode corr(method=’spearman’) sur des variables numériques on obtient la figure ci-dessous :**



Apres analyse nous optons pour le choix des variables d’entrée numérique les variables suivantes :

* Median\_icome
* Longitute
* Latitude
* Housing\_median\_age

Rappelons que notre variable cible est ‘ median\_house\_value’.

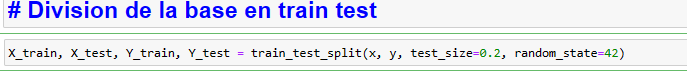
# Construction de L’arbre de décision

## Choix des critères de division et des mesures d’évaluations

Lors de la construction d’un arbre de décision, il est important de choisir les critères de division et les mesures d’évaluation appropriés pour obtenir un modèle précis et généralisable.

Cela se fait grace train\_test\_split. train\_test\_split est une fonction de la bibliothèque scikit-learn qui permet de diviser un ensemble de données en deux sous-ensembles aléatoires: un ensemble d’entraînement et un ensemble de test. Cela permet de mesurer les performances d’un modèle sur des données qu’il n’a jamais vues auparavant.

Utilisation de cette fonction dans notre cas :



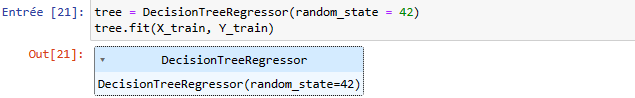
Dans ce cas, X et y sont les ensembles de données à diviser, test\_size est le pourcentage de données à utiliser pour l’ensemble de test et random\_state est une graine aléatoire pour garantir que la division est reproductible. train\_test\_split renvoie quatre tableaux: X\_train, X\_test, y\_train et y\_test, qui contiennent respectivement les données d’entraînement et de test pour les caractéristiques et les étiquettes.

## Construction de l’arbre

Comme Nous l’avons dit plus haut nous allons construire un arbre de décision.

Pour cela nous allons utiliser le model DecisionTreeRegressor. DecisionTreeRegressor est une classe de la bibliothèque scikit-learn qui implémente un arbre de décision pour la régression. Il s’agit d’un algorithme d’apprentissage supervisé qui construit un arbre de décision à partir des données d’entraînement pour prédire une variable cible continue.

Puis de la fonction fit qui prend deux paramètre X et y qui sont des données d’entraînement.  La fonction fit est utilisée pour entraîner un modèle d’apprentissage automatique sur un ensemble de données d’entraînement. Elle ajuste les paramètres du modèle pour minimiser l’erreur de prédiction sur les données d’entraînement.



# Evaluation et validation de l’arbre de décision

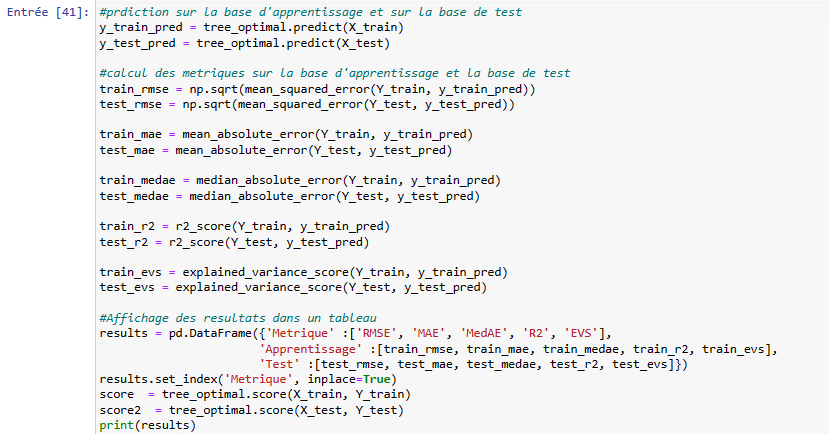
L’évaluation et la validation d’un arbre de décision sont des étapes cruciales dans le processus de construction d’un modèle de Machine Learning. L’évaluation permet de mesurer les performances du modèle sur un ensemble de données de test, tandis que la validation permet de s’assurer que le modèle est généralisable à de nouvelles données.

## Mesure de la performance de l’arbre

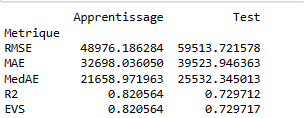
Pour évaluer notre arbre de décision, nous avons utilisé des métriques telles que :

* La RMSE (Root Mean Square Error) est une métrique qui mesure la distance moyenne entre les valeurs prédites et les valeurs réelles d’un modèle de régression ;
* la MAE (Mean Absolute Error) est une mesure d’erreur qui calcule la moyenne des valeurs absolues des différences entre les valeurs prédites et les valeurs réelles
* Le coefficient de détermination R2, également appelé R-squared, est une métrique de régression qui mesure la proportion de la variance totale de la variable dépendante expliquée par le modèle de régression [1](https://kobia.fr/regression-metrics-r2-score/). Il est compris entre 0 et 1, où 0 indique que le modèle ne fournit aucune information sur la variable dépendante, tandis que 1 indique que le modèle explique parfaitement la variance de la variable dépendante
* explained\_variance\_score est une métrique de régression qui mesure la proportion de la variance totale de la variable dépendante expliquée par le modèle de régression

le code python correspondant est :



Le résultat :



## Test pour évaluer la généralisation de l’arbre

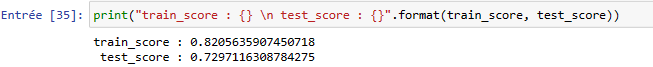
Pour tester la généralisation de notre arbre de décision, nous avons diviser notre ensemble de donnée en un ensemble d’apprentissage(train\_set) et en un ensemble de test(test\_set) puis évaluer l’arbre sur ces deux ensembles.

L’évaluation c’est fait par la détermination du score de l’arbre sur les données d’apprentissage et sur les données de test.

Code python :



Résultat :

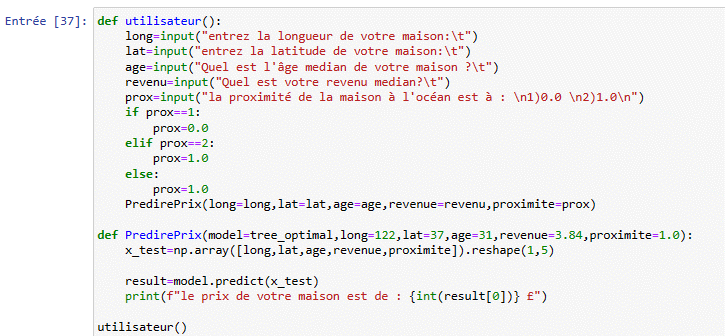


# Application Pratique et recommandation

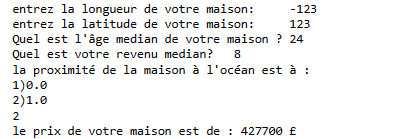
## Prise de décision commercial A partir de l’arbre de décision

Pour la prédiction, ici nous avons implémenter une procédure dont le rôle est de prédire le prix d’une maison par le biais de certaines valeurs caractéristiques de la maison passer en paramètre.

Code python :



Exécution :



## Recommandation

La recommandation se fait en fonction des utilisateurs et selon les besoins de chaque utilisateur. Grâce aux valeurs prédire, vous pouvez évaluer les différentes alternatives et choisir la meilleure option en fonction des critères et des objectifs spécifiques.

# Conclusion

EN conclusion, nous avons réalisé un model d’arbre de décision avec une profondeur de 10 avec un pourcentage de 72% de prédiction du prix de vente des maisons grâce aux méthodes statistiques et la construction du model d’arbre de décision.

# Bibliographie et hébergement

Bibliographie

* Google.com
* Wikipedia.com
* Youtube.com
* chatGpts .

Hébergé sur GitHub

* <https://github.com/yann214/Analyse_de_donnee>