|  |
| --- |
| [Tapez le nom de la société] |
| CAS PRATIQUE 1 RAPPORT E2 |
| Prédiction de la valeur médiane d’un quartier de logements californiens |
|  |
| **Marianne DEPIERRE** |
| **20/02/2023** |

|  |
| --- |
| Titre professionnel RNCP "Développeur en intelligence artificielle" |

Table des matières

[1. Introduction - 2 -](#_Toc123741718)

[1.1. Contexte - 2 -](#_Toc123741719)

[1.2. Analyse du besoin et enjeux - 2 -](#_Toc123741720)

[1.3. Organisation de travail - 2 -](#_Toc123741721)

[2. Description de l’existant - 2 -](#_Toc123741722)

[2.1. Jeu de données - 2 -](#_Toc123741723)

[2.1.1. Variable cible ou « target » - 3 -](#_Toc123741724)

[2.1.2. Variables explicatives - 3 -](#_Toc123741725)

[2.2. Application - 4 -](#_Toc123741726)

[2.3. Modèle - 4 -](#_Toc123741727)

[3. Améliorations - 4 -](#_Toc123741728)

[3.1. Amélioration du modèle - 5 -](#_Toc123741729)

[3.2. Amélioration de l’application - 5 -](#_Toc123741730)

[4. Vérifications de la non régression - 5 -](#_Toc123741731)

[4.1. MLFlow - 5 -](#_Toc123741732)

[4.2. Tests unitaires - 5 -](#_Toc123741733)

[4.3. Tests fonctionnels - 5 -](#_Toc123741734)

[5. Conclusion - 6 -](#_Toc123741735)

[6. Annexes - 6 -](#_Toc123741736)

# Introduction

Ce rapport décrit un projet de machine learning dont l’objectif principal est de valider les compétences pour le titre professionnel RNCP « Développeur en Intelligence Artificielle ». Dans ce but, un modèle et son application créés en cours de formation sont repris afin d’être améliorés. Ce projet utilisait le jeu de données « California Housing Prices » contenant des informations provenant du recensement californien de 1990. Les données concernent les logements appartenant à un quartier californien (ou « bloc » en anglais).

## Contexte

La startup Predimmo fournit des services pour les entreprises du domaine de l’investissement immobilier. Il y a plusieurs années, les chargés de relation client de cette startup ont relevé une augmentation de la demande dans la Silicon Valley. Il devenait difficile de répondre rapidement à toutes les demandes d’expertise pour un investissement immobilier dans cette région du monde. L’entreprise souhaitant automatiser cette tâche, elle a chargée un prestataire en intelligence artificielle de leur fournir un modèle prédictif.

Aujourd’hui, les chargés de mission font de nouvelles constatations. Les prédictions du modèle semblent en effet de moins en moins cohérentes avec les valeurs du marché. De plus, l’interface est jugée sobre et aucun élément ne permet à un nouvel utilisateur de se représenter les valeurs possibles d’un quartier californien. Predimmo a donc chargé un prestataire d’améliorer ces aspects de l’application et du modèle.

## Enjeu et objectifs

Un modèle prédictif et une application ont été développés précédemment afin de répondre à d’estimation de la valeur d’un quartier de logements en Californie. En renseignant diverses informations dans un formulaire, l’utilisateur obtient une estimation du prix. Une révision du modèle et une amélioration graphique est souhaitée par le client. Ainsi plusieurs objectifs ont été établis avec le client :

* Evaluation des performances du modèle existant
* Amélioration des performances de l’ancien modèle
* Amélioration graphique de l’application

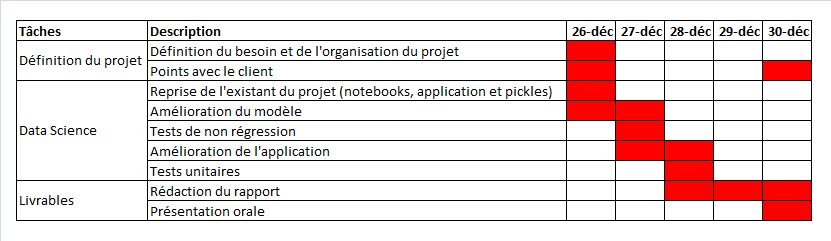
Pour répondre aux besoins du client, un environnement de développement a été définit comme suit.

|  |  |  |
| --- | --- | --- |
| Langage | Librairies python | Outils |
| Python | Application : streamlit | Versioning : Git et GitHub |
|  | Modèle et traitement de données : scikit-learn, pickle |  |
|  | Utilitaires : os, pandas, numpy |  |
|  | Versioning : mlflow |  |
|  | Tests unitaires unittest |  |
|  | Environnement virtuel : virtualenv |  |

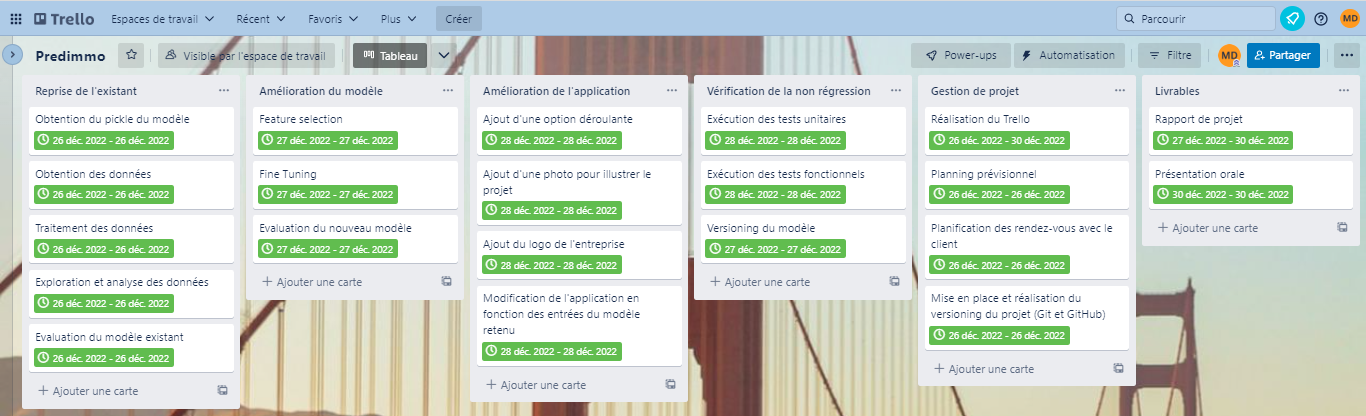
Le langage python, ses librairies et les outils mentionnés permettent de lire et évaluer les performances de l’ancien modèle, d’optimiser celles-ci tout en vérifiant la non régression du modèle, ainsi que d’améliorer l’interface de l’application.

## Organisation de travail

Dans un premier temps, un planning prévisionnel a été défini, montrant les dates et le temps alloué pour chacune des tâches identifiées.



Un planning réel a été effectué avec l’outil Trello. Il s’agit d’un outil de gestion de projet en ligne qui organise des projets avec des cartes représentant des tâches.



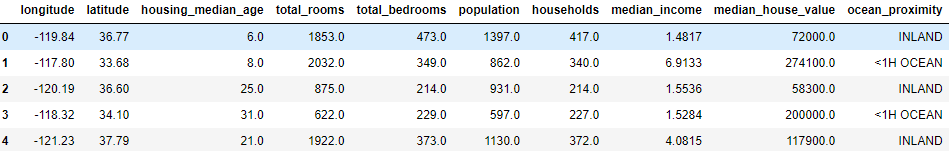
Globalement, le planning prévisionnel a été respecté. En effet, cinq jours étaient prévus pour la réalisation de ce projet et cinq jours y ont été consacrés. Les tâches ont été effectuées légèrement différemment. En effet, le rapport a été rédigé en parallèle de l’exécution des autres tâches, et l’amélioration du modèle et de l’application ont débuté le jour suivant. Des points avec le client ont été organisés et effectués en début et en fin de session de travail, pour le rendu des livrables.

Le versioning du projet s’est fait avec Git et GitHub (capture d’écran du repository en annexes)

# Description de l’existant

## Jeu de données

Le jeu de données est sous forme de fichier CSV.



Il est composé de 16 336 lignes (ou observations) et 10 colonnes, soit une variable cible (ou « target ») et 9 variables explicatives (ou « features »). La tâche à réaliser est de type supervisée, plus précisément il s’agit d’un problème de régression linéaire.

### Variable cible ou « target »

La variable cible est la valeur médiane des logements pour les ménages d’un quartier, en dollars américain. Elle est nommée median\_house\_value dans le jeu de données. Il s’agit d’une variable numérique continue. Ci-dessous se trouvent un histogramme et des informations (pd.DataFrame.describe()) sur sa distribution.

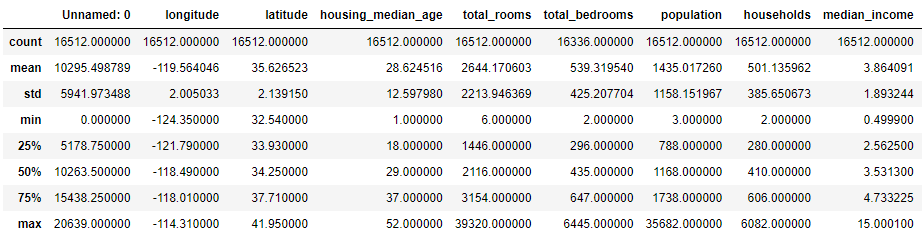
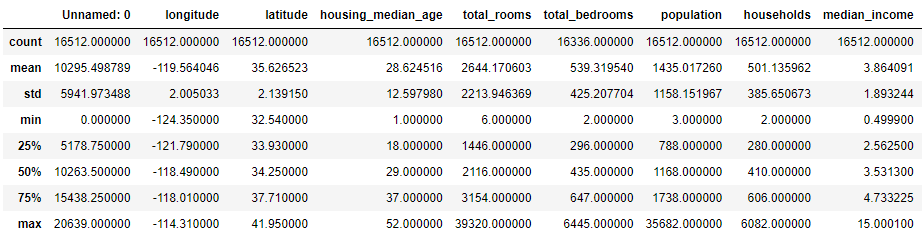
|  |  |
| --- | --- |
| distribution_median_house_value |  |
| *Histogramme de la distribution de la variable cible* | *Informations sur la distribution de la variable cible* |

### Variables explicatives

Le tableau suivant récapitule les informations pour chacune des variables. Il résume aussi les informations accessibles grâces à la fonction pd.DataFrame.info() appliquées au dataframe du jeu de données.

|  |  |  |
| --- | --- | --- |
| Variable | Description | Type |
| longitude | Distance à l’ouest d’une maison. | float |
| latitude | Distance nord d’une maison. | float |
| house\_median\_age | Age médian d’une maison dans un quartier. | float |
| total\_rooms | Nombre total de pièces dans un quartier. | float |
| total\_bedrooms | Nombre total de chambres dans un quartier. | float |
| population | Nombre total de résidents dans un quartier. | float |
| households | Nombre total de ménages (ou groupe de personnes résidant dans un logement) | float |
| median\_income | Revenus médian des ménages dans un quartier (en dizaines de milliers de dollars américains) | float |
| ocean\_proximity | Distance à l’océan. Cinq catégories : « INLAND », « NEAR BAY », « NEAR OCEAN », « <1H OCEAN » et « ISLAND » | object |

Ci-dessous, il s’agit du résultat de la fonction pd.DataFrame.describe() appliquée au dataframe du jeu de données.



## Application

L’application déjà en place utilise streamlit, une bibliothèque python qui permet de développer des applications à partir d’un script. En entrant des valeurs pour chacune des variables nécessaires à la prédiction, cette application estime le prix médian d’un quartier de logements en Californie.

## Modèle

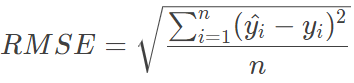
Le modèle ainsi que le traitement des données existants sont récupérés sous formes de pickles. Les pickles en python sont utilisés pour sérialiser et dé-séréaliser une structure objet python. En résumé, un objet python est convertit en flux d’octets et est stocké dans un fichier, le pickle. Ce fichier contient toutes les informations pour reconstruire un objet python.

Le modèle existant est un modèle de « k-nearest neighbors » ou « KNN ». Les paramètres utilisés pour ce modèle sont ceux par défaut. Avant d’entraîner le modèles, les données ont été normalisées avec une fonction MinMaxScaler() et les données qualitatives ont été converties en valeurs numériques avec une fonction OneHotEncoder(). Les données manquantes ont été supprimées du jeu de données. Il n’y avait pas de doublons.

Dans un notebook Jupyter Notebook, les pickles sont exécutés sur le nouveau jeu de données et les scores du modèle existant sont récupérés.

|  |  |
| --- | --- |
| Score | Valeur |
| R² | 0.6896264484170945 |
| RMSE | 63898.09826508343 |

* Le R² ou coefficient de détermination quantifie la corrélation dune ou des variables explicatives avec une variable cible. Il représente donc la proportion de la variation totale des données qui est expliquée par le modèle de régression. Ce coefficient peut être compris entre 0 et 1 et plus il est élevé, plus le modèle est considéré comme précis. Dans notre cas, il est égal à environ 0.69 ; cette valeur est un peu faible.
* Le RMSE (Root Mean Square Error) mesure la différence moyenne entre les valeurs prédites par le modèle et les valeurs réelles.



Plus le RMSE est faible, meilleure est la performance du modèle. Le RMSE du modèle de base est d’environ 63 898. En d’autres terme, lorsque le modèle prédit une valeur,

Si l’on compare à la moyenne (206 422 environ) et l’écart-type (115 264 environ) de la variable cible, le RMSE paraît trop important.

# Améliorations

## Amélioration du modèle

A la suite d’une feature importance les variables « ISLAND » et « <1H OCEAN » ont été supprimées.

(Fine Tuning definition)

Afin d’améliorer les performances du modèle, un GridSearchCV a été appliqué avec plusieurs grilles de paramètres. Celle retenue est la suivante :

XXX

Un nouvel entraînement du modèle a été exécuté avec les paramètres suivant : XXX. Les scores du nouveau modèle sont ceux affichés dans le tableau ci-dessous.

|  |  |
| --- | --- |
| Score | Valeur |
| R² |  |
| RMSE |  |

## Amélioration de l’application

Basiquement, l’application est très sobre et ne montre que les champs à remplir pour la prédiction. Une image a donc été ajoutée. Un menu déroulant à été ajouté et permet, selon le désir de l’utilisateur, d’afficher en un clic des informations sur la variable à prédire et sur le modèle de prédiction utilisé.

# Vérifications de la non régression

## Versioning avec MLflow

Du versioning a été réalisée avec la librairie MLflow. Le modèle est suivi pour vérifier qu’il n’y a pas de baisse de performance au cours de son cycle de vie, notamment après la modification de ses paramètres ou après de nouvelles alimentations du jeu de données.

## Tests unitaires

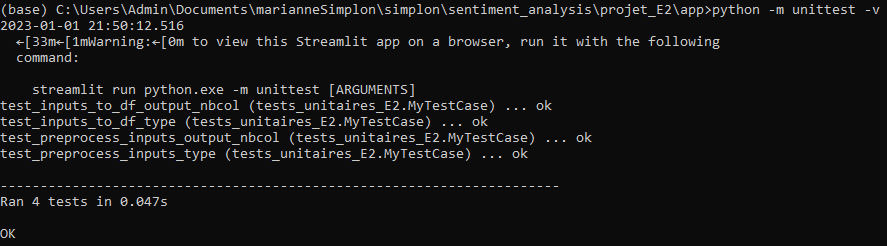
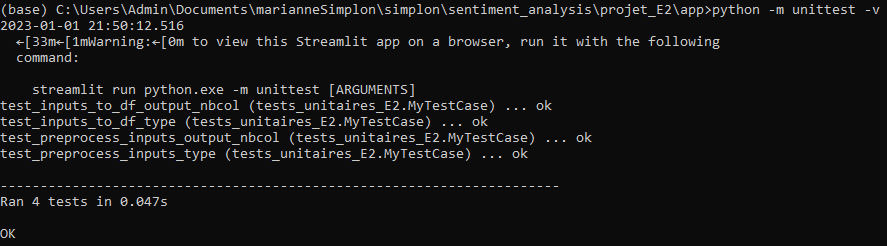
Pour s’assurer que l’application n’avait pas régressée, les tests unitaires codés précédemment sont exécutés. Ces tests unitaires sont codés avec la librairie unittest et vérifient que la conformité :

* Du nombre de colonnes après collecte des entrées soumises par le formulaire
* Du type de la variable retournée après collecte des entrées soumises par le formulaire
* Du nombre de colonnes après traitement des données
* Du type de la variable retournée après traitement des données

Ci-dessous un extrait du code des tests unitaires réalisés.

XXXX

Après exécution du fichier (Annexe XXX), les tests on été passés avec succès.



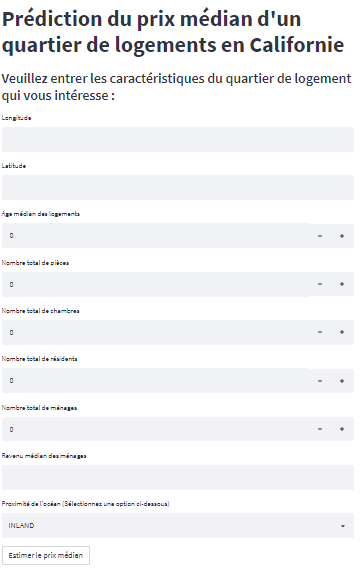
## Tests fonctionnels

Des tests fonctionnels de l’application ont été réalisés sous forme de jeu d’essais.

|  |  |  |  |
| --- | --- | --- | --- |
| Cas d’essai | Résultat attendu | Résultat obtenu | Commentaires |
| Visibilité de l’image d’illustration | L’image s’affiche au chargement de la page, quelle que soit la taille de la fenêtre. | Conforme |  |
| Affichage du menu déroulant | Le menu déroulant se déroule au clic de l’utilisateur | Conforme |  |
| Affichage du graphique de la distribution de la variable cible | Le graphique est affiché lorsque le menu est déroulé. | Conforme |  |
| Visibilité du formulaire | Le formulaire est visible quelle que soit la taille de la fenêtre | Conforme |  |
| Affichage du formulaire | Le formulaire affiche les champs suivants :  Longitude, Latitude, Age médian des logements, Nombre total de pièces, Nombre total de chambres, Nombre total de ménages, Revenu médian des ménages et Proximité de l’océan.  Un bouton « Estimer le prix médian » permet de valider le formulaire. | Conforme |  |
| Réussite de la prédiction | La prédiction est affichée en dessous du formulaire après soumission du formulaire | Conforme |  |
| Echec de la prédiction | Un message d’erreur est affiché à l’utilisateur si la prédiction a échouée | Conforme | Prévoir un message personnalisé selon le type d’erreur améliorait l’expérience utilisateur. |

# Bilan

# Annexes



Capture de l’application existante



Extrait du script python réalisant les tests unitaires



