Cars Price

Développeur Data / IA
2019-2020

SIMPLON.CO

SOMMAIRE

1 - Introduction du Projet

L’objectif de ce projet est de prédire le prix selon plusieurs caractéristiques (années, modèle...) dans l’optique d’obtenir une cotation concernant les voitures d’occasions.

Les besoins sont:

- La création d’une base de données
- Un modèle d’intelligence artificielle
- Une API du modèle pour l’intégrer à d’autres services
- Une interface web affichant les résultats obtenus

2 - Analyse de la demande

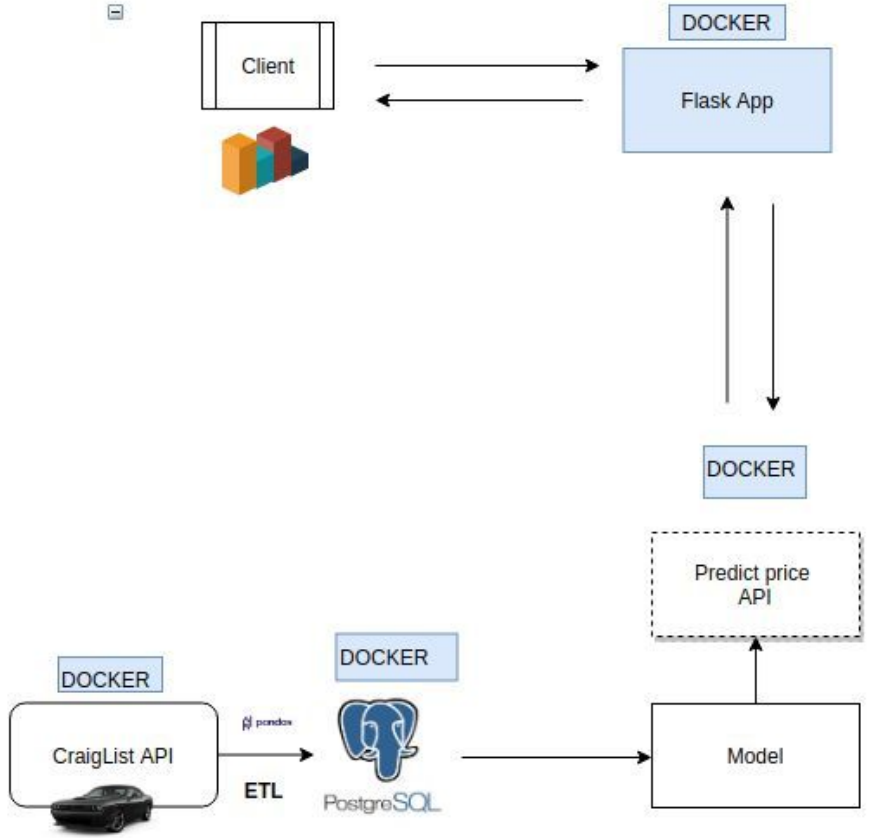
Le projet ne doit pas contenir d'informations personnelles dans notre base de données comme plaque d'immatriculation.

Les utilisateur: des particuliers ou des entreprises

Planning:

- Création de l'architecture du projet, de la base de données
- Alimentation de la base de données
- Création du modèle
- Création d'une API
- Création d'une APP

Architecture du projet



3 - Conception de la structure des données

Base de données relationnelle: Postgresql

Shema: voir repo

Utilisation de Docker + volume

4 - Collecte des données

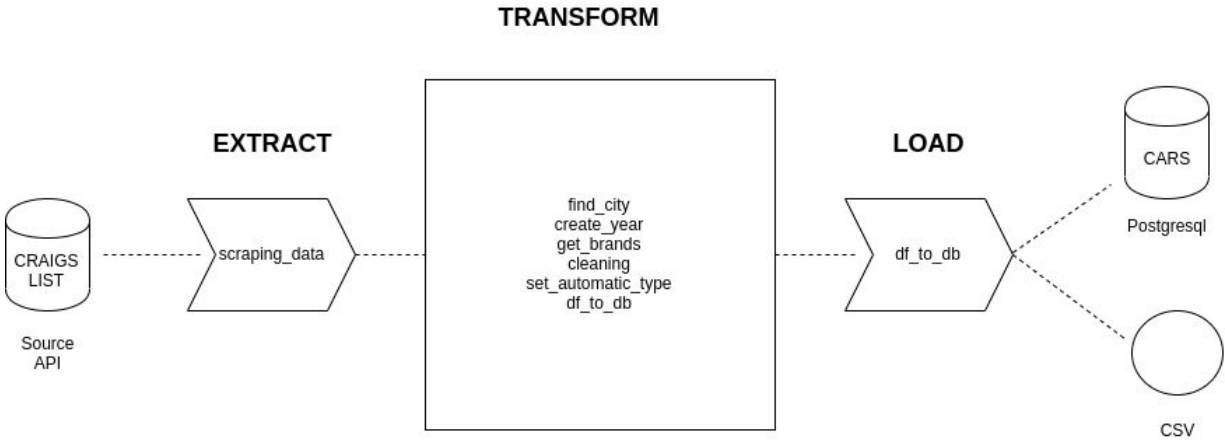
Sources: API Craiglists

- `CraigslistCommunity` (craigslist.org > community)
- `CraigslistHousing` (craigslist.org > housing)
- `CraigslistJobs` (craigslist.org > jobs)
- **`CraigslistForSale` (craigslist.org > for sale)**
- `CraigslistEvents` (craigslist.org > event calendar)
- `CraigslistServices` (craigslist.org > services)
- `CraigslistGigs` (craigslist.org > gigs)
- `CraigslistResumes` (craigslist.org > resumes)

5 - Pipeline de données

ETL (pandas Pipe):

- Extract (Api Source format json)
- Transform (reverse_geocode, feature engineering, cleaning, set_automatic_type)
- Load (csv + Postgresql)



6 - L’extraction

Format Json


```
[{'id': '7233831584',
  'repost_of': None,
  'name': 'Smart 2 seater',
  'url': 'https://sfbay.craigslist.org/eby/cto/d/oakland-smart-seater/7233831584.html',
  'datetime': '2020-11-19 14:10',
  'last_updated': '2020-11-19 14:10',
  'price': '$3,000',
  'where': 'alameda',
  'has_image': True,
  'geotag': (37.8471, -122.2223),
  'deleted': False,
  'fuel': 'gas',
  'color': 'black',
  'cylinder': '3 cylinders'}]
```


7 - Traitement - Création de features + cleaning

Création d'une colonne city

```
def find_city(df):
    def reverseGeocode(coordinates):
        import reverse_geocoder as rg
        try:
            result = rg.search(coordinates)
            return result[0]['name']
        except: #NoneType
            return "NaN"
    df['city'] = df['geotag'].apply(lambda x: reverseGeocode(x))
    return df
```

Regex: Création d'une colonne year et brands

Cleaning: Supprimer les $, transformer "6 cylinder" en 6

Traitement: set_automatic_type

```
def set_automatic_type(df):
    tests = [
            (int, int),
            (float, float),
            (datetime, lambda value: datetime.strptime(value, "%Y-%m-%d %H:%S"))
        ]
    def getType(value):
            #Get type of data to create database
             for typ, test in tests:
                 try:
                     test(value)
                     return typ
                 except ValueError:
                     continue
             # No match
             return "str"

    for r,c in zip(df.iloc[0],df.columns):
      try:
        get_type = getType(r)
        if get_type == int:
          df[c] = pd.to_numeric(df[c])
        elif get_type == float:
          df[c] = df[c].astype("float8")
        elif get_type == datetime:
          df[c] = pd.to_datetime(df[c])
      except TypeError:
        pass
    return df
```

Aperçu de nos données

	id	name	url	price	geotag	fuel	color	cylinder	city	year	brand
1	7233279609	2009 Smart Fortwo	https://sfbay.craigslist.org/sby/cto/d/campbel...	5100	(37.273233, -121.937428)	gas	black	3	Cambrian Park	2009	smart
2	7232819826	2019 Mini Cooper Convertible, auto, nav, camer...	https://sfbay.craigslist.org/eby/ctd/d/san-ram...	19900	(37.78335, -121.98015)	gas	black	3	San Ramon	2019	seat
3	7231976837	2012 Smart ForTwo Passion Cabrio	https://sfbay.craigslist.org/eby/cto/d/hayward...	6400	(37.7015, -122.0782)	gas	black	3	Castro Valley	2012	smart
4	7228057298	2020 Hyundai Santa Fe SE 2.4 suv Twilight Black	https://sfbay.craigslist.org/sby/ctd/d/san-jos...	23900	(37.274813, -121.875391)	gas	black	3	Seven Trees	2020	hyundai
7	7221478254	2019 Mini Cooper Convertible, auto, nav, camer...	https://sfbay.craigslist.org/eby/ctd/d/san-ram...	19900	(37.78335, -121.98015)	gas	black	3	San Ramon	2019	seat

8 - Chargement dans PostgreSQL + csv

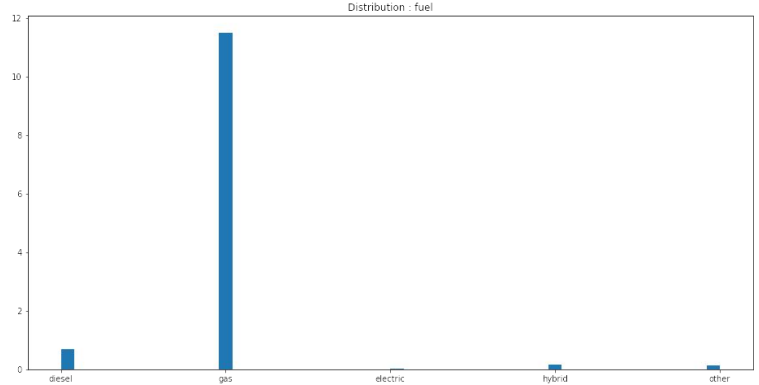
- Connexion à la base de données avec psycopg2
- Création d'une fonction df_to_db qui récupère notre dataframe puis le découpe en plusieurs dataframes correspondant à la structure de notre base de données.
- Chargement vers un csv

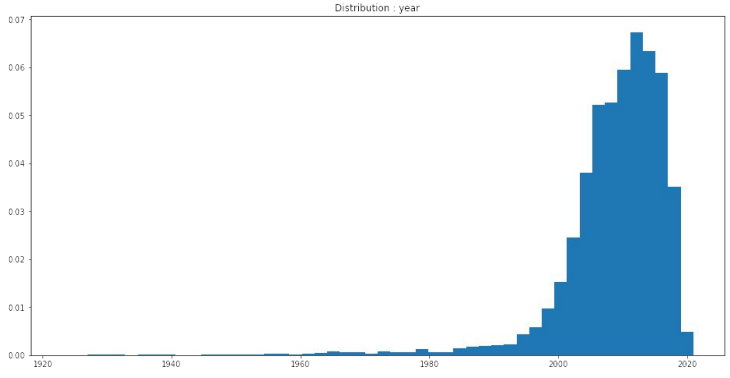
9 - Process pour l’entraînement du modèle

Dataset externe historique disposant de 85000 lignes.

- Analyse exploratoire
- Suppression outliers
- Label encoding
- Normalisation
- Split train test
- Comparaison de plusieurs modèles

10 - Analyse exploratoire + outliers





- Suppression des prix en dessous de 500$ et au-dessus de 60000$
- Conservation uniquement du carburant gas
- Conservation des véhicules datant entre 2000 et 2020

11 - Label encoding + normalisation

Label encoding des colonnes catégoriques

	price	year	state	manufacturer	model	condition	cylinders	fuel	odometer	transmission	drive	type	paint_color
5	13995	112	23	13	3614	2	5	0	188406	0	0	10	5
6	7995	110	23	7	3251	2	3	0	108124	0	0	0	5
7	8995	111	23	7	8350	2	5	0	178054	0	0	0	10
8	10995	114	23	13	3473	2	5	0	170259	0	0	0	10
10	10995	111	23	7	7310	2	6	0	210865	0	0	10	9

Normalisation: obtenir des données comprises entre l’intervalle 0 et 1

12 - Split train - test

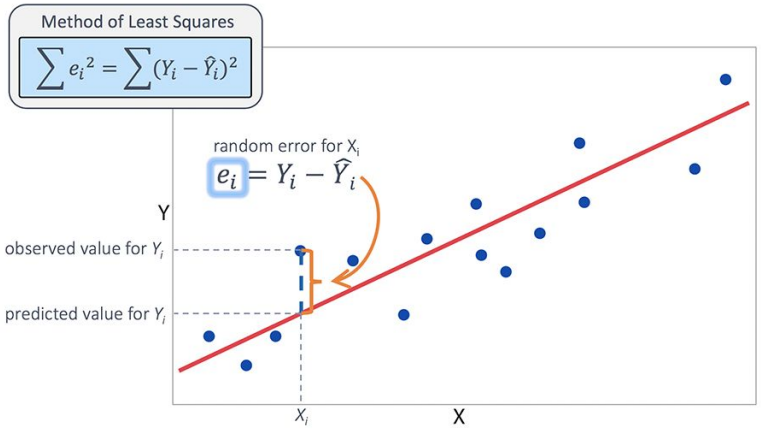
Librairie train_split de sklearn 20% pour le test

- X_train_shape (54089, 12)
- y_train_shape (54089,)
- X_test_shape(13523, 12)
- y_test_shape(13523,)

13 - Comparaison des modèles

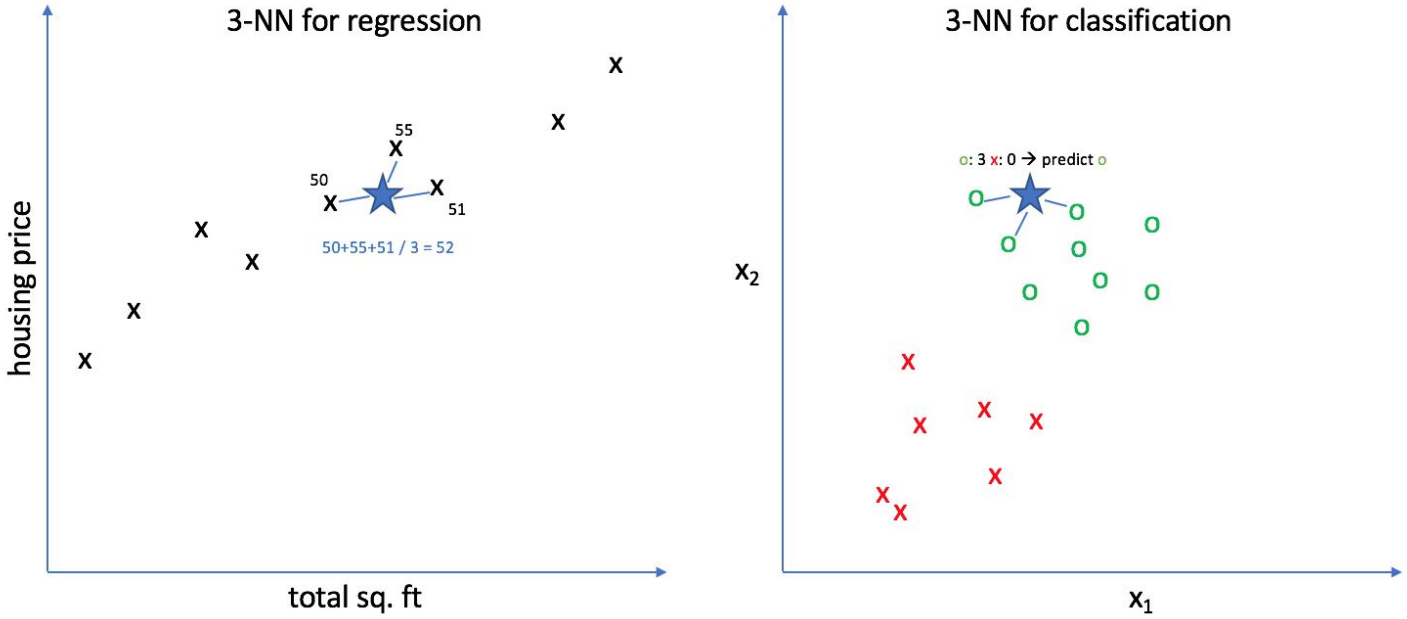
	RMSE TRAIN	RMSE TEST	r2 score TRAIN	r2 score TEST
Linear Regression	7700	8032	28	33
Ridge Regression	7710	7957	34	30
KNN + k param	7403	6257	56	34
XGBOOST	2745	3317	90	88

Linear regression, Ridge regression

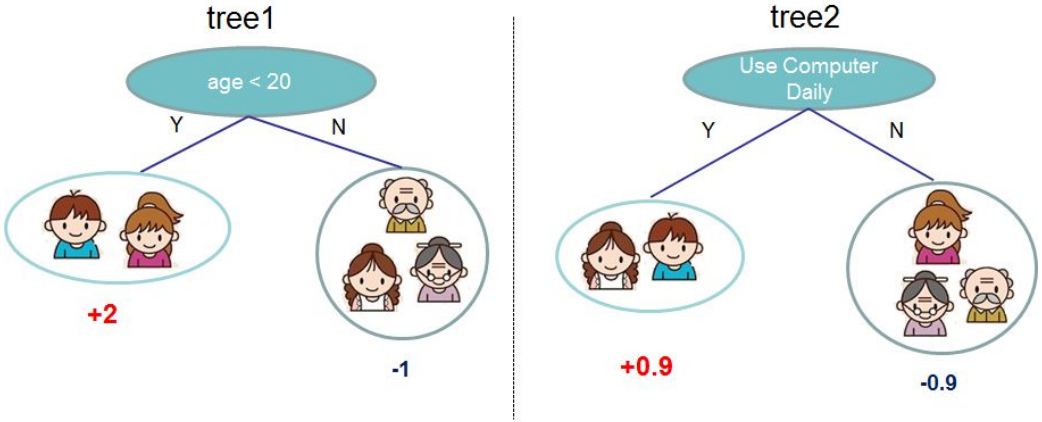


Ridge: Ajout d’une pénalité (moins overfitting)

KNN


3-NN for regression
housing price
total sq. ft
55
50
51
50+55+51 / 3 = 52
3-NN for classification
o: 3 x: 0 → predict o
x_2
x_1

XGBOOST



f( ) = 2 + 0.9= 2.9 f( )= -1 - 0.9= -1.9

14 - API Flask

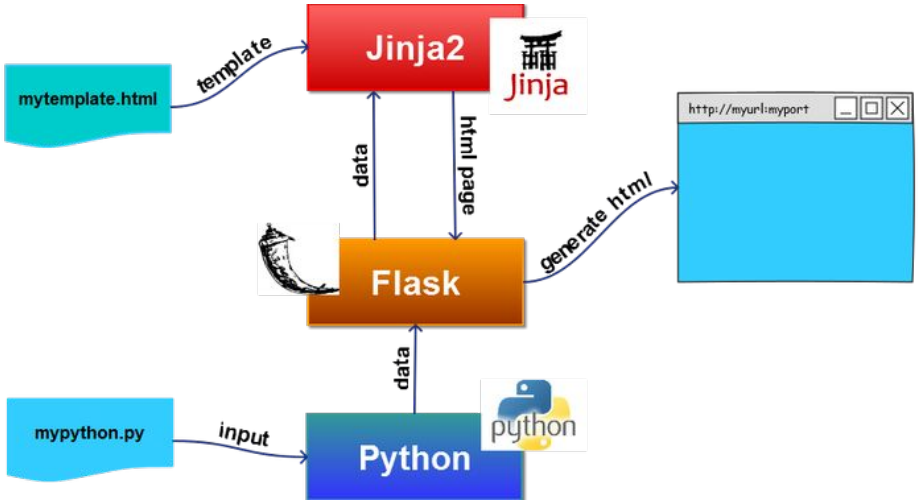
- Pourquoi utiliser une API?
- Utilisation librairie Flask_restful
- Chargement de notre modèle sous format pkl
- Création d’un container via un fichier Dockerfile

```
(base) zack@zack-ThinkPad-X1-Carbon-4th:~/Desktop/simplon/simplon_project_cars_nlp_sql$ curl -X GET http://127.0.0.1:5
001/ -d answer="[2010,'ny','nissan','rogue','fair','4 cylinders','gas', 731313.0 ,'automatic','4wd','SUV','black']"
"\"16702.451\""
```

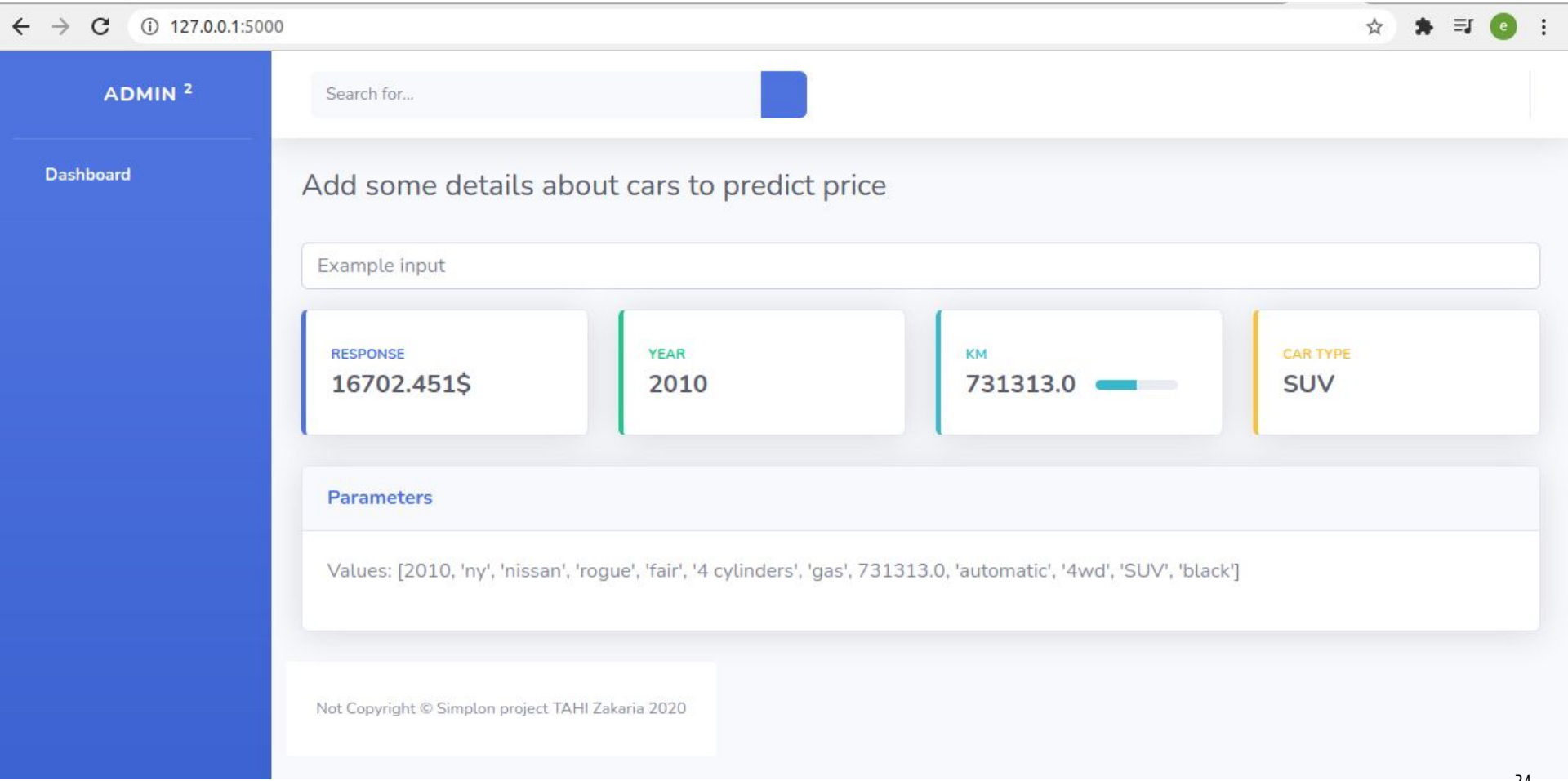
15 - L’application

Front: Framework boostrap (JS, html, css), jinja (python)

Back-end: Flask (Python)



Interface web



16 - Conclusion

Ce projet a permis d’explorer toutes les étapes d’un projet IA du besoin client jusqu’à la solution. La réalisation de ce MVP de bout en bout, notamment via l’utilisation de flask, docker et d’une base de données relationnelle a rendu le développement de la solution plus rapide.

17 - Améliorations

Traitement de la source de la Data (temps, mémoire).

Enrichir les données (source externes).

Amélioration de la robustesse du modèle:

- Plus de données
- Temporalité (décôte)