L’ia au service des agents immobiliers

Préparation au baseline model quick view

	Unnamed: 0	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
0	2072	-119.84	36.77	6.0	1853.0	473.0	1397.0	417.0	1.4817	72000.0

```
#first quick view for types
df_brut.dtypes
```

```
Unnamed: 0              int64
longitude             float64
latitude              float64
housing_median_age    float64
total_rooms           float64
total_bedrooms        float64
population            float64
households            float64
median_income         float64
median_house_value    float64
ocean_proximity        object
dtype: object
```

```
df_brut = df_brut.drop(columns=['Unnamed: 0'])
```

Préparation au baseline model missing data

```
#missing data
df_brut.isna().sum()
```

```
longitude             0
latitude              0
housing_median_age    0
total_rooms           0
total_bedrooms      176
population            0
households            0
median_income         0
median_house_value    0
ocean_proximity       0
dtype: int64
```

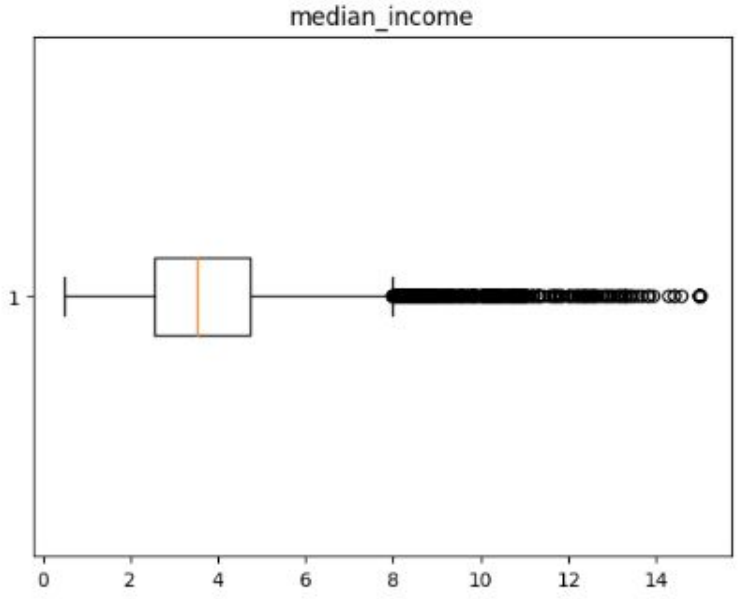
```
#Inputation of missing value by the median
median = df_brut["total_bedrooms"].median()
df_brut['total_bedrooms'].fillna(median, inplace=True)
```

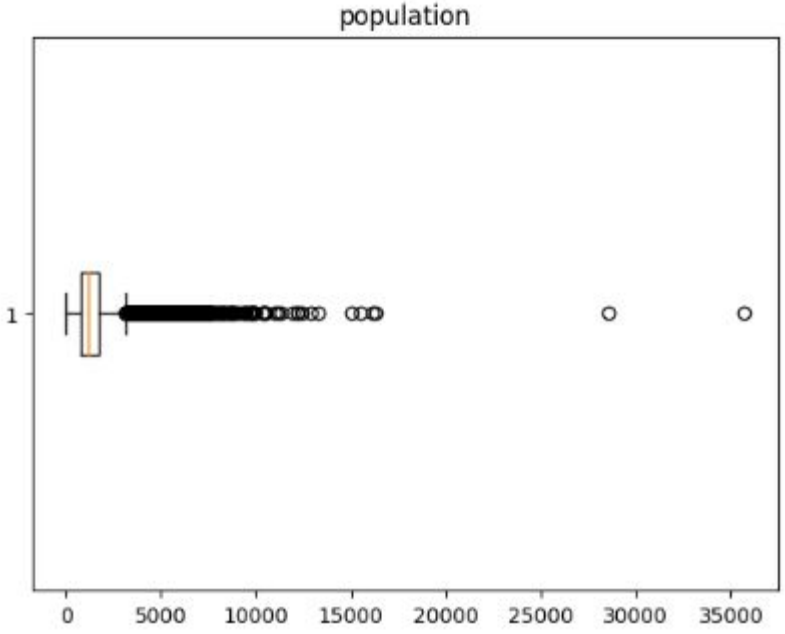
Baseline model

```
# Choose your feature and your target
X = df_clean[features]
y = df_clean['median_house_value']

# Splitting the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.2, random_state=3)

# Choose your model
model = LinearRegression()

# Fit the model with the train set
model.fit(X_train, y_train)

# Evaluate the model with the test set
baseline_score = model.score(X_test, y_test)
baseline_score
```

```
0.639149507560891
```

Itération outlier





Itération outlier

```
#The 2 location wich are real heavy outlier
df_clean[df_clean["population"] > 20000]
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
7471	-117.42	33.35	14.0	25135.0	4819.0	35682.0	4769.0	2.5729	134400.0	<1H OCEAN
14503	-121.79	36.64	11.0	32627.0	6445.0	28566.0	6082.0	2.3087	118800.0	<1H OCEAN

```
# both this location are military infrastructure , so it's not somthing relevent for house price , we can remove them

from IPython import display
display.Image("https://user-images.githubusercontent.com/104862908/212533897-cfaeeda6-9f09-48aa-a47f-7a6f40cb8dc7.PNG")
```

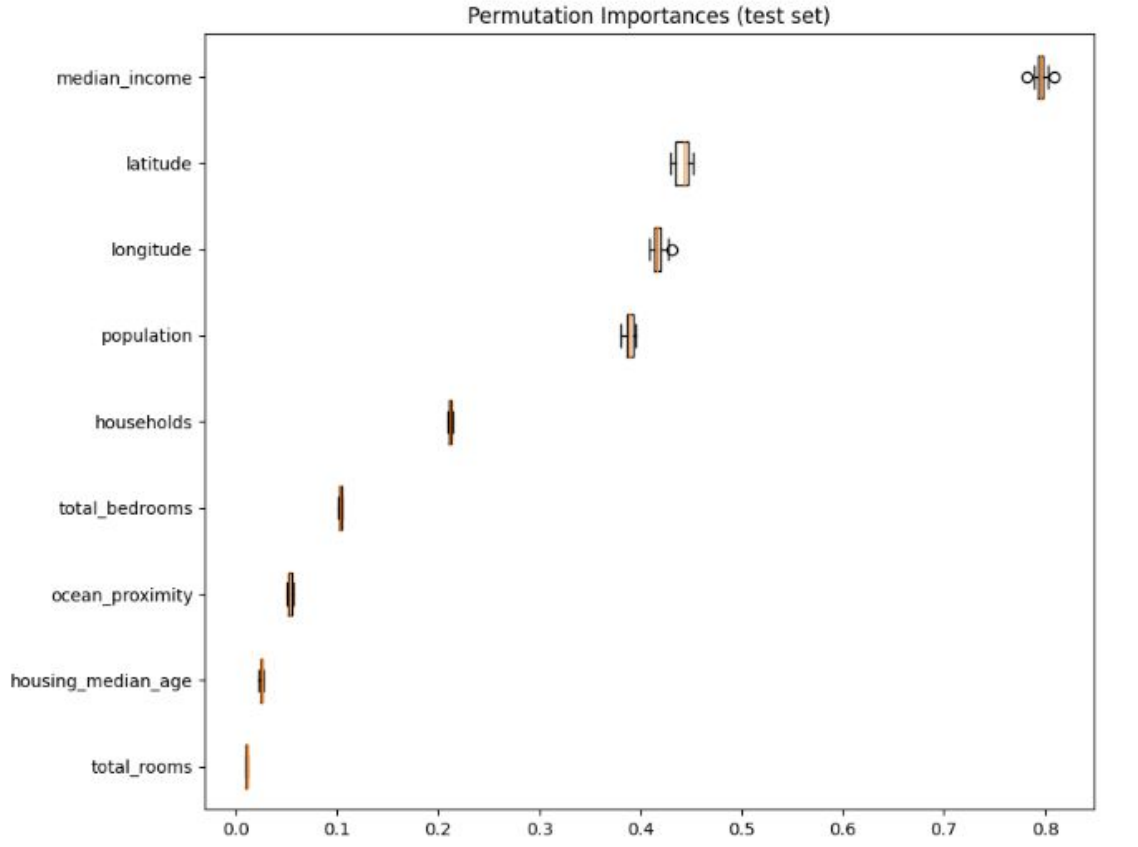
Itération OneHot Encoder

```
feature_cols = [
 'longitude',
 'latitude',
 'housing_median_age',
 'total_rooms',
 'total_bedrooms',
 'population',
 'households',
 'median_income',
 'ocean_proximity']


X = df_clean[feature_cols]
y = df_clean['median_house_value']
```

```
column_trans = make_column_transformer((OneHotEncoder(), ['ocean_proximity']),

                                       remainder='passthrough')
```

Itération Feature sélection

```
from sklearn.inspection import permutation_importance
```



Meilleur model

```
Average R2:  0.6494682364540078
```

```
Average RMSE:  68196.6689864473
```

Conclusion et ouverture

```
import pandas as pd
from pickle import *
```

```
fichier_pickle = open ("fichier_pickle","rb")
```

```
pipe = load(fichier_pickle)
```

```
pipe
```