OPENCLASSROOMS

CentraleSupélec | université PARIS-SACLAY

Soutenance du Projet 4

Anticipez les besoins en consommation électrique de bâtiments

Par
Elisée TCHANA

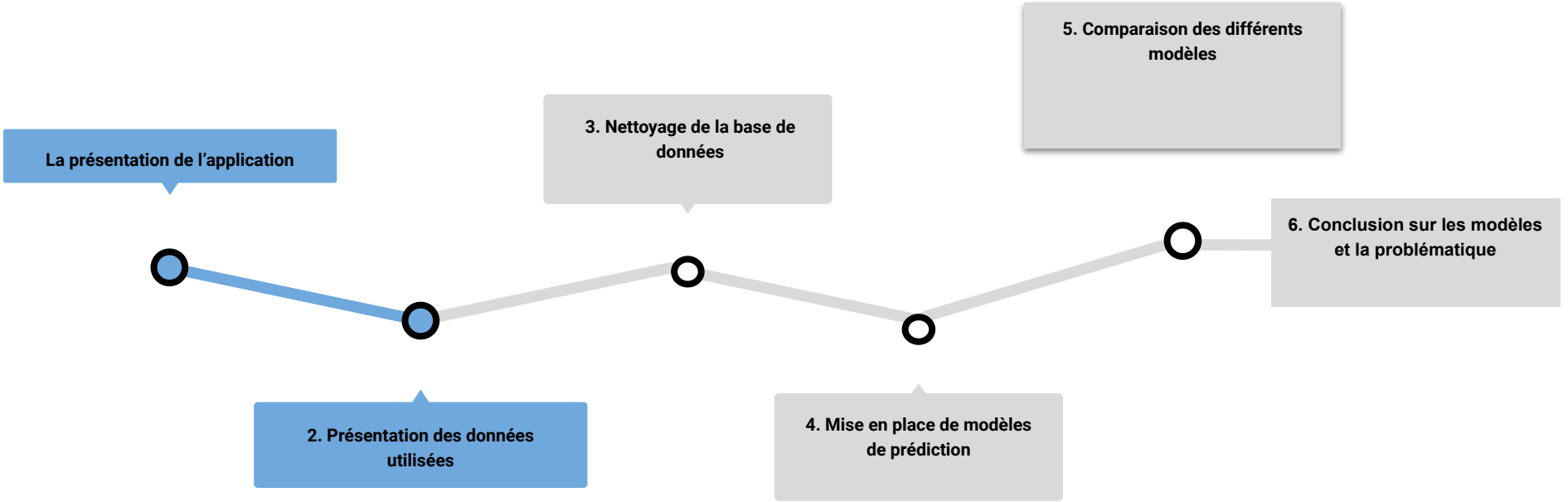
Mentor
Cyril MONTI

Avril 2021

OPENCLASSROOMS

CentraleSupélec | université PARIS-SACLAY

Plan de Soutenance



Problématique du Projet

1. Prédire les performances énergétiques de bâtiments situés à Seattle
2. Se passer des relevés faits sur place (opérations très coûteuses et fastidieuses)

Seattle

Objectifs

- Prédire la consommation en énergie des bâtiments
- Prédire les émissions en CO2 des bâtiments
- Evaluer l'intérêt de « l' ENERGYSTARScore »

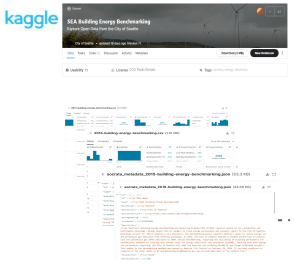
ENERGY STAR

1. Indicateur à l'échelle nationale permettant de refléter les performances énergétiques d'un bâtiment
2. Score allant de 1 (mauvaise performance) à 100 (excellente performance)
3. Un score de 50 représente la médiane national

OPENCLASSROOMS

CentraleSupélec | université PARIS-SACLAY

Data Analysis Workflow



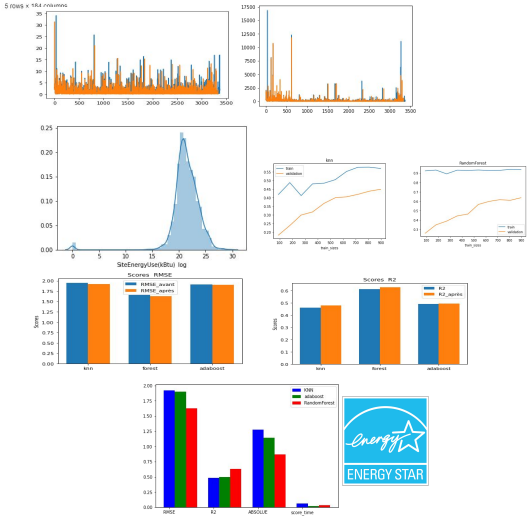
SEA Building Energy Benchmarking
Explore Open Data from the City of Seattle

```
#import packages
%matplotlib inline
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import math
import pickle
from sklearn.model_selection import train_test_split
from sklearn.impute import KNNImputer
from sklearn.preprocessing import StandardScaler
from scipy import spatial,stats

#import data
data= pd.read_csv("openfoodfacts.csv", sep='\t')
data.head()
```

	code	url	creator	created_t	created_datetime	last_modified_t	last_modified_datetime	product_name	abbrevia
0	0000000000017	http://world-en.openfoodfacts.org/product/00000...	kiliweb	1529056080	2018-06-15T19:38:00Z	1561463718	2019-06-25T11:55:18Z	Vitória crackers	
1	0000000000031	http://world-en.openfoodfacts.org/product/00000...	isagoofy	1539464774	2018-10-13T21:06:14Z	1539464817	2018-10-13T21:06:57Z	Cacao	
2	0000000000033327986	http://world-en.openfoodfacts.org/product/00000...	kiliweb	1574175736	2019-11-19T15:02:16Z	1574175737	2019-11-19T15:02:17Z	Filetes de pollo empanado	
3	0000000000100	http://world-en.openfoodfacts.org/product/00000...	del51	1444572561	2015-10-11T14:09:21Z	1444659212	2015-10-12T14:13:32Z	moutarde au moût de raisin	
4	0000000000111111111111	http://world-en.openfoodfacts.org/product/00000...	openfoodfacts-contributors	1560020173	2019-06-08T18:56:13Z	1560020173	2019-06-08T18:56:13Z	Sfudinn	

5 rows × 184 columns



Data Cleaning & Analysis

Data Sharing

Présentation du jeu de données

kaggle

Dataset

SEA Building Energy Benchmarking

Explore Open Data from the City of Seattle

City of Seattle • updated 18 days ago (Version 7)

Data Tasks Code (2) Discussion Activity Metadata

Download (3 MB) New Notebook

Usability 7.1 License CC0: Public Domain Tags socrata, energy, electricity

2015-building-energy-benchmarking.csv (1.51 MB)

2016-building-energy-benchmarking.csv (1.18 MB)

Detail Compact Column 10 of 46 columns

OSEBuildingID	DataYear	BuildingType	PrimaryPropertyT...	PropertyName
	2016	NonResidential 43%	Low-Rise Multifam... 29%	3362 unique values
		Multifamily LR (1-4) 30%	Mid-Rise Multifamily 17%	
		Other (898) 27%	Other (1825) 54%	
1	2016	NonResidential	Hotel	Mayflower park hotel

socrata_metadata_2015-building-energy-benchmarking.json (53.3 KB)

socrata_metadata_2016-building-energy-benchmarking.json (44.69 KB)


```
"root" : {
  "id" : string "2bpz-gwpy"
  "name" : string "2016 Building Energy Benchmarking"
  "assetType" : string "dataset"
  "attribution" : string "City of Seattle"
  "attributionLink" : string "https://www.seattle.gov/energybenchmarking"
  "averageRating" : int 0
  "createdAt" : int 1521070051
  "description" :
string "Seattle's Building Energy Benchmarking and Reporting Program (SMC 22.920) requires owners of non-residential and multifamily buildings (20,000 square feet or larger) to track energy performance and annually report to the City of Seattle. Buildings account for 33% of Seattle's core emissions. The benchmarking policy supports Seattle's goals to reduce energy use and greenhouse gas emissions from existing buildings. In 2013, the City of Seattle adopted a Climate Action Plan to achieve zero net greenhouse gas (GHG) emissions by 2050. Annual benchmarking, reporting and disclosing of building performance are foundational elements of creating more market value for energy efficiency. Per Ordinance (125000), starting with 2015 energy use performance reporting, the City of Seattle will make the data for all building 20,000 SF and larger available annually. This update to the benchmarking mandate was passed by Seattle City Council on February 29, 2016. If you have questions or comments on the data, email us at energybenchmarking@seattle.gov and include Open Data in the subject line."
  "displayType" : string "table"
  "downloadCount" : int 2301
```


Seattle

Données de 2015 :

- 3340 lignes
- 47 colonnes

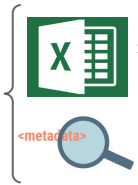
Données de 2016 :

- 3376 lignes
- 46 colonnes

```
print(df_2016.shape)
print(df_2015.shape)
```

```
(3376, 46)
(3340, 47)
```

Nettoyage des données - 1



SEA Building Energy Benchmarking Explore Open Data from the City of Seattle

2016

```
print(df_2016.shape)
print(df_2015.shape)
```

```
(3376, 46)
(3340, 47)
```

2015

Suppression des outliers
+
Suppression des colonnes avec 80% de NA

```
df_2016.shape
```

```
(3344, 41)
```

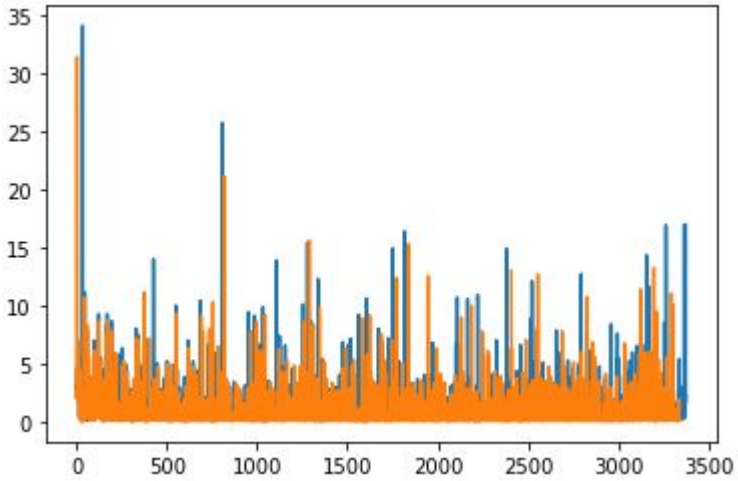
```
df_2015.shape
```

```
(3340, 40)
```

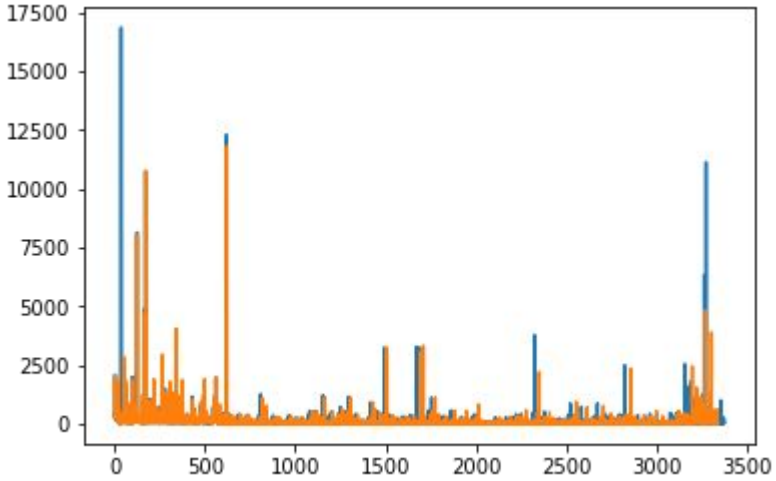
```
colonnes présentes sur données 2015 et absentes sur 2016 :
 {'GHGEmissionsIntensity(kgCO2e/ft2)', 'Seattle Police Department Micro Communi
ty Policing Plan Areas', 'SPD Beats', 'GHGEmissions(MetricTonsCO2e)', 'OtherFue
lUse(kBtu)', 'Zip Codes'}

colonnes présentes sur données 2016 et absentes sur 2015 :
 {'GHGEmissionsIntensity', 'TotalGHGEmissions'}
```

Nettoyage des données - 2




```
2016['GHGEmissionsIntensity'])
2015['GHGEmissionsIntensity(kgCO2e/ft2)'])
```




```
2016['TotalGHGEmissions'])
2015['GHGEmissions(MetricTonsCO2e)'])
```

Nettoyage des données - 3

- Dans la colonne **buldingtype** : garder seulement les « non residential », « sps-district k12 », « Campus ».
- Dans certaines colonnes comme « Neighborhood » rassembler les mots avec ou sans majuscule comme : north & NORTH
- Remplacer les dernières valeurs manquantes par un KNNimputer

```
df['Neighborhood'].replace('North', 'NORTH', inplace=True)
df['Neighborhood'].replace('CENTRAL', 'Central', inplace=True)
df['Neighborhood'].replace('Northwest', 'NORTHWEST', inplace=True)
df['Neighborhood'].replace('Ballard', 'BALLARD', inplace=True)
df['Neighborhood'].replace('DELRIDGE NEIGHBORHOODS', 'DELRIDGE', inplace=True)
df['Neighborhood'].replace('Delridge', 'DELRIDGE', inplace=True)
df['City'].replace('SEATTLE', 'Seattle', inplace=True)
df['ComplianceStatus'].replace('Error - Correct Default Data', np.nan, inplace=
df['ComplianceStatus'].replace('Not Compliant', 'Non-Compliant', inplace=True)
df['ComplianceStatus'].replace('Missing Data', np.nan, inplace=True)
```

TotalGHGEmissions

SiteEnergyUse(kBtu)

Analyse et exploitation

1. Quelle énergie est utilisée : stream, natural et stream
2. Quelle énergie est la plus utilisée : stream, natural ou electric
3. L'âge des bâtiments = datayear – yearbuild

```
df['AgeBat'] = df['DataYear']-df['YearBuilt']
df = df.drop(['DataYear','YearBuilt'],axis=1)


#créer 3 colonne avec des 0
df['most_use_steam'] = 0
df['most_use_natural'] = 0
df['most_use_electric'] = 0

### l'énergie qui est la plus utilisé (electrcité/natural/streamuse), rajouter 1
df.loc [(df['SteamUse(kBtu)'] > df['Electricity(kBtu)']) & (df['SteamUse(kBtu)']
df.loc [(df['NaturalGas(kBtu)'] > df['Electricity(kBtu)']) & (df['NaturalGas(kBt
df.loc [(df['Electricity(kBtu)'] > df['NaturalGas(kBtu)']) & (df['Electricity(kB

# quels sont les énergies utilisés
df.loc [df ['SteamUse(kBtu)'] == 0, 'SteamUse(kBtu)'] = 0
df.loc [df ['SteamUse(kBtu)'] != 0, 'SteamUse(kBtu)'] = 1
df.loc [df ['Electricity(kBtu)'] == 0, 'Electricity(kBtu)'] = 0
df.loc [df ['Electricity(kBtu)'] != 0, 'Electricity(kBtu)'] = 1
df.loc [df ['NaturalGas(kBtu)'] == 0, 'NaturalGas(kBtu)'] = 0
df.loc [df ['NaturalGas(kBtu)'] != 0, 'NaturalGas(kBtu)'] = 1

### renommer pour ne pas croire que j'utilise des données fuites
df = df.rename(columns = {'SteamUse(kBtu)': 'use_stream', 'Electricity(kBtu)': '
                          'NaturalGas(kBtu)':'use_natural'})
#df=df.drop(['SteamUse(kBtu)','Electricity(kBtu)','NaturalGas(kBtu)'],axis=1)
```

Suppression des données qui sont pas utilisables

```
df = df.drop(['OSEBuildingID','Address','City','State','TaxParcelIdentificationN
              'PropertyName','SourceEUI(kBtu/sf)','PropertyGFATotal','TotalGHGEmi
```

Analyse et exploitation

Faire un pipeline pour faire un preprocessor et transformer les variables qualitatives en 0 et 1 grâce à binaryencoder et pour les variables quantitatives standardiser les données.

Tester plusieurs modèles différents en comparaison avec le modèle de base DummyRegressor.

Couper le dataset en deux parties : train set (80%) et un test set (20%)

Évaluer les modèles grâce à différentes métriques telles que : RMSE, R2, valeur absolue, le temps de traitement… avec une validation croisé (3) et 40n_iter.

RMSE : l'erreur quadratique moyenne , permet de voir la différence entre la valeur prédite et la valeur réelle.

R2 : c'est le coefficiente de détermination (entre 0 et 1). Permet de juger la qualité de la régression linéaire.

Absolue : mesure des erreurs entre les observations prédites et réelles.

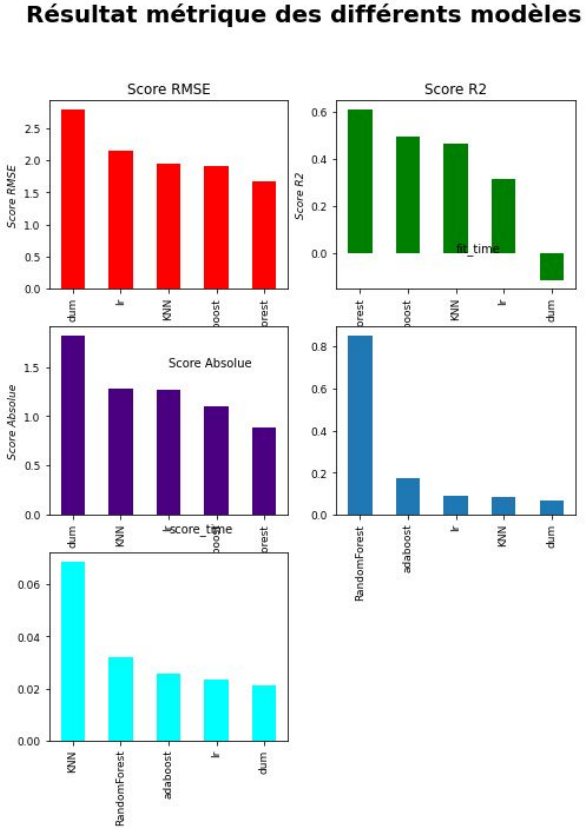
Fit time : le temps d'ajustement de l'estimateur sur l'ensemble de trains pour chaque fractionnement de CV.

Score time : le temps de notation de l'estimateur sur l'ensemble de tests pour chaque fractionnement de CV.

Visualisation

TotalGHGEmissions

- **Random Forest**
- **adaboost**
- **KNN**



le modèle dummy de base à un R2 qui se rapproche de 0 c'est-à-dire que peu importe l'entrée il va sortir la même réponse

Le modèle avec le meilleur R2 et RMSE est le random forest mais c'est aussi celui qui met le plus de temps à s'entraîner

Visualisation

SiteEnergyUse(kBtu)

le modèle dummy de base à un R2 qui se rapproche de 0 c'est-à-dire que peu importe l'entrée il va sortir la même réponse

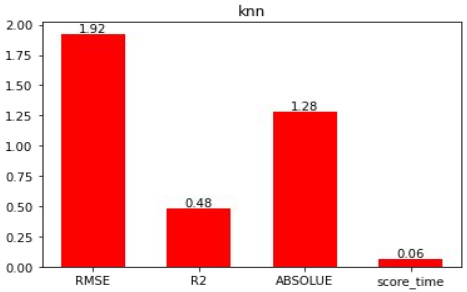
Le modèle avec le meilleur R2 et RMSE est le random forest mais c'est aussi celui qui met le plus de temps à s'entraîner

Les 3 modèles retenus sont random forest/ adaboost et KNN.

Même conclusion que la target précédente mais avec des scores moins performants

Analyse et exploitation

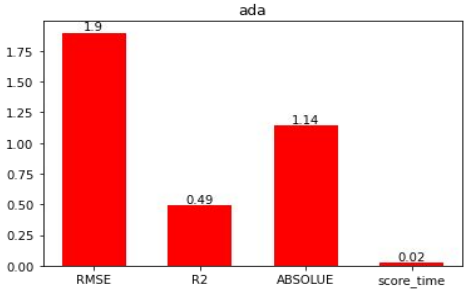
KNN



Les meilleurs paramètres :
Algorithm = brute
Leaf size = 39
Neighbors = 10

Modèle basé sur les voisins les plus proches. Le score est prédit par interpolation local des cibles associées des voisins les plus proches dans l'ensemble d'apprentissage.

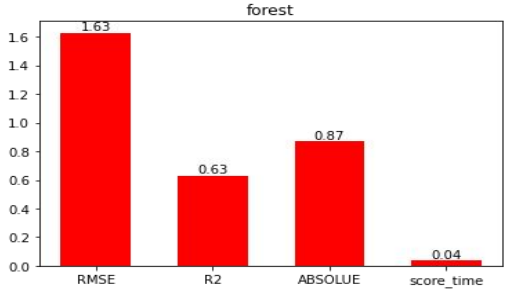
Adaboost



Les meilleurs paramètres :
N_estimator = 95
Learning rate= 4
loss = exponential

Un régresseur adaboost, s'ajuste sur l'ensemble des données d'origine puis ajuste les copies supplémentaires du régresseur sur le même ensemble de données mais le poids varie en fonction des erreurs

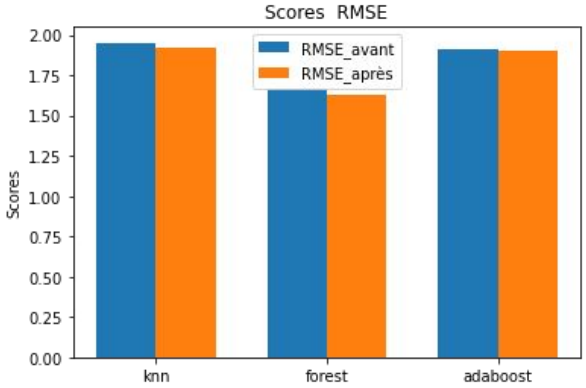
Random Forest



Les meilleurs paramètres :
N_estimator = 115
Bootstrap= true
Max sample leaf = 1

C'est un méta-estimateur qui ajuste un certain nombres d'arbres de décision sous divers sous-échantillons et utilise la moyenne pour améliorer la précision prédictive et contrôler le sur-ajustement.

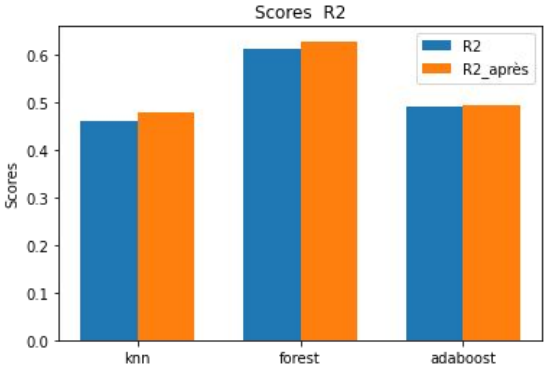
Analyse et exploitation



Plus le score est proche de 0, plus le modèle est performant

Les modèles sont plus performants en cherchant les meilleurs paramètres

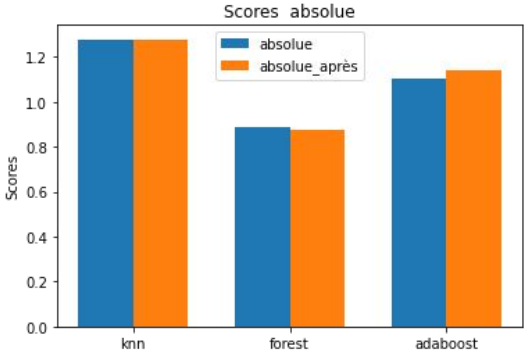
Le meilleurs modèle par rapport au RMSE est le random forest



Plus le score est proche de 1, plus le modèle est performant

Les modèles sont plus performants en cherchant les meilleurs paramètres

Le meilleurs modèle par rapport au R2 est le random forest



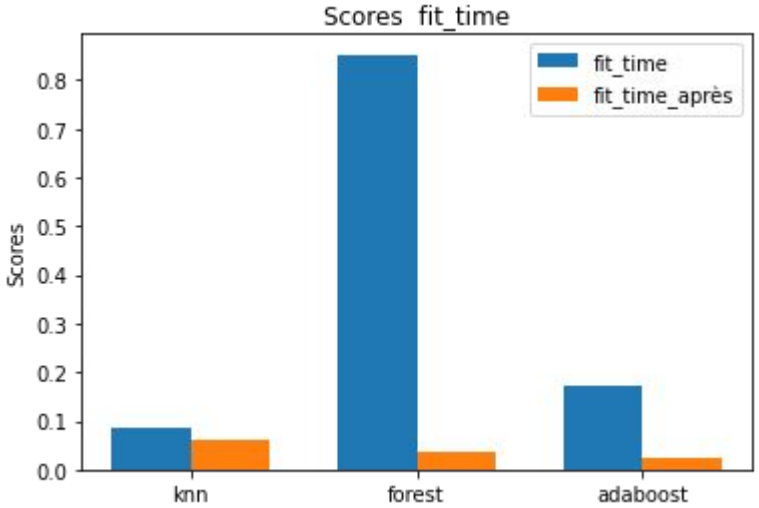
Plus le score est proche de 0, plus le modèle est performant

Les modèles sont plus performants en cherchant les meilleurs paramètres

Le meilleur modèle par rapport aux absolus est le random forest

Comparaison avant après

Analyse et exploitation

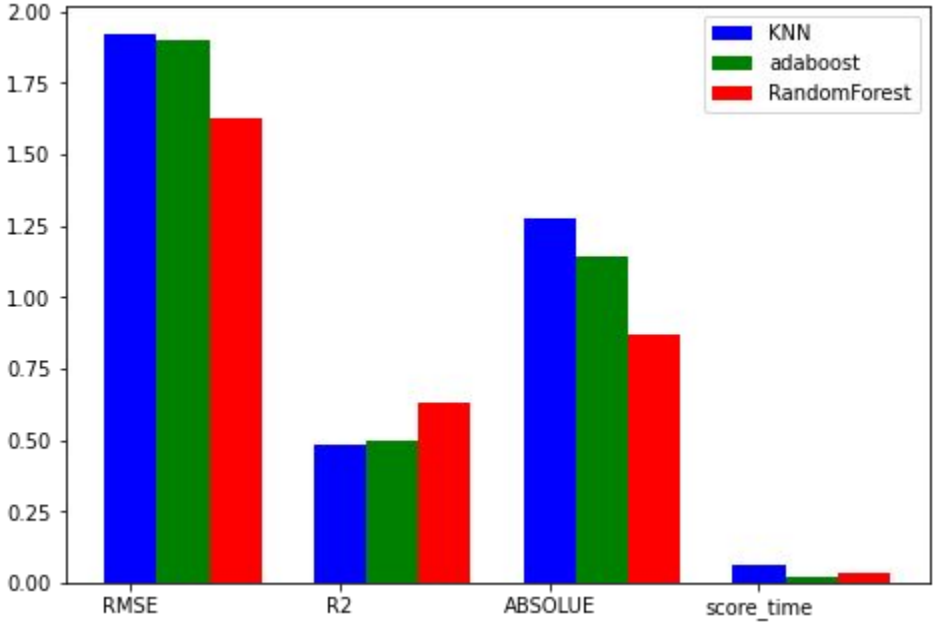


Lorsqu'on cherche les meilleurs paramètres, les modèles sont plus longs.

Le random forest est très long car n_estimator est très grand et il y a beaucoup de combinaisons.

Comparaison avant après

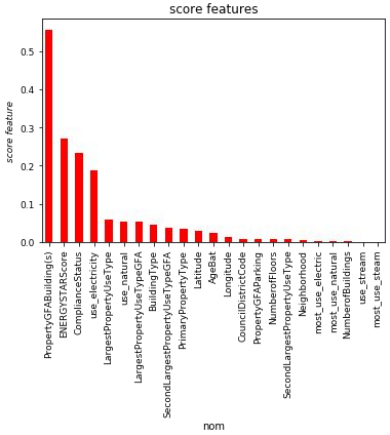
Analyse et exploitation

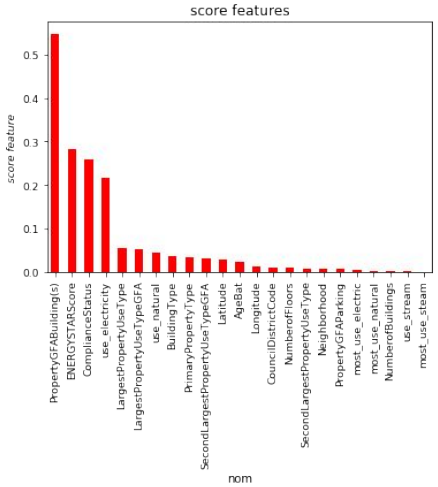


Le meilleur modèle par rapport aux metrics est le random forest.

Le random forest est le modèle le plus long en terme de temps d'entraînement

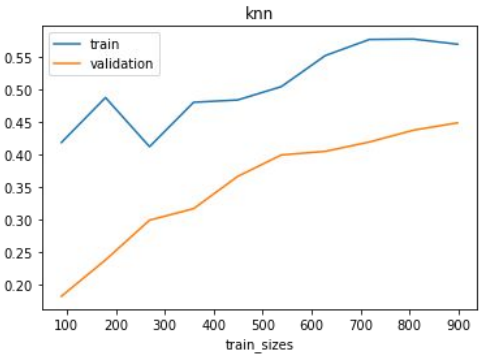
Analyse et exploitation

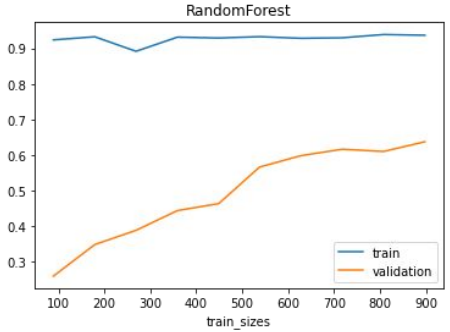


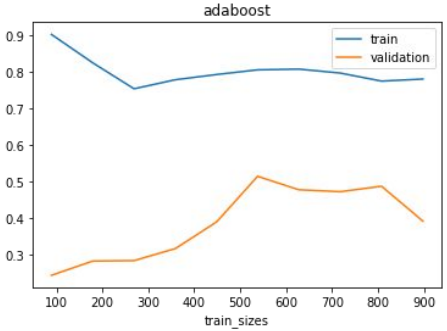


Avec ENERGYSTARScore
RMSE : 1,88
R2 : 0,66

Sans ENERGYSTARScore
RMSE : 1,89
R2 : 0,65







Les courbes nous montrent que des données supplémentaires nous permettraient d'avoir surement de meilleurs résultats.

Conclusion

- Le **ENERGYScore** ne joue pas un rôle très important dans l'amélioration des différents modèles et pour les deux targets.
- Le meilleur modèle pour les deux targets est le Random forest mais c'est aussi le plus long à exécuter.
- Malgré les datasets il manque un peu de données pour que les modèles soient plus efficaces et qu'ils soient plus performants.

Merci pour votre attention !!