



# Anticiper les besoins en énergie de la ville de Seattle

# Le Besoin

## Qui?

- La ville de Seattle
- Non-résidentiel
- Energie et gaz

## Quoi?

- Prédire les besoins futurs
- Etudier la valeur de EnergyStarScore dans le modèle

## Comment?





- Appliquer des modèles de prédiction linéaires
- Ne pas utiliser les variables énergétiques directement
- Définir le meilleur modèle

# Ressources & méthode de pensée

## Plusieurs Datasets

## Quel dataset prioriser?

## Quelle méthode?

Nom	Type	Taille
 2015-building-energy-benchmarking	Fichier CSV	1 552 Ko
 2016-building-energy-benchmarking	Fichier CSV	1 207 Ko
 socrata_metadata_2015-building-energy-...	Fichier JSON	54 Ko
 socrata_metadata_2016-building-energy-...	Fichier JSON	45 Ko

- Données utiles à la question?
- Indicateurs pouvant être mis en relation?

- Découverte du dataset
- Nettoyage
- Affinage
  - Tri des variables
  - Remplissage
- Phase exploratoire
- Application des modèles et comparaison

# Méthode d'analyse exploratoire

## Etudier les datasets

Comprendre ce qu'on a à l'intérieur:

- Beaucoup de donnée?
- Présente? Manquante?
- Eparses?
- Constante? Régulière?
- Organisée?
- Optimisée?
- Quels indicateurs?

## Créer un jeu d'étude propre

- Variables claires
- Donnée organisée
- Gestion des données manquantes et aberrantes
- Feature engineering

## Explorer Modéliser Comparer

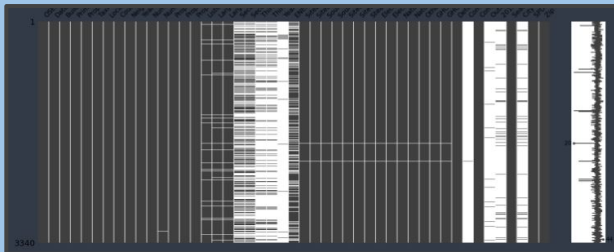
- Création de pipeline & cross validation
- Choix et application des modèles via les pipe
- Comparaison

## Etudier les datasets

## Avant nettoyage

Comprendre ce qu'on a à l'intérieur:

- Beaucoup de donnée?
- Présente? Manquante?
- Eparses?
- Constante? Régulière?
- Organisée?
- Optimisée?
- Quels indicateurs?



```
1 diff_col5 = data_2015.columns.difference(data_2016.columns)
2 diff_col5
```

```
Index(['2010 Census Tracts', 'City Council Districts', 'Comment',
      'GHGEmissions(MetricTonsCO2e)', 'GHGEmissionsIntensity(kgCO2e/ft2)',
      'Location', 'OtherFuelUse(kBtu)', 'SPD Beats',
      'Seattle Police Department Micro Community Policing Plan Areas',
      'Zip Codes\r\n'],
      dtype='object')
```

```
1 diff_col6 = data_2016.columns.difference(data_2015.columns)
2 diff_col6
```

```
Index(['Address', 'City', 'Comments', 'GHGEmissionsIntensity\r\n', 'Latitude',
      'Longitude', 'State', 'TotalGHGEmissions', 'ZipCode'],
      dtype='object')
```

```
1 def doublons(k):
2     print("Il y a " + str(k.index.size - k.drop_duplicates().index.size) + " doublons")
3
4     doublons(data)
5
6
7
8
9
10
11 y = e doublons
```

```
1 print(data_2015.shape)
2 print(data_2016.shape)
```

(3340, 47)

(3376, 46)

# Les méthodes de nettoyage

```
1 data_energyuse.shape
```

```
(3190, 220)
```

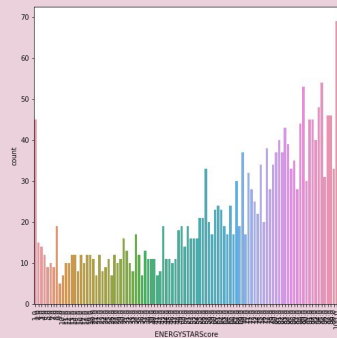
- Suppression des colonnes/lignes vides, inutiles(variables énergies), aberrantes, ou en double
- Filtrage sur les bâtiments non-résidentiels
- Changement des noms de variables communes
- Elimination des outliers (energystarscore,...)
- Vérification de variables sommes (gfa, siteenergyuse)
- Ajout d'une variable otherfuels et categorielle energie
- Récupération de la longitude et latitude
- Concaténation de 2015 et 2016 dans un même dataset
- Encoder sur les variables object
- Remplissage des valeurs manquantes via mediane
- Passage au log des variables target
- séparation des deux dataset contenant les variables target respectives

## Etudier les datasets

## Analyses Univariées

Comprendre ce qu'on a à l'intérieur:

- Beaucoup de donnée?
- Présente? Manquante?
- Eparses?
- Constante? Régulière?
- Organisée?
- Optimisée?
- Quels indicateurs?



```
les variables suivant une loi normale sont:  
['NumberOfFloors', 'LargestPropertyUseTypeGFA', 'SecondLargestPropertyUseTypeGFA', 'ThirdLargestPropertyUseTypeGFA', 'ENERGYSTARScore']
```

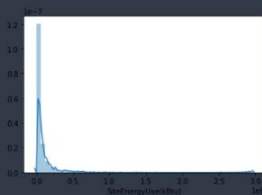
```
les variables ne suivant pas une loi normale sont:  
['OSEBuildingID', 'DataYear', 'CouncilDistrictCode', 'YearBuilt', 'NumberOfBuildings', 'PropertyGFATotal', 'PropertyGFAParking', 'Property  
GFABuilding(s)', 'SiteEnergyUse(kBtu)', 'SiteEnergyUseWN(kBtu)', 'SteamUse(kBtu)', 'Electricity(kBtu)', 'NaturalGas(kBtu)', 'GHGEmissions  
(MetricTonsCO2e)', 'ZipCode', 'Latitude', 'Longitude', 'OtherFuels']
```

```
1 stats, pval = shapiro(data_NR['SiteEnergyUse(kBtu)'])  
2 print(pval)  
3  
4 stats, pval = shapiro(np.log(data_NR['SiteEnergyUse(kBtu)']))  
5 print(pval)
```

```
0.8  
1.0
```

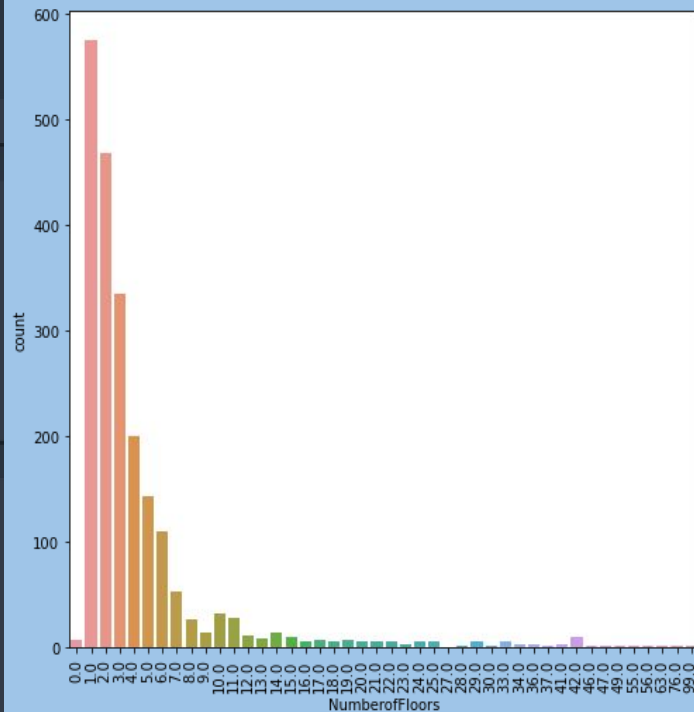
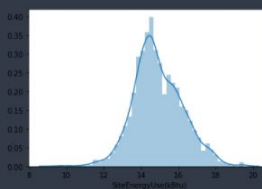
```
1 sns.distplot(data_NR['SiteEnergyUse(kBtu)'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2241c1c828>
```



```
1 sns.distplot(np.log(data_NR['SiteEnergyUse(kBtu)'])[data_NR['SiteEnergyUse(kBtu)']!=0])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x22414fe737b>
```

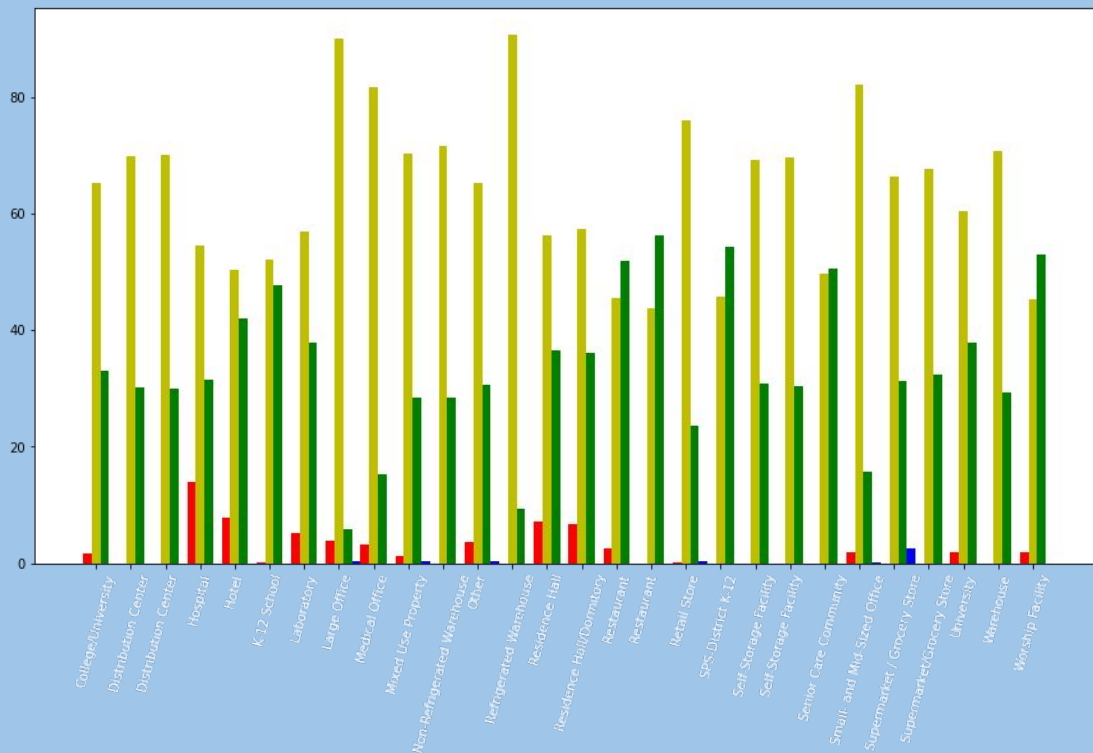


## Etudier les datasets

## Analyse multivariée

Comprendre ce qu'on a à l'intérieur:

- Beaucoup de donnée?
- Présente? Manquante?
- Eparses?
- Constante? Régulière?
- Organisée?
- Optimisée?
- Quels indicateurs?



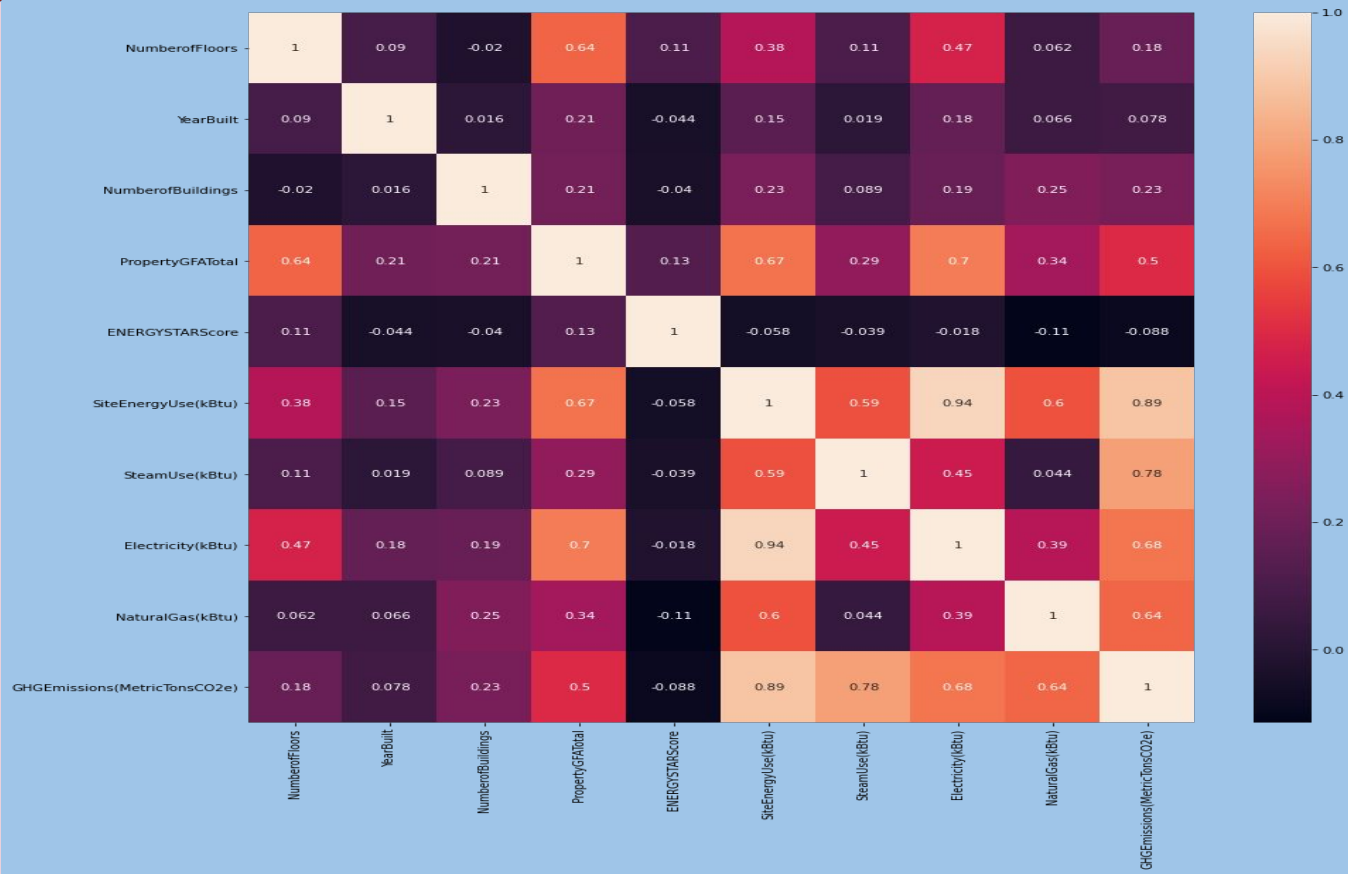


## Etudier les datasets

## Analyse multivariée

Comprendre ce qu'on a à l'intérieur:

- Beaucoup de donnée?
- Présente? Manquante?
- Eparses?
- Constante? Régulière?
- Organisée?
- Optimisée?
- Quels indicateurs?

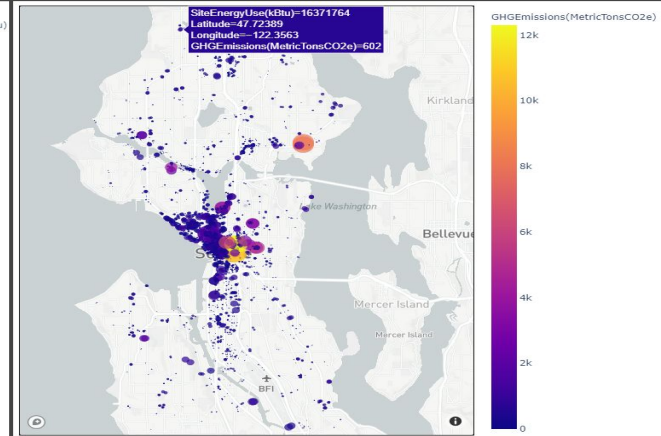
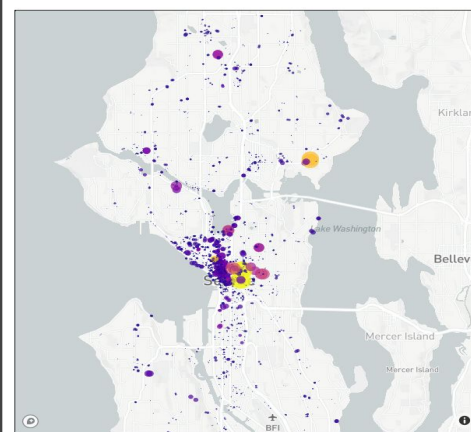
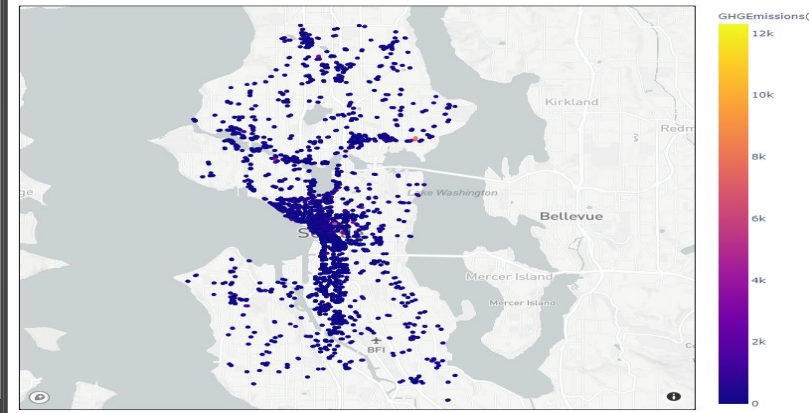
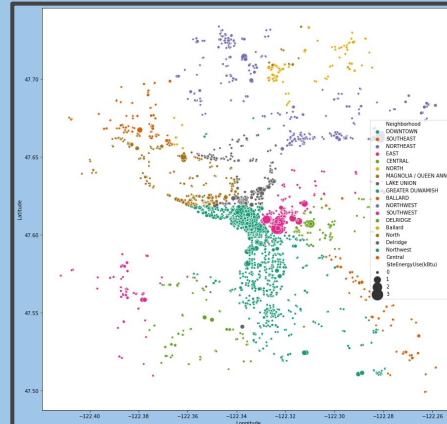


## Etudier les datasets

## Data présente par Région sur les indicateurs choisis

Comprendre ce qu'on a à l'intérieur:

- Beaucoup de donnée?
- Présente? Manquante?
- Eparse?
- Constante? Régulière?
- Organisée?
- Optimisée?
- Quels indicateurs?



## Créer un jeu d'étude propre

- Indicateurs clairs
- Donnée organisée
- Gestion des données manquantes
- Gestion des indicateurs

## Machine Learning et méthodologie

Les mots d'ordre:

Modularité &  
automatisation

Optimisation

Robustesse

Modèles linéaires et  
classiques

Scoring

Comparaison

Importance des  
variables

En découle:

Pipelines

Gridsearch

Crossvalidation

Hyperopt sklearn

xgboost

Lasso, Ridge, SVR,  
Random Forest

DATAFRAME!

r2 et rmse

## Créer un jeu d'étude propre

- Indicateurs clairs
- Donnée organisée
- Gestion des données manquantes
- Gestion des indicateurs

## Machine Learning et méthodologie

Les mots d'ordre:

Modularité & automatisisation

Optimisation

Robustesse

Modèles linéaires et classiques

Scoring

Comparaison

Importance des variables

En découle:

Pipelines

Gridsearch

Crossvalidation

Hyperopt sklearn

xgboost

Lasso, Ridge, SVR,  
Random Forest

DATAFRAME!

r2 et rmse



## Créer un jeu d'étude propre

- Indicateurs clairs
- Donnée organisée
- Gestion des données manquantes
- Gestion des indicateurs

## Machine Learning et méthodologie

Les mots d'ordre:

Modularité & automatisisation

Optimisation

Robustesse

Modèles linéaires et classiques

Scoring

Comparaison

Importance des variables

En découle:

Pipelines

Gridsearch

Crossvalidation

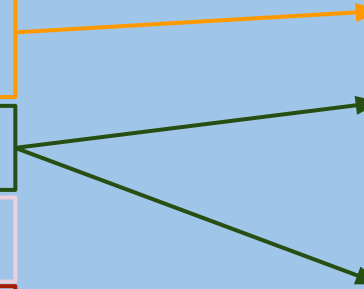
Hyperopt sklearn

xgboost

Lasso, Ridge, SVR,  
Random Forest

DATAFRAME!

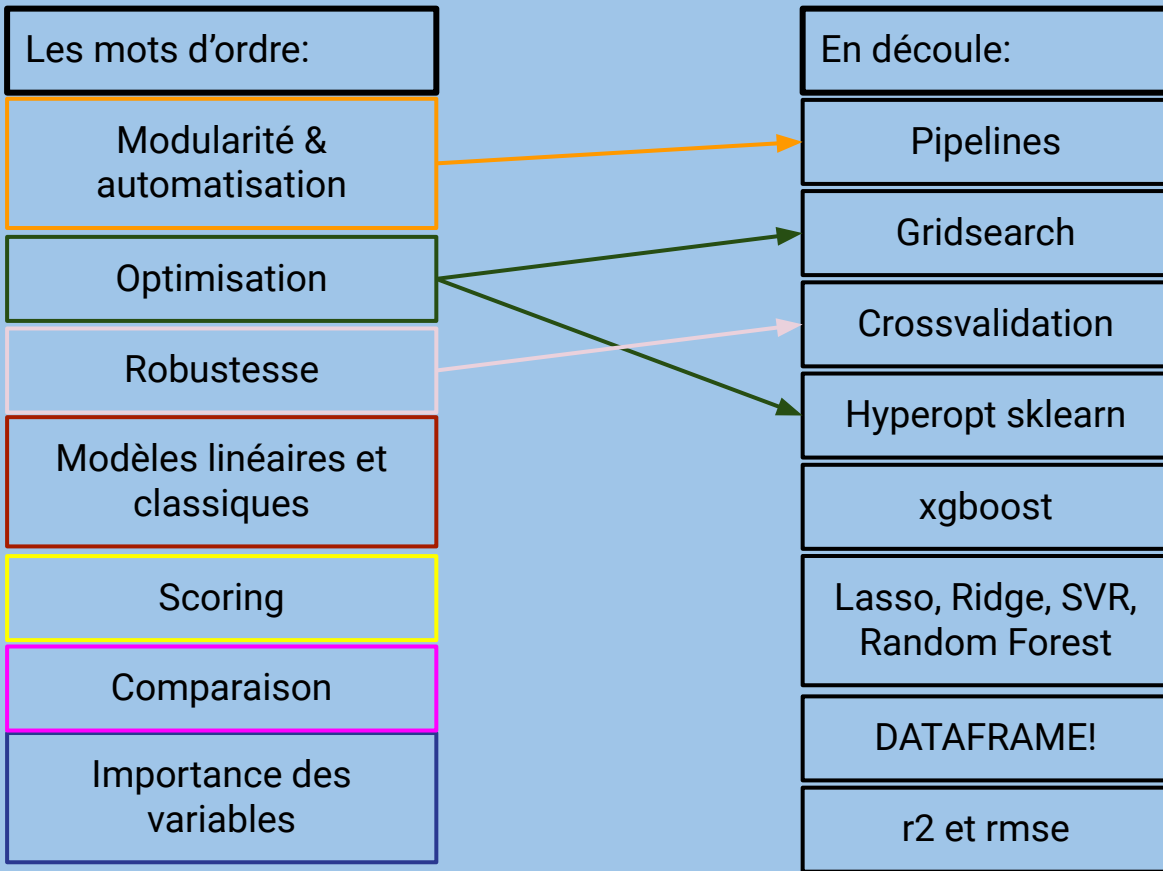
r2 et rmse



## Créer un jeu d'étude propre

- Indicateurs clairs
- Donnée organisée
- Gestion des données manquantes
- Gestion des indicateurs

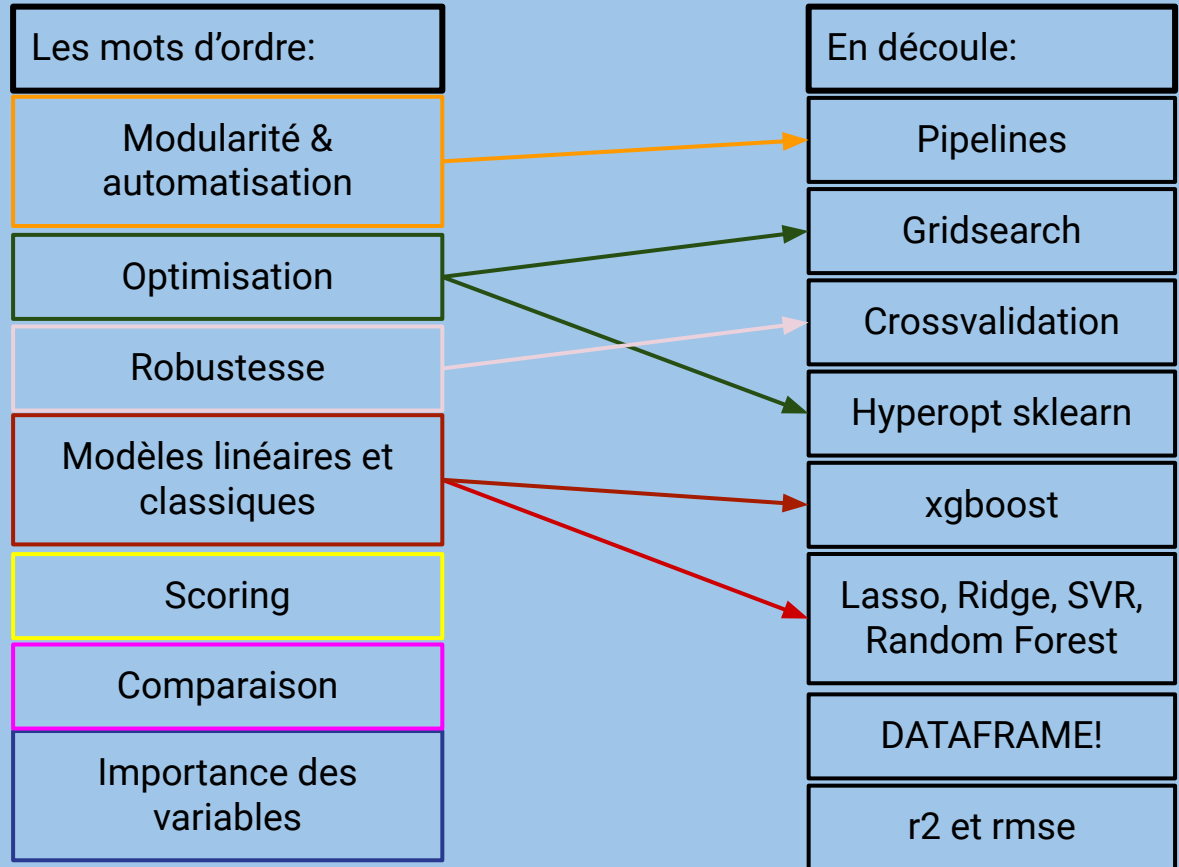
## Machine Learning et méthodologie



## Créer un jeu d'étude propre

- Indicateurs clairs
- Donnée organisée
- Gestion des données manquantes
- Gestion des indicateurs

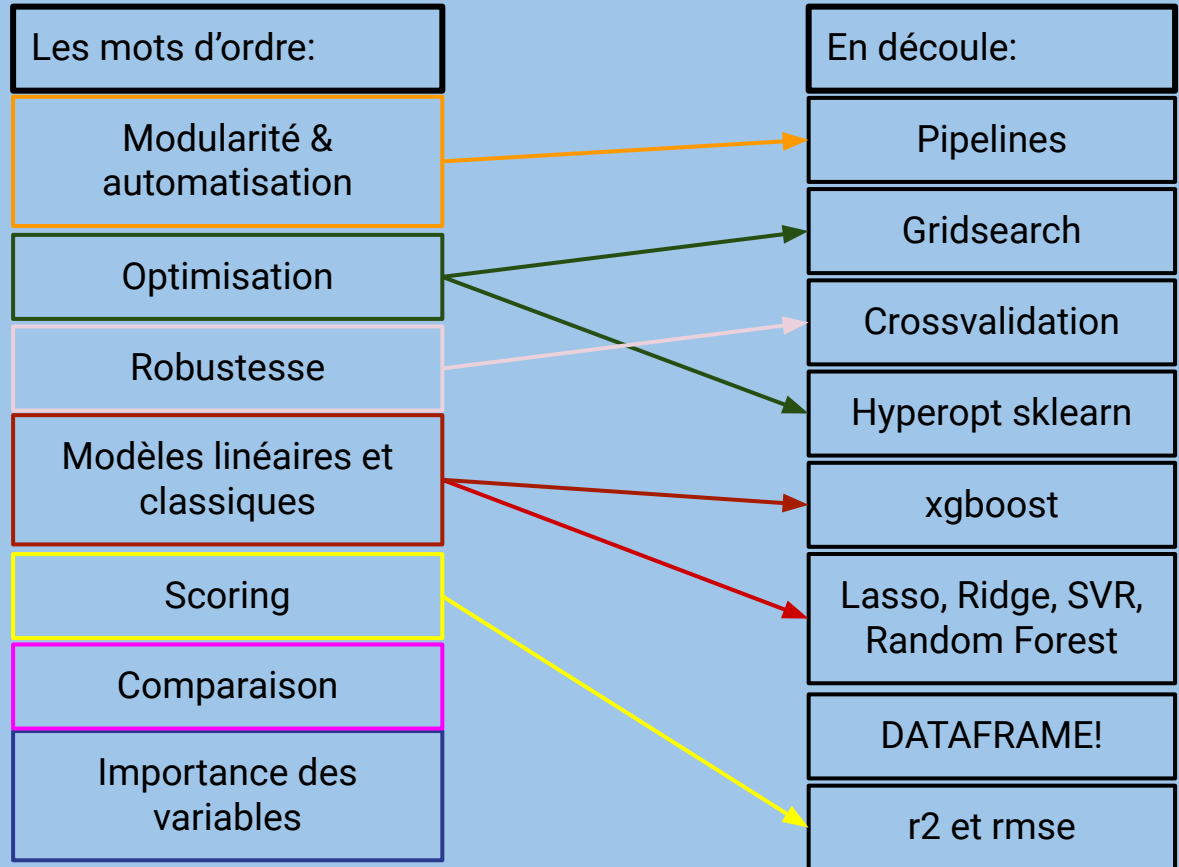
## Machine Learning et méthodologie



## Créer un jeu d'étude propre

- Indicateurs clairs
- Donnée organisée
- Gestion des données manquantes
- Gestion des indicateurs

## Machine Learning et méthodologie

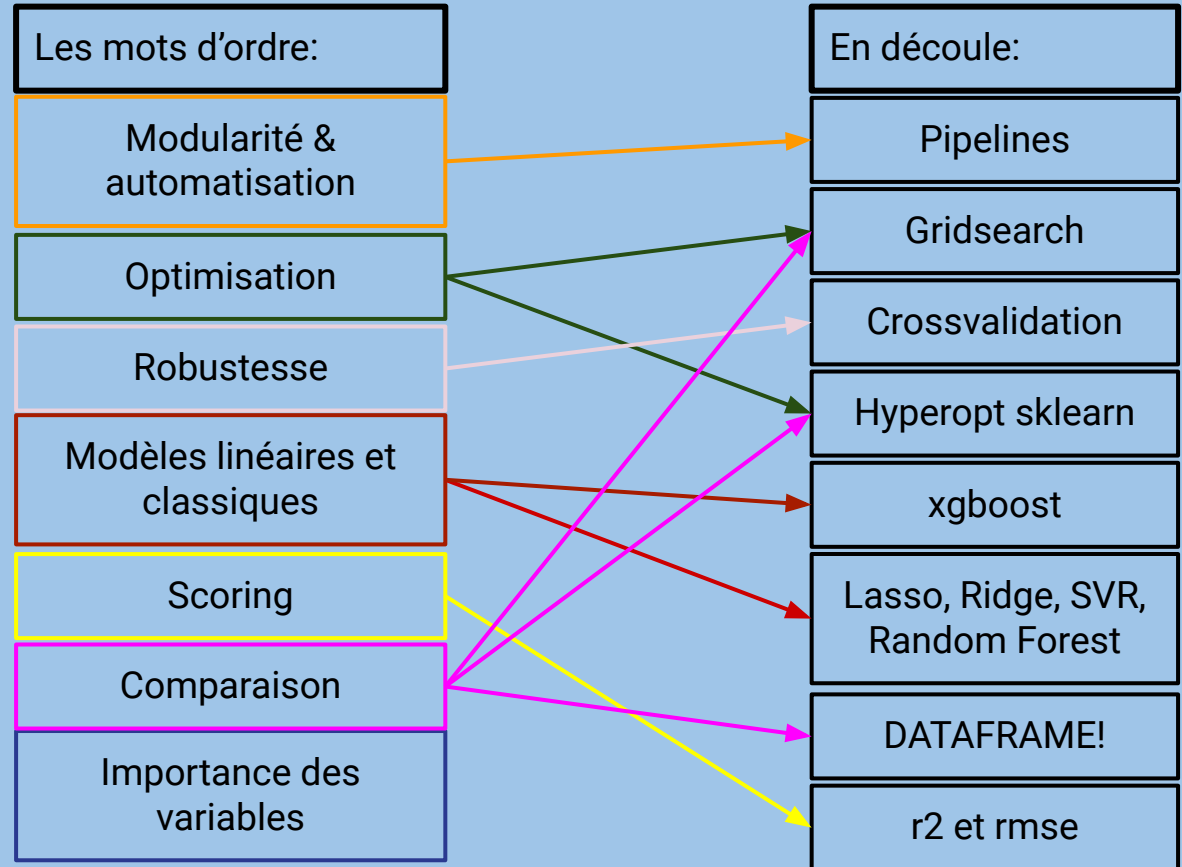




## Créer un jeu d'étude propre

- Indicateurs clairs
- Donnée organisée
- Gestion des données manquantes
- Gestion des indicateurs

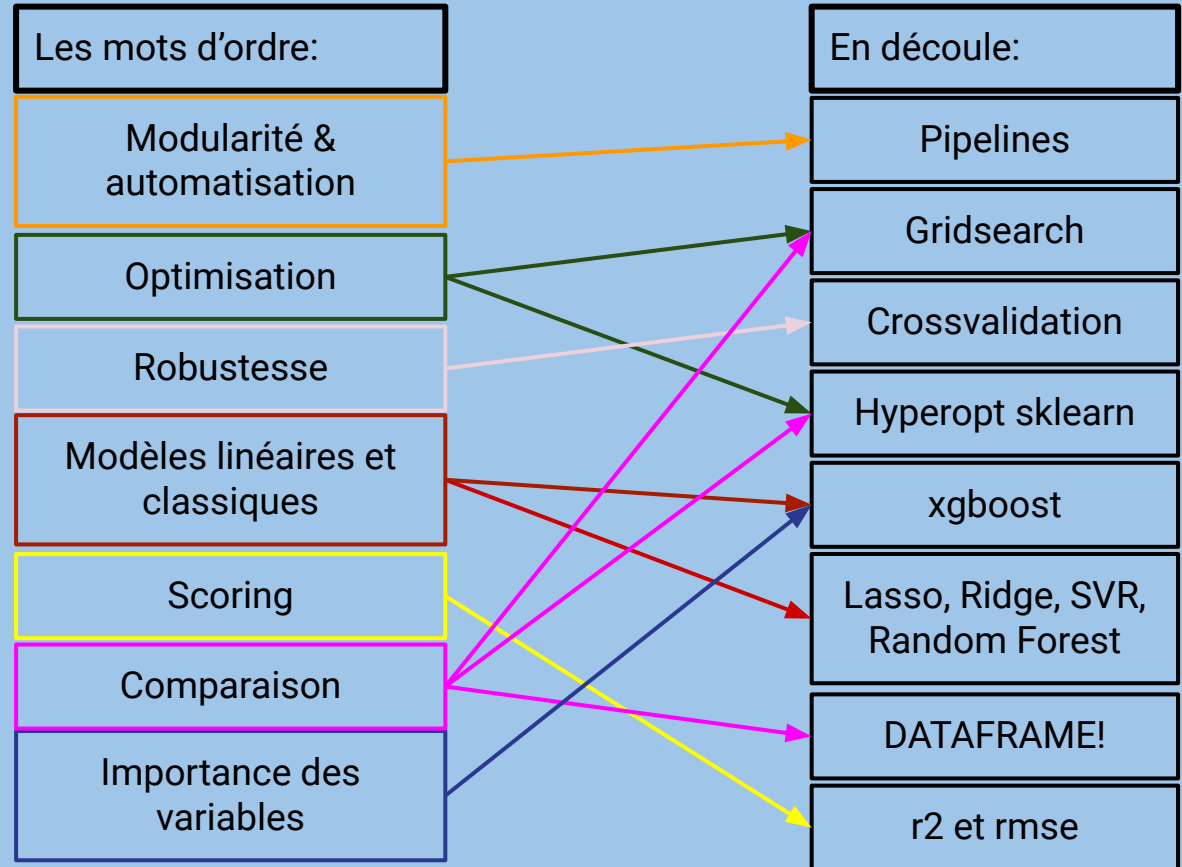
## Machine Learning et méthodologie



## Créer un jeu d'étude propre

- Indicateurs clairs
- Donnée organisée
- Gestion des données manquantes
- Gestion des indicateurs

## Machine Learning et méthodologie

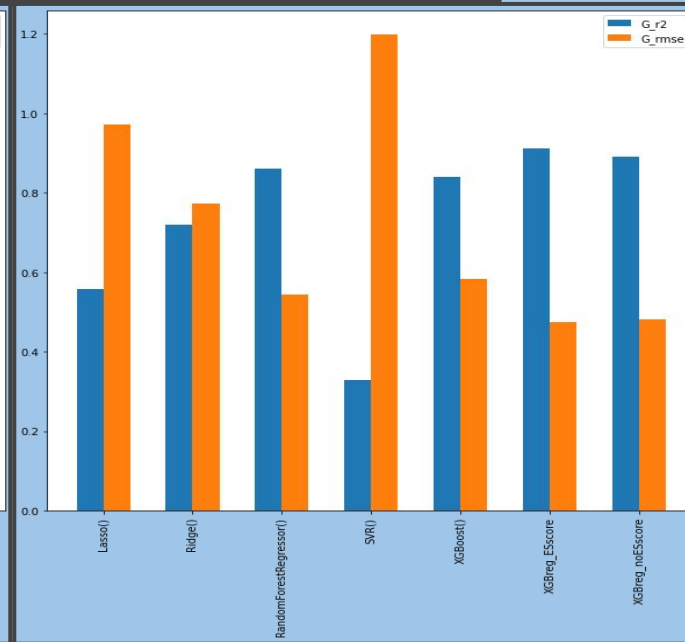
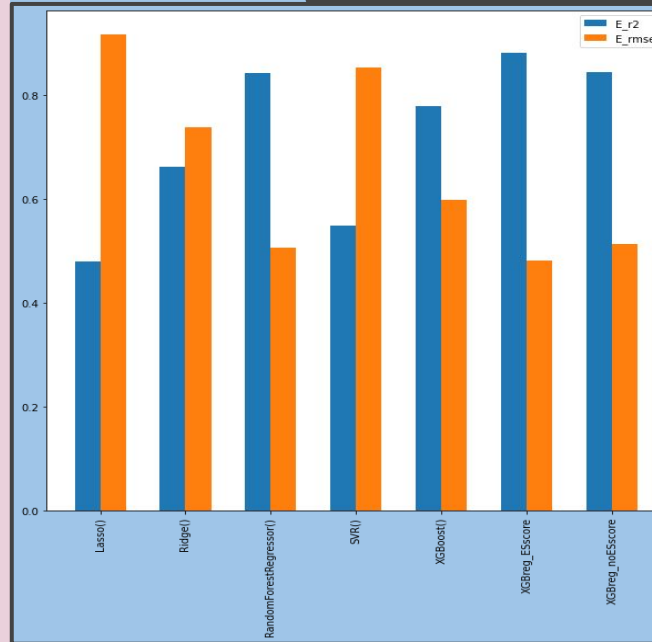


## Créer un jeu d'étude propre

- Indicateurs clairs
- Donnée organisée
- Gestion des données manquantes
- Gestion des indicateurs

## Résultat final

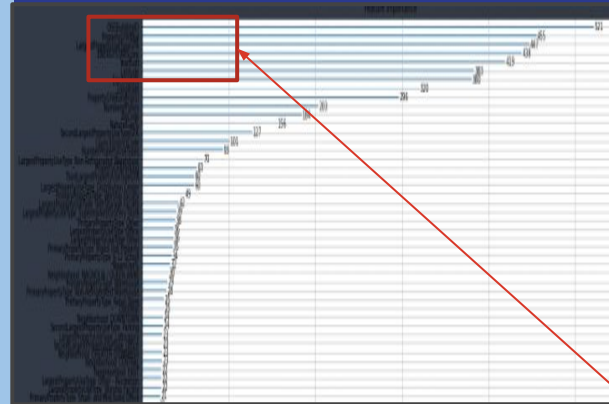
	E_r2	E_rmse	G_r2	G_rmse
<b>Lasso()</b>	0.479274	0.916199	0.557335	0.972132
<b>Ridge()</b>	0.661862	0.738299	0.720652	0.772255
<b>RandomForestRegressor()</b>	0.841542	0.505409	0.861889	0.543002
<b>SVR()</b>	0.549199	0.852467	0.327411	1.198293
<b>XGBoost()</b>	0.778460	0.597601	0.841075	0.582483
<b>XGBreg_ESScore</b>	0.886400	0.448432	0.912419	0.474390



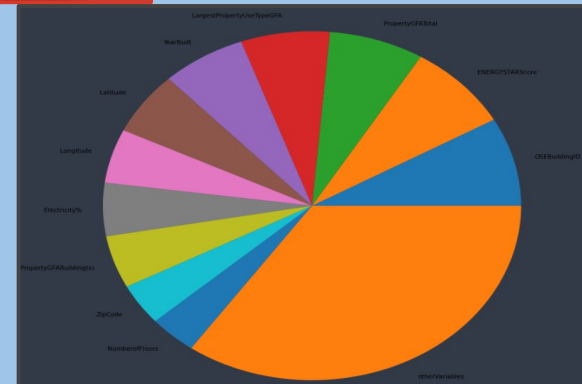
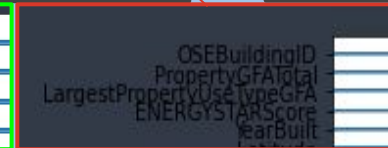
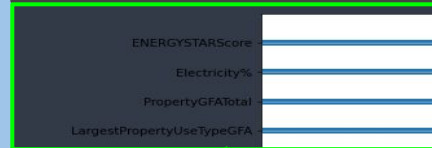
## Créer un jeu d'étude propre

- Indicateurs clairs
- Donnée organisée
- Gestion des données manquantes
- Gestion des indicateurs

## De l'importance d'EnergyStarScore



	E_r2	E_rmse	G_r2	G_rmse
Lasso()	0.479274	0.916199	0.557335	0.972132
Ridge()	0.661862	0.738299	0.720652	0.772255
RandomForestRegressor()	0.841542	0.505409	0.861889	0.543002
SVR()	0.549199	0.852467	0.327411	1.198293
XGBoost()	0.778460	0.597601	0.841075	0.582483
XGBreg_ESscore	0.880932	0.480780	0.912419	0.474390
XGBreg_noESscore	0.843689	0.513560	0.891202	0.480167



# Apprentissage et ressenti

Critique:  
Qu'est-ce que j'ai appris et ma  
pensée sur les technologies  
abordées

Croissance max.

les meilleurs modeles: ¶

## Energy

```
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=0.8829119548287974,  
colsample_bynode=1, colsample_bytree=0.6628937541385697, gamma=7.317511939125787e-07, gpu_id=-1,  
importance_type='gain', interaction_constraints="", learning_rate=0.06089857803867583, max_delta_step=0,  
max_depth=10, min_child_weight=2, missing=nan, monotone_constraints=(), n_estimators=2600, n_jobs=16,  
num_parallel_tree=1, objective='reg:linear', random_state=0, reg_alpha=0.178742166256787,  
reg_lambda=2.002400735833588, scale_pos_weight=1, seed=0, subsample=0.5638344363863476,  
tree_method='exact', validate_parameters=1, verbosity=None)
```

## gas

```
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=0.6726521725890362,  
colsample_bynode=1, colsample_bytree=0.7974066587372607, gamma=0.0023815763092893145, gpu_id=-1,  
importance_type='gain', interaction_constraints="", learning_rate=0.04480475685904021, max_delta_step=0,  
max_depth=4, min_child_weight=6, missing=nan, monotone_constraints=(), n_estimators=1600, n_jobs=16,  
num_parallel_tree=1, objective='reg:linear', random_state=4, reg_alpha=0.46901677521628776,  
reg_lambda=2.2344712857026297, scale_pos_weight=1, seed=4, subsample=0.6572741862686372,  
tree_method='exact', validate_parameters=1, verbosity=None)
```

Croissance max.