Seattle Building Energy Benchmarking

Anticipez les besoins en consommation électrique de bâtiments

Plan de présentation

1. **Présentation de problématique**
2. **Analyse exploratoire**
   - Cleaning
   - Exploration de données
   - Feature Engineering
3. **Modélisation**

   A – Emissions de CO2

   B – Consommation d'énergie
   - Entraînement de plusieurs modèles
   - Validation croisée
   - Choix de modèles
4. **Finalisation de modèles**

   A – Emissions de CO2

   B – Consommation d'énergie
   - Ajustement de paramètres
   - Méthodes ensemblistes
   - Evaluation de modèle final
   - Prédiction en fonction de type de bâtiment
5. **Evaluation d'intérêt d'ENERGY STAR Score dans les prédictions d'émissions**
6. **Conclusion**

1. PRÉSENTATION DE LA PROBLÉMATIQUE

1. Présentation de la problématique
But de projet

- Pour atteindre l'objectif de Seattle, neutralité en émissions de carbone en 2050, notre équipe s'intéresse aux émissions des bâtiments non destinés à l'habitation
- Nous disposons de relevés annuels de consommation, mais ces relevés sont couteux à obtenir
- But de projet :
  - Etablir une méthode basé sur les données déclaratives du permis d'exploitation commerciale
  - Prédire la consommation de bâtiments pour lesquels les relevés n'ont pas encore été mesurés
  - Evaluer l'intérêt d'ENERGY STAR Score pour les prédictions, car le calcul de score est fastidieux

<u>De point de vu pratique :</u>

Le projet consiste à trouver et à optimiser un modèle pertinent à l'aide des algorithmes de Machine Learning pour prédire :

A) Les émissions de CO2
B) La consommation totale d'énergie

⇒ Trouver des features appropriées à l'aide d'exploration de données

⇒ Transformer les features existantes (ex. relevés de consommations)

⇒ Evaluer l'importance de feature ENERGY STAR Score dans le modèle A)

1. Présentation de la problématique

Description de données

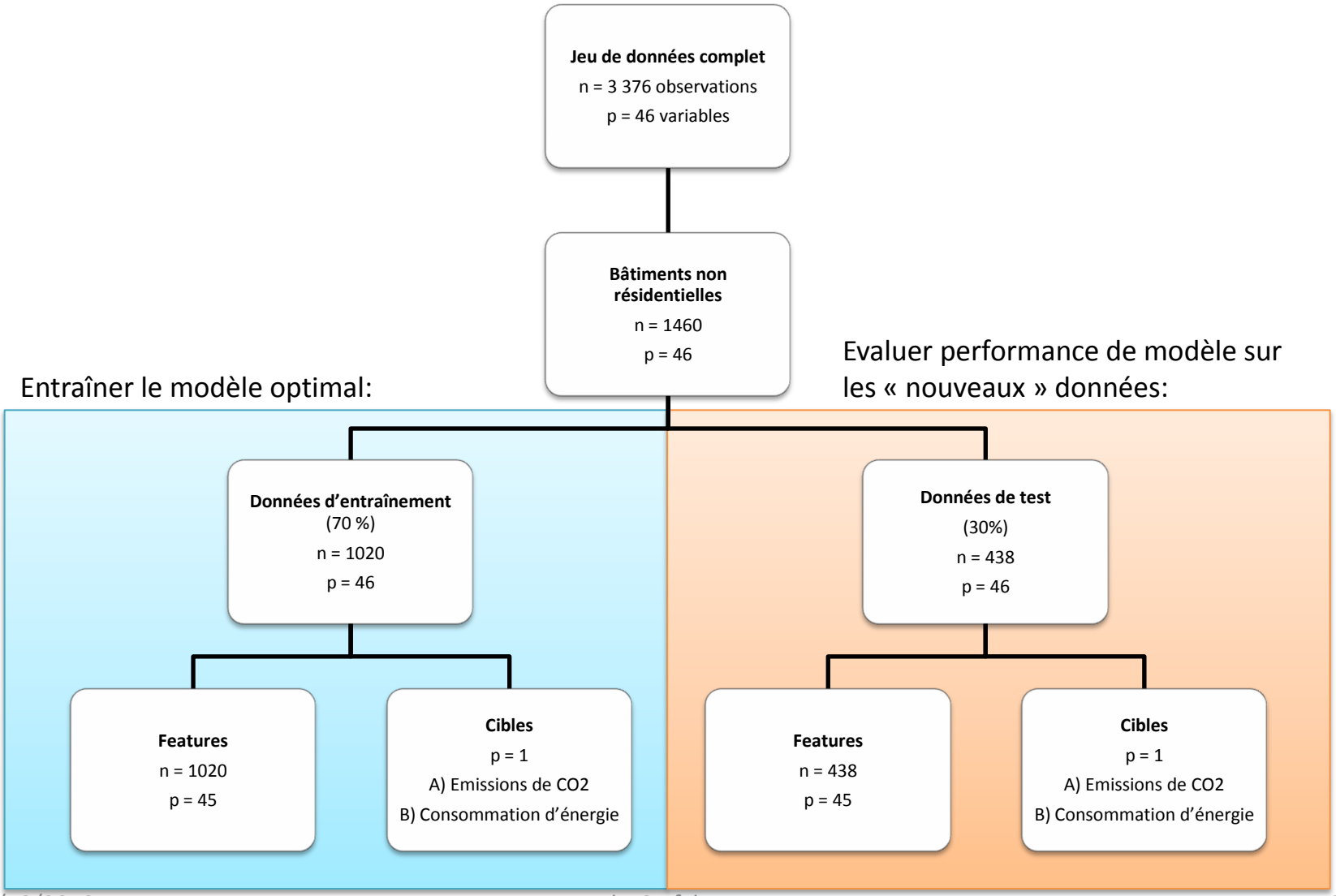
- 2 fichiers de données de 2015 et 2016
  - Données déclaratives du permis d'exploitation commerciale
  - Données relatives à la consommation d'énergie

- Après avoir comparé les deux fichiers, nous avons décidé de garder les données plus récentes

- Le fichier 2015 était utilisé pour imputer certaines valeurs manquantes dans le fichier 2016 (LargestPropertyUseType, LargestPropertyUseTypeGFA et ENERGYSTARScore)

1. Présentation de la problématique

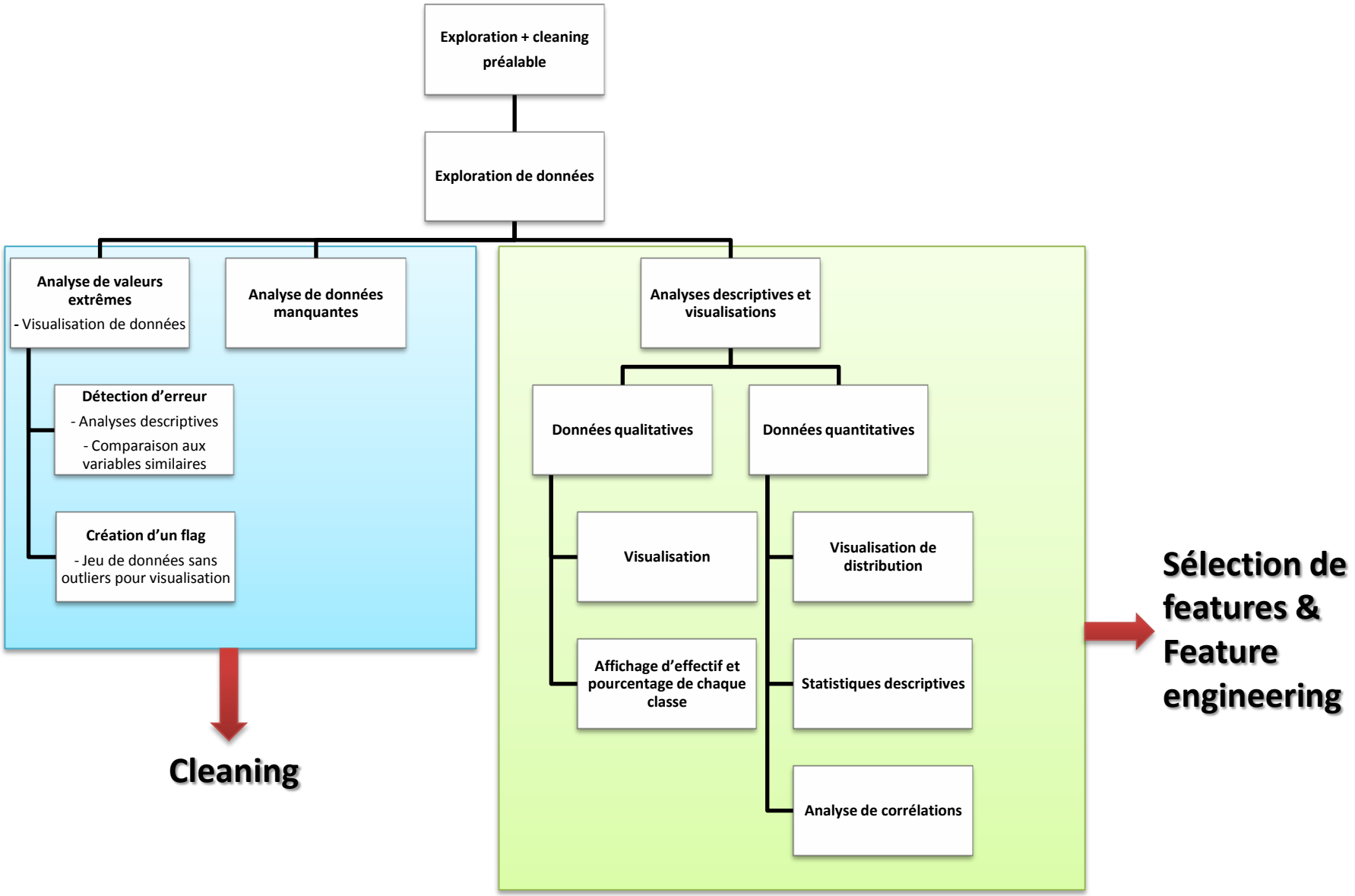
Data flow



2. ANALYSE EXPLORATOIRE

2. Analyse exploratoire

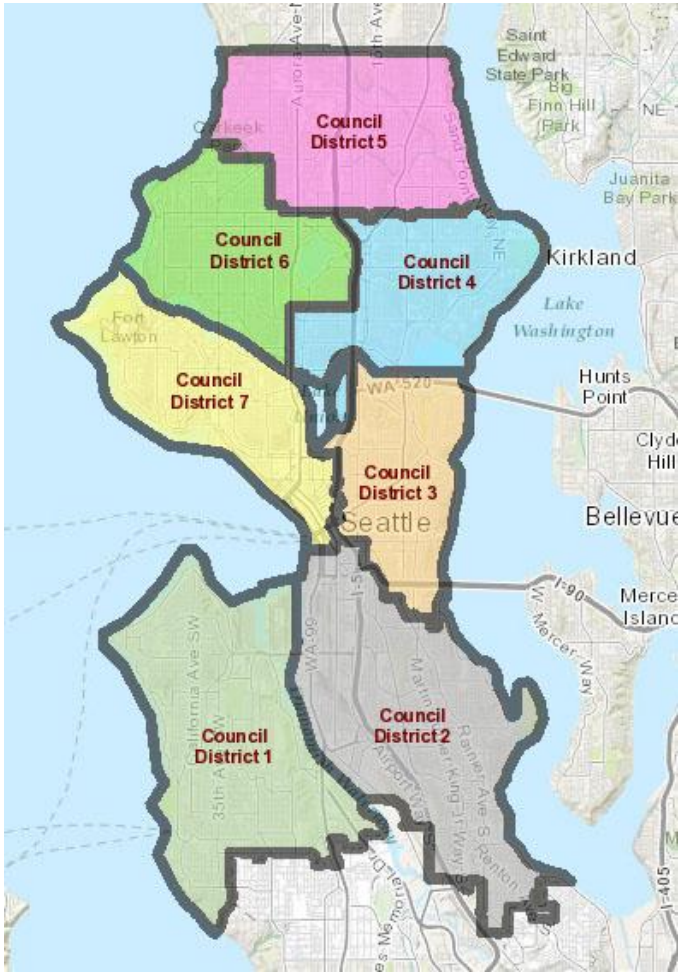
Workflow

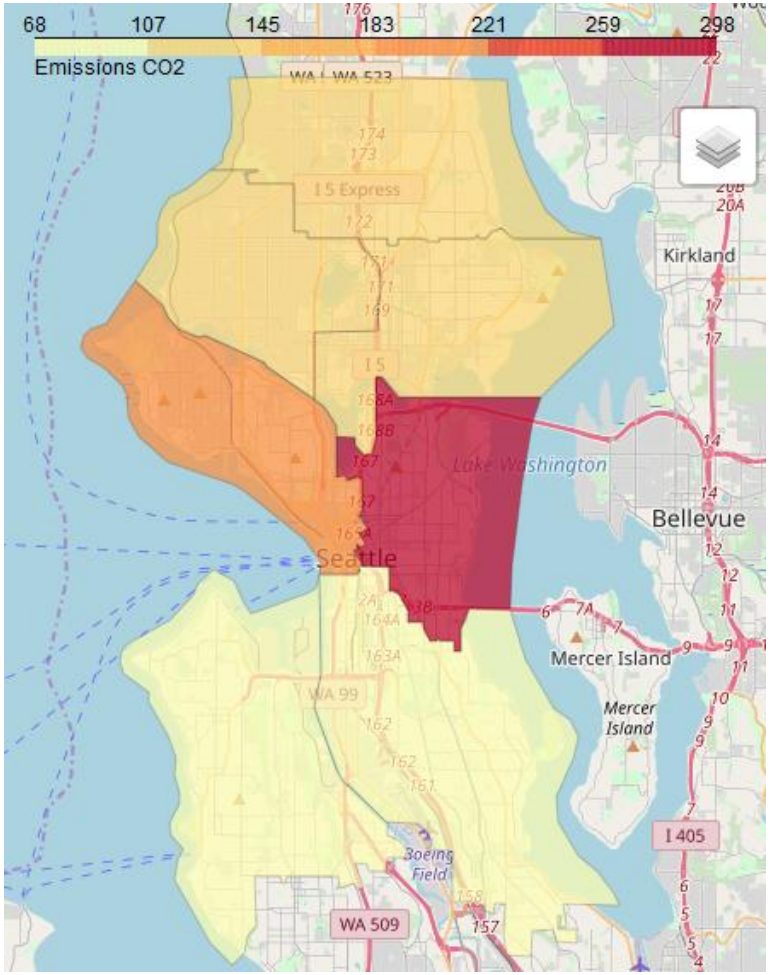


2. Analyse exploratoire
Exploration de données

Emissions de CO2 en fonction de district

- La figure à gauche représente la carte de districts
- La figure à droite représente une carte interactive construite avec le package *folium.Choropleth*

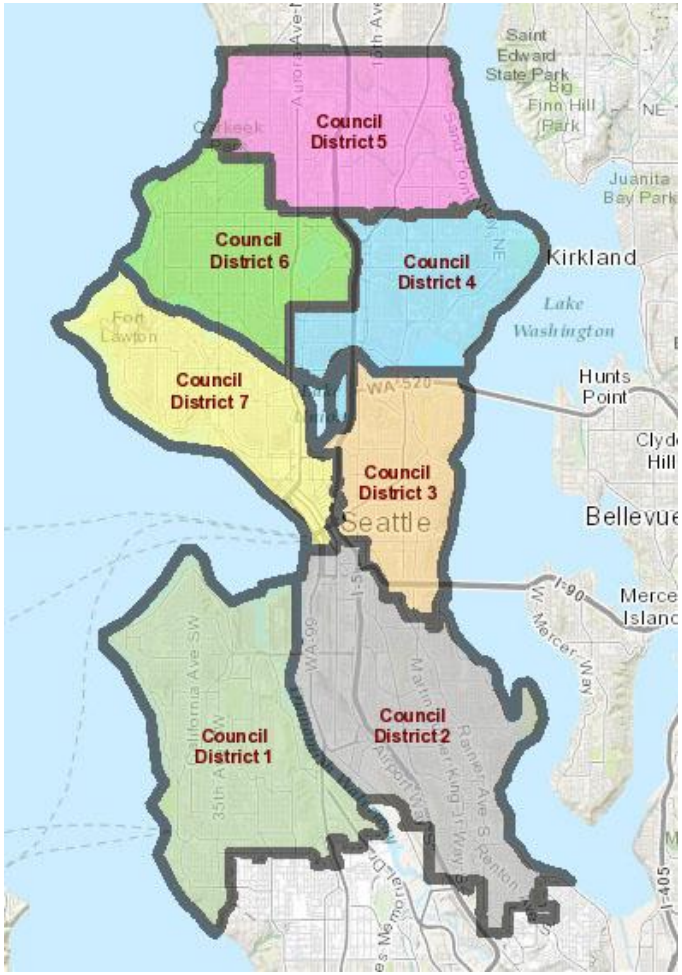


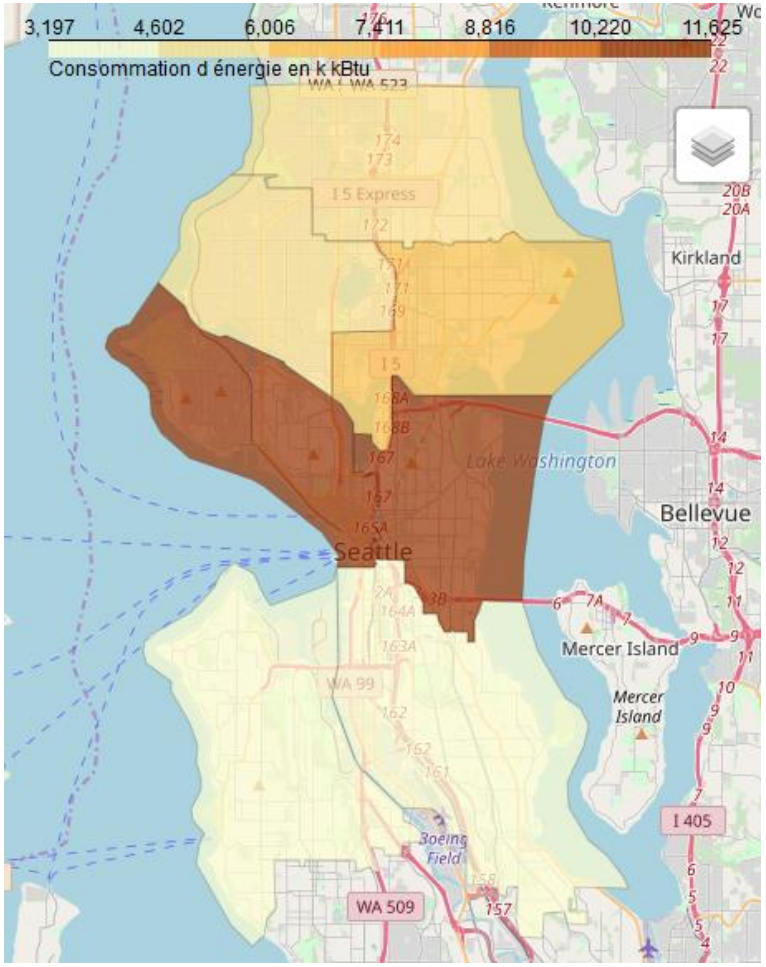


2. Analyse exploratoire
Exploration de données

Consommation d'énergie en fonction de district

- La figure à gauche représente la carte de districts
- La figure à droite représente une carte interactive construite avec le package *folium.Choropleth*

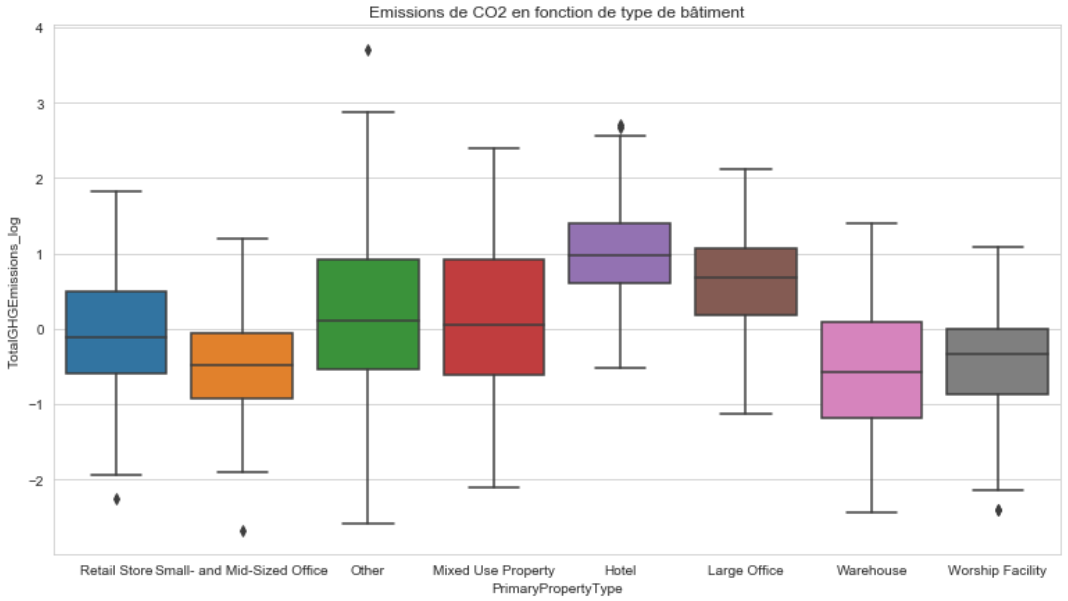


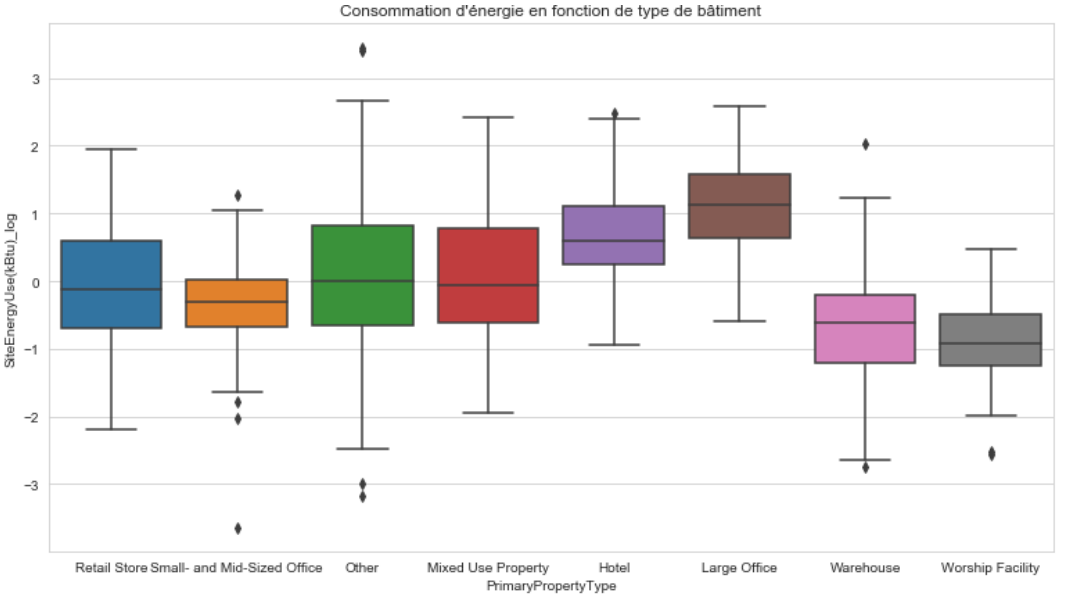


2. Analyse exploratoire
Exploration de données

Type d'utilisation principal (Primary Property Type)

- Initialement 22 classes
- Les classes avec < 5% d'effectifs recodées en 'Other' pour réduire le nombre de classes et enlever celles avec faible effectif
- Utile notamment lors de recodage de variable

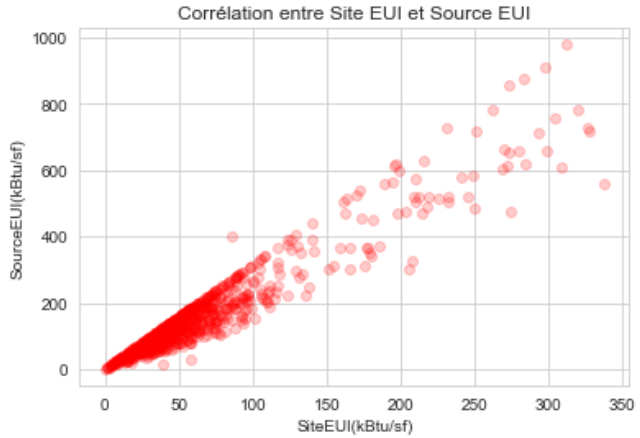


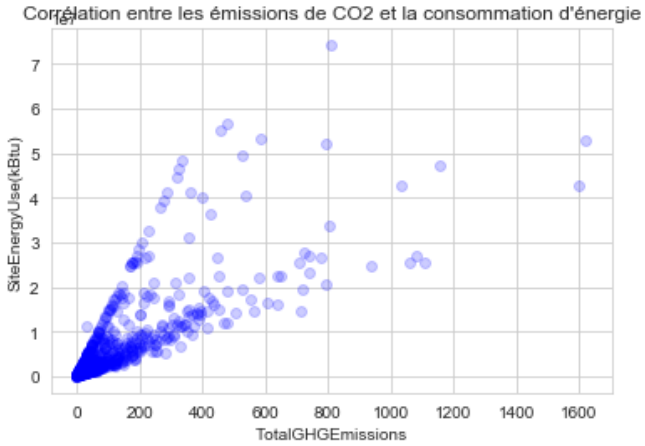


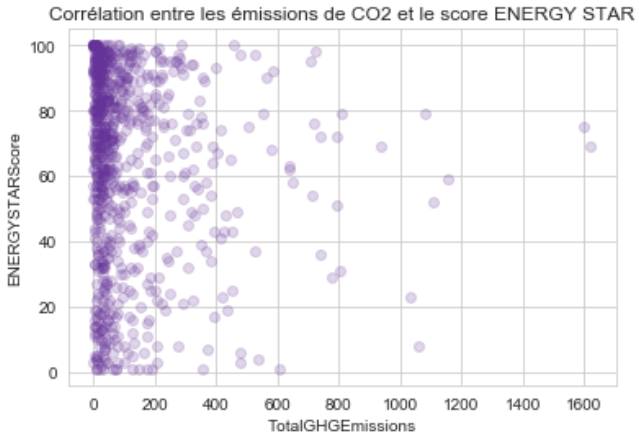
2. Analyse exploratoire
Exploration de données

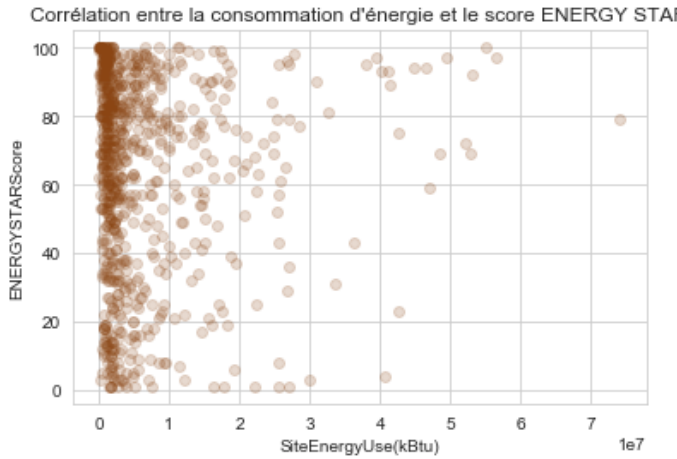
Analyse de corrélations

- Pour trouver s'il existe des lien entre les variables afin de pouvoir les introduire dans le même modèle
- Corrélations détectés entre :
  - Consommation d'énergie totale et les relevés de consommation
  - Emissions de CO2 et les données de relevés de consommation
  - Relation linéaire entre les différentes données représentant la surface de bâtiment

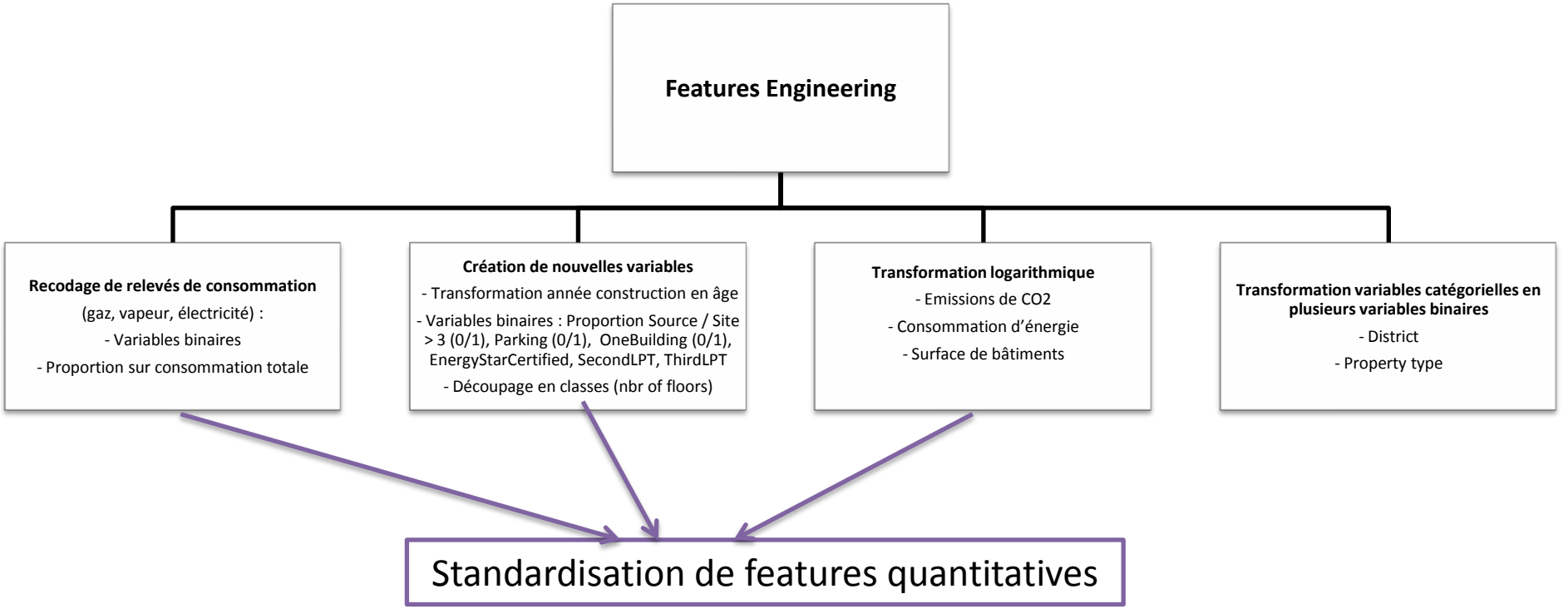








2. Analyse exploratoire
Features Engineering

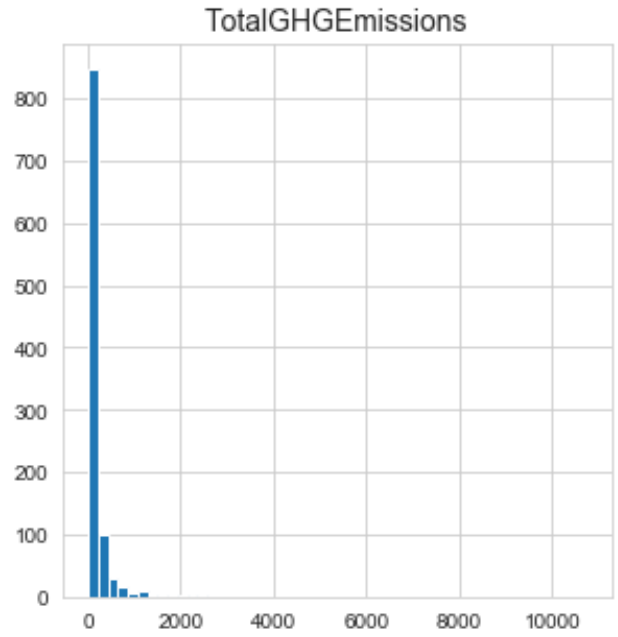


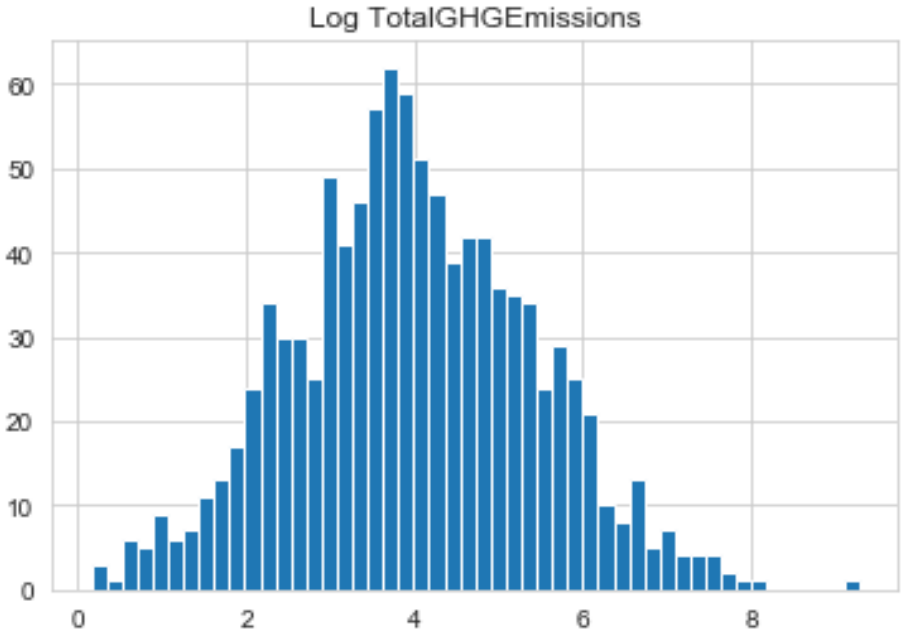
2. Analyse exploratoire
Transformation cible

Emissions de CO2 (TotalGHGEmissions)

- Distribution aplatie à gauche

=> Transformation en log afin de valider des hypothèses de certains modèles



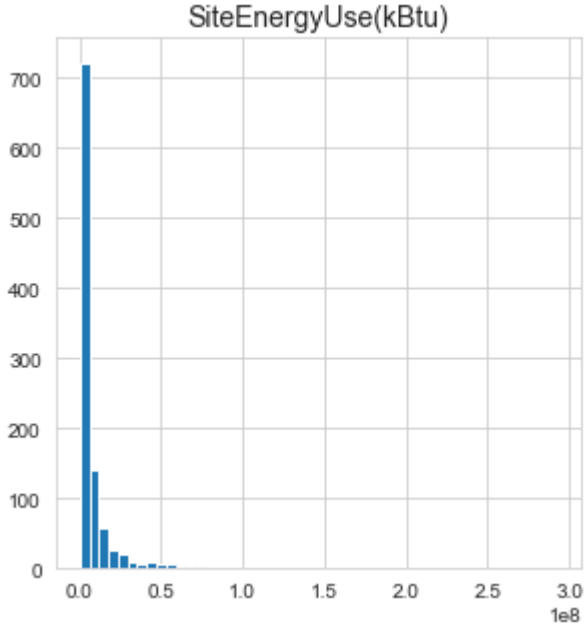


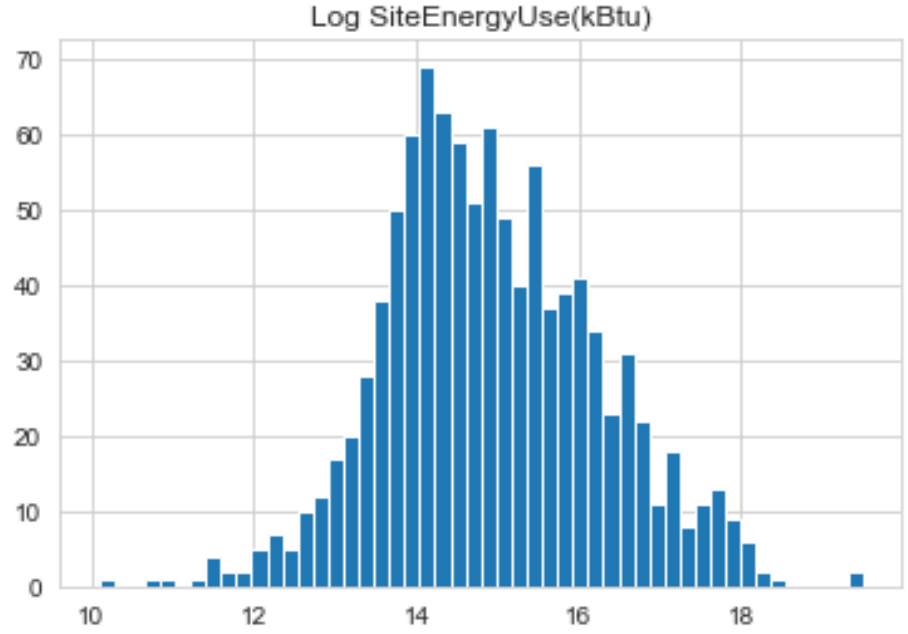
2. Analyse exploratoire
Transformation cible

Consommation d'énergie (SiteEnergyUse(kBtu))

- Aplatie vers la gauche

=> Nous allons également utiliser la transformation log



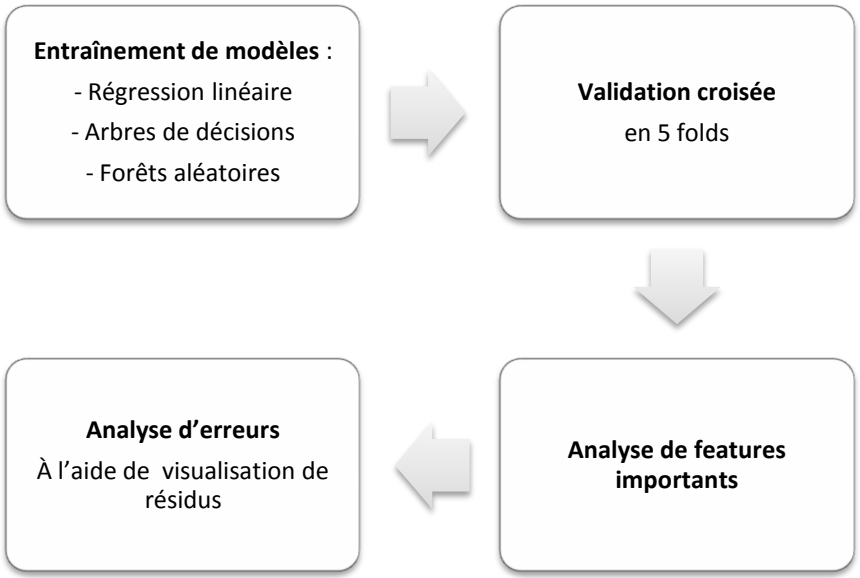


A – Emissions de CO2

B – Consommation d’énergie

3. MODÉLISATION

3. Modélisation
Schéma de recherche d'un modèle optimale



Nous allons choisir un modèle parmi les suivants :

- Régression linéaire
- Arbres de décision
- Forêts aléatoires

Critère de choix du modèle :
- RMSE moyen (validation croisée)
- Analyse d'erreurs

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

3. Modélisation

Recherche de modèles en pratique

Création d'une classe Python « Modelisation » qui prend en charge les trois modèles (Régression linéaire, Arbres de décision, Forêts aléatoires) et permet d'effectuer 4 opérations grâce aux méthodes suivantes :

- Obtenir le score RMSE

```
def getRMSE(self):
    if self.model == 'lin':
        print("Le modèle linéaire prédit les valeurs avec RMSE =", round(self.rmse, 4))
    elif self.model == 'tree':
        print("Le modèle arbres de décisions prédit les valeurs avec RMSE =", round(self.rmse, 4))
    else:
        print("Le modèle forêts aléatoires prédit les valeurs avec RMSE =", round(self.rmse, 4))
```

- Effectuer la validation croisée

```
def cross_valid(self, cv):
    self.cv=cv
    scores = cross_val_score(self.mod, self.X, self.y, scoring="neg_mean_squared_error", cv=self.cv)
    rmse_scores = np.sqrt(-scores)
    print("Scores:", rmse_scores)
    print("Mean:", rmse_scores.mean())
    print("Standard deviation:", rmse_scores.std())
```

- Evaluer l'importance de features

```
def features_importance(self):
    if self.model == 'lin':
        return sorted(zip(self.mod.coef_, self.X.columns), reverse=True)
    else:
        return sorted(zip(self.mod.feature_importances_, self.X.columns), reverse=True)
```

- Tracer le graphique de résidus

```
def scatter_residuals(self, alpha=0.2, xmin=-4, xmax=4, n_estimators=100):
    self.alpha=alpha
    self.xmin=xmin
    self.xmax=xmax
    self.n_estimators=n_estimators

    # Scatter plot the training data
    train = plt.scatter(self.pred,(self.y-self.pred),c='b',alpha=0.2)

    # Plot a horizontal axis line at 0
    plt.hlines(y=0,xmin=-4,xmax=4)

    #Labels
    plt.title('Residual Plots')
    plt.xlabel('Prédictions')
    plt.ylabel('Résidus')
```

3. Modélisation
A) Prédiction d'émissions de CO2

RMSE de modèle simple
- Régression linéaire: RMSE = 0.114
- Arbres de décision: RMSE = 0.0
- Forêts aléatoires (110 arbres): RMSE = 0.0517

RMSE moyen de validation croisée en 5 folds
- Régression linéaire: RMSE = 0.1154
- Arbres de décision: RMSE = 0.2038
- Forêts aléatoire (110 arbres): RMSE = 0.1504

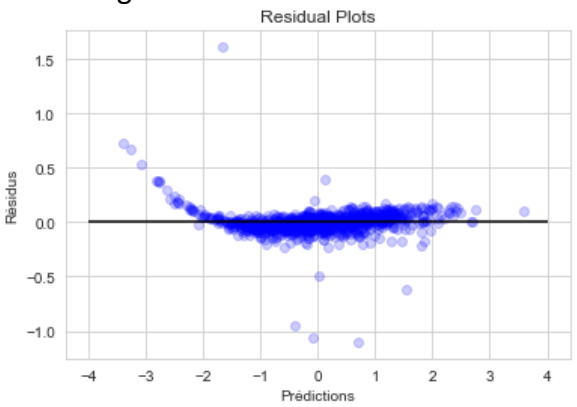
Les arbres de décision ont tendance de surapprentissage (RMSE de modèle simple << RMSE de validation croisée. La même tendance chez les forêts aléatoires.

Features les plus importantes

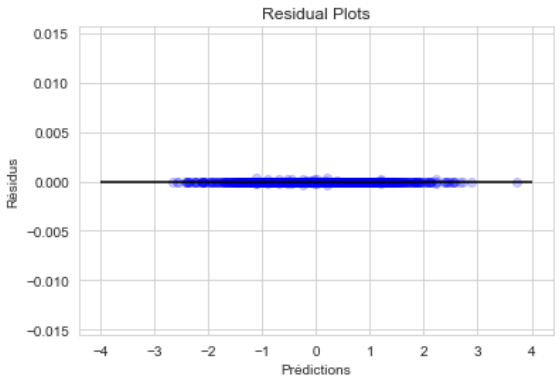
	Régression linéaire		Arbres de décision		Forêts aléatoires	
Rang	Feature	Score	Feature	Score	Feature	Score
1	SiteEnergyUse(kBtu)_log	0.8339	SiteEnergyUse(kBtu)_log	0.7584	SiteEnergyUse(kBtu)_log	0.7805
2	NaturalGas(kBtu)_pct	0.3289	RatioSourceSite_sup3	0.1868	RatioSourceSite_sup3	0.1246
3	SteamUse(kBtu)_pct	0.1431	NaturalGas(kBtu)_pct	0.0333	NaturalGas(kBtu)_pct	0.0728
4	SteamUse	0.1279	SteamUse(kBtu)_pct	0.0097	SteamUse(kBtu)_pct	0.0092
5	PrimaryPropertyType_Hotel	0.0563	ENERGYSTARScore	0.0044	ENERGYSTARScore	0.0029

Analyse de résidus

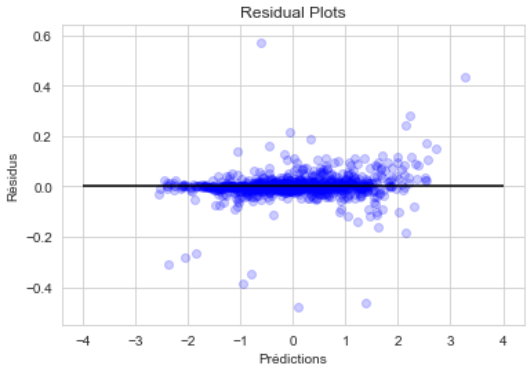
Régression linéaire



Arbres de décision



Forêt aléatoire



3. Modélisation

A) Prédiction d'émissions de CO2 – modèle retenu

RMSE de modèle simple

- Régression linéaire: RMSE = 0.114
- Arbres de décision: RMSE = 0.0
- Forêts aléatoires (110 arbres): RMSE = 0.0517

RMSE moyen de validation croisée en 5 folds

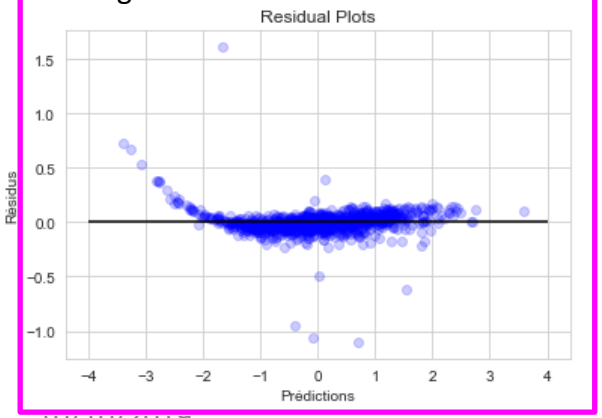
- Régression linéaire: RMSE = 0.1154
- Arbres de décision: RMSE = 0.2038
- Forêts aléatoire (110 arbres): RMSE = 0.1504

Features les plus importantes

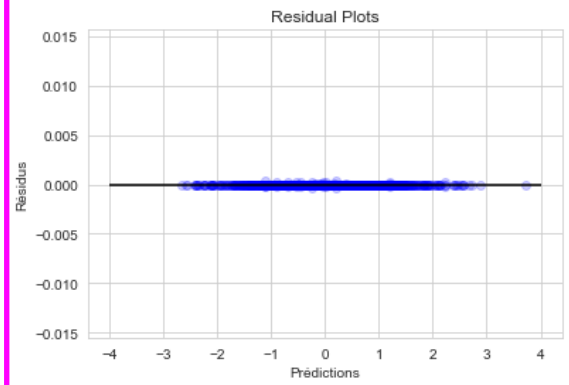
	Régression linéaire		Arbres de décision		Forêts aléatoires	
Rang	Feature	Score	Feature	Score	Feature	Score
1	SiteEnergyUse(kBtu)_log	0.8339	SiteEnergyUse(kBtu)_log	0.7584	SiteEnergyUse(kBtu)_log	0.7805
2	NaturalGas(kBtu)_pct	0.3289	RatioSourceSite_sup3	0.1868	RatioSourceSite_sup3	0.1246
3	SteamUse(kBtu)_pct	0.1431	NaturalGas(kBtu)_pct	0.0333	NaturalGas(kBtu)_pct	0.0728
4	SteamUse	0.1279	SteamUse(kBtu)_pct	0.0097	SteamUse(kBtu)_pct	0.0092
5	PrimaryPropertyType_Hotel	0.0563	ENERGYSTARScore	0.0044	ENERGYSTARScore	0.0029

Analyse de résidus

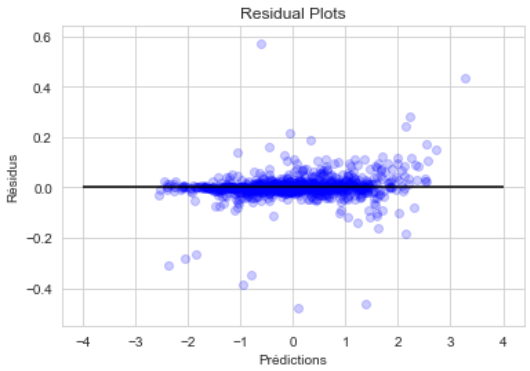
Régression linéaire



Arbres de décision



Forêt aléatoire



3. Modélisation
B) Prédiction de consommation d'énergie

RMSE de modèle simple
- Régression linéaire: RMSE = 0.132
- Arbres de décision: RMSE = 0.0
- Forêts aléatoires (110 arbres): RMSE = 0.0568

RMSE moyen de validation croisée en 5 folds
- Régression linéaire: RMSE = 0.1339
- Arbres de décision: RMSE = 0.2154
- Forêts aléatoire (110 arbres): RMSE = 0.1585

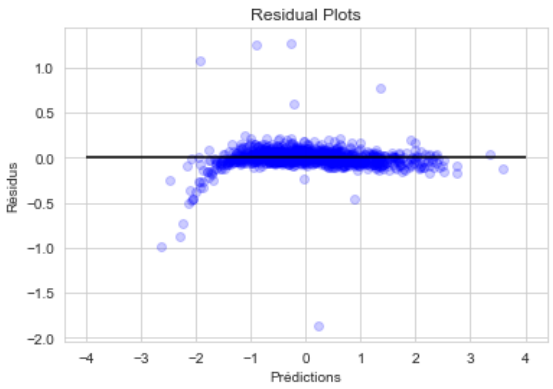
Les arbres de décision ont tendance de surapprentissage (RMSE de modèle simple << RMSE de validation croisée. La même tendance chez les forêts aléatoires.

Features les plus importantes

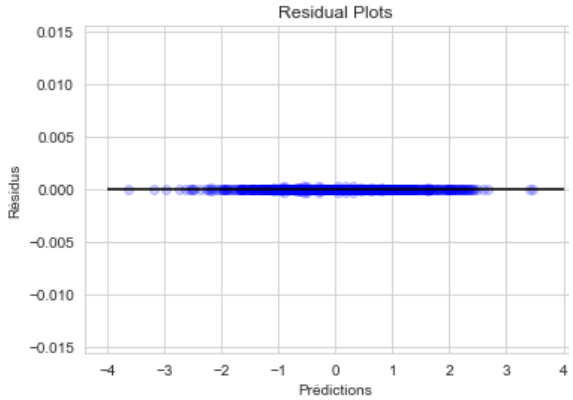
	Régression linéaire		Arbres de décision		Forêts aléatoires	
Rang	Feature	Score	Feature	Score	Feature	Score
1	TotalGHGEmissions_log	1.1175	TotalGHGEmissions_log	0.8607	TotalGHGEmissions_log	0.8407
2	RatioSourceSite_sup3	0.2893	NaturalGas(kBtu)_pct	0.0711	NaturalGas(kBtu)_pct	0.0699
3	CouncilDistrictCode_2	0.0450	RatioSourceSite_sup3	0.0470	LargestPropertyUseTypeGFA_log	0.0370
4	PrimaryPropertyType_Large Office	0.0416	SteamUse(kBtu)_pct	0.0072	RatioSourceSite_sup3	0.0337
5	CouncilDistrictCode_4	0.0317	LargestPropertyUseTypeGFA_log	0.0046	SteamUse(kBtu)_pct	0.0049

Analyse de résidus

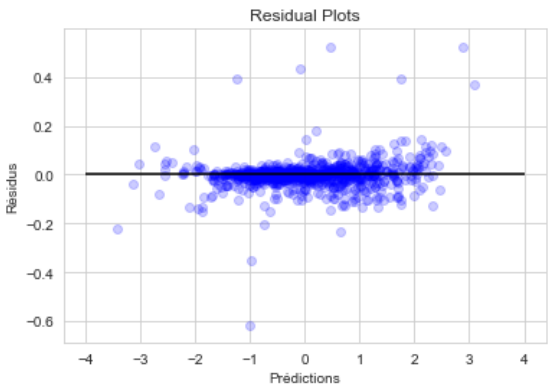
Régression linéaire



Arbres de décision



Forêt aléatoire



3. Modélisation

B) Prédiction de consommation d'énergie – modèle retenu

La différence entre RMSE de la régression linéaire et les forêts aléatoires n'est pas très grande. Pour des raisons pédagogiques nous allons choisir les forêts aléatoires avec une validation croisée pour éviter le surapprentissage.

RMSE de modèle simple

- Régression linéaire: RMSE = 0.132
- Arbres de décision: RMSE = 0.0
- Forêts aléatoires (110 arbres): RMSE = 0.0568

RMSE moyen de validation croisée en 5 folds

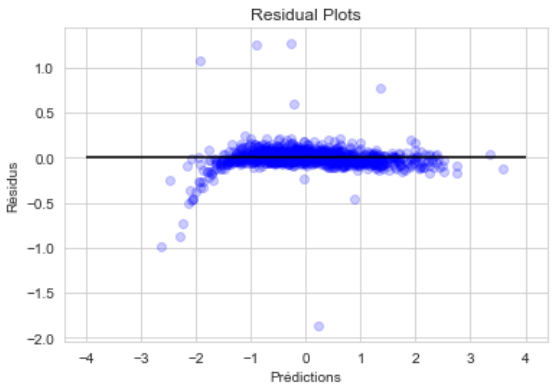
- Régression linéaire: RMSE = 0.1339
- Arbres de décision: RMSE = 0.2154
- Forêts aléatoire (110 arbres): RMSE = 0.1585

Features les plus importantes

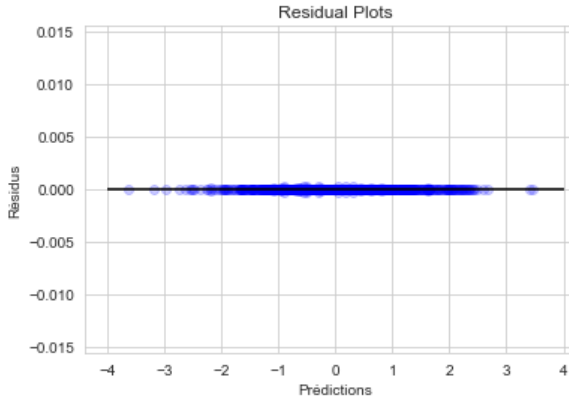
	Régression linéaire		Arbres de décision		Forêts aléatoires	
Rang	Feature	Score	Feature	Score	Feature	Score
1	TotalGHGEmissions_log	1.1175	TotalGHGEmissions_log	0.8607	TotalGHGEmissions_log	0.8407
2	RatioSourceSite_sup3	0.2893	NaturalGas(kBtu)_pct	0.0711	NaturalGas(kBtu)_pct	0.0699
3	CouncilDistrictCode_2	0.0450	RatioSourceSite_sup3	0.0470	LargestPropertyUseTypeGFA_log	0.0370
4	PrimaryPropertyType_Large Office	0.0416	SteamUse(kBtu)_pct	0.0072	RatioSourceSite_sup3	0.0337
5	CouncilDistrictCode_4	0.0317	LargestPropertyUseTypeGFA_log	0.0046	SteamUse(kBtu)_pct	0.0049

Analyse de résidus

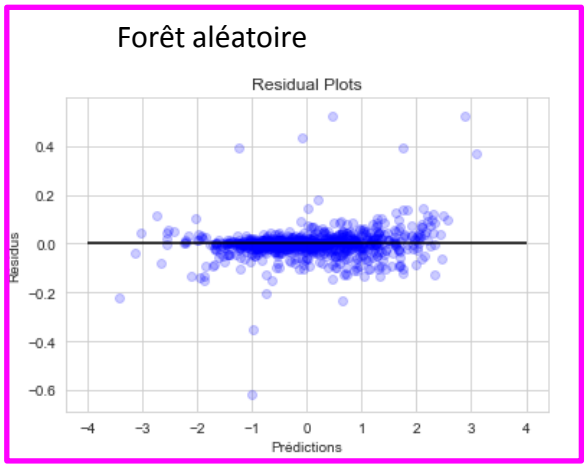
Régression linéaire



Arbres de décision



Forêt aléatoire



A – Emissions de CO2

B – Consommation d’énergie

4. FINALISATION DE MODÈLES

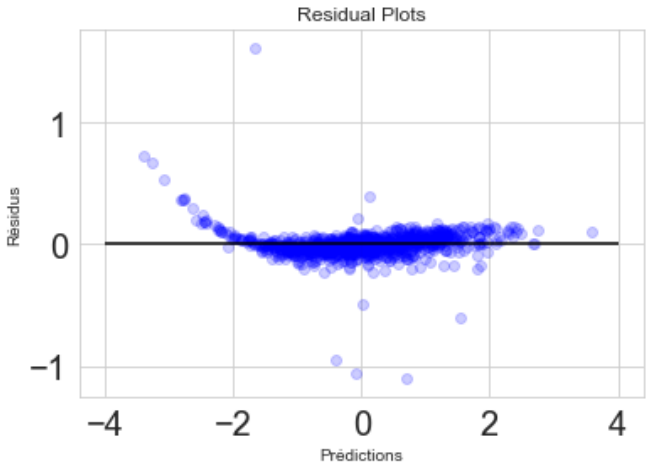
4. Finalisation de modèles : **A) Prédiction d'émissions de CO2 Ajustement de paramètres**

Le choix de modèle sera fait parmi les modèles suivants:

- **Régression linéaire**
  - Ridge : pour éviter le surapprentissage en restreignant l'amplitude des poids
  - Lasso : pour obtenir un modèle parcimonieux
  - ElasticNet : pour combiner les deux normes l1 et l2 utilisés dans Ridge et Lasso

Nous utilisons les modules de scikit-learn qui intègrent la validation croisée avec une option de 5 folds (cv=5)

	MSE	RMSE	R2
Régression Ridge	0.01300	0.11402	0.98700
Régression Lasso	0.01319	0.11485	0.98681
Régression ElasticNet	0.01319	0.11485	0.98681



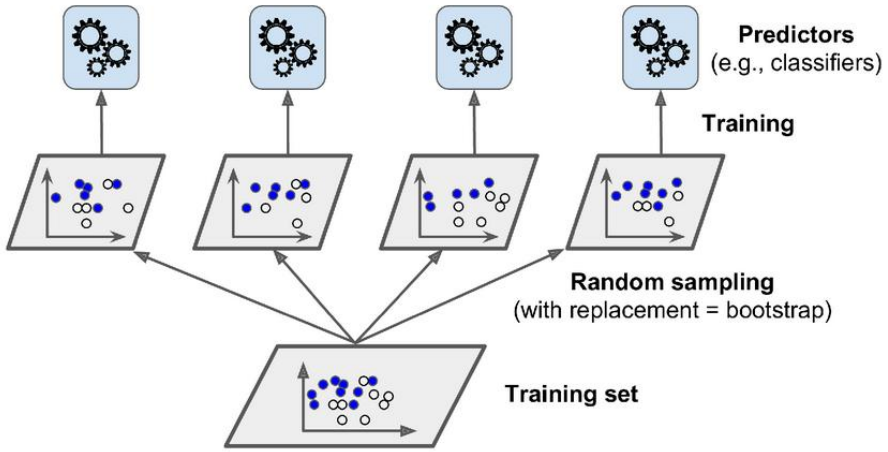
Le meilleur modèle obtenu est celui de régression Ridge avec RMSE = 0.1140 et R2 = 98.70 %

4. Finalisation de modèles : **A) Prédiction d'émissions de CO2 Méthodes ensemblistes**

Utilisation de bagging:

- Méthode ensembliste, qui consiste à entraîner le modèle plusieurs fois sur des échantillons aléatoires de notre jeu de données tirés avec remise
- Le but est notamment de réduire la variance des estimateurs individuels et obtenir une prédiction plus performante et plus stable
- Entraînement de 500 modèles Ridge sur des échantillons de 100 observations tirés avec remise
- Coefficient Out-of-bag : calculé sur les données qui ne rentrent pas dans le modèle => qui font office de données test
- Out-of-bag de notre modèle = 98.58 %

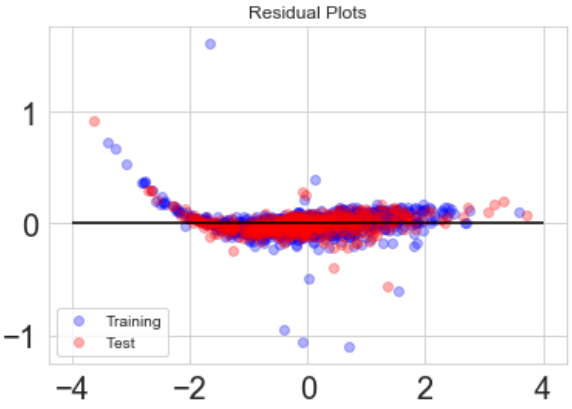
=> Le modèle va probablement prédire des données test avec 98 % de précision



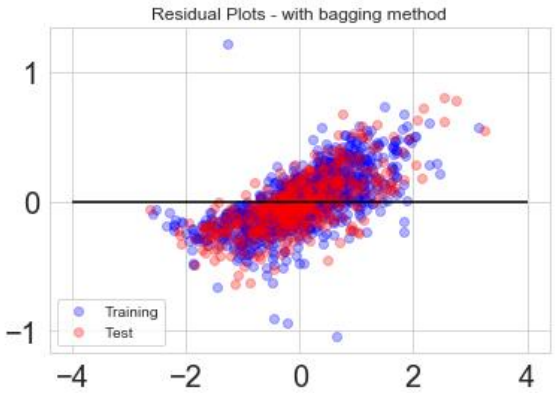
Source : A. Géron - Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow

4. Finalisation de modèles : **A) Prédiction d'émissions de CO2 Evaluation de modèle final**

- Scores pour prédiction de données test :
  - MSE = 0.00868
  - RMSE = 0.09317
  - R2 = 99.20 %



- En utilisant le bagging :
  - MSE = 0.0508
  - RMSE = 0.2253
  - R2 = 95.35 %



- En utilisant le bagging, R2 diminue de 99.2 % à 95.35 %. Nous perdons la précision, mais le modèle est plus robuste et overfit moins les données
- Nous observons une tendance dans les résidus :

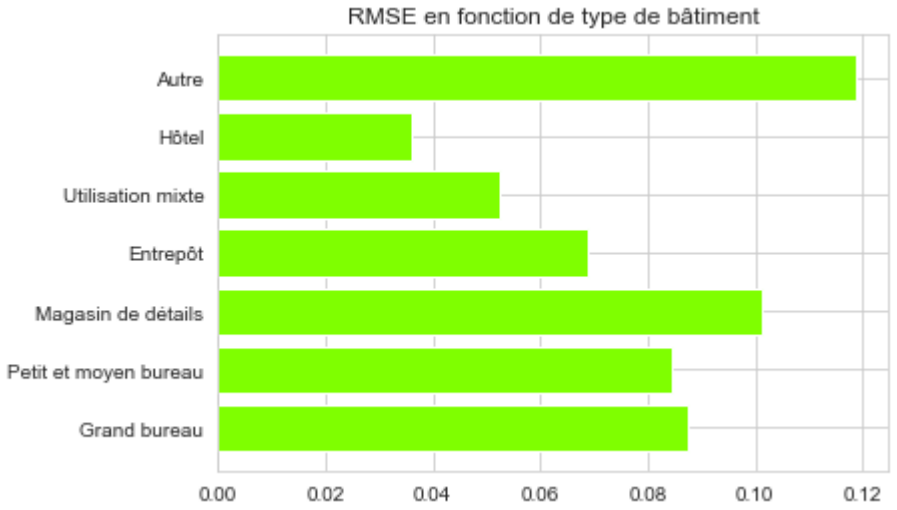
  -Introduction de modèle polynomiale ? Difficile, car nous n'avons pas assez de données

  - Utiliser une régression Ridge à noyau ? Piste à explorer…

4. Finalisation de modèles : **A) Prédiction d'émissions de CO2 Prédictions en fonction de type d'immeuble**

- Nous avons séparé les données en fonction de type de bâtiment et évalué le modèle de régression Ridge pour chaque type séparément

	MSE	RMSE	R2
Grand bureau	0.00763	0.08735	0.97922
Petit et moyen bureau	0.00715	0.08456	0.98307
Magasin de détails	0.01025	0.10124	0.98544
Entrepôt	0.00472	0.06870	0.99374
Utilisation mixte	0.00274	0.05235	0.99709
Hôtel	0.00130	0.03606	0.99872
Autre	0.01413	0.11887	0.98982



- Le modèle prédit mieux les émissions pour les hôtels et utilisation mixte, que pour les bâtiments type bureau, magasin ou autre.

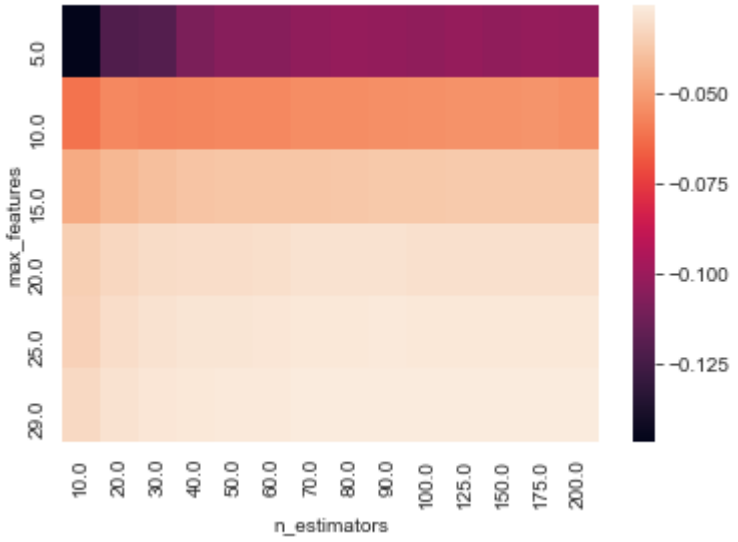
4. Finalisation de modèles : **B) Prédiction de consommation d'énergie Ajustement de paramètres**

La recherche des paramètres optimales du modèle Random Forest était fait avec une recherche sur grille :

1. Avec des différents paramètres en utilisant le bootstrap ou non, comme illustre l'extrait de code :

```
from sklearn.model_selection import GridSearchCV

param_grid = [
    {'n_estimators': [20, 30, 40, 50, 100, 150], 'max_features': [20,25,29]},
    {'bootstrap': [False], 'n_estimators': [20, 30, 40, 50, 100, 150], 'max_features': [20, 25, 29]},
  ]

forest_reg = RandomForestRegressor(random_state = 42)

grid_search = GridSearchCV(forest_reg, param_grid, cv=5,
                           scoring='neg_mean_squared_error',
                           return_train_score=True)

grid_search.fit(X_train_B, y_train_B)
```

2. Etant donné que la procédure avec bootstrap donne globalement meilleurs résultats, nous avons élargi le nombre de coefficients pour la recherche sur grille avec bootstrap et refait la procédure de recherche sur grille



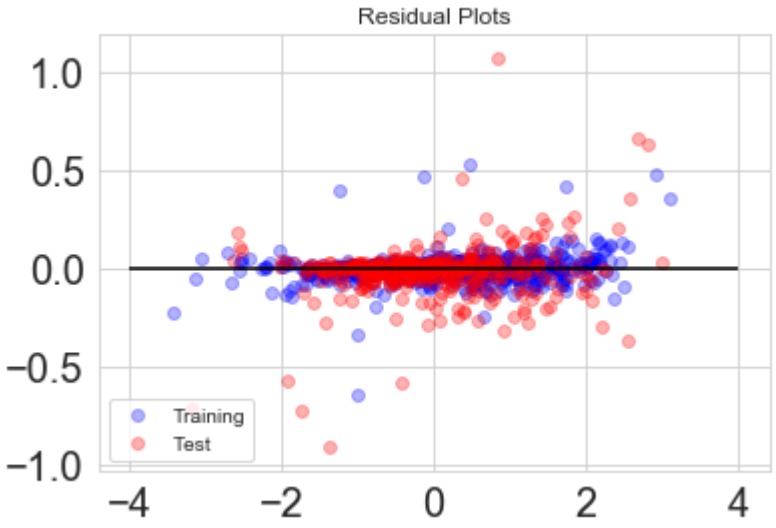
Les meilleures résultats sont obtenus à partir de 20 features qui rentrent dans le modèle

Etant donné que nous avons peu d'observations, nous pouvons opter pour les meilleurs paramètres sans être obligé à optimiser le temps de calcul

Les meilleurs résultats :
Max_features = 29
N_estimators = 200

4. Finalisation de modèles : **B) Prédiction de consommation d'énergie Evaluation de modèle final**

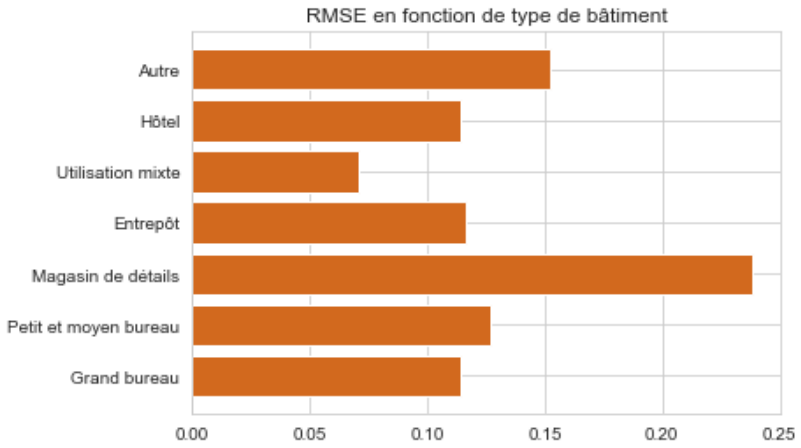
- Scores pour prédiction de données test :
  - MSE = 0.0181
  - RMSE = 0.1344
  - R2 = 98.23 %



4. Finalisation de modèles : **B) Prédiction de consommation d'énergie Prédictions en fonction de type d'immeuble**

- Nous avons séparé mes données en fonction de type de bâtiment et évalué le modèle de forêts aléatoires pour chaque type de bâtiment

	MSE	RMSE	R2
Grand bureau	0.01304	0.11419	0.95653
Petit et moyen bureau	0.01601	0.12653	0.94867
Magasin de détails	0.05686	0.23845	0.91553
Entrepôt	0.01355	0.11640	0.97361
Utilisation mixte	0.00503	0.07092	0.99338
Hôtel	0.01298	0.11393	0.98011
Autre	0.02326	0.15251	0.98283



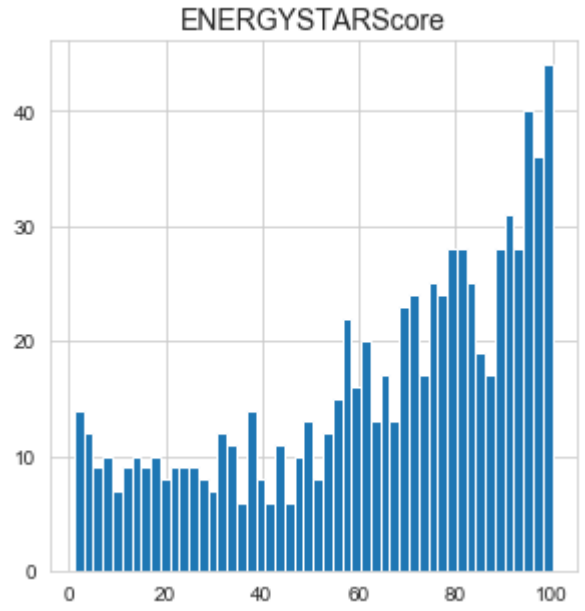
- Le modèle est moins performant pour prédire la consommation de bâtiments de type magasins de détail.
- La meilleure performance est atteint s'il s'agit de bâtiments avec une utilisation mixte

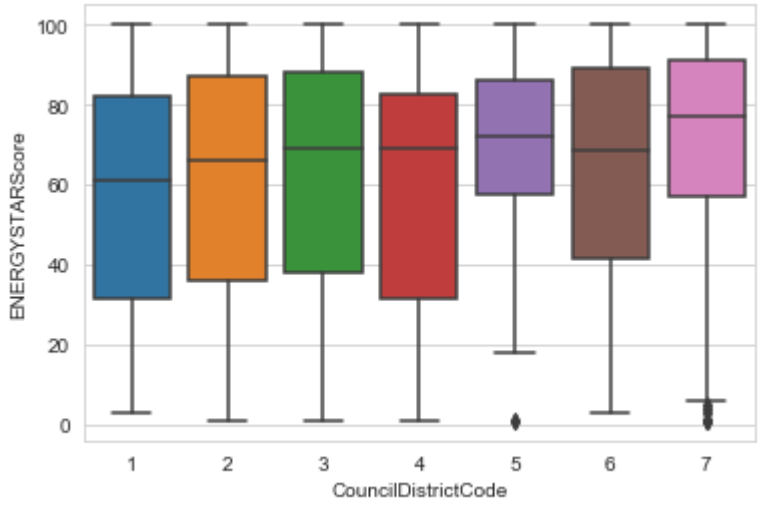
5. EVALUER L'INTÉRÊT D'ENERGY STAR SCORE POUR LA PRÉDICTION D'EMISSIONS DE CO2

2. Analyse exploratoire
Exploration de données

ENERGY STAR Score (ENERGYSTARScore)

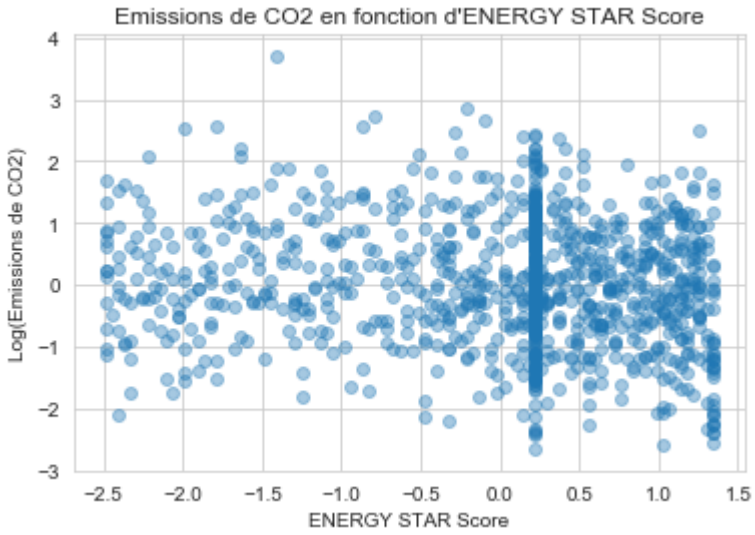
- Score en 0 et 100
- Les bâtiments avec Score 100 ont la meilleure performance énergétique
- 22% de valeurs manquantes imputées par la médiane





5. Evaluer l'intérêt d'energy star score pour la prédiction d'emissions de CO2

- Visualisation de données : Corrélation entre les deux variables



5. Evaluer l'intérêt d'energy star score pour la prédiction d'emissions de CO2

Evaluation d'importance de feature avec forêts aléatoires :

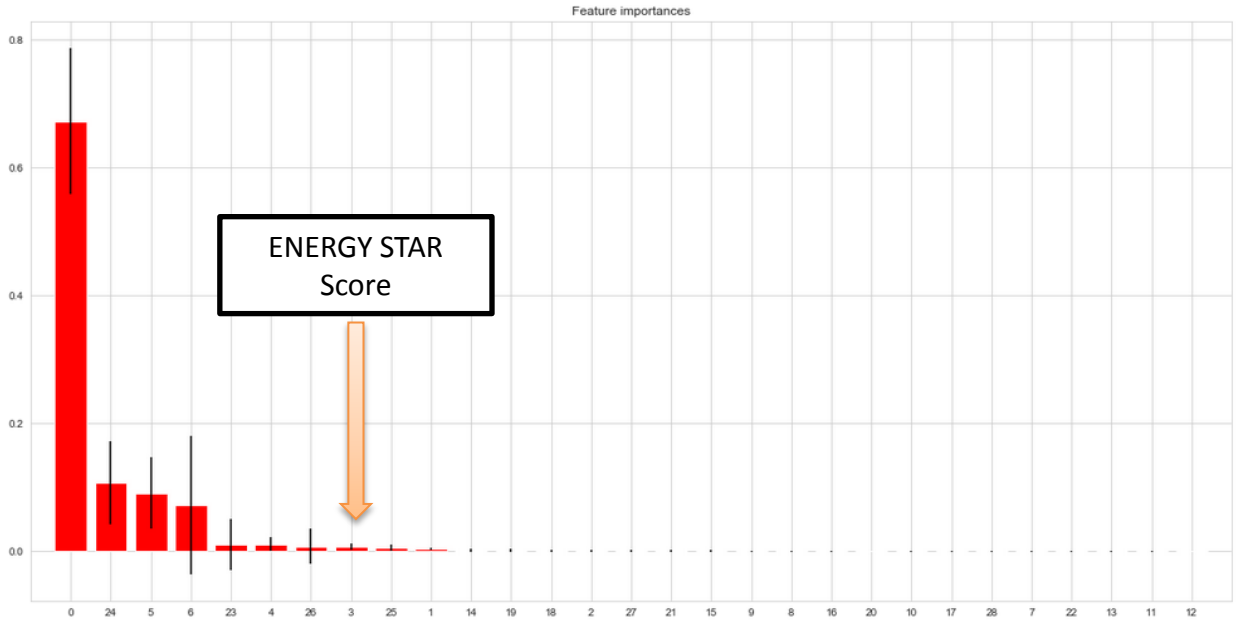
- Recherche d'un modèle optimale : recherche sur grille

⇒ meilleurs paramètres :

  - Max_features = 20
  - N_estimators = 125

- Construction de modèle
  - RMSE = 0.1226

```
Entrée [152]: sorted(zip(forest_A.feature_importances_, X_train_A.columns), reverse=True)

Out[152]: [(0.672996560698921, 'SiteEnergyUse(kBtu)_log'),
           (0.10784340338780601, 'RatioSourceSite_sup3'),
           (0.09151113649571363, 'NaturalGas(kBtu)_pct'),
           (0.07211511168226664, 'LargestPropertyUseTypeGFA_log'),
           (0.011404766924942916, 'NumberofFloors_cat'),
           (0.010922894908352071, 'SteamUse(kBtu)_pct'),
           (0.008236320224683126, 'NaturalGas'),
           (0.0071679749280832, 'ENERGYSTARScore'),
```



6. CONCLUSION

6. Conclusion

- **Prédiction d'émissions de CO2**
  - Le meilleur modèle trouvé:
    - Régression Ridge avec bagging
  - Performance sur les données test:
    - RMSE = 0.2253
    - R2 = 95.35 %
  - Possibilité d'améliorer le modèle en explorant l'introduction d'une régression non-linéaire à noyau (régression polynomiale pas possible car peu d'observations)

- **Prédiction de consommation d'énergie**
  - Le meilleur modèle trouvé :
    - Forêts aléatoires avec 200 arbres et toutes les features
  - Performance sur les données test:
    - RMSE = 0.1344
    - R2 = 98.23 %

- **Evaluation d'intérêt d'ENERGY STAR Score dans les prédictions d'émissions:**
  - Aucun lien apparent en visualisant les émissions en fonction d'ENERGY STAR Score
  - Analyse d'importance de features avec forêt aléatoire :
    - 8ème feature la plus importante / 29
    - La représentation graphique montre une faible importance par rapport aux 4 premières features
    - ⇒ L'importance d'ENERGY STAR Score pour les prédictions d'émissions est assez faible
    - ⇒ Recommandation : utiliser un autre score composé de consommation totale d'énergie, ratio source / site, surface de bâtiment et proportion d'utilisation de gaz / vapeur sur la consommation totale