Université Claude Bernard Lyon 1

PROJET

MODELE DE REGRESSION

Année universitaire: 2019 - 2020

Etudiant 1: NGUYEN Mai Ngoc Linh

Etudiant 2: HOA Minh Luan Hoa

Tuteur pédagogique : François WAHL

TABLE DES MATIÈRES

INTRODUCTION

On étudie un jeu de données issu de données industrielle provenant de diverses raffineries dans le monde.

Les raffineries mettent en œuvre des procédés (de raffinage) qui ont pour objectif de transformer une entrée (une charge pétrolière) en sorties (ou effluents) plus aptes à être valorisées.

On souhaite prédire certaines propriétés des sorties d'un procédé donné en fonction des entrées.

Pour ce faire, on va modéliser la sortie à l'aide d'un modèle linéaire. Ce modèle va être construit site par site et globalement (tous les sites confondus).

On travaille en langage de programmation Python à l'aide des bibliothèques Numpy, Pandas, Sklearn pour la prépration et la modélisation, Pyplot pour la visualisation.

1. JEU DE DONNÉES

Les données ont déjà été partiellement prétraitées mais ils restent quelques étapes à faire avant de la construction d'un modèle linéaire.

En entrée, on dispose des informations suivantes :

- "**FEED_D154**" : Densité de la charge
- "**FEED_SULFUR**" : Soufre (de la charge)
- "**FEED_NITROGEN**" : Azote de la charge
- "**FEED_MeABP**" : Mean Average Boiling Point de la charge
- "**FEED_KWATSON**" : Kwatson (c'est un index utilisé par les raffineurs)
- "**FEED_DS_005**", "**FEED_DS_010**", "**FEED_DS_030**", "**FEED_DS_050",** "**FEED_DS_070**", "**FEED_DS_090**", "**FEED_DS_095**" : Distillation de la charge (c'est un ensemble de points qui relient le pourcentage distillé à la température nécessaire pour obtenir le pourcentage en question)
- "**DIESEL_DS_005**", "**DIESEL_DS_010**", "**DIESEL_DS_030**", "**DIESEL_DS_050**", "**DIESEL_DS_070**", "**DIESEL_DS_090**", "**DIESEL_DS_095**" : Distillation de la coupe GO (Gas-Oil)
- "**TMP**" : c'est une donnée sur la charge exploitée par les raffineurs
- "**X370PLUS**"
- "**Proc1**" : donnée procédé
- "**Proc2**" : donnée procédé
- "**Proc3**" : donnée procédé
- "**actual.values**" : la variable de sortie
- "**PLANT_NAME**" : numéro de site
- "**CYCLE**" : numéro du cycle
- "**DATE**" : dates des observations

Ce jeu de données contient 8000 observations et 27 variables (inclus 2 variables supplémentaires "**CYCLE**" et "**DATE**").

2. REPRÉSENTATION DES DONNÉES

Afin de représenter graphiquement le jeu de données, on applique les “pairplots” et les histogrammes.

Premièrement, on visualise les “pairplot” de chaque variable avec la variable sortie “actual.values” afin de comprendre mieux la relation entre eux.

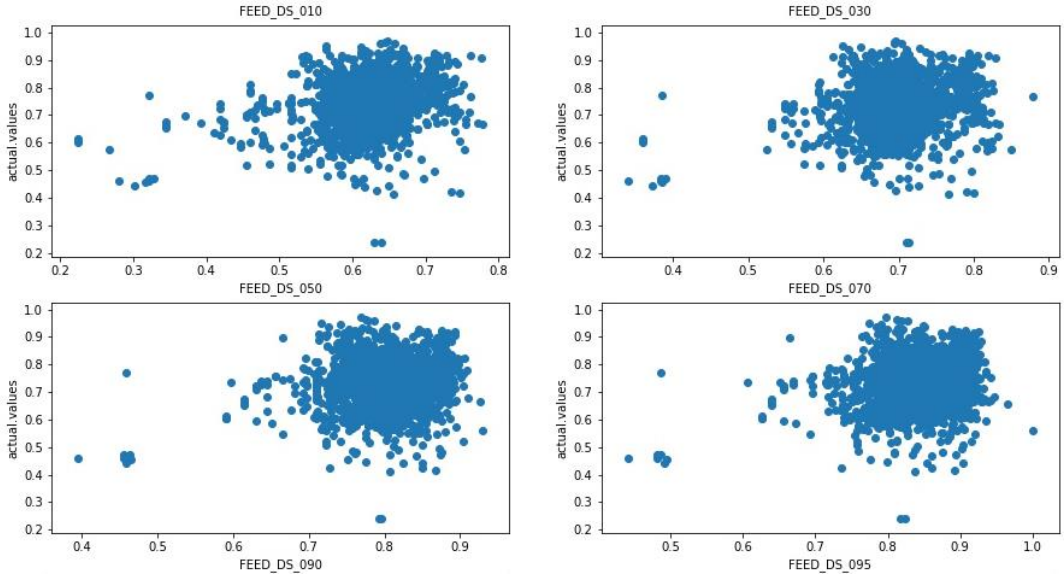

Figure 1 : “Pairplots” de quelques variables avec “actual.values”

Grâce au graphe, on vois qu’il y a des outliers dans les observations. Ainsi, il faut les enlever avant de la modélisation.

Deuxièmement, on affiche l’histogramme de chaque variables.

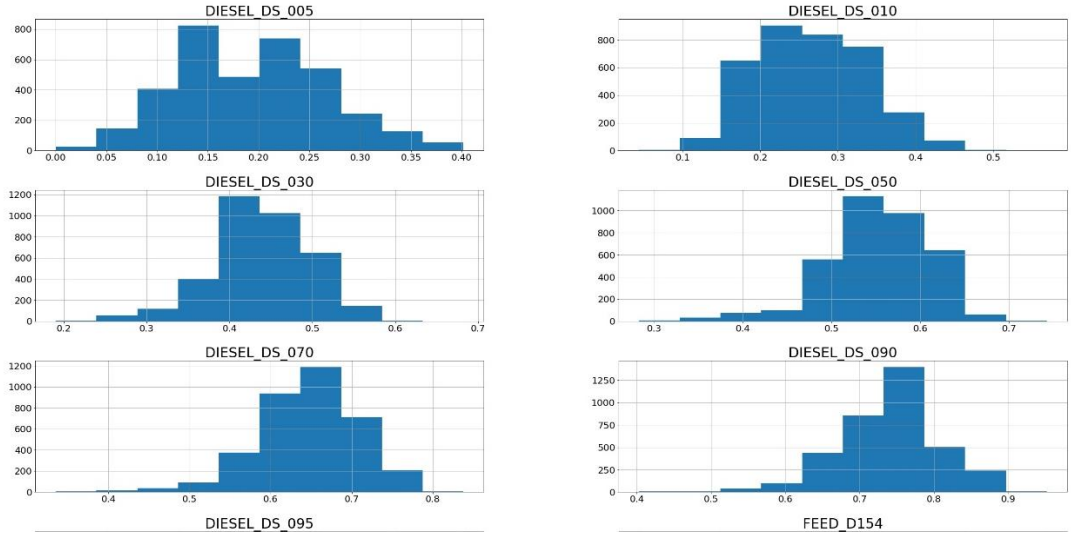

Figure 2 : Histogrammes de 6 premières variables numériques

On peut voir que la majorité des variables suivent la loi Normale.

3. PRÉPARATION DES DONNÉES

Dans cet étape, on va supprimer les colonnes et les lignes inutiles, et aussi les sites insuffisamment renseignés.

3.1. Suppression des variables redondantes

D'abord, on souhaite étudier la correlation entre variables. Ainsi, on calcule la matrice de correlation.

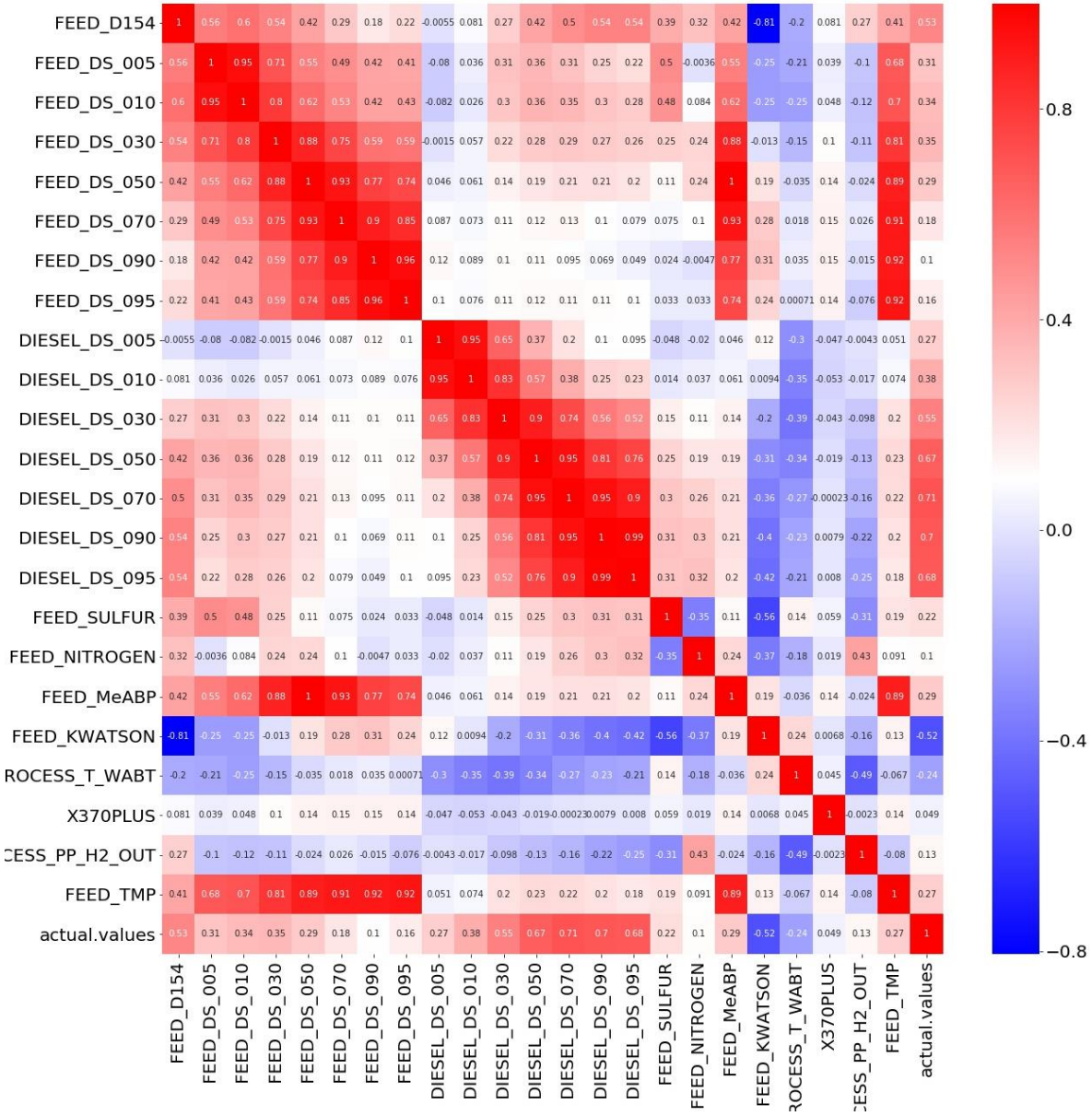

Figure 3 : Matrice de correlation

Cette matrice montre la forte correlation entre les variables de distillations. On peut voit aussi que 2 variables "FEED_MeABP" et "FEED_TMP" sont très corrélées aux variables de distillation de la charge. Ainsi, on doit les supprimer et enlever aussi les variables redondantes recommendées par des experts:

- Le TMP se déduit des valeurs de la distillation.
- Les valeurs à 30, 50 et 70% des distillations se déduisent avec une très grande fiabilité des autres valeurs de la distillation
- La MeABP est une donnée construite aussi à partir de la distillation.

Après de la suppression, il reste 15 variables et une variable de sortie “actual.values”.

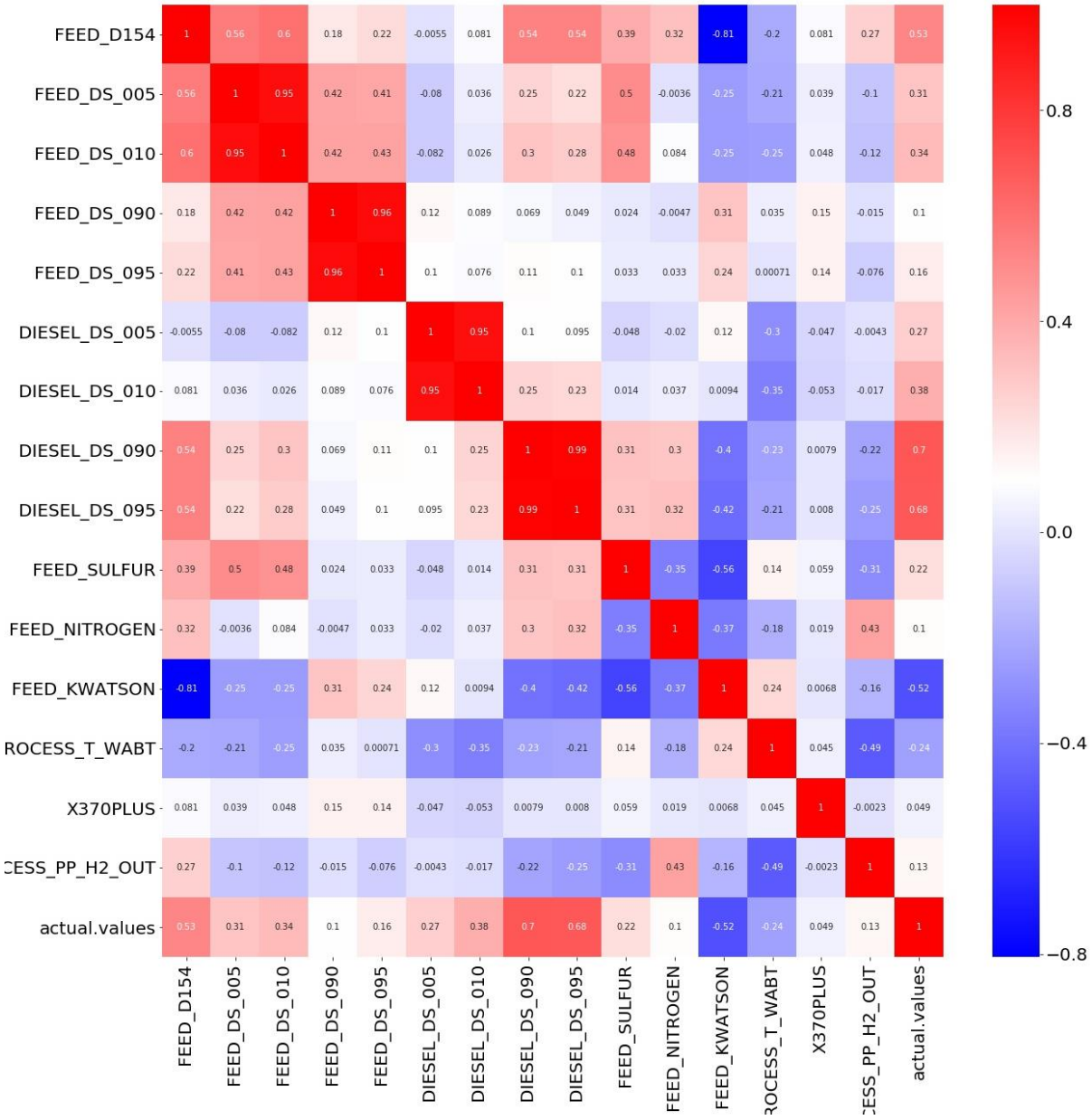

Figure 4 : Matrice de correlation après de la suppression

Le jeu de données reste maintenant 8000 lignes x 19 colonnes (15 variables expliquatives quantitatives, “actual.values”, “PLANT_NAME”, “CYCLE” et “DATE”).

3.2. Suppression des lignes inutiles

Il y a des observations contenant les valeurs manquantes (NaN value). Pour qu'une observation soit prise en compte dans les régressions, il faut qu'elle ne contienne aucune valeur manquante. Ainsi, on doit supprimer toutes les lignes ayant ces valeurs.

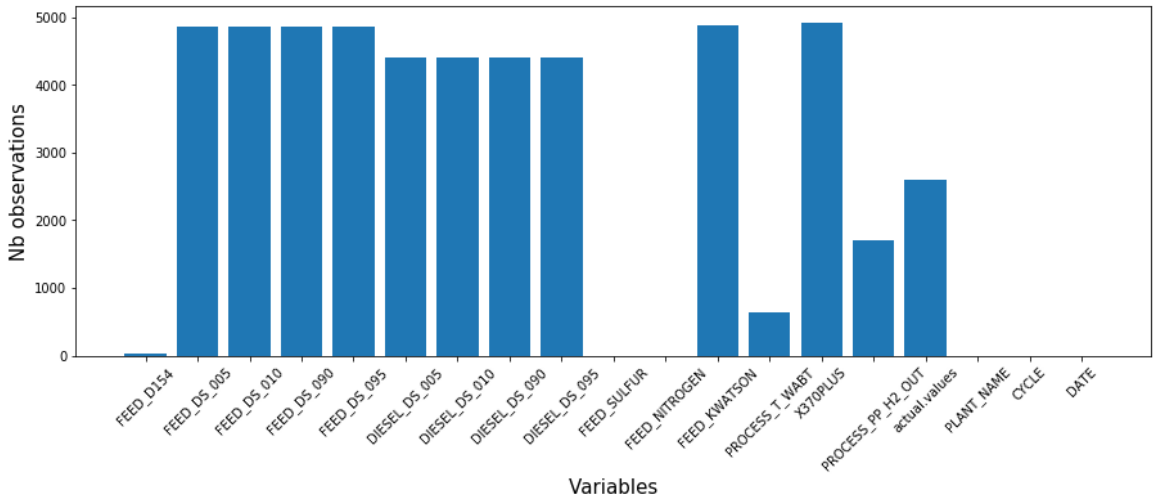

Figure 5 : Nombre de valeurs manquantes par variable

A la fin de cet étape, le jeu de données contient 2074 observations (au lieu de 8000).

3.3. Suppresion des sites insuffisamment renseignés

Les sites insuffisamment renseignés sont lesquels contiennent moins de 50 observations. Afin de les détecter, on calcule la fréquence pour chacun des sites et ensuite garde les sites satisfaisant à la condition.

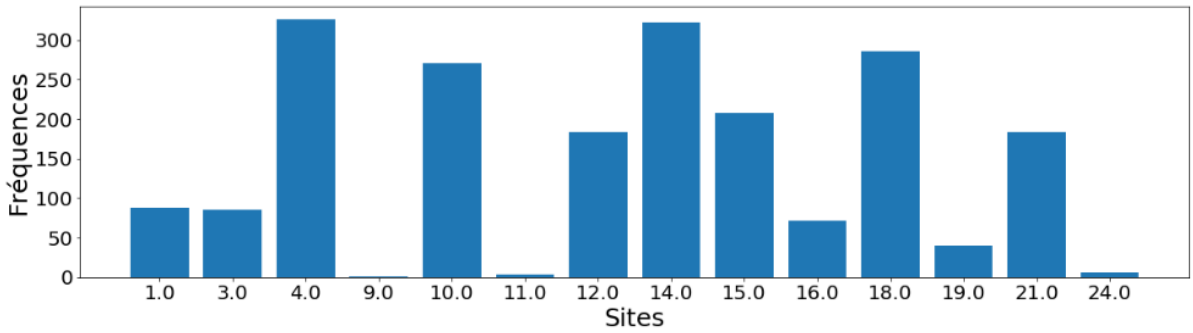

Figure 6 : Fréquence par site

Les sites 9, 11, 19, 24 contiennent moins de 50 observations, on supprime donc tous les individus pour chacun des sites.

Il reste maintenant 2024 observations.

3.4. Suppression des valeurs suspectes

On veut supprimer toutes les lignes contenant la valeur susprecte dans au moins une colonne.

Pour chaque colonne, premièrement, on calcule la valeur absolute du Z-score de cette colonne en utilisant sa moyenne et sa écarte-type. Deuxièmement, on ne garde que les lignes au dessous du seuil.

Après de la préparation, le jeu de données contient 1908 lignes x 19 colonnes.

4. CONSTRUCTION DU MODÈLE

4.1. Construire une base de calibration et de validation

On sépare le jeu de données au training set (70%) et au test set (30%). Pour assurer que les bases de calibration (et surtout de validation) contiennent des observations de tous les sites, on applique le stratégie “stratified sampling” qui permet de répartir les observations dans les 2 bases de façon équilibrée.

Après de la separation, on visualise la nombre des observations des 2 bases dans chacun des sites.

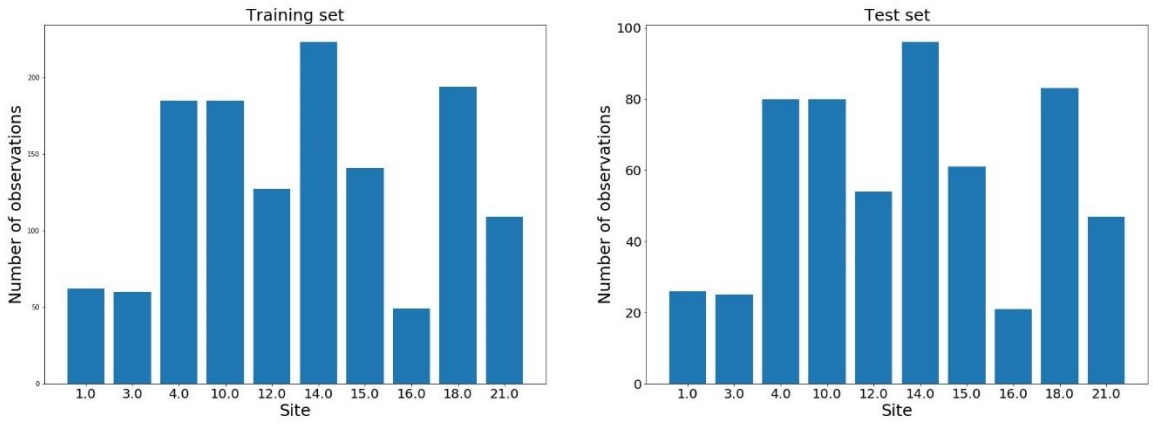

Figure 7 : Distributions des observations par site dans 2 bases

Comme les 2 graphes sont similaires, on peut assure que les proportions des sites dans 2 bases sont pareilles.

4.2. Les critères

Pour répondre au problème posé, nous devons avoir recours à un modèle linéaire de la forme :

$$Y = a_0 + a_1 X_{1,i} + a_2 X_{2,i} + \ldots + a_p X_{p,i} + \epsilon_i$$

Les critères de selection du modèle sont RMSEP (Root Mean Square Error of Prediction)

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

et MAE (Mean Absolute Error)

$$MAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n}$$

Avec y_i est la valeur de données et $\hat{y}_i$ est la valeur de prediction.

Pour utiliser RMSEP et MAE, on découpe des données en deux sous-échantillons:

- Premier échantillon utilisé pour estimé les paramètres du modèle (base de calibration - train).
- Second échantillon utilisé pour évaluer les prédictions en les comparant aux valeurs observes (base de validation-test).

Le meilleur modèle est lequel a RMSEP et MAE minimal.

4.3. Les modèles

On construit une régression linéaire globale avec tous les sites confondus et un modèle site par site. De plus, on veut construit un modèle qui ne contient que les variables significatives.

4.3.1. Modèle global

D'abord, on applique la regression linéaire avec 15 variables explicatives quantitatives pour construire un modèle prédictif. Ensuite, on évalue la performance de prediction par les 2 critères RMSEP et MAE. RMSEP de ce modèle est **0.040849** et son MAE est **0.030343**.

Afin d'évaluer mieux ce modèle, on continue travailler avec les modèles site par site et le méthode "exhaustive" et comparer leur performance.

4.3.2. Modèles site par site

Dans cette partie, on fait un même méthode que le modèle global mais avant de la modélisation, on groupe les observations étant même site et construit un modèle pour chacun de groupe. Puis, on calcule RMSEP et MAE de chaque groupe.

Table 1 : RMSEP et MAE de chacun modèle des sites

Sites	RMSEP	MAE
1	0.02252436732807503	0.018863455834622605
3	0.04523540841290938	0.02340362750052709
4	0.028532686725709484	0.022781246587503905
10	0.03179565653309578	0.024245496676472977
12	0.02319920473686336	0.01593251929185253
14	0.018482515416897712	0.012328584113020619
15	0.01800523307117602	0.015035625357605878
16	0.03201516037505363	0.022047042057226834

18	0.03319385168635647	0.025363944805770523
21	0.029097906386742475	0.021537787781948366

Enfin, on calcule la valeur moyenne de ces critères pour comparer avec le modèle global.

- meanRMSEP = **0.028208**
- meanMAE = **0.021787**

On peut voir si on divise le jeu de données aux groups, la performance du modèle améliore: **0.028208** vs **0.040849** pour RMSEP et **0.021787** vs **0.030343** pour MAE.

4.3.3. Méthode "exhaustive"

Maintenant, on veut trouver un ensemble des variables significatives en fabriquant tous les modèles possibles et ne retenant que ceux dont la RMSEP est la meilleure. Avec 15 variables, on va faire en tout $2^{15} - 1 = 32767$ modèles. On applique ce méthode pour le modèle global parce qu'il y aura 32767*10 modèles (c'est trop) si on applique le méthode "exhaustive" pour les modèles site par site.

Pour le faire, on doit d'abord implementer une function permettant obtenir tous les sous-ensembles d'un set des variables.

Ensuite, pour chaque sous-ensemble des variables, on construit un modèle de regression linéaire et sauvegarde sa valeur de RMSEP et de MAE.

Finalement, on trouve un ensemble des variables ayant RMSEP et MAE meilleurs. On garde 13 variables significatives : "**FEED_D154**", "**FEED_DS_005**", "**FEED_DS_010**", "**FEED_DS_090**", "**FEED_DS_095**", "**DIESEL_DS_005**", "**DIESEL_DS_090**", "**DIESEL_DS_095**", "**FEED_SULFUR**", "**FEED_NITROGEN**", "**PROCESS_T_WABT**", "**X370PLUS**", "**PROCESS_PP_H2_OUT**". Ce modèle donne :

- RMSEP = **0.040808**
- MAE = **0.030061**

Malgré sa performance meilleur que le modèle global classique, ce méthode n'est pas très bon en comparant avec les modèles site par stie.

4.4. Comparaison avec modèle PLS

On veut maintenant comparer le modèle regression linéaire avec le modèle PLS. On va étudier la performance de ces modèle par 2 stratégies : global et site par site.

Grâce aux parties précédentes, on a déjà eu RMSEP et MAE de modèle linéaire (global et aussi site par site). Ainsi, on ne construit que les modèles PLS et calcule leur RMSEP et MAE.

On sommaire leurs critères dans la table ci-dessous.

Table 2 : LR versus PLS

	RMSEP		**MAE**	
	Régression linéaire	PLS	Régression linéaire	PLS
Global	0.040849	0.044962	0.030343	0.034238
Site par site (valeur moyenne)	0.028208	0.030316	0.020153	0.021787

On peut conclure que la régression linéaire est meilleur dans ce cas.

CONCLUSION

La projet nous aide à comprendre mieux les méthodes de regressions utilise.

En comparant les RMSEP et MAE entre les méthodes, entre la regression linéaire et PLS, on peut choisir le meilleur modèle : la regression linéaire site par site. Il donne la meilleure performance sous les critères RMSEP (**0.028208**) et MAE (**0.020153**).

Ce modèle peut être utilise pour la prediction. Pour l'appliquer, on entre 15 valeurs explicatives quantitatives et le site, il va sortir une valeur pour la "**actual.values**".