:rezelIA

Introduction aux compétitions Kaggle

Atelier 4

Noé Vernier
Jacques Sun
Rezel IA

7 Mars 2024

Sommaire

1 Introduction

2 Comprendre la data

3 Feature engineering

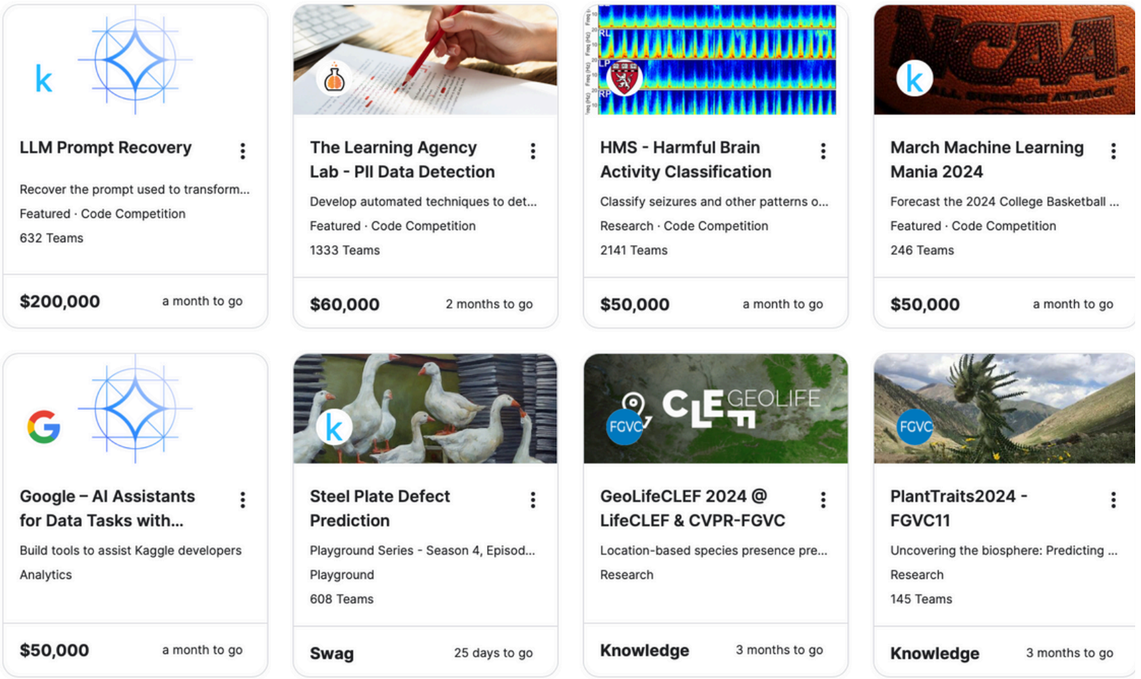
4 Data cleaning

5 Data splitting

C’est quoi ?

Site proposant des **bases de données**, **notebook** et **tutoriel** pour réaliser des projets ou compétitions

kaggle



House Prices

Advanced Regression Techniques

Goal

It is your job to predict the sales price for each house. For each Id in the test set, you must predict the value of the SalePrice variable.

Dataset

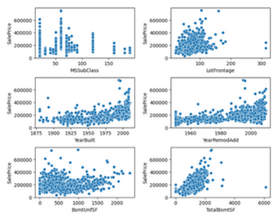
Categorical Features

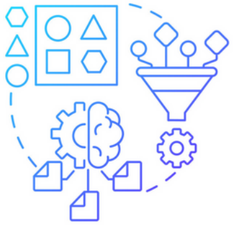
- Nominal: MSSubClass, MSZoning, Street, ...
- Ordinal: OverallQual, OverallCond, ExterQual, ...

Numerical Features

- Continuous: LotFrontage, LotArea, ...
- Descrete: YearBuilt, YearRemodAdd, ...

Step of resolution





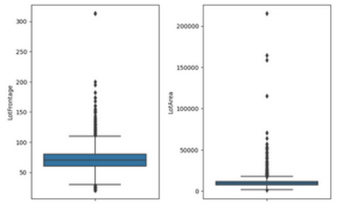
kaggle

1

Understanding Data

2

Data Cleaning

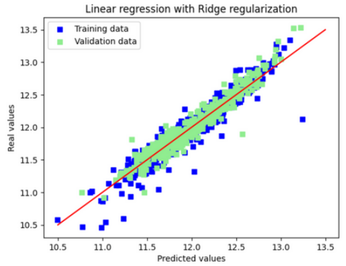


3

Feature Engineering

4

Modeling



5

Submission

Understanding data - Type of data

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape
0	1	60	RL	65.0	8450	Pave	NaN	Reg
1	2	20	RL	80.0	9600	Pave	NaN	Reg
2	3	60	RL	68.0	11250	Pave	NaN	IR1

1. **Numeric** → Stock Prices, Temperature Data, ...
2. **Categorical** → Gender, Education Level, ...
3. **Missing value** → NaN, -, NA, ...

Understanding data - Categorical encoding

Label Encoding

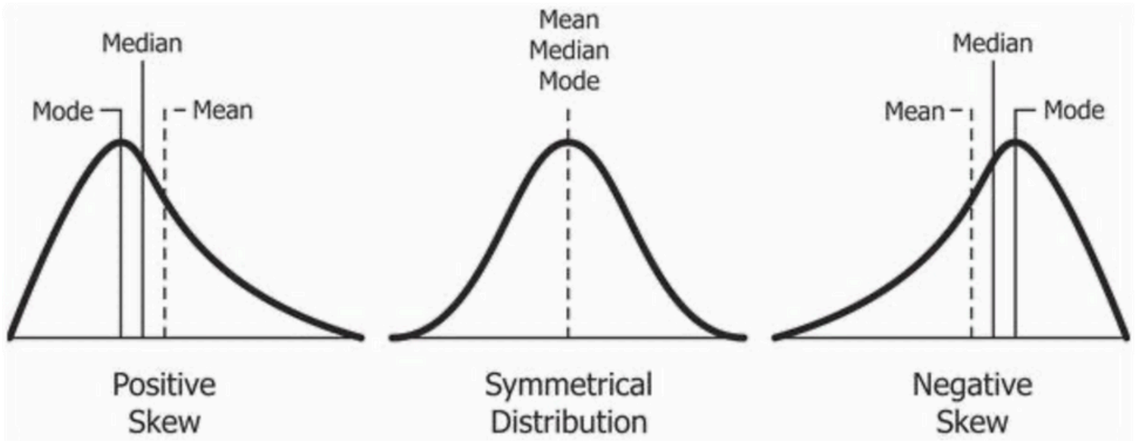
Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

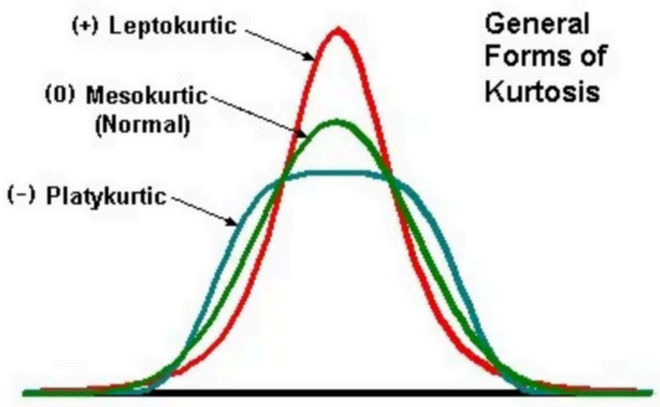
→

One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

Understanding data - Distribution





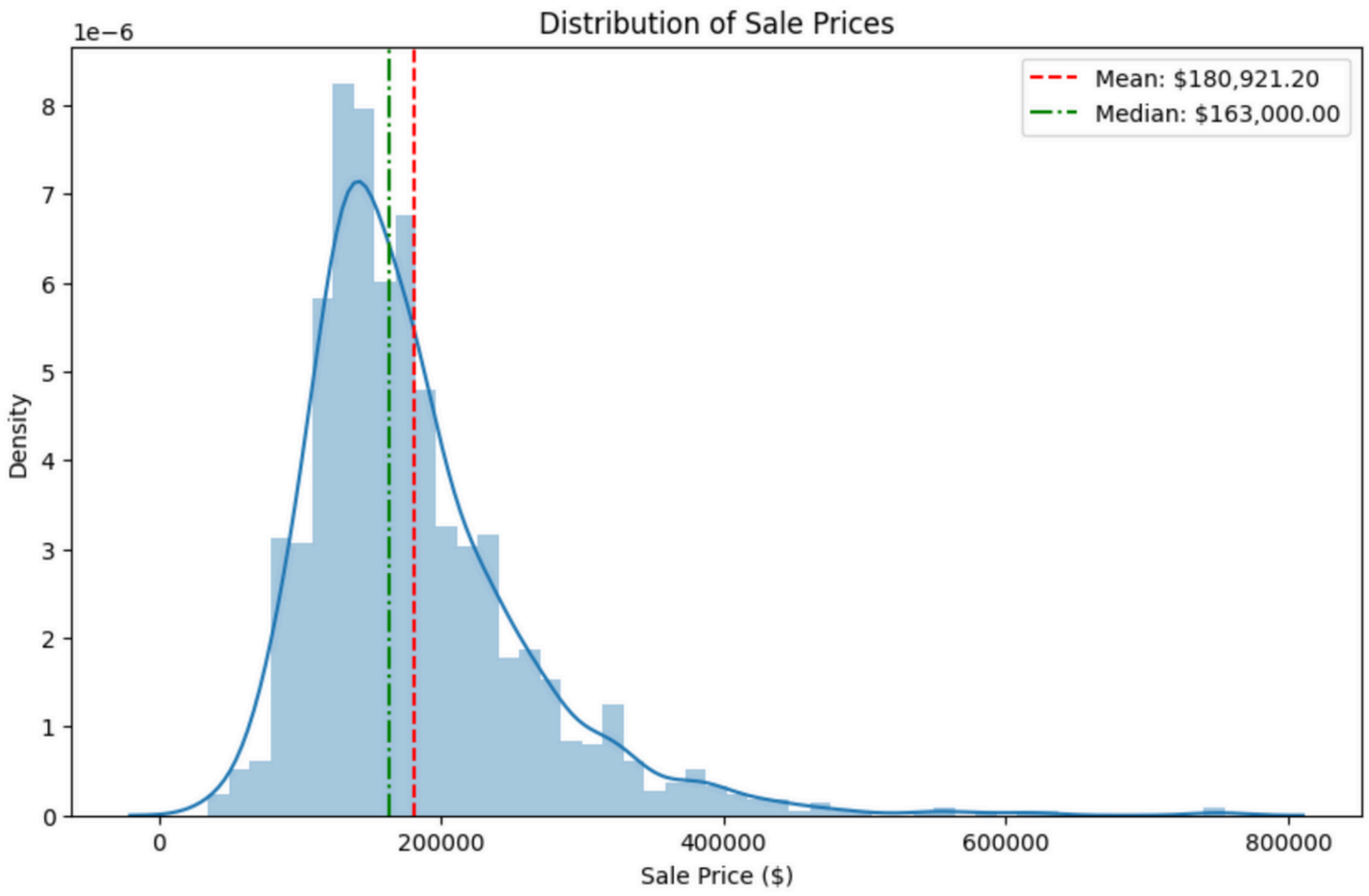
Skewness

$$\text{skew}(X) = \mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^3\right]$$

Kurtosis

$$\text{kurt}(X) = \mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^4\right]$$

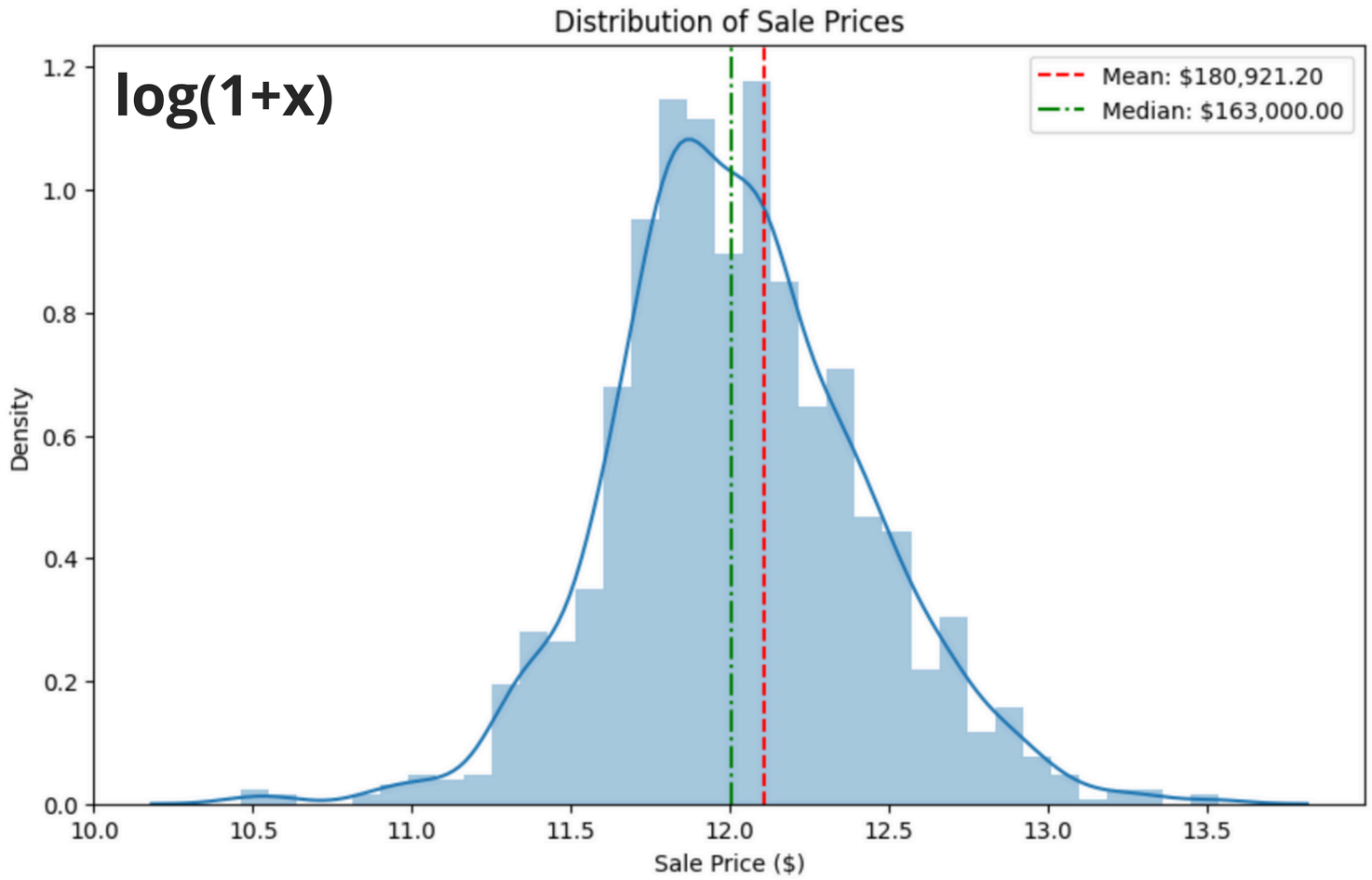
Understanding data - Distribution



Skewness = 1.882876

Kurtosis = 6.536282

Understanding data - Distribution

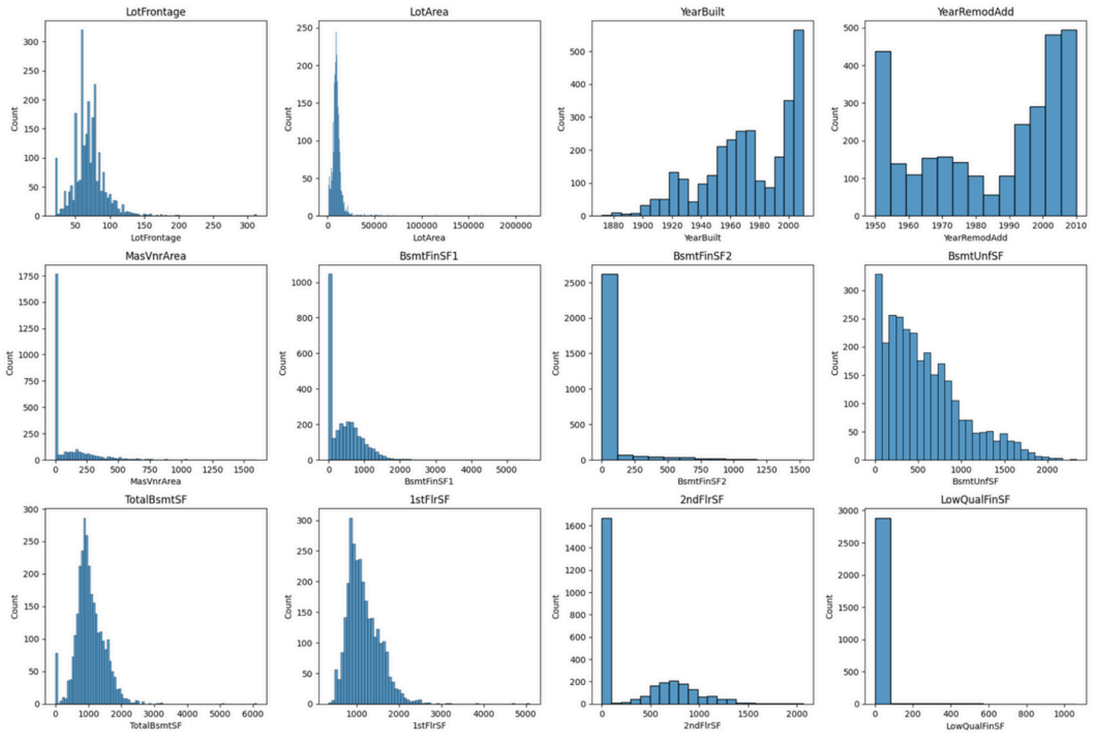


Skewness = 0.121347

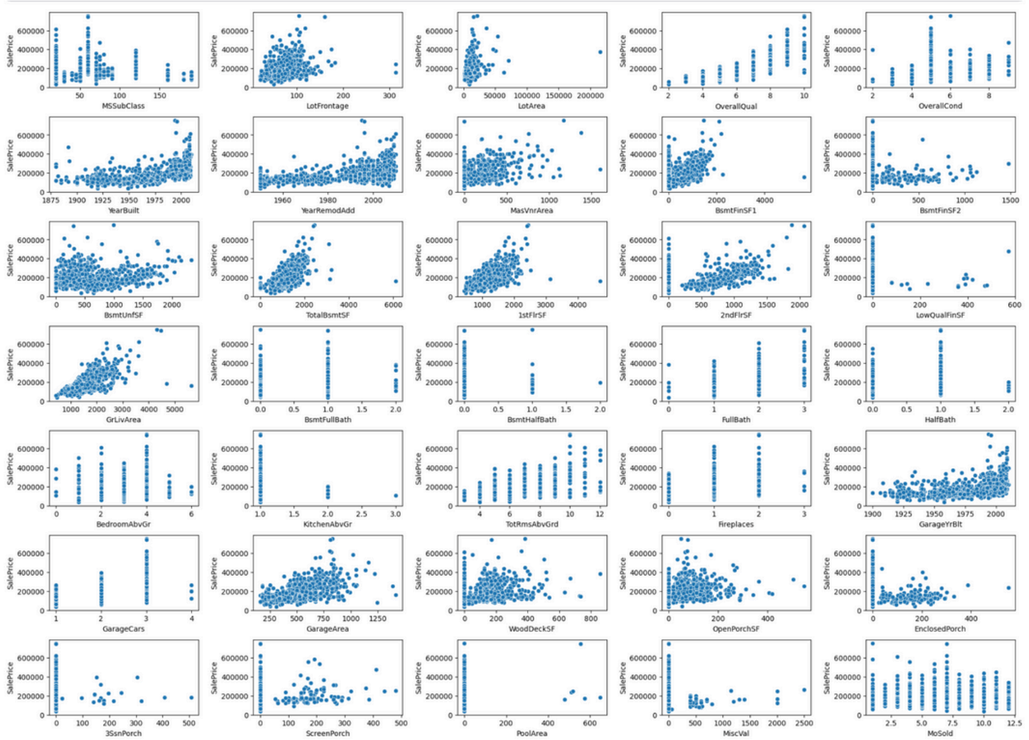
Kurtosis = 0.809519

→ Normaliser pour éviter le pb d'échelle, améliore les performances

Understanding data - Univariate / Bi-Variate

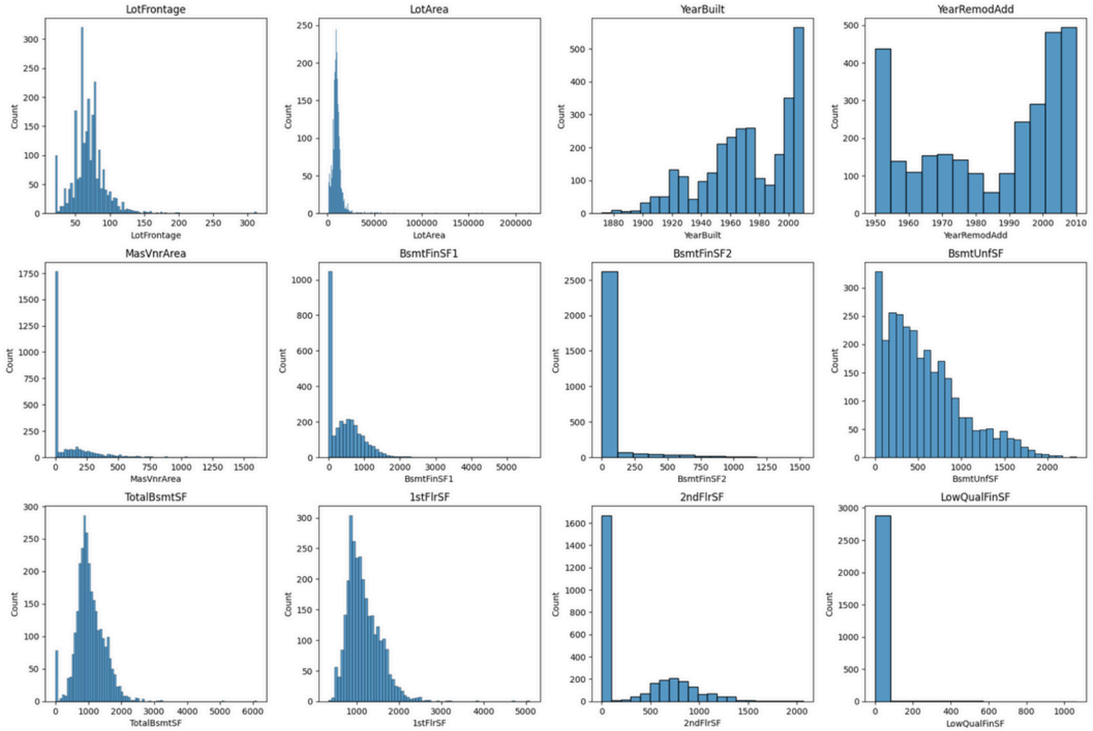


Univariate

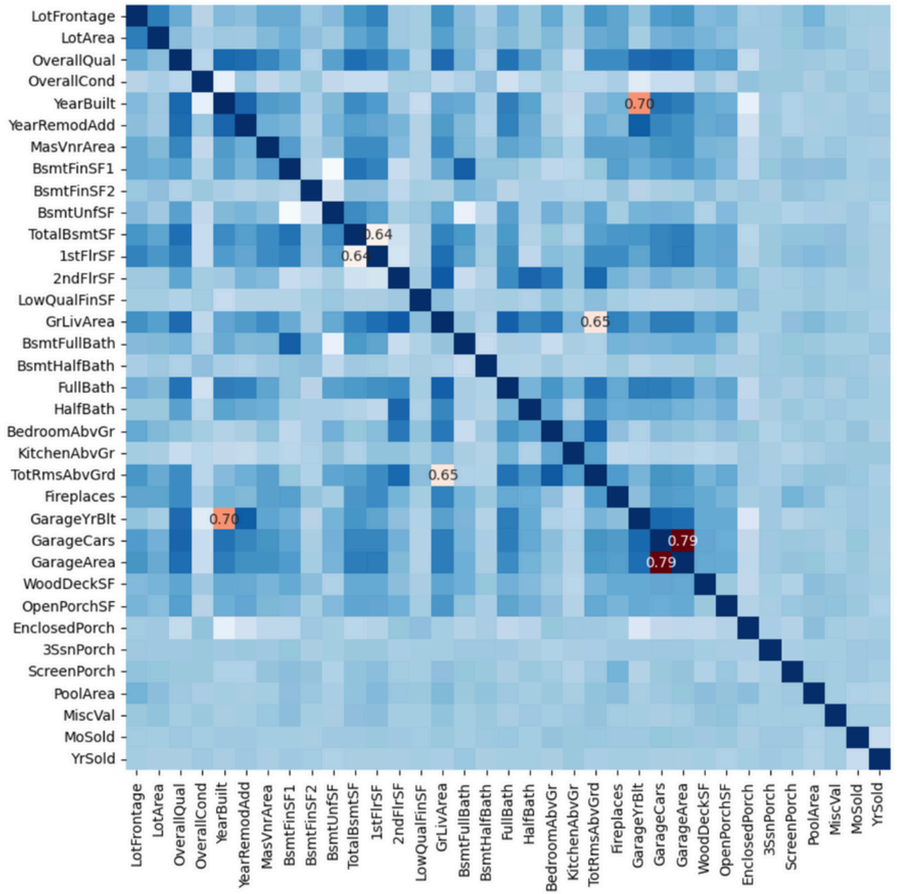


Bi-Variate

Understanding data - Univariate / Bi-Variate



Univariate



Bi-Variate

Feature Engineering

Feature Engineering

Feature Engineering

Feature Engineering is a process that involves **transforming** raw data into features that **more precisely** represent the underlying problem for a predictive model

1

Feature Simplification

→

2

Combinations of Features

→

3

Polynomials on Features

Feature Engineering - Feature Simplification

```
all_data["SimplOverallQual"] = all_data.OverallQual.replace({1 : 1, 2 : 1, 3 : 1, # bad
                                                             4 : 2, 5 : 2, 6 : 2, # average
                                                             7 : 3, 8 : 3, 9 : 3, 10 : 3 # good
                                                            })
```

Feature Engineering - Feature Combination

```
all_data["AllSF"] = all_data["GrLivArea"] + all_data["TotalBsmtSF"]

all_data['TotalLot'] = all_data['LotFrontage'] + all_data['LotArea']

all_data['TotalBsmtFin'] = all_data['BsmtFinSF1'] + all_data['BsmtFinSF2']

all_data['TotalSF'] = all_data['TotalBsmtSF'] + all_data['2ndFlrSF']
```

Feature Engineering - Feature Polynomials

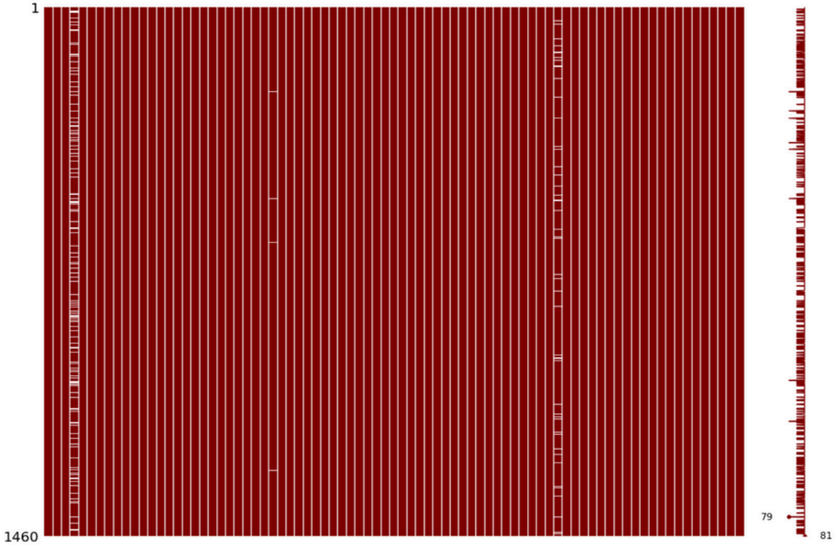
```
all_data["AllSF-2"] = all_data["AllSF"] ** 2
all_data["AllSF-3"] = all_data["AllSF"] ** 3
all_data["AllSF-Sq"] = np.sqrt(all_data["AllSF"])

all_data["OverallQual-s2"] = all_data["OverallQual"] ** 2
all_data["OverallQual-s3"] = all_data["OverallQual"] ** 3
all_data["OverallQual-Sq"] = np.sqrt(all_data["OverallQual"])
```

Data Cleaning

Data Cleaning

Data Cleaning

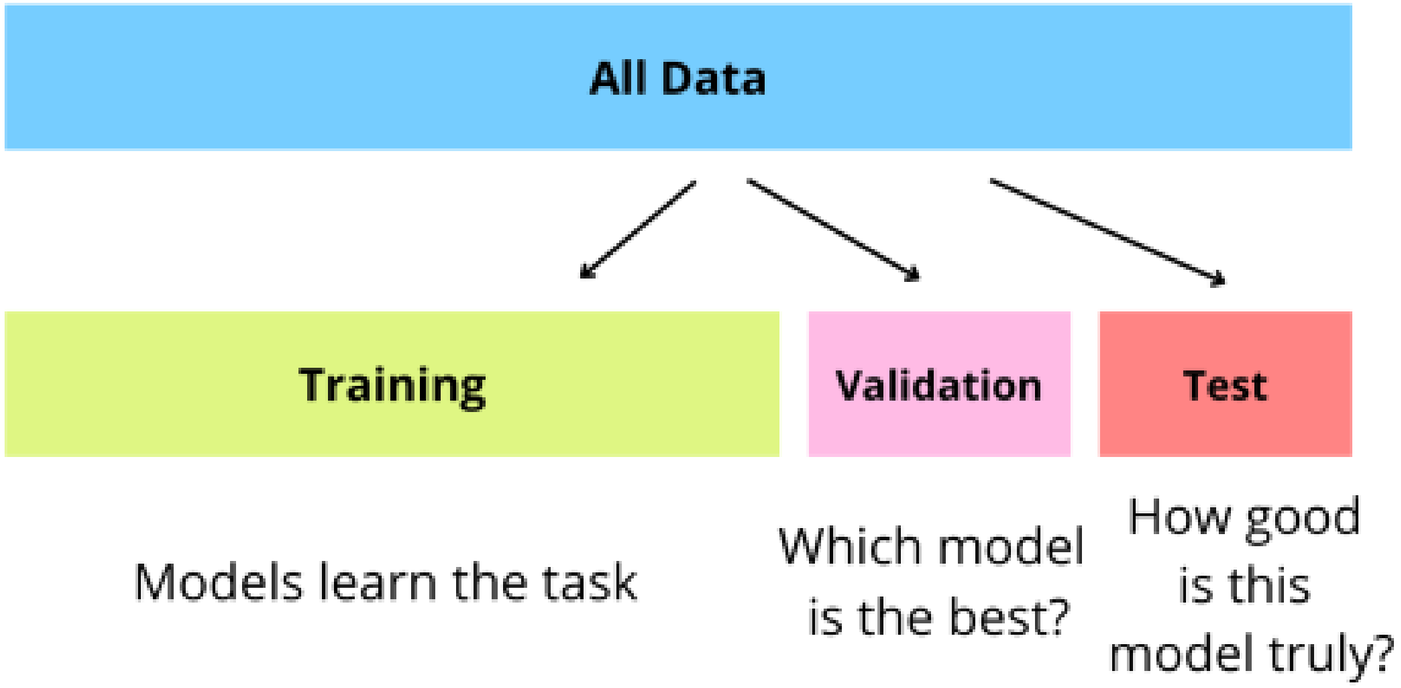
Data Cleaning is the process of **improving** data quality by identifying and **correcting** or **removing** inaccurate, **incomplete**, **irrelevant**, or **corrupted** records from a dataset



Data Cleaning

- **Enlever colonne** : Lorsque qu'il y a trop de NaN dans cette colonne
- **Enlever ligne**
- **Remplace les NaN** : valeur médiane, moyenne
- **Enlever outlier**
- **Features corrélés**
- **Feature inintéréssante** (une seule valeur quasi tout le temps)

Data Splitting



Data Splitting - K-Fold Cross Validation