**Historique**

Les premiers termes pour désigner la fouille de données sont apparut dans les années 1960. Les statisticiens utilisent des termes comme « Pêche de données » pour désigner ce qu’ils considéraient comme une mauvaise pratique de l’analyse de données sans hypothèse. Le mot « DATAMINING » est apparu dans les années 1990. Gregory Piatesky-Shapiro a inventé le « KNOWLEDGE Discovery in Databases », ce terme est devenu populaire en apprentissage communautaire. Quant au terme « Datamining », il apparait en 1991 et plus utilisé dans les milieux d’affaires et de presse. Aujourd’hui, les termes Datamining et Knowledge Discovery sont tous deux utilisés.

L’extraction de données existe depuis des siècles, en effet, auparavant, des travaux sur les méthodes d'identification des données ont été réalisés. La prolifération, ainsi que la puissance croissante des nouvelles technologies, ont contribué fortement à l’augmentation des collectes des données,la manipulation et la capacité de stockage. Les données se sont multipliées en taille et en complexité, corrélées avec des découvertes de techniques telles que les réseaux de neurones et les arbres de décision.

Le Datamining est un processus d’application de ces méthodes, dont le but est de découvrir les tendances cachées. En effet, on peut dire que le Dataming comble l’écart entre les statistiques appliquées et l’intelligence artificielle de gestion de bases de données. Les techniques utilisées ne sont pas récentes. En 1875, la méthode des régressions linéaires de Galton fait son apparition. Simultanément, des méthodes et des techniques voient le jour, telles que les analyses factorielles par Guttman, en 1941, les réseaux de neurones de Mac Culloch et Pitts, en 1943 et les arbres de décision, en 1984, ces techniques permettent alors d'exploiter et de découvrir des modèles de plus en plus précis.

**Machine learning vs data mining**

**Le machine Learning**

Peut-être décrit comme une technologie d’apprentissage par la donnée. L’objectif est de rendre un ordinateur « intelligent » en lui soumettant un algorithme et des modèles de données afin d’automatiser des tâches complexes.



**Le data mining**

Également appelé forage de données, quant à lui est une méthode d’exploration des données. Il sert à analyser des volumes colossaux de data afin de déduire des relations et liens entre celles-ci. L’objectif est de transformer les données Big Data en informations compréhensibles et exploitables.



**Data Mining ?**

1990 : les entreprises commencent à stocker de plus en plus de données concernant leur clients, sans planification expérimentale. Les méthodes statistiques classiques sont massivement utilisées pour extraire de la connaissance de ces données (CRM, gestion de la relation client). C’est la naissance du data mining.

Le Data Mining est l’ensemble des :

Le data-mining est un processus de découverte de règle, relations, corrélations et/ou dépendances à travers une grande quantité de données, grâce à des méthodes statistiques, mathématiques et de reconnaissances de formes.

Ensemble de méthodes et de techniques d’analyse de données et d’extraction d’information structurée en vue d’aider à la prise de décision:

* Mettre en évidence des informations présentes mais noyées par le volume de données : Datamining descriptif (ML Non-Supervisé)

Ex: achat de riz + limonade==> achat de poissons

* Extrapoler des nouvelles informations à partir de données existantes: Datamining prédictif (ML Supervisé)

Ex: Les achats des clients d’un tel produit le mois 3 de l’année prochaine sera 3655124 DH.

**Cleaning DATA**

Le nettoyage des données est le processus de préparation des données en supprimant ou en modifiant des données incorrectes, incomplètes, non pertinentes, dupliquées ou mal formatées, c’est-à-dire en les rendant exactes et utilisables. Le nettoyage des données n’est pas toujours une question de suppression, mais aussi de recherche de documents en double, de fichiers ou d’enregistrements incomplets et de les enrichir. Il s’agit d’une étape essentielle pour assurer la précision.

**Cause de cette saleté :**

Les données sales sont produites en raison d’erreurs humaines (par exemple, erreurs de faute de frappe), de problèmes de conception de formulaires (par exemple, le champ d’identification manquant ci-dessus n’était pas obligatoire) et de nombreux autres facteurs. En outre, les méthodes de collecte de données sont souvent vaguement contrôlées, ce qui entraîne des valeurs manquantes et hors de portée. Les données sales sont inévitables.

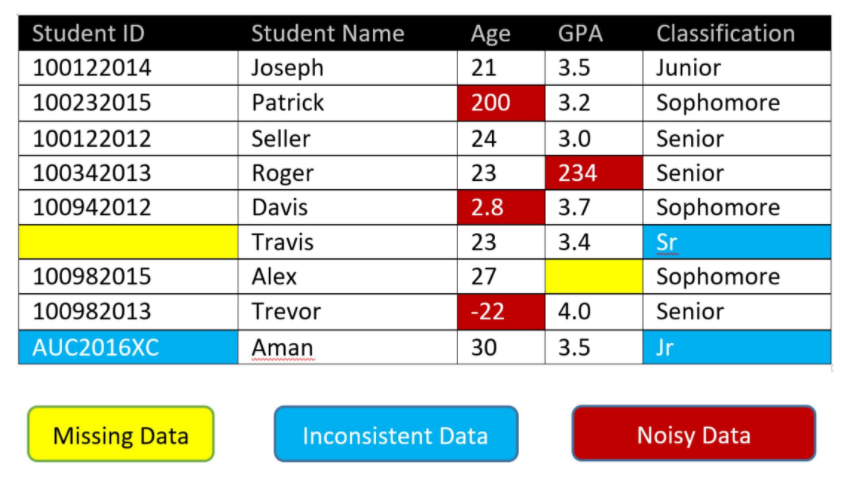
**Les différents types des bruits dans la base de données :**

Missing data : Pas de valeur enregistrer dans telle colonne.

Noise :Une valeur qui n’a pas du sens . Comme une personne remplissant la valeur numérique -689 dans le domaine du salaire.

Outlier :Une valeur qui n’a pas de sens par Exemple : âge de quelqu’un est 200 ans .

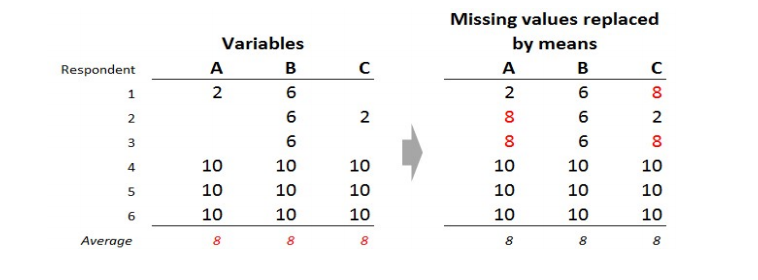
Data inconsistency : Pas de standardisation dans le format du input Exemple : la date -2010/12/25- vs -25/12/2010-



**SOLUTIONS**

Missing data :

* Supprimer
* Tolération
* Imputation
  + Machine Learning imputation techniques( KNN SVR DT …)
  + Statistical imputation techniques ( la moyenne ,mode …)



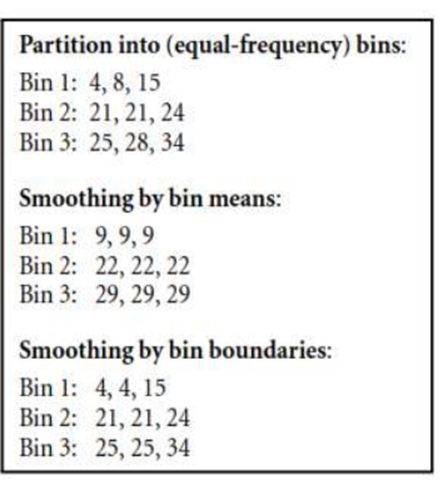
8 ! C’est grave à la valeur moyenne calculer : M=(10+10+10+2)/4=8

Noisy data :

1.Binning :

Etape 1 : diviser en bins





çç

Etape 2 :

Bin means :

Pour l’exemple on a calculer la moyenne du bin 1 M=(4+8+15)/3=9

De même pour les bins 2 et 3

Bin boundaries :

Pour l’exemple on a fixer le Max et le Min et puis on compare la distance des autres valeurs par rapport aux deux valeurs et on les changes par la valeur la plut proche des deux.

2.Regression

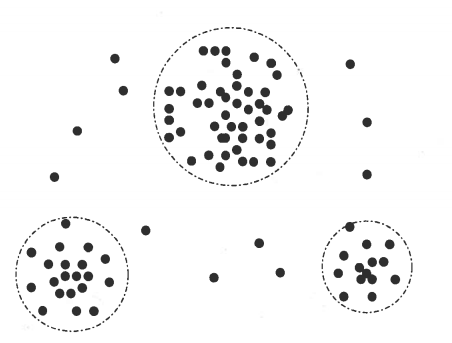
Méthode qui serve a construire une fonction qui représente toute les points (DATA).la régression linière inclut la recherche du meilleure droite pour lier deux attributs (ou variables) a fin de prédire l’un par l’autre .

Linière régression :

Y = b0 + b1\*X

3.Outliers

Apres le regroupement des points similaire on construire des (clusters).donc les points qui se situent à l’extérieure de ces groupes on les considèrent des outliers.



**Data Reduction**

**Feature Extraction**

L’apprentissage automatique fonctionne selon une règle simple – si vous mettez des ordures, vous n’obtiendrez que des ordures à sortir. Par ordures ici, je veux dire le bruit dans les données.

Cela devient encore plus important lorsque le nombre de fonctionnalités est très important. Vous n’avez pas besoin d’utiliser toutes les fonctionnalités à votre disposition pour créer un algorithme. Vous pouvez aider votre algorithme en alimentant uniquement les fonctionnalités qui sont vraiment importantes. J’ai moi-même vu des sous-ensembles de fonctionnalités donnant de meilleurs résultats que l’ensemble complet de fonctionnalités pour le même algorithme.

Cette réduction sert à :

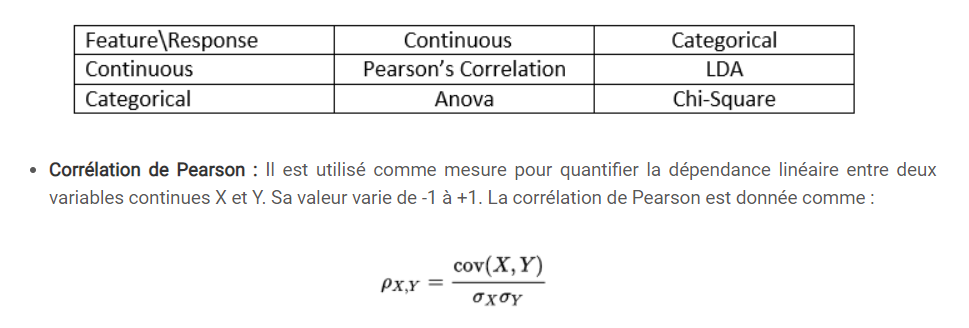
* Il permet à l’algorithme d’apprentissage automatique de s’entraîner plus rapidement.
* Il réduit la complexité d’un modèle et le rend plus facile à interpréter.
* Il améliore la précision d’un modèle si le sous-ensemble droit est choisi.

Ensuite, nous discuterons des différentes méthodologies et techniques que vous pouvez utiliser pour sous-ensemble de votre espace de fonctionnalités et aider vos modèles à mieux performer et efficacement. Alors, commençons.

**Méthodes de filtre**



Les méthodes de filtre sont généralement utilisées comme étape de prétraitement. La sélection des fonctionnalités est indépendante de tous les algorithmes d’apprentissage automatique. Au lieu de cela, les fonctionnalités sont sélectionnées en fonction de leurs scores dans divers tests statistiques pour leur corrélation avec la variable de résultats. La corrélation est ici un terme subjectif. Pour obtenir des conseils de base, vous pouvez vous référer au tableau suivant pour définir les co-efficacités de corrélation.



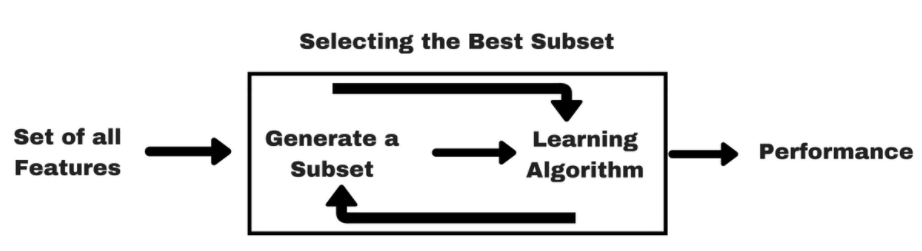
**LDA**: L’analyse linéaire discriminante est utilisée pour trouver une combinaison linéaire de caractéristiques qui caractérise ou sépare deux classes ou plus (ou niveaux) d’une variable catégorique.

**ANOVA**: ANOVA signifie Analyse de la variance. Il est similaire à LDA, sauf pour le fait qu’il est exploité en utilisant une ou plusieurs caractéristiques indépendantes catégoriques et une fonction dépendante continue. Il fournit un test statistique pour déterminer si les moyens de plusieurs groupes sont égaux ou non.

**Chi-Carré**: Il s’agit d’un test statistique appliqué aux groupes de caractéristiques catégoriques pour évaluer la probabilité de corrélation ou d’association entre eux à l’aide de leur distribution de fréquences.

Une chose qu’il faut garder à l’esprit est que les méthodes de filtre n’éliminent pas la multicollinarité. Ainsi, vous devez faire face à la multicollinarité des fonctionnalités ainsi avant de former des modèles pour vos données.

## Méthodes wrapper



Dans les méthodes d’emballage, nous essayons d’utiliser un sous-ensemble de fonctionnalités et de former un modèle en les utilisant. Sur la base des inférences que nous tirons du modèle précédent, nous décidons d’ajouter ou de supprimer des fonctionnalités de votre sous-ensemble. Le problème est essentiellement réduit à un problème de recherche. Ces méthodes sont généralement très coûteuses sur le plan informatique.

Une des meilleures façons d’implémenter la sélection de fonctionnalités avec des méthodes d’emballage est d’utiliser le paquet Boruta qui trouve l’importance d’une fonctionnalité en créant des fonctionnalités d’ombre.

Il fonctionne dans les étapes suivantes:

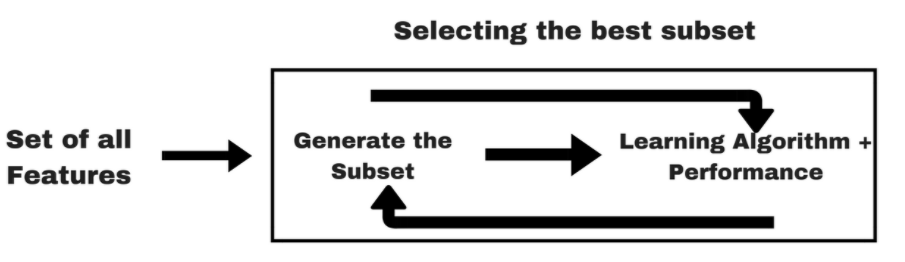
1.Tout d’abord, il ajoute le caractère aléatoire à l’ensemble de données donné en créant des copies mélangées de toutes les fonctionnalités (qui sont appelés fonctionnalités d’ombre).

2.Ensuite, il forme un classificateur de forêt aléatoire sur l’ensemble de données étendu et applique une mesure d’importance de fonctionnalité (la valeur par défaut est précision moyenne de diminution) pour évaluer l’importance de chaque fonctionnalité où des moyens plus élevés plus importants.

3.À chaque itération, il vérifie si une fonctionnalité réelle a une importance plus élevée que le meilleur de ses fonctionnalités d’ombre (c.-à-d. si la fonctionnalité a un score Z plus élevé que le score Z maximum de ses entités d’ombre) et supprime constamment les fonctionnalités qui sont considérées comme très peu importantes.

4.Enfin, l’algorithme s’arrête soit lorsque toutes les fonctionnalités sont confirmées ou rejetées, soit il atteint une limite spécifiée de séries forestières aléatoires.

## Méthodes embarquées



Les méthodes intégrées combinent les qualités des méthodes de filtre et d’emballage. Il est implémenté par des algorithmes qui ont leurs propres méthodes intégrées de sélection de fonctionnalités. Certains des exemples les plus populaires de ces méthodes sont la régression LASSO et RIDGE qui ont intégré des fonctions de pénalisation pour réduire la suraresse.

## Différence entre les méthodes de filtre et d’emballage

Les principales différences entre le filtre et les méthodes d’emballage pour la sélection des fonctionnalités sont les suivantes :

* Les méthodes de filtre mesurent la pertinence des fonctionnalités par leur corrélation avec la variable dépendante tandis que les méthodes d’emballage mesurent l’utilité d’un sous-ensemble de fonctionnalités en y enseignant un modèle.
* Les méthodes de filtre sont beaucoup plus rapides par rapport aux méthodes d’emballage car elles n’impliquent pas la formation des modèles. D’autre part, les méthodes d’emballage sont informatiques très coûteux ainsi.
* Les méthodes de filtrage utilisent des méthodes statistiques pour l’évaluation d’un sous-ensemble de fonctionnalités tandis que les méthodes d’emballage utilisent la validation croisée.
* Les méthodes de filtre peuvent ne pas trouver le meilleur sous-ensemble de fonctionnalités dans de nombreuses occasions, mais les méthodes d’emballage peuvent toujours fournir le meilleur sous-ensemble de fonctionnalités.
* L’utilisation du sous-ensemble de fonctionnalités des méthodes d’emballage rend le modèle plus sujet à un sura modelage par rapport à l’utilisation de sous-ensemble de fonctionnalités des méthodes de filtre.

## Feature Extraction

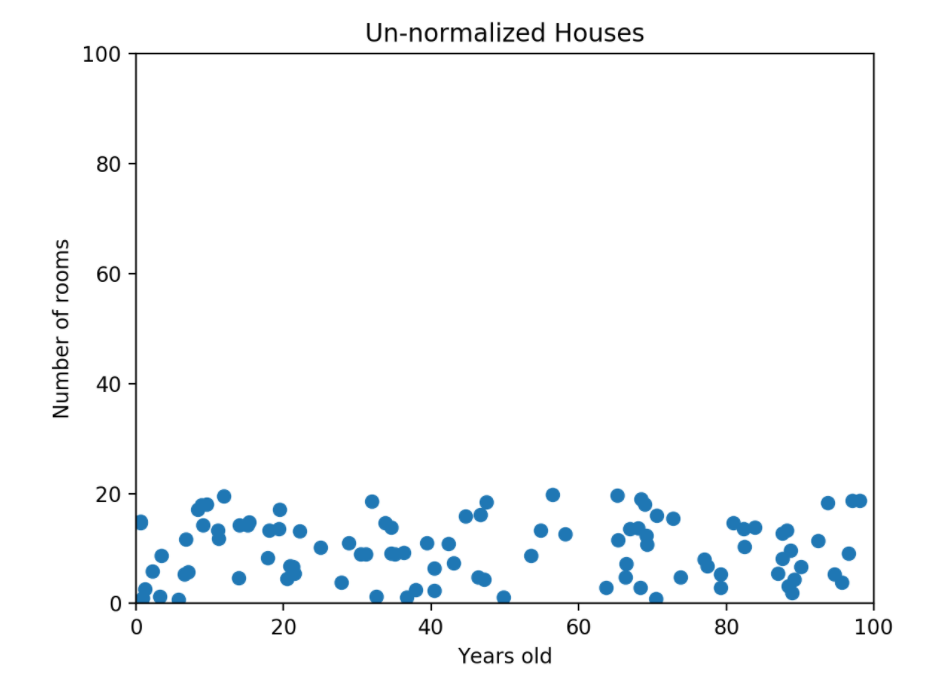
L’extraction des entités est un processus de réduction de la dimensionnalité par lequel un ensemble initial de données brutes est réduit à des groupes plus gérables pour le traitement. Une caractéristique de ces grands ensembles de données est un grand nombre de variables qui nécessitent beaucoup de ressources informatiques à traiter. L’extraction des fonctionnalités est le nom des méthodes qui sélectionnent et/ou combinent des variables en fonctionnalités, réduisant efficacement la quantité de données qui doivent être traitées, tout en décrivant toujours avec précision et complètement l’ensemble de données d’origine.

# **DATA Normalization**

L’objectif de la normalisation est de transformer les fonctionnalités pour qu’ils soient à une échelle similaire. Cela améliore la stabilité des performances et de la formation du modèle.

De nombreux algorithmes d’apprentissage automatique tentent de trouver des tendances dans les données en comparant les caractéristiques des points de données. Cependant, il ya un problème lorsque les fonctionnalités sont sur des échelles radicalement différentes.

Par exemple, considérez un ensemble de données de maisons. Deux caractéristiques potentielles pourraient être le nombre de chambres dans la maison, et l’âge total de la maison en années. Un algorithme d’apprentissage automatique pourrait essayer de prédire quelle maison serait la meilleure pour vous. Toutefois, lorsque l’algorithme compare les points de données, la fonctionnalité avec la plus grande échelle dominera complètement l’autre. Jetez un oeil à l’image ci-dessous:



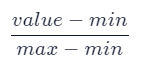
### Quand les données ont l’air écrasées comme ça, nous savons que nous avons un problème. L’algorithme d’apprentissage automatique devrait se rendre compte qu’il ya une énorme différence entre une maison avec 2 chambres et une maison avec 20 chambres. Mais à l’heure actuelle, parce que deux maisons peuvent être à 100 ans d’intervalle, la différence dans le nombre de chambres contribue moins à la différence globale.

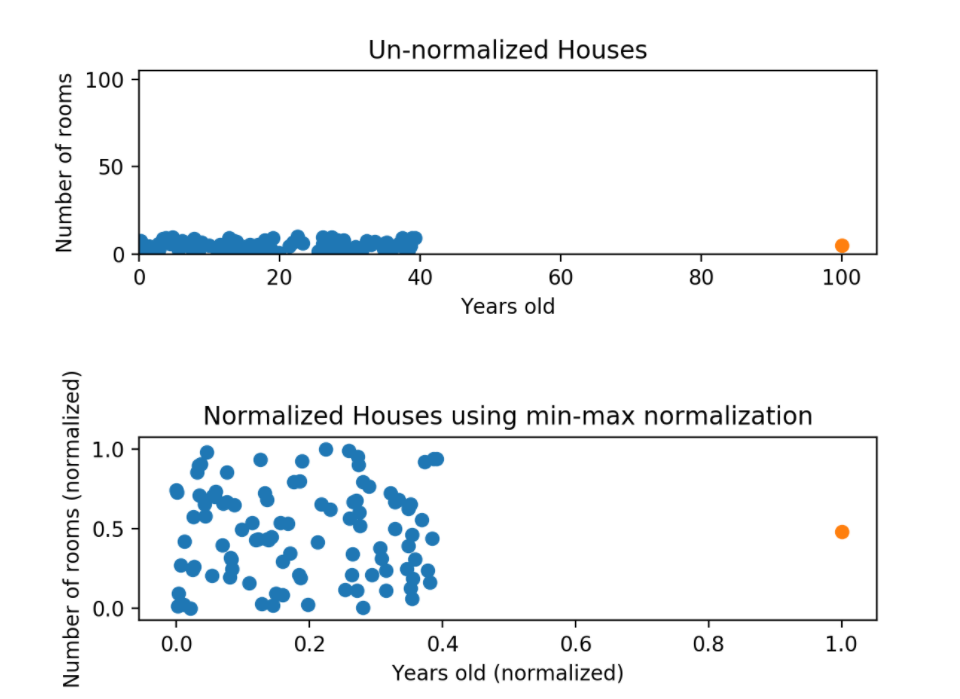
L’objectif de la normalisation est de faire en sorte que chaque point de données a la même échelle de sorte que chaque fonctionnalité est tout aussi importante.

### Normalisation min-max

La normalisation min-max est l’un des moyens les plus courants de normaliser les données. Pour chaque fonctionnalité, la valeur minimale de cette fonctionnalité se transforme en 0, la valeur maximale se transforme en 1, et chaque autre valeur se transforme en décimale entre 0 et 1.

Par exemple, si la valeur minimale d’une entité était de 20, et que la valeur maximale était de 40, alors 30 serait transformée en environ 0,5 puisqu’elle est à mi-chemin entre 20 et 40. La formule est la suivante :





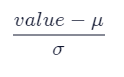
Inconvénient !!

La normalisation min-max a un inconvénient assez important : elle ne gère pas très bien les valeurs aberrantes.

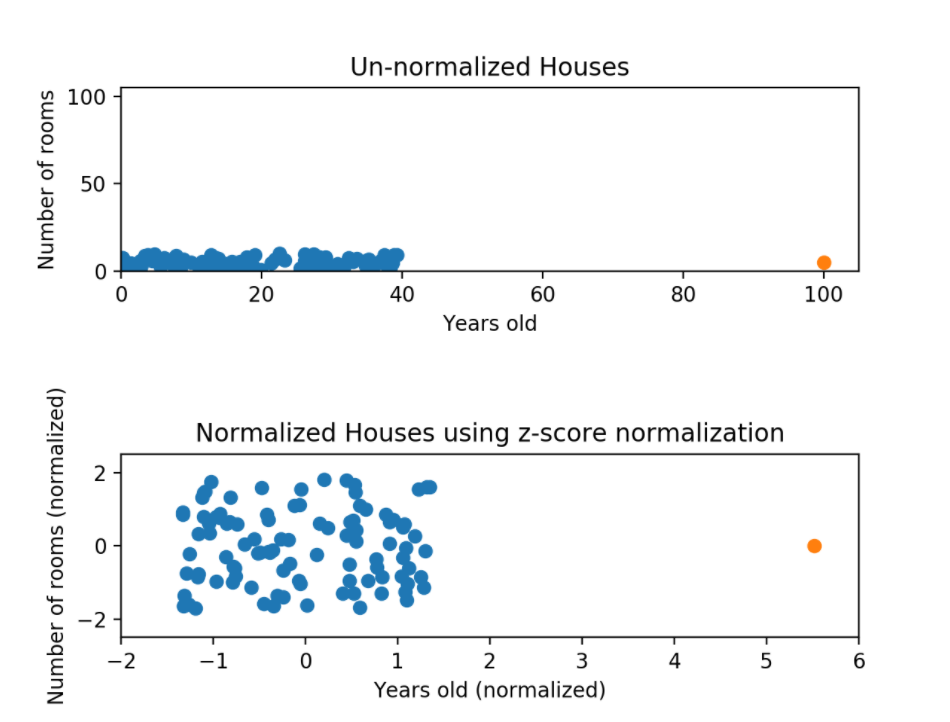
La normalisation a corrigé le problème de squishing sur l’axe y, mais l’axe x est toujours problématique. Maintenant, si nous comparions ces points, l’axe y dominerait; l’axe y peut différer de 1, mais l’axe x ne peut différer que de 0,4.

### Normalisation z-score

La normalisation du score Z est une stratégie de normalisation des données qui évite ce problème aberrant. La formule de normalisation du score Z est ci-dessous :



Ici, μ est la valeur moyenne de la fonctionnalité et σ est l’écart type de la fonctionnalité. Si une valeur est exactement égale à la moyenne de toutes les valeurs de la fonctionnalité, elle sera normalisée à 0. S’il est inférieur à la moyenne, ce sera un nombre négatif, et s’il est supérieur à la moyenne, ce sera un nombre positif. La taille de ces nombres négatifs et positifs est déterminée par l’écart type de la caractéristique originale. Si les données non normalisées avaient un écart type important, les valeurs normalisées seront plus proches de 0.



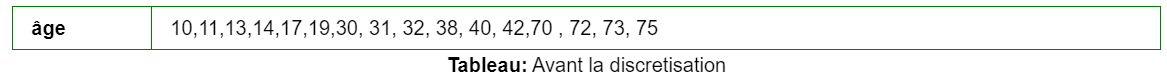
Bien que les données semblent toujours écrasées, notez que les points sont maintenant sur à peu près la même échelle pour les deux fonctionnalités - presque tous les points sont entre -2 et 2 à la fois sur l’axe x et y-axe. Le seul inconvénient potentiel est que les fonctionnalités ne sont pas sur la même échelle exacte.

Avec la normalisation min-max, nous étions assurés de remodeler nos deux caractéristiques pour être entre 0 et 1. En utilisant la normalisation z-score, l’axe x a maintenant une gamme d’environ -1,5 à 1,5 tandis que l’axe y a une portée d’environ -2 à 2. C’est certainement mieux qu’avant ; l’axe x, qui avait auparavant une portée de 0 à 40, ne domine plus l’axe y.

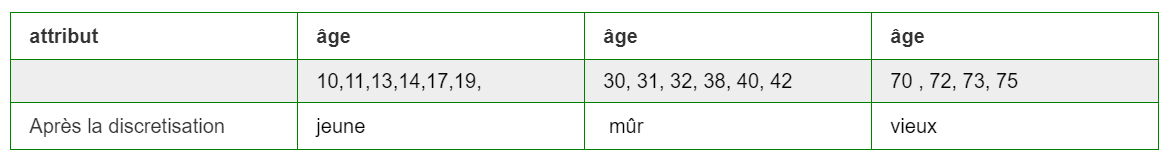
**Data Transformation -Discretization**

La discretisation des données convertit un grand nombre de valeurs de données en plus petites une fois, de sorte que l’évaluation des données et la gestion des données deviennent très faciles.

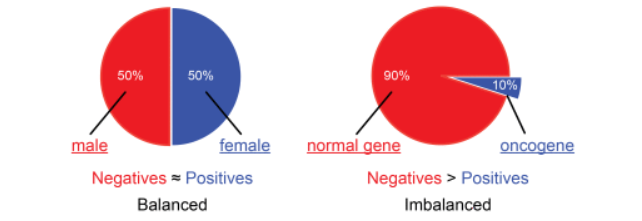
nous avons un attribut de l’âge avec les valeurs suivantes.



Après discrétization :



**Data Balancing**

****

**Problème avec un ensemble de données déséquilibré**

Supposons que vous travaillez dans une entreprise de technologie de premier plan et l’entreprise vous donne une tâche de former un modèle sur la détection de la fraude. Mais voilà le piège. La transaction de fraude est relativement rare.Ainsi, vous commencez à la formation que vous modélisez et obtenir plus de 95% de précision.Vous vous sentez bien et présentez votre modèle devant le PDG de l’entreprise et les actionnaires.Lorsqu’ils donnent des entrées à votre modèle afin que votre modèle prédit « Pas une transaction de fraude » à chaque fois.C’est clairement un problème parce que de nombreux algorithmes d’apprentissage automatique sont conçus pour maximiser la précision globale.

Maintenant, que se passe-t-il? Vous obtenez 95% de précision, mais votre modèle en prédisant mal à chaque fois?

## ****Qu’est-ce que les ensembles de données équilibrés et déséquilibrés?****

**Ensemble de données équilibré :**

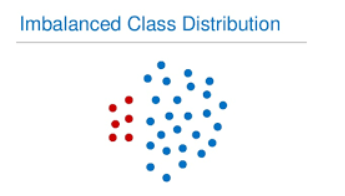
 Prenons un exemple simple si, dans notre ensemble de données, nous avons des valeurs positives qui sont à peu près les mêmes que les valeurs négatives. Ensuite, nous pouvons dire que notre ensemble de données en équilibre



Considérez la couleur orange comme une valeur positive et la couleur bleue comme une valeur négative. Nous pouvons dire que le nombre de valeurs positives et négatives dans à peu près le même.

**Ensemble de données déséquilibré :**

S’il y a la très haute différence entre les valeurs positives et les valeurs négatives. Ensuite, nous pouvons dire notre jeu de données dans Imbalance Dataset.

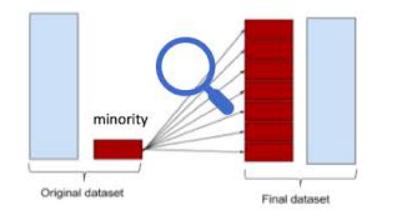


## ****Techniques pour convertir l’ensemble de données déséquilibré en ensemble de données équilibré****

Les données déséquilibrées ne sont pas toujours une mauvaise chose, et dans les ensembles de données réels, il y a toujours un certain degré de déséquilibre. Cela dit, il ne devrait pas y avoir d’impact important sur le rendement de votre modèle si le niveau de déséquilibre est relativement faible.Maintenant, nous allons couvrir quelques techniques pour résoudre le problème de déséquilibre de classe.

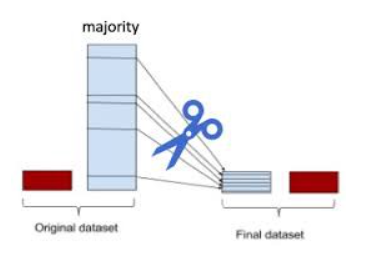
**Suréchantillon (échantillonnage en hausse) :**

 Cette technique est utilisée pour modifier les classes de données inégales afin de créer des ensembles de données équilibrés. Lorsque la quantité de données est insuffisante, la méthode de suréchantillonnement tente d’équilibrer en incrémentant la taille des échantillons rares.Le suréchantillon augmentation du nombre de membres de la catégorie minoritaire dans l’ensemble de formation. L’avantage du suréchantillon est qu’aucune information provenant de l’ensemble de formation initial n’est perdue, car toutes les observations des classes minoritaires et majoritaires sont conservées. D’autre part, il est enclin à trop s’adapter.



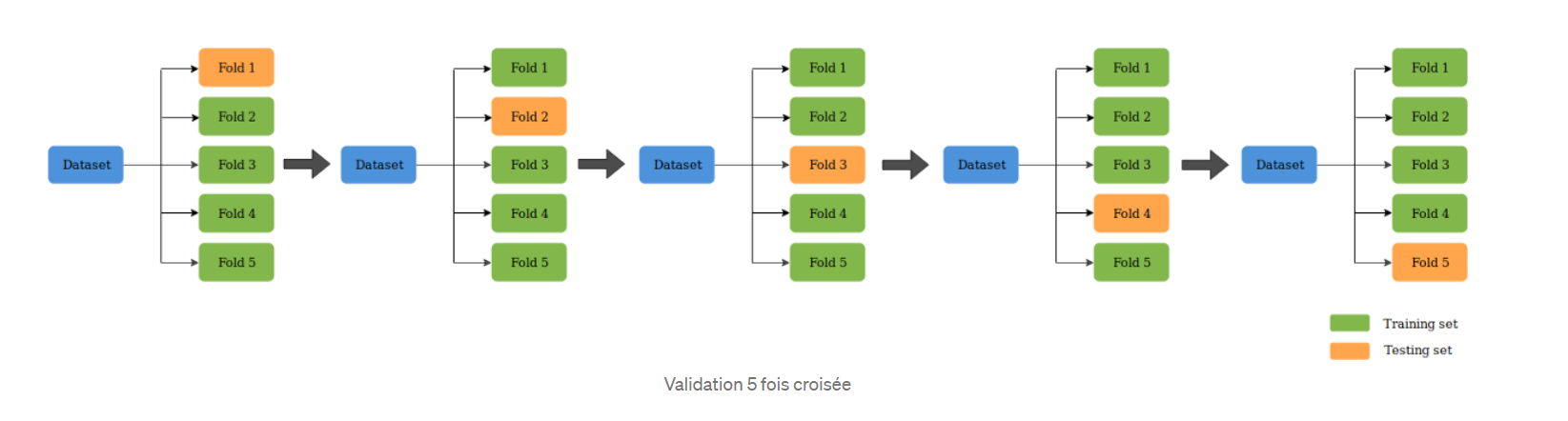
**Sous-échantillonnage (échantillonnage à la baisse) :**

Contrairement au suréchantillon, cette technique équilibre l’ensemble de données de déséquilibre en réduisant la taille de la classe qui est en abondance. Il existe diverses méthodes pour les problèmes de classification tels que les centroïdes de cluster et les liens Tomek. Les méthodes centroïdes de cluster remplacent le groupe d’échantillons par le centroïde de cluster d’un algorithme K-means et la méthode de lien De Tomek élimine le chevauchement non désiré entre les classes jusqu’à ce que tous les voisins les plus proches minimalement éloignés soient de la même classe.Le sous-échantillonnage, contrairement au suréchantillonnement, vise à réduire le nombre d’échantillons majoritaires afin d’équilibrer la répartition des classes. Étant donné qu’il supprime les observations de l’ensemble de données d’origine, il peut rejeter des informations utiles.



**K-flod Validation croisée**

K-Fold CV est l’endroit où un ensemble de données donné est divisé en un nombre K de sections / plis où chaque pli est utilisé comme un ensemble de test à un moment donné. Prenons le scénario de validation croisée 5 fois (K=5). Ici, l’ensemble de données est divisé en 5 plis. Dans la première itération, le premier pli est utilisé pour tester le modèle et le reste est utilisé pour former le modèle. Dans la deuxième itération, le 2ème pli est utilisé comme ensemble de test tandis que le reste sert d’ensemble de formation. Ce processus est répété jusqu’à ce que chaque pli des 5 plis ait été utilisé comme jeu de test.



**Classification**

**Classement**