

# **MÉMOIRE DE RECHERCHE DE FIN D'ÉTUDES**

## **AUTOMATISATION DU MACHINE LEARNING DANS LE CADRE DE L'APPRENTISSAGE SUPERVISÉ AVEC PYTHON**

**Mastère in Artificial Intelligence & Management**

**Etienne KOA**

**2021-2022**

### **RÉSUMÉ :**

Ce mémoire s'inscrit dans le cadre de mon stage et celle de la finalisation et l'obtention de mon master en Intelligence Artificielle. Il s'agit de travailler sur une solution en Intelligence qui permet d'automatiser et sélectionner le modèle adéquat et précis dans le cas des traitements des données en Apprentissage Supervisé avec python. Le gain de temps et la sélection des modèles est une question sans doute majeure dans le cas des traitements. Elle part d'un besoin dans le cadre de mon stage qui était celui à un besoin en peu de temps et de manière fiable.

### **Introduction Générale**

#### **1) Préparation des données dans un projet d'apprentissage automatique.**

La préparation des données est peut-être l'une des étapes les plus difficiles de tout projet d'apprentissage automatique. La raison en est que chaque ensemble de données est différent et hautement spécifique au projet. Néanmoins, il existe suffisamment de points communs entre les projets de modélisation prédictive pour que nous puissions définir une séquence souple d'étapes et de sous-tâches que vous êtes susceptible de réaliser.

Ce processus fournit un contexte dans lequel nous pouvons considérer la préparation des données requises pour le projet, informée à la fois par la définition du projet effectuée avant

la préparation des données et par l'évaluation des algorithmes d'apprentissage machine effectuée après.

Nous essayerons de montrer comment considérer la préparation des données comme une étape dans un projet plus large d'apprentissage automatique de modélisation prédictive.

Chaque projet de modélisation prédictive avec l'apprentissage automatique est différent, mais il existe des étapes communes réalisées sur chaque projet.

La préparation des données implique la meilleure exposition de la structure sous-jacente inconnue du problème aux algorithmes d'apprentissage.

Les étapes avant et après la préparation des données dans un projet peuvent informer sur les méthodes de préparation des données à appliquer, ou du moins à explorer.

Nous essayerons de clarifier les trois parties suivantes :

1. Processus d'apprentissage automatique appliqué
2. Qu'est-ce que la préparation des données ?
3. Comment choisir les techniques de préparation des données

### **a) Processus d'apprentissage automatique appliqué**

Chaque projet d'apprentissage automatique est différent car les données spécifiques au cœur du projet sont différentes.

Les bonnes caractéristiques ne peuvent être définies que dans le contexte du modèle et des données ; étant donné que les données et les modèles sont si diversifiés, il est difficile de généraliser la pratique de l'ingénierie des fonctionnalités dans tous les projets.

Cela rend chaque projet d'apprentissage automatique unique. Personne ne peut vous dire quels sont ou pourraient être les meilleurs résultats ou pourraient être, ou quels algorithmes utiliser pour les obtenir.

Vous devez établir une ligne de base de performance comme point de référence pour comparer tous vos modèles et vous devez découvrir quel algorithme fonctionne le mieux pour votre ensemble de données spécifique. Même si votre projet est unique, les étapes sur la voie d'un bon ou même du meilleur résultat sont généralement les mêmes d'un projet à l'autre.

C'est ce que l'on appelle parfois le processus d'apprentissage automatique appliqué, processus de science des données ou, plus anciennement, découverte de connaissances dans les bases de données (KDD). Le processus d'apprentissage automatique appliqué consiste en une séquence d'étapes.

Les étapes sont les mêmes, mais les noms des étapes et des tâches effectuées peuvent différer d'une description à l'autre.

En outre, les étapes sont écrites de manière séquentielle, mais nous allons sauter d'une étape à l'autre pour un projet donné.

On peut ainsi, définir le processus à l'aide des quatre étapes de haut niveau suivantes

Étape 1 : Définir le problème.

Étape 2 : Préparer les données.

Étape 3 : Évaluation des modèles.

Étape 4 : finaliser le modèle.

Examinons de plus près chacune de ces étapes.

### **Étape 1 : Définition du problème**

Cette étape consiste à en apprendre suffisamment sur le projet pour sélectionner le ou les cadres de la tâche de prédiction.

Par exemple, s'agit-il de classification ou de régression, ou d'un autre type de problème d'ordre supérieur ?

Il s'agit de collecter les données qui sont censées être utiles pour faire une prédiction et de définir clairement la forme que prendra la prédiction. Il peut également s'agir de parler aux parties prenantes du projet et d'autres personnes ayant une expertise approfondie du domaine. Cette étape implique également un examen attentif des données, et peut-être même une exploration des données à l'aide de statistiques sommaires et de la visualisation des données.

### **Étape 2 : Préparation des données**

Cette étape consiste à transformer les données brutes qui ont été collectées en une forme qui peut être utilisée pour la modélisation.

Les techniques de prétraitement des données font généralement référence à l'ajout, la suppression ou la transformation des données de l'ensemble d'entraînement.

### **Étape 3 : Évaluation des modèles**

Cette étape concerne l'évaluation des modèles d'apprentissage automatique sur votre ensemble de données. Elle exige que vous conceviez un ensemble de test robuste utilisé pour évaluer vos modèles afin que les résultats obtenus soient fiables et puissent être utilisés pour sélectionner les modèles que vous avez évalués.

Cela implique des tâches telles que la sélection d'une métrique de performance pour évaluer la compétence d'un modèle, l'établissement d'une base de performance auquel toutes les évaluations de modèles peuvent être comparées, et une technique de rééchantillonnage pour diviser les données en ensembles de formation et de test afin de simuler la façon dont le modèle final sera utilisé.

Pour des estimations rapides et sales de la performance d'un modèle, ou pour un très grand ensemble de données, une seule division des données en ensembles d'entraînement et de

test peut être effectuée. Il est plus courant d'utiliser la validation croisée k-fold comme technique de rééchantillonnage des données, souvent avec des répétitions du processus pour améliorer la robustesse du résultat.

Cette étape comprend également des tâches visant à tirer le meilleur parti des modèles performants, telles que l'ajustement des hyperparamètres et les ensembles de modèles.

#### **Étape 4 : Finalisation du modèle**

Cette étape concerne la sélection et l'utilisation d'un modèle final. Une fois qu'un ensemble de modèles a été évalué, vous devez choisir un modèle qui représente la solution au projet. Cela s'appelle sélection du modèle et peut impliquer une évaluation plus poussée des modèles candidats sur un ensemble de données de validation, ou une sélection selon d'autres critères spécifiques au projet, comme la complexité du modèle. Il peut également s'agir de résumer la performance du modèle d'une manière standard pour les parties prenantes du projet, ce qui est une étape importante.

#### **b) Qu'est-ce que la préparation des données ?**

Dans un projet de modélisation prédictive, tel que la classification ou la régression, les données brutes ne peuvent généralement pas être utilisées directement. Cela est dû à des raisons telles que :

- Les algorithmes d'apprentissage automatique nécessitent que les données soient chiffrées.
- Certains algorithmes d'apprentissage automatique imposent des exigences aux données.
- Le bruit statistique et les erreurs dans les données peuvent devoir être corrigés.
- Des relations non linéaires complexes peuvent être extraites des données.

Ainsi, les données brutes doivent être prétraitées avant d'être utilisées pour ajuster et évaluer un modèle d'apprentissage automatique.

Cette étape d'un projet de modélisation prédictive est appelée préparation des données, bien qu'elle porte de nombreux autres noms, tels que le traitement des données, le nettoyage des données, le prétraitement des données et l'ingénierie des caractéristiques.

Certains de ces noms peuvent être considérés comme des sous-tâches du processus plus large de préparation des données.

Nous pouvons définir la préparation des données comme la transformation des données brutes en une forme plus adaptée à la modélisation.

Ce processus est très spécifique à vos données, aux objectifs de votre projet et aux algorithmes qui seront utilisés pour modéliser vos données.

Nous parlerons davantage de ces relations dans la section suivante.

Néanmoins, il existe des tâches communes ou standard que vous pouvez utiliser ou explorer au cours de l'étape de préparation des données dans un projet d'apprentissage automatique.

Ces tâches comprennent :

- Le nettoyage des données : Identifier et corriger les erreurs ou les fautes dans les données.
- Sélection des caractéristiques : Identification des variables d'entrée les plus pertinentes pour la tâche.
- Transformations des données : Modification de l'échelle ou de la distribution des variables.
- Ingénierie des caractéristiques : Détermination de nouvelles variables à partir des données disponibles.
- Réduction de la dimensionnalité : Créer des projections compactes des données.

Chacune de ces tâches constitue un domaine d'étude à part entière, avec des algorithmes spécialisés. La préparation des données ne se fait pas à l'aveuglette.

Dans certains cas, les variables doivent être codées ou transformées avant que nous puissions appliquer un algorithme d'apprentissage automatique, comme la conversion de chaînes de caractères en nombres.

Dans d'autres cas, c'est moins évident, par exemple : la mise à l'échelle d'une variable peut ou non être utile à un algorithme.

La philosophie générale de la préparation des données consiste à découvrir comment exposer au mieux la structure sous-jacente du problème aux algorithmes d'apprentissage.

Nous ne connaissons pas la structure sous-jacente du problème ; si c'était le cas, nous n'aurions pas besoin d'un algorithme d'apprentissage pour la découvrir et apprendre à la transformer en une solution.

En outre, il se peut que nous devions rechercher de nombreuses représentations alternatives des prédicteurs pour améliorer la performance du modèle.

De plus, différentes variables ou sous-ensembles de variables d'entrée peuvent nécessiter différentes séquences de méthodes de préparation des données.

Le site peut sembler écrasant, étant donné le grand nombre de méthodes, chacune d'entre elles pouvant avoir sa propre configuration et ses propres exigences.

Néanmoins, les étapes du processus d'apprentissage automatique avant et après la préparation des données peuvent aider à déterminer les techniques à envisager.

### **c) Comment choisir les techniques de préparation des données**

Comment savoir quelles techniques de préparation des données utiliser pour nos données ? Comme pour de nombreuses questions de statistiques, la réponse à la question " quelles méthodes d'ingénierie des caractéristiques sont les meilleures ? est que cela dépend.

Plus précisément, cela dépend du modèle utilisé et de la relation réelle avec le résultat.

En apparence, il s'agit d'une question difficile, mais si l'on considère l'étape de préparation des données dans le contexte de l'ensemble du projet, la question devient plus simple.

Les étapes d'un projet de modélisation prédictive avant et après l'étape de préparation des données informent sur la préparation des données qui peut être nécessaire.

L'étape précédant la préparation des données consiste à définir le problème.

Dans le cadre de la définition du problème, cela peut impliquer de nombreuses sous-tâches, telles que :

- Recueillir les données du domaine du problème.
- Discuter du projet avec des experts en la matière.
- Sélectionner les variables à utiliser comme entrées et sorties d'un modèle prédictif.
- Examiner les données qui ont été recueillies.
- Résumer les données recueillies à l'aide de méthodes statistiques.
- Visualiser les données recueillies à l'aide de diagrammes et de graphiques.

Les informations connues sur les données peuvent être utilisées pour sélectionner et configurer les méthodes de préparation des données.

Par exemple, les graphiques des données peuvent aider à identifier si une variable a des valeurs aberrantes. Cela peut faciliter les opérations de nettoyage des données.

Elles peuvent également donner un aperçu de distribution de probabilité qui sous-tend les données. Cela peut aider à déterminer si les transformations de données qui modifient la distribution de probabilité d'une variable sont appropriées.

Les méthodes statistiques, telles que les statistiques descriptives, peuvent être utilisées pour déterminer si des opérations de mise à l'échelle sont nécessaires. Les tests d'hypothèse statistique peuvent être utilisés pour déterminer si une variable correspond à une distribution de probabilité donnée.

Les graphiques et les statistiques par paire peuvent être utilisés pour déterminer si les variables sont liées et, si oui, dans quelle mesure, ce qui permet de savoir si une ou plusieurs variables sont redondantes ou non pertinentes pour la variable cible.

Ainsi, il peut y avoir beaucoup d'interactions entre la définition du problème et la préparation des données.

Il peut également y avoir une interaction entre l'étape de préparation des données et l'évaluation des modèles. L'évaluation du modèle peut impliquer des sous-tâches telles que :

- Sélectionner une métrique de performance pour évaluer la capacité prédictive du modèle.
- Sélectionner une procédure d'évaluation de modèle.

- Sélectionner les algorithmes à évaluer.
- Régler les hyperparamètres des algorithmes.
- Combiner les modèles prédictifs en ensembles.

Les informations connues sur le choix des algorithmes et la découverte des algorithmes performants peuvent également éclairer la sélection et la configuration des méthodes de préparation des données. Par exemple, le choix des algorithmes peut imposer des exigences et des attentes quant au type et à la forme des variables d'entrée dans les données. Cela peut exiger que les variables aient une distribution de probabilité spécifique, la suppression des variables d'entrée corrélées et/ou la suppression des variables qui ne sont pas fortement liées à la variable cible.

Le choix de la métrique de performance peut également nécessiter une préparation minutieuse de la variable cible afin de répondre aux attentes, par exemple en notant les modèles de régression en fonction de l'erreur de prédiction, l'utilisation d'une unité de mesure spécifique, nécessitant l'inversion de toutes les transformations d'échelle appliquées à cette variable pour la modélisation.

Ces exemples, et bien d'autres, soulignent que, bien que la préparation des données soit une étape importante d'un projet de modélisation prédictive, elle ne se suffit pas à elle-même.

Au contraire, elle est fortement influencée par les tâches effectuées avant et après la préparation des données. Cela met en évidence la nature hautement itérative de tout projet de modélisation prédictive.

## **2) Tour d'horizon des techniques de préparation des données**

Les projets d'apprentissage automatique de modélisation prédictive, tels que la classification et la régression, impliquent toujours une certaine forme de préparation des données.

La préparation spécifique des données requise pour un ensemble de données dépend des spécificités des données, comme les types de variables, ainsi que des algorithmes qui seront utilisés pour les modéliser et qui peuvent imposer des attentes ou des exigences aux données.

Néanmoins, il existe une collection d'algorithmes standard de préparation des données qui peuvent être appliqués aux données structurées (par exemple, les données qui forment un grand tableau comme dans une feuille de calcul).

Ces algorithmes de préparation des données peuvent être organisés ou regroupés par type dans un cadre qui peut être utile pour comparer et sélectionner des techniques pour un projet spécifique.

Dans cette section, nous verrons les tâches courantes de préparation des données effectuées dans une tâche d'apprentissage automatique. L'objectif étant, de connaître :

- Les techniques telles que le nettoyage des données peuvent identifier et corriger les erreurs dans les données, comme les valeurs manquantes.
- Les transformations de données peuvent modifier l'échelle, le type et la distribution de probabilité des variables de l'ensemble de données.
- Des techniques telles que la sélection de caractéristiques et la réduction de la dimensionnalité peuvent réduire le nombre de variables d'entrée.

Dans cette partie, nous parlerons de ces 6 parties :

1. Tâches courantes de préparation des données
2. Nettoyage des données
3. Sélection des caractéristiques
4. Transformation des données
5. Ingénierie des caractéristiques
6. Réduction de la dimensionnalité

### **a) Tâches courantes de préparation des données**

Nous pouvons définir la préparation des données comme la transformation des données brutes en une forme plus adaptée à la modélisation.

Néanmoins, il y a des étapes dans un projet de modélisation prédictive avant et après l'étape de préparation des données qui sont importantes.

Avant et après l'étape de préparation des données, certaines étapes d'un projet de modélisation prédictive sont importantes et renseignent sur la préparation des données à effectuer. Le processus d'apprentissage automatique appliqué consiste en une séquence d'étapes. Nous pouvons passer d'une étape à l'autre pour un projet donné, mais tous les projets comportent les mêmes étapes générales :

- Étape 1 : Définir le problème.
- Étape 2 : Préparer les données.
- Étape 3 : évaluation des modèles.
- Étape 4 : finaliser le modèle.

Nous nous intéressons à l'étape de préparation des données (étape 2), et il existe des tâches communes ou standard que vous pouvez utiliser ou explorer pendant l'étape de préparation des données dans un projet d'apprentissage automatique.

Les types de préparation des données effectués dépendent de vos données, comme vous pouvez vous y attendre. Néanmoins, au fur et à mesure que vous travaillez sur plusieurs projets de modélisation prédictive, vous voyez et exigez à nouveau les mêmes types de tâches de préparation des données à plusieurs reprises. Ces tâches comprennent :

- Le nettoyage des données : Identifier et corriger les erreurs ou les fautes dans les données.
- Sélection des caractéristiques : Identification des variables d'entrée les plus pertinentes pour la tâche.



- Transformations des données : Modification de l'échelle ou de la distribution des variables.
- Ingénierie des caractéristiques : Détermination de nouvelles variables à partir des données disponibles.
- Réduction de la dimensionnalité : Créer des projections compactes des données.

Ceci fournit un cadre approximatif que nous pouvons utiliser pour réfléchir et naviguer dans les différents algorithmes de préparation des données que nous pouvons envisager pour un projet donné avec des données structurées ou tabulaires.

## **b) Nettoyage des données**

Le nettoyage des données consiste à corriger des problèmes ou des erreurs systématiques dans des données désordonnées. Le nettoyage de données le plus utile implique une expertise approfondie du domaine et peut impliquer l'identification et le traitement d'observations spécifiques qui peuvent être incorrectes. Il existe de nombreuses raisons pour lesquelles les données peuvent présenter des valeurs incorrectes, comme les erreurs de frappe, la corruption, les doublons, etc.

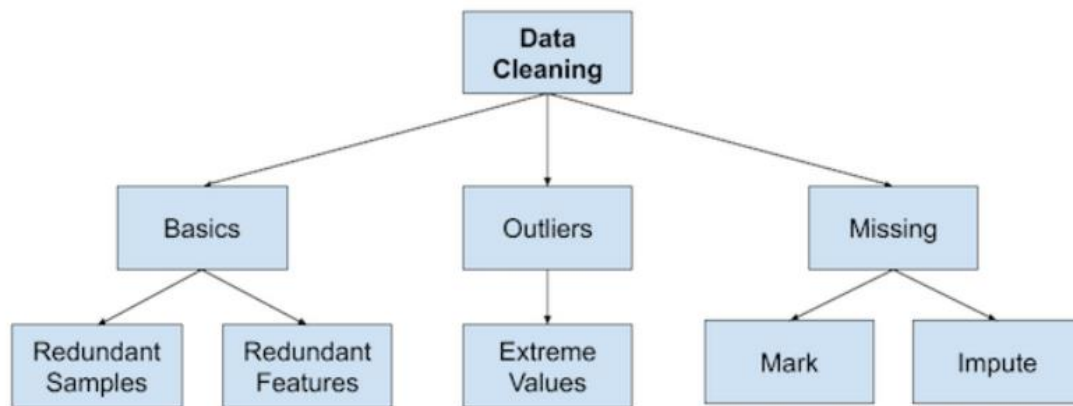
L'expertise du domaine peut permettre d'identifier des observations manifestement erronées car elles sont différentes de ce qui est attendu.

Une fois que les observations désordonnées, bruyantes, corrompues ou erronées sont identifiées, elles peuvent être traitées. Il peut s'agir de supprimer une ligne ou une colonne. Ou bien de remplacer les observations par de nouvelles valeurs. Ainsi, il existe des opérations générales de nettoyage des données qui peuvent être utilisés, comme par exemple

- Utiliser les statistiques pour définir les données normales et identifier les valeurs aberrantes
- Identifier les colonnes qui ont la même valeur ou aucune variance et les supprimer.
- Identifier les lignes de données en double et les supprimer
- Marquer les valeurs vides comme manquantes.
- Imputer les valeurs manquantes en utilisant des statistiques ou un modèle appris.

Le nettoyage des données est une opération qui est généralement effectuée en premier, avant d'autres opérations de préparation des données.

## Overview of Data Cleaning



### c) Sélection des caractéristiques

La sélection des caractéristiques fait référence aux techniques de sélection d'un sous-ensemble de caractéristiques d'entrée qui sont les plus pertinentes pour la variable cible à prédire. Ceci est important car les variables d'entrée non pertinentes et redondantes peuvent distraire ou tromper les algorithmes d'apprentissage, ce qui peut entraîner une baisse de la performance prédictive plus faible.

De plus, il est souhaitable de développer des modèles en utilisant uniquement les données nécessaires à la prédiction. En outre, il est souhaitable de développer des modèles en utilisant uniquement les données nécessaires pour faire une prédiction, par exemple pour favoriser le modèle le plus simple possible et le plus performant.

Les techniques de sélection de caractéristiques peuvent généralement être regroupées en deux catégories : celles qui utilisent la variable cible (supervisées) et celles qui ne le font pas (non supervisées).

En outre, les techniques supervisées peuvent être subdivisées en modèles qui sélectionnent automatiquement les caractéristiques dans le cadre de l'ajustement du modèle (intrinsèque), ceux qui choisissent explicitement les caractéristiques qui aboutissent au modèle le plus performant (wrapper) et ceux qui évaluent chaque caractéristique d'entrée et permettent de sélectionner un sous-ensemble (filtre).

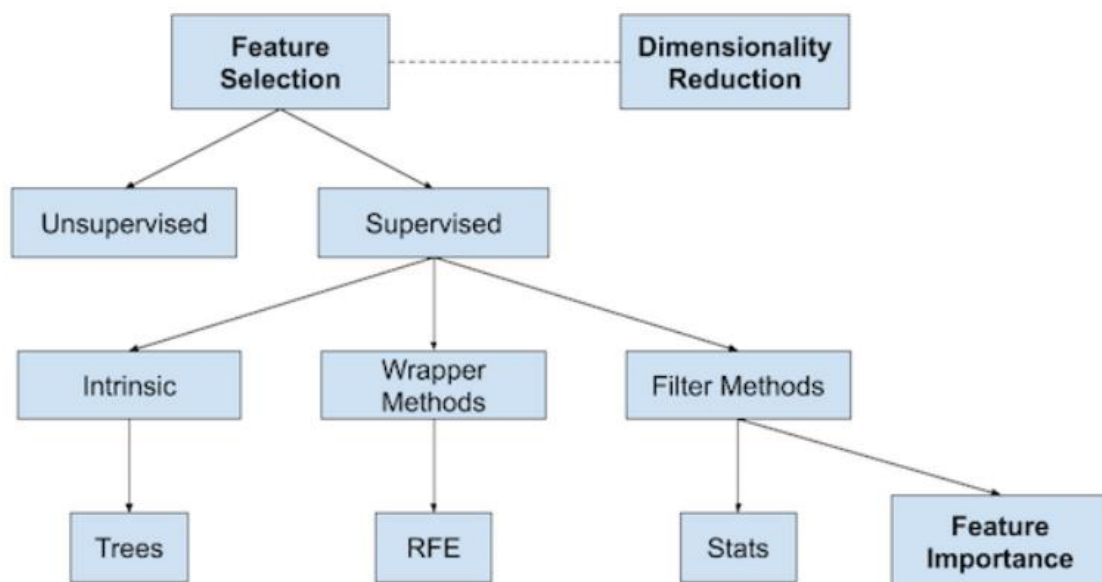
Les méthodes statistiques, telles que la corrélation, sont populaires pour évaluer les caractéristiques d'entrée. Les caractéristiques peuvent ensuite être classées en fonction de leurs scores et un sous-ensemble ayant les scores les plus élevés est utilisé comme entrée d'un modèle. Le choix de la mesure statistique dépend des types de données des variables d'entrée. En outre, il existe différents cas d'utilisation de la sélection de caractéristiques que l'on peut rencontrer dans un projet de modélisation prédictive, comme :

- Des entrées catégorielles pour une variable cible de classification.
- Entrées numériques pour une variable cible de classification.

- Des entrées numériques pour une variable cible de régression.

Lorsqu'un mélange de types de données de variables d'entrée est présent, différentes méthodes de filtrage peuvent être utilisées. Il est également possible d'utiliser une méthode enveloppante, telle que la célèbre méthode Recursive Feature Elimination (RFE) qui ne tient pas compte du type de variable d'entrée. Le domaine plus large de l'évaluation de l'importance relative des caractéristiques d'entrée est appelé " importance des caractéristiques ", dont les résultats peuvent être utilisés pour faciliter l'interprétation du modèle, l'interprétation de l'ensemble de données ou la sélection de caractéristiques pour la modélisation.

#### Overview of Feature Selection Techniques



#### d) Transformations de données

Les transformations de données sont utilisées pour changer le type ou la distribution des variables de données. Il s'agit d'une vaste plage de techniques différentes et elles peuvent être appliquées aussi bien aux variables d'entrée qu'aux variables de sortie. Rappelez-vous que les données peuvent être de plusieurs types, tels que numériques ou catégoriels, avec des sous-types pour chacun d'eux, tels que les valeurs entières et les valeurs réelles à virgule flottante pour les données numériques, et les valeurs nominales, ordinales et booléennes pour les variables d'entrée et de sortie, ordinal et booléen pour les données catégorielles.

Type de données numériques : Valeurs numériques.

- Entier : Entiers sans partie fractionnaire.
- Flottant : Valeurs à virgule flottante.

Type de données catégoriques : Valeurs d'étiquettes.

- Ordinaux : Étiquettes avec un ordre de classement.
- Nominal : Étiquettes sans ordre de classement.
- Booléen : Valeurs True et False.

#### Overview of Data Variable Types

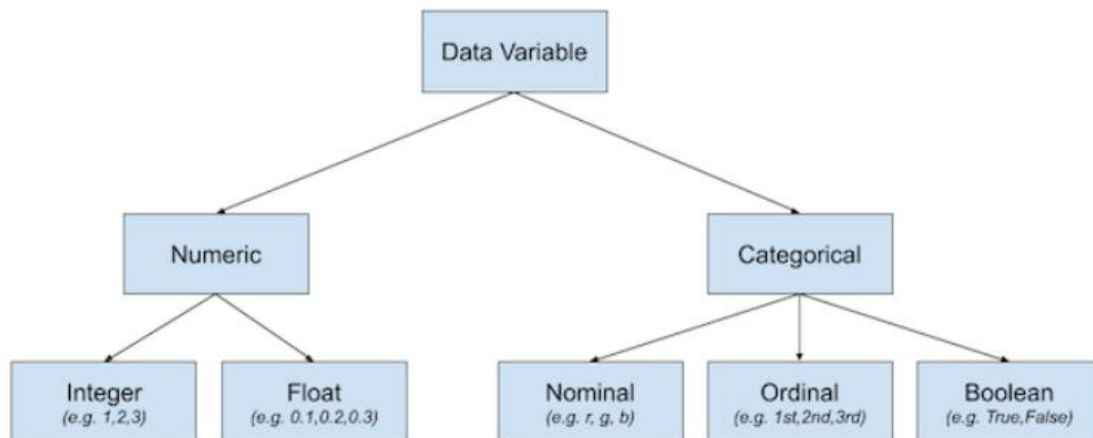


Figure 3.3: Overview of Data Variable Types.

Nous pouvons souhaiter convertir une variable numérique en une variable ordinale dans un processus appelé discrétisation. Nous pouvons également coder une variable catégorielle sous forme d'entiers ou de variables booléennes, qui sont nécessaires dans la plupart des tâches de classification.

- **Transformation de discrétisation** : Encoder une variable numérique en une variable ordinale.
- **Transformation ordinale** : Encode une variable catégorielle en une variable entière.
- **Transformation à un point** : Encode une variable catégorielle en variable binaire.

Pour les variables numériques à valeurs réelles, la façon dont elles sont représentées dans un ordinateur signifie que la résolution est nettement plus élevée dans la plage 0-1 que dans la plage plus large du type de données.

Il peut donc être souhaitable de mettre les variables à l'échelle de cette plage, ce que l'on appelle la normalisation. Si les données ont une distribution de probabilité gaussienne, il peut être plus utile d'adapter les données à une gaussienne standard avec une moyenne de 0 et un écart-type de 1.

- **Transformation de normalisation** : Mettre une variable à l'échelle entre 0 et 1.
- **Transformation de normalisation (ou standardisation)** : Mettre une variable à l'échelle d'une gaussienne standard.

La distribution de probabilité des variables numériques peut être modifiée.

Par exemple, si la distribution est presque gaussienne, mais qu'elle est asymétrique ou décalée, elle peut être rendue plus gaussienne en utilisant une transformation de puissance. Les transformations de quantile peuvent également être utilisées pour forcer une distribution de probabilité, telle qu'une distribution uniforme ou gaussienne, sur une variable dont la distribution naturelle est inhabituelle.

- **Transformation de puissance** : Modifier la distribution d'une variable pour qu'elle soit plus gaussienne.
- **Transformation en quantile** : Impose une distribution de probabilité telle qu'un uniforme ou gaussienne.

Un aspect important des transformations de données est que les opérations sont généralement effectuées séparément pour chaque variable. Ainsi, nous pouvons souhaiter effectuer différentes opérations sur différents types de variables.

Nous pouvons également vouloir utiliser la transformation sur de nouvelles données à l'avenir. Pour ce faire, il suffit d'enregistrer les objets de transformation dans un fichier avec le modèle final formé sur toutes les données disponibles.

Overview of Data Transforms

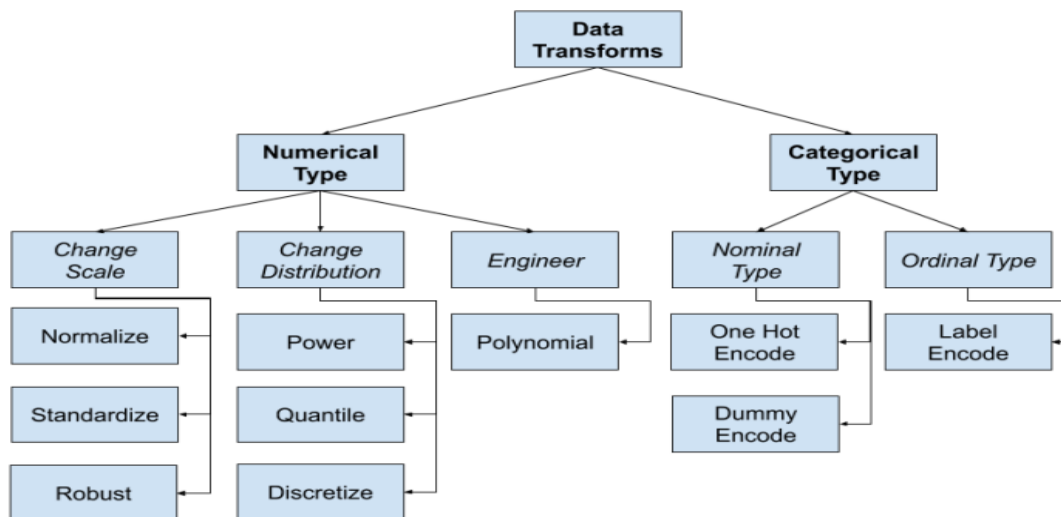


Figure 3.4: Overview of Data Transform Techniques.

## e) Ingénierie des caractéristiques

L'ingénierie des caractéristiques fait référence au processus de création de nouvelles variables d'entrée à partir des données disponibles.

L'ingénierie de nouvelles caractéristiques est très spécifique à vos données et à vos types de données.

En tant que telle, elle nécessite souvent la collaboration d'un expert en la matière pour aider à identifier les nouvelles fonctionnalités qui pourraient être construites à partir des données.

Cette spécialisation en fait un sujet difficile à généraliser à des méthodes générales. Néanmoins, certaines techniques peuvent être réutilisées, par exemple :

- L'ajout d'une variable drapeau booléenne pour un certain état.
- L'ajout d'une statistique sommaire de groupe ou globale, telle qu'une moyenne.
- Ajouter de nouvelles variables pour chaque composant d'une variable composée, telle qu'une date-heure.

Une approche populaire tirée des statistiques consiste à créer des copies de variables d'entrée numériques qui ont été modifiées par une opération mathématique simple, telle que l'élévation à une puissance ou multipliées par d'autres variables d'entrée, appelées caractéristiques polynomiales.

- **Transformation polynomiale** : Créez des copies de variables d'entrée numériques élevées à une puissance.

Le thème de l'ingénierie des caractéristiques est d'ajouter un contexte plus large à une observation unique ou de décomposer une variable complexe, dans le but de fournir une perspective plus simple sur les données d'entrée.

## **f) Réduction de la dimensionnalité**

Le nombre de caractéristiques d'entrée pour un ensemble de données peut être considéré comme la dimensionnalité des données. Par exemple, deux variables d'entrée peuvent définir ensemble un espace bidimensionnel où chaque ligne de données définit un point dans cet espace. Cette idée peut ensuite être mise à l'échelle de n'importe quel nombre de variables d'entrée pour créer de grands hypervolumes multidimensionnels.

Le problème est que plus cet espace a de dimensions (par exemple, plus le nombre de variables d'entrée est élevé), plus il est probable que l'ensemble de données représente un espace très clairsemé et probablement non représentatif. Un échantillonnage très clairsemé et probablement non représentatif de cet espace. C'est ce que l'on appelle la malédiction de la dimensionnalité. C'est ce qui motive la sélection de caractéristiques, bien qu'une alternative à la sélection de caractéristiques soit de créer une projection des données dans un espace de plus faible dimension qui conserve les propriétés les plus importantes des propriétés les plus importantes des données d'origine. C'est ce qu'on appelle généralement la réduction de la dimensionnalité et constitue une alternative à la sélection des caractéristiques. Contrairement à la sélection de caractéristiques, les variables des données projetées ne sont pas directement liées aux variables d'entrée originales, ce qui rend la projection difficile à interpréter. L'approche la plus courante de la réduction de la dimensionnalité consiste à utiliser une technique de factorisation de la matrice.

- L'analyse en composantes principales.

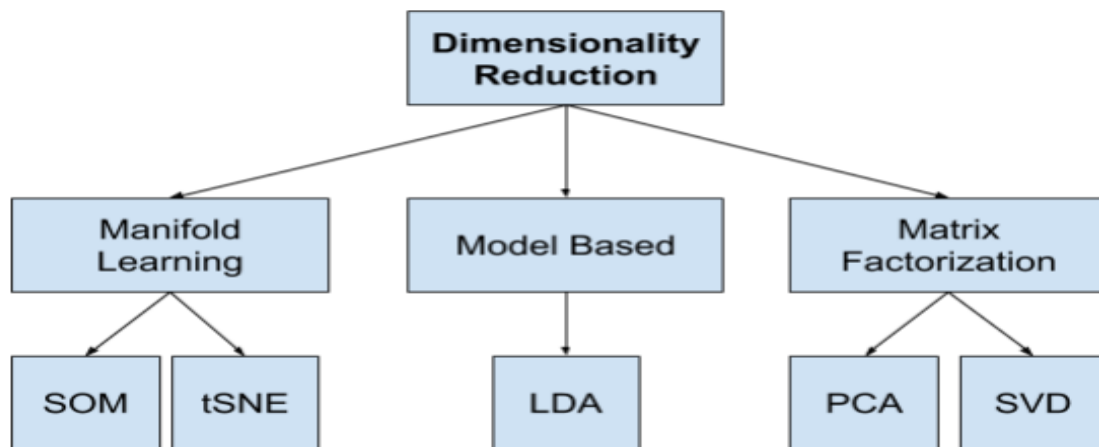
- Décomposition en valeur singulière.

Le principal impact de ces techniques est qu'elles suppriment les dépendances linéaires entre les variables d'entrée, par exemple les variables corrélées. Il existe d'autres approches qui découvrent une réduction de dimensionnalité plus faible.

- Réduction de la dimension. Il s'agit de méthodes basées sur des modèles, telles que l'analyse discriminante linéaire et les auto-encodeurs.
- L'analyse discriminante linéaire.
- 

Parfois, des algorithmes d'apprentissage multidimensionnel peuvent également être utilisés, tels que les cartes auto-organisatrices de Kohonen (SOME) et le t-Distributed Stochastic Neighbor Embedding (t-SNE).

### Overview of Dimensionality Reduction Techniques



## **CHAP I : TOUR D'HORIZON DE L'APPRENTISSAGE SUPERVISE**

### **1) Introduction**

Deux notions sont essentielles avant de pouvoir l'apprentissage supervisé, à savoir :

– la statistique décisionnelle (ou inférentielle) qui utilise les bases de données pour prédire la valeur de variables non observées.

– la statistique descriptive qui a pour but de décrire les liens existants entre les différentes variables observées.

En statistique décisionnelle, les deux types de modèles considérés sont les modèles paramétriques (qui basent leur prédiction sur un nombre de paramètres fini indépendant de la taille de la base de données) et les modèles non paramétriques.

L'apprentissage statistique est la branche non paramétrique de la statistique décisionnelle qui s'intéresse aux bases de données composées de  $n$  **couples**, souvent appelées couples **entrée/sortie**, supposées **indépendants et identiquement distribués**.

Le but d'un algorithme d'apprentissage statistique est de proposer pour toute nouvelle entrée une prédiction de la sortie associée à cette entrée. Les procédures d'apprentissage statistique sont utiles lorsqu'une modélisation paramétrique de la loi générant les données n'est pas accessible ou lorsque la complexité du modèle est telle qu'elle empêche son utilisation pour la prédiction.

Ces méthodes sont devenues incontournables dans de nombreuses applications pratiques (classement et analyse d'images, reconnaissance d'objets, classement de documents textuels (par exemple : spam vs non-spam), diagnostic médical, analyse de séquences génétiques ou de protéines, prédictions du rendement d'actifs financiers, interface cerveau-machine, ...).

## 2) Description formelle et exemples

### 2.1) Problématique

Nous observons une base de données composée de  $n$  couples  $Z_1 = (X_1, Y_1), \dots, Z_n = (X_n, Y_n)$  que nous supposons être des réalisations indépendantes d'une même loi  $\mathbb{P}$  inconnue.

Les  $X_1, \dots, X_n$  appartiennent à un espace  $X$  et s'appellent les **entrées**. Typiquement,  $X = \mathbb{R}^d$  pour un grand entier  $d$ . Les  $Y_1, \dots, Y_n$  appartiennent à un espace  $Y$ , et s'appellent les **sorties**. Typiquement,  $Y$  est fini ou  $Y$  est un sous-ensemble de  $\mathbb{R}$ .

**But de l'apprentissage statistique** : prédire la sortie  $Y$  associée à toute nouvelle entrée  $X$ , où il est sous-entendu que la paire  $(X, Y)$  est une nouvelle réalisation de la loi  $\mathbb{P}$ , cette réalisation étant indépendante des réalisations précédemment observées.

Une fonction de prédictions est une fonction (mesurable) de  $X$  dans  $Y$ . Dans cette partie, nous supposons que toutes les quantités que nous manipulons sont mesurables.

L'ensemble de toutes les fonctions de prédictions est noté  $F(X, Y)$ .

La base de données  $Z_1, \dots, Z_n$  est appelée ensemble d'apprentissage, et sera parfois notée  $Z_1^n$ . Un algorithme d'apprentissage est une fonction qui à tout ensemble d'apprentissage renvoie une fonction de prédictions, i.e. une fonction de l'union  $\bigcup_{n \geq 1} Z_1^n$  dans  $F(X, Y)$ , où  $Z = (X, Y)$ .



C'est un estimateur de "la meilleure" fonction de prédiction, où le terme "meilleure" sera précisé ultérieurement.

Soit  $\ell(y, y')$  la perte encourue lorsque la sortie réelle est  $y$  et la sortie prédite est  $y'$ . **La fonction  $\ell: \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$  est appelée fonction de perte.**

Exemple du classement :

$$\ell(y, y') = 1_{y \neq y'} \text{ (i.e. } \ell(y, y') = 1 \text{ si } y \neq y' \text{ et } \ell(y, y') = 0 \text{ sinon).}$$

Un problème d'apprentissage pour lequel cette fonction de perte utilisée est appelé problème de classement (ou plus couramment par anglicisme classification).

L'ensemble  $\mathcal{Y}$  considéré en classement est le plus souvent fini, voire de cardinal deux en classement binaire.

Exemple de la régression :

$$L_p: \mathcal{Y} = \mathbb{R} \text{ et } \ell(y, y') = |y - y'|^p \text{ où } p \geq 1 \text{ est un réel fixe.}$$

Dans ce cas, on parle de régression  $L_p$ . La tâche d'apprentissage lorsque  $p = 2$  est aussi appelée **régression aux moindres carrés**.

La qualité d'une fonction de prédiction  $g: \mathcal{X} \rightarrow \mathcal{Y}$  est mesurée par son risque (ou erreur de généralisation) :

$$\mathcal{R}(g) = \mathbb{E}[l(Y, g(X))].$$

Le risque est donc l'espérance par rapport à loi  $\mathbb{P}$  de la perte encourue sur la donnée  $(X, Y)$  par la fonction de prédiction  $g$ .

La qualité d'un algorithme d'apprentissage  $\hat{g}_n$ , construit à partir de  $Z_1^n$ , peut être mesurée par son risque moyen  $\mathbb{E}\mathcal{R}[\hat{g}_n]$ , où il est sous-entendu que l'espérance est prise par rapport à la loi de l'ensemble d'apprentissage.

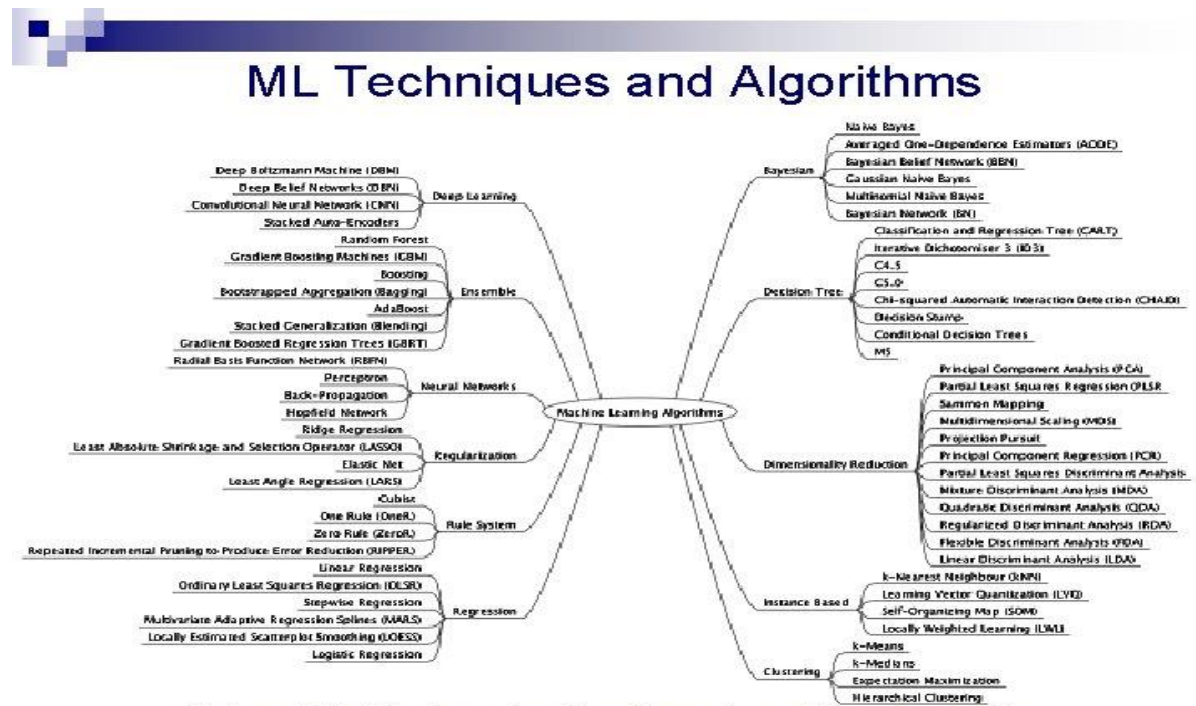
La "**meilleure**" fonction de prédiction est la (ou plus rigoureusement une) fonction de  $F(\mathcal{X}, \mathcal{Y})$  minimisant  $\mathcal{R}$ . Une telle fonction n'existe pas nécessairement mais existe pour les fonctions de pertes usuelles (notamment celles que nous considérerons par la suite). Cette "meilleure" fonction sera appelée **fonction cible**.

**Exemple du classement :**

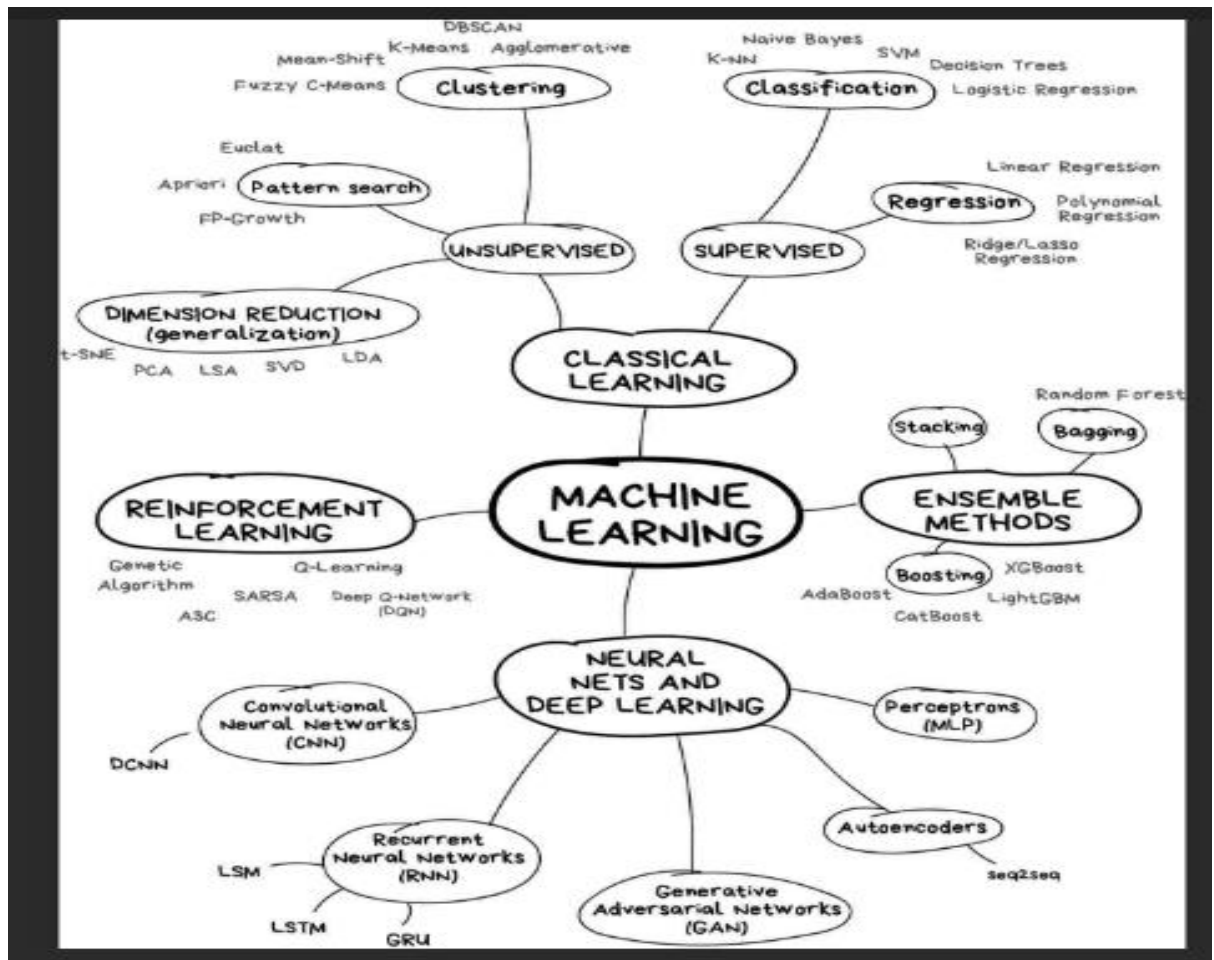
$\ell(y, y') = 1_{y \neq y'}$ . La fonction qui à une entrée  $x$  renvoie la sortie la plus probable (au sens de la distribution conditionnelle de  $Y$  sachant  $X = x$ )  $\mathcal{L}(Y|X = x)$  est "**la**" **fonction cible en classement**.

**Exemple de la régression aux moindres carrés :**

$\mathcal{Y} = \mathbb{R}$  et  $\ell(y, y') = |y - y'|^2$ . La fonction qui à une entrée  $x$  renvoie la sortie moyenne  $\mathbb{E}(Y|X = x)$  est “la” fonction cible en régression aux moindres carrés.



A Tour of Machine Learning Algorithms. Jason Brownlee. 2013.



Présentation générale de quelques modèles.

### 3) Tour d'horizon des métriques d'évaluation des modèles

#### 3.1) Cas de la classification

Un classificateur est aussi bon que la métrique utilisée pour l'évaluer. Si vous choisissez la mauvaise métrique pour l'évaluation de vos modèles, vous risquez de choisir un mauvais modèle ou, dans le pire des cas, d'être trompé sur les performances attendues de votre modèle.

Dans cette partie, nous découvrirons les métriques que vous pouvez utiliser pour la classification. L'objectif de cette partie est de comprendre les parties suivantes :

- Le défi que représente le choix des métriques pour la classification, et comment cela est particulièrement difficile lorsqu'il y a une distribution de classe asymétrique.
- Comment il existe trois principaux types de métriques pour évaluer les modèles de classificateurs, appelés **rang**, **seuil** et **probabilité**.
- Comment choisir une métrique pour la classification déséquilibrée si vous ne savez pas par où commencer.

#### a) Défi des mesures d'évaluation

## Une métrique d'évaluation quantifie la performance d'un modèle prédictif.

Cela implique généralement l'apprentissage d'un modèle sur un ensemble de données, l'utilisation du modèle pour faire des prédictions sur un ensemble de données d'attente qui n'a pas été utilisé pendant la formation, puis de comparer les prédictions aux valeurs attendues dans l'ensemble de données d'attente.

Pour les problèmes de classification, les mesures consistent à comparer l'étiquette de classe attendue à l'étiquette de classe prédite ou à interpréter les probabilités prédites pour les étiquettes de classe du problème.

La sélection de modèle, et même les méthodes de préparation des données, constituent un problème de recherche guidé par la métrique d'évaluation.

Des expériences sont réalisées avec différents modèles et le résultat de chaque expérience est quantifié par une métrique. Les mesures d'évaluation jouent un rôle crucial à la fois pour évaluer les performances de classification et pour guider la modélisation du classificateur.

Il existe des mesures standard largement utilisées pour évaluer les modèles prédictifs de classification, telles que **l'exactitude ou la précision de la classification**. Les métriques standard fonctionnent bien sur la plupart des problèmes, c'est pourquoi elles sont largement adoptées.

Mais toutes les métriques font des hypothèses sur le problème ou sur ce qui est important dans le problème. Il faut donc choisir une métrique d'évaluation qui capture au mieux ce que vous ou les parties prenantes de votre projet pensez être important à propos du modèle ou des prédictions, ce qui rend le choix des métriques d'évaluation de modèle difficile.

### b) Taxonomie des métriques d'évaluation des classificateurs

Il existe des dizaines de métriques à choisir pour évaluer les modèles de classificateurs, et peut-être même des centaines, si l'on considère toutes les versions secondaires des métriques proposées par les universitaires.

Afin d'avoir une idée sur les métriques que vous pouvez choisir, nous allons utiliser une taxonomie proposée par Cesar Ferri dans son article de 2008 intitulé **An Experimental Comparison Of Performance Measures** For classification. Nous pouvons diviser les mesures d'évaluation en trois groupes utiles ; ce sont :

1. **Métriques de seuil**
2. **Métriques de classement**
3. **Métriques de probabilité.**

Cette division est utile car les principales métriques utilisées par les praticiens pour les classificateurs en général. Plusieurs chercheurs en apprentissage automatique ont identifié trois familles de mesures d'évaluation utilisées dans le contexte de la classification. Il s'agit des métriques à seuil (par ex, l'exactitude), les méthodes et les mesures de classement (par exemple, l'analyse des caractéristiques d'exploitation du récepteur (ROC) et les mesures probabilistes (par exemple, l'erreur quadratique moyenne).

## Métriques de seuil pour la classification

**Les métriques de seuil sont celles qui quantifient les erreurs de prédiction de la classification.** En d'autres termes, elles sont conçues pour résumer la fraction, le rapport ou le taux des cas où une classe prédite ne correspond pas à la classe attendue dans un ensemble de données d'attente.

**Les mesures basées sur un seuil** et une compréhension qualitative de l'erreur sont utilisées lorsque nous voulons qu'un modèle minimise le nombre d'erreurs.

**La métrique de seuil** la plus utilisée est peut-être **la précision de classification**.

$$\textit{Exactitude} = \frac{\textit{prédictions correctes}}{\textit{prédictions totales}}$$

Et le complément de la précision de classification appelé **erreur de classification**.

$$\textit{Erreur} = \frac{\textit{prédictions non correctes}}{\textit{prédictions totales}}$$

**La matrice de confusion** fournit un meilleur aperçu non seulement de la performance d'un modèle prédictif, mais aussi les classes qui sont prédites correctement, celles qui sont prédites incorrectement, et le type d'erreurs commises, résumé comme suit :

- **Métriques de sensibilité-spécificité**

**La sensibilité** fait référence au taux de vrais positifs et résume la qualité de la prédiction de la classe positive.

**La spécificité** est le complément de la sensibilité, ou le taux de vrais négatifs, et résume la façon dont la classe négative a été prédite.

- **Mesures de précision et de rappel**

La précision résume la fraction d'exemples affectés à la classe positive qui appartiennent à la classe positive. **Le rappel résume la qualité de la prédiction de la classe positive.**

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) <b>Type II Error</b>	<b>Sensitivity</b> $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) <b>Type I Error</b>	True Negative (TN)	<b>Specificity</b> $\frac{TN}{(TN + FP)}$
		<b>Precision</b> $\frac{TP}{(TP + FP)}$	<b>Negative Predictive Value</b> $\frac{TN}{(TN + FN)}$	<b>Accuracy</b> $\frac{TP + TN}{(TP + TN + FP + FN)}$

### Résumé des métriques de seuil.

#### Mesures de classement pour la classification

**Les métriques de classement** sont plus concernées par l'évaluation des classificateurs en fonction de leur efficacité à séparer les classes.

Les mesures basées sur le classement des exemples par le modèle sont importantes pour de nombreuses applications où les classificateurs sont utilisés pour sélectionner les n meilleures instances d'un ensemble de données ou lorsqu'une bonne séparation des classes est cruciale.

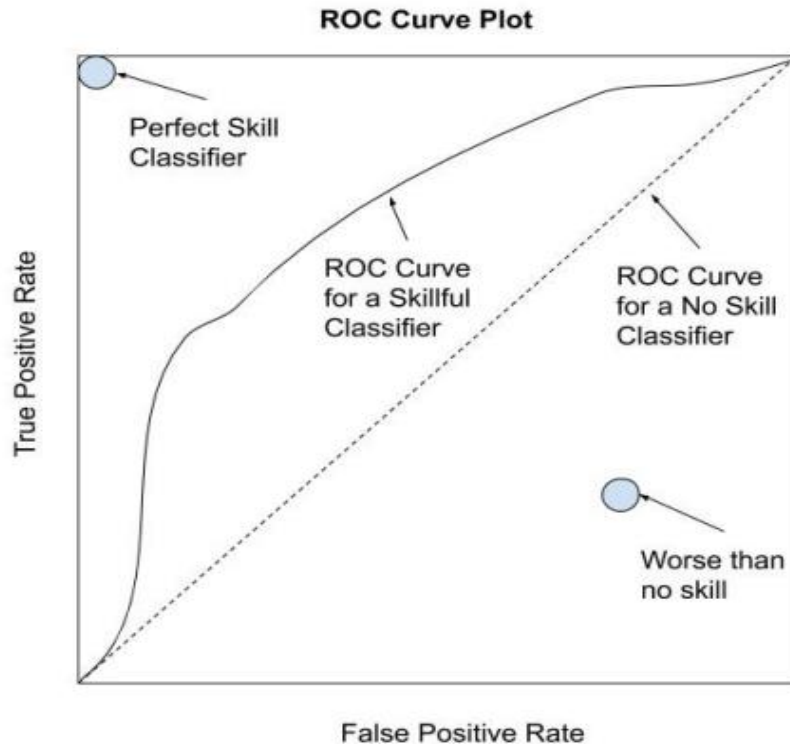
Ces mesures exigent qu'un classificateur prédise un score ou une probabilité d'appartenance à une classe A à partir de ce score, différents seuils peuvent être appliqués pour tester l'efficacité des classifieurs. Les modèles qui maintiennent un bon score à travers une gamme de seuils auront une bonne séparation des classes et seront mieux classés.

**La métrique de classement la plus couramment utilisée est la courbe ROC ou l'analyse ROC.**

ROC est un acronyme qui signifie caractéristique d'exploitation du récepteur et qui résume un champ d'étude pour l'analyse des classificateurs binaires en fonction de leur capacité à discriminer les classes. Une courbe ROC est un graphique de diagnostic permettant de résumer le comportement d'un modèle en calculant le taux de faux positifs et le taux de vrais positifs pour un ensemble de prédictions par le modèle sous différents seuils.

Chaque seuil est un point sur le graphique et les points sont reliés pour former une courbe. Un classificateur qui n'a aucune compétence (c'est-à-dire qui prédit la classe majoritaire sous tous les seuils) sera représenté par une ligne diagonale allant du bas à gauche vers le haut à droite. Tous les points en dessous de cette ligne ont pire qu'aucune compétence. Un modèle parfait sera un point en haut à gauche du graphique.

#### 4.3. Taxonomy of Classifier Evaluation Metrics



- La courbe ROC est un diagnostic utile pour un modèle.
- L'aire sous la courbe ROC peut être calculée et fournit un score unique pour résumer le tracé qui peut être utilisé pour comparer les modèles.
- Un classificateur sans compétence aura un score de 0.5, tandis qu'un classificateur parfait aura un score de 1.0.

*Bien que généralement efficaces, la Courbe ROC et l'AUC ROC peuvent être optimistes dans le cas d'un grave déséquilibre de classe, en particulier lorsque le nombre d'exemples dans la classe minoritaire est faible.*

Une alternative à la courbe ROC est la **courbe précision-rappel** qui peut être utilisée de manière similaire, mais se concentre sur la performance du classificateur sur la classe minoritaire.

Encore une fois, différents seuils sont utilisés sur un ensemble de prédictions par un modèle, et dans ce cas, la précision et le rappel sont calculés.

**Les points forment une courbe** et les classificateurs qui sont plus performants sous une gamme de seuils différents seront classés plus haut.

Un classificateur sans compétence sera une ligne horizontale sur le graphique avec une précision proportionnelle au nombre d'exemples positifs de l'ensemble de données. Pour un ensemble de données équilibré, cette précision sera de 0,5. Un classificateur parfait est représenté par un point en haut à droite.

#### 4.3. Taxonomy of Classifier Evaluation Metrics

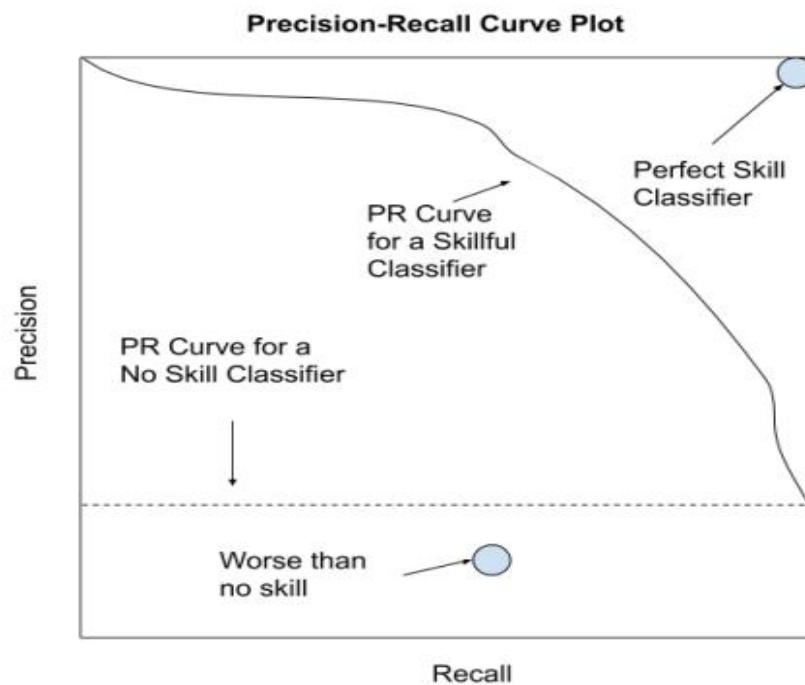
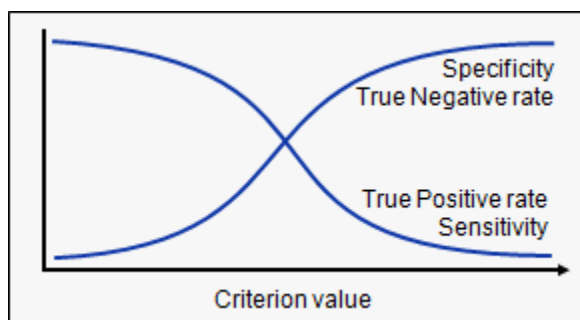


Figure 4.2: Depiction of a Precision-Recall Curve.

Comme la courbe ROC, la courbe précision-rappel est un outil de diagnostic utile pour évaluer un seul classificateur, mais difficile à comparer. Et comme l'AUC de ROC, nous pouvons calculer l'aire sous la courbe comme un score et utiliser ce score pour comparer les classificateurs.



Exemple de courbe AUC.

#### Métriques probabilistes pour la classification déséquilibrée



**Les métriques probabilistes sont conçues spécifiquement pour quantifier l'incertitude des prédictions d'un classificateur.** Elles sont utiles pour les problèmes où l'on s'intéresse moins aux prédictions de classes incorrectes vs. et plus intéressés par l'incertitude du modèle dans les prédictions et la pénalisation des prédictions mais hautement confiantes.

Les métriques basées sur une compréhension probabiliste de l'erreur, c'est-à-dire la mesure de l'écart par rapport à la vraie probabilité. Ces mesures sont particulièrement utiles lorsque l'on veut évaluer la fiabilité des classificateurs, en mesurant non seulement quand ils échouent mais aussi s'ils ont sélectionné la mauvaise classe avec une probabilité élevée ou faible.

Pour évaluer un modèle sur la base des probabilités prédites, il faut que les probabilités soient calibrées. Certains classificateurs sont formés à l'aide d'un cadre probabiliste, tel que l'estimation du maximum de vraisemblance, ce qui signifie que leurs probabilités sont déjà calibrées. Un exemple serait la régression logistique.

De nombreux classificateurs non linéaires ne sont pas entraînés dans un cadre probabiliste et nécessitent donc que leurs probabilités soient calibrées par rapport à un ensemble de données avant d'être évaluées par une mesure probabiliste. **Les exemples peuvent inclure les machines à vecteurs de support et les k-plus proches voisins.**

La métrique la plus courante pour évaluer les probabilités prédites est peut-être la perte logarithmique pour la classification binaire (ou la probabilité logarithmique négative), ou plus généralement l'entropie croisée. Pour un ensemble de données de classification binaire où les valeurs attendues sont  $y$  et les valeurs prédites sont  $y_{pred}$ , cela peut être calculé comme suit :

$$logloss = -((1 - y) \log(1 - y_{pred}) + \log(y_{pred}))$$

Le score peut être généralisé à plusieurs classes en ajoutant simplement les termes ; par exemple :

$$logloss = - \sum_{c \in C} y_c \log(y_{pred_c})$$

### c) Métrique d'évaluation de la régression

*Les métriques d'évaluation jouent un rôle très important dans la construction de tout modèle d'apprentissage automatique. L'erreur quadratique moyenne, l'erreur quadratique moyenne racine, l'erreur absolue moyenne, le R-carré et le R-carré ajusté sont utilisés pour évaluer les performances du modèle dans les algorithmes de régression.*

- **MSE / Perte quadratique / Perte L2 :**

L'erreur quadratique moyenne, ou perte MSE, est la perte par défaut à utiliser pour les problèmes de régression. Mathématiquement, c'est la fonction de perte préférée dans le

cadre d'inférence du maximum de vraisemblance si la distribution de la variable cible est gaussienne. C'est la fonction de perte qui doit être évaluée en premier et modifiée uniquement si vous avez une bonne raison.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

Où :

- $N$  : est le nombre de points de données
- $\hat{y}_i$  : la valeur renvoyée par le modèle
- $y_i$  : la valeur réelle pour le point de données  $i$ .

MSE est calculé comme la moyenne des différences au carré entre les valeurs prévues et réelles. Le résultat est toujours positif quel que soit le signe des valeurs prédites et réelles et une valeur parfaite est 0,0. La mise au carré signifie que des erreurs plus importantes entraînent plus d'erreurs que des erreurs similaires, ce qui signifie que le modèle est puni pour avoir commis des erreurs plus importantes.

- **Erreur absolue moyenne / Perte L1 :**

Sur certains problèmes de régression, la distribution de la variable cible peut être principalement gaussienne mais peut avoir de nombreuses valeurs aberrantes, par exemple des valeurs grandes ou petites éloignées de la valeur moyenne.

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

Où :

- $N$  : est le nombre de points de données
- $\hat{y}_i$  : la valeur renvoyée par le modèle
- $y_i$  : la valeur réelle pour le point de données  $i$ .

MAE, d'autre part, est mesuré comme la somme moyenne de la différence absolue entre les prédictions et les observations réelles. Comme MSE, cela mesure également l'ampleur de l'erreur sans tenir compte de leur direction. Contrairement à MSE, MAE a besoin d'outils plus compliqués tels que la programmation linéaire pour calculer les gradients. MAE est plus robuste aux valeurs aberrantes car il n'utilise pas de carré.

- **La racine de l'erreur quadratique moyenne (RMSE)** n'est rien d'autre que la racine carrée de l'erreur quadratique moyenne que nous avons calculée précédemment. Elle mesure la racine de la variance, c'est-à-dire l'écart-type des résidus.

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

- **Le R-carré** mesure le degré auquel la variance de la variable dépendante peut être expliquée par les variables indépendantes (caractéristiques). Il sera toujours compris entre 0 et 1, plus le R-carré est élevé, meilleur est mon modèle. Il s'agit d'une mesure relative qui peut être utilisée pour comparer lorsque nous appliquons plus d'un algorithme.

$$R^2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2}$$

- **Le R<sup>2</sup> ajusté** est une légère modification de la valeur du R<sup>2</sup>, il mesure la variance de la variable cible, expliquée uniquement par les caractéristiques qui sont utiles pour

faire des prédictions, il sera toujours inférieur ou égal au  $R^2$  car il vous pénalisera pour avoir ajouté des caractéristiques qui ne sont pas utiles pour faire des prédictions.

Dans la formule ci-dessous,  $n$  est le nombre de lignes dans les données et  $k$  est le nombre de colonnes dans les données.

$$R^2_{adj} = 1 - \left[ \frac{(1 - R^2)(n - 1)}{n - k - 1} \right]$$

### Que choisir entre MSE, RMSE, MAE et MAPE ?

Lorsque vous devez pénaliser les valeurs aberrantes présentes dans l'ensemble de données, vous utilisez soit **MSE** soit **RMSE**. Personnellement, il est préférable de choisir **RMSE** à **MSE** car ce dernier n'a pas la même unité que ma valeur prédite.

- Utilisez **MAE** lorsque vos prédictions ne sont pas affectées par les valeurs aberrantes car **MAE** est robuste aux valeurs aberrantes.
- **RMSE** et **MSE** sont tous deux différentiables et sont généralement préférés à **MAE**, car **MAE** n'est pas différentiable lorsque la valeur réelle = la valeur prédite.
- Lorsque vos valeurs moyennes prédites sont importantes (en milliers ou en millions), choisissez **MAPE** (pourcentage d'erreur absolu moyen) plutôt que les autres, car les valeurs d'erreur très importantes peuvent être trompeuses.
- **R-Squared** est une mesure relative et devrait être préféré pour comparer les modèles lorsque nous appliquons plus d'un algorithme.
- **Le R-carré** et la mesure de l'erreur contiennent tous deux des informations importantes, car le premier nous renseigne sur la relation entre les variables indépendantes et les variables dépendantes et le second sur la proximité des valeurs réelles et prédites.

Nous avons suffisamment d'outils afin d'appréhender la notion d'AutoML & AutoEDA, dans le cadre de l'apprentissage supervisé.

# **CHAP II : AutoEDA et AutoML**

## **Introduction**

### **Besoin d'automatiser l'analyse exploratoire des données**

Le mouvement élargi des clients sur le Web, les instruments raffinés pour filtrer le trafic Web, la multiplication des téléphones portables, les gadgets Web et les capteurs IoT sont les éléments essentiels qui accélèrent le rythme de l'ère de l'information de nos jours. À l'ère de l'informatique, les associations de toutes tailles comprennent que l'information peut jouer un rôle crucial dans l'amélioration de leurs compétences, de leur rentabilité et de leurs capacités dynamiques, ce qui entraîne des transactions, des revenus et des avantages accrus.

De nos jours, la plupart des organisations abordent d'immenses ensembles de données, mais le fait de n'avoir que d'énormes mesures d'informations n'améliorent pas l'entreprise, sauf si les entreprises étudient les données accessibles et conduisent un développement faisant autorité. Dans le cycle de vie d'un projet de science des données ou de tout projet d'apprentissage automatique, plus de 60 % de votre temps est consacré à des tâches telles que l'analyse de données, la sélection de fonctionnalités, l'ingénierie de fonctionnalités, etc. Parce que c'est la partie ou l'épine dorsale la plus importante d'un projet de science des données, est cette partie particulière où vous devez effectuer de nombreuses activités telles que le nettoyage des données, la gestion des valeurs manquantes, la gestion des valeurs aberrantes, la gestion des ensembles de données déséquilibrés, la gestion des fonctionnalités catégorielles, etc. Donc, si vous voulez gagner du temps dans l'analyse exploratoire des données, nous pouvons utiliser des bibliothèques python comme dtale, pandas profiling, sweetviz et autoviz pour automatiser nos tâches.

### Description des données :

Nos données proviennent de l'entreprise où je suis en stage, à savoir Caplogy. Il s'agit d'une start-up qui offre ses services aux différents : caplogy, innovation, data, novatiel. Les données étudiées représentent le bilan annuel des intervenants de caplogy auprès des différents pôles au cours de l'année 2022.

Noms des colonnes	Description
Intervenants	Ensemble des intervenants de Caplogy
Intervenants_reel	Ensemble des intervenants ayant eu à intervenir sur différents secteurs à Caplogy
Matiere	Matière enseignée aux seins des différents écoles partenaires de Caplogy
Campus	Noms des différents campus ou écoles
Year	Année 2022
Month	Mois de l'année 2022
Week	Rang trimestriel au cours de l'année 2022
Total hours	Nombre total d'interventions effectués par chaque intervenant sur chaque campus.
Type_d'intervention	Type d'interventions : TP – TD – Cours...etc.
Referencie	Services proposés par secteur : caplogy, innovation...etc.
Date	Répartition des jours sur l'ensemble des mois de l'années 2022.

## Automate EDA Library & Descriptions

Library	Descriptions
<ul style="list-style-type: none"><li>Pandas_Profiling</li></ul>	<ul style="list-style-type: none"><li><b>Pandas_profiling</b> génère des rapports de profil à partir d'un pandas DataFrame.</li><li>La fonction <b>pandas df. describe()</b> est pratique mais un peu basique pour l'analyse exploratoire des données.</li><li><b>Pandas_profiling</b> étend pandas DataFrame avec <b>df.profile_report()</b>, qui génère automatiquement un rapport univarié et multivarié standardisé pour la compréhension des données.</li><li>Pour chaque colonne, les informations suivantes (si elles sont pertinentes pour le type de colonne) sont présentées dans un rapport HTML interactif.</li><li>Le rapport contient trois sections supplémentaires :<ul style="list-style-type: none"><li><b>Vue d'ensemble</b> : détails principalement globaux sur l'ensemble de données (nombre d'enregistrements, nombre de variables, erreurs et doublons globaux, empreinte mémoire)</li><li><b>Alertes</b> : une liste complète et automatique des problèmes potentiels de qualité des données (forte corrélation, asymétrie, uniformité, zéros, valeurs manquantes, valeurs constantes, entre autres)</li><li><b>Reproduction</b> : détails techniques sur l'analyse (heure, version et configuration)</li></ul></li></ul>
<ul style="list-style-type: none"><li>Sweetviz</li></ul>	<ul style="list-style-type: none"><li><b>Sweetviz est une bibliothèque Python open-source qui génère de belles visualisations haute densité pour démarrer l'EDA (Exploratory Data Analysis) avec seulement deux lignes de code.</b></li><li>La sortie est une application HTML entièrement autonome.</li><li>Le système est construit autour de la visualisation rapide des valeurs cibles et de la comparaison des ensembles de données.</li><li>Son objectif est d'aider à une analyse rapide des caractéristiques cibles, des données de formation par rapport aux tests et d'autres tâches de caractérisation des données.</li><li>Fonctionnalités<ul style="list-style-type: none"><li>Analyse cible</li><li>Visualisez et comparez</li><li>Associations de type mixte</li><li>Inférence de type</li><li>Analyse numérique</li></ul></li></ul>
<ul style="list-style-type: none"><li>AutoViz</li></ul>	<ul style="list-style-type: none"><li>Visualisez automatiquement n'importe quel jeu de données, de n'importe quelle taille avec une seule ligne de code.</li></ul>

	<ul style="list-style-type: none"> <li>Vous pouvez maintenant enregistrer automatiquement ces graphiques interactifs sous forme de fichiers HTML.</li> </ul>
<ul style="list-style-type: none"> <li>Dataprep</li> </ul>	<ul style="list-style-type: none"> <li><b>DataPrep permet de préparer vos données à l'aide d'une seule bibliothèque avec quelques lignes de code.</b></li> <li>Actuellement, vous pouvez utiliser DataPrep pour : <ul style="list-style-type: none"> <li>Collecter des données à partir de sources de données communes (via dataprep.connector)</li> <li>Effectuez votre analyse exploratoire des données (via dataprep.eda)</li> <li>Nettoyer et normaliser les données (via dataprep.clean)</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>data-purifier</li> </ul>	<ul style="list-style-type: none"> <li>Une bibliothèque Python pour l'analyse exploratoire automatisée des données, le nettoyage automatisé des données et le prétraitement automatisé des données pour les applications d'apprentissage automatique et de traitement du langage naturel en Python.</li> </ul>
<ul style="list-style-type: none"> <li>Dora</li> </ul>	<ul style="list-style-type: none"> <li>Dora est une bibliothèque Python conçue pour automatiser les parties douloureuses de l'analyse exploratoire des données.</li> <li>La bibliothèque contient des fonctions pratiques pour le nettoyage des données, la sélection et l'extraction des fonctionnalités, la visualisation, le partitionnement des données pour la validation du modèle et les transformations de version des données.</li> <li>La bibliothèque utilise et est destinée à être un complément utile aux outils d'analyse de données Python courants tels que pandas, scikit-learn et matplotlib.</li> </ul>
<ul style="list-style-type: none"> <li>Bamboolib</li> </ul>	<p>Bamboolib est une <b>interface graphique pour pandas DataFrames</b> qui permet à quiconque de travailler avec Python dans Jupyter Notebook ou JupyterLab :</p> <ul style="list-style-type: none"> <li>Interface graphique intuitive qui <b>exporte du code Python</b></li> <li>Prend en charge toutes les transformations et visualisations courantes</li> <li>Les transformations sont livrées avec <b>un contrôle complet du clavier</b></li> <li>Fournit des analyses des meilleures pratiques pour l'exploration des données</li> <li>Ajoutez des transformations personnalisées, des visualisations et des chargeurs de données via de simples <b>plugins Python</b></li> </ul>
<ul style="list-style-type: none"> <li>Klib</li> </ul>	<p>klib.describe : fonctions de visualisation des ensembles de données</p> <ul style="list-style-type: none"> <li><b>klib.cat_plot(df)</b> : renvoie une visualisation du nombre et de la fréquence des caractéristiques catégorielles.</li> <li><b>klib.corr_mat(df)</b> : renvoie une matrice de corrélation codée en couleur</li> <li><b>klib.corr_plot(df)</b> : renvoie une carte thermique codée en couleur, idéale pour les corrélations.</li> </ul>



	<ul style="list-style-type: none"> <li>• <b>klib.dist_plot(df)</b> : renvoie un graphique de distribution pour chaque caractéristique numérique</li> </ul> <p>klib.clean - fonctions pour nettoyer les ensembles de données</p> <ul style="list-style-type: none"> <li>• <b>klib.data_cleaning(df)</b></li> <li>• <b>klib.clean_column_names(df)</b></li> <li>• <b>klib.convert_datatypes(df)</b></li> <li>• <b>klib.drop_missing(df)</b></li> </ul>
<ul style="list-style-type: none"> <li>• Dtale</li> </ul>	<ul style="list-style-type: none"> <li>• Il s'agit d'une bibliothèque lancée en février 2020 qui nous permet de visualiser facilement la trame de données des pandas.</li> <li>• Il possède de nombreuses fonctionnalités très pratiques pour l'analyse exploratoire des données. Il est réalisé à l'aide du backend flask et réagit au frontend.</li> <li>• Il prend en charge les tracés interactifs, les tracés 3D, les cartes thermiques, la corrélation entre les fonctionnalités, crée des colonnes personnalisées et bien d'autres. C'est l'une des meilleures librairies actuelles.</li> </ul>
<ul style="list-style-type: none"> <li>• Mito</li> </ul>	<p>Mito est une feuille de calcul conçue pour accélérer les analyses Python. Lorsque vous modifiez la Mitosheet, le code Python est généré pour vous. Voici les avantages du package Mito en Python :</p> <ul style="list-style-type: none"> <li>• <b>Tableaux croisés dynamiques de type Excel en Python</b></li> <li>• <b>Analysez vos données rapidement</b></li> <li>• <b>Suivez et communiquez votre analyse</b></li> </ul> <p>Voici les types d'opérations d'analyse de données qui peuvent être effectuées à l'aide de Mito en Python :</p> <ul style="list-style-type: none"> <li>• Graphiques exploratoires</li> <li>• Tableaux croisés dynamiques</li> <li>• Fusion de trames de données</li> <li>• Formules de feuille de calcul</li> <li>• Exploration de données</li> <li>• Filtrage des colonnes</li> </ul>
<ul style="list-style-type: none"> <li>• Data_dashboarder</li> </ul>	<ul style="list-style-type: none"> <li>• La bibliothèque vous permet de créer un tableau de bord HTML visualisant non seulement les données et les relations entre les fonctionnalités, mais également de rechercher automatiquement le meilleur modèle compatible avec sklearn "de base".</li> </ul>

## Automate ML Library & Descriptions

Librairies	Descriptions
<ul style="list-style-type: none"><li>• <b>Lazypredict</b></li></ul>	<ul style="list-style-type: none"><li>• <b>Lazy Predict</b> permet de créer de nombreux modèles de base sans trop de code et aide à comprendre quels modèles fonctionnent le mieux sans aucun réglage de paramètre.</li></ul>
<ul style="list-style-type: none"><li>• <b>Auto_ViML</b></li></ul>	<ul style="list-style-type: none"><li>• <b>Auto_ViML</b> a été conçu pour créer des modèles interprétables hautes performances avec le moins de variables nécessaires.</li><li>• Le "V" dans Auto_ViML signifie Variant car il essaie plusieurs modèles avec plusieurs fonctionnalités pour vous trouver le modèle le plus performant pour votre jeu de données. Le "i" dans Auto_ViML signifie "interprétable" car Auto_ViML sélectionne le moins de fonctionnalités nécessaires pour construire un modèle plus simple et plus interprétable.</li><li>• Dans la plupart des cas, Auto_ViML construit des modèles avec 20 % à 99 % de fonctionnalités en moins qu'un modèle performant similaire avec toutes les fonctionnalités incluses (ceci est basé sur mes essais. Votre expérience peut varier).</li></ul>
<ul style="list-style-type: none"><li>• <b>H2O</b></li></ul>	<ul style="list-style-type: none"><li>• H2O fait faire des maths à Hadoop ! H2O met à l'échelle les statistiques, l'apprentissage automatique et les mathématiques sur BigData.</li><li>• H2O est extensible et les utilisateurs peuvent construire des blocs en utilisant de simples legos mathématiques dans le noyau. H2O conserve des interfaces familières telles que python, R, Excel et JSON afin que les passionnés et les experts du BigData puissent explorer, modifier, modéliser et évaluer des ensembles de données à l'aide d'une gamme d'algorithmes simples à avancés.</li><li>• La collecte des données est facile. La prise de décision est difficile. H2O permet d'obtenir rapidement et facilement des informations à partir de vos données grâce à une modélisation prédictive plus rapide et de meilleure qualité. H2O a une vision du scoring et de la modélisation en ligne sur une plateforme unique.</li></ul>

<ul style="list-style-type: none"> <li>• <b>Outil d'optimisation de pipeline basé sur l'arborescence (TPOT)</b></li> </ul>	<ul style="list-style-type: none"> <li>• <b>TPOT</b> signifie <b>T</b>ree-<b>b</b>ased Pipeline <b>O</b>ptimization <b>T</b>ool. Considérez TPOT comme votre <b>assistant en science des données</b>. TPOT est un outil Python d'apprentissage automatique automatisé qui optimise les pipelines d'apprentissage automatique à l'aide de la programmation génétique.</li> <li>• Une fois que TPOT a terminé la recherche (ou que vous en avez assez d'attendre), il vous fournit le code Python du meilleur pipeline qu'il a trouvé afin que vous puissiez bricoler le pipeline à partir de là.</li> </ul>
<ul style="list-style-type: none"> <li>• <b>AutoSklearn</b></li> </ul>	<ul style="list-style-type: none"> <li>• <b>Auto-sklearn</b> libère un utilisateur d'apprentissage automatique de la sélection d'algorithmes et du réglage des hyperparamètres. Il exploite les avantages récents de l'optimisation bayésienne, du méta-apprentissage et de la construction d'ensembles.</li> </ul>
<ul style="list-style-type: none"> <li>• <b>LigthAutoML</b></li> </ul>	<p><b>LightAutoML</b> est une bibliothèque Python open source destinée à l'apprentissage automatique automatisé. Il est conçu pour être léger et efficace pour diverses tâches avec des données textuelles tabulaires. <b>LightAutoML</b> fournit une création de pipeline facile à utiliser, qui permet :</p> <ul style="list-style-type: none"> <li>• Réglage automatique des hyperparamètres, traitement des données.</li> <li>• Saisie automatique, sélection de fonctionnalités.</li> <li>• Utilisation automatique du temps.</li> <li>• Création de rapport automatique.</li> <li>• Schéma modulaire facile à utiliser pour créer vos propres pipelines.</li> </ul>
<ul style="list-style-type: none"> <li>• <b>Hyperpot (Optimisation d'hyperparamètres asynchrones distribués)</b></li> </ul>	<p>L'optimisation séquentielle basée sur un modèle (également connue sous le nom d'optimisation bayésienne) est l'une des méthodes les plus efficaces (par évaluation de fonction) de minimisation de fonction.</p> <p>Cette efficacité le rend approprié pour optimiser les hyperparamètres des algorithmes d'apprentissage automatique qui sont lents à s'entraîner. La bibliothèque Hyperopt fournit des algorithmes et une infrastructure de parallélisation pour effectuer l'optimisation des hyperparamètres (sélection de modèle) en Python.</p>
<ul style="list-style-type: none"> <li>• <b>EvalML</b></li> </ul>	<p>EvalML est une bibliothèque AutoML qui crée, optimise et évalue des pipelines d'apprentissage automatique à l'aide de fonctions d'objectif spécifiques à un domaine.</p> <p><b>Fonctionnalité clé</b></p> <ul style="list-style-type: none"> <li>• <b>Automatisation</b> - Facilite l'apprentissage automatique. Évitez d'entraîner et de régler les modèles à la main. Comprend des vérifications de la qualité des données, une validation croisée et plus encore.</li> <li>• <b>Vérifications des données</b> - Détecte et avertit des problèmes avec vos données et la configuration des problèmes avant la modélisation.</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>De bout en bout</b> - Construit et optimise les pipelines qui incluent un prétraitement de pointe, l'ingénierie des fonctionnalités, la sélection des fonctionnalités et une variété de techniques de modélisation.</li> <li>• <b>Compréhension du modèle</b> - Fournit des outils pour comprendre et introspecter les modèles, pour apprendre comment ils se comporteront dans votre domaine problématique.</li> <li>• <b>Spécifique au domaine</b> - Comprend un référentiel de fonctions d'objectif spécifiques au domaine et une interface pour définir la vôtre.</li> </ul>
<ul style="list-style-type: none"> <li>• <b>Autogluon</b></li> </ul>	<ul style="list-style-type: none"> <li>• <b>AutoGluon</b> automatise les tâches d'apprentissage automatique vous permettant d'obtenir facilement de fortes performances prédictives dans vos applications. Avec seulement quelques lignes de code, vous pouvez former et déployer des modèles d'apprentissage automatique et d'apprentissage en profondeur de haute précision sur des images, du texte et des données tabulaires.</li> </ul>
<ul style="list-style-type: none"> <li>• <b>Pycaret</b></li> </ul>	<ul style="list-style-type: none"> <li>• PyCaret est une bibliothèque d'apprentissage automatique à code source libre en Python qui automatise les workflows d'apprentissage automatique. Il s'agit d'un outil d'apprentissage automatique et de gestion de modèles de bout en bout qui accélère le cycle d'expérimentation de manière exponentielle et vous rend plus productif.</li> <li>• En comparaison avec les autres bibliothèques d'apprentissage automatique open source, PyCaret est une bibliothèque alternative low-code qui peut être utilisée pour remplacer des centaines de lignes de code par quelques lignes seulement. Cela rend les expériences exponentiellement rapides et efficaces. PyCaret est essentiellement un wrapper Python autour de plusieurs bibliothèques et frameworks d'apprentissage automatique tels que scikit-learn, XGBoost, LightGBM, CatBoost, spaCy, Optuna, Hyperopt, Ray et quelques autres.</li> <li>• La conception et la simplicité de PyCaret s'inspirent du rôle émergent des scientifiques citoyens des données, un terme utilisé pour la première fois par Gartner. Les Citizen Data Scientists sont des utilisateurs expérimentés qui peuvent effectuer des tâches analytiques simples et modérément sophistiquées qui auraient auparavant nécessité une plus grande expertise technique.</li> </ul>

### Variable Cible Cas de la Régression & Classification

Modèle	Cible
Régression	Total Hours
Classification	Referencie

### Visualisations des résultats de l'AutoEDA

1) Pandas\_Profiling

DataPrep Report

Overview

Variables

Interactions

Correlations

### Overview

#### Dataset Statistics

Number of Variables	11
Number of Rows	2168
Missing Cells	0
Missing Cells (%)	0.0%
Duplicate Rows	72
Duplicate Rows (%)	3.3%
Total Size in Memory	1003.5 KB
Average Row Size in Memory	474.0 B
Variable Types	Categorical: 8 Numerical: 3

#### Dataset Info

**Total hours** is skewed

Dataset has 72 (3.32%) duplicate rows

**Matiere** has a high cardinality: 75 values

**Year** has constant value "2022"

**Year** has constant length 6

**Month** has constant length 3

Sort by

Feature order

☐ Reverse order

Intervenants

categorical

Show Details

Approximate Distinct Count	31
Approximate Unique (%)	1.4%
Missing	0
Missing (%)	0.0%
Memory Size	166.6 KB

#### Intervenants

Top 10 of 31

search.google.com/drive/1yx581WOC#077zQJ-1VMgJ2M8fc-ydbO#scrollTo=e0tzLKDT4DIn&printMode=true 22/37

Automated\_EDA\_MEMOIRE - Colaboratory

0:00

Intervenants\_reel

categorical

Show Details

Approximate Distinct Count	38
Approximate Unique (%)	1.8%
Missing	0
Missing (%)	0.0%
Memory Size	166.7 KB

#### Intervenants\_reel

Top 10 of 38

Matiere

categorical

Show Details

Approximate Distinct Count	75
Approximate Unique (%)	3.5%
Missing	0
Missing (%)	0.0%
Memory Size	220.8 KB

#### Matiere

Top 10 of 75

Campus

categorical

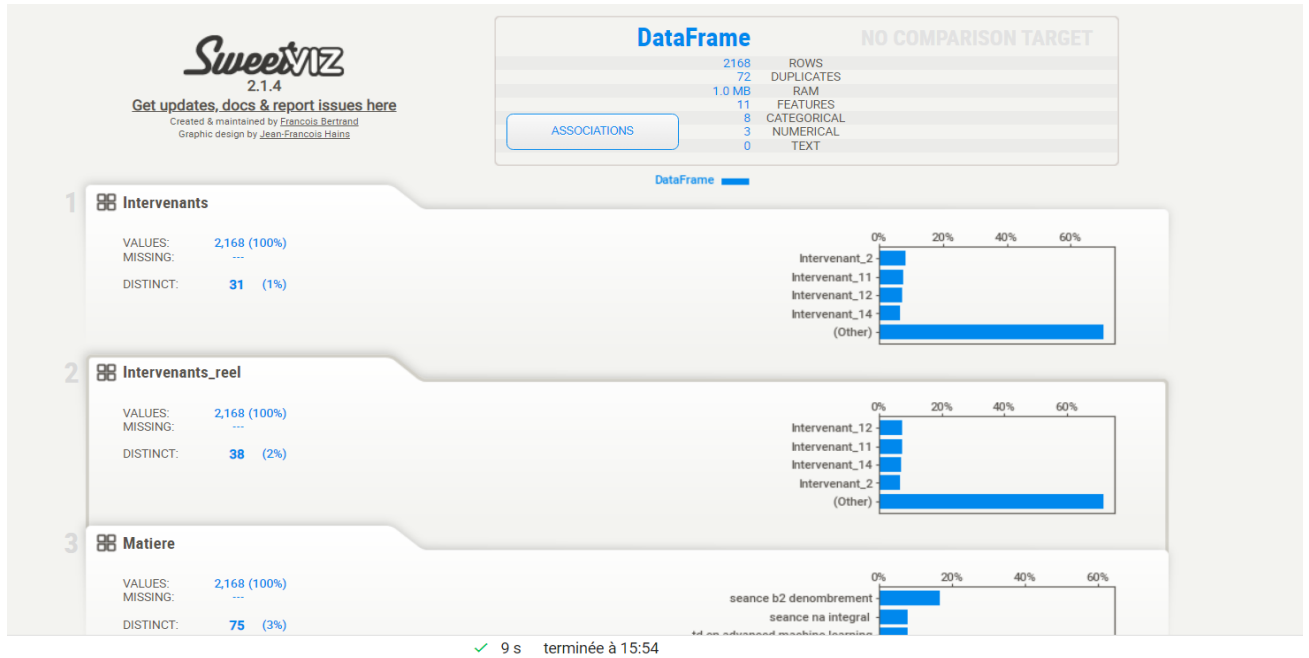
Show Details

Approximate Distinct Count	27
Approximate Unique (%)	1.2%
Missing	0
Missing (%)	0.0%
Memory Size	158.6 KB

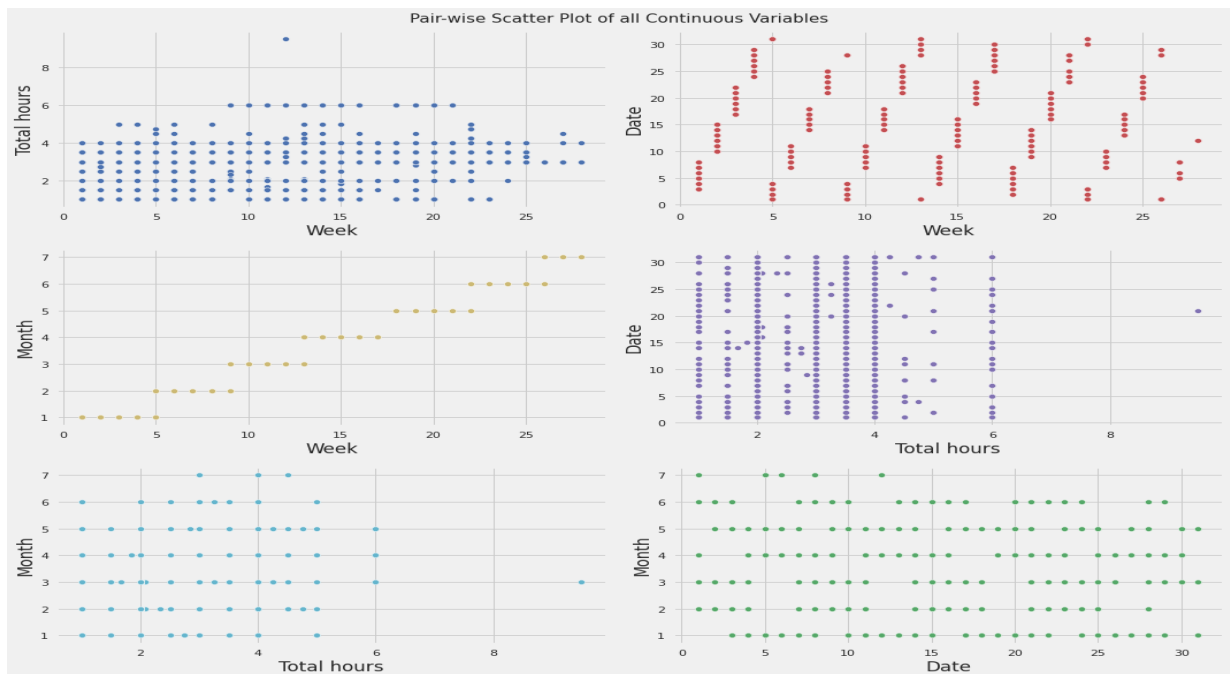
#### Campus

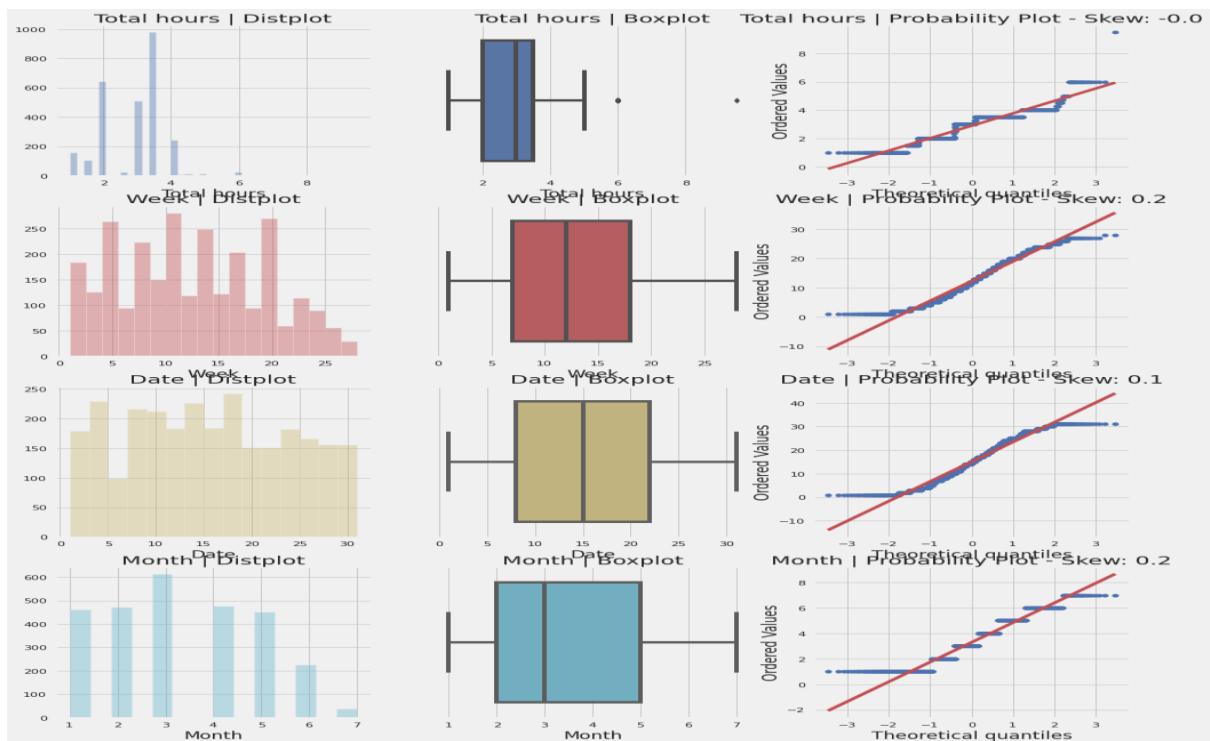
Top 10 of 27

## 2) Sweetviz



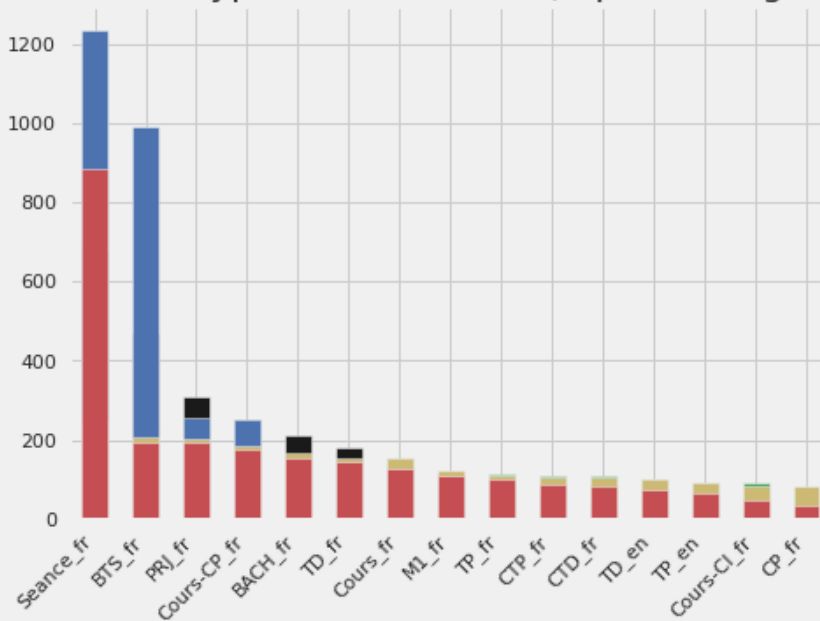
## 3) AutoViz





Histograms (KDE plots) of all Continuous Variables

### Distribution of Type d'intervention (top 15 categories only)



## 4) Dataprep

```
[ ] !pip install -U dataprep
```

```
[65] from dataprep.eda import create_report  
      create_report(df)
```

DataPrep Report

Overview

Variables

Interactions

Correlations

Missing Values

## Overview

### Dataset Statistics

Number of Variables	11
Number of Rows	2168
Missing Cells	0
Missing Cells (%)	0.0%
Duplicate Rows	72
Duplicate Rows (%)	3.3%
Total Size in Memory	1003.5 KB
Average Row Size in Memory	474.0 B
Variable Types	Categorical: 8

### Dataset Insights

Total hours is skewed	Skewed
Dataset has 72 (3.32%) duplicate rows	Duplicates
Matiere has a high cardinality: 75 distinct values	High Cardinality
Year has constant value "2022.0"	Constant
Year has constant length 6	Constant Length
Month has constant length 3	Constant Length

✓ 9 s terminée à 15:54

```
[ ] !pip install -U dataprep
```

```
from dataprep.eda import create_report  
create_report(df)
```

DataPrep Report

Overview

Variables

Interactions

Correlations

Missing Values

## Variables

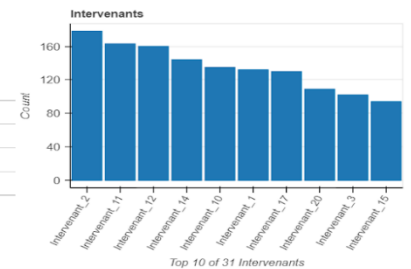
Sort by Feature order ☐ Reverse order

### Intervenants

categorical

Show Details

Approximate Distinct Count	31
Approximate Unique (%)	1.4%
Missing	0
Missing (%)	0.0%
Memory Size	166.6 KB



## 5) Data\_purifier



Shape of DataFrame: (2168, 11)

Sample of DataFrame:

	Intervenants	Intervenants_reel	Matiere	Campus	Year	Month	Week	Total hours	Date	Type_d'intervention	Referencie
1989	Intervenant_16	Intervenant_16	ctp langages et compilation	EFREI	2022.0	2.0	6.0	3.0	7.0	CTD_fr	Caplogy
1682	Intervenant_15	Intervenant_15	prj na solution factory	EFREI	2022.0	6.0	24.0	3.0	14.0	PRJ_fr	Caplogy
368	Intervenant_26	Intervenant_26	seance na integral	IA_School_Paris	2022.0	5.0	20.0	3.5	17.0	Seance_fr	Innovation
1776	Intervenant_3	Intervenant_3	bts atelier de prof	ENSITECH_Cergy	2022.0	2.0	6.0	1.5	7.0	BTS_fr	Innovation
572	Intervenant_3	Intervenant_3	seance b2 denombrement	IA_School_Paris	2022.0	3.0	12.0	3.5	22.0	Seance_fr	Innovation
1204	Intervenant_14	Intervenant_14	td na mesh&lpwan	ESILV	2022.0	2.0	7.0	3.0	16.0	TD_fr	Caplogy
631	Intervenant_0	Intervenant_0	prj na solution factory	EFREI	2022.0	7.0	26.0	3.0	1.0	PRJ_fr	Caplogy
2107	Intervenant_4	Intervenant_8	ing3 traitement du signal numérique	ECE	2022.0	4.0	16.0	1.0	22.0	Cours-TP_fr	Caplogy
282	Intervenant_4	Intervenant_4	cours b2 économie générale	IA_School_Paris	2022.0	2.0	5.0	3.5	4.0	Cours_fr	Innovation
2105	Intervenant_4	Intervenant_4	int2 en introduction to databases	ESME_Paris	2022.0	4.0	14.0	2.0	8.0	BACH_fr	Caplogy

There are total 6 categorical and 5 numerical columns

There are total 6 categorical and 5 numerical columns

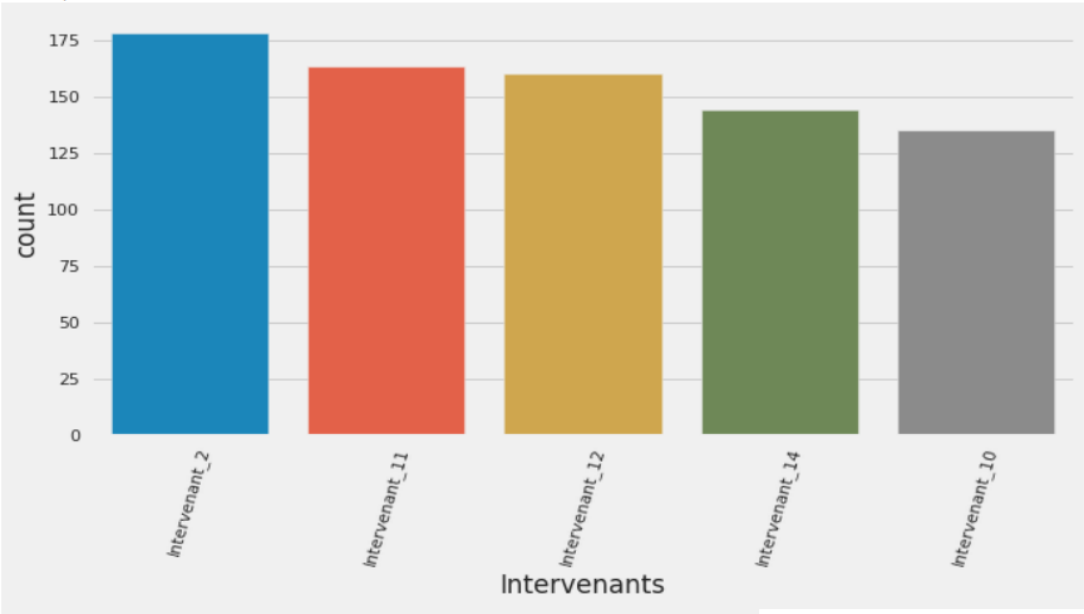
Description of Data:

	Year	Month	Week	Total hours	Date
count	2168.0	2168.000000	2168.000000	2168.000000	2168.000000
mean	2022.0	3.333487	12.437269	2.976707	15.042435
std	0.0	1.591667	6.799979	0.905151	8.573311
min	2022.0	1.000000	1.000000	1.000000	1.000000
25%	2022.0	2.000000	7.000000	2.000000	8.000000
50%	2022.0	3.000000	12.000000	3.000000	15.000000
75%	2022.0	5.000000	18.000000	3.500000	22.000000
max	2022.0	7.000000	28.000000	9.500000	31.000000

Information regarding data:

<class 'pandas.core.frame.DataFrame'>

Scatter plot of Total hours by Intervenant



6) Data\_dashboarder

Features

[Overview](#)

[Features](#)

[Models](#)

≡

Campus

Features

x

000. Campus

001. Date

002. Intervenants

003. Intervenants\_reel

004. Matière

005. Month

006. Reference

007. Total\_hours

008. Type\_d'intervention

009. Week

Chosen Feature Information

Summary Statistics

Intervenants\_reel

DescriptionDescription not Available

Type: Categorical

Mean: 10.5826

Median: 7.0000

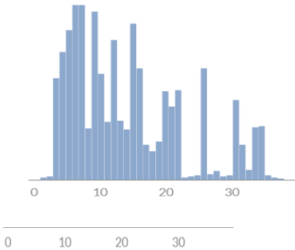
Min: 1.0000

Max: 27.0000

Standard deviation: 7.4572

# of Missing: 0.0000

Feature Distribution



Transformed Feature Information

Transformers (fitted on Train Data)

SimpleImputer(strategy='most\_frequent')

OneHotEncoder(handle\_unknown='ignore')

Applied Transformations (First 5 Rows) - Test Data

Original_Intervenants	Intervenants_Intervenant_0	Intervenants_Intervenant_1	Intervenants_Intervenant_10	Intervenants_Intervenant_11	Intervenants_Intervenant_12	Intervenants_Intervenant_13	Intervenants_Intervenant_14
Intervenant_2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Intervenant_24	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Intervenant_26	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Intervenant_5	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Intervenant_18	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Transformers (fitted on Train Data)

SimpleImputer(strategy='most\_frequent')

OneHotEncoder(handle\_unknown='ignore')

Applied Transformations (First 5 Rows) - Test Data

Original_Intervenants_reel	Intervenants_reel_ ANNE Sophie	Intervenants_reel_ Ahmed FADEL	Intervenants_reel_ Intervenant_0	Intervenants_reel_ Intervenant_1	Intervenants_reel_ Intervenant_10	Intervenants_reel_ Intervenant_11
Intervenant_2	0.0	0.0	0.0	0.0	0.0	0.0
Intervenant_24	0.0	0.0	0.0	0.0	0.0	0.0
Intervenant_26	0.0	0.0	0.0	0.0	0.0	0.0
Intervenant_5	0.0	0.0	0.0	0.0	0.0	0.0
Intervenant_18	0.0	0.0	0.0	0.0	0.0	0.0

7) Dtale

Y Elec Data Imb Ense Col Cap ML Autc Feat Doc Dee Autc Git+ Redi http data +

pvou91izfkl-496ff2e9c6d22116-40000-colab.googleusercontent.com/dtale/main/1

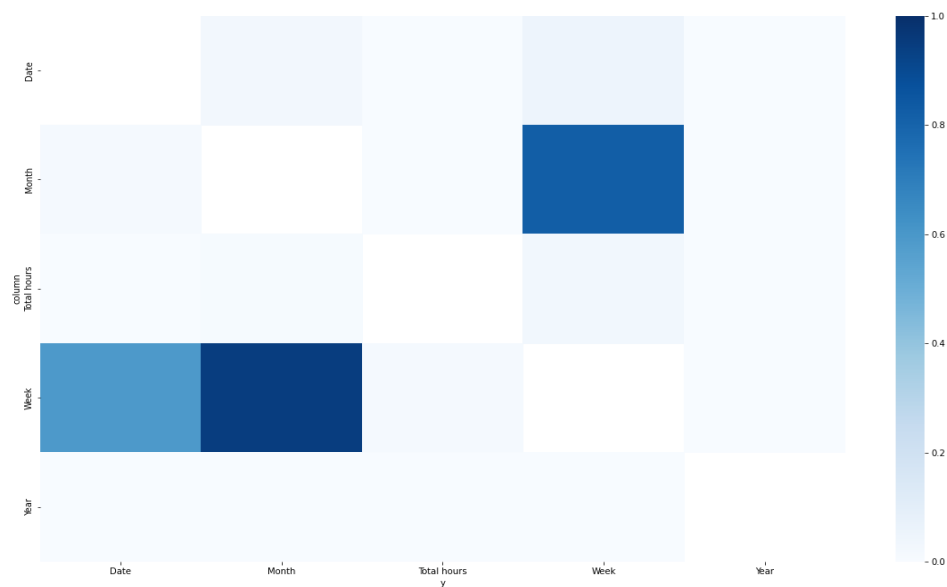
Documents pour l'é... sujets entiers ESD dossiers 2016... MégaMaths Classic sujets et corrigés d... Forfaits et tarifs Sujets LYON 5 (EML... Bibliothèque d'exer... Club de mathémati...

**D-TALE** Actions Visualize Highlight Settings

	Intervenants_reel	Matiere	Campus	Year	Month	Week	Total hours	Date	Type_d'intervention	Reference
Convert To XArray	Intervenant_1	cours b2 économie générale	IA_School_Paris	2022.00	2.00	5.00	3.50	3.00	Cours_fr	Innovation
Describe	Intervenant_5	ctp langages et compilation	EFREI	2022.00	2.00	5.00	3.00	1.00	CTP_fr	Caplogy
Custom Filter	Intervenant_5	prj I1 projet transverse	EFREI	2022.00	2.00	5.00	2.00	1.00	PRJ_fr	Caplogy
Show/Hide Columns	Intervenant_5	ctp en operating systems	EFREI	2022.00	2.00	5.00	2.00	1.00	CTP_en	Caplogy
Dataframe Functions	Intervenant_5	td en advanced machine learning	ESILV	2022.00	2.00	5.00	3.00	2.00	TD_en	Caplogy
Clean Column	Intervenant_5	ctp en operating systems	EFREI	2022.00	2.00	5.00	2.00	2.00	CTP_en	Caplogy
Merge & Stack	Intervenant_5	ctd devops - architecture n-tiers	EFREI	2022.00	2.00	5.00	3.00	3.00	CTD_fr	Caplogy
Summarize Data	Intervenant_5	prj I1 projet transverse	EFREI	2022.00	2.00	5.00	2.00	4.00	PRJ_fr	Caplogy
Time Series Analysis	Intervenant_5	et structures de données 1	EFREI	2022.00	2.00	5.00	2.00	4.00	TP_fr	Caplogy
Duplicates	Intervenant_5	ctp en operating systems	EFREI	2022.00	2.00	6.00	2.00	7.00	CTP_en	Caplogy
Missing Analysis	Intervenant_5	ctp langages et compilation	EFREI	2022.00	2.00	6.00	3.00	8.00	CTP_fr	Caplogy
Feature Analysis	Intervenant_5	ctp en operating systems	EFREI	2022.00	2.00	6.00	2.00	8.00	CTP_en	Caplogy
Correlations	Intervenant_5	et structures de données 1	EFREI	2022.00	2.00	6.00	3.00	11.00	TP_fr	Caplogy
Predictive Power Score	Intervenant_5	ctp langages et compilation	EFREI	2022.00	2.00	7.00	3.00	15.00	CTP_fr	Caplogy
Charts	Intervenant_5	et structures de données 1	EFREI	2022.00	2.00	7.00	2.00	18.00	TP_fr	Caplogy
Network Viewer	Intervenant_5	td en advanced machine learning	ESILV	2022.00	1.00	1.00	3.00	5.00	TD_en	Caplogy
Heat Map	Intervenant_5	cm en computer vision	ESILV	2022.00	1.00	1.00	3.00	7.00	CM_en	Caplogy
Highlight Dtypes	Intervenant_5	td en advanced machine learning	ESILV	2022.00	1.00	2.00	3.00	12.00	TD_en	Caplogy
Highlight Missing	Intervenant_5	ctp langages et compilation	EFREI	2022.00	1.00	3.00	3.00	18.00	CTP_fr	Caplogy
Highlight Outliers	Intervenant_5	td en advanced machine learning	ESILV	2022.00	1.00	3.00	3.00	19.00	TD_en	Caplogy
	Intervenant_5	ctd devops - architecture n-tiers	EFREI	2022.00	1.00	3.00	5.00	21.00	CTD_fr	Caplogy
	Intervenant_5	ctp langages et compilation	EFREI	2022.00	1.00	4.00	2.00	25.00	CTP_fr	Caplogy

Taper ici pour rechercher

17:03 30/08/2022



## Présentations des résultats de l'AutoML

Lazypredict

## a) Cas de la régression

- Résultat

Model	Adjusted R-Squared	R-Squared	RMSE	Time Taken
LGBMRegressor	0.6721541986167248	0.6776081953125649	0.4988676901525992	0.09722375869750977
HistGradientBoostingRegressor	0.6684361407887224	0.6739519905722742	0.5016885127863085	0.5665683746337891
RandomForestRegressor	0.6415427916985725	0.6475060354595945	0.5216380024481942	0.5819482803344727
BaggingRegressor	0.6303258978198562	0.6364757442516885	0.5297367191677969	0.07768630981445312
GradientBoostingRegressor	0.6175211423485836	0.6238840068936164	0.5388331182624929	0.2311115264892578
XGBRegressor	0.5828436649824367	0.5897834191694202	0.5627298925057977	0.32167649269104004
ExtraTreesRegressor	0.5766179965061757	0.5836613200393448	0.5669134502829565	0.358717679977417
NuSVR	0.472854412428211	0.48162393236933143	0.6325805753614286	0.32239627838134766
SVR	0.4681481286118625	0.47699594162941006	0.6353980972168186	0.23412442207336426
KNeighborsRegressor	0.4674644927313355	0.476323678619354	0.6358063325391821	0.028404712677001953
MLPRegressor	0.46076082260744744	0.4697315298099114	0.6397956498172154	2.565183162689209
DecisionTreeRegressor	0.4444653392080604	0.45370713578315747	0.6493908190253065	0.06181502342242121
AdaBoostRegressor	0.37983392127501314	0.39015091703938454	0.6861270921444919	0.16696786880493164
Ridge	0.3284266825480477	0.3395988819145156	0.7139984688283932	0.024261951446533203
RidgeCV	0.3283769572911629	0.3395498387966484	0.7140249016090265	0.032767295837402344
TransformedTargetRegressor	0.32822398487077276	0.3393995562869707	0.7141062120255108	0.020534992218017578
LinearRegression	0.32822398487077276	0.3393995562869707	0.7141062120255108	0.01708531379699707
LassoLarsCV	0.3282239848707703	0.3393995562869683	0.7141062120255122	0.06968498229980469
BayesianRidge	0.3271878840360718	0.3383806918802037	0.7146566938514641	0.025620460510253906
SGDRegressor	0.3266520637448399	0.3378537854200644	0.7149412096466232	0.055759429931640625
ElasticNetCV	0.32619652261266185	0.3374058226061666	0.7151830089045612	0.17638468742370605
LassoLarsIC	0.3260867187645812	0.3372978454394773	0.7152412800845308	0.023154020309448242
LassoCV	0.3259764607267346	0.33718942163885923	0.7152997875193886	0.17011117935180664

9 s terminée à 02:51

## b) Cas de la classification

- Résultat

Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	Time Taken
BaggingClassifier	1.0	1.0	null	1.0	0.17905187606811523
XGBClassifier	1.0	1.0	null	1.0	0.627349853515625
DecisionTreeClassifier	0.9981549815498155	0.9989035087719298	null	0.9981628735110464	0.027942657470703125
LGBMClassifier	0.9981549815498155	0.9945652173913043	null	0.9981470996095813	0.3567485809326172
RandomForestClassifier	0.992619926199262	0.9841276935545384	null	0.9925872371722814	2.5960946083068848
ExtraTreesClassifier	0.992619926199262	0.9797894021739131	null	0.9925364318364958	0.4625979469848633
ExtraTreeClassifier	0.9538745387453874	0.934039637737246	null	0.9535920815610305	0.027111291885375977
LabelSpreading	0.8800738007380073	0.8268192359895903	null	0.8797437724707806	0.48591085406799316
LabelPropagation	0.8782287822878229	0.8255937457935119	null	0.8780541647644411	0.3253657817840576
KNeighborsClassifier	0.8450184501845018	0.7929622649975321	null	0.8431385091359658	0.10497355461120605
SVC	0.8154981549815498	0.7135067878696102	null	0.809339620437564	0.20321130752563477
GaussianNB	0.6180811808118081	0.6694604758379324	null	0.5960197083660264	0.026546001434326172
QuadraticDiscriminantAnalysis	0.6328413284132841	0.6388202413402432	null	0.6136746248710792	0.053403615951538086
BernoulliNB	0.6051660516605166	0.6195787132072509	null	0.6031492280887878	0.09041547775268555
NearestCentroid	0.5701107011070111	0.6103412653677032	null	0.5806691729059292	0.049538373947143555
LogisticRegression	0.7287822878228782	0.5789854020841746	null	0.7163749592878159	0.25073719024658203
PassiveAggressiveClassifier	0.6217712177121771	0.5466306731502869	null	0.6179129322581879	0.04243302345275879
CalibratedClassifierCV	0.7177121771217713	0.5452244792367748	null	0.6954377278500334	1.9433839321136475
LinearDiscriminantAnalysis	0.7011070110701108	0.5440238774285459	null	0.6866810107190706	0.0370173454284668
AdaBoostClassifier	0.3597785977859778	0.5283994032395567	null	0.3665350461224498	0.4904055595397949
LinearSVC	0.7011070110701108	0.49425428781352354	null	0.6642787882654286	0.24120545387268066
Perceptron	0.6531365313653137	0.4654661349665724	null	0.6268440622213096	0.061701059341430664
SGDClassifier	0.6734317343173432	0.4505047084623323	null	0.6323914672720267	0.10597395896911621
RidgeClassifier	0.6826568265682657	0.44051938209494323	null	0.61937087395983	0.03204059600830078
RidgeClassifierCV	0.6826568265682657	0.44051938209494323	null	0.61937087395983	0.046515703201293945

Show 25 per page

12

AutoViML

## a) Cas de la régression

- Résultat:

imported Auto\_ViML version: 0.1.710. Call using:

```
m, feats, trainm, testm = Auto_ViML(train, target, test,
    sample_submission="",
    scoring_parameter="", KMeans_Featurizer=False,
    hyper_param='RS', feature_reduction=True,
    Boosting_Flag='CatBoost', Binning_Flag=False,
    Add_Poly=0, Stacking_Flag=False, Imbalanced_Flag=False,
    verbose=1)
```

Imported Auto\_NLP version: 0.1.01.. Call using:

```
train_nlp, test_nlp, nlp_pipeline, predictions = Auto_NLP(
    nlp_column, train, test, target, score_type='balanced_accuracy',
    modeltype='Classification', top_num_features=200, verbose=0,
    build_model=True)
```

##### DATA SET ANALYSIS #####

Training Set Shape = (1951, 11)

Training Set Memory Usage = 0.16 MB

Test Set Shape = (217, 11)

Test Set Memory Usage = 0.02 MB

Single\_Label Target: ['Total hours']

##### Regression VISUALIZATION Started #####

No shuffling of data set before training...

Using RandomizedSearchCV for Hyper Parameter Tuning. This is 3X faster than GridSearchCV...

#####

##### CLASSIFYING VARIABLE

S #####

#####

Classifying variables in data set...

Number of Numeric Columns = 3

Number of Integer-Categorical Columns = 0

Number of String-Categorical Columns = 6

Number of Factor-Categorical Columns = 0

Number of String-Boolean Columns = 0

Number of Numeric-Boolean Columns = 0

Number of Discrete String Columns = 0

Number of NLP String Columns = 0

Number of Date Time Columns = 0

Number of ID Columns = 0

Number of Columns to Delete = 1

10 Predictors classified...

1 variables removed since they were ID or low-information variables

['Year']

Number of Processors on this device = 1

CPU available

GPU active on this device

#####  
#####

## ##### DATA PREPARATION AND CLEANING #####

#####  
#####

No Missing Values in train data set

Test data has no missing values. Continuing...

Completed Label Encoding and Filling of Missing Values for Train and Test Data

Regression problem: hyperparameters are being optimized for mae

#####  
#####

##### SULOV: Searching for Uncorrelated List Of Variables in 3 features #####

#####  
#####

there are no null values in dataset...

Removing (1) highly correlated variables:

['Month']

Following (2) vars selected: ['Date', 'Week']

### How SULOV Method Works by Removing Highly Correlated Features

In SULOV, we repeatedly remove features with lower mutual info scores among highly correlated pairs (see figure),  
SULOV selects the feature with higher mutual info score related to target when choosing between a pair.



Splitting selected features into float and categorical (integer) variables:

(2) float variables ...

(6) categorical vars...

#####  
#####

## ##### FEATURE SELECTION BY XGBOOST #####

#####  
#####

Current number of predictors = 8

Finding Important Features using Boosted Trees algorithm...

using 8 variables...

using 6 variables...

using 4 variables...

using 2 variables...

Found 8 important features

Performing limited feature engineering for binning, add\_poly and KMeans\_Featurizer flags ...

Train CV Split completed with TRAIN rows = 1560 , CV rows = 391

Binning\_Flag set to False or there are no float vars in data set to be binned

KMeans\_Featurizer set to False or there are no float variables in data

Skipping MinMax scaling since perform\_scaling flag is set to False

#####  
#####  
##### CatBoost MODEL TRAINING #####  
#####  
#####

Rows in Train data set = 1560

Features in Train data set = 8

Rows in held-out data set = 391

Finding Best Model and Hyper Parameters for CatBoost model...

CPU Count = 2 in this device

Using CatBoost Model, Estimated Training time = 0.116 mins

Warning: Overfitting detector is active, thus evaluation metric is calculated on every iteration.

'metric\_period' is ignored for evaluation metric.

Learning rate set to 0.016661

0:	learn: 0.8780462	test: 0.9253394	best: 0.9253394 (0)	total: 47.6ms
	remaining: 5m 32s			
500:	learn: 0.4461443	test: 0.5122626	best: 0.5122626 (500)	total: 1.07s
	remaining: 13.9s			
1000:	learn: 0.3827057	test: 0.4819224	best: 0.4819224 (1000)	total: 2.12s
	remaining: 12.7s			
1500:	learn: 0.3520860	test: 0.4729143	best: 0.4728032 (1498)	total: 3.25s
	remaining: 11.9s			
2000:	learn: 0.3342581	test: 0.4690141	best: 0.4690069 (1996)	total: 4.07s
	remaining: 10.2s			
2500:	learn: 0.3200404	test: 0.4688980	best: 0.4683644 (2171)	total: 5.25s
	remaining: 9.44s			
3000:	learn: 0.3101162	test: 0.4705618	best: 0.4683644 (2171)	total: 6.18s
	remaining: 8.24s			
3500:	learn: 0.3024220	test: 0.4733562	best: 0.4683644 (2171)	total: 6.87s
	remaining: 6.87s			
4000:	learn: 0.2961659	test: 0.4757572	best: 0.4683644 (2171)	total: 7.61s
	remaining: 5.71s			

Stopped by overfitting detector (2000 iterations wait)

bestTest = 0.4683644473

bestIteration = 2171

Shrink model to first 2172 iterations.

Actual training time (in seconds): 8

##### Single\_Label MODEL RESULTS #####

5-fold Cross Validation RMSE Score = 0.4684

CatBoost Best Parameters for Model: Iterations = 2171, learning\_rate = 0.02

#####

CatBoost Model Prediction Results on Held Out CV Data Set:

Regression Plots completed in 0.130 seconds

MAE = 0.3005

MAPE = 13% (MAPE will be very high when zeros in actuals)

RMSE = 0.4684

Normalized MAE (as % std dev of Actuals) = 32%

Normalized RMSE (% of Std Dev of Actuals) = 50%

##### ENSEMBLE MODEL #####

Time taken = 3 seconds

Based on trying multiple models, Best type of algorithm for this data set is RF\_Regressor

Displaying results of weighted average ensemble of 5 regressors

#####  
#####

Regression Plots completed in 0.159 seconds

MAE = 0.3387

MAPE = 14% (MAPE will be very high when zeros in actuals)

RMSE = 0.4938

Normalized MAE (as % std dev of Actuals) = 36%

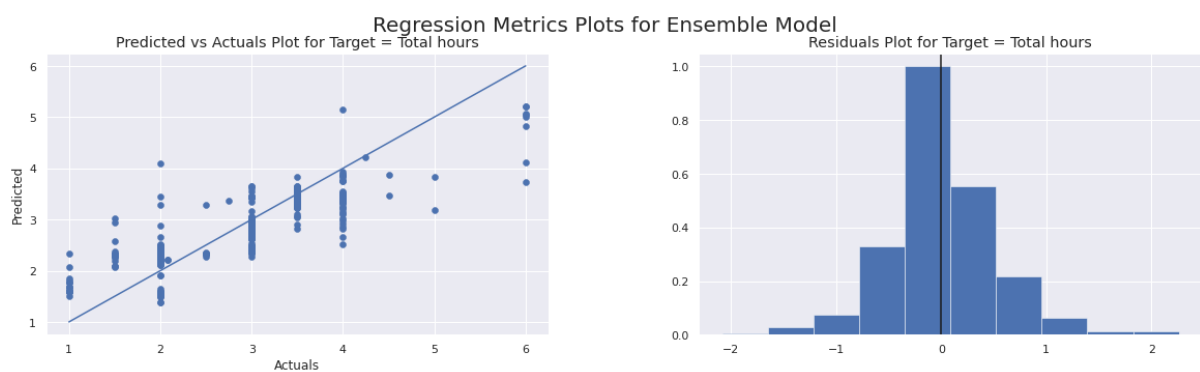
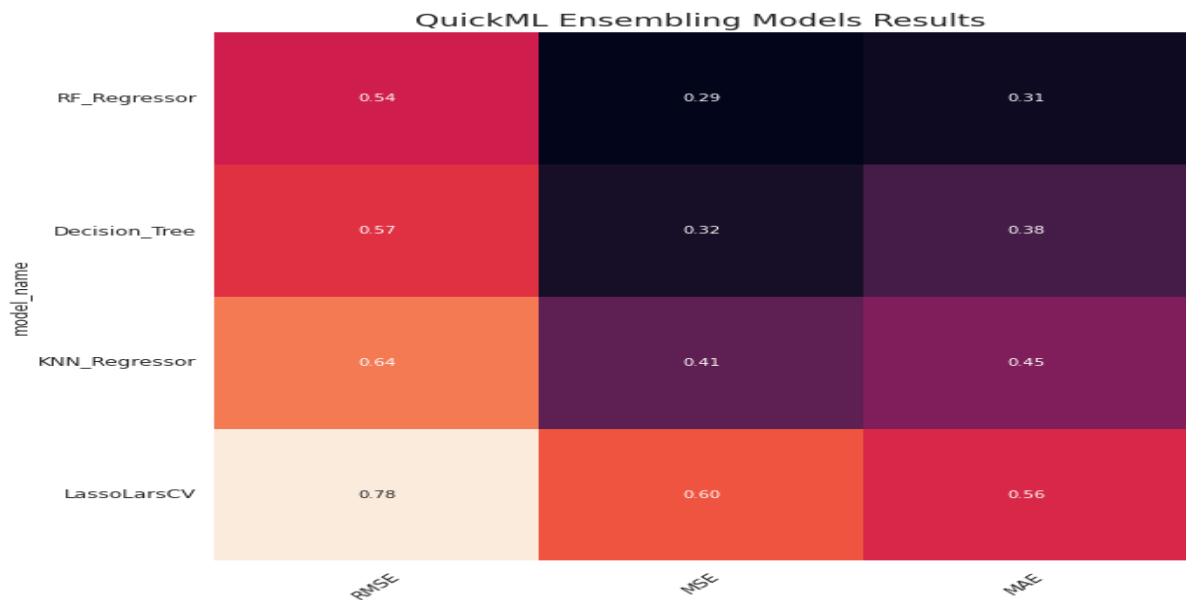
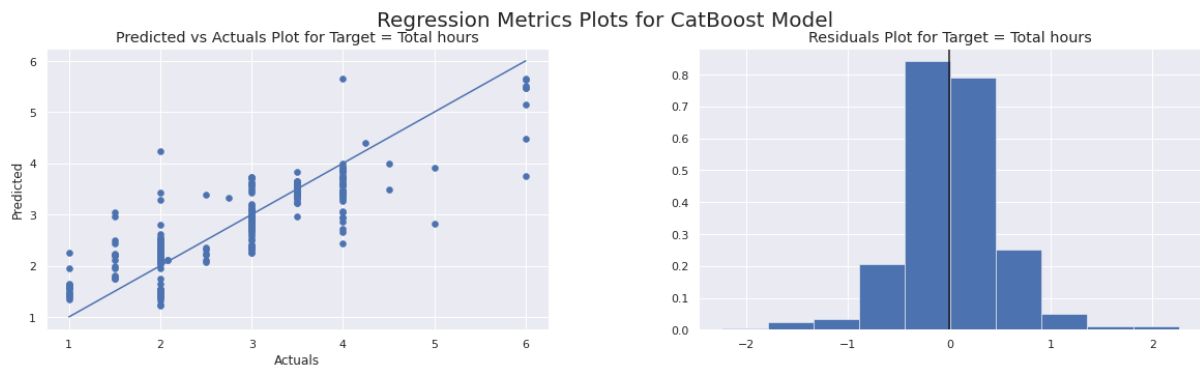
Normalized RMSE (% of Std Dev of Actuals) = 53%

After multiple models, Ensemble Model Results:

RMSE Score = 0.49382

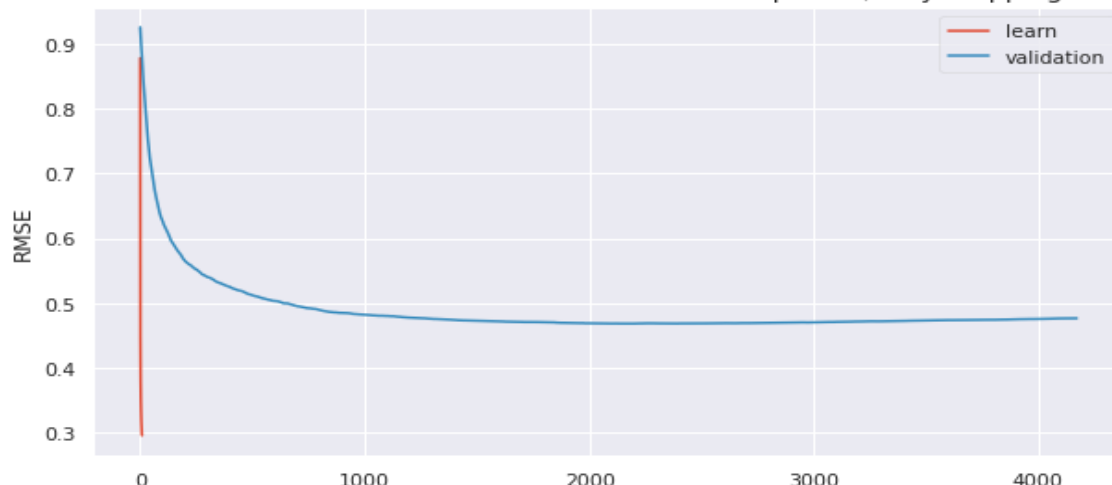
#####  
#####

Single Model is better than Ensembling Models for this data set.





Total hours Results Train and Validation Metrics across Epochs (Early Stopping in effect)



Time taken for this Target (in seconds) = 22

Binning\_Flag set to False or there are no float vars in data set to be binned

Setting best params for CatBoost model from Initial State since you cannot change params to a fitted Catboost model

Number of Categorical and Integer variables used in CatBoost training = 6

No MinMax scaling performed since scaling flag is set to false

#####  
###

##### FINALIZING MODEL ON FULL TRAIN #####

#####  
###

0:	learn: 0.8884884	total: 1.25ms	remaining: 2.72s
500:	learn: 0.4482528	total: 931ms	remaining: 3.1s
1000:	learn: 0.3914659	total: 1.69s	remaining: 1.98s
1500:	learn: 0.3642615	total: 2.47s	remaining: 1.1s
2000:	learn: 0.3445734	total: 3.8s	remaining: 323ms
2170:	learn: 0.3396609	total: 4.31s	remaining: 0us

Actual Training time taken in seconds = 5

Training of models completed. Now starting predictions on test data...

Calculating weighted average ensemble of 5 regressors

Completed Ensemble predictions on held out data

Plotting Feature Importances to explain the output of model

##### PREDICTION ON TEST COMPLETED #####

Time taken thus far (in seconds) = 30

Writing Output files to disk...

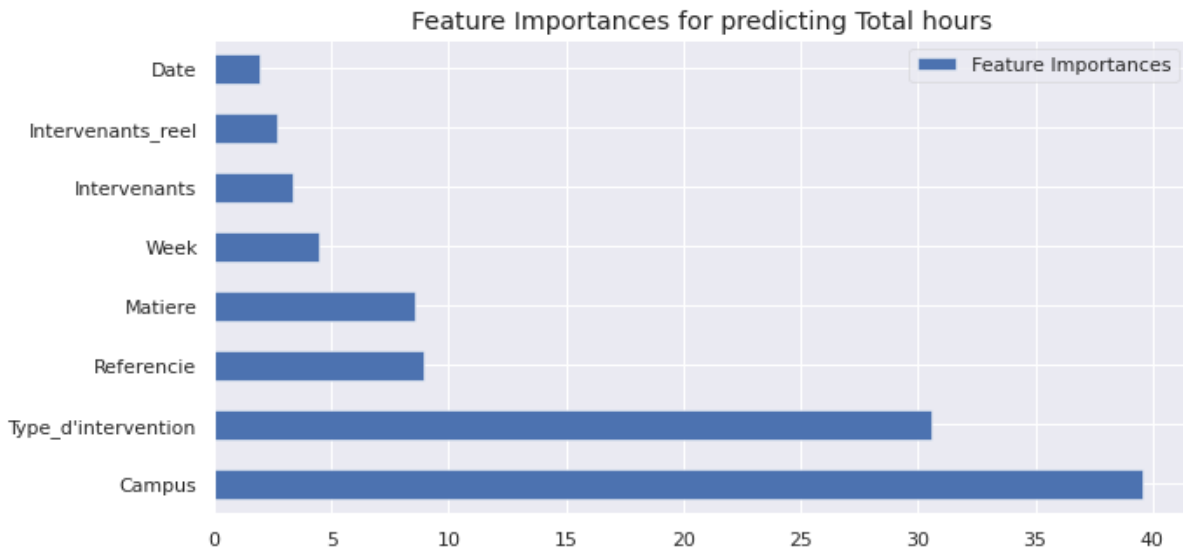
Saving predictions to ./Total hours/Total hours\_Regression\_test\_modified.csv

Saving predictions to ./Total hours/Total hours\_Regression\_submission.csv

Saving predictions to ./Total hours/Total hours\_Regression\_train\_modified.csv

##### COMPLETED #####

Time Taken in mins = 0.5 for the Entire Process



a) Cas de la classification (en cas de besoin me contacter)

## LigthAutoML

a) Cas de la régression

Résultat :

```
INFO:lightautoml.automl.base:Layer 1 training completed.
```

```
INFO:lightautoml.automl.blend:Blending: optimization starts with equal weights and score -0.21501998802412425
```

```
INFO:lightautoml.automl.blend:Blending: iteration 0: score = -0.21343638624740086, weights = [0.18597613 0.46900022 0.34502366 0. 0. ]
```

```
INFO:lightautoml.automl.blend:Blending: iteration 1: score = -0.21312282508928546, weights = [0.24134026 0.4193296 0.3393302 0. 0. ]
```

```
INFO:lightautoml.automl.blend:Blending: iteration 2: score = -0.2131218122098189, weights = [0.24281742 0.41204774 0.34513482 0. 0. ]
```

```
INFO:lightautoml.automl.blend:Blending: iteration 3: score = -0.21312169324504918, weights = [0.24332635 0.4095056 0.3471681 0. 0. ]
```

```
INFO:lightautoml.automl.blend:Blending: iteration 4: score = -0.21312168105156976, weights = [0.24350215 0.40863055 0.3478673 0. 0. ]
```

```
INFO:lightautoml.automl.presets.base:Automl preset training completed in 125.70 seconds
```

```
INFO:lightautoml.automl.presets.base:Model description:
```

```
Final prediction for new objects (level 0) =
```

```
0.24350 * (5 averaged models Lvl_0_Pipe_0_Mod_0_LinearL2) +
```

0.40863 \* (5 averaged models Lvl\_0\_Pipe\_1\_Mod\_0\_LightGBM) +  
0.34787 \* (5 averaged models Lvl\_0\_Pipe\_1\_Mod\_1\_Tuned\_LightGBM)

## b) cas de la classification

### Résultat

INFO:lightautoml.automl.base:Layer 1 training completed.

INFO:lightautoml.automl.blend:Blending: optimization starts with equal weights and score -0.005618870523066639

INFO:lightautoml.automl.blend:Blending: iteration 0: score = -0.0023137717937401435, weights = [0. 0. 0. 0. 1.]

INFO:lightautoml.automl.blend:Blending: iteration 1: score = -0.0022907775892676225, weights = [0.09621898

0. 0. 0. 0.903781 ]

INFO:lightautoml.automl.blend:Blending: iteration 2: score = -0.0022907775892676225, weights = [0.09621898

0. 0. 0. 0.903781 ]

INFO:lightautoml.automl.blend:Blending: no score update. Terminated

INFO:lightautoml.automl.presets.base:Automl preset training completed in 81.90 seconds

INFO:lightautoml.automl.presets.base:Model description:

Final prediction for new objects (level 0) =

0.09622 \* (3 averaged models Lvl\_0\_Pipe\_0\_Mod\_0\_LinearL2) +

0.90378 \* (5 averaged models Lvl\_0\_Pipe\_1\_Mod\_3\_Tuned\_CatBoost)

## AutoSklearn

### a) Cas de la régression

#### Résultat

auto-sklearn results:

Dataset name: 2c08c724-2543-11ed-8045-0242ac1c0002

Metric: r2

Best validation score: 0.643255

Number of target algorithm runs: 29

Number of successful target algorithm runs: 11

Number of crashed target algorithm runs: 3

Number of target algorithms that exceeded the time limit: 15

Number of target algorithms that exceeded the memory limit: 0

RMSE: 0.5345525108613502

### b) Cas de la classification

## Résultat

```
auto-sklearn results:
Dataset name: 7defe6c0-2543-11ed-8045-0242ac1c0002
Metric: accuracy
Best validation score: 0.998138
Number of target algorithm runs: 31
Number of successful target algorithm runs: 25
Number of crashed target algorithm runs: 0
Number of target algorithms that exceeded the time limit: 6
Number of target algorithms that exceeded the memory limit: 0

Accuracy: 0.9944649446494465
```

## Outil d'optimisation de pipeline basé sur l'arborescence (TPOT)

### a) Cas de la régression :

## Résultat

```
{ 'learner': XGBRegressor(base_score=0.5, booster='gbtree',
callbacks=None,
      colsample_bylevel=0.8932227866988688, colsample_bynode=1,
      colsample_bytree=0.5185014892651896,
early_stopping_rounds=None,
      enable_categorical=False, eval_metric=None,
      gamma=0.04808306169864229, gpu_id=-1,
grow_policy='depthwise',
      importance_type=None, interaction_constraints='',
      learning_rate=0.045475659855726464, max_bin=256,
      max_cat_to_onehot=4, max_delta_step=0, max_depth=7,
max_leaves=0,
      min_child_weight=2, missing=nan,
monotone_constraints='()',
      n_estimators=1200, n_jobs=1, num_parallel_tree=1,
predictor='auto',
      random_state=3, reg_alpha=0.7546089993311615,
      reg_lambda=2.235486969230257, ...), 'preprocs':
(StandardScaler(),), 'ex_preprocs': () }
```

### b) Cas de la classification (en cas de besoin veuillez me contacter)

## **Conclusion :**

En résumé, l'AutoEDA et l'AutoML avec python dans le cas de l'analyse de données, aident l'analyse dans la réalisation d'un projet. Elles permettent à l'Analyste, de gagner un temps précieux, de pouvoir sélectionner le modèle adéquat et l'aide à chaque étape de son projet de la préparation des données à la finalisation de son projet, avec une précision ou minimisation d'erreur qui est phénoménale. Ces bibliothèques d'AutoEDA et d'AutoML ont l'avantage d'être courtes et de n'être exhaustives en ce qui concerne leur implémentation. Néanmoins, certains modèles restent parfois trop abstraits et d'autres avec très peu d'éléments d'analyse quant aux résultats qu'ils renvoient (exemple : `dataile` (EDA) et `lazy_predict`). On peut, également souligner le fait que certaines, ne disposent pas assez de modèles comme les méthodes d'Ensemble (Boosting, Stacking, Bagging...etc.). Tout de même, ils restent très précieux car le gain de temps et la sélection des modèles et paramètres sont des éléments essentiels pour qu'un projet soit bien achevé. Par ailleurs, on pourrait se demander, s'ils existent des bibliothèques d'automatisation pour la détection faciale, portfolio, série temporelle, scrapping ?

## **Annexe :**

<https://siyab.medium.com/auto-eda-using-dtale-and-autoviz-b3aaf5703d65>

<https://machinelearningmastery.com/ensemble-machine-learning-with-python-7-day-mini-course/>

<https://perso.lpsm.paris/~aguyader/papers.html>

<https://www.analyticsvidhya.com/blog/2021/04/top-python-libraries-to-automate-exploratory-data-analysis-in-2021/>

[https://hastie.su.domains/ElemStatLearn/printings/ESLII\\_print12.pdf](https://hastie.su.domains/ElemStatLearn/printings/ESLII_print12.pdf)

<https://app.datacamp.com/learn/career-tracks/machine-learning-scientist-with-python?version=1>

