Analyse des techniques et outils des professionnels de la Data

Projet de formation Data Analyst

Promotion bootcamp décembre 2021

Auteurs : Thomas Henry, Alexandre Tournieux

Responsable projet : Mage Merceron

Sommaire

[**Introduction**](#_76zqbzgeiwv3) **3**

[**Préparation des données**](#_8yin6kn3wxsw) **4**

[**Data Cleaning**](#_ivj6ii8q39ev) **5**

[**Data Visualization**](#_y0md97wngf57) **5**

[Fonctions pour la data viz](#_fx2uj7v3udu5) 5

[Biais du dataset](#_gbmvtrqn3b8a) 8

[Périmètre de l’analyse](#_go8dbckeoduj) 9

[Analyse des données](#_df1dg2xnl5ny) 10

[Q7 : Langage de programmation](#_5an5hcbugcei) 10

[Q14 : Librairies de visualisation](#_7kpmt2gs67tx) 12

[Q16 : Librairies de machine learning](#_g1p7sfw7xlop) 14

[Q17 : Types de modèles utilisés pour le machine learning](#_xqghnufmy6i9) 16

[Q34 : Outils de Business Intelligence](#_clmm8sp5mre3) 18

[Q41 : Types de logiciel privilégié pour analyser des données](#_6fqmsmcgt68o) 20

[Conclusion](#_bl1nbqcyce0) 21

[**Modélisation**](#_69qv0kwp7py5) **22**

[Préparation des données](#_uw7tihb2rjdf) 22

[Première itération de modèles](#_rypd9l7y9onx) 23

[Optimisation des performances](#_ylovxuqo5gso) 24

[Rééchantillonnage](#_9ee09x6eytjc) 24

[Changement de la taille du groupe de test](#_ncogyxuph577) 24

[Validation croisée](#_ipctudpcb91n) 25

[Réduction du jeu de données](#_d52nld2uj5tl) 25

[Réduction du nombre de classe](#_ygertfdu1ty1) 25

[Conclusion](#_5wne5vk69yae) 26

# 

# 

# Introduction

Notre projet porte sur l’analyse des techniques et outils utilisés par les professionnels de la data. Cette étude est réalisée à partir des résultats d’un sondage publié sur le site Kaggle. Le dataset est disponible au lien suivant :

[https://www.kaggle.com/c/kaggle-survey-2020](https://www.kaggle.com/c/kaggle-survey-2020/overview)

Il contient les réponses à un questionnaire adressé à différents professionnels de la data ainsi qu'à des étudiants. Les 39 questions de ce questionnaire ont pour but de mettre en avant les outils et techniques utilisés dans le domaine de la data.

Afin d’enrichir notre projet, nous avons pris la décision d’ajouter le jeu de données de 2021 à celui de 2020. Ce second dataset est disponible au lien suivant :

<https://www.kaggle.com/c/kaggle-survey-2021>

# 

# Préparation des données

A la comparaison des rapports de 2020 et de 2021, notre premier constat a été que les deux questionnaires n’étaient pas identiques.

En effet, le rapport de 2021 contenait plus de colonnes que celui de 2020. Aussi, nous nous sommes rendu compte que la numérotation des questions n’était pas la même. Par exemple, la Q41 d’un questionnaire correspondait à la Q38 de l’autre.

L’agrégation de ces deux rapports nécessitait donc une première étape de préparation des données. Nous avons pris le parti d’aligner le questionnaire de 2020 sur celui de 2021 qui était plus complet et plus récent. Pour se faire, nous avons créé différents dictionnaires et différentes listes :

* Un dictionnaire permettant d’aligner les Qx (noms de colonne) de 2020 sur les Qx de 2021
* Une liste contenant les Qx à créer et à remplir en full NaN dans le questionnaire de 2020 (celles-ci étant uniquement présentes dans le questionnaire de 2021)
* Une liste contenant les colonnes du rapport de 2020 non présentes dans le rapport de 2021. La cible étant de supprimer ces colonnes pour aligner les deux rapports sur celui de 2021.
* Une liste contenant les questions à supprimer car non pertinentes. Certaines questions étant hypothétiques et uniquement destinées aux étudiants (Exemple : In the next 2 years, do you hope to become more familiar with any of these specific machine learning products?)

Après utilisation de ces dictionnaires et listes pour ajout, suppression et modification de colonnes dans les dataframes, nous avons aussi :

* Ajouté une colonne “Year” afin de bien dissocier les répondants de 2020 et 2021
* Supprimé l’énoncé des questions du dataframe de 2020 afin de ne pas l’avoir en doublon dans le futur dataframe fusionné

Toutes ces modifications ont permis de joindre facilement les deux questionnaires à partir des numéros de question via la fonction concat de panda.

Nous avons ensuite décidé de supprimer les lignes non-pertinentes. Nous avons jugé un questionnaire non-pertinent lorsque le répondant :

* Répondait à moins de 10 questions

Ou

* Prenait moins de 120 secondes à répondre au questionnaire

Enfin, nous avons réaffecté l’indexation du dataframe fusionné afin d’éviter les doublons d’index. Le dataframe final est composé de 44590 lignes et 268 colonnes.

# 

# Data Cleaning

Une fois nos données correctement compilées, nous avons dû effectuer un nettoyage du dataframe car le format des réponses aux questions n’était pas toujours identique entre 2020 et 2021. Voici quelques exemples de nettoyage que nous avons réalisés :

* Pour la Q10, “Google Cloud Datalab” en 2020 vs “Google Cloud Datalab Notebooks” en 2021
* Pour la Q35, “Einstein Analytics” en 2020 est devenu “Tableau CRM” en 2021. Tableau CRM est en fait le nouveau nom d’Einstein Analytics depuis 2020.
* Pour la Q9, la réponse Visual Studio était entrée avec un espace devant et un espace derrière en 2021, ce qui n’était pas le cas en 2021.

# Data Visualization

## Fonctions pour la data viz

Une fois le data cleaning effectué, nous avons pu commencer notre analyse du dataset. Pour nous faciliter la tâche, nous avons créé différents types de fonctions nous permettant de mettre en forme les données ou de générer des graphiques. En voici la liste :

1. Les fonctions de mise en forme / filtrage des données
2. La fonction QxToQuestion

Cette fonction permet de renvoyer l'intitulé d'une question à partir de son ID (au format 'Qx').

1. La fonction MultiQCount

Certaines questions sont présentées sous forme de plusieurs colonnes dans le dataframe. Chaque colonne représentant une réponse différente à cette même question.

​​Cette fonction renvoie un dataframe avec deux colonnes :

* 'Answer' contenant les différentes réponses possibles associées à la question concernée.
* 'Count' faisant le décompte du nombre d’apparition de cette question dans le dataframe.

Le dataframe résultant contiendra donc autant de lignes qu’il y a de réponses différentes à la question.

Les paramètres de cette fonction sont l’intitulé de la question (récupérable à partir de la fonction QxToQuestion) et le dataframe sélectionné.

A noter que cette fonction est aussi adaptée pour les questions présentées en une seule colonne.

1. La fonction MultiQCountPerc

Cette fonction est similaire à MultiQCount. La seule différence réside dans le fait qu’au lieu d’un ‘Count’, c’est une proportion en ‘Pourcentage’ qui sera renvoyée.

Les paramètres de cette fonction sont l’intitulé de la question ainsi que le dataframe sélectionné.

1. La fonction NewBarhCountList

Dans le cas des questions présentées sur plusieurs colonnes, cette fonction retourne la proportion en pourcentage de chacune des réponses sous forme de liste. Cette donnée nous sera utile pour la génération d’une visualisation via la fonction survey qui sera présentée plus bas.

Les paramètres de cette fonction sont l’intitulé de la question ainsi que le dataframe sélectionné.

1. Les fonctions permettant la génération de visualisations
2. La fonction BarGraph

Fonction retournant un histogramme horizontal affichant toutes les réponses différentes à une même question ainsi que le compte de chacune de ces réponses.

Les paramètres de cette fonction sont l’intitulé de la question ainsi que le dataframe sélectionné.

1. La fonction BarGraphPerc

Fonction retournant un histogramme horizontal affichant toutes les réponses différentes à une même question ainsi que le ratio en pourcentage de chacune de ces réponses.

Les paramètres de cette fonction sont l’intitulé de la question ainsi que le dataframe sélectionné.

1. La fonction BarYearGraphPerc

Fonction retournant un histogramme horizontal affichant toutes les réponses différentes à une même question ainsi que le ratio en pourcentage de chacune de ces réponses avec dissociation des données de 2021 et 2020.

Les paramètres de cette fonction sont l’intitulé de la question ainsi que le dataframe sélectionné.

1. La fonction survey

Fonction retournant un histogramme horizontal indiquant la proportion de chacune des réponses à une même question en fonction du poste des répondants.

Cette fonction a été reprise au lien ci-dessous puis adaptée :

<https://matplotlib.org/stable/gallery/lines_bars_and_markers/horizontal_barchart_distribution.html>

1. La fonction TopAnswersPerJob

Fonction permettant de générer un graphique représentant les proportions des 2, 3 ou 4 principales réponses à une question pour chaque métier. Elle réalise dans l'ordre :

- l'identification des principales réponses de la question pour l’ensemble du sondage

- le stockage dans un dataframe des proportions de ces réponses pour chaque métier

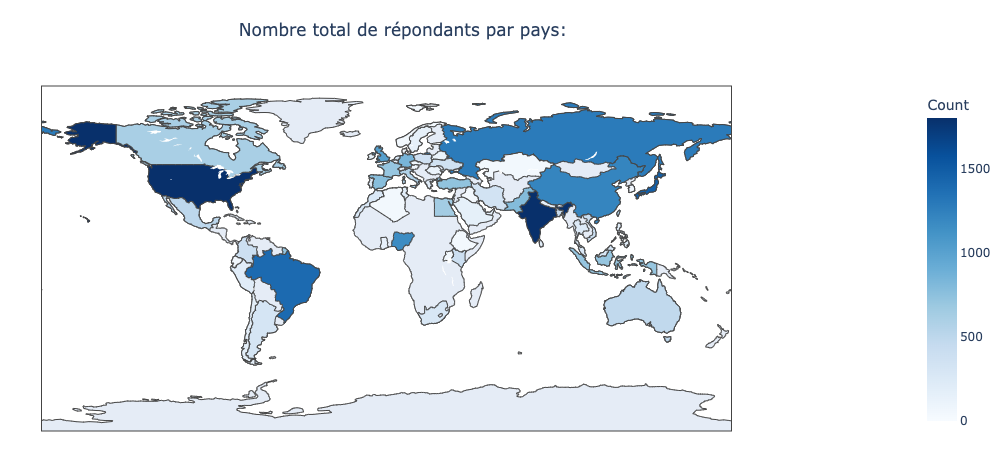
- l'affichage du graphique correspondant

1. La fonction CountMap

Fonction permettant de retourner une carte en couleur mettant en avant le pays de résidence des répondants.

## Biais du dataset

Voici une première analyse concernant le lieu de résidence des répondants et qui est importante pour comprendre le biais du dataset.



Cette carte met en avant la localisation géographique des différents répondants. Nous constatons que l’Inde est le pays le plus représenté (plus de 12,800 répondants), suivi par les Etats-Unis (4700 répondants). Le Japon, la Chine, le Royaume-Uni, le Brésil, le Nigéria et la Russie comptent quant à eux entre 1000 et 1500 représentants.

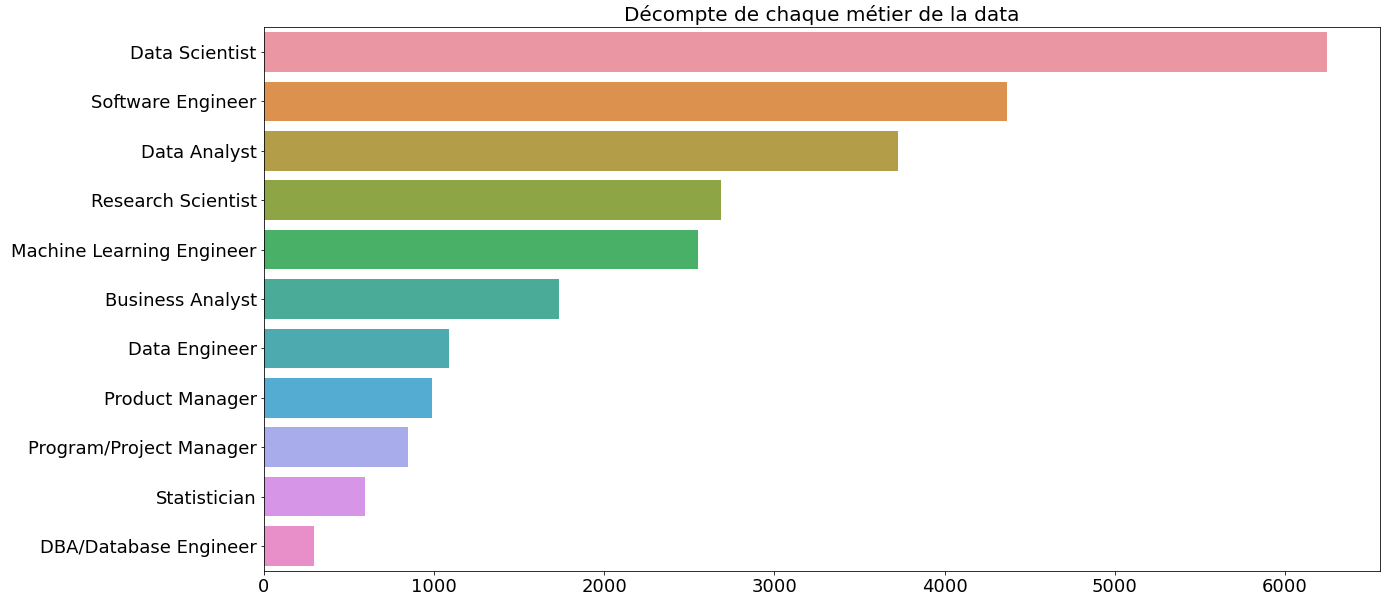
Ce premier constat nous appelle à faire preuve de vigilance quant à la pertinence de ce dataset. En effet, l’Inde est sur-représentée et ne figure qu’à la 51ème place du classement mondial des compétences en Data Science. Le détail du classement Coursera, ainsi que la méthodologie d’évaluation sont disponibles au lien suivant :

<https://www.sertacdoganay.com/wp-content/uploads/Coursera-Global-Skills-Index-2020.pdf>

Il est à noter que les pays comptabilisant moins de 50 répondants ne sont pas représentés sur cette carte.

## Périmètre de l’analyse

Nous nous sommes ensuite intéressés aux métiers de la data décrits dans le dataset. Voici leur répartition :



L’objectif de notre étude étant d’analyser les techniques et outils utilisés par les professionnels de la data, nous nous sommes focalisés sur les métiers les plus représentés afin de rendre les visualisations claires et facilement lisibles. Nous avons pris le parti d’afficher uniquement les 7 métiers les plus représentés dans le questionnaire : Data Scientist, Software Engineer, Data Analyst, Research Scientist, Machine Learning Engineer, Business Analyst et Data Engineer.

Dans le même souci de clarté, nous avons concentré notre étude sur les questions qui nous semblaient les plus pertinentes et en lien à l’énoncé du projet, à savoir :

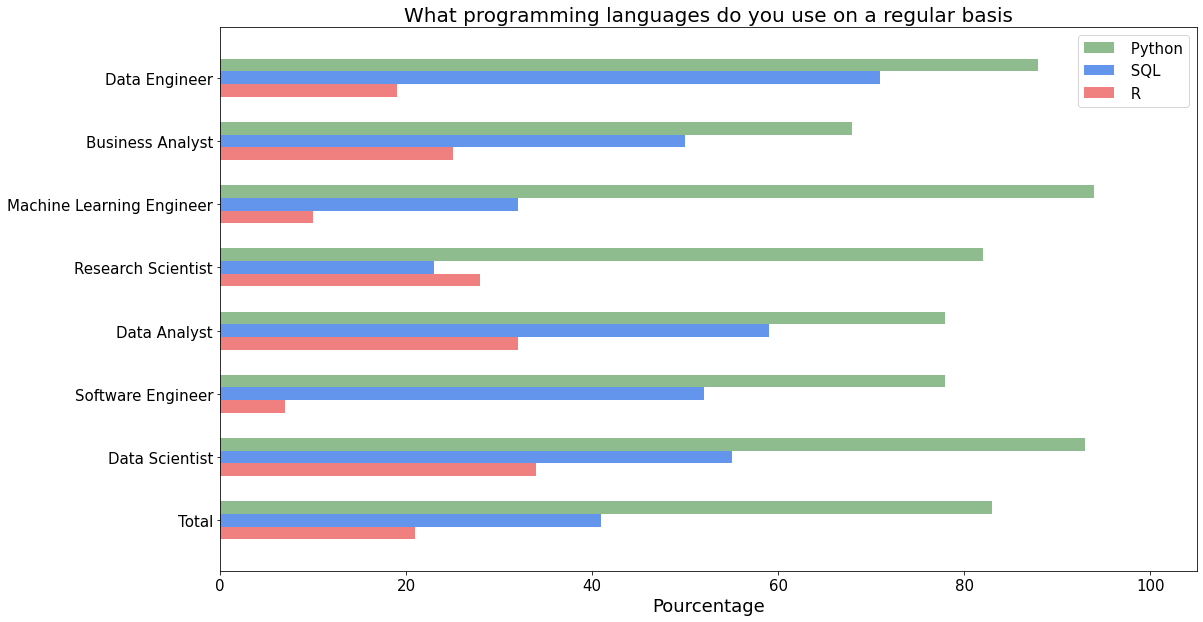
* La question 7 : Les langages de programmation
* La question 14 : Les librairies de visualisation
* La question 16 : Les librairies de machine learning
* La question 17 : Les types de modèles utilisés pour le machine learning
* La question 34 : Les outils de Business Intelligence
* La question 41 : Le type de logiciel privilégié pour analyser des données

Dans chacun de nos graphiques, nous avons affiché les proportions de réponses de chaque métier en sélectionnant au moins les 2 réponses les plus représentées (nous sommes parfois allés jusqu’à sélectionner 3 ou 4 réponses, selon la question).

## Analyse des données

### Q7 : Langage de programmation

La première question porte sur les langages de programmation utilisés par les répondants. Ci-dessous un premier graphique représentant les pourcentages d’utilisation des 3 langages les plus utilisés par les principaux métiers sur le total des répondants :



Sur le premier histogramme, nous pouvons voir que Python reste le langage de référence en Data Science. En effet, c’est le langage le plus utilisé dans chacun des métiers et il est maîtrisé par plus de 80% des répondants.

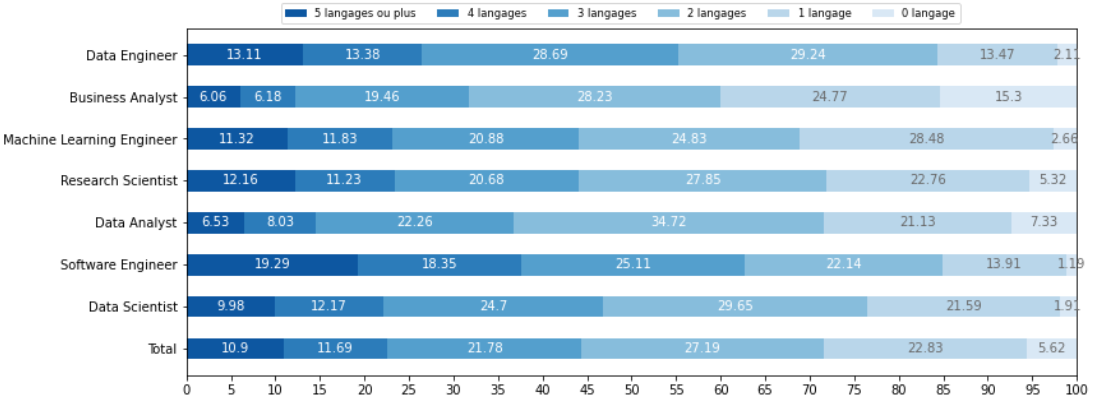
La hiérarchie des langages est la même partout (Python, puis SQL, puis R). Exception faite des Research Scientists qui privilégieront l’utilisation de R à celle de SQL.

Nous pouvons également remarquer que les Data Engineer et les Software Engineer ont un profil assez proche. En effet, nous constatons pour ces deux professions une proportion importante de Python et de SQL mais une utilisation moindre de R.

Enfin, nous nous apercevons que les Business Analyst et les Data Analyst ont une répartition très semblable pour les 3 langages. Les proportions pour les Data Scientist sont d’ailleurs ressemblantes à celles de ces deux métiers mais la prépondérance de Python est plus marquée.

Les deux derniers métiers, que sont Machine Learning Engineer et Research Scientist, forment un groupe à part avec une préférence marquée pour le langage Python.

Ci-dessous, un deuxième graphique représentant la proportion du nombre de langages utilisés dans chaque métier :



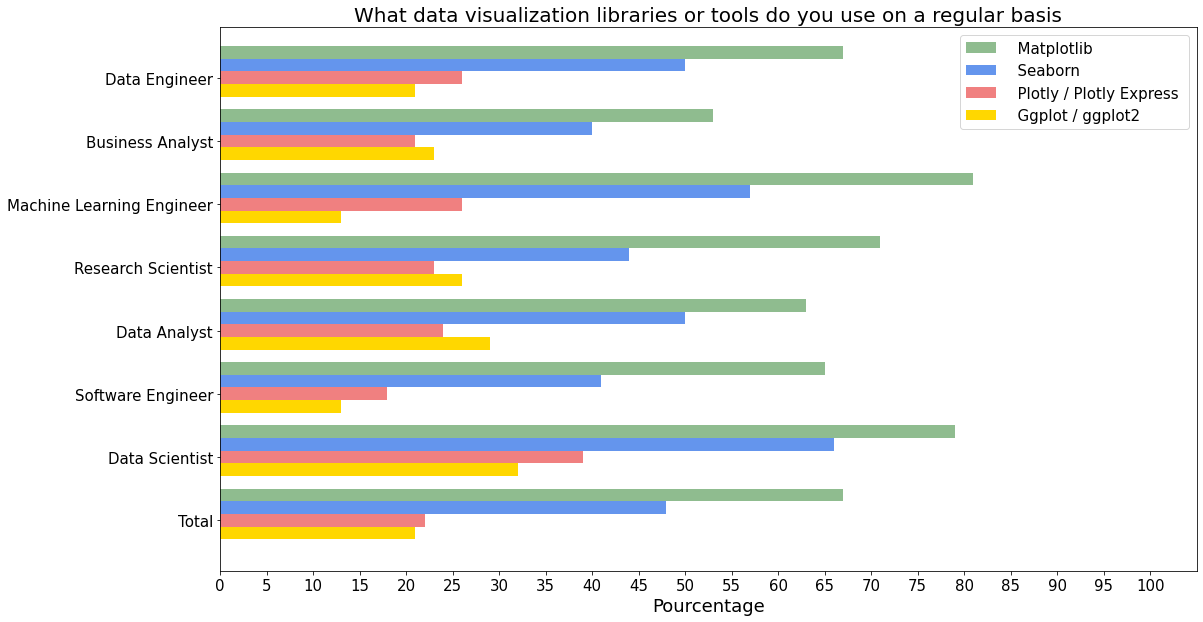
Nous constatons que seuls 5,62% des répondants n’en maîtrisent aucun. Ce chiffre confirme le caractère essentiel des langages de programmation dans le secteur de la Data.

Cet histogramme met aussi en exergue les métiers les plus techniques. Seuls les Software Engineers et les Data Engineers maîtrisent 2 langages ou plus dans une proportion supérieure à 85%.

A contrario, nous constatons que 15% des Business Analysts ne maîtrisent aucun langage : une proportion bien supérieure à la moyenne générale. De plus, seuls 60% d’entre eux maîtrisent 2 langages ou plus.

### Q14 : Librairies de visualisation

La deuxième question retenue porte sur les librairies de visualisation utilisées par les répondants. Ci-dessous un graphique représentant le taux d’utilisation des 4 librairies de visualisation les plus utilisées par les principaux métiers sur le total des répondants :

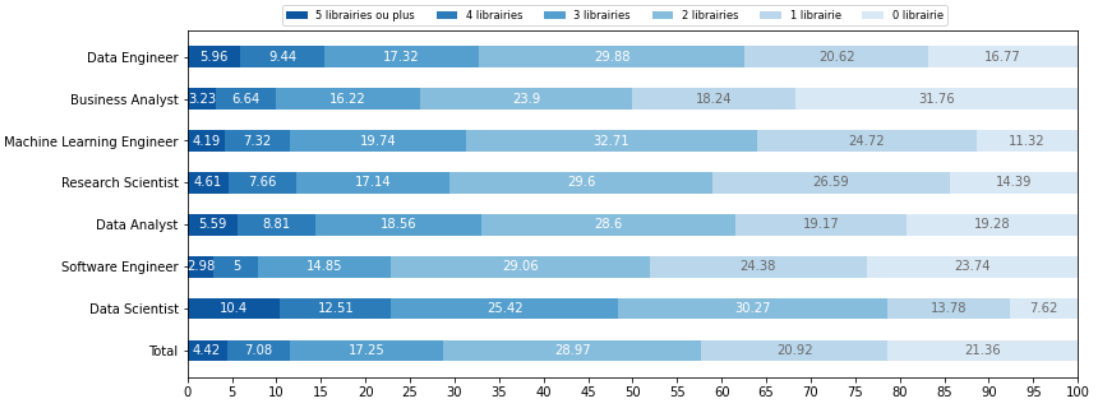


Le top 2 est identique pour tous les métiers (Matplotlib et Seaborn). Cependant, nous pouvons nous apercevoir que Plotly et Ggplot se partagent les 3ème et 4ème places.

Les Data Engineers, ML Engineers, Software Engineers et les Data Scientists préfèreront Plotly quand les Business Analysts, Research Scientists et les Data Analysts choisiront plutôt Ggplot.

Nous constatons que seuls 67% des Data Engineers utilisent Matplotlib. Ce score les classe uniquement au 4ème rang derrière les ML Engineers, les Data Scientists et les Research Scientists. Ce score semble logique puisque, même si la frontière entre les différents métiers de la Data est parfois fluctuante, le Data Engineer n’a pas pour vocation première la conception de visualisations et graphiques.

Voici ensuite un deuxième graphique représentant la proportion du nombre de librairies de visualisation utilisées dans chaque métier :

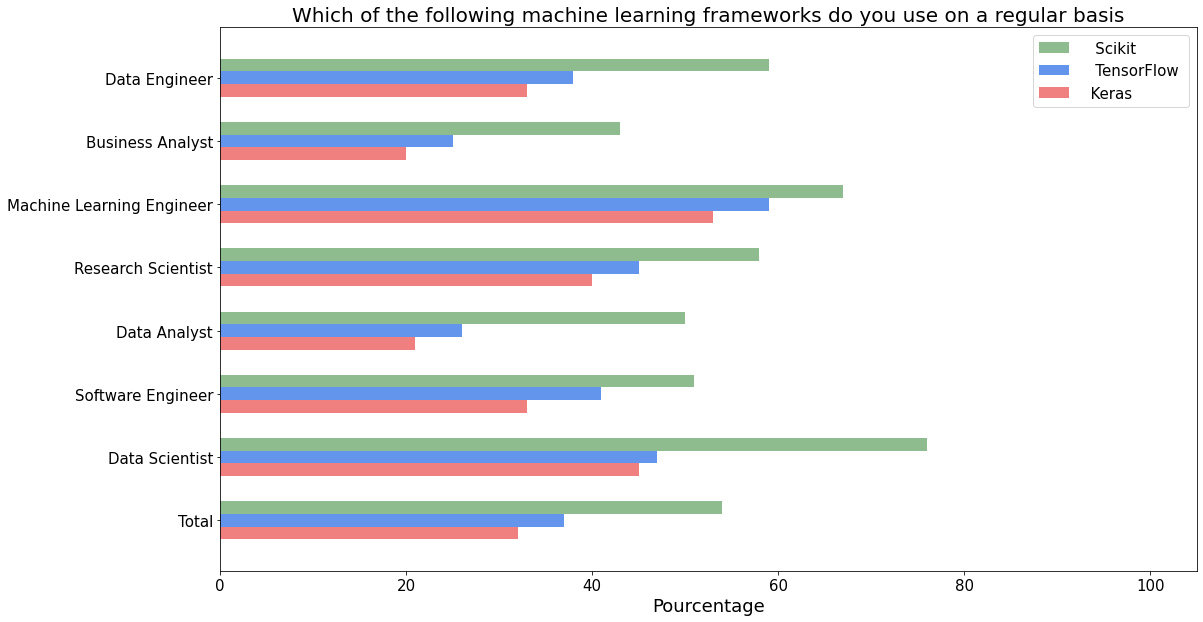


Ici on remarque que près d’un tiers des Business Analysts n’utilisent aucune librairie. Une proportion record sur le panel.

De leur côté, 79% des Data Scientists en maîtrisent à minima 2. Un ratio bien supérieur à celui des autres professions (15% de plus que les ML Engineers en deuxième position), confirmant ainsi, l’importance des connaissances techniques nécessaires au métier de Data Scientist.

### Q16 : Librairies de machine learning

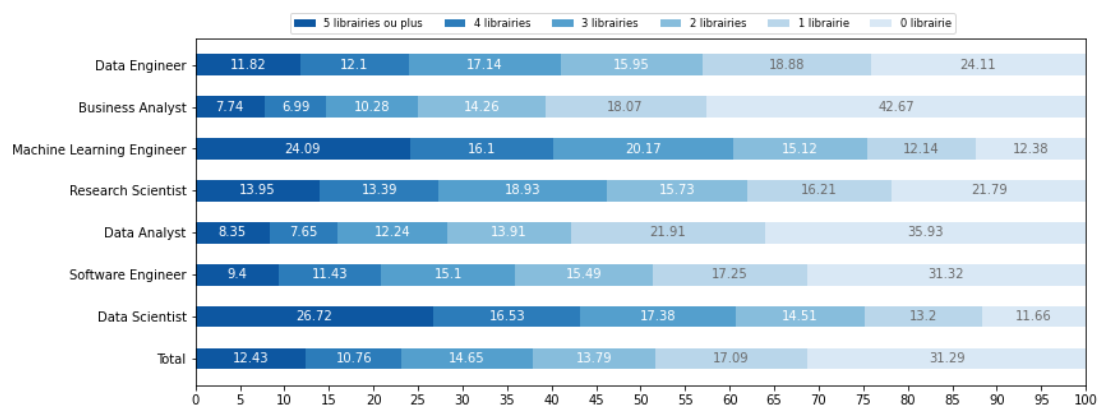
La troisième question sélectionnée porte sur les librairies de machine learning utilisées par les répondants. Ci-dessous un graphique représentant le taux d’utilisation des 3 librairies de machine learning les plus utilisées par les principaux métiers sur le total des répondants :



La hiérarchie est identique pour tous les métiers (Scikit, puis TensorFlow, et enfin Keras).

Les Data Scientists ont la plus grande proportion d’utilisateurs de Scikit (75%) et ce devant les ML Engineers (67%).

Voici ensuite un deuxième graphique représentant la proportion du nombre de librairies de machine learning utilisées dans chaque métier :

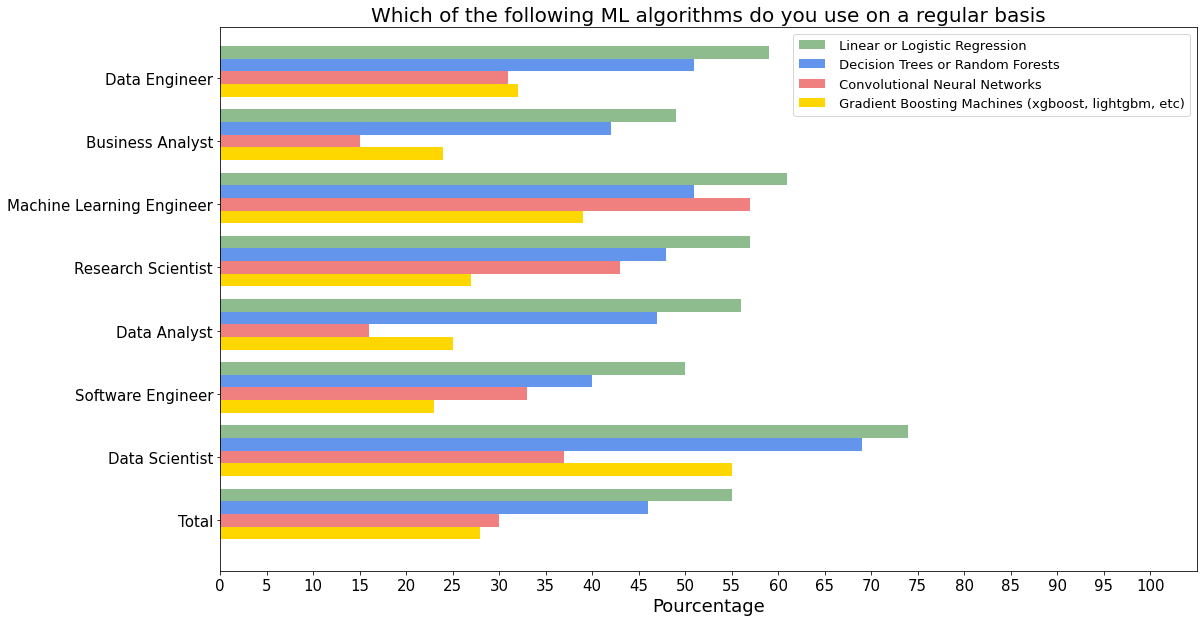


On peut constater que la proportion de ML Engineers maîtrisant 2 frameworks ou plus est équivalente à celle des Data Scientists (soit près de 75%), les plaçant loin devant les Data Engineers ou les Research Scientists pointant respectivement à 57% et 61%.

A nouveau, ces graphiques mettent en avant le niveau de polyvalence nécessaire à la profession de Data Scientist puisque ces derniers semblent avoir des connaissances équivalentes ou supérieures à celles de ML Engineers en ce qui concerne les frameworks de machine learning.

### Q17 : Types de modèles utilisés pour le machine learning

La quatrième question porte sur les types de modèles utilisés par les répondants pour le machine learning. Ci-dessous un graphique représentant les taux d’utilisation des 4 types de modèles les plus utilisés par les principaux métiers sur le total des répondants :

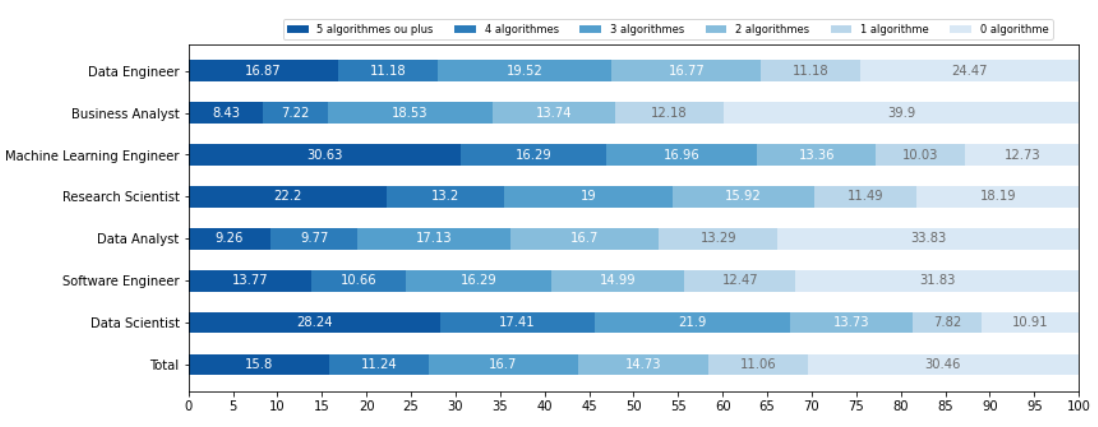


Les régressions linéaires ou logistiques sont les algorithmes les plus utilisés dans tous les métiers de la Data.

En seconde place, nous retrouvons, pour tous, les arbres de décision et les forêts aléatoires. Exception faite des ML Engineers qui préfèreront les réseaux neuronaux convolutifs. Ce constat semble logique puisque de nombreux ingénieurs se spécialisent en Deep Learning. Les réseaux neuronaux leurs sont alors essentiels pour développer des algorithmes basés sur la reconnaissance visuelle.

Là encore, nous pouvons constater la polyvalence nécessaire au métier de Data Scientist: eux seuls utilisent à plus de 55% au moins 3 algorithmes (les régressions linéaires, les arbres de décisions et les méthodes de Gradient Boosting).

Voici ensuite un deuxième graphique représentant la proportion du nombre d’algorithme de ML utilisés dans chaque métier :



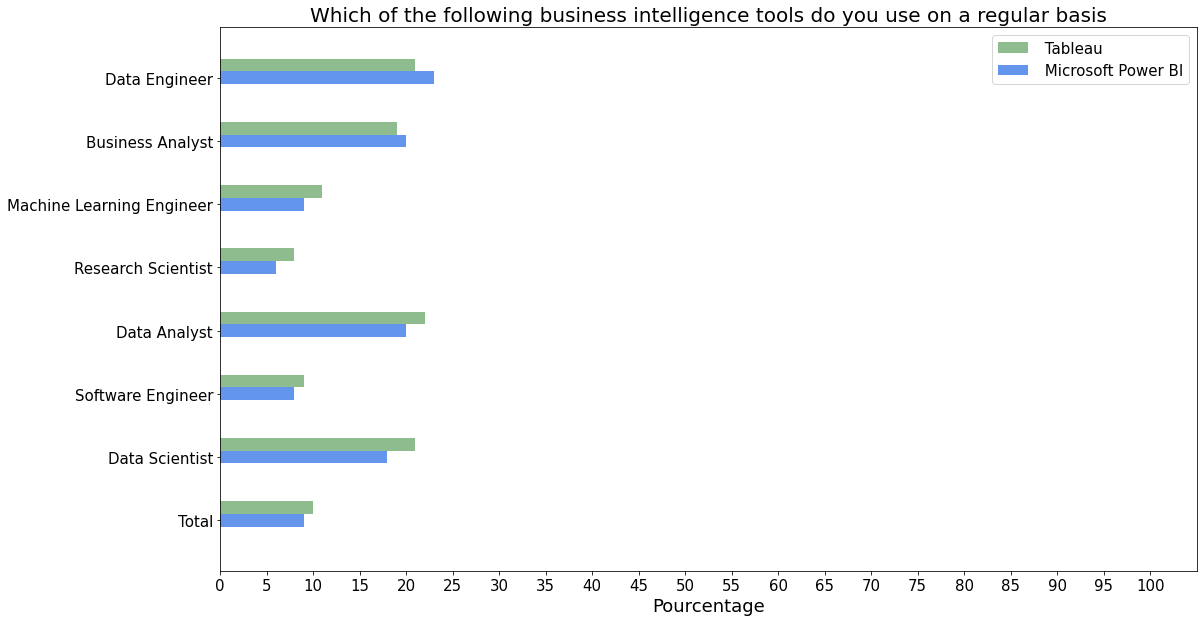
Ce graphique met en exergue les métiers les plus demandeurs en Machine Learning. Comme nous pouvions nous en douter, ce sont les ML Engineers et les Data Scientists qui utilisent le plus régulièrement des algorithmes de ML. En effet, environ 65% d’entre eux utilisent au moins 3 algorithmes différents dans leurs tâches quotidiennes.

Les Research Scientists ont eux aussi un ratio d’utilisation bien supérieur à la moyenne : 55% d’entre eux utilisent régulièrement 3 algorithmes quand la moyenne générale du panel est de moins de 45%.

Les métiers de Data Analysts, Business Analysts et Software Engineers sont les moins demandeurs en Machine Learning. 40% des Business Analysts et 34% des Data Analysts n’utilisent aucun algorithme de ML. Ces proportions semblent cohérentes puisque ces types de postes ne nécessitent pas toujours l’établissement de modèles prédictifs.

### Q34 : Outils de Business Intelligence

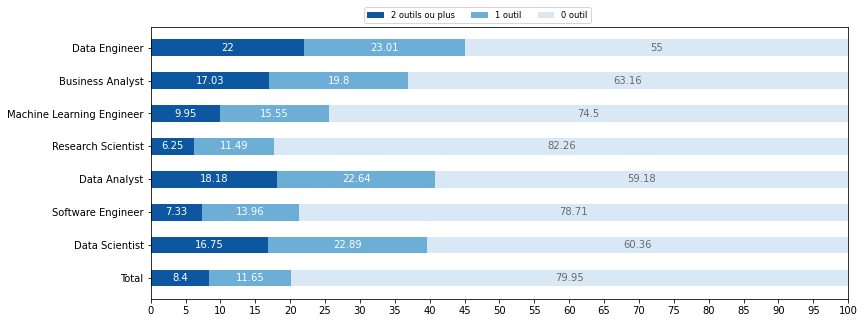
La cinquième question porte sur les outils de Business Intelligence utilisés par les répondants. Ci-dessous un graphique représentant les taux d’utilisation des 2 outils les plus utilisés par les principaux métiers sur le total des répondants :



Nous constatons que les taux d’utilisations de PowerBI et Tableau sont très proches.

Parmi les utilisateurs d’outils de BI, les Data Engineers et les Business Analysts préfèrent légèrement PowerBI. Les ML Engineers, les Research Scientists, les Data Analysts, les Software Engineers et les Data Scientists privilégient, quant à eux, l’utilisation de Tableau.

Voici ensuite un deuxième graphique représentant la proportion du nombre d’outils BI utilisés pour chaque métier :



Nous remarquons que les outils de Business Intelligence sont peu utilisés par les répondants. Près de 80% d’entre eux n’utilisent ni PowerBI ni Tableau.

Les principaux utilisateurs de ces outils sont les Data Scientists, les Data Analysts, les Data Engineers ainsi que les Business Analysts. Plus de 35% d’entre eux en utilisent au moins régulièrement.

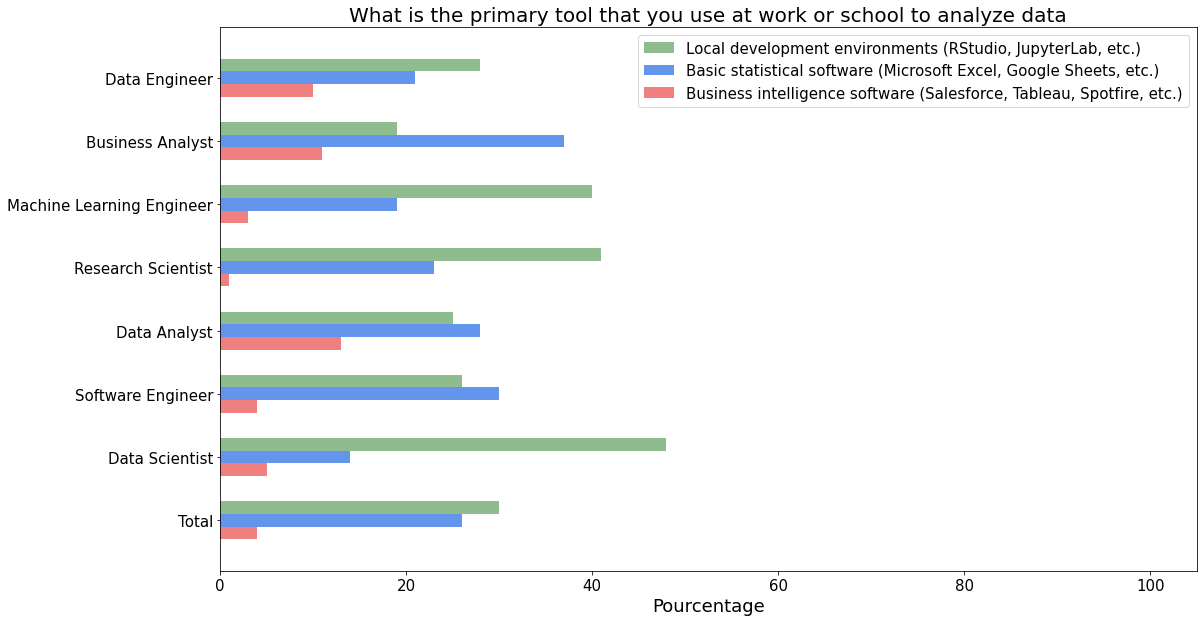
Le pourcentage semble cependant très faible. Les explications les plus plausibles sont les suivantes :

* Le dataset n’est pas entièrement représentatif de la réalité des métiers de la Data (sur-représentation de répondants venant d’un même pays. En effet, comme énoncé plus haut, plus d’un quart des sondés travaillent en Inde)
* Les répondants utilisent dans leur grande majorité des outils de BI mais pas de manière régulière. Nous pouvons imaginer que ces derniers sont utilisés seulement pour les projets les plus importants par exemple. Ou encore uniquement dans les grandes entreprises car les licences peuvent s’avérer coûteuses.

### 

### Q41 : Types de logiciel privilégié pour analyser des données

La sixième question porte sur les types de logiciel privilégiés par les répondants pour analyser des données. Ci-dessous un graphique représentant les taux d’utilisation des 3 types de logiciels les plus utilisés par les principaux métiers sur le total des répondants :



Sur cet histogramme nous constatons que les environnements de développement tels que RStudio et JupyterLab ne constituent pas les outils principaux de tous les métiers de la Data.

En effet, les Business Analysts, les Data Analysts ainsi que les Software Engineers préfèrent des outils basiques tels que Google Sheets ou Excel.

Les outils de Business Intelligence, quant à eux, sont davantage utilisés par les Data Analyst et les Business Analyst.

Ces constats confirment nos précédentes déductions. Les métiers d’analystes nécessitent moins de connaissances en langages de programmation que les autres métiers du panel. Cependant, les analystes devront rendre les données accessibles et compréhensibles par des personnes moins techniques. C’est pourquoi, la maîtrise des outils de Business Intelligence leur est très souvent essentielle.

## 

## Conclusion

Au travers de ces différentes visualisations, nous pouvons nous rendre compte que tous les métiers de la data utilisent globalement les mêmes outils, mais pas tout à fait de la même manière.

En effet, même si les langages de programmation tels que Python, SQL ou R paraissent essentiels dans ce secteur, les connaissances des Research Scientist, Machine Learning Engineer, Software Engineer et Data Engineer semblent plus approfondies dans ces langages.

De par cette première analyse nous pouvons définir une frontière “technique” entre les professions.

Ce postulat est confirmé par l’analyse de l'utilisation des outils de Machine Learning et des librairies de visualisations. A nouveau, les Research Scientist, Machine Learning Engineer, Software Engineer et Data Engineer semblent avoir des connaissances plus poussées. Ces derniers maîtrisent mieux et utilisent plus régulièrement ce type d’outils ou de librairies.

Les professions de Data Analyst et Business Analyst se caractérisent, quant à elles, par une utilisation plus régulière des logiciels de Business Intelligence. Ceci, de par la nécessité de rendre les données compréhensibles par les profils moins techniques.

Enfin, cette étude a permis de mettre en avant la polyvalence nécessaire au métier de Data Scientist. Ces derniers figurent toujours parmi les profils les plus techniques dans chacun des domaines.

On peut donc conclure que certains métiers de la data ont des similitudes dans leurs pratiques et semblent avoir un profil technique commun. Cependant, tous les métiers favorisent généralement les mêmes outils et librairies, et il sera peut-être compliqué de les différencier lors de la modélisation.

# 

# Modélisation

Après l’analyse de notre jeu de données, la deuxième partie de notre projet a été de développer un modèle capable de prédire le métier d’un répondant à partir de ses réponses au sondage.

## Préparation des données

En vue de la modélisation, nous avons transformé nos données pour pouvoir y appliquer des modèles de machine learning.

Pour ce faire, nous avons d’abord retiré toutes les réponses des répondants qui sortaient du cadre de l’objectif du modèle. Ce retrait concernait :

* Les étudiants
* Les personnes sans emplois
* Les personnes ayant sélectionné l’option “Other” lorsqu’il leur a été demandé leur profession
* Les personnes ayant sélectionné l’option “Developer Relations/Advocacy” lorsqu’il leur a été demandé leur profession. Nous avons fait ce choix car cette catégorie ne correspondait qu’à 99 répondants (très peu représentatif donc)

Nous avons aussi retiré la colonne ‘Year’ car elle risquait de fausser les résultats.

Suite à celà, nous avons modifié la colonne target en remplaçant chaque type d’emploi restant par un chiffre ou un nombre différent :

‘Data Scientist' : 1,

'Software Engineer' : 2,

'Data Analyst' : 3,

'Research Scientist' : 4,

'Machine Learning Engineer' : 5,

'Business Analyst' : 6,

'Data Engineer' : 7,

'Product Manager' : 8,

'Program/Project Manager' : 9,

'Statistician' : 10,

'DBA/Database Engineer' : 11

Les colonnes restantes étaient divisées en 2 catégories :

* Catégorie 1 : les colonnes ‘mono’ correspondant à des questions présentées en plusieurs colonnes. Chaque colonne contenant une seule possibilité de réponse à la question.
* Catégorie 2 : les colonnes ‘multi’ correspondant aux questions dont toutes les réponses se situaient dans la même colonne

Comment identifier rapidement la catégorie de chacune des colonnes ?

Nous sommes partis du constat que les colonnes de la catégorie 1 ne contenaient que 2 types de réponses : soit la réponse correspondant à la colonne, soit un NaN.

A contrario, les colonnes correspondant à la catégorie 2 contenaient toujours 3 types de réponses ou plus.

Le nombre de réponses différentes par colonne nous a un ainsi permis de dissocier facilement les colonnes appartenant la catégorie 1 de celles appartenant à la catégorie 2.

Une fois chaque colonne catégorisée, la préparation du dataframe était facile puisqu’il nous suffisait de :

* Remplacer les Not Nan par des 1 et les Nan par des 0 sur les colonnes de la catégorie 1
* Appliquer un get\_dummies sur les colonnes de la catégorie 2

Notre dataframe était alors prêt pour appliquer les différents modèles de Machine Learning.

## Première itération de modèles

Nous avons isolé 20% du jeu de données comme groupe de test. Ensuite nous avons entraîné et mesuré les performances des principaux modèles de classification, à savoir :

* Régression logistique
* SVM
* KNN
* Arbre de décision
* Boosting
* Bagging
* Forêt aléatoire

Le meilleur score était celui de la régression logistique avec une précision de 45,7% seulement. Ce résultat battait largement celui d’un modèle aléatoire (9% de précision pour un modèle aléatoire avec 11 classes à prévoir) mais il était insuffisant pour un modèle de machine learning.

Nous avons donc cherché à modifier notre dataframe en amont afin de tenter d’améliorer cette précision.

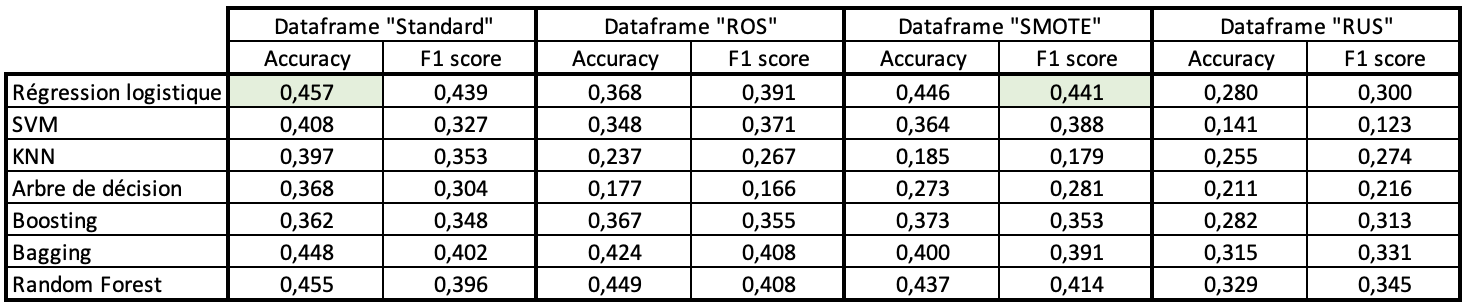
## Optimisation des performances

### Rééchantillonnage

Notre première piste d’amélioration a été de rééquilibrer les classes de notre target en appliquant différentes data-level solutions :

* Du sous-échantillonnage grâce à la classe RandomUnderSampler de la librairie Imblearn
* Du sur-échantillonnage grâce aux classes RandomOverSampler et SMOTE de la librairie Imblearn

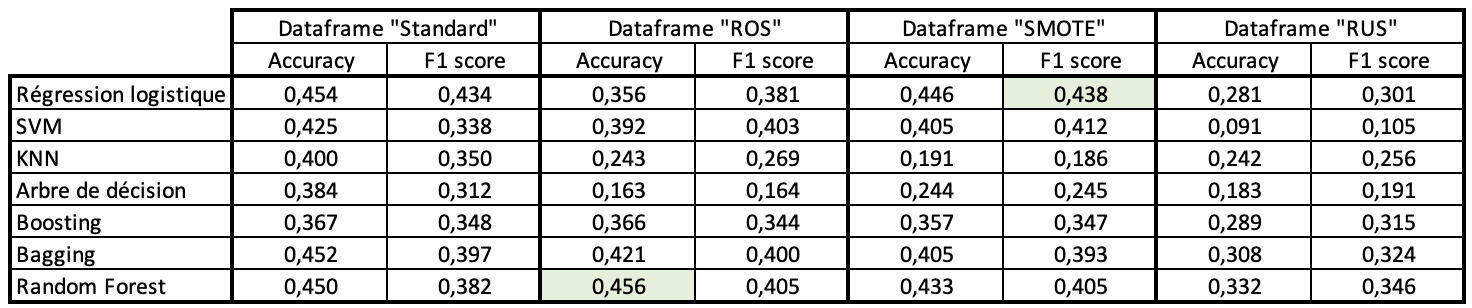
Ces résultats n’ont pas été concluants, les performances ont été soit équivalentes soit inférieures. Vous trouverez ci-dessous le détail des Accuracy et F1 Score pour chacun des Dataframes et des modèles :



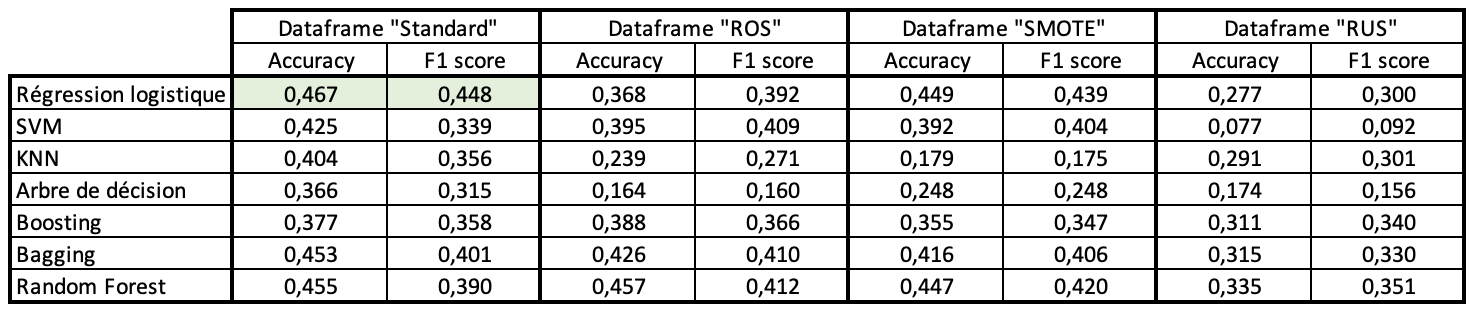
### Changement de la taille du groupe de test

Une autre piste que nous avions était de modifier la taille de l’échantillon de test. Ci-dessous les résultats avec des test\_size de 15% et 25%.

Taille d’échantillon de 15% :



Taille d’échantillon de 25% :



Comme nous pouvons le constater, cette modification de test\_size a amélioré les résultats de certains modèles. En effet, avec un échantillon test de 25% sur un DataFrame “Standard”, nous avons obtenu une régression logistique avec une précision de 46,7% et un F1 Score de 0,448. Mais cette amélioration restait négligeable.

### Validation croisée

Nous avons ensuite utilisé une fonction GridSearchCV pour trouver les meilleurs hyperparamètres pour notre modèle de régression logistique car c’est lui qui donnait les meilleurs résultats parmi tous les modèles.

La validation croisée nous a permis de trouver les hyperparamètres suivants :

* C = 0.1
* solver = 'newton-cg'
* max\_iter = 100

Cependant, cela n’a amélioré la précision de notre modèle que de 1 à 2%, donc encore une fois négligeable.

### Réduction du jeu de données

Nous avons également essayé de réduire le jeu de données :

* Soit en sélectionnant uniquement des questions du sondage qui nous semblaient pertinentes
* Soit en supprimant du jeu de données complet des questions qui nous semblaient trop éloignées du sujet

Dans un cas comme dans l’autre, et même en essayant différentes combinaisons de questions, les performances étaient systématiquement inférieures à celles d’un modèle entraîné sur le jeu de données complet.

### Réduction du nombre de classe

Enfin, notre dernière piste a été de diminuer le nombre de métiers à prédire. En effet, certaines professions étant pratiquées uniquement par moins de 2,000 répondants, ce “petit” nombre n’était peut-être pas assez représentatif pour établir un profil métier. Nous avons alors supprimé par étape les réponses du dataframe correspondant aux métiers les moins représentés.

Cette solution est celle qui a donné les meilleurs résultats. Avec 11 classes, le modèle de régression logistique atteignait environ 45% de précision. Plus le nombre de classes à prédire était réduit et plus la précision augmentait. Finalement, en réduisant le jeu de données aux 3 métiers les plus représentés (Data Scientist, Software Engineer et Data Analyst), la précision des différents types de modèle atteignait environ 70%. C’était certes un taux bien meilleur qu’avec les autres solutions, mais il restait décevant si on considérait qu’un modèle de machine learning devait avoir au minimum une précision de 80%. De plus, ce score a été atteint au prix d’un modèle ne pouvant prédire que 3 métiers différents, ce qui diminuait grandement son intérêt.

## Conclusion

Nos premiers essais de modélisation nous ont permis d’identifier les modèles prédictifs les plus performants sur notre jeu de données : régression logistique, SVM, bagging et forêt aléatoire.

Malgré nos multiples tentatives pour améliorer les résultats de notre modèle le plus efficace basé sur la régression logistique, nous n’avons pas réussi à augmenter sa précision à un niveau satisfaisant.

Si l’on fait le lien entre ces résultats et les analyses que nous avons pu faire pendant la partie data visualisation de ce projet, nous pouvons émettre l’hypothèse que les différents métiers de la data ont trop de choses en commun et utilisent, relativement, les mêmes techniques et outils. Les proportions d’utilisation de ces outils changent évidemment d’un métier à l’autre, mais trop peu pour que nous puissions établir un profil technique précis associé à chaque métier de la data.