RAPPORT

PROJET FIL ROUGE
DATA JOB

Table des matières

Introduction :

Ce rapport rend compte de notre travail sur le projet fil rouge de la formation Data Analyst dispensée par l'organisme de formation Datascientest.

Le sujet de notre projet était une étude du profil des personnes qui travaillent dans les métiers de la data.

Le but du projet est de créer un outil qui permet par le choix de critère de définir le métier de la data correspondant.

La réalisation de notre projet s'est déroulée en plusieurs étapes :

- **Première étape :** prise de connaissance du sujet et premières observations/conclusions.
- **Deuxième étape :** prise de connaissance du jeu de données et premières questions.
- **Troisième étape :** appropriation et préparation du jeu de données.
- **Quatrième étape :** transformation de nos questions/observations en graphique.
- **Cinquième étape :** preprocessing et nettoyage des données.
- **Sixième étape :** première itération de modèles de machine learning.
- **Septième étape :** seconde itération de modèles de machine learning.
- **Conclusion.**

Première étape : prise de connaissance du sujet et premières observations/conclusions

Lorsque nous avons pris connaissance du sujet qui nous a été attribué pour notre projet fil rouge, voici les premières observations que nous avons eues :

Sujet : Analyse des techniques et outils utilisés par les professionnels de la Data

Nature des données : Notre projet s'appuie sur une enquête réalisée par le site Kaggle.com. Elle est composée de 39 questions, 18 d'entre elles sont des questions à choix unique, le reste étant à choix multiples.

L'enquête a été réalisé sur une période de 3 mois, de juillet 2020 à septembre 2020 et a été transmise à l'ensemble des personnes qui sont inscrites sur le site Kaggle.com.

Un total de 171 nationalités ont pris part à cette enquête.

Initialement ce jeu de données a été constitué pour la compétition annuelle Machine Learning et Data Science de Kaggle qui consiste à réaliser un notebook qui présente et raconte l'histoire des données récoltées dans l'enquête.

Quel est l'objectif du projet ?

- Identifier les différents profils techniques qui se sont créés dans l'industrie de la Data.
- Quelles sont les tâches effectuées par les personnes qui travaillent dans la Data ?
- Quels sont les outils qui sont utilisés par les personnes qui travaillent dans la Data ?
- Définir quels sont les outils et les compétences attendues par les personnes qui travaillent / souhaitent travailler dans la data.

Conclusion :

Assez rapidement nous avons compris que les objectifs finaux de notre projet seraient d'une part, l'analyse et la présentation des données. Et d'autre part la création d'un modèle machine learning capable de classer en fonction d'une liste de critères un individu sur l'un des métiers cibles retenus.

Deuxième étape : prise de connaissance du jeu de données et premières questions

Une fois que nous avons pris connaissance du sujet de notre projet fil rouge, nous avons découvert le jeu de données.

Celui-ci est composé de :

- 20036 lignes de données.
- 355 colonnes.

La majorité des données sont des données textuelles. Il y a très peu de données numériques dans celui-ci.

Le premier sentiment partagé par le groupe a été la surprise face au nombre de lignes et de colonnes que nous allions avoir à analyser. Notre groupe est composé de profils assez divers et aucun d'entre nous n'avait été jusqu'ici confronté à une telle masse de données.

Avec le recul que nous avons aujourd'hui, nous réalisons que notre jeu de données n'est pas si imposant que cela au regard de ce que les Data Analyst en poste peuvent avoir à traiter chaque jour.

Lors de cette première lecture du jeu de données, nous avons relevé les points suivants :

1. Les deux premières lignes font office de titre pour chaque colonne.
2. Les titres des colonnes sont en fait les questions en entier de l'enquête d'où sont issues les données.
3. Il sera nécessaire de réaliser un premier travail de simplification des titres de chaque colonne.

Conclusion :

Au moment de la découverte du jeu de données, nous sommes sûrs de plusieurs choses, un travail de retraitement/harmonisation des données sera nécessaire. Nous pensons qu'avec la masse de données en notre possession, la création de l'outil de classification semble plus que réalisable.

Troisième étape : appropriation et préparation du jeu de données

Pour cette étape, Datascientest a mis à notre disposition un tableau : « Rapport exploration des données » pour nous aider à nous approprier le jeu de données.

Le but ici était des plus simples : réaliser une analyse exhaustive du jeu de données. En pratique, il s'agissait d'identifier pour chaque colonne les éléments suivants :

- Le type de variable,
- Décrire les variables,
- La disponibilité de cette variable,
- Le type informatique (int, str, etc.),
- Le taux de NA (valeur manquante),
- Comment est-ce que l'on envisageait la gestion des NA,
- Etc.

La complétion de ce tableau a été très utile pour nous aider à nous approprier le jeu de données. Cela nous a permis d'identifier les colonnes qui nous permettent de répondre à la problématique projet. Ainsi que l'identification, de celles qui ne nous apportent aucune valeur ajoutée.

Une fois le rapport d'exploration des données complété, nous avons été en mesure de réaliser un premier « retraitement » du jeu de données.

Celui-ci a consisté en la suppression de l'une des deux lignes de titre des colonnes et en la simplification des questions afin de les transformer en titre exploitable dans python (raccourcissement de ceux-ci, suppression des espaces, suppression des majuscules, etc.)

Exemple de transformation des titres de chaque colonne :

Méthodologie : suppression des espaces, des majuscules, des caractères spéciaux.

What is your age (# years) ? => *age*.
What is your gender? => *gender*.
In which country do you currently reside? => *country*.
What is the highest level of formal education that you have attained or plan to attain within the next 2 years? => *diploma*.

Conclusion :

Cette étape du projet fil rouge, nous a permis de rendre exploitable dans python le jeu de données. Nous avons de plus effectué un premier tri sur les colonnes qui nous semblaient pertinentes pour le projet. Cette étape était primordiale pour les prochaines étapes du projet, notamment pour la partie visualisation et pour la partie machine learning.

Quatrième étape : transformation de nos questions/observations en graphique.

L'étape de Data-Viz' nous a permis de mettre en perspective les premières observations et questions que nous avions sur notre jeu de données.

Un peu de contexte :

Cible : les personnes qui sont intéressées par les métiers de la data et qui veulent savoir quelles sont les compétences attendues dans ce domaine d'activité.

Ce que l'on veut montrer :

1. Visualisation des jobs de la data.
2. Visualisation des diplômes des personnes qui travaillent dans la data.
3. Visualisation de l'âge des personnes qui travaillent dans la data.
4. Visualisation de la répartition par genre des personnes qui travaillent dans la data.
5. Visualisation des langages de programmation utilisés dans la data.
6. Visualisation du/des langages de programmation(s) recommandé(s).
7. Autre(s).

Comment : à l'aide de graphiques (matplotlib, seaborn, etc.) mettre en valeur la répartition des données à notre disposition.

Réaliser une courte analyse des données :

- Qu'est-ce que l'on voit/constate ?
- Qu'est-ce que l'on peut en déduire ?

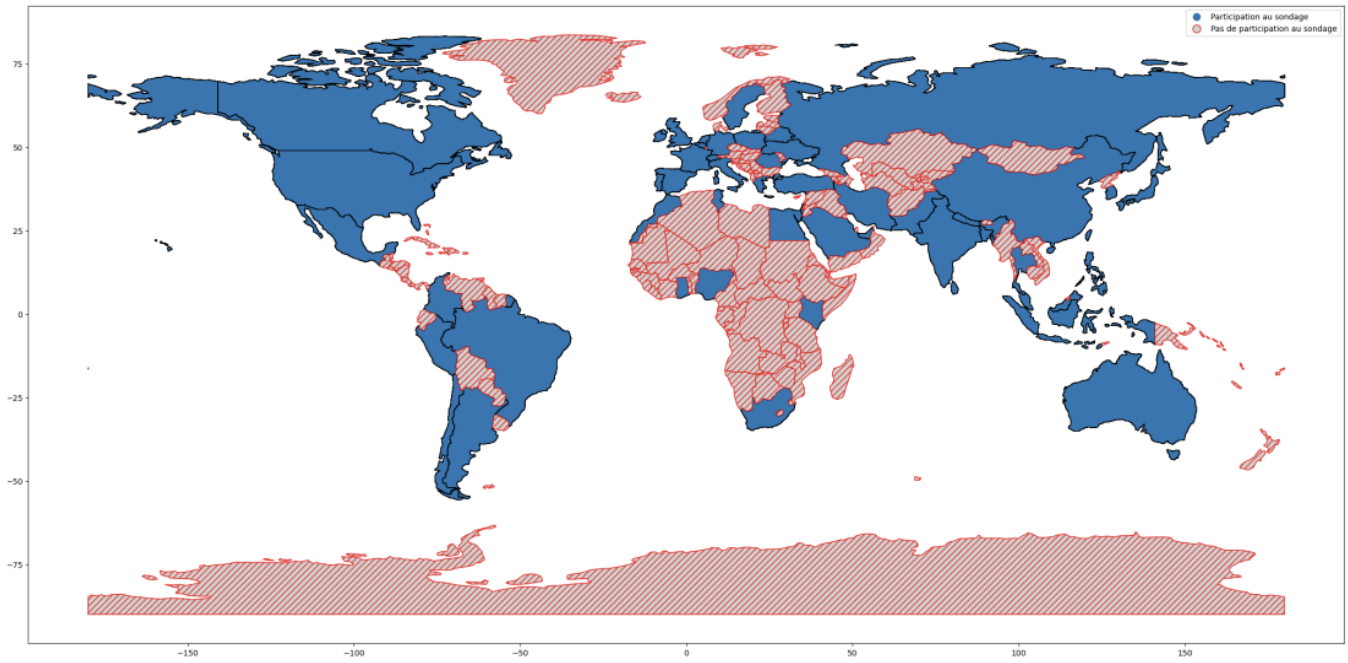
Nous avons ici décidé de procéder avec la méthode de l'entonnoir, en partant d'une analyse générale puis en rétrécissant l'axe d'analyse de plus en plus afin de voir s'il était possible avec le jeu de données en l'état d'identifier les critères/colonnes pour la classification des métiers de la data.

Nous avons ainsi, dans la première partie de notre data-viz réalisée une identification et classification des répondants, via des graphiques qui représentaient :

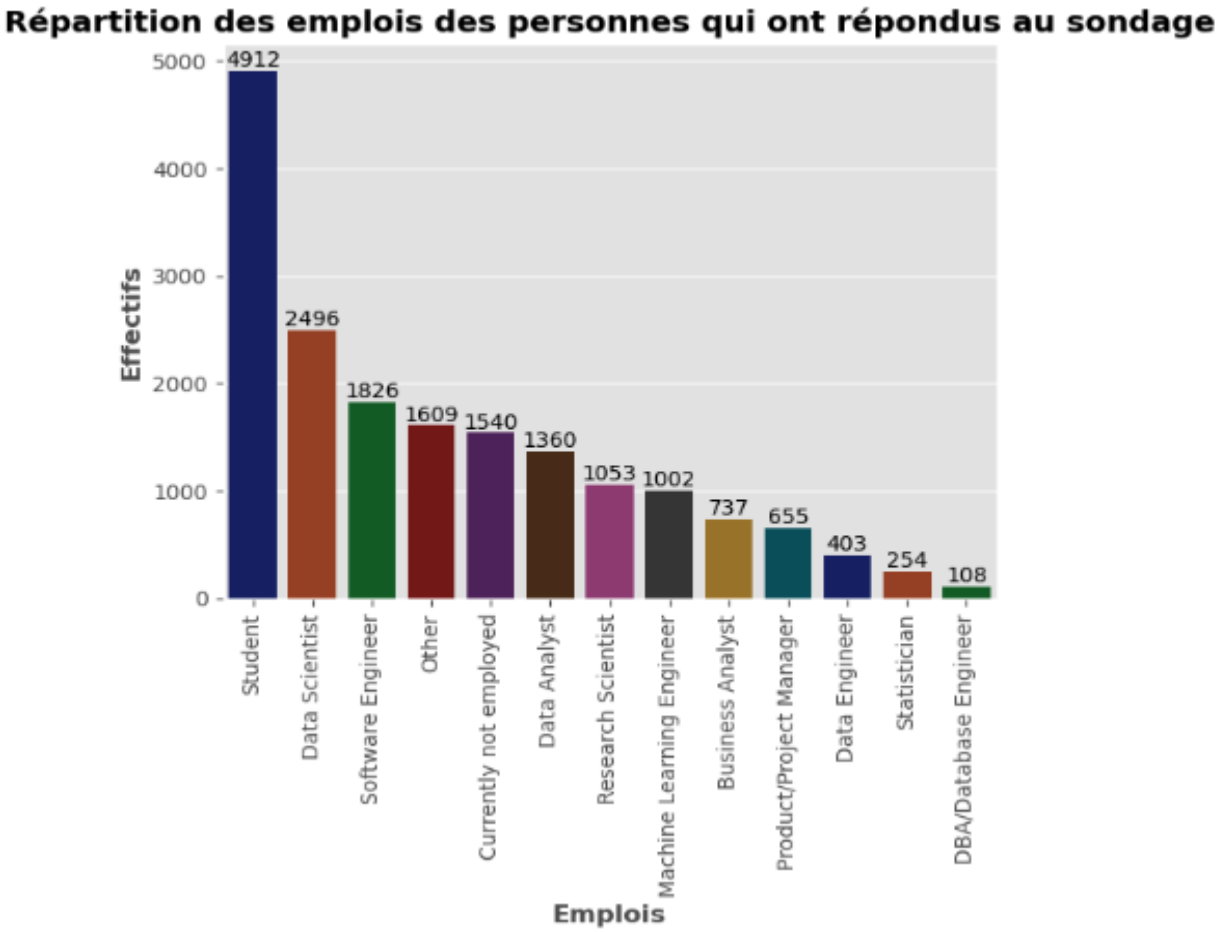
- Pays de résidence des répondants.
- Âges des répondants.
- Genres des répondants.
- Diplômes des répondants.
- Postes actuels des répondants.

Ci-après quelques graphiques pour illustrer cette première partie de l'analyse :

Pays de résidence des répondants de l'enquête :



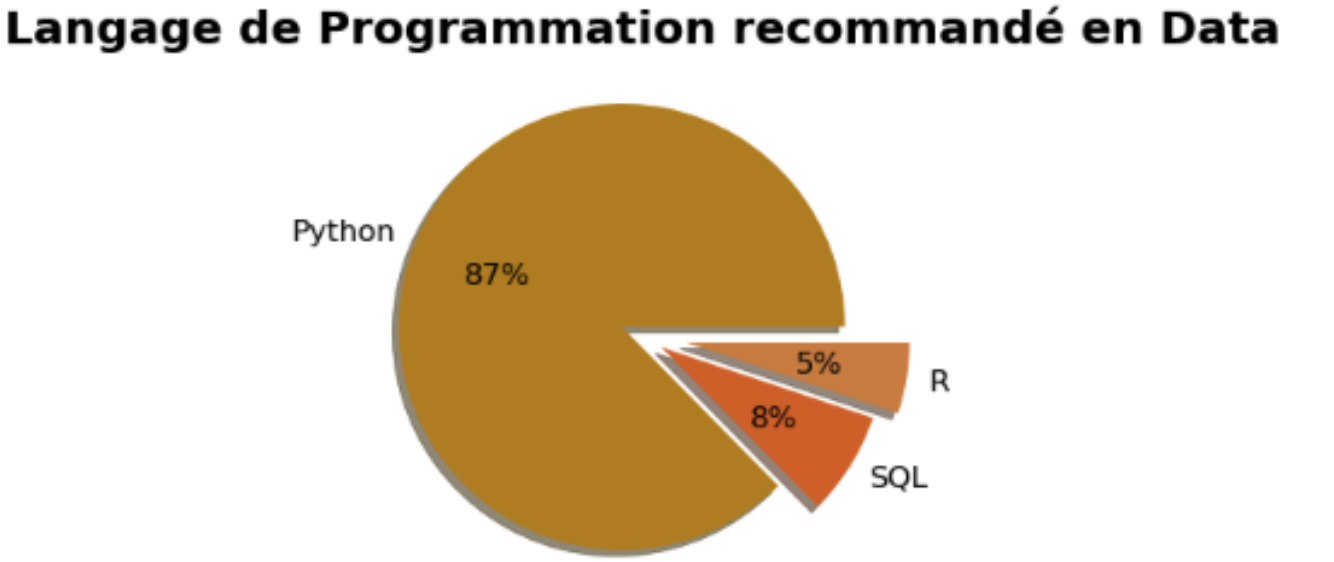
Emplois des répondants de l'enquête :

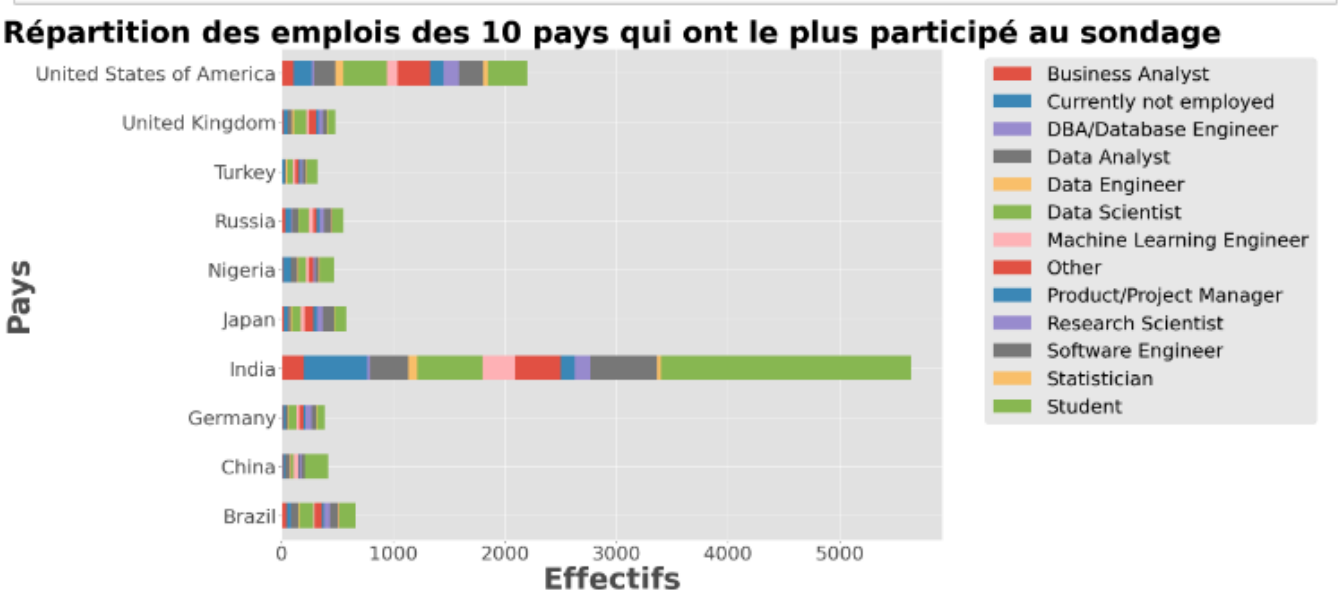


Nous nous sommes ensuite intéressés au 10 pays les plus représentés dans le jeu de données ainsi qu'aux 3 langages de programmation les plus présents :

- Top 10 pays qui ont répondu à l'enquête.
- Répartition des langages de programmations recommandés.
- Top 3 des langages de programmation.
- Répartition des diplômes du top 10 pays.
- Répartition des postes du top 10 pays.

Ci-après quelques graphiques pour illustrer cette première partie de l'analyse :

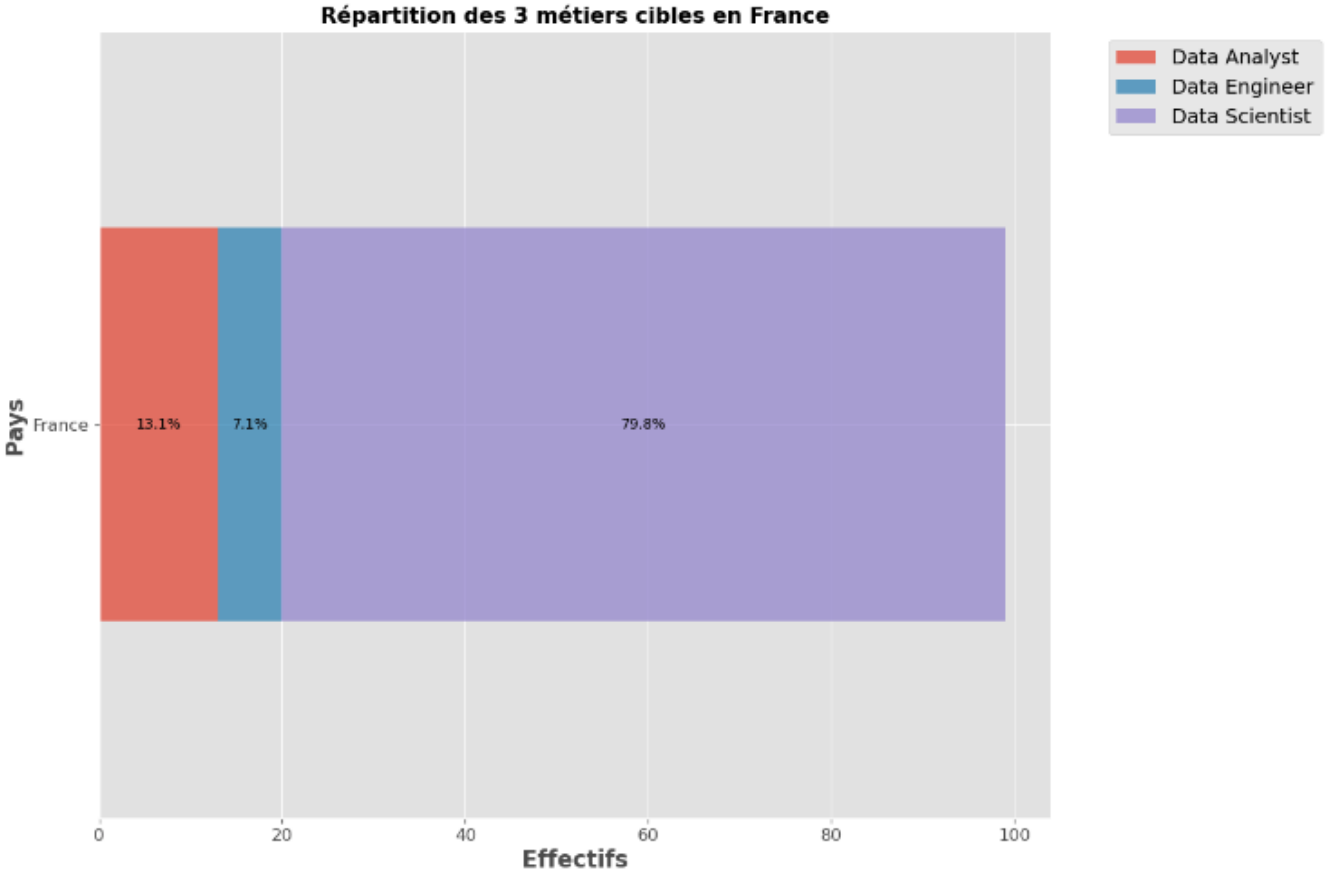


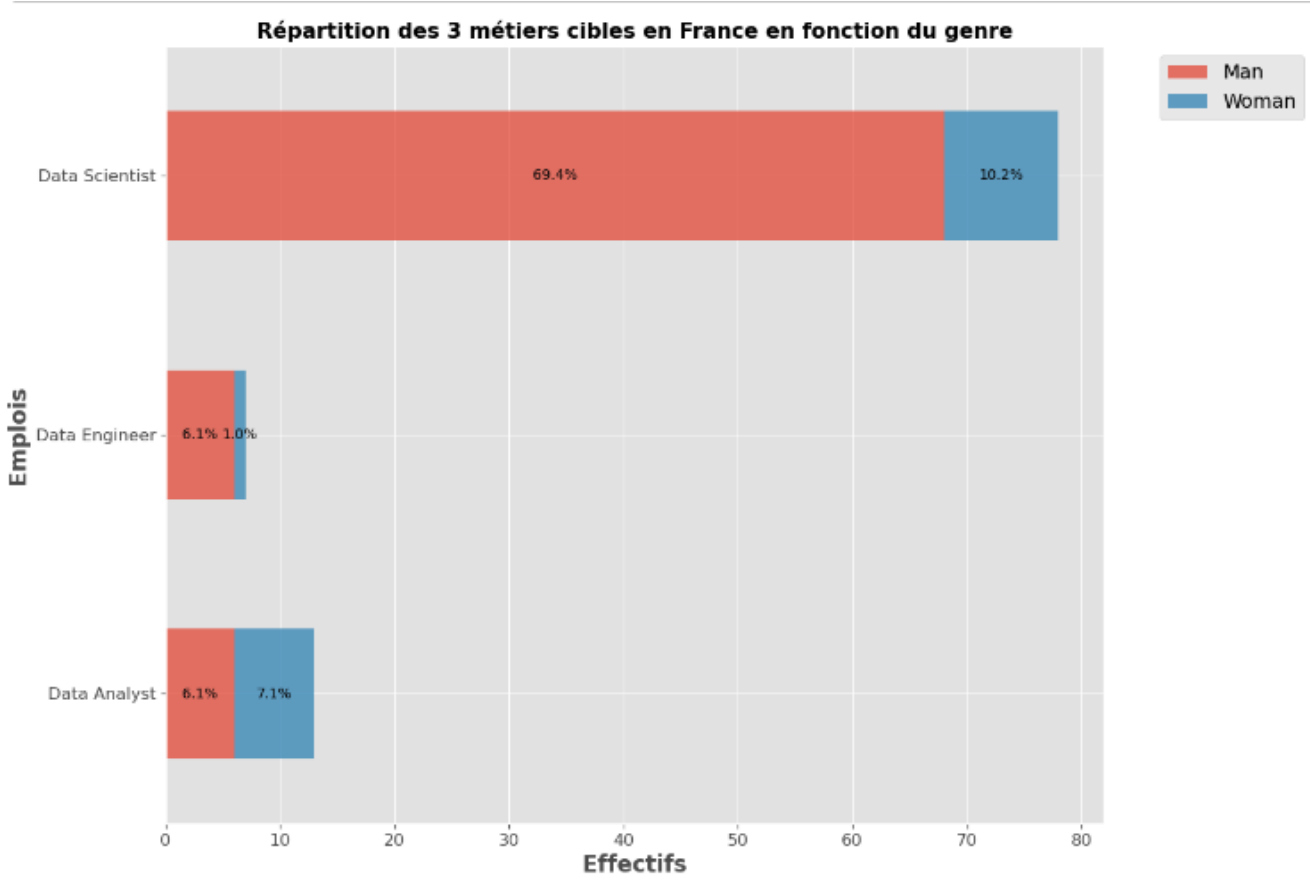


Et enfin nous nous sommes intéressés à trois métiers : Data Analyst, Data Engineer, Data Scientist :

- Répartition des diplômes en fonction des 3 métiers cibles.
- Répartition des 3 langages de programmation en fonction des 3 métiers cibles.
- Répartition des 3 métiers cible vs top 10 pays.
- Répartition des 3 métiers cibles en France.
- Répartition des 3 métiers cibles en fonction du Genre.
- Répartition des 3 métiers cibles en France en fonction du Genre.

Ci-après quelques graphiques pour illustrer cette première partie de l'analyse :





Nous avions décidé de focaliser notre analyse statistique sur deux critères principaux :

- Le niveau d'étude.
- Les langages de programmation.

Nous espérions ainsi, vérifier que ces critères seraient suffisants pour l'entraînement d'un modèle de machine learning afin d'être en mesure de prédire vers quel métier peut s'orienter un apprenant en fonction de son niveau d'étude et de ses compétences dans un langage de programmation.

Force est de constater que le langage de programmation **Python** et **le Master** sont surreprésentés au sein de nos 3 métiers cibles.

Conclusion :

L'étape de data-viz est très importante, car elle permet de valider ou d'invalider les premières hypothèses obtenues à la lecture du jeu de données. Cette étape n'est vraiment pas à négliger, car elle permet d'identifier la bonne direction à prendre soit au niveau business, soit pour tout projet de machine learning.

Néanmoins, il est important de faire attention aux biais statistiques et notamment au biais de sélection des données en s'assurant que l'échantillon de données qui est utilisé est bien le plus représentatif pour la réussite du projet d'analyse.

Cinquième étape : preprocessing et nettoyage des données.

L'étape de preprocessing et de nettoyage des données est l'étape charnière pour l'utilisation de modèle de machine learning.

Pour rappel notre jeu de données est issu d'une enquête de 36 questions qui ont été adressées aux personnes qui visitent le site kaggle.com et qui évoluent dans le monde de la data.

Sur les 36 questions qui le composent, 18 d'entre elles sont des questions à réponses uniques, les autres questions sont à choix multiples. Pour notre première itération de preprocessing et de machine learning, nous avons fait le choix de nous concentrer uniquement sur les questions à réponses uniques.

Nettoyage :

La première étape que nous avons réalisée a été de dupliquer le jeu de données dans un nouveau classeur Excel afin de pouvoir appliquer quelques filtres. Nous avons donc dans un premier temps réalisé les étapes suivantes :

- Suppression de toutes les colonnes qui correspondent à des questions à choix multiples.
- Filtrage des données de la colonne 'position' pour conserver uniquement nos 4 métiers cibles : Data Analyst, Data Scientist, Data Engineer et Machine Learning Engineer. Ces 4 métiers représentent nos données cibles.

Une fois cette première opération de tri et de nettoyage effectuée nous avons importé les données dans un nouveau Jupyter Notebook.

Nous avons dès lors suivi les étapes suivantes :

1. Utilisation d'un script python afin d'obtenir les informations pertinentes sur nos colonnes :
    a. Type de données.
    b. Pourcentage de valeurs manquantes.
    c. Nombre de valeurs uniques.
    d. Valeurs uniques.
    e. Mode ou la valeur moyenne de chaque colonne.
2. Nettoyage des colonnes les unes après les autres :
    a. Normalisation / simplification des données.
    b. Création de règles pour le traitement des valeurs manquantes : remplacer les NA par le mode ou la médiane.
3. Création de plusieurs dataframe pour stocker les données :
    a. Séparation des valeurs cibles dans un dataframe Target et des autres données dans un dataframe Features.
    b. Standardisation et centralisation des données du fataframe Features.

Preprocessing :

Une fois le nettoyage et la standardisation de données faites, nous avons pu passer à l'étape de preprocessing.

Nous avons décidé d'appliquer une étape **d'oversampling** : *RandomOverSampler* et *SMOTE* et **d'undersampling** : *RandomUnderSampler* et *Centroids* à nos données du fait que la répartition de nos valeurs cibles n'était pas homogène. En effet, comme nous avions pu le constater lors de l'étape de Data-Visualisation, les métiers Data Engineer et Machine Learning Engineer, sont sous représentés en comparaison des Data Analyst et Data Scientist. L'oversampling permet d'homogénéiser l'ensemble des valeurs cibles sur la valeur la plus haute. L'undersampling quant à lui est le même procédé, mais appliqué sur la valeur la plus basse.

Conclusion Nettoyage & Preprocessing :

Cette étape a été pour nous l'occasion de réfléchir à une stratégie à mettre en place pour l'utilisation de modèle de machine learning. Que ce soit par la sélection des colonnes les plus pertinentes de notre jeu de données ou par la gestion des valeurs manquantes (remplacement par la valeur la plus fréquente pour les valeurs catégorielles et par la médiane pour les valeurs numériques).
Il a été également nécessaire de standardiser et centrer les valeurs contenues dans le dataframe features_dm. Et enfin d'appliquer une étape d'over et d'undersampling à l'ensemble de nos jeux d'entraînements pour limiter au maximum tout risque d'erreur lors de l'utilisation de modèle de machine learning.

Sixième étape : première itération de modèles de machine learning.

L'objectif de notre projet fil rouge est de réussir à identifier les différents profils des personnes qui travaillent dans le milieu de la data, soit à réaliser une classification de ceux-ci. L'une des utilisations possibles du machine learning est la classification via les modèles : KNN, SVM, Decision Tree et Random Forest. C'est donc tout naturellement que ce sont ces modèles que nous avons mis à l'épreuve de nos jeux d'entraînements et de test.

Nous avons réalisé un premier tour des modèles sans utiliser d'hyper paramètre afin d'avoir un premier aperçu de la performance de ceux-ci.

Nous avons obtenu les résultats suivants :

Pour les jeux d'entraînement avec un Oversampling : RandomOverSampler

Score du Knn : 37%
Score du SVM : 43%
Score du DT : 41%
Score du RF : 48%

Nous avons pu remarquer que les résultats sont concentrés autour des 40% avec le RandomOverSampler.

Pour les jeux d'entraînement avec un Oversampling : SMOTE

Score du Knn : 40%
Score du SVM : 47%
Score du DT : 42%
Score du RF : 49%

Nous avons pu constater que les résultats sont un peu meilleurs avec SMOTE qu'avec le RandomOverSampler.

Pour les jeux d'entraînement avec un Undersamling : RandomUnderSampler

Score du Knn : 38%
Score du SVM : 43%
Score du DT : 33%
Score du RF : 41%

Nous avons pu relever que les résultats sont concentrés autour des 35/40% avec le RandomUnderSampler.

Pour les jeux d'entraînement avec un Undersamling : Centroids

Score du Knn : 28%
Score du SVM : 25%
Score du DT : 20%
Score du RF : 19%

Nous avons pu noter que les résultats sont concentrés autour des 20/25% avec le Centroids.

Premières conclusions :

Nous avons pu observer que les modèles choisis avec les paramètres par défaut n'offrent pas tous le même niveau de performance brut. Nous observons que deux modèles semblent sortir mieux que les autres, peu importe que les jeux d'entraînement aient été Over ou Undersamplé. Il s'agit des modèles SVM et Random Forest. Nous constatons également que la méthode SMOTE est celle qui a permis d'obtenir les résultats les plus élevés, peu importe le modèle de machine learning testé.

Définition des hyper paramètres :

Nous avons utilisé la méthode du GridSearchCV afin de pouvoir tester plusieurs hyper paramètres d'un coup sur l'ensemble des modèles de machine learning retenus.

Nous avons de plus ajouté la bibliothèque de son *chime* afin d'ajouter une alerte sonore dès lors qu'un modèle de machine learning a terminé de tourner. Cette opération pouvant prendre un certain temps, cela nous permettait de faire autre chose en attendant que le script python ait terminé ses opérations.

Une fois que chaque modèle a terminé son processus de calcul, nous avons obtenu les meilleurs scores en fonction des hyper paramètres les plus performants, ci-après les différents résultats sur nos jeux d'entraînement :

SCORE KNN :

RandomOverSampler : 69%
SMOTE : 66%
RandomUnderSampler : 39%
Centroids : 49%

SCORE SVM :

RandomOverSampler : 69%
SMOTE : 62%
RandomUnderSampler : 42%
Centroids : 56%

SCORE Decision Tree :

RandomOverSampler : 78%
SMOTE : 56%
RandomUnderSampler : 35%
Centroids : 49%

SCORE Random Forest :

RandomOverSampler : 80%
SMOTE : 68%
RandomUnderSampler : 43%
Centroids : 56%

Premières conclusions :

Nous avons pu constater ici que c'est la méthode de RandomOverSampler qui offre le meilleur score, peu importe le modèle de machine learning utilisé.
Deux modèles se dégagent nettement, il s'agit du *Decision Tree* avec un score de **78%** et du *Random Forest* avec un score de **80%**. Le KNN et le SVM quant à eux ne dépassent pas les 69% au mieux.

Une fois que nous avons identifié les meilleurs hyper paramètres, nous avons réalisé une matrice de confusion sur notre jeu de test afin de constater l'efficacité de chaque modèle de machine learning dans la classification de nos 4 métiers cibles (DATA Analyst, Data Scientist, Data Engineer et Machine Learning Engineer).

Matrice de Confusion :

KNN :

Classe prédite / Classe réelle	Data Analyst	Data Engineer	Data Scientist	Machine Learning Engineer
Data Analyst	179	22	82	32
Data Engineer	33	15	25	15
Data Scientist	162	52	248	74
Machine Learning Engineer	61	19	77	38

SVM :

Classe prédite / Classe réelle	Data Analyst	Data Engineer	Data Scientist	Machine Learning Engineer
Data Analyst	157	34	76	48
Data Engineer	32	12	24	20
Data Scientist	134	56	244	102
Machine Learning Engineer	51	28	57	59

Decision Tree :

Classe prédite / Classe réelle	Data Analyst	Data Engineer	Data Scientist	Machine Learning Engineer
Data Analyst	131	36	94	54
Data Engineer	28	13	31	16
Data Scientist	116	57	266	97
Machine Learning Engineer	31	15	93	56

Random Forest :

Classe prédite / Classe réelle	Data Analyst	Data Engineer	Data Scientist	Machine Learning Engineer
Data Analyst	156	23	95	41
Data Engineer	29	12	29	18
Data Scientist	109	34	324	69
Machine Learning Engineer	36	9	92	58

Conclusion :

La première chose que nous avons observée avec la création de ces matrices de confusion est que peu importe le modèle de machine learning, certains des métiers cibles ne sont pas correctement identifiés.

Les métiers : Data Engineer et Machine Learning Engineer, ne sont pas du tout classés correctement. Contrairement aux deux autres métiers qui sont dans une majorité des cas bien identifiés par les différents modèles.

Bien que les scores obtenus par les modèles Decision Tree et Random Forest semblaient significatifs, on constate qu'il reste encore une énorme marge de progression dans la classification des différents métiers.

Au moment de la réalisation de cette première itération de machine learning, nous avions posé les hypothèses suivantes concernant les raisons sur cette hétérogénéité dans la classification des différents métiers cibles :

- Les colonnes utilisées pour cette première itération ne proposaient pas suffisamment de données différenciantes pour chaque métier.
- Le jeu de données étant composé de plus de 300 colonnes, se limiter à seulement 18 ne permet peut-être pas aux modèles d'avoir suffisamment de données pour réussir une meilleure classification.
- Lors de l'étape de datavisualisation, nous avions déjà pu mettre à l'épreuve certains biais d'analyse que nous avions. Par exemple : le niveau d'étude que nous attendions de nos différents métiers cibles : les Data Scientist auraient majoritairement un niveau d'étude bac+7 (doctorat) ou que tous les Data Engineer auraient un niveau d'étude d'au moins bac +5 (master). Alors que la majorité des répondants à l'enquête Kaggle.com avaient en réalité un niveau bac+5 (master).

Nous avions pris la décision d'appliquer une stratégie de sélection des données pertinentes sur le modèle de l'entonnoir (du plus général au plus précis).

Nous allons donc pouvoir lors d'une seconde itération nous intéresser aux colonnes qui offrent des informations plus différenciantes et précises sur les tâches qu'effectuent (ou sont censés effectuer) nos 4 métiers cibles.

Septième étape : seconde itération de modèles de machine learning.

Une nouvelle approche :

À l'issue de la première itération de machine learning, nous avons obtenu des scores plutôt satisfaisants avec les modèles Decision Tree et Random Forest. Néanmoins, nous avons également constaté des faiblesses de la part de ces modèles dans la classification des 4 métiers cibles (Data Analyst, Data Scientist, Data Engineer et Machine Learning Engineer).

Lors de la première itération, nous nous étions concentrés sur les colonnes faisant référence aux questions à réponses uniques de l'enquête du site Kaggle.com. Afin d'améliorer la performance de nos modèles et notamment sur la partie classification des métiers Data Engineer et Machine Learning Engineer, nous avons décidé d'élargir significativement le nombre de critères.

Nous avons procédé comme suit :

1. Consultation de fiches métiers et d'offres d'emploi pour les Data Engineer et les Machine Learning Engineer.
2. Identification des tâches et activités principales de ces deux métiers cibles.
3. Nouvelle sélection de colonnes au sein de notre jeu de données afin d'avoir davantage de critères différenciants entre nos 4 métiers cibles.
4. Création d'un nouveau fichier CSV de travail pour la seconde itération de machine learning.

Le nouveau fichier sur lequel nous avons travaillé était constitué comme suit :

- 5670 lignes et 200 colonnes.
- Uniquement des colonnes contenant des valeurs catégorielles.

Nettoyage et Preprocessing :

Comme pour la première itération de machine learning, nous avons commencé par une étape de nettoyage des données et de preprocessing.

Peu de chose à décrire ici, si ce n'est le choix de remplacer l'ensemble des valeurs manquantes par "no_reply" et la simplification des réponses qui avaient été faites sous forme de phrase ou contenant des caractères spéciaux (parenthèses par exemple).

Le preprocessing a consisté en l'utilisation de la fonction get_dummies puis d'une nouvelle étape d'over et d'undersampling de nos jeux d'entraînement et de test.

Premier résultat :

Lors de la première itération, nous avions utilisé 4 modèles de machine learning différents (KNN, SVM, Decision Tree & Random Forest). Ce sont les modèles Random Forest et Decision Tree qui ont été les plus performants, nous avons donc fait le choix de nous concentrer uniquement sur ces ceux-ci pour cette seconde itération de machine learning.

Là encore, nous avons réalisé un premier tour des modèles Decision Tree (DT) et Random Forest (RF) sur nos jeux d'entraînements avec les paramètres par défauts. Ce qui nous a permis d'obtenir les résultats suivants :

En rouge les scores de la première itération et en bleu ceux de la seconde itération :

Pour les jeux d'entraînement avec un Oversampling : RandomOverSampler

Score du DT : 47%
Score du RF : 59%
Score du DT : 41%
Score du RF : 48%

Nous observons une augmentation d'environ 10% du score de performance de ces deux modèles en comparaison des scores de la première itération.

Pour les jeux d'entraînement avec un Oversampling : SMOTE

Score du DT : 48%
Score du RF : 60%
Score du DT : 42%
Score du RF : 49%

Nous constatons ici aussi une augmentation d'environ 10% entre les deux itérations, nous notons également des scores proches entre les deux méthodes d'oversampling.

Pour les jeux d'entraînement avec un Undersamling : RandomUnderSampler

Score du DT : 38%
Score du RF : 52%
Score du DT : 33%
Score du RF : 41%

Nous remarquons une progression intéressante de la part du modèle Random Forest qui dépasse les 50% lors de cette deuxième itération.

Pour les jeux d'entraînement avec un Undersamling : Centroids

Score du DT : 23%
Score du RF : 23%
Score du DT : 20%
Score du RF : 19%

La méthode d'undersampling Centroids et celle qui voit le score de performance des deux modèles progresser le moins uniquement un gain de 3 et 4% entre les deux itérations.

Premières conclusions :

Dès le premier tour de machine learning sur ces nouveaux jeux d'entraînements, nous constatons une augmentation de 10% pour les deux modèles retenus. À première vue, il semble que l'ajout de critères permet à nos modèles de machine learning de mieux classer nos métiers cibles.

A ce stade de notre analyse, nous espérions obtenir des résultats significatifs pour la classification des métiers Data Engineer et Machine Learning Engineer.

Définition des hyper paramètres :

Nous avons une nouvelle fois utilisé la fonction GridSearchCV pour la définition des hyper paramètres des modèles Decision Tree et Random Forest, là encore les scores en rouge représentent ceux de la première itération et ceux en bleu ceux de la seconde itération.

SCORE Decision Tree :

RandomOverSampler : 81%
SMOTE : 64%
RandomUnderSampler : 39%
Centroids : 50%

RandomOverSampler : 78%
SMOTE : 56%
RandomUnderSampler : 35%
Centroids : 49%

SCORE Random Forest :

RandomOverSampler : 84%
SMOTE : 76%
RandomUnderSampler : 50%
Centroids : 61%

RandomOverSampler : 80%
SMOTE : 68%
RandomUnderSampler : 43%
Centroids : 56%

Premières conclusions :

Nous constatons que malgré l'ajout de nombreux critères supplémentaires, les scores n'ont que peu évolué entre la première et la seconde itération une fois les hyper paramètres définis. Il y a eu en moyenne **4%** d'amélioration pour le Decision Tree et **6%** d'amélioration pour le Random Forest.

L'utilisation d'une matrice de confusion sur les meilleurs hyper paramètres pour ces deux modèles nous permettra de définitivement constater s'il y a une amélioration ou non dans la classification de nos 4 métiers cibles.

Matrice de Confusion :

Decision Tree :

Première itération :

Classe prédite / Classe réelle	Data Analyst	Data Engineer	Data Scientist	Machine Learning Engineer
Data Analyst	131	36	94	54
Data Engineer	28	13	31	16
Data Scientist	116	57	266	97
Machine Learning Engineer	31	15	93	56

Seconde itération :

Classe prédite / Classe réelle	Data Analyst	Data Engineer	Data Scientist	Machine Learning Engineer
Data Analyst	140	24	81	26
Data Engineer	30	21	27	5
Data Scientist	110	51	316	83
Machine Learning Engineer	38	18	82	82

Random Forest :

Première itération

Classe prédite / Classe réelle	Data Analyst	Data Engineer	Data Scientist	Machine Learning Engineer
Data Analyst	156	23	95	41
Data Engineer	29	12	29	18
Data Scientist	109	34	324	69
Machine Learning Engineer	36	9	92	58

Seconde itération :

Classe prédite / Classe réelle	Data Analyst	Data Engineer	Data Scientist	Machine Learning Engineer
Data Analyst	153	11	88	19
Data Engineer	31	10	36	6
Data Scientist	86	8	405	61
Machine Learning Engineer	28	2	96	94

Conclusion Matrice de Confusion :

Que ce soit pour le modèle Decision Tree ou Random Forest, nous constatons une amélioration de la classification des métiers Data Analyst et Data Scientist. L'ajout des critères supplémentaires a permis aux modèles d'affiner la classification de ceux-ci.

Concernant les deux autres métiers, il n'y a quasiment pas d'amélioration pour le modèle Random Forest pour la classification des Data Engineer. Et il y a une petite amélioration de la classification des Data Engineer et Machine Learning Engineer pour le modèle Decision Tree.

Lors de la sélection des critères supplémentaires, nous avons constaté que les répondants de nos 4 métiers cibles partagent des compétences et utilisations de logiciels similaires, ce qui empêche de facto les modèles de machine learning d'être en mesure de classifier avec précision les différents postes.

Conclusion Seconde Itération de Machine Learning :

Malgré une progression de 10% du score de performance pour les deux modèles avec les paramètres par défaut, nous n'avons obtenu qu'une amélioration d'en moyenne 4 et 6% des scores de performances pour le Decision Tree et la Random Forest.

L'ajout de nombreux critères supplémentaires n'a pas non plus permis d'améliorer significativement la classification des Data Engineer et des Machine Learning Engineer. Cette seconde itération nous permet néanmoins de poser les hypothèses suivantes pour expliquer ces résultats :

- La raison la plus probable quant à ce problème de classification vient de la nature de nos données. Pour rappel, il s'agit des réponses à une enquête réalisée par le site Kaggle.com sur les compétences et logiciels utilisés par les personnes qui travaillent dans la data. Il est possible que les questions posées soient trop générales et ne permettent pas de différencier suffisamment les répondants.
- Il est également possible que les personnes qui ont participé à l'enquête occupent des postes sur lesquels ils réalisent aussi bien des tâches propres aux : Data Analyst, Data Scientist, Data Engineer et Machine Learning Engineer.
- Une autre possibilité est également un *enthousiasme* trop prononcé dans les réponses aux questionnaires, c'est-à-dire, indiquer l'utilisation de tels ou tels logiciels ou compétences, car ils en ont connaissance et/ou l'ont déjà utilisé, sans que cela ne reflète strictement leurs activités journalières.
- Enfin, il est bon de rappeler que Kaggle.com est une plateforme d'apprentissage et que les personnes qui visitent ce site le font avant tout pour se former. De ce fait, ce n'est pas surprenant d'avoir des Data Analyst, Data Scientist, Data Engineer et Machine Learning Engineer qui partagent des compétences et connaissances similaires.

Ces hypothèses nous amènent à penser que nous avons atteint un niveau de performance satisfaisant avec nos modèles de machine learning au regard des données à notre disposition. Pour améliorer encore plus la classification, il faudrait envisager l'utilisation d'un nouveau jeu de données, qui cette fois-ci, s'appuierait par exemple sur des données provenant directement d'offres d'emploi propre à chacun des métiers cibles que nous avons étudiées.

Conclusion Projet Fil Rouge :

A travers les différentes étapes du projet, nous avons pu mettre à l'épreuve notre première impression du jeu de données :

A savoir que nous disposions de suffisamment de données pour être en mesure de dresser un profil clair et précis pour chacun de nos métiers cibles. Profil qui nous aurait ainsi permis d'obtenir une classification homogène de ces métiers par nos modèles de machine learning.

La partie datavisualisation nous a permis de construire une cartographie des répondants et d'observer que même si les métiers de Data Analyst, Data Scientist, Data Engineer et Machine Learning Engineer sont différents, ils partagent tous un tronc commun de compétences et d'utilisation de logiciels.

Il nous est apparu clair assez rapidement, qu'une classification des 4 métiers ne serait probablement pas possible en utilisant uniquement quelques critères de différenciations. Hypothèse que nous avons pu confirmer lors de la première itération de machine learning.

La seconde itération quant à elle nous a amené à augmenter drastiquement le nombre de critères, en passant d'une dizaine de critères pour la première itération à deux cents pour la seconde. Bien qu'une augmentation du pourcentage de performance du modèle ait été observée avec l'ajout de ces nombreux critères, la classification des métiers cibles et notamment des Data Engineer et Machine Learning Engineer, n'est toujours pas satisfaisante.

Cet état peut s'expliquer par de nombreux facteurs, que nous avons détaillé dans l'analyse de la seconde itération. L'un des principaux facteurs est très probablement, le fait que les questions posées dans l'enquête ne sont pas suffisamment différenciantes pour permettre une classification précise des métiers des répondants.

La piste la plus pertinente à explorer pour aller plus loin dans ce travail d'analyse et de classification, serait la consolidation des données par l'ajout de nouveaux critères. Que ce soit par une évolution du questionnaire du site Kaggle.com qui s'appuierait cette fois-ci directement sur des fiches de postes des différents métiers de la data pour construire son enquête. Ou directement par le *scrapping* d'offre d'emploi des métiers de la data pour être en mesure de construire le profil de ces métiers le plus précis possible. Il serait ainsi possible de construire un modèle qui pourrait, au travers des critères retenus proposer une classification la plus précise possible.

Appréciation du projet fil rouge

Ce projet fil rouge a été pour nous l'occasion d'avoir un aperçu complet du métier de Data Analyst, que ce soit pour l'étape de prise de connaissance des données, la réalisation des premières analyses et conclusion. Nous avons pu également nous familiariser avec l'aspect présentation et synthèse via la création de visuels d'analyse de notre jeu de données.

Et nous avons enfin pu nous expérimenter au machine learning. Bien que cet aspect ne fasse pas partie des tâches principales du Data Analyst, il est important que nous en ayons une bonne connaissance, afin d'être en mesure de comprendre et de partager les conclusions obtenues.

Ce projet nous a permis aussi d'avoir un autre regard sur les "Données" et de dépasser une certaine "appréhension" face à des fichiers de plusieurs (centaines de) milliers de lignes.

Datascientest et ce projet fil rouge sont le premier pas vers de plus en plus de projets stimulants et intéressants dans la Data !

Remerciements :

Nous tenons à remercier :

- Datascientest (et ses équipes) pour la qualité des supports de formation et la disponibilité pour répondre à nos interrogations.
- Romain et Robin nos supers mentors projet qui ont su nous épauler et nous accompagner tout au long de la réalisation de ce projet.