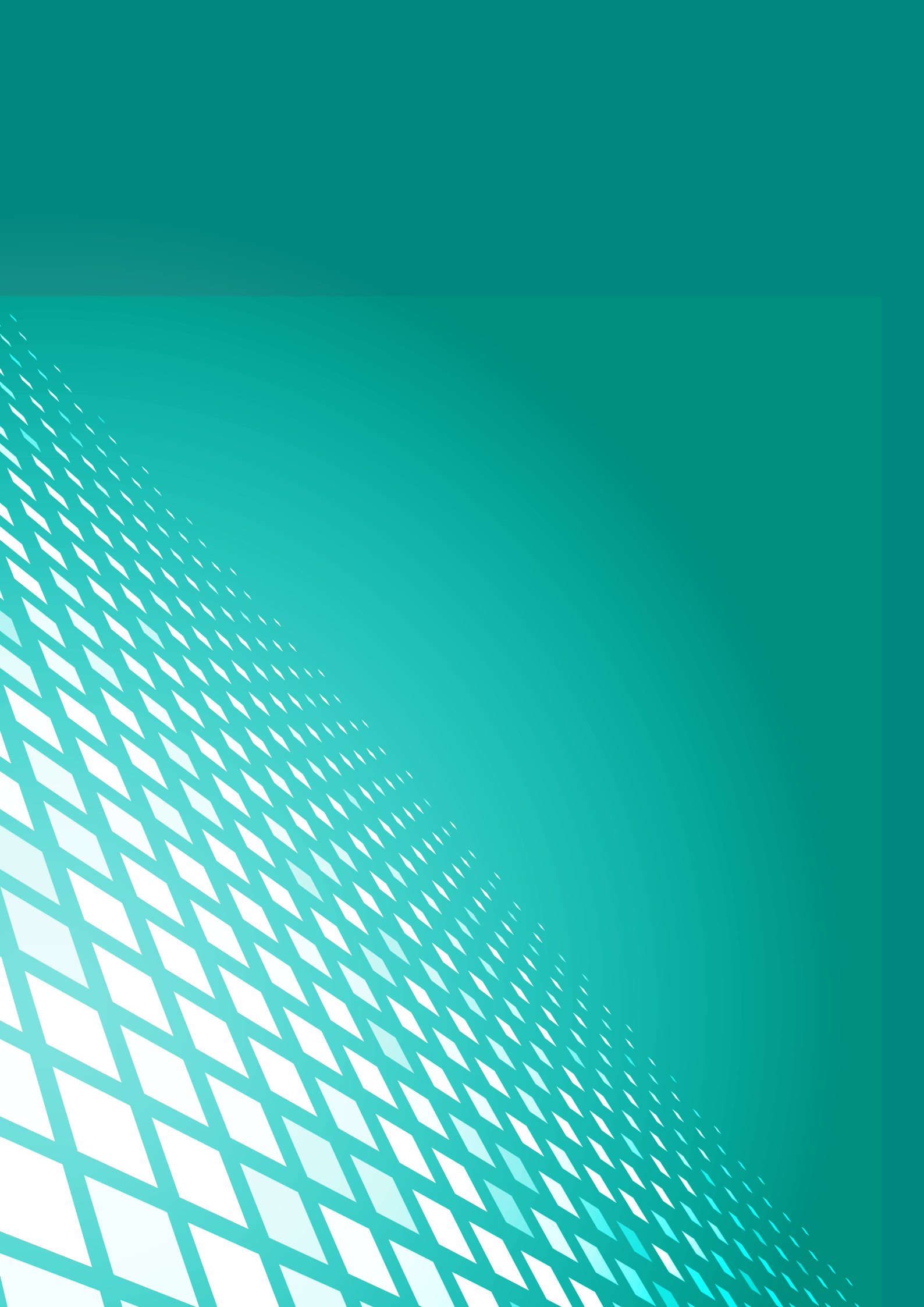
GUIDE DE BONNES PRATIQUES POUR LES SCIENTIFIQUES DES DONNÉES DU SECTEUR PUBLIC

LE GOUVERNEMENT
DU GRAND-DUCHÉ DE LUXEMBOURG
Ministère de la Digitalisation

LE GOUVERNEMENT
DU GRAND-DUCHÉ DE LUXEMBOURG
Ministère de la Recherche
et de l'Enseignement supérieur



Préambule

Dans le paysage en constante évolution de la science des données, il est primordial de naviguer dans les complexités et considérations éthiques inhérentes à l'exploitation de connaissances guidées par les données. Alors que le volume et la vitesse des données continuent d'augmenter, il devient impératif d'établir un cadre solide qui délimite les principes, les objectifs et les buts guidant nos efforts dans ce domaine et de s'assurer que toute utilisation se fera dans le respect de l'éthique et de la loi.

Ce document est conçu pour répondre aux besoins uniques des scientifiques des données (*data scientists*) travaillant dans le secteur public au Grand-Duché de Luxembourg et dans l'écosystème législatif et technique actuel. Son objectif principal est de fournir un cadre clair et complet qui guide les professionnels à travers les défis techniques, éthiques et sociétaux inhérent à leur travail.

En promouvant les principes d'intégrité, de reproductibilité et de maintenabilité, ces recommandations visent à permettre aux *data scientists* du secteur public d'exploiter tout le potentiel de leur discipline.

Grâce à un engagement en faveur de l'apprentissage continu, de la collaboration de de l'adhésion aux meilleures pratiques, nous nous efforçons de cultiver une communauté scientifique qui se consacre à l'avancement du bien public par le biais d'une prise de décision éthique et responsable fondée sur les données.

Ce document fournit à la fois un glossaire basé sur des mots-clés pour trouver rapidement des informations spécifiques et des idées plus approfondies sur tous les aspects d'une science des données réussie.

La section 1 présente quelques définitions importantes pour définir le champ d'application. La section 2 présente les objectifs généraux que les *data scientists* devraient s'efforcer d'atteindre, et la section 3 poursuit avec des pratiques et lignes directrices plus spécifiques. Enfin, les sections 4 et 5 sont davantage axées sur les objectifs à moyen et long terme et présentent le contrôle et l'amélioration continus, ainsi que la gestion du cycle de vie.

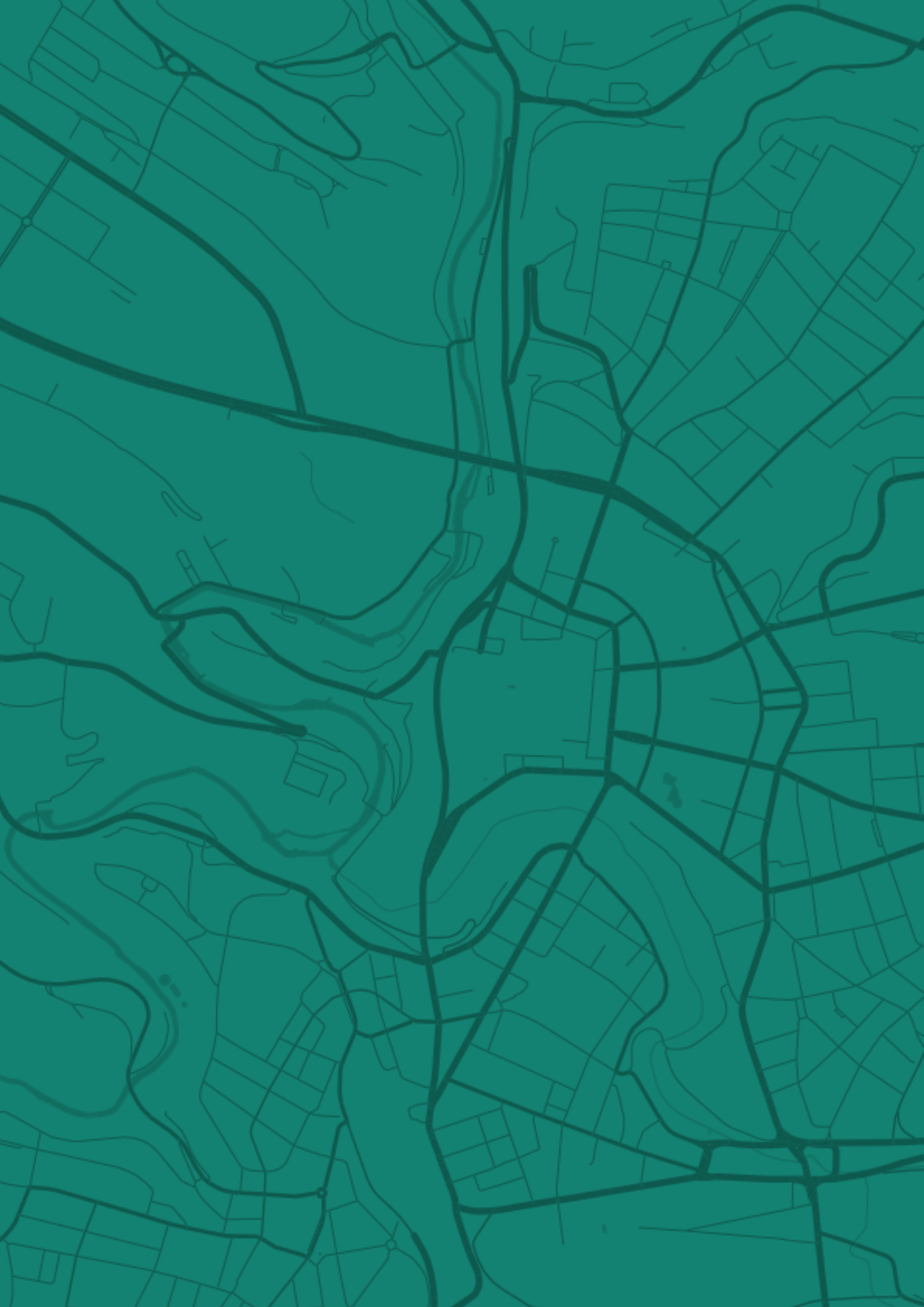


Table des matières

1. Définitions

Lors de leurs tâches quotidiennes, les scientifiques des données sont confrontés à un large éventail de concepts différents, qui font l'objet de la présente section. Leur expertise de base concerne des domaines tels que les statistiques, les mathématiques, le *machine learning*, et le développement informatique, tandis qu'ils sont également confrontés à des préoccupations telles que l'éthique, le droit, l'ingénierie logicielle et la gestion de projet. Plus concrètement, les *data scientists* mettent en œuvre des pipelines de données pour manipuler les données afin d'en extraire des informations, des rapports et des informations visuelles. En outre, si leur travail inclut des outils de *machine learning*, ils peuvent également utiliser des modèles prêts à l'emploi ou fournir leurs propres modèles réentraînés ou raffinés

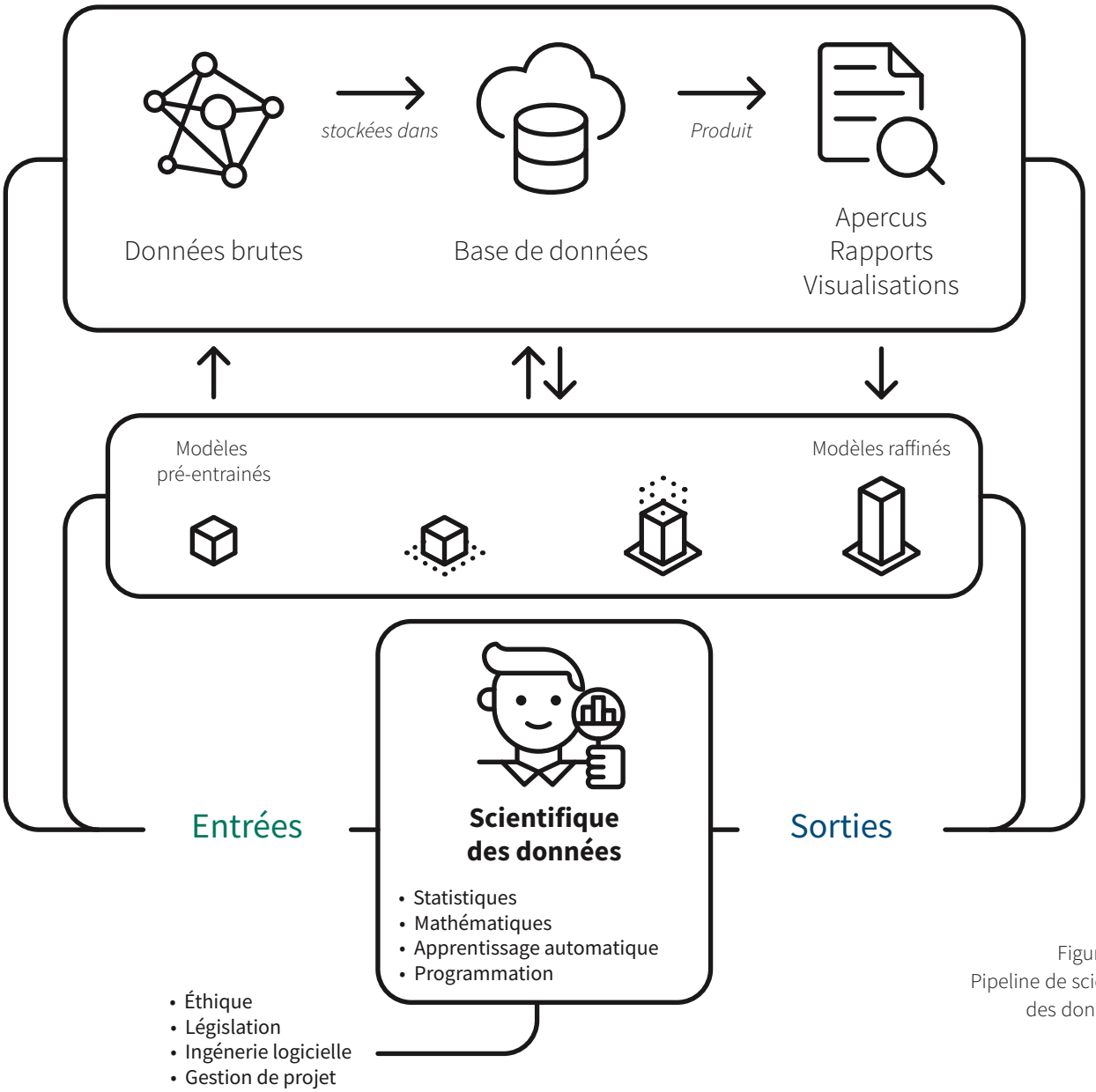


Figure 1 :
Pipeline de science
des données

- **Données :** Une donnée représente une information, une statistique ou un fait collecté et stocké à des fins de référence, d'analyse ou d'utilisation future. Elles peuvent exister sous différentes formes, telles que du texte, des nombres, des images ou du multimédia, et servent de base à l'analyse et à la prise de décision dans divers domaines. Dans certains cas, le code informatique peut également être considéré comme une donnée.
- **Ensemble de données / jeu de données (*dataset*) :** Un *dataset* est une collection structurée de données, ou accompagnées de métadonnées permettant de les structurer, ou d'informations organisés dans un format spécifique, généralement stockée dans des bases de données ou des fichiers. Il sert d'unité cohérente pour l'analyse, la recherche ou la résolution de problèmes, et contient de multiples observations ou instances ainsi que leurs attributs ou variables associés.
- **Métadonnée :** Une métadonnée est une information supplémentaire qui fournit un contexte, une structure et des détails sur les données ou *datasets* et contribue à en faciliter l'organisation, la gestion et la compréhension.
- **Analyse des données :** L'analyse des données implique le processus d'examen, de nettoyage, de transformation et d'interprétation des données afin d'en extraire des informations précieuses, des motifs ou des tendances. Elle utilise divers outils, techniques et méthodes statistiques pour découvrir des informations significatives qui peuvent aider à la prise de décision et à la formulation de stratégies.
- **Science des données (*data science*) :** La science des données est un domaine interdisciplinaire qui englobe les méthodologies, les algorithmes et les outils permettant d'extraire des informations ou des connaissances à partir de données. Elle fait appel à l'analyse statistique, au *machine learning*, à la programmation et à l'expertise dans le domaine pour résoudre des problèmes complexes, développer des modèles prédictifs et tirer des enseignements exploitables des ensembles de données.
- **Scientifique des données (*data scientist*) :** Un *data scientist* est un professionnel compétent dans le traitement et l'analyse de grands volumes de données à l'aide d'analyses statistiques, de programmation et de connaissances dans le métier. Il met à profit son expertise dans les techniques de science des données pour explorer les données, développer des modèles et interpréter les résultats afin de résoudre les problèmes de l'entreprise ou d'éclairer les processus décisionnels.
- **Pipeline de la science des données :** Un processus systématique et automatisé qui couvre la collecte, l'exploration, le traitement, l'analyse et le déploiement des données pour résoudre les problèmes spécifiques à la science des données. Il orchestre une séquence d'étapes connectées pour rationaliser et automatiser ces tâches, fournissant ainsi aux projets de science des données des flux de travail cohérents et reproductibles.

- **Modèle :** Dans le contexte de la science des données, un modèle fait référence à une représentation d'un système ou d'un phénomène du monde réel. Il constitue un résultat essentiel d'un pipeline de science des données et souvent la principale motivation pour démarrer un projet. Les modèles peuvent également exister dans le commerce et servir d'entrée ou de moyen de transformation des données. La conception et l'évolution des modèles au cours d'un projet de science des données sont étroitement liées aux observations faites pendant l'analyse des données. En général, on commence par un modèle de base que l'on affine ou que l'on perfectionne par itérations pour créer la version finale.
- **Génie logiciel :** Le génie logiciel est une discipline de l'informatique qui se concentre sur la conception, le développement, le test et la maintenance systématiques des systèmes logiciels. Certains des principes du génie logiciel s'appliquent aussi directement à la science des données (par exemple, les meilleures pratiques de programmation). Cependant, le génie logiciel, en tant que discipline, devrait plutôt être considéré comme quelque chose qui existe en dehors de la science des données, mais qui devra être pris en compte pour les projets qui vont au-delà de l'analyse des données et qui doivent fournir une suite logicielle fonctionnelle en tant que résultat.

1.1. Types d'analyse de données

En général, un *data scientist* peut effectuer six types d'analyse :

- **Analyse descriptive :** L'analyse descriptive consiste à décrire un événement passé à l'aide de techniques appelées statistiques descriptives. Elle permet de résumer et de présenter les données afin de mieux comprendre les modèles et les tendances. En règle générale, les analyses descriptives permettent de répondre à des questions telles que « Combien d'amendes pour conduite en état d'ivresse ont été infligées chaque jour au cours de l'année écoulée ? » Il existe plusieurs façons de répondre à cette question, par exemple en fournissant une moyenne journalière des amendes sur l'ensemble de la période, en établissant une moyenne pour les jours de semaine et les week-ends séparément, ou en présentant un graphique illustrant le total journalier sur la période. Il n'y a pas nécessairement une seule réponse correcte, et la réponse dépendra du type de données analysées, des aspects à mettre en évidence (par exemple, si l'accent doit être mis sur une augmentation des amendes pendant les week-ends ou même sur les fluctuations saisonnières, un graphique pourrait être plus approprié), et de la personne qui pose la question.
- **Analyse diagnostique :** Ce type d'analyse implique l'examen des données pour comprendre pourquoi certains événements se sont produits ou quels facteurs ont influencé ces événements. Il permet d'identifier les corrélations et les liens de causalité.
- **L'analyse en temps réel** implique le traitement et l'analyse des données au fur et à mesure qu'elles arrivent, ce qui permet d'obtenir des informations instantanées. Cette approche permet de prendre des décisions en temps réel et de réagir rapidement à de nouvelles conditions ou à de nouveaux événements. Un des principaux cas d'utilisation de l'analyse en temps réel est la nécessité de générer des alertes dans des conditions prédéfinies ou pré-identifiées, qui peuvent ensuite être intégrées dans des systèmes d'alerte existants ou être développées dans des tableaux de bord dédiés en temps réel.

- **L'analyse prédictive :** En reprenant l'exemple précédent, l'analyse prédictive répondrait à une question telle que « À combien d'amendes peut-on s'attendre au cours de la semaine du 15 août ? L'approche consiste généralement à modéliser des séries temporelles (*timeseries*) sur la base de données antérieures, ce qui permet de faire des prédictions sur la base de ce modèle. Cependant, il est essentiel d'être prudent : l'analyse prédictive ne peut pas répondre à une question similaire telle que « À combien d'amendes peut-on s'attendre pendant la semaine du 15 août si nous installons un radar au carrefour ABC ? » L'analyse prédictive s'appuie souvent sur des données de *timeseries*. Des méthodes appropriées tenant compte des spécificités des timeseries (fixité, saisonnalité, cointégration, etc.) doivent être employées.
- **L'analyse prescriptive :** Alors que l'analyse prédictive recommande des actions ou des stratégies basées sur des modèles prédictifs afin d'optimiser les résultats, l'analyse prescriptive fournit des conseils sur les actions à entreprendre pour atteindre le résultat souhaité. L'analyse prescriptive utilise des techniques d'optimisation et de simulation pour suggérer le meilleur plan d'action sur la base des prédictions faites par les modèles prédictifs. Exemple : Recommandation de stratégies de marketing basées sur des modèles prédictifs pour maximiser les ventes.
- **L'analyse causale :** L'analyse causale vise à déterminer l'impact d'une intervention. Il n'existe pas de méthodologie unique, mais en général, il est essentiel de déterminer ce que l'on appelle dans le jargon une « stratégie d'identification. » Idéalement, cette stratégie implique une expérimentation. Cependant, des limitations pratiques, telles que le coût, l'éthique ou la faisabilité, peuvent empêcher sa mise en œuvre. En l'absence d'expérimentation, les approches statistiques permettent d'approcher les conditions expérimentales idéales pour déterminer l'impact causal d'une intervention. Il est fortement recommandé de dessiner un graphe orienté acyclique pour déterminer les variables à prendre en compte ou à écarter dans la stratégie d'identification.

Alors que l'analyse descriptive se concentre sur le résumé de données historiques, l'analyse prédictive prévoit des événements futurs, l'analyse prescriptive suggère des actions basées sur ces prévisions, et l'analyse causale cherche à comprendre les relations entre les causes et les effets dans les données. Chaque type d'analyse sert un objectif différent et aide à la prise de décision dans divers domaines, de l'entreprise à la recherche scientifique.

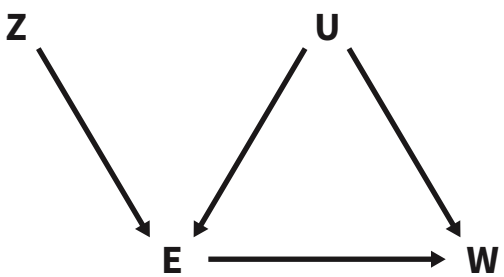


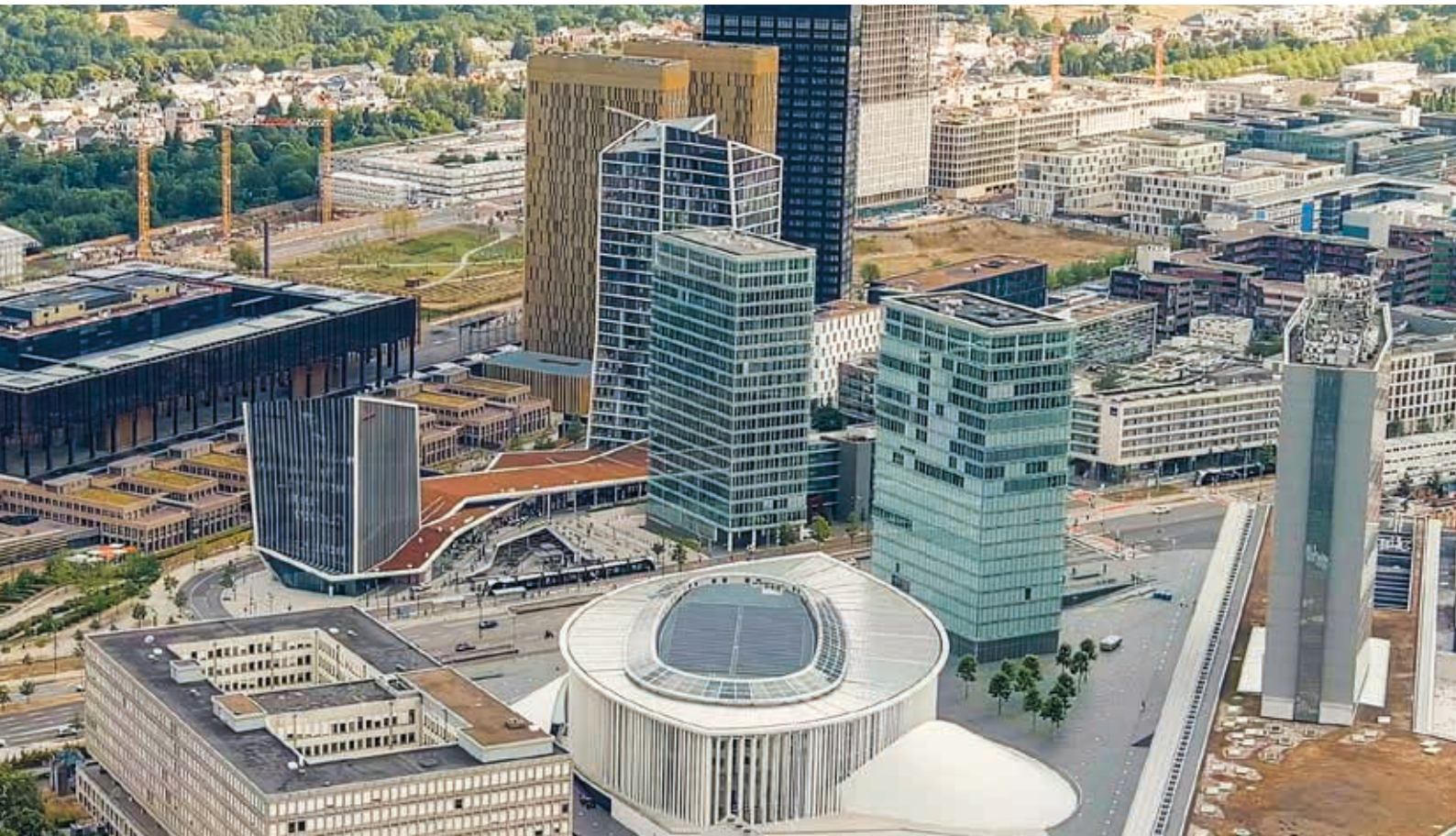
Figure 2 : Un graphe orienté acyclique illustrant l'approche par « variables instrumentales » pour l'inférence causale. Source: Statistical rethinking with brms, ggplot2 and the tidyverse [1]

1.2. Méthodes d'ingénierie des logiciels et des données

Diverses méthodes existent pour promouvoir la collaboration dans les projets de science des données et soutenir leur déploiement et leur intégration.

- **Intégration continue (CI) :** L'intégration continue est une pratique de développement dans laquelle les développeurs fusionnent régulièrement leurs modifications de code dans un référentiel partagé. L'objectif est de détecter rapidement les problèmes d'intégration en automatisant la construction et l'exécution des tests, afin de garantir que les nouveaux ajouts de code s'intègrent de manière transparente dans la base de code existante.
- **Livraison continue (CD) :** La livraison continue étend le CI en déployant automatiquement les modifications du code dans des environnements de production ou de test après avoir passé la phase de test automatisée. Elle garantit que le logiciel peut être mis à disposition de manière fiable à tout moment, en réduisant l'intervention manuelle nécessaire au déploiement.
- **Conteneur (OCI - *Open Container Initiative*) :** Les conteneurs sont une forme d'empaquetage logiciel léger, portable et exécutable qui encapsule une application et ses dépendances. L'OCI est une spécification standard pour les formats de conteneurs et leurs moteurs d'exécution, qui garantit la compatibilité et l'interopérabilité entre les différentes plateformes de conteneurs.
- **DevOps :** DevOps est une approche culturelle et technique qui vise à rapprocher les équipes de développement (Dev) et d'exploitation (Ops). Elle met l'accent sur la collaboration, l'automatisation et le partage des responsabilités tout au long du cycle de développement des logiciels afin de fournir des logiciels de haute qualité plus rapidement et plus efficacement.

- **DataOps :** DataOps est une méthodologie qui applique les principes DevOps à l'ingénierie et à la gestion des données. Elle se concentre sur la rationalisation et l'automatisation des flux de données, en assurant la collaboration entre les équipes de données, et en améliorant l'agilité et la qualité des applications basées sur les données.
- **MLOps :** MLOps est un ensemble de pratiques qui combinent le ML (pour *Machine Learning*) et DevOps pour améliorer la gestion du cycle de vie des modèles ML. Il s'agit d'automatiser le déploiement, la surveillance et la gestion des modèles ML à l'échelle, en garantissant leur fiabilité, leur reproductibilité et leur amélioration continue.
- **Tests logiciels :** Les processus systématiques et automatisés qui vérifient la fonctionnalité, détectent les défauts et garantissent le respect des exigences d'un projet :
  - **Tests unitaires :** Processus consistant à tester des unités ou des composants individuels, de petite taille et isolés, afin de s'assurer de leur exactitude et de leur fonctionnalité, ce qui permet de détecter les erreurs à un stade précoce et à un niveau peu élevé.
  - **Tests fonctionnels :** Évaluation des fonctionnalités d'un projet par rapport à des spécifications prédéfinies concernant leur comportement et leur performance, garantissant ainsi que le projet fonctionne comme prévu par rapport aux exigences spécifiées.
  - **Tests d'intégration :** Évaluation du déploiement dans un environnement similaire à la production afin de prévenir les problèmes liés au flux de données et aux ressources disponibles.
  - **Tests de régression :** Empêche les modifications récentes de la base de code d'introduire des ruptures dans les fonctionnalités existantes, des erreurs dans le projet ou des dégradations qualitatives.



2. Aspects fondamentaux de la science des données

2.1. Comprendre le domaine

Un projet de science des données peut être essentiellement considéré comme l'usage et l'implantation de plusieurs outils et pratiques analytiques en vue de résoudre un problème métier, c'est-à-dire une tâche relevant d'un domaine d'expertise précis. À cet égard, aligner les efforts de la science des données avec l'expertise métier est un aspect essentiel dans la couverture de chacune des étapes d'un projet de science des données réussi :

1. Pertinence vis-à-vis des buts fixés : le fait de comprendre et garder à l'esprit le problème métier garantit que le projet ne déviera pas de ses objectifs.
2. Expertise métier : une compréhension approfondie, non seulement des objectifs spécifiques au projet, mais également du champ plus large dans lequel il s'inscrit, permet la bonne interprétation des résultats préliminaires et ouvre des pistes et perspectives à explorer pertinentes au projet.
3. Extraction de connaissance et développement de modèles : identifier les caractéristiques (ou features) pertinentes vis-à-vis du problème métier est de la plus haute importance en vue de l'implantation de modèles à la fois précis et efficaces. L'expertise du métier peut ici apporter des corrélations pré-identifiées entre features et objectifs.
4. Une vigilance et une remise en question continues des besoins et solutions développées sont essentielles à l'obtention de bénéfices tangibles à la fin du projet.

Pour ces raisons, un *data scientist* doit toujours chercher, ce depuis le tout début et tout au long du développement d'un projet, à construire et approfondir ses connaissances du domaine auquel le projet est rattaché.

2.2. Comprendre et définir le projet

Afin de garantir la réussite de tout projet, il est de la plus haute importance de définir les buts et objectifs de manière claire et traçable. C'est également vrai pour les projets de science des données. Entraîner des modèles sans une définition claire de l'évaluation des performances empêchera in fine la mise en production. Au commencement d'un projet de science des données les parties prenantes devraient dès lors :

1. Définir les besoins logiciels associés.
2. Décider s'il s'agit bien d'un projet de science des données, ou si l'objectif fixé est atteignable sans développement logiciel. Dans cette seconde hypothèse les étapes suivantes peuvent être ignorées.
3. Définir les besoins en termes de performance des modèles, et définir des métriques associées.
4. Sélectionner un (sous-)ensemble des données à conserver pour l'évaluation.
5. S'assurer que les données d'entraînement comme les données d'évaluation sont représentatives de l'environnement de production.
6. Itérer sur les points précédents plusieurs fois tout au long de la mise en œuvre du projet. Redéfinir les objectifs au besoin.

2.3. Aspects légaux et éthiques

Les aspects légaux et éthiques sont également des considérations cruciales qui couvrent toute l'étendue d'un projet de science des données. En effet, chaque étape peut être l'objet de régulations et de mise en conformité légale.

1. **Traitement de données personnelles :** les principes suivants devraient être observés :

   a. Définition claire de la finalité, de la réponse spécifique à la question concrète posée dans le cadre du projet. Cette définition doit déterminer quelles données sont nécessaires à l'obtention de ces résultats.
   b. Établissement d'une base juridique.
   c. Apport de l'information aux personnes concernées.
   d. Documentation dans les dossiers de traitement.
   e. Conformité avec les dispositions de la loi du 1er août 2018 relative à la protection des données.

2. **Gestion des données :** Tout au long du cycle de vie d'un projet, les aspects légaux et éthiques infusent la gestion des données :

   a. **Collecte des données et lois sur la confidentialité :** la confidentialité des données et la légalité de la collecte de types spécifiques de données doivent être au cœur des considérations dès le départ. Cela couvre le respect du Règlement général sur la protection des données [2] (RGPD) si les données contiennent des données à caractère personnel. En outre, suivez l'évolution du Règlement sur la gouvernance des données [3] (*Data Governance Act*), du Règlement sur l'intelligence artificielle [4] (*AI Act*) et du Règlement sur les données [5] (*Data Act*). Si votre projet nécessite des données d'une autre administration publique, sachez que cet accès n'est pas automatique et pourrait être tout bonnement impossible. Discutez des données nécessaires au plus tôt dans le processus et impliquez les délégués à la protection des données de vos administrations dès que possible. Vous pouvez également consulter le CGPD.
   b. **Traitement, stockage et sécurité des données :** les obligations légales relatives à la sécurité et la protection des données imposent des restrictions sur le lieu et la manière dont les données sont stockées et traitées.
   c. **Conclusion du projet, rétention et suppression des données :** Les considérations légales quant à la rétention des données et le droit à l'oubli doivent être scrupuleusement observées à la conclusion d'un projet. En toute prudence, toute donnée non nécessaire à la production à long terme devrait être alors supprimée.

3. **Exploration et utilisation équitable des données :** l'utilisation des données en vue de leur exploration ou de l'ingénierie des caractéristiques (*feature engineering*) doit respecter les standards légaux et éthiques. Au-delà des droits à la propriété intellectuelle, cela a trait également à des préoccupations éthiques telles que les biais dans des données qui concernent des individus ou des populations, par exemple.

4. **Développement, évaluation et équité des données :** les considérations éthiques et légales en ce qui concerne les biais et la non-discrimination dépassent l'exploration des données et couvrent toute l'implantation de l'analyse de données au sein d'un modèle, de même que sa validation et son évaluation.
5. **Déploiement, intégration et transparence :** certains domaines et applications, plus spécifiquement dans des secteurs tels la finance ou la santé, s'accompagnent d'une stricte application de la réglementation concernant la transparence et l'explicabilité des applications fournies.
6. **Maintenance à long terme et suivi de la conformité :** au fur et à mesure que les cadres légaux évoluent, un projet en production doit être contrôlé et mis à jour pour rester en conformité avec lesdites évolutions.

À ces égards il est crucial, dans tout projet d'analyse de données, d'assurer l'usage responsable et équitable des données. Les pratiques suivantes visent à apporter un usage plus éthique des données :

- **L'anonymisation des données** se réfère à une désidentification irréversible de toute donnée personnelle. Cette approche est recommandée quand l'aspect personnel des données n'est pas nécessaire à la mise en œuvre réussie du projet. Cependant, l'anonymisation irréversible des données personnelles telle que l'entend le RGPD est difficile à réaliser en pratique. Dès lors, la faisabilité de l'anonymisation des données devrait être discutée avec un expert qualifié de la protection des données, tel que le délégué à la protection des données ou l'organisme qui accueille le *data scientist*.
- **Consentement éclairé :** dans certaines circonstances, l'obtention du consentement éclairé et explicite des individus (personnes concernées) est une obligation légale lors de la collecte ou de la réutilisation de données personnelles. Lors de l'obtention de ce consentement éclairé, l'individu doit être informé explicitement de l'usage qui sera fait de ses données personnelles. Les individus doivent avoir la possibilité d'accepter ou refuser l'usage de leurs données personnelles pour tout ou partie des objectifs du projet (par exemple en cochant une case dans un formulaire approprié).
- **La transparence** sur la collecte des données, leur stockage et leur utilisation est fondamentale afin de communiquer sans équivoque aux individus comment leurs données seront utilisées et qui y aura accès.
- **La responsabilité et le professionnalisme** dans le traitement des données, effectuée dans le respect des règles de l'organisation responsable, sont essentiels.
- **Éviter les préjudices** aux individus ou communautés doit toujours être au cœur des préoccupations dans le traitement des données qui concernent des individus, des populations, des sujets sociaux, etc. Il est alors crucial de considérer les impacts négatifs potentiels et de prendre des mesures pour minimiser les risques.
- **Les implications sociales et éthiques** sur des considérations sociétales plus larges doivent être scrutées. Évaluez les impacts éventuels de votre projet sur la société, des communautés ou des groupes vulnérables.
- **L'évaluation continue** permet aux *data scientists* d'évaluer régulièrement les implications éthiques de leurs pratiques. Restez informé de l'évolution des standards éthiques et adaptez vos pratiques en conséquence.

2.4. Qualité des données

La qualité des données concerne entre autres la pertinence des données par rapport aux objectifs du projet. Cela comprend différentes caractéristiques afférentes aux données :

- **FAIR :** veillez à ce que les données adhèrent aux principes FAIR [6] (*Findable Accessible Interoperable Reusable*).
- **L'exactitude** mesure à quel point les données transmettent fidèlement l'information relative au monde réel qu'elles sont censées transmettre. L'exactitude peut être mise à mal par des capteurs mal calibrés ou des erreurs de mesures, par exemple.
- **L'exhaustivité** s'intéresse au degré de disponibilité de toutes les données nécessaires au projet. Des données manquantes peuvent réduire la qualité de l'analyse ou même rendre l'analyse impossible en pratique.
- **La cohérence,** ou la manière dont les données issues de différentes sources s'alignent, que ce soit en termes de format, d'unité ou de temps.
- **La précision** correspond au niveau de détail des données. Cela assure que chaque donnée est précisément représentée sans ambigüité.
- **La fiabilité** indique si les données restent cohérentes dans le temps. Généralement, des données non fiables changent fréquemment ou révèlent des incohérences.
- **La pertinence** vis-à-vis de la tâche à accomplir ; il importe de déterminer si les données sont directement utilisables, telles quelles, en vue des objectifs du projet, ou si au contraire elles requièrent soit un prétraitement, soit des informations supplémentaires.
- **L'actualité,** la fraîcheur des données, concerne plus précisément le cas où des données historiques sont censées être et rester représentatives de données non-observées et d'événements associés à venir. Dans ce contexte il est essentiel de contrôler l'éventuelle péremption et déviation des données dans le temps.
- **La validité** vis-à-vis de règles et contraintes prédéfinies. Cela peut globalement concerner des formats attendus, des conditions numériques nécessaires au bon fonctionnement de certains algorithmes d'analyse des données, ou encore la mise en conformité légale.
- **La clarté** assure que les données peuvent être facilement comprises par leur public cible. Des définitions et labels clairs, ainsi que l'apport de métadonnées contribuent à une meilleure compréhension des données.
- **L'unicité** des données vise à prévenir la présence de doublons inattendus dans les données qui risquent de dégrader la robustesse de leur analyse.

2.5. Pipeline de la science des données

Un pipeline de la science des données vise à structurer l'ensemble des processus d'un projet de science des données comme une succession d'étapes qui partagent toutes les mêmes concepts de base. La succession de ces étapes mène *in fine* à l'implantation fructueuse de la transformation des données brutes en résultats attendus. Les étapes en question varient d'un projet à l'autre, en fonction des données brutes, des résultats à produire, des approches implantées, mais se retrouvent généralement, du moins en partie, dans la liste suivante :

1. Préparation des données
2. *Feature engineering*
3. Modélisation
4. Entraînement des modèles
5. Évaluation
6. Inférence
7. Interprétation

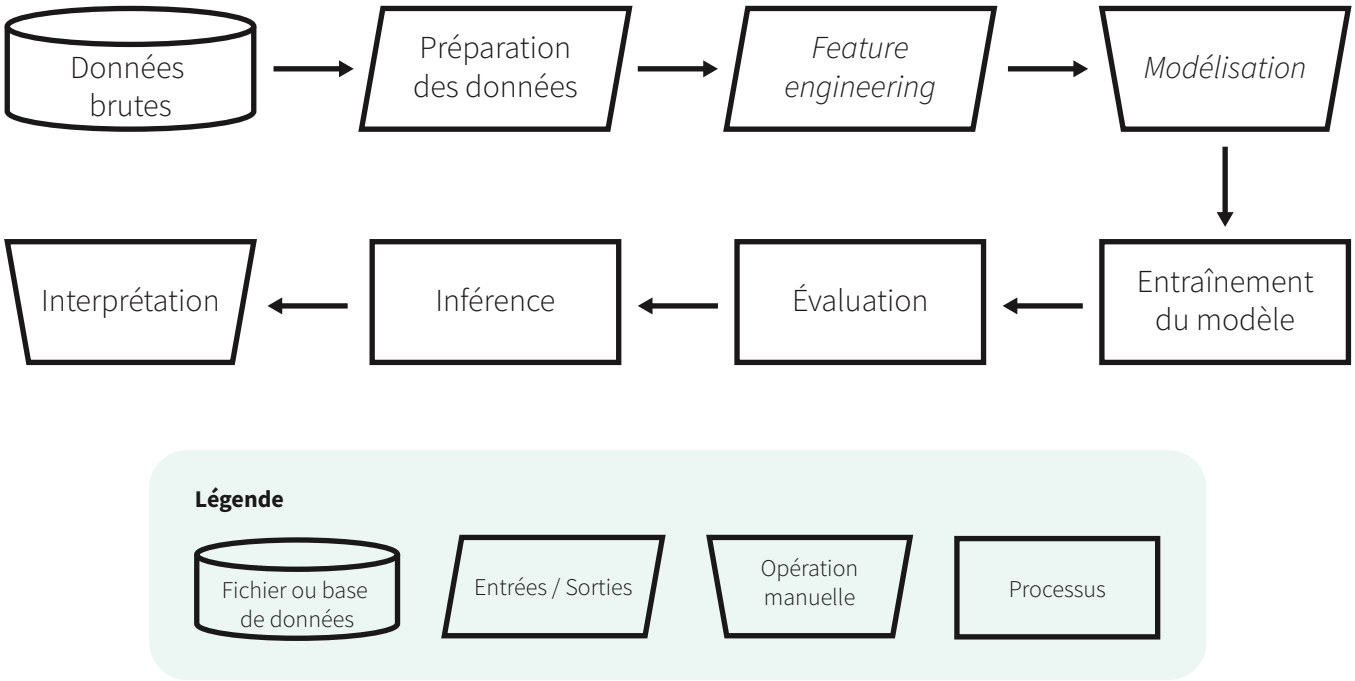


Figure 3:

Diagramme de flux [7] d'un pipeline de science des données

Ce document n'a pas pour objectif de décrire ces étapes en détail, mais plutôt de fournir des bonnes pratiques pour les implanter de manière reproductible et interchangeable dans un pipeline de science des données. Dès lors, toutes ces étapes peuvent être décomposées en trois phases :

1. Collecte des données : les données nécessaires à l'étape courante sont récupérées depuis leur source.
2. Transformation des données : les données en entrées sont analysées et traitées pour produire des données en sortie, qu'il s'agisse de données nettoyées, de modèles du *machine learning* entraînés sur un sous-ensemble d'entraînement des données, etc.
3. Stockage des données : les données en sortie sont stockées et conservées pour un usage ultérieur (que ce soit dans les étapes suivantes du pipeline ou lors de l'intégration en production).

Ces considérations mettent en évidence que toute étape relève des mêmes besoins techniques dans sa mise en œuvre :

1. Un stockage des données pour les données d'entrées et les données de sortie.
2. Un environnement d'exécution pour l'implantation des transformations des données.

Afin de minimiser les efforts dans la mise à dispositions de telles solutions, et dans un souci de systématisation des déploiements, tout projet de science des données devrait idéalement s'appuyer sur une solution unique et standardisée pour chacun de ces deux aspects. Avec une solution unique pour le stockage des données, il est alors possible de se concentrer uniquement sur les particularités des données à chaque étape du pipeline. L'environnement logiciel d'exécution (ou de calcul) devrait complètement s'abstraire du matériel qui l'héberge afin de permettre au *data scientist* de se concentrer seulement sur le code. La conteneurisation telle que définie par l'initiative OCI est habituellement la bonne manière de faire, en ce qu'elle simplifie grandement la maintenabilité comme la reproductibilité des projets, ce tout au long de leur conception et au-delà.

3. Bonnes pratiques

3.1. Pipeline de science des données

Les *data scientists* devraient connaître l'écosystème informatique et légal actuel. Le principal fournisseur d'outils et d'infrastructure pour le secteur public est le CTIE (Centre des Technologies de l'Information et de l'État). Dès lors et pour toute question relative à des besoins informatiques, contactez l'organisme[1]. Pour certaines administrations le fournisseur sera le CGIE (Centre de Gestion Informatique de l'Éducation), notamment pour le ministère de l'Éducation, ou le SIGI (Syndicat Intercommunal de Gestion Informatique), dans le cas des communes. Dans tous les cas, contactez d'abord vos responsables / départements informatiques afin de voir ce qu'ils ont à vous proposer.

Les *data scientists* devraient connaître les autres acteurs de l'écosystème des données : les organismes publics mêmes, qu'ils soient détenteurs ou responsables du traitement des données, le *Luxembourg National Data Service* (LNDS) en charge de faciliter les échanges de données, le CNPD (Commission Nationale pour la Protection des Données) et son homologue du secteur public le CGPD, les acteurs de la cybersécurité tels le GOVCERT (dès lors que les données doivent être protégées), ou le Portail *Open Data* comme fournisseur de données sans frais et prêtes à l'emploi. Le STATEC, en tant qu'institut national de statistiques est un autre acteur important, et il est important d'apprendre à connaître ses agents ainsi que leur métier.

Afin de mieux connaître l'écosystème des données et de l'informatique, envisagez de rejoindre les rencontres Meetup "*Data science*" et "*Business intelligence*". Ces groupes se rencontrent fréquemment et des représentants de tous les acteurs susmentionnés y participent pour discuter de projets, d'idées, et plus généralement apprendre à se connaître. Contactez-nous à l'adresse datascience@digital.etat.lu pour rejoindre ces groupes.

Chaque année, le ministère de la Digitalisation organise deux appels à projets, DATA4GOV et AI4GOV, qui sont de belles opportunités offertes aux *data scientists* pour soumettre des idées de projets allant au-delà des activités habituelles de leurs administrations respectives. Les projets sélectionnés bénéficient de l'expertise et du support du ministère de la Digitalisation.

1 Vous pouvez consulter la plateforme LogON

3.2. Biais

Les biais d'analyse sont des erreurs systémiques qui peuvent survenir lors de la collecte, de l'analyse, de l'interprétation ou de la publication des données. Comme ils conduisent à des conclusions erronées ou trompeuses, il est important d'identifier les différents types de biais pour les reconnaître et les prévenir :

- Le **biais de sélection** peut se produire lorsqu'une analyse qui vise à s'appliquer à une large population ne peut traiter qu'un sous-ensemble de celle-ci, généralement parce que la taille de la population totale rend impossible la collecte de données relatives à l'ensemble de la population ou la mise en œuvre d'approches à l'échelle de celle-ci. Le biais en question est introduit lorsque la sélection ne garantit pas que l'échantillon soit représentatif de l'ensemble. À partir de là, les conclusions reposeront sur une hypothèse fausse, ce qui produira des données biaisées qui reflètent la perte (non détectée) de la représentativité des données. Le biais de sélection peut également être introduit lors de l'étape de nettoyage des données, au cours de laquelle l'application de certains filtres aux données pourrait produire un échantillon non aléatoire de la population. La comparaison des statistiques descriptives générées à partir de l'échantillon avec les statistiques agrégées disponibles sur la population peut aider à détecter les biais de sélection.
- Le **biais du survivant** apparaît lorsque seule une partie de ce qui doit être représenté est parvenue à la collecte de données, ne conservant souvent que des données bien structurées et bien formatées, donc en supposant que seules les données conformes aux attentes contiennent des informations exploitables. Bien qu'elle soit essentiellement vraie, cette hypothèse ne tient pas compte du fait que les données inutiles sont informatives en elles-mêmes (par exemple, les données dégradées d'un capteur en disent long sur la santé du capteur ou du réseau par lequel il envoie des informations), mais conduit également à une surestimation des comportements sans faille dans les systèmes auxquels les données se rapportent.

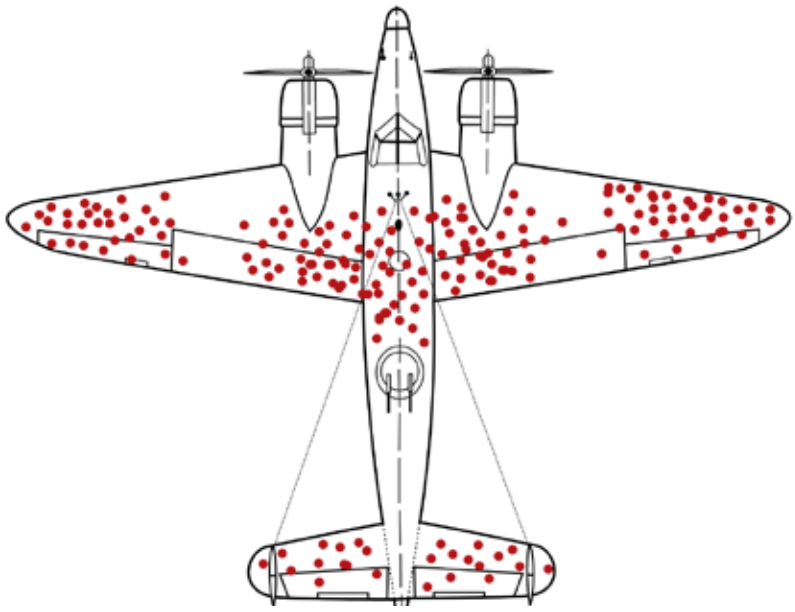


Figure 4 : Localisation des impacts dans l'avion. McGeddon, CC BY-SA 4.0, via Wikimedia Commons. Illustration du biais des survivants, inspirée de l'article fondateur d'Abraham Wald "A method of estimating plane vulnerability based on damage of survivors" (1943) [8]

- Le **biais de confirmation / biais de l'observateur** fait référence à la tendance à privilégier les données qui confirment des croyances préexistantes ou valident des hypothèses. Cela conduit à une interprétation orientée et incomplète des données qui ne fait que contribuer à leur propagation.
- Le **biais d'échantillonnage** survient lorsque l'échantillon de données utilisé pour l'analyse n'est pas représentatif de la population sous-jacente, entraînant des conclusions erronées. Même à partir de données représentatives, il est possible d'échantillonner un sous-ensemble non représentatif et biaisé de celles-ci. Cela conduit à des résultats et des conclusions non généralisables, qui ne s'appliquent qu'à l'échantillon biaisé.
- Le **biais des valeurs aberrantes (*outliers*)** peut apparaître lorsque les données contiennent des observations qui sont : a) radicalement distinctes de la majorité des autres données au point qu'elles restent aberrantes tout au long de l'analyse ; et b) représentatives d'une partie des données beaucoup plus importante que leur part dans la population générale. Cela conduit à un biais significatif dans l'analyse, où les valeurs aberrantes entravent l'analyse des normes et des tendances dans les données.
- Le **biais de publication** fait référence à la tendance à identifier, sélectionner ou publier uniquement des résultats significatifs et positifs, en omettant ceux qui ne le sont pas, ce qui peut fausser la perception des données, déformer les résultats réels et surestimer les effets sous-jacents.

En plus d'entraver la qualité d'un projet en science des données, ces biais introduisent également le risque de se propager jusque dans la publication de résultats et la prise de décision en fonction de ceux-ci. Par exemple, des biais dans n'importe quelle partie d'une analyse de science des données axée sur les individus peuvent conduire à des conclusions inéquitables, voire discriminatoires, lorsqu'elles sont généralisées à l'ensemble de la population.

3.3. Sécurité et conformité

Il est important d'assurer la sécurité des données et le respect des réglementations, en particulier lorsqu'il s'agit d'informations sensibles ou de données à caractère personnel.

Le RGPD (Règlement Général sur la Protection des Données) définit la notion de protection de la vie privée dès la conception [9] comme une « *protection des données dès la conception* ». Avec son article 25, « *Protection des données dès la conception et protection des données par défaut* », le RGPD oblige les organisations à intégrer des mesures de protection de nature technologique tout au long du cycle de vie du traitement des données. L'accent est mis sur les données personnelles, et l'obligation de les limiter à ce qui est strictement nécessaire au projet, en termes de collecte, de traitement, de conservation et d'accessibilité.

Au-delà du RGPD et lorsque les données personnelles sont essentielles à une solution, le concept de *secure by design* (sécurité dès la conception) fournit des lignes directrices pour des mesures de sécurité robustes à mettre en œuvre dès le début d'un projet. Les principales mesures sont le chiffrage, le contrôle d'accès rigoureux et des évaluations régulières de la sécurité. Celles-ci garantissent que les données personnelles incorporées dans une solution déployée sont traitées avec une sécurité maximale, prévenant ainsi les risques et gardant le traitement des données personnelles conforme au RGPD.

Dans une réflexion plus large sur le respect des lois, un *data scientist* doit disposer des compétences pour discerner les données sensibles et la raison pour laquelle elles sont sensibles (vie privée, propriété intellectuelle, secret gouvernemental ou européen, *etc.*). En outre, se familiariser avec la législation qui touche à ces raisons permet de respecter des obligations légales plus larges et des considérations éthiques.

Enfin, la veille législative est primordiale pour se tenir informé de l'évolution des exigences, qu'elles proviennent du RGPD ou d'autres instruments juridiques. Cela s'applique non seulement au développement et à la mise en production d'un projet, mais aussi aux données stockées, qui peuvent rester stockées pour une utilisation ultérieure, en particulier lorsqu'on essaie d'assurer la reproductibilité. Cette approche proactive permet de maintenir les projets alignés sur les dernières normes juridiques, de protéger la vie privée des individus et de respecter les normes éthiques, en promouvant la responsabilité et la discipline au sein de la communauté de la science des données à l'égard de ces obligations.

3.4. Extraction de connaissances

L'extraction de connaissances est le processus d'extraction d'informations significatives, d'identification de motifs ou de comportements à partir de données brutes grâce à des techniques d'analyse et de modélisation.

- **Comprendre le champ du problème :** pour mieux cerner le problème, il est essentiel de développer une compréhension approfondie de la situation que vous cherchez à résoudre ainsi que des connaissances spécifiques au domaine concerné. La recherche d'une expertise dans le domaine atténue les risques de s'écarter des objectifs du projet, en guidant l'analyse des données avec une compréhension plus approfondie de la signification réelle des variables et de leurs interactions.

- **L'analyse exploratoire des données (AED)** permet de mieux comprendre les relations entre les variables et d'identifier des modèles ou des corrélations entre les données. Il s'agit d'extraire des données des statistiques, de visualiser des modèles, d'identifier et de nettoyer les incohérences, et d'établir des relations entre les variables. Grâce à des techniques capables de révéler des corrélations entre les variables, les tendances dans les données, l'AED permet à un *data scientist* d'obtenir de meilleures informations et de formuler des hypothèses pour le développement d'analyses et de modèles.
- **Traiter les valeurs manquantes :** l'absence de données transmet une information ; il est donc primordial de l'aborder d'une manière ou d'une autre. Il faut comprendre pourquoi ces valeurs sont absentes : si le processus de génération de données est lié à un processus métier, parlez-en dans votre administration. Il peut y avoir une très bonne raison pour laquelle ces données sont manquantes ; essayer d'imputer ou d'écarter ces observations pourrait être absurde. L'imputation à l'aide de statistiques simples est à éviter, car elle pourrait introduire un biais statistique (l'imputation par la moyenne n'est conseillée que lorsque les absences de données se produisent de manière complètement aléatoire – missing completely at random dans la littérature – ce qui est très rarement le cas). Idéalement, il conviendrait d'employer des méthodes prédictives pour imputer ces données, et le faire par plusieurs méthodes pour tenir compte de l'incertitude introduite par la procédure d'imputation elle-même. Envisagez des outils d'imputation de données tels que ceux fournis par *sklearn* [10] pour Python ou *mice* [11] pour R.
- **Encoder des variables catégorielles :** il peut être nécessaire de convertir des variables catégorielles en représentations numériques à l'aide de techniques telles que l'encodage 1 parmi n (*one-hot*) ou l'encodage cible. Il convient d'explorer des méthodes d'encodage avancées telles que l'encodage par fréquence ou l'encodage par moyenne pour les variables catégorielles à cardinalité élevée.
- **Détection et traitement des valeurs aberrantes :** pour détecter des valeurs aberrantes, il est d'abord nécessaire de profiler les données. Les approches courantes se concentrent sur la caractérisation de la distribution sous-jacente qui est déterminée par la moyenne et l'écart-type. Ces informations peuvent ensuite être utilisées pour déterminer les valeurs aberrantes qui se trouvent aux extrêmes de la distribution. Le traitement des valeurs aberrantes peut donner lieu à des techniques communes au traitement des valeurs manquantes. Les valeurs aberrantes peuvent également être un indice que la population est mal définie : si vous analysez des données sur des maisons, mais que vous avez une maison de 20 chambres, il est probable que cette « maison » soit un « hôtel ». Il faut s'assurer qu'on travaille avec un échantillon bien défini et représentatif de la population qu'on souhaite étudier. Il est important de réfléchir à la signification des valeurs aberrantes. Par exemple, si l'on suppose que le processus de génération de données suit une loi de puissance, alors on peut s'attendre à des valeurs très élevées, inhabituelles (mais rares) (vente de livres, morts dans des catastrophes naturelles, *etc.*).
- Le **traitement de données textuelles** vise à extraire des données d'un texte pour alimenter les outils numériques, tels que des statistiques ou l'entraînement de modèles du *machine learning* dédiés aux données numériques. Il s'agit non seulement de symbolisation (*tokenisation*) ou de racinisation pour les tâches de traitement du langage naturel, mais aussi d'analyses telles que TF-IDF [12] pour extraire des statistiques sur l'utilisation de chaque mot spécifique dans les données textuelles.

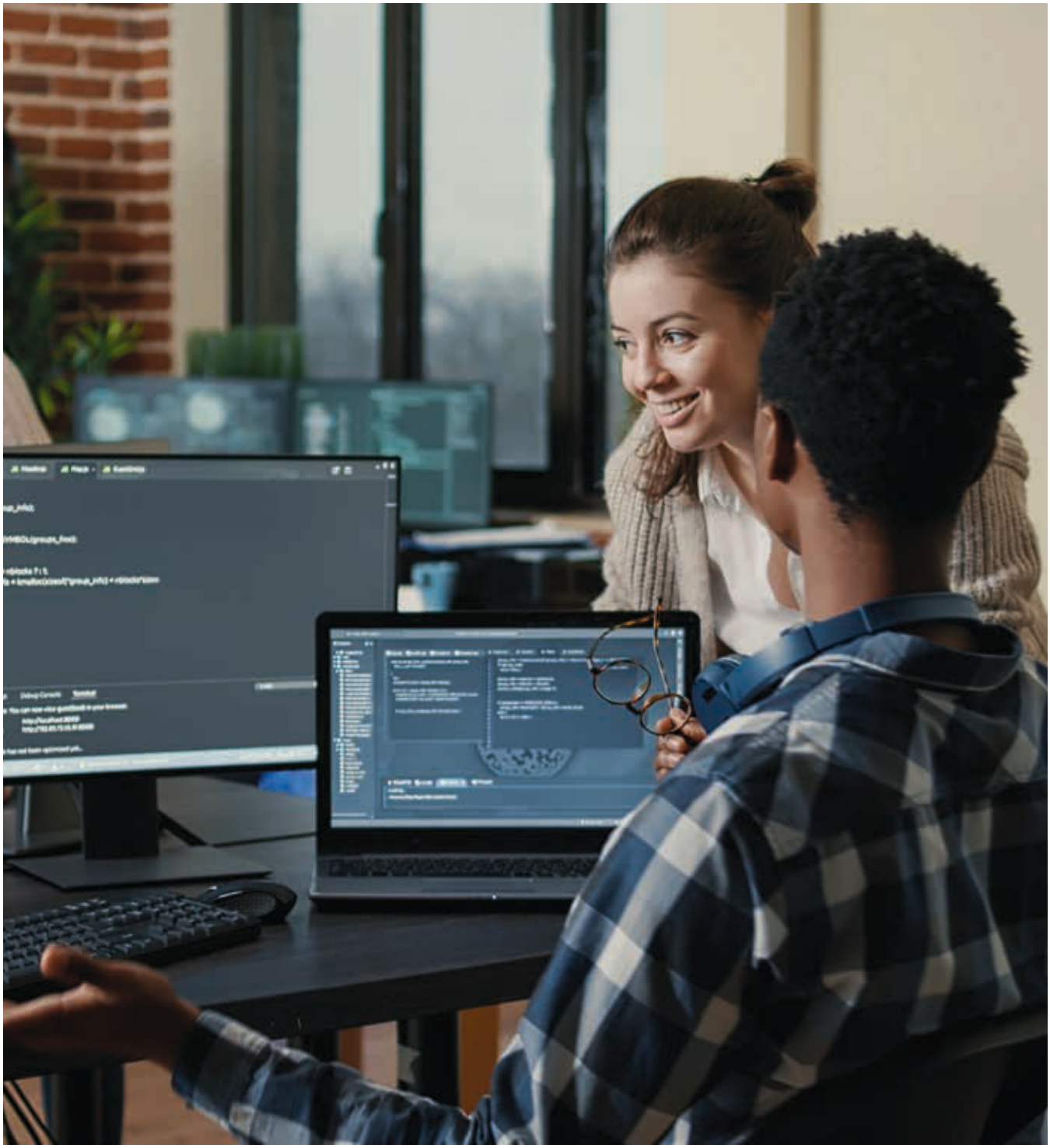
3.5. Visualisation des données

La visualisation des données concerne la présentation graphique d'informations complexes afin d'exposer des informations significatives. Une visualisation efficace des données facilite la communication avec un large éventail de publics, et vise donc à rendre les modèles, les comportements et les relations dans les données plus compréhensibles. Chenxin Li fournit une liste complète, bien que subjective, de directives plus techniques [13].

- **Identifiez votre public :** comprenez qui recevra vos visualisations et adaptez-les à leur niveau d'expertise et à leurs besoins spécifiques. Présentez avec clarté pour le public visé.
- **Choisissez les bons graphiques :** sélectionnez les types de graphiques appropriés, qui représentent efficacement les relations ou les modèles de vos données. Les diagrammes à bandes, les graphiques linéaires, les nuages de points, les histogrammes, etc. ont des objectifs différents.
- **Étiqueter et annoter :** étiquetez les axes, les points et fournissez des titres et des légendes clairs pour transmettre un contexte et faciliter l'interprétation. Utilisez des annotations pour mettre en évidence les principales conclusions ou tendances.
- **Simplifiez et épurez :** évitez l'encombrement visuel en supprimant les éléments inutiles. Mettez l'accent sur les informations les plus critiques et simplifiez la conception pour une meilleure compréhension.
- **Réfléchissez aux couleurs :** utilisez la couleur à bon escient pour mettre en évidence des points ou des tendances importants. Maintenez la cohérence et assurez la lisibilité, en tenant compte des palettes adaptées aux daltoniens.
- **Mettez en contexte :** ajoutez du contexte à vos visualisations en incluant des légendes, des descriptions ou des résumés qui expliquent l'importance des données et les informations qui en découlent.
- **Assurez la lisibilité et l'accessibilité :** utilisez des tailles de police appropriées, un texte lisible et un contraste suffisant pour une lecture facile. Rendez les visualisations accessibles à tous les publics, y compris aux personnes handicapées.
- **Conception réactive et interactivité :** concevez des visualisations réactives et adaptables à différents appareils. Utilisez des éléments interactifs (le cas échéant) pour permettre aux utilisateurs d'explorer davantage les données.
- **Validez et testez :** testez vos visualisations sur un échantillon du public cible pour recueillir des commentaires et assurez-vous qu'elles transmettent efficacement le message souhaité.
- **Mettez l'accent sur la narration :** créez un flux narratif dans vos visualisations pour raconter une histoire. Guidez l'auditoire à travers les données en présentant une séquence logique d'informations.
- **Amélioration continue :** Apprenez des commentaires et itérez sur vos visualisations. Efforcez-vous d'améliorer continuellement la conception et l'efficacité en fonction des réponses des utilisateurs et de l'évolution des besoins.

3.6. *Machine Learning*

Le *machine learning* fait référence à un ensemble de techniques algorithmiques et statistiques qui sont utilisées pour déduire automatiquement des relations et des modèles dans les données. Ce processus d'apprentissage produit un modèle d'inférence qui peut être employé pour extrapoler, anticiper des observations, des événements ou des comportements en se basant sur de nouvelles données conformes, dans leur format, à celles observées pendant l'apprentissage.



- **L'hypothèse de modèle** concerne la représentation des données, car elle englobe les schémas observés dans les données qui sont appris durant le processus du *machine learning*. Comme cette hypothèse repose sur le résultat de la découverte de données, de l'exploration et de l'extraction de connaissances, elle s'entraîne sur des parties des données identifiées comme des features en entrée, qui proviennent d'observations du monde réel ou de préanalyses basées sur celles-ci, et sur d'autres parties traitées comme des labels en sortie, ou des conclusions à tirer des *features*.
- **La préparation des données** nécessite une expérimentation et une validation afin de comprendre les données et de préparer les features et les labels pour l'apprentissage du modèle. Outre les techniques d'extraction de connaissances, diverses approches spécifiques au *machine learning* peuvent être appliquées, en fonction des données et des objectifs du projet :

  - **Transformer les variables numériques :** Envisager d'appliquer des transformations (logarithme, racine carrée, *etc.*) aux variables numériques afin de rendre leurs distributions plus gaussiennes ou de réduire l'asymétrie. Cela peut même être obligatoire lorsque certaines couches du modèle nécessitent exclusivement des données non nulles.
  - **Normaliser ou standardiser** les *features* numériques s'il est nécessaire de s'assurer qu'elles soient du même ordre de grandeur. Cette opération est étroitement liée à la transformation numérique et est obligatoire pour que de nombreuses techniques aient une chance de converger lorsqu'elles s'appuient fortement sur le fait que toutes les features sont comparables dans l'espace des *features*.
  - ***Features* temporelles :** En général, les horodatages sont soit des horodatages machine, soit une variante d'un horodatage formaté selon la norme ISO 8601. Bien que les deux transmettent des informations non ambiguës, il n'est pas rare que les modèles temporels et les comportements soient caractérisés par des facteurs plus cycliques et saisonniers, tels que l'heure du jour, le mois de l'année, etc. Préparer les *features* de l'heure et de la date consiste à en extraire les informations pertinentes, telles que le jour de la semaine, le mois, la saison ou la différence de temps entre les événements, afin de tenir compte de la nature cyclique de ces *features*.
  - **Traitement du déséquilibre des classes (pour la classification) :** bien qu'il ne soit pas toujours conseillé de traiter le déséquilibre des classes dans la préparation des données [14], étant donné que le déséquilibre des classes peut être intrinsèquement important lors de l'entraînement d'un classificateur (le suréchantillonnage d'une classe rare peut conduire à une sur-prédiction), il y a certains cas où cela peut être intéressant. Par exemple, un modèle de *machine learning* qui serait formé pour prédire les catastrophes naturelles est extrêmement susceptible d'apprendre à n'en prédire aucune, ce tout en atteignant une précision extrêmement élevée. À cet égard, le traitement du déséquilibre des classes implique des techniques telles que le suréchantillonnage, le sous-échantillonnage ou l'utilisation de fonctions de perte pondérées pour garantir un apprentissage équitable du modèle [15]. Mais il convient également de porter un regard critique sur le problème en question : vaut-il la peine de fournir des efforts pour prédire des événements très rares, ou serait-il préférable d'investir du temps pour essayer de se préparer et d'atténuer les effets lorsque la catastrophe se produit ?

- **Séparation entraînement / validation / test :** La formation de modèles de *machine learning* au moins deux ensembles de données qui ne doivent jamais se chevaucher afin d'éviter les fuites de données. De manière générale, le modèle affine ses paramètres internes sur la base des corrélations découvertes entre les *features* et les labels dans l'ensemble d'entraînement. Le deuxième ensemble est ensuite utilisé pour mesurer combien ces paramètres transmettent toujours les mêmes corrélations sur des exemples non vus ; il sert également à affiner les hyperparamètres (par exemple, l'architecture du modèle ou les détails de la procédure d'apprentissage). Si possible, un troisième ensemble distinct doit être préparé par un *data scientist* pour s'assurer que les paramètres et hyperparamètres appris se généralisent bien sur des données inconnues. Habituellement, le deuxième ensemble est appelé ensemble de validation et le troisième, ensemble de test, mais ces deux termes sont parfois inversés. En effet, étant donné que les premier (apprentissage) et deuxième (validation) ensembles sont tous deux utilisés pendant la formation, ils peuvent tous deux introduire un sur-apprentissage (*overfitting*) ou des biais inhérents à ces ensembles. S'assurer que le modèle fonctionne toujours bien sur un troisième ensemble indépendant permet d'atténuer ce risque. Étant donné que le troisième ensemble vise à empêcher le surapprentissage sur le deuxième ensembre, il devrait idéalement appartenir à une tierce partie. Cela garantit qu'aucune décision ne peut être prise qui, en favorisant la conformité avec le troisième ensemble, introduirait un biais indésirable. La séparation entraînement / validation / test doit être effectuée avant toute autre opération sur les données afin d'éviter les fuites de données : par exemple, l'ingénierie des *features* doit être effectuée après la séparation et dans le cadre de la boucle de validation croisée, ce qui signifie que toutes les étapes de l'ingénierie des *features* doivent être effectuées sur chaque ensemble séparé.

- **Validation croisée :** La validation croisée ou *cross validation* (CV) développe l'idée de diviser les données avant l'apprentissage [16]. Au lieu d'un seul découpage statique, elle évalue la généralisation du modèle sur des données inédites en validant le modèle sur plusieurs découpages des mêmes données en deux ensembles, l'un primaire et l'autre secondaire. L'ingénierie des *features* doit être effectuée dans la boucle de CV afin d'éviter les fuites. La CV est également une bonne option lorsque le *dataset* est trop petit pour créer la répartition entraînement / validation / test suggérée au préalable. Les données temporelles et les données spatiales sont deux types de données spécifiques pour lesquelles il convient d'être prudent avec la validation croisée. Comme les techniques habituelles de validation croisée reposent sur la répartition de points de données uniques entre les ensembles d'apprentissage et de test sans tenir compte de leur contexte, cela entraîne une perte de colocalisation temporelle ou spatiale des données. Le regroupement des données dans une fenêtre temporelle de blocs spatiaux (par exemple, blockCV) avant la validation croisée est obligatoire pour que l'approche présente un quelconque intérêt.
- **L'optimisation des hyperparamètres** fait référence à l'ajustement des paramètres de configuration externes d'un modèle qui ne sont ni appris ni modifiés pendant l'apprentissage, mais qui peuvent augmenter ou diminuer ses performances de manière significative, tels que les paramètres de validation croisée, la répartition des données, le taux d'apprentissage, etc. Il s'appuie sur des techniques telles que la recherche en grille, la recherche aléatoire ou l'optimisation bayésienne pour trouver la meilleure combinaison afin d'améliorer les performances. L'objectif est d'affiner les hyperparamètres afin d'optimiser la précision et les capacités de généralisation du modèle, qui peuvent être mesurées par les résultats de l'apprentissage ainsi que par l'exécution du modèle sur l'ensemble de validation. Il est important de noter que ce réglage ne doit pas intervenir trop tôt dans un projet de science des données, et qu'il est possible d'envisager de l'omettre si les résultats et les performances sont à la hauteur des attentes. Des modèles et des hyperparamètres simples facilitent la reproductibilité, la maintenabilité et l'extensibilité.

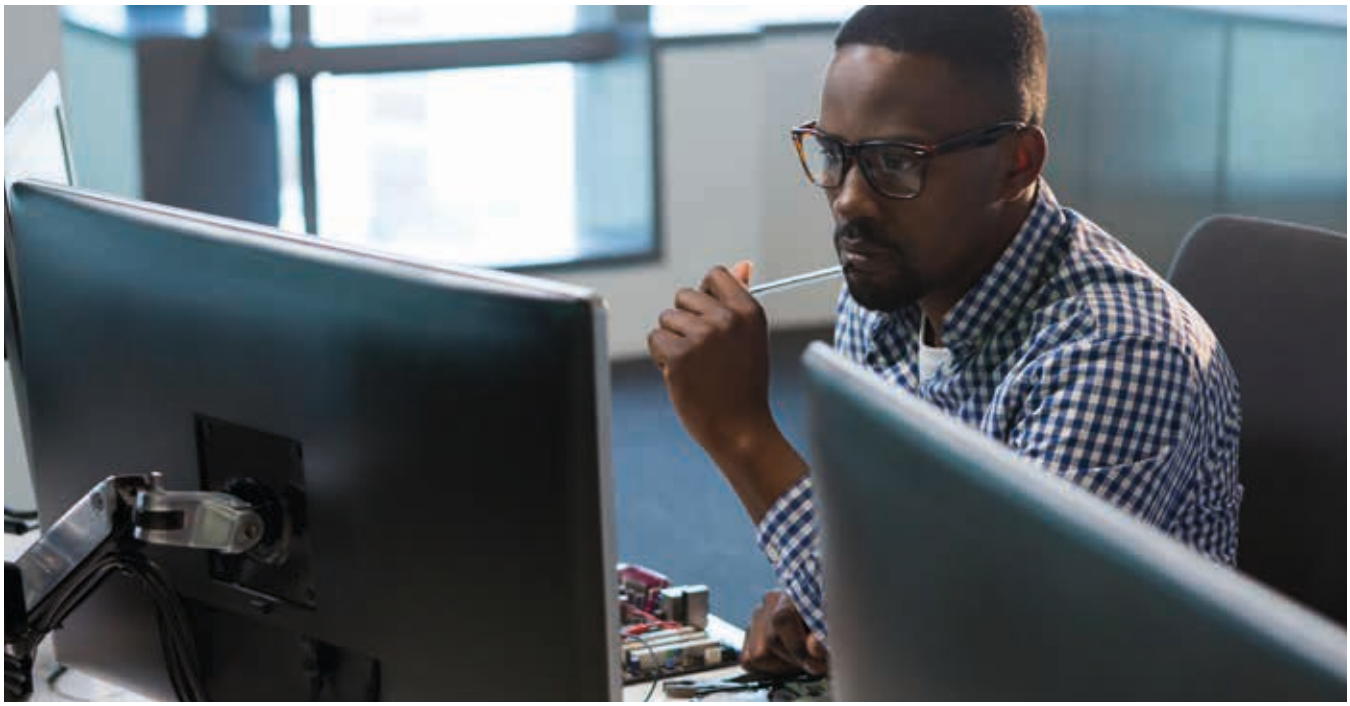
- **La fixation de la graine aléatoire (*random seed*)** pour toutes les tâches de *machine learning* est obligatoire pour garantir la reproductibilité des résultats. En s'assurant que toutes les parties pseudo-aléatoires de l'algorithme de *machine learning* (initialisation des paramètres d'apprentissage, division des données, etc.) suivent la même génération d'une exécution à l'autre, on peut a) reproduire exactement les expériences passées et obtenir exactement les mêmes résultats ; et b) travailler sur des paramètres manuels ou sur la préparation des données et comparer des résultats qui ne peuvent différer qu'en raison de ces changements. Néanmoins, le fait de s'en tenir à une seule graine comporte le risque d'utiliser accidentellement une graine qui favorise considérablement l'entraînement. L'analyse de la manière dont le changement de graine affecte le résultat est une bonne pratique pour contourner ce problème. En outre, si le changement de graine affecte systématiquement les résultats, cela peut même remettre en question l'approche ou le modèle utilisé. Cependant, cette analyse ne doit pas une quête de la meilleure graine, favorisant ainsi les résultats escomptés et introduisant alors un biais de confirmation dans le pipeline de données. Une meilleure pratique consiste plutôt à réaliser exactement les mêmes expériences sur plusieurs graines aléatoires et à rendre compte de la performance moyenne, lorsque cela est possible d'un point de vue calculatoire.
- **Sélection du modèle :** Choisir les algorithmes ou les modèles appropriés en fonction du défi posé et des caractéristiques des données. Expérimentez avec plusieurs modèles et comparez leurs performances à l'aide de techniques de validation croisée. Bien qu'ayant peu ou pas d'effet entre les modèles qui peuvent utiliser des architectures internes très différentes, une graine aléatoire fixée pour fractionner les données reste très importante pour garantir la validité de la comparaison des performances des modèles.
- **Régularisation** : L'utilisation de techniques de régularisation (par exemple, la régularisation L1/L2) permet d'éviter le sur-apprentissage (*overfitting*) et le sous-apprentissage (*underfitting*) en contenant ou en lissant les paramètres appris et en pénalisant les modèles complexes, ce qui améliore en fin de compte la capacité du modèle à traiter de nouvelles données inédites.
- **Méthodes d'ensemble :** Envisager d'utiliser des méthodes d'ensemble (par exemple, *bagging, boosting*, *stacking*) pour combiner plusieurs modèles afin d'améliorer les performances et la robustesse.
- **Interprétabilité du modèle :** Dans la mesure du possible, il faut s'efforcer de rendre le modèle interprétable, en particulier dans les scénarios de prise de décision critiques. Comprendre comment le modèle fait des prédictions contribue à la confiance en celui-ci et une compréhension plus approfondie du problème sous-jacent. Il est de plus en plus important de comprendre comment les modèles parviennent à leurs prédictions ou à leurs décisions, en particulier dans les scénarios exigeant l'interprétabilité ou le respect de la législation.

3.7. Ingénierie logicielle

La science des données englobe aujourd'hui une série de pratiques d'ingénierie logicielle visant à assurer un développement, une collaboration et une maintenance efficace et performante :

- **Gestion de versions (pour le code) :** Utiliser un système de contrôle de version (par exemple, *git* [17]) pour gérer les changements dans le code, le modèle et les données. Cela facilite la collaboration, le suivi des modifications et aide à revenir à une version précédente si nécessaire.
    - Les branches permettent aux *data scientists* de travailler sur une base de code isolée de la base principale. Elles facilitent à la fois le développement parallèle en quittant la branche principale et la collaboration par le biais de fusions vers la branche principale. Les branches sont cependant une épée à double tranchant, avec le risque que des branches périmées passent inaperçues et encombrent la base de code. À cet égard, la suppression périodique des branches périmées permet de rationaliser la gestion des branches et de ne garder en vie que les branches pertinentes. Envisagez d'adopter le développement basé sur le tronc [18] si vous travaillez au sein d'une équipe ou d'un projet.
    - De plus, les fichiers *.gitignore* [19] permettent de s'assurer que les fichiers qui ne doivent pas être transmis, que ce soit en raison de leur taille, de leur type ou pour des raisons légales ou éthiques, ne le seront pas.

- **Les tests unitaires** visent à valider de petites portions individuelles de code. En garantissant la fiabilité des fonctions, les tests unitaires permettent de prévenir les erreurs de bas niveau et d'améliorer la robustesse des algorithmes qui servent à l'analyse des données.
- **Les revues de code** doivent être menées par une tierce personne et visent à inspecter le code pour en estimer la lisibilité, assurer la mise en œuvre des meilleures pratiques, détecter les erreurs et maintenir une qualité de code globalement élevée. Le retour d'information des collègues, en plus de respecter les normes de bonnes pratiques et de garantir la fiabilité du code, permet également de susciter de nouvelles idées et de nouvelles pistes à suivre.
- **La documentation** est un aspect essentiel de la collaboration et de la maintenabilité. À cet effet, l'utilisation d'outils de génération de documentation (par exemple., sphinx [20], doxygen [21], roxygen [22], *etc.*) permet de regrouper le code et la documentation dans le même fichier. Par exemple, les descriptions de classes et de fonctions écrites dans les commentaires peuvent être utilisées pour générer de la documentation.
- **Les plateformes collaboratives** (par exemple *GitHub* [23], *GitLab* [24], *Bitbucket* [25], *etc.*) permettent le travail en équipe en fournissant non seulement le contrôle des versions, mais aussi le suivi des problèmes, des fonctions de revue de code et une documentation intégrée.
- **Les pipelines CI/CD,** aujourd'hui intégrés dans toutes les plateformes collaboratives, permettent l'automatisation des tests et des déploiements, accélérant ainsi ces processus tout au long du cycle de vie d'un projet de science des données. Ils systématisent également les tests et les déploiements, atténuant ainsi le risque d'introduire des erreurs.
- Il est essentiel de **tester les pipelines de données** pour garantir la fiabilité du traitement des données, en termes de stabilité, de fiabilité, de fonctionnalités et de résultats. Par exemple, on peut utiliser testthat [26] pour R ou pytest [27] pour Python. Si possible, exécutez les tests après chaque transfert vers une plateforme collaborative à l'aide d'environnements d'exécution (*runners*) automatisés.



3.8. Reproductibilité

La capacité à répliquer et à vérifier les analyses est cruciale pour garantir la validité et la fiabilité des résultats, ce qui en fait une pratique clé de la science des données. Pour un projet de science des données, il s'agit de la possibilité, pour toute personne autorisée :

1. D'accéder au moins aux données brutes, éventuellement aussi aux données préparées et nettoyées.
2. D'accéder aux pipelines de données.
3. D'exécuter des pipelines de données sur données brutes ou nettoyées.
4. De répliquer et comparer les résultats.

Ces accès doivent être garantis le plus tôt possible et tout au long du cycle de vie d'un projet et même rester en place par la suite. Plusieurs solutions et pratiques existent et doivent être utilisées, mises en œuvre, dans un souci de reproductibilité :

- **Scripts au format texte :** rédigez votre analyse en scripts au format texte, à savoir .py ou . R pour Python et R respectivement. Les scripts sont par nature prêts pour exécution, car ils peuvent généralement être déployés tels quels. Les outils interactifs, tels que les *Jupyter Notebooks*, doivent être évités. Les outils interactifs peuvent être utiles lors de l'exploration des données à un stade préliminaire, mais ils doivent être clairement séparés du pipeline du projet de science des données et ne doivent donc pas être distribués. Les outils sans-code sont également à proscrire.
- **Horodatage des données d'entraînement :** Fixez les *datasets* d'entraînement et de test pour pouvoir reproduire les mêmes expériences.
- **Horodatage des modèles utilisés et entraînés** pour le *machine learning* afin d'améliorer la traçabilité et l'évolution du modèle sur l'ensemble du projet.
- **Enregistrement des paramètres d'entraînement avec les modèles entraînés.** Utilisez des formats lisibles par l'homme et analysables par la machine pour stocker, mettre à jour et accéder à ces paramètres, tels que JSON ou YAML.
- **Fixer les versions de logiciel et des bibliothèques utilisées :** n'utilisez jamais la balise latest, utilisez plutôt des versions fixes, par exemple « ==3.2.1 ». De plus, les versions de logiciels (par exemple Python 3.9.17) doivent être spécifiées sans ambiguïté dans les instructions (fichier *README*). Il est également fortement recommandé d'utiliser des environnements virtuels individuels pour chaque projet. Pour Python, *pyenv* [28] est couramment utilisé pour gérer différentes versions de Python ainsi que des environnements virtuels au sein de chaque version de Python. Pour R, l'utilisation du paquet *renv* [29] est également fortement recommandée pour au moins protéger les versions des paquets par la génération d'un fichier verrou.
- **Transparence :** Fournir des explications et une documentation claire sur les méthodes, les hypothèses et les limites favorise la confiance, la compréhension et la prise de décision éclairée.
- **Fournissez des instructions claires** (fichier *README*), qui décrivent étape par étape la façon d'exécuter le code. Bien que certaines analyses de données soient complexes et donc difficiles à maîtriser, elles ne doivent jamais être difficiles à exécuter ni à reproduire.

- **Conteneuriser votre analyse :** fournissez votre analyse sous forme de conteneur, déterministe et reproductible (par exemple, Docker [30], Kubernetes [31]). Assurez-vous que la même image est générée lors de la création du conteneur en utilisant des empreintes plutôt que des balises. Utilisez des fichiers verrou (par exemple, à l'aide de la commande *pip freeze* si vous utilisez *pyenv* ou un fichier *renv.lock* si vous utilisez *renv* dans R) pour installer les dépendances au moment de la génération. Dans le doute, privilégier les solutions conformes à l'*Open Container Initiative* [32] est un choix prudent en vue d'une maintenabilité à long terme. Les références externes, les secrets et les informations d'identification doivent être exclus du répertoire de code, ainsi que de toute image de conteneur générée.
- **Automatisez,** dans la mesure du possible et autant que faire se peut afin d'éviter les erreurs humaines. L'automatisation améliorera la vitesse et la précision de l'analyse, grâce à l'utilisation d'outils d'automatisation de la configuration et de l'exécution du flux de travaux tels que *Snakemake* [33] ou *Nextflow* [34], par exemple.

3.9. Maintenabilité

Un code bien structuré, lisible et documenté garantit que les analyses peuvent être facilement maintenues, mises à l'échelle et améliorées, contribuant ainsi à la longévité d'un projet.

- **Démarrer tout nouveau projet avec les bibliothèques les plus récentes** garantit l'accès à des fonctionnalités optimisées et à jour. Cela atténue également le risque de dépendre de versions non maintenues. À l'inverse, les bibliothèques obsolètes (sans mise à jour récente) ou sans information de support explicite doivent être traitées avec prudence lorsqu'elles sont intégrées dans un projet, et idéalement simplement évitées.
- **Maintenez à jour les versions des bibliothèques,** si possible (dépendances, vulnérabilités), mais vérifiez que vous disposez de suffisamment de tests unitaires pour vous assurer que le code continue de produire les mêmes résultats avec ces nouvelles bibliothèques.
- **Envisagez de refondre votre code** pour l'améliorer. Les améliorations portent sur la lisibilité, l'efficacité, la gestion des erreurs, la correction des vulnérabilités potentielles et l'évolution des exigences du projet.
- **Continuez à mettre à jour le logiciel,** mais assurez-vous de pouvoir revenir en arrière en cas de problème.
- **Continuez de mettre à jour les modèles** avec de nouvelles données pour remédier à la dérive des modèles. Au fil du temps les données évoluent, les modèles entraînés au cours du développement seront finalement confrontés à des données qui ne sont pas entièrement conformes à leurs hypothèses et exigences. Le réentraînement à intervalles prolongés permet non seulement de maintenir en vie un projet, mais aussi de réaligner les modèles déployés afin qu'ils restent fidèles à leurs objectifs et à leur efficacité.
- La **mise à jour continue des *datasets*** est primordiale pour remédier à la dérive des données et doit être mise en œuvre conjointement avec la pratique précédente. En effet, l'inclusion de nouvelles données permet de rattraper les changements de comportements ou de modèles, réalignant ainsi la précision et la représentativité des modèles réentraînés.
- **Utilisez le style de programmation recommandé** pour le langage de votre choix. Pour R, essayez de suivre celui de Google [35] ou Posit [36], pour Python, suivez la PEP 8 [37].

Il est recommandé de définir et de respecter une structure générale de projet au sein d'une organisation. Ce qui suit illustre un exemple de structure qui peut être utilisée. Dans l'exemple ci-dessous et si vous collaborez en utilisant *git*, vous ne devez versionner que les fichiers au niveau racine ainsi que le contenu du dossier src et inclure les autres répertoires dans votre fichier *.gitignore*.

Figure 5 : Exemple d'un projet de science des données

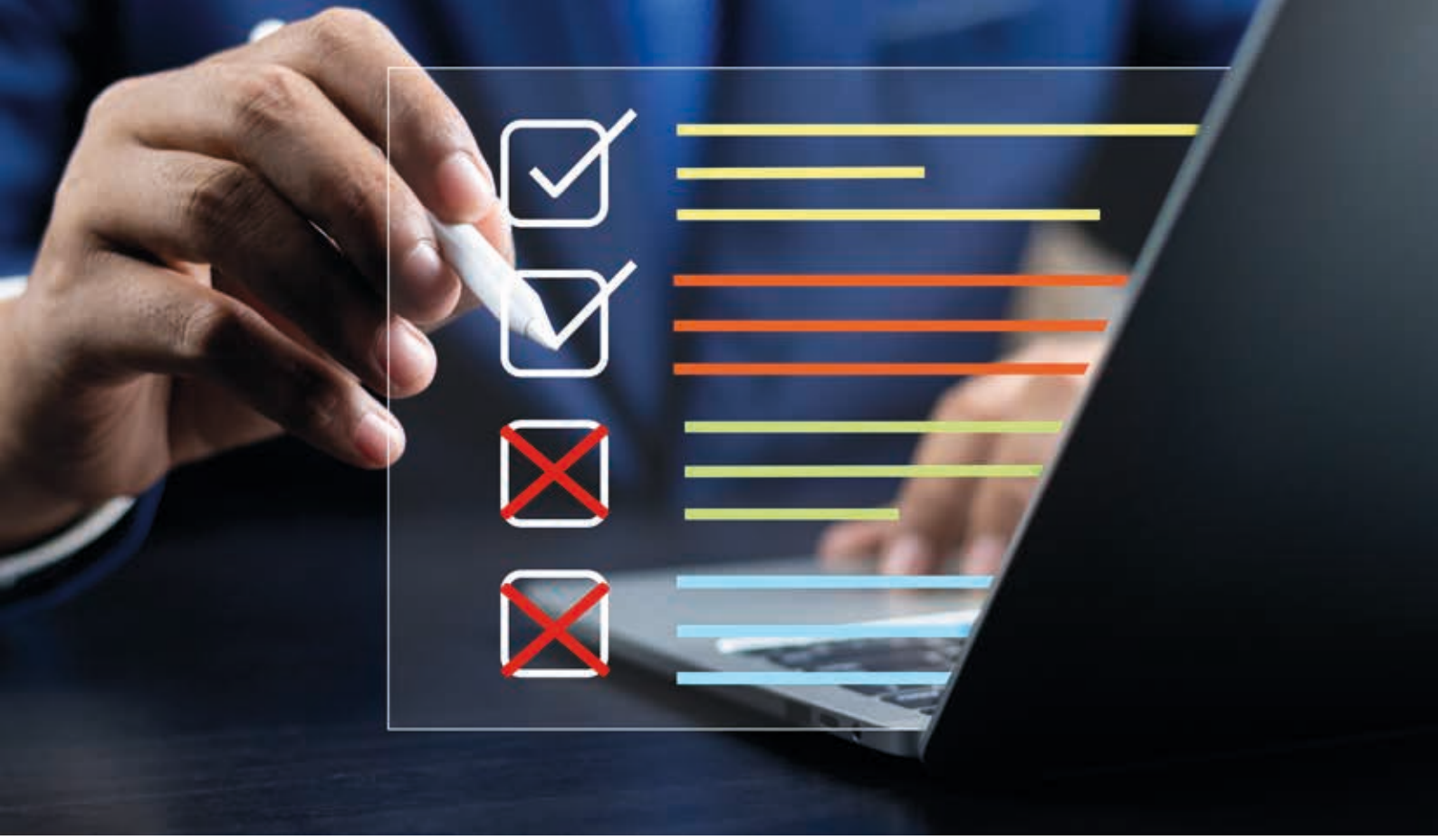
```
.                    # niveau racine
├── .dockerignore    # exclut fichiers des commandes Dockerfile COPY et ADD
├── .gitignore       # exclut des contenus de commits (sauf fichiers racine et src)
├── Dockerfile       # script pour construire ou exécuter le projet dans un conteneur
├── README.md        # présente le projet et comment le reproduire
├── data
│   ├── external     # données tierces
│   ├── raw          # données brutes, suffisantes pour reproduire l'exécution
│   └── processed    # sortie de pipeline de traitement de données
├── exploration      # scripts et notebooks à ne pas partager
├── models           # modèles entrainés et sauvés à utiliser et tester en production
├── references       # documents relatifs au domaine métier
├── reports          # comptes-rendus des résultats
└── src              # répertoire du code du pipeline de traitement de données
    ├── data         # scripts de transformation de données
    ├── models       # scripts d'entrainement
    └── visualization  # scripts de communication visuelle des résultats
```

3.10. À faire et à ne pas faire

En conclusion de cette section nous proposons ici une liste de recommandations à suivre et d'écueils à éviter.

À faire

- **Définissez des objectifs clairs :** des objectifs techniques, commerciaux et des résultats attendus décrits sans ambiguïté constituent l'épine dorsale d'un projet de science des données.
- **Comprendre le domaine et les besoins métier :** une collaboration étroite avec les utilisateurs finaux est ici nécessaire et permet d'aligner les solutions de science des données aux attentes.
- **Collecte et préparation des données de qualité** (GIGO : *Garbage In Garbage Out*) : non seulement les données brutes doivent être de haute qualité pour permettre des résultats de haute qualité, mais leur préparation ne doit jamais réduire cette qualité, ce afin de maintenir la fiabilité tout au long du projet. Restez à l'affût des valeurs manquantes, aberrantes et des biais.
- **Modélisation et évaluation itératives :** cela permet d'éviter les dérives des données, conceptuelles et des modèles, mais aussi de détecter tout problème le plus tôt possible.
- **Reproductibilité et documentation** sont la clé de la collaboration, du support des travaux futurs, de la maintenabilité et de la continuité à long terme.
- **Formation et amélioration continues :** rester à jour sur les derniers outils et techniques, être ouvert à la mise en œuvre de nouvelles approches est crucial pour rester à la hauteur des attentes autour d'un projet de science des données.
- **Faites un pré-mortem :** prenez le temps d'identifier les risques et les raisons pour lesquelles un projet pourrait échouer. Réfléchissez à la façon de les éviter.
- **Commencez par définir des modèles et des outils :** commencez chaque projet en configurant les modèles, outils et infrastructures logicielles requis. Par exemple, commencez par créer un dépôt git, mettez en place les actions par défaut qui doivent se déclencher à chaque modification (lancer des tests par exemple), à chaque fusion, *etc*.



À ne pas faire

- **Surapprentissage et sous-apprentissage des modèles :** l'ajustement au plus près des modèles aux données d'apprentissage (surapprentissage) ou la simplification excessive des modèles (sous-apprentissage) peut réduire la précision prédictive des nouvelles données.
- **Biais dans les données et les modèles :** les données peuvent refléter des biais présents dans la société, conduisant à des modèles biaisés. Le fait de ne pas détecter et atténuer les préjugés peut entraîner des résultats injustes ou discriminatoires.
- **S'appuyer uniquement sur les algorithmes :** les algorithmes sont importants, mais il est tout aussi crucial de comprendre le domaine (métier) du problème et les besoins de l'entreprise ainsi que le contexte des données. Les algorithmes seuls ne peuvent pas résoudre tous les problèmes.
- **Préoccupations en matière de confidentialité et de sécurité des données :** une mauvaise gestion des données sensibles ou personnelles peut causer des atteintes à la vie privée, des problèmes juridiques ou nuire à la réputation d'une organisation.
- **Modèles trop compliqués :** n'utilisez pas un modèle complexe pour la seule complexité.
- **Le mieux est l'ennemi du bien** : ne vous perdez pas dans trop de détails et ne visez pas la perfection. Référez-vous aux objectifs du projet : lorsqu'ils sont atteints à la fois en temps et en qualité, passez à d'autres tâches.

4. Suivi et amélioration continue

Il est essentiel d'évaluer et d'affiner régulièrement les modèles pour s'assurer de leur pertinence et de leur précision au fil du temps, en particulier dans un environnement dynamique. Le suivi et l'amélioration continue comprennent à la fois des aspects techniques ainsi qu'une collaboration et une communication appropriées.

4.1. Aspects techniques

L'aspect technique dans le suivi et l'amélioration est principalement axé sur la façon de suivre vos propres progrès pendant le développement et de vous assurer que les modèles proposés sont adaptés aux exigences du projet.

- **Suivi de la qualité des données :** évaluez régulièrement la qualité des données entrantes, en adressant les problèmes tels que des valeurs manquantes, aberrantes ou des incohérences afin de maintenir l'intégrité des données.
- **Suivi de la performance des modèles :** au début du projet, définissez des critères et sélectionnez les données à utiliser pour évaluer les performances des modèles entraînés. Les parties prenantes doivent collaborer pour élaborer une stratégie d'évaluation qui soit réaliste et se rapproche autant que possible de l'environnement de production prévu, veillant à maintenir un effort gérable.
- **Échouer vite :** les exigences métier dans les projets en science des données peuvent souvent être écrasantes au début. Dans ce cas, il peut être utile de réaliser une preuve de concept (PDC) dans des conditions simplifiées pour déterminer si la solution pourrait fonctionner et pour mieux estimer l'effort nécessaire pour modéliser le produit final. Attention cependant à ne pas survendre une PDC d'une complexité considérablement réduite.
- **Détection et adaptation des dérives :** détectez les dérives conceptuelles et les changements de distribution, en déclenchant des ajustements de modèle ou un réentraînement pour s'adapter à l'évolution des modèles de données.
- **Réentraînement du modèle et mises à jour du *feature engineering* :** planifiez un réentraînement périodique du modèle à l'aide de données mises à jour et réévaluez les techniques de *feature engineering* pour améliorer l'interprétation et les performances.
- **Exploration et réglage des algorithmes :** restez informé des nouveaux algorithmes, expérimentez avec différentes options et affinez les hyperparamètres pour améliorer les performances du modèle.
- **Sécurité, conformité et suivi des ressources :** examinez régulièrement les protocoles de sécurité, assurez-vous de la conformité aux réglementations en rapport avec la confidentialité et surveillez l'utilisation des ressources pour un traitement efficace et sécurisé des données.

4.2. Collaboration et communication

Une collaboration et communication efficaces facilitent la compréhension commune et le partage d'informations, mais peuvent ne pas avoir d'impact direct sur les aspects techniques de l'analyse.

- **Documentation et partage de connaissances :** documenter l'architecture, les hypothèses et les limites du modèle, tout en favorisant une culture de partage des connaissances pour faciliter l'apprentissage continu au sein de l'équipe.
- **Retour d'expérience :** organisez des retours d'expérience pour une amélioration continue en fonction des commentaires des utilisateurs/parties prenantes. Planifiez des ateliers avec les utilisateurs cibles du modèle dès le début du processus de développement pour vous assurer que les exigences présentées en début de projet sont bien en accord avec les besoins des utilisateurs. Au besoin, n'hésitez pas à adapter les exigences, mais assurez-vous de documenter et de communiquer ces changements de manière transparente et compréhensible.
- **Communication des résultats et performance :** ne sous-estimez pas l'importance et l'impact d'un excellent compte rendu et d'une bonne présentation. En travaillant quotidiennement sur un projet, il est facile de garder une bonne vue d'ensemble de ses différents aspects. Les décideurs, qui sont moins en contact avec le développement doivent recevoir des rapports et des visuels qui aident à la compréhension de l'avancement et qui sont suffisamment détaillés pour permettre un bon retour d'information et de bonnes décisions métier.
- **Enseignements de fin de projet :** à la fin d'un projet, qu'il soit une réussite ou un échec, prenez l'habitude de documenter les étapes franchies, les défis rencontrés (notamment imprévus) et de partager les enseignements au sein de l'équipe afin de permettre un meilleur transfert de connaissances à d'autres projets et permettre de prendre des précautions pour éviter les problèmes rencontrés.

5. Gestion du cycle de vie

Il est essentiel d'évaluer et d'affiner régulièrement les modèles pour s'assurer de leur pertinence et de leur précision au fil du temps, en particulier dans un environnement dynamique. Le suivi et l'amélioration continue comprennent à la fois des aspects techniques ainsi qu'une collaboration et une communication appropriées.

5.1. Intégration continue (CI)

L'intégration continue pour la science des données comprend l'automatisation de l'intégration du pré-traitement des données, du *feature engineering*, de l'entraînement des modèles et de leur évaluation, le tout dans un même pipeline. En tirer parti implique les pratiques suivantes :

- **Effectuer des *commits* réguliers** permet d'identifier rapidement des changements introduisant potentiellement des ruptures dans l'existant. De telles ruptures sont plus faciles à réparer quand leur ampleur reste faible, non seulement parce qu'elles impliquent moins de code à revoir, mais aussi parce que le moment où elles sont adressées est plus proche du moment où ces ruptures ont été introduites. À long terme cela réduit drastiquement le temps nécessaire au débogage.
- **Les tests ne doivent pas être écrits par celui qui écrit le code.** Cela réduit non seulement le risque d'introduire les mêmes erreurs logiques dans le code et les tests, mais cela assure également la lisibilité et la reproductibilité du code via la collaboration.
- **Écrire et maintenir des tests** qui assurent **la couverture du code.** Cela requiert une compréhension approfondie de la variété des données et des cas d'utilisation auxquels le projet peut faire face. Maintenir les tests pour qu'ils supportent les nouvelles fonctionnalités de même que de nouvelles données est crucial, dès lors que chacun peut introduire des conditions qui rompent avec les hypothèses précédentes implantées dans les tests.
- Quand la compilation ou les tests échouent, **comprendre pourquoi.** Il se peut alors qu'un changement récent introduise un bug, mais il se pourrait également que les phases de compilation ou de test soient elles-mêmes périmées, qu'elle cessent d'être en adéquation avec les changements introduits et qu'elles doivent alors être mises à jour en ce sens. Si tel est le cas, essayez de trouver une tierce personne pour identifier et mettre en oeuvre les changements nécessaires au pipeline d'intégration.
- **Inclure des données structurellement représentatives** et maintenir ces structures à jour au fur et à mesure que les données évoluent. Cela assure que le pipeline de données puisse accepter et manipuler des données en temps réel, issues de la réalité de la production.

5.2. Livraison continue (CD)

La livraison continue pour la science des données comprend l'automatisation du déploiement des modèles, pipelines et changements de code associés au sein d'environnement de production et de démonstration. Ces tâches doivent être menées dans le but d'assurer la cohérence et la reproductibilité des pipelines de données et du déploiement des modèles, et incluent les pratiques suivantes :

- **Le versionnage et l'horodatage** tant des modèles que des données est crucial pour assurer la reproductibilité dans le temps : ainsi, un modèle dont on connaît les résultats précis sur un jeu de données spécifiques sera plus facilement réutilisable sur les mêmes données. En outre, la différence, en termes d'horodatage, entre modèles et données est essentiel dans l'identification d'une éventuelle dérive des données ou du modèle.
- **Implantez un mécanisme de retour en arrière** qui tire plein avantage du versionnage et de l'horodatage des modèles et des données. Cela permet des changements rapides et des retours à des versions précédentes des modèles et des données utilisées en déploiement.
- **Cohérence des environnements :** l'utilisation de scripts (*playbooks*) Ansible, par exemple, permet de spécifier les besoins nécessaires à l'exécution sans heurt du pipeline de données, ce tout en assurant que le déploiement dépende de ressources pré-identifiées (puissance de calcul, mémoire, *etc.*).
- **Implanter des contrôles de santé** réduit le risque de panne et de dégradation de performances dans les pipelines déployés. Cela vise à vérifier le fonctionnement des dits pipelines avec la vérification des résultats attendus quant aux entrées et sorties, à la latence, à la qualité des résultats, *etc*.
- **Planifier des déploiements collaboratifs** aide à établir une couverture des fonctionnalités à vérifier lors du déploiement d'une nouvelle version du pipeline. Sont ici concernés le respect des objectifs du projet, l'assurance que les modèles fonctionnent comme prévu, que les services s'exécutent sans heurt, *etc*. À ce stade tous les membres du projet devraient être impliqués, qu'ils soient *data scientists*, ingénieurs logiciels ou systèmes, experts du domaine, *etc*.

5.3. MLOps (*Machine Learning Operations*)

Le MLOps est un ensemble de pratiques et d'outils qui rationnalise et simplifie le cycle de vie des modèles d'*machine learning*, depuis le développement jusqu'au déploiement et au suivi. Les bonnes pratiques visent ici à assurer l'automatisation, la reproductibilité et le passage à l'échelle des processus de la science des données, qu'il s'agisse d'entraînement des modèles, de leur validation, de leur déploiement ou de leur suivi :

- **L'entraînement continu des modèles** devrait être implanté pour déclencher régulièrement et automatiquement les pipelines d'entraînement sur des nouvelles données. Cela permet de prévenir la dérive des données et des modèles en entraînant ces derniers sur des nouvelles tendances et des nouveaux comportements dans des données plus à jour.
- **L'orchestration des pipelines** *via* des outils dédiés simplifie la gestion d'analyses de données de bout en bout au sein de pipelines de données.
- La construction d'une **banque de variables (*feature store*)** permet l'identification, la gestion et la réutilisation d'une partie des *features* qui doivent être utilisées comme entrées, ce possiblement à n'importe quelle phase de l'analyse des données. Cela permet de garder trace des données nécessaires au bon fonctionnement de ces différentes phases, assurant ainsi la reproductibilité et la cohérence.
- **Les tableaux de bord de suivi et de journalisation** fournissent un aperçu non seulement des performances dans l'inférence des modèles, mais aussi des contrôles de santé des modèles en développement et en déploiement.

5.4. DevOps (*Development Operations*)

De manière similaire, le DevOps est un ensemble de pratiques et d'outils qui visent à aligner les tâches de développement d'un projet de science des données avec des considérations concernant leur déploiement et leur suivi. Ces pratiques promeuvent la collaboration et simplifient les processus en vue de la fiabilité et de l'efficacité de l'intégration de la science des données en déploiement :

- **Les boucles de rétroaction continue** visent à établir des processus d'orchestration du développement, des tests et des déploiements, généralement dans cet ordre, de telle sorte que chaque nouveau développement est testé avant d'être déployé, et que les informations collectées lors du déploiement orientent en retour les nouveaux développements.
- **La planification de réponse aux incidents** réduit l'introduction de changements de rupture via l'établissements de procédures répétables en réponses aux échecs de compilation ou de tests, de telle sorte que ces procédures deviennent des habitudes avec le temps.
- **L'amélioration continue** encourage la collaboration et la reproductibilité au travers de la systématisation de revues de code régulières et de sessions de refonte du code qui visent à produire du code plus efficace, plus lisible et plus testé.
- **La sécurité continue** couvre l'entièreté du projet de science des données et assure que les bonnes pratiques en termes de sécurité soient toujours mises en œuvre, ce à toutes les phases du projet.

Références

[1] Solomon Kurz, "Statistical Rethinking with brms, ggplot2, and the tidyverse", *Retrieved from osf. io/97t6w*, 2019.

[2] (2024) General Data Protection Regulation (GDPR) Compliance Guidelines. [En ligne]. https://gdpr.eu/

[3] "Regulation (EU) 2022/868 of the European Parliament and of the Council of 30 May 2022 on European data governance and amending Regulation (EU) 2018/1724 (Data Governance Act) (Text with EEA relevance)", *OJ L 152, 3.6.2022, p. 1–44*, 2022.

[4] "Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE (ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION LEGISLATIVE ACTS", *COM/2021/206 final*, 2021.

[5] "Regulation (EU) 2023/2854 of the European Parliament and of the Council of 13 December 2023 on harmonised rules on fair access to and use of data and amending Regulation (EU) 2017/2394 and Directive (EU) 2020/1828 (Data Act)", *OJ L, 2023/2854, 22.12.2023*, 2023.

[6] (2018) Open Consultation on FAIR Data Action Plan - LIBER Europe. [En ligne]. https://libereurope.eu/article/fairdataconsultation/

[7] "Information processing - Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts", International Organization for Standardization, Geneva, CH, Standard February 1985.

[8] Abraham Wald, "A method of estimating plane vulnerability based on damage of survivors", *Statistical Research Group, Columbia University. CRC*, vol. 432, 1943.

[9] (2024) Privacy by Design - General Data Protection Regulation (GDPR). [En ligne]. https://gdpr-info.eu/issues/privacy-by-design

[10] (2024) 6.4. Imputation of missing values — scikit-learn 1.4.0 documentation. [En ligne]. https://scikit-learn.org/stable/modules/impute.html

[11] (2024) Multivariate Imputation by Chained Equations • mice. [En ligne]. https://amices.org/mice/

[12] (2023) tf-idf. [En ligne]. https://en.wikipedia.org/wiki/Tf%E2%80%93idf

[13] (2024) GitHub - cxli233/FriendsDontLetFriends: Friends don't let friends make certain types of data visualization - What are they and why are they bad. [En ligne]. https://github.com/cxli233/FriendsDontLetFriends

[14] Ruben van den Goorbergh, Maarten van Smeden, Dirk Timmerman, and Ben Van Calster, "The harm of class imbalance corrections for risk prediction models: illustration and simulation using logistic regression", *Journal of the American Medical Informatics Association*, vol. 29, pp. 1525–1534, 2022.

[15] Khan Md Hasib et al., "A survey of methods for managing the classification and solution of data imbalance problem", *arXiv preprint arXiv:2012.11870*, 2020.

[16] Daniel Berrar and others, Cross-Validation., 2019.

[17] (2024) Git - Documentation. [En ligne]. https://git-scm.com/doc

[18] (2024) Trunk-based Development | Atlassian. [En ligne]. https://www.atlassian.com/continuous-delivery/continuous-integration/trunk-based-development

[19] (2024) Git - gitignore Documentation. [En ligne]. https://git-scm.com/docs/gitignore

[20] (2024) Welcome - Sphinx documentation. [En ligne]. https://www.sphinx-doc.org/en/master/

[21] (2024) Doxygen homepage. [En ligne]. https://www.doxygen.nl/index.html

[22] (2024) Get started with roxygen2. [En ligne]. https://cran.r-project.org/web/packages/roxygen2/vignettes/roxygen2.html

[23] (2024) GitHub: Let's build from here · GitHub. [En ligne]. https://github.com/

[24] (2024) The DevSecOps Platform | GitLab. [En ligne]. https://about.gitlab.com/

[25] (2024) Bitbucket | Git solution for teams using Jira. [En ligne]. https://bitbucket.org/

[26] (2024) Unit Testing for R • testthat. [En ligne]. https://testthat.r-lib.org/

[27] (2024) pytest: helps you write better programs — pytest documentation. [En ligne]. https://docs.pytest.org/en/8.0.x/

[28] (2024) GitHub - pyenv/pyenv: Simple Python version management. [En ligne]. https://github.com/pyenv/pyenv

[29] (2024) Introduction to renv • renv. [En ligne]. https://rstudio.github.io/renv/articles/renv.html

[30] (2024) Docker: Accelerated Container Application Development. [En ligne]. https://www.docker.com/

[31] (2024) Kubernetes. [En ligne]. https://kubernetes.io/

[32] (2024) Open Container Initiative - Open Container Initiative. [En ligne]. https://opencontainers.org/

[33] (2024) Snakemake | Snakemake 8.4.8 documentation. [En ligne]. https://snakemake.readthedocs.io/en/stable/

[34] (2023) A DSL for parallel and scalable computational pipelines | Nextflow. [En ligne]. https://www.nextflow.io/

[35] (2024) Google's R Style Guide | styleguide. [En ligne]. https://google.github.io/styleguide/Rguide.html

[36] (2024) Welcome | The tidyverse style guide. [En ligne]. https://style.tidyverse.org/

[37] (2013) PEP 8 – Style Guide for Python Code | peps.python.org. [En ligne]. https://peps.python.org/pep-0008/

Abréviations

- **AED** Analyse Exploratoire des Données
- **AI** Artificial Intelligence
- **CD** Continuous Delivery
- **CGIE** Centre de gestion informatique de l'éducation
- **CGPD** Commissariat du gouvernement à la protection des données auprès de l'État
- **CI** Continuous Integration
- **CNPD** Commission nationale pour la protection des données
- **CTIE** Centre des technologies de l'information de l'État
- **CV** Cross Validation
- **Dev** Développement
- **FAIR** Findable Accessible Interoperable Reusable
- **GIGO** Garbage In Garbage Out
- **JSON** JavaScript Object Notation
- **LNDS** Luxembourg National Data Service
- **ML** Machine Learning
- **OCI** Open Container Initiative
- **Ops** Opérations
- **PDC** Preuve de concept
- **PEP** Python Enhancement Proposal
- **RGPD** Règlement général sur la protection des données
- **SIGI** Syndicat intercommunal de gestion informatique
- **SRC** Source
- **TF-IDF** Term Frequency - Inverse Document Frequency
- **YAML** Yet Another Markup Language

Publié par
Ministère de la Digitalisation
info@digital.etat.lu
www.digitalisation.lu

Mai 2024