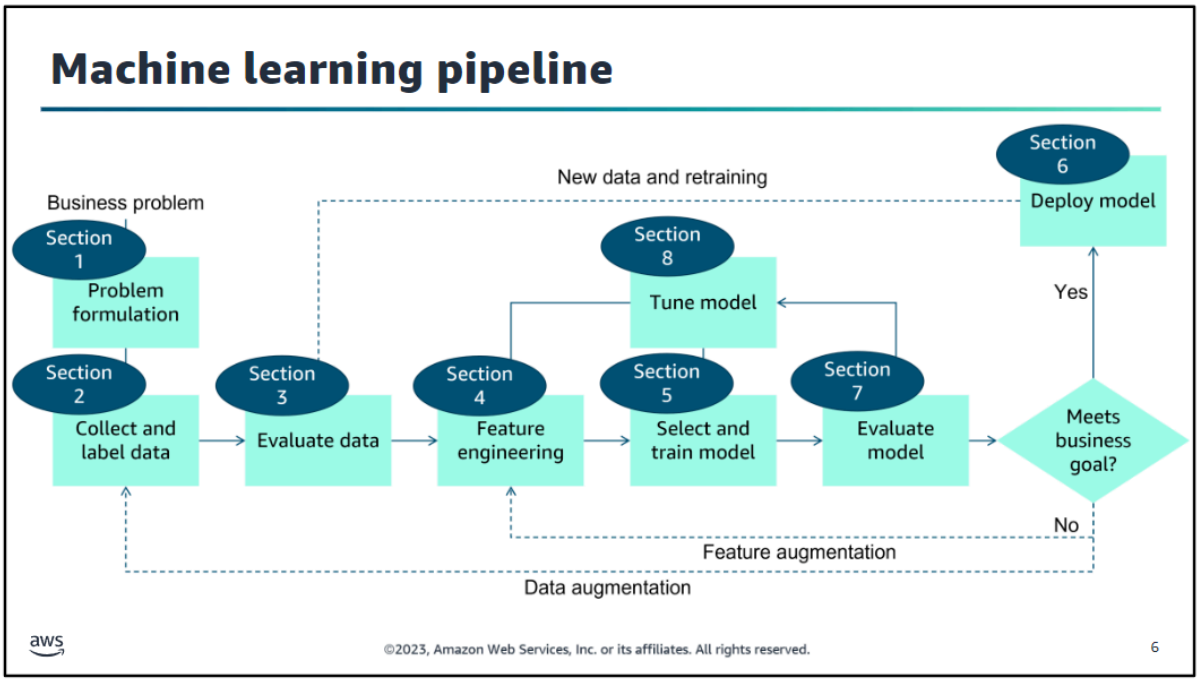


1. Section 1:
    a. Problem formulation:

Ceci correspond à comment formuler au mieux notre problème à fin de le comprendre au maximum et de le résoudre au mieux. Pour cela nous devons nous poser quelques questions clés suivantes :

- Est-ce que le problème est un problème de machine learning ?
- Le problème est-il supervisé ou non ?
- Quelle est la cible à prédire ?
- Avons nous accès aux données ?
- Quelle est la performance minimale ?
- Comment peut-on résoudre le problème manuellement ?
- Quelle est la solution la plus simple ?

Pour résumer ils nous faut donc :

- Définir des objectifs clairs (par exemple un problème de falsification de données)
- Choisir un type de problème (par exemple une anomalie)
- Identifier les données d'évaluation qui indiqueront le succès de notre modèle

2. Section 2:
    a. Collect and label data:

Cette section correspond aux techniques de collection et de sécurisation des données pour avoir un apprentissage automatique. Pour cela nous devons d'abord catégorisé nos différentes source de données :

- Données privées : Données que l'utilisateurs créer
- Données commerciale : AWS Data Exchange, AWS Marketplace, et autres sources externes
- Données open-source: Données à accès public (Kaggle)

Ensuite Amazon nous met à disposition un emplacement de stockage qui s'appelle Amazon Simple Storage Service (Amazon S3) pour stocker un grand ensemble de données. Avec ce dernier nous pouvons utiliser un ETL qui va nous permettre une extraction, une transformation et un chargement de la données.

3. Section 3:
    a. Evaluate data:

Cette étape correspondait à l'évaluation des données fournies pour être sur qu'elles sont au bon format. Ensuite nous utilisons Pandas qui est une librairie Python pour travailler avec nos données. Ceci nous permet de décrire nos statistiques et d'apprendre des choses sur notre dataset. Enfin pandas permet de visualiser nos données en détail en créant des graphiques (à point, courbe, bâtons, etc…).

4. Section 4:
    a. Feature engineering:

L'étape du feature engineering correspond à la sélection ou l'extraction des meilleures caractéristiques pour le machine learning. Il y a un prétraitement de données ce qui permet de travailler avec de meilleures données donc d'avoir de meilleurs résultats. Ce dernier possèdent 2 catégories qui sont :

- La conversion des données
- Le nettoyage de données manquantes ou imparfaite

Cependant il faut faire attention car la manière dont les données imparfaite sont gérées dans notre modèle influe sur ce dernier. Pour ce faire, on peut supprimer ou remplacer les lignes contenant des données imparfaites.

5. Section 5:
    a. Select and train model:

Cette étape permet de diviser nos données en trois ensembles : un ensemble d'entraînement, de test et de validation. Cela nous permet d'avoir un modèle plus précis. En général, nous devons allouer 80% des données à l'entraînement, 10% à la validation et 10% au test. L'ensemble d'entraînement est utilisé pour ajuster les paramètres du modèle, tandis que l'ensemble de validation permet d'évaluer la performance du modèle pendant l'entraînement. Enfin, l'ensemble de test est utilisé pour vérifier la capacité du modèle à généraliser sur des données jamais vues.

Pour un ensemble de données plus petit nous pouvons utiliser la validation croisée K-fold. On peut utiliser deux algorithmes pour l'apprentissage supervisé — XGBoost et Linear Learner. On peut utiliser k-means pour de l'apprentissage non supervisé. Ce dernier permet de regrouper les données en fonction de leur similarité. Enfin pour entraîner des modèles on peut utiliser Amazon SageMaker pour entraîner notre modèle.

6. Section 6:
    a. Deploy model

L'objectif de la phase de déploiement est de fournir un environnement géré pour héberger les modèles et permettre des inférences de manière sécurisée et avec une faible latence.

Une fois que le modèle est déployé en production, il est important de surveiller les données de production et de réentraîner le modèle si nécessaire. Les modèles nouvellement déployés doivent refléter les données actuelles de production. C'est un processus continu. SageMaker s'occupe du reste : il lancera les instances, déploiera le modèle et configurera un point de terminaison HTTPS sécurisé pour l'application.

L'application n'a besoin d'inclure qu'un appel API à ce point de terminaison pour réaliser des inférences avec une faible latence et un débit élevé. Grâce à cela, on peut intégrer les nouveaux modèles dans l'application en quelques minutes, car les modifications du modèle n'exigent plus de changements dans le code de l'application.
Il peut effectuer des vérifications d'état, appliquer des correctifs de sécurité et réaliser d'autres opérations de maintenance de routine, le tout avec une surveillance et une journalisation intégrées via Amazon CloudWatch.

Après avoir entraîné le modèle, on peut créer le endpoint soit via du code, soit en utilisant la console SageMaker. Les endpoints multi-modèles offrent une solution évolutive et rentable pour déployer un grand nombre de modèles.

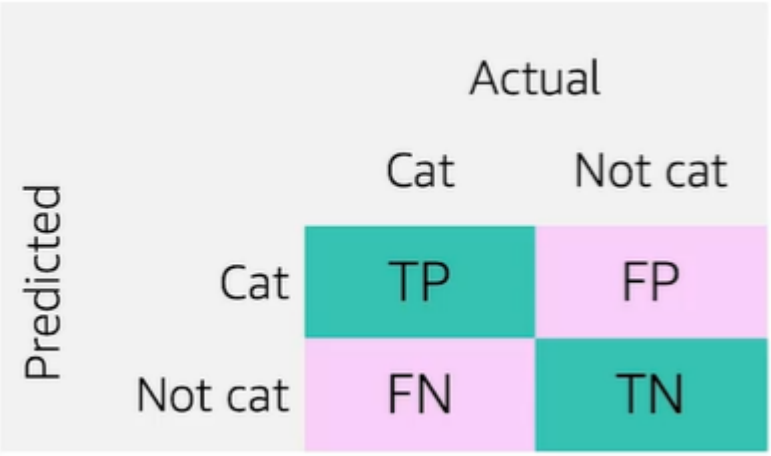
Résumé :
2 moyens de déploiements :
- Amazon SageMaker
- Batch transform

On déploie uniquement après avoir testé le modèle → générer des prédictions pour l'application cliente.
Créer un endpoint :

- endpoint pour un seul modèle pour des cas d'utilisation simples.
- endpoint multi-modèles pour prendre en charge plusieurs cas d'utilisation.

7. Section 7:
    a. Evaluate model

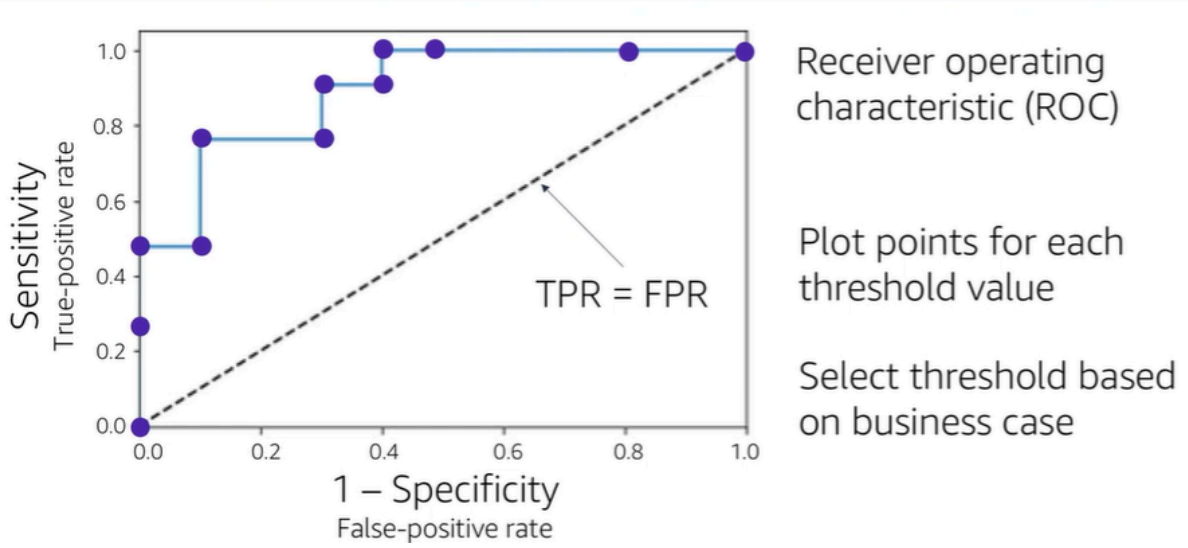
Il est nécessaire d'évaluer comment le modèle se comportera sur des données pour lesquelles tu connais déjà la réponse cible. 80 % des données pour l'entraînement, 10 % pour le test et 10 % pour la validation. Une partie importante de cette phase consiste à choisir la métrique la plus appropriée pour ta situation professionnelle. Pour aider à examiner la performance du modèle, tu peux comparer les valeurs prédites avec les valeurs réelles



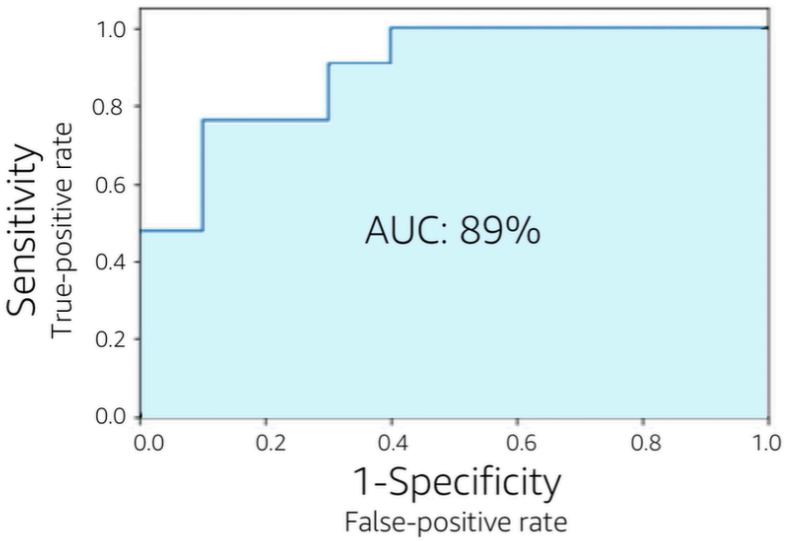
.

Dans une matrice de confusion, on peut obtenir une comparaison générale de la façon dont les classes prédits correspondent aux classes réelles.

Les modèles de classification vont renvoyer une probabilité pour la cible. Il s'agit d'une valeur indiquant à quel point l'entrée appartient à la classe cible, et elle sera comprise entre zéro et un. Pour convertir cette valeur en une classe, il est nécessaire de déterminer le seuil à utiliser.



Un graphique des caractéristiques de fonctionnement du récepteur, également connu sous le nom de graphique ROC, résume toutes les matrices de confusion générées par chaque seuil. Pour en construire un, tu calcules et traces la sensibilité ou le taux de vrais positifs par rapport au taux de faux positifs sur un graphique pour chaque valeur de seuil. Tu peux calculer le taux de faux positifs en soustrayant le spécificité de un.



Area under the curve

AUC-ROC is used to easily compare one model to another.

La partie AUC (Area Under the Curve) est l'aire sous la ligne tracée.

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}$$ Model's score

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$ Proportion of positive results that were correctly classified

$$\text{F1 score} = 2 \cdot \frac{\text{precision} \cdot \text{sensitivity}}{\text{precision} + \text{sensitivity}}$$ Combines precision and sensitivity

Résumé :

Plusieurs méthodes permettent de valider le modèle, notamment la méthode de séparation (Hold-out) ou la validation croisée K-fold. Il existe deux types d'évaluation de modèle : la classification (matrice de confusion, AUC-ROC) et les tests de régression (erreur quadratique moyenne).

8. Section 8:
    a. Tune model

L'ajustement des modèles, ou tuning des hyperparamètres, est une étape cruciale dans le développement de modèles de machine learning. Ce processus vise à améliorer les performances des modèles en optimisant les paramètres qui ne sont pas appris pendant l'entraînement.

Compréhension des Hyperparamètres

Les hyperparamètres sont des paramètres définis avant le processus d'apprentissage qui contrôlent le comportement du modèle et de l'algorithme d'optimisation. Parmi les hyperparamètres courants, on trouve :

- **Modèle** : type d'algorithme utilisé (par exemple, arbre de décision, SVM).
- **Optimiseur** : méthode d'optimisation pour ajuster les poids (ex. : Adam, SGD).
- **Data :** définir les attributs des données elles-mêmes.

Importance de l'Ajustement

L'ajustement des hyperparamètres est crucial pour :

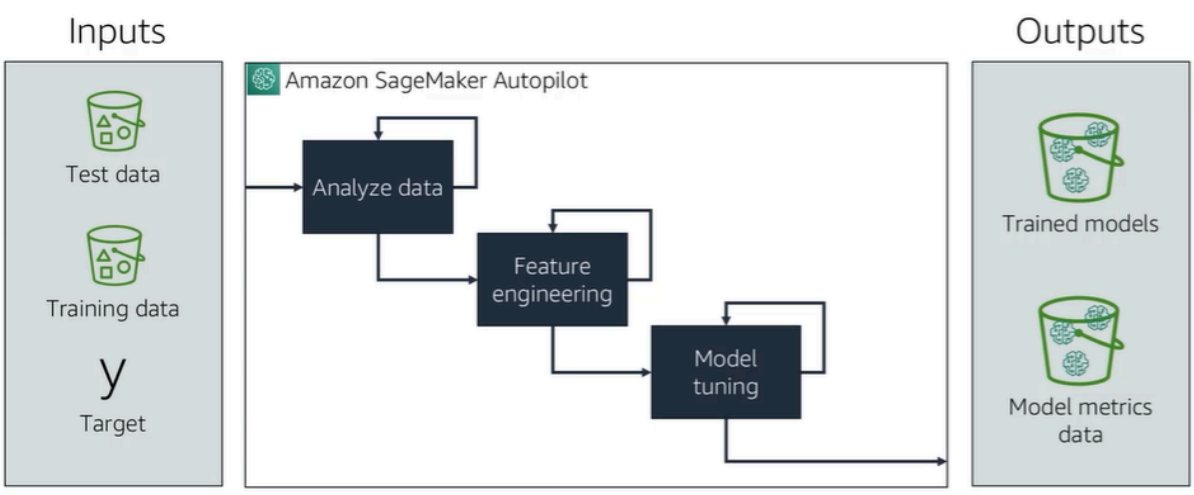
- **Améliorer la performance du modèle** : Un modèle bien ajusté peut améliorer significativement les métriques de performance, telles que la précision et le rappel.
- **Éviter le Surapprentissage** : Un ajustement adéquat aide à prévenir le surapprentissage en trouvant un équilibre entre la complexité du modèle et les données.

Méthodes d'Ajustement

Meilleures pratiques pour l'ajustement.

- Ne pas tout ajuster
- Limiter votre gamme de valeurs à ce qui est le plus efficace.
- lancer un job d'entraînement à la fois au lieu de plusieurs jobs en parallèle
- Dans les tâches d'entraînement distribué, s'assurer que la métrique d'objectif que l'on souhaite est bien celle qui est rapportée.
- Avec Amazon SageMaker, convertir les hyperparamètres à échelle logarithmique en hyperparamètres à échelle linéaire lorsque cela est possible.

Avec Autopilot, on peut créer un job qui fournit les données de test, d'entraînement et la cible. Autopilot analysera les données, sélectionne les fonctionnalités appropriées, puis entraînera et ajustera les modèles.



Résumé :

L'ajustement du modèle aide à trouver la meilleure solution au problème commercial. Les hyperparamètres peuvent inclure des éléments tels que le modèle, l'optimiseur, les données et le processus d'ajustement.

Utiliser Amazon SageMaker pour aider à ajuster les hyperparamètres, et utiliser Autopilot pour un développement plus rapide.