**Régression Linéaire : Définition et Fonctionnement**

La régression linéaire modélise la relation entre une variable dépendante yyy et une ou plusieurs variables indépendantes xxx en ajustant une ligne droite (également appelée "ligne de régression") à travers les points de données. L'équation de cette ligne est généralement de la forme :

Une image contenant texte, Police, ligne, capture d’écran

Description générée automatiquement

**Exemple en Python avec scikit-learn (Voir jupiter Notebook)**

Scikit-learn est une bibliothèque populaire pour le machine learning en Python. Elle offre des outils simples et efficaces pour l'analyse de données et la modélisation statistique, y compris la régression linéaire.

Imaginons un cas simple où nous avons des données sur la durée de vie restante de moteurs d'hélicoptères en fonction de l'heure totale de fonctionnement. Nous pouvons utiliser ces données pour entraîner un modèle de régression linéaire. Voici comment on pourrait faire cela en Python :

Dans le contexte de la régression linéaire, les matrices colonne (ou vecteurs colonne) sont souvent utilisées pour représenter les données d'entrée (features) et les étiquettes (labels) en raison de la manière dont les algorithmes de régression et d'apprentissage machine traitent les données. Voici pourquoi :

1. **Compatibilité avec les algorithmes** :
   * Les algorithmes de régression linéaire, tels que celui fourni par la bibliothèque scikit-learn, s'attendent à ce que les données d'entrée soient sous forme de matrice 2D où chaque colonne représente une caractéristique (feature) et chaque ligne représente une observation.
   * En utilisant une matrice colonne pour XXX, nous explicitons que chaque observation (heure de fonctionnement du moteur) est une caractéristique unique, ce qui est nécessaire pour l'algorithme de régression linéaire.
2. **Forme des données** :
   * Les algorithmes de machine learning nécessitent souvent une distinction claire entre les observations (lignes) et les caractéristiques (colonnes). Même si dans ce cas spécifique, chaque observation n'a qu'une seule caractéristique, il est toujours nécessaire de le formater en une matrice 2D.



1. **Compatibilité avec l'apprentissage supervisé** :
   * La régression linéaire implique la multiplication matricielle entre les coefficients (poids) et les caractéristiques des données d'entrée. Pour que cette multiplication soit valide, XXX doit être une matrice 2D.
2. **Consistance des prédictions** :
   * Lorsque vous effectuez des prédictions, la méthode predict s'attend également à ce que les nouvelles données soient sous forme de matrice 2D. C'est pourquoi np.array([[700]]) est utilisé pour les nouvelles données de prédiction, garantissant que les dimensions sont compatibles avec les attentes du modèle.



Le graphique montre une relation linéaire négative entre les heures de fonctionnement du moteur (X) et la durée de vie restante (y). Chaque point bleu représente une observation des données, et la ligne rouge est la ligne de régression linéaire ajustée. La pente de la ligne indique que la durée de vie restante du moteur diminue à mesure que les heures de fonctionnement augmentent.

Ce tracé peut être utilisé pour prédire la durée de vie restante du moteur pour un nombre donné d'heures de fonctionnement qui ne figurent pas dans le jeu de données initial.

--

**Régression Logistique : Définition et Fonctionnement**

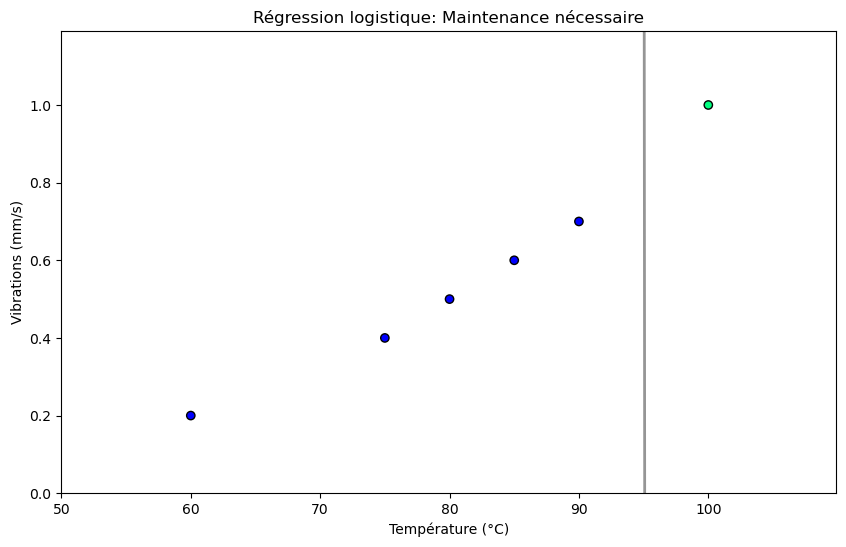
C’est un autre algorithme très courant en machine learning, particulièrement adapté à la classification binaire, où la variable cible prend deux valeurs distinctes (comme "oui" ou "non", "défectueux" ou "non défectueux").

La régression logistique est utilisée pour estimer la probabilité qu'une observation appartienne à une certaine catégorie. Au lieu de simplement ajuster une ligne droite comme en régression linéaire, la régression logistique ajuste une courbe en forme de S (fonction logistique ou sigmoïde) aux données. Cette courbe permet de prédire la probabilité de l'une des deux catégories possibles.

Une image contenant texte, Police, ligne, algèbre

Description générée automatiquement

Pour illustrer, imaginons que nous voulons classifier des moteurs d'hélicoptères comme nécessitant une maintenance ou non, basé sur les lectures des capteurs (par exemple, la température et les vibrations du moteur).



**Interprétation du Graphique**

1. **Points de Données**:
   * Les points bleus indiquent les observations où la maintenance n'est pas nécessaire (étiquetés avec 0).
   * Le point vert indique une observation où la maintenance est nécessaire (étiqueté avec 1).

 **Distribution des Points**:

* La plupart des points bleus (pas de maintenance nécessaire) sont situés à des températures plus basses et des niveaux de vibration inférieurs.
* Le point vert (maintenance nécessaire) est situé à une température élevée de 100°C et à un niveau de vibration de 1.0 mm/s, ce qui suggère que des valeurs élevées de ces deux mesures peuvent indiquer une probabilité accrue qu'une maintenance soit nécessaire.

 **Frontière de Décision**:

* La ligne verticale noire semble représenter une frontière de décision proposée par le modèle. Cependant, cette frontière apparaît comme une ligne verticale, ce qui est inhabituel pour un modèle de régression logistique traitant de deux caractéristiques. Normalement, on s'attendrait à voir une courbe ou une ligne oblique qui divise l'espace des caractéristiques en deux zones basées sur les probabilités calculées par le modèle.

-

Les coefficients du modèle indiqueront combien chaque caractéristique (température et vibrations) contribue à la décision :

* Un coefficient plus élevé pour la température signifie que la température a une influence plus significative sur la probabilité de maintenance.
* Un coefficient plus faible pour les vibrations signifie que les vibrations ont une influence moindre.

 **Distribution des Données**:

* Si la caractéristique des vibrations n'apporte pas beaucoup de variance ou ne contribue pas significativement à la décision, le modèle peut se concentrer principalement sur la température. Dans votre ensemble de données, il semble que la température ait une plus grande influence sur la décision de maintenance.

 **Fonctionnement du Modèle**:

* La régression logistique crée une frontière de décision linéaire entre les classes. Si une seule des caractéristiques est dominante (ici, la température), la frontière de décision peut sembler alignée sur cette caractéristique.

 **Balance des Données**:

* Si le modèle détecte que les changements dans la température ont un impact beaucoup plus significatif sur la nécessité de maintenance comparé aux vibrations, il peut prioriser cette caractéristique.

**Forêts Aléatoires : Définition et Fonctionnement**

les Forêts Aléatoires (Random Forests) sont une technique puissante et très populaire en machine learning pour aborder à la fois les problèmes de classification et de régression.

La méthode des forêts aléatoires est un algorithme d'apprentissage supervisé qui utilise un ensemble d'arbres de décision pour améliorer la stabilité et la précision des prédictions. Elle combine les résultats de plusieurs arbres de décision, construits à partir de sous-ensembles aléatoires de l'ensemble de données, pour produire une prédiction finale. Cette approche est particulièrement efficace pour réduire le risque de surapprentissage (overfitting) et pour améliorer la précision générale des prédictions.

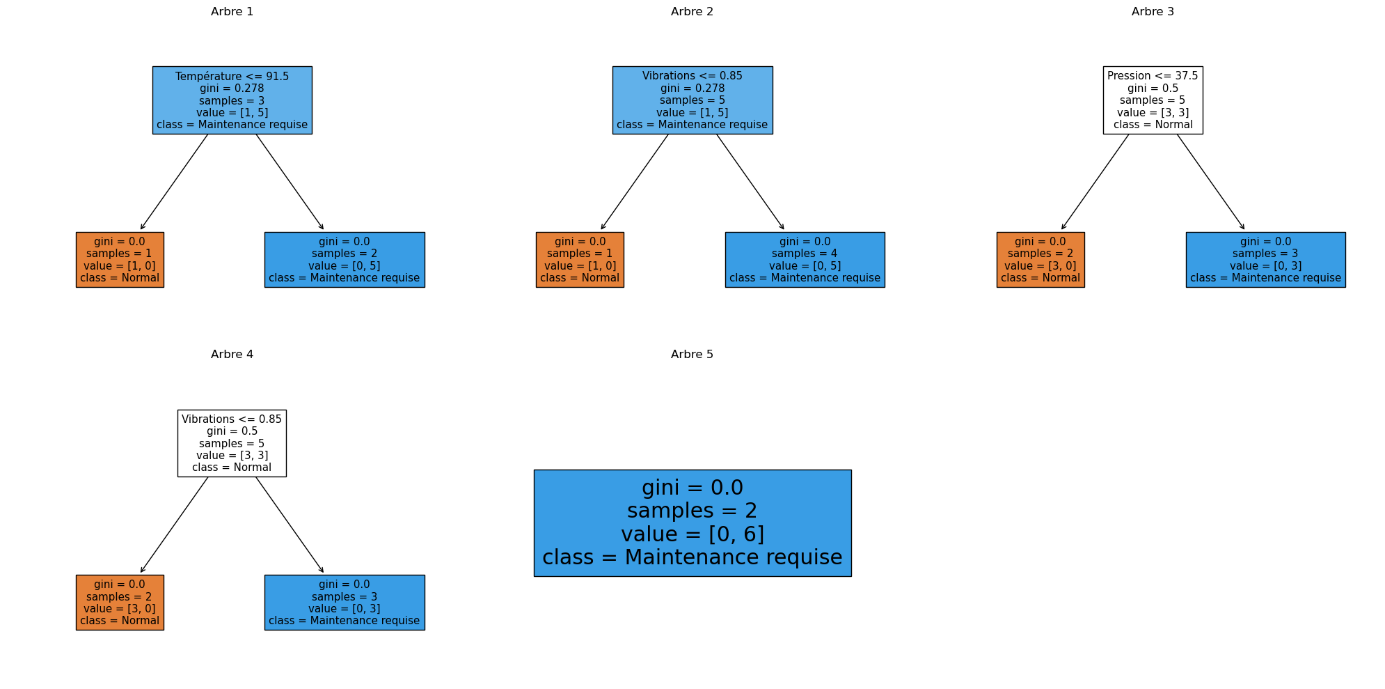
**Fonctionnement des Forêts Aléatoires**

1. **Sélection aléatoire de sous-ensembles** : À partir de l'ensemble de données original, des sous-ensembles sont créés en sélectionnant des échantillons de manière aléatoire avec remplacement (bootstrap samples).
2. **Construction d'arbres de décision** : Un arbre de décision est construit pour chaque sous-ensemble. Pour chaque nœud de l'arbre, un nombre limité de caractéristiques est sélectionné aléatoirement pour déterminer la meilleure scission.
3. **Agrégation (Bagging)** : Les prédictions de chaque arbre de décision sont ensuite agrégées pour former la prédiction finale. Pour la classification, c'est généralement le mode des prédictions (la classe la plus fréquente) ; pour la régression, c'est la moyenne

**Consensus dans les Forêts Aléatoires**

Le consensus dans une forêt aléatoire (ou random forest) est obtenu par une méthode appelée **"majority voting"** pour la classification et **"averaging"** pour la régression.

1. **Classification** : Chaque arbre de la forêt fait une prédiction de classe pour une observation donnée. La classe qui obtient le plus de "votes" (prédite par le plus grand nombre d'arbres) est choisie comme la prédiction finale de la forêt aléatoire.
2. **Régression** : Chaque arbre fait une prédiction numérique pour une observation donnée. La prédiction finale est la moyenne de toutes ces prédictions numériques.



**Remarque :** Si vous avez besoin de visualiser la logique de décision collective de la forêt, vous devriez vous concentrer sur des métriques comme l'importance des caractéristiques, les courbes ROC, ou d'autres formes d'évaluation statistique, plutôt que d'essayer de visualiser l'ensemble de la structure arborescente, ce qui n'est pas possible avec les forêts aléatoires en tant que telles.

des exemples individuels de décisions prises par des arbres de décision au sein d'une forêt aléatoire. Chaque arbre est indépendant des autres et est construit en utilisant un sous-ensemble de données aléatoires, ce qui aide à diversifier les décisions et à réduire le surapprentissage.

Les arbres montrent des décisions basées sur les caractéristiques des moteurs d'hélicoptères (température, pression, vibrations), et classifient chaque moteur soit comme 'Normal' soit comme nécessitant 'Maintenance'.

**Observation de la foret aléatoire**

**Arbre 1**

1. **Nœud Racine** :
   * **Critère** : Température <= 91.5
   * **Indice Gini** : 0.278
     + L'indice Gini mesure l'impureté d'un nœud. Un Gini de 0 indique un nœud parfaitement pur (toutes les instances sont de la même classe). Un indice plus élevé indique une plus grande mixité des classes.
   * **Échantillons** : 6
     + Nombre total d'échantillons de données passant par ce nœud.
   * **Valeur** : [1, 5]
     + Le tableau montre le nombre d'échantillons dans chaque classe; ici, 1 pour 'Normal' et 5 pour 'Maintenance requise'.
   * **Classe** : Maintenance requise
     + La classe majoritaire pour ce nœud est 'Maintenance requise', donc toute nouvelle observation qui atteint ce nœud serait classée ainsi si aucune subdivision supplémentaire n'était faite.
2. **Branches du Nœud Racine** :
   * **Gauche (Vrai)** : Tous les moteurs avec une température <= 91.5, classés comme 'Normal'. Gini = 0.0, indiquant une pureté parfaite.
   * **Droite (Faux)** : Tous les moteurs avec une température > 91.5, également classés comme 'Maintenance requise'. Gini = 0.0, indiquant aussi une pureté parfaite.

**Arbre 2**

1. **Nœud Racine** :
   * **Critère** : Vibrations <= 0.85
   * **Indice Gini** : 0.278
   * **Échantillons** : 6
   * **Valeur** : [1, 5]
   * **Classe** : Maintenance requise
2. **Branches du Nœud Racine** :
   * **Gauche (Vrai)** : Tous les moteurs avec des vibrations <= 0.85, classés comme 'Normal'. Gini = 0.0.
   * **Droite (Faux)** : Tous les moteurs avec des vibrations > 0.85, classés comme 'Maintenance requise'. Gini = 0.0.

**Comment interpréter ces informations ?**

* **Critère de Division** : Chaque nœud utilise un critère pour diviser les données. Ce critère est basé sur une caractéristique qui est la plus efficace pour réduire l'incertitude (impureté) concernant la classe cible.
* **Indice Gini** : Donne une idée de l'efficacité de la séparation effectuée par le nœud. Un indice Gini faible à un nœud signifie que la séparation a bien clarifié la classification des échantillons entre les classes.
* **Valeur** : Le nombre d'échantillons de chaque classe qui passe par ce nœud aide à comprendre la distribution des classes à ce stade de la décision.

Rmq : Dans le contexte des forêts aléatoires et plus généralement de l'apprentissage supervisé, les "feuilles" ou "nœuds terminaux" de chaque arbre de décision servent à faire des prédictions basées sur les données d'entrée. Le but ultime est d'utiliser ces prédictions pour prendre des décisions éclairées ou faire des classifications précises

Dans le contexte des forêts aléatoires et plus généralement de l'apprentissage supervisé, les "feuilles" ou "nœuds terminaux" de chaque arbre de décision servent à faire des prédictions basées sur les données d'entrée. Le but ultime est d'utiliser ces prédictions pour prendre des

Dans le cadre de notre projet sur la classification des moteurs d'hélicoptères.

**Objectif des Feuilles dans les Forêts Aléatoires**

1. **Finalisation des Prédictions** :
   * Chaque feuille dans un arbre de décision représente une décision finale basée sur les conditions (les tests dans les nœuds précédents) menant à cette feuille. Les feuilles stockent généralement la classe majoritaire parmi les échantillons de données qui atteignent cette feuille.
2. **Agrégation des Prédictions** :
   * Dans une forêt aléatoire, pour une classification, la prédiction finale pour une observation donnée est déterminée par un vote majoritaire des prédictions de toutes les feuilles correspondantes de différents arbres (chaque arbre donne un "vote" basé sur sa prédiction).

Dans votre projet, les "feuilles" de chaque arbre dans la forêt aléatoire servent à faire des prédictions précises sur l'état des moteurs d'hélicoptères, en utilisant les informations fournies par les capteurs. L'objectif est d'anticiper les besoins de maintenance avant que des problèmes graves ne surviennent, optimisant ainsi les opérations de maintenance et la durabilité des moteurs.

**Utilité des Forêts Aléatoires**

1. **Robustesse** : Les forêts aléatoires sont moins susceptibles de surapprendre par rapport aux arbres de décision individuels car elles agrègent les résultats de nombreux arbres, ce qui réduit la variance sans augmenter le biais.
2. **Précision** : En combinant plusieurs arbres, cet algorithme tend à offrir une précision de prédiction élevée, même pour des ensembles de données complexes avec de nombreuses variables.
3. **Gestion des Données Manquantes** : Les forêts aléatoires peuvent gérer les valeurs manquantes dans les données et maintenir une bonne précision même quand une partie des données est manquante.
4. **Importance des Caractéristiques** : Elles permettent de mesurer l'importance relative de chaque caractéristique dans la prédiction, ce qui peut fournir des insights précieux pour la compréhension du problème.

**Finalité de Prédiction**

* **But** : Prédire la sortie (classe pour classification, valeur pour régression) pour de nouvelles données basées sur les apprentissages des données historiques.
* **Base** : Les prédictions se fondent sur les modèles appris de l'ensemble des arbres, où chaque arbre a été entraîné sur un sous-ensemble aléatoire des données avec des caractéristiques sélectionnées aléatoirement. Cela assure diversité et généralisation.
* **Aide à la Décision** : En fournissant des prédictions précises et des mesures de l'importance des différentes variables, les forêts aléatoires aident les décideurs à comprendre les facteurs influençant les résultats et à prendre des décisions éclairées.

La décision basée sur une forêt aléatoire s'appuie généralement sur deux résultats principaux : les prédictions de classe (ou de valeur, dans le cas d'une régression) et les probabilités associées à ces prédictions. Voyons comment ces éléments sont utilisés pour prendre des décisions concrètes et pratiques, en utilisant l'exemple de la maintenance prédictive des moteurs d'hélicoptères.

**Résultats Concrets d'une Forêt Aléatoire**

1. **Prédictions de Classe** :
   * Dans un contexte de classification, la forêt aléatoire prédit la classe de chaque entrée (par exemple, "Maintenance requise" ou "Pas de maintenance requise").
   * La décision de maintenance peut être directement basée sur cette prédiction.
2. **Probabilités de Classe** :
   * Pour chaque prédiction, la forêt aléatoire peut également fournir la probabilité de chaque classe. Cela signifie combien le modèle est sûr de sa prédiction.
   * Par exemple, pour un moteur donné, la forêt aléatoire pourrait prédire "Maintenance requise" avec une probabilité de 80%.

**Utilisation des Probabilités pour la Prise de Décision**

* **Seuil de Décision** :
  + Vous pouvez définir un seuil de probabilité pour déclencher une action. Par exemple, si la probabilité que le moteur nécessite une maintenance est supérieure à 70%, alors une maintenance est programmée.
  + Ce seuil peut être ajusté en fonction de la tolérance au risque, des coûts de maintenance inutile, ou de la gravité potentielle d'une défaillance du moteur.

**Visualisation Simple et Concrète pour la Décision**

1. **Graphiques de Probabilité** :
   * Des graphiques peuvent être créés pour montrer la probabilité de maintenance nécessaire pour chaque moteur. Ces graphiques aident les gestionnaires à visualiser rapidement quels moteurs sont à risque.
2. **Tableaux de Bord** :
   * Un tableau de bord peut afficher à la fois les prédictions et les probabilités pour chaque moteur, permettant aux utilisateurs de voir facilement quels moteurs requièrent une attention immédiate.