Les types de variables

- variable qualitatife
- variable ordiannale
- variable quantitatives (disctrète thé & café)
- variable quantitatives (continue taille, poids et age)

Quand les données (quantitative) ne sont pas comparable

- Centrage et réduction => favorise les algo qui utilise la déscente de gradiant. Pour pouvoir comparer les choux et les carrote et il faut centré et réduire les variable quantitative
- Analyse en composant principale, ne peut être faîte que sur les variables quantitative. Pour faire des traitement il va devoir calculer des moyen, médianne, écrat type, mais cela n'a aucun sens mathématique sur les variable qualitative.

Quand les variables sont qualitatives

(moyen de payement - état de santé)

- Donner à la place de chaque modalité une valeur numérique. Jour de la semaine (1 - 7). Les valeurs n'ont aucun impacte, ce sont des modalité, (1 - 7) === (15 - 22).
- On peut transformer ces modalité en beans, chaque modalité devient une colonne Samedi (0 / 1)

Clustering

comparer un individu avec un autre => clustering, call non supervisé

> *Non supervisé, j'ai un jeu de données et je n'ai aucune idée de comment comparer ces individu. C'est une première étape pour classifier un jeu de données préscis.*

Classification

déterminé qu'un indiv. appartient à une class => classification

> *Ici on a des données par le métier, on a une colonne qui devra être déterminer, es ce que l'individu appartient à la class A ou B, malade pas malade, toxic, comestible.*

Régression

déterminé les caractéristique d'un individu => régression

> *La colonne a déterminer est réel avec une infinité de valeur, elle n'est pas sur deux modalité.*

Classification => variable qualitative

Régression => variable quantitative

Analyse en composant principale

C'est une transformation linéaire, il n'y a pas de déformation des axes. l'ACP calcule la valeur par laquelle on va devoir multiplier chaque axe pour réaliser cette opération de transformation linéaire.

Quand le jeu de données n'est pas linaire, ce n'est pas applicable. D'en le pourcentage d'inertie, s'il n'y a pas d'éboulie, cela signifiques que l'analyse en composant princiaple n'a pas d'impacte. Il faut observé des coudes dans l'éboulis des valeur propres.

Algorithme supervisés et non supervisé

> *e.g. donnez moi le nom d'un algorythme supervisés et dite moi comment es ce qu'il fonctionne.*

Le clustering (non supervisé)

Hiérarchique ascendent & kmeans :

On se base sur la distance en les différents individu.

Classification Hiérarchique ascendent

Aglomératif

On recherche la distance la plus petite entre les individus plus les groupes.

La hauteur des branches représente la différence entre les différents cluster.

> *Cette arbre n'a qu'un role, permettre de définir un découpage en nombre de classes. Il est donc possible de couper cette arbre à n'importe quelle hauteur et de trouver un nombre de classe. Cela sert à déterminer le nombre de class qu'il est intéressant d'étudier.*

E.g. découper toutes la France en deux (ville / météo) n'est pas pertinent.

Divisif

On fait l'inverse, on part d'un groupe d'individu avec une forte hétérogénéité et on essaye de trouver des feuilles et tous les différencier. Pour le culstering cette algorythme n'a jamais été mi en place mais pour la classification oui.

Dans le cas du cancer du seins si une des colonne de mon tableau peut séparer les individus en malade / pas malade c'est gagner.

Inconvégnient

Il est impossible d'utilisé ce type d'algorythme sur des grands jeu de données. Il serait trop long à sortir un résultat.

K-means (Lloyd)

- définir aléatoirement deux point
- calculer les distences des individus entre ces deux points
- déplacer ces points dans les baricentre de ces classes.

Il existe en deux version soit on commence en lui donnant 2 individu, soit 2 points aléatoire. L'algo s'arrête lorsqu'il n'y a plus de variation entre deux itérations, il est possible de que l'aglo n'arrivent pas à faire trois catégories et tourne en boucle.

On détermine le nombre de class en :

- on fait toutes les classifications possible
- on utilise l'aglo silhouette pour déterminer le quel est le meilleurs
    - pour chaque individu, on calcule la distance entre lui même contre tout les individu de sa class, puis de lui même contre tout les individus d'un autre groupe, on fait la moyen et on regarde pour un autre groupe.
    - es ce que même point n'était pas mieux dans la classe X plutôt que Y ?
    - Ou se trouve la distance minimal, entre les individus de son groupe ou d'un autre groupe ?

> *Es ce que la silhouette fonctionne dans tous les cas ? Non quand il y a des groupe qui entoure un autre (des cercles imbriquer les un dans les autres) la silhouette ne peut pas fonctionner, car un point à une extrémité du cercle sera forcément plus loins de l'autre extrémité du cercle que du cercle étant situé à l'intérieur de lui.*

K-means peut aussi être utilisé pour diminuer le nombre de valeur, diminuer la complexité des données. En faisant des clustering, je peut rajouter des individus qui vont être dans un cluster particulier. Je peux ainsi rajouter artificelement le nombre d'individu, je ne crée pas d'information mais on peut avoir plus de valeur ce qui permet au algo de classification de mieux se comporter par la suite.

On en fait k-means que sur les colonnes centré réduite.

DB Scan

Oon donne le rayon de la boule & combien d'indivdu on veut avoir dans cette boule pour que l'individu qui a crée cette boule fasse partie de la même classe. On a A plus 3 individu et l'algo s'arrête avec C car il n'a qu'une seul individu proche.

- A la fin il reste des individus qui ne sont dans aucun groupe, qui représente els cas extrème.
- Le calcule de re rayon n'est jamais simple, un faible variation entre le nb d'individu proche et le rayon du cercle peut donner des résultats très différents

Clustering hiérarchique ascendant

C'est la creation d'un arbre. Tout les individus sont unique et on essaye faire dse groupe de plus en plus grand jusqu'à n'avoir plus qu'un seul cluster. Si on jeu de données à N individu je peux avoir N-1 classes. La hauteur des branches entre deux groupes symbolise leurs éloignements.

Les plus proche voisin (supervisé)

On continue de faire rechercher les proximités mais on donne un nombre de voisin à trouver pour aiguillé la classification.

Apprentissage

Il faut découper notre jeu de donné en apprentissage & test pour controller les prédictions de notre algo.

La complexité permet de découper les groupes par rapport à notre jeu de donnée de manière de plus en plus préscise (une ligne ou un dessin qui englobe l'intégralité des point). En augmentant la complexité de l'ago on peut avoir une taux de réussite de 100% sur notre jeu de données.

généralisation Cependant notre modèle une fois entrainé dois rester général. Il ne doit pas servir juste à prédire notre jeu de données initial. Si la complexité est trop élevé notre algo ne sera plus général et sera utilisable que pour sont jeu d'entrainement.

Surapprentissage => faible biais, variance élevé

Sous-apprentissage => biais élevé, variance faible

L'échantillonnage

stratification => permet de s'assurer que le découpage de données soit équilibré. Il faut faire attention, car la variable a stratifier doit être qualitative.

prédiction croisé => Comment faire pour faire une prédiction à partir de 5 modèle. les algo donne deux métriques la réponse (O/1)+ la probabilité. Il faut donc additionner les probabilités données par chaque modèle.

Choix des métrique de performance

Classification (qualitatif)

Matrice de confusion

Cette matrice permet d'analyser les performances d'un modèle de classification en regardant où il fait des erreurs et en évaluant la pertinence des prédictions

Exemple d'application :

- Diagnostic médical
- Détection de spam
- Reconnaissance faciale

La courbe Roc

Définition

La courbe ROC (Receiver Operating Characteristic) est un outil permettant d'évaluer la performance d’un modèle de classification binaire.

 Pourquoi l’utiliser ?

Elle mesure la capacité du modèle à bien classifier les classes positives et négatives. Elle permet d’analyser l’impact du seuil de décision sur les performances du modèle. L'aire sous la courbe (AUC - Area Under Curve) donne un indicateur global de la qualité du modèle

Les Axes de la Courbe ROC

Axe	Explication
Axe Y (ordonnée) = Sensibilité (Recall)	Proportion de **vrais positifs détectés** par rapport à tous les positifs réels.
Axe X (abscisse) = Taux de faux positifs (1 - Spécificité)	Proportion de **faux positifs détectés** par rapport à tous les négatifs réels.

Interprétation d'un Point sur la Courbe ROC

- Chaque point de la courbe ROC correspond à un seuil de classification différent.
- Le seuil de probabilité permet de décider à partir de quelle probabilité on classe une observation comme "positive".
- Si on diminue le seuil, on détecte plus de vrais positifs mais aussi plus de faux positifs.
- Si on augmente le seuil, on réduit les faux positifs, mais on risque de manquer des vrais positifs. Exemple :

Seuil = 50% → Le modèle classe "positif" si la probabilité est ≥ 50%. Seuil = 70% → On devient plus strict, donc moins de faux positifs mais plus de faux négatifs.

L'Aire Sous la Courbe (AUC - Area Under Curve)

L'AUC (Area Under the Curve) représente la probabilité qu’un modèle classe un exemple positif avant un négatif.

- AUC = 1 → Modèle parfait (sépare parfaitement les classes).
- AUC = 0.5 → Modèle aléatoire (aucune capacité de classification).

- AUC < 0.5 → Modèle inversé (prédit le contraire de la vérité).

Regression (quantitatif)

Corrélation de Pearson (quantitatif)

Mesure la relation linéaire entre deux variables numériques.

Lorsqu'on entraîne un modèle de régression, on cherche à prédire une variable continue (ex: prix d'une maison, température, chiffre d'affaires, etc.)

La corrélation de Pearson (r) mesure le degré de linéarité entre les prédictions et les valeurs réelles :

- Si r est proche de 1 → Le modèle suit bien la tendance des données réelles.
- Si r est proche de 0 → Aucune corrélation linéaire entre les prédictions et les valeurs réelles (le modèle est mauvais).
- Si r est négatif → Les prédictions sont à l'opposé des valeurs réelles (erreur systématique).

En complément des erreurs absolues (comme RMSE et MAE), la corrélation de Pearson aide à voir si le modèle suit la bonne dynamique générale.

RSS (Residual Sum of Squares)

Le résidu rpz l'erreur pour chacune des prédictions.

C'est la somme des erreurs élevées au carré. Plus le RSS est faible, plus le modèle ajuste bien les données.

Ne permet pas de comparer des modèles entre différents jeux de données, car sa valeur dépend de la taille de l'échantillon.

MSE (Mean Squared Error)

 Comparaison des performances d'un modèle entre différents jeux de test ou algorithmes.

 Permet de choisir la meilleure optimisation ou le meilleur kernel (par exemple en SVM ou régression).

 Problème : l'unité du MSE n'est pas la même que celle de la variable cible Y.

RMSE (Root Mean Squared Error)

 Permet d'avoir une erreur interprétable car elle est dans la même unité que Y.

 Très utile pour comparer des modèles de régression et voir l'ampleur des erreurs en unités réelles.

 Ne fonctionne PAS pour des prédictions qualitatives ! (À corriger dans tes notes) → Il est exclusivement utilisé en régression et non en classification.

Kernel

Un kernel est une fonction mathématique qui transforme les données pour les rendre séparables linéairement dans un espace de plus haute dimension.

 Utilisation principale : Il permet d'utiliser des algorithmes linéaires (comme SVM ou régression) sur des données non linéaires, en évitant de calculer explicitement toutes les dimensions.

⚠ Si on choisit le mauvais kernel, la transformation peut rendre les données encore plus complexes, ce qui dégrade la performance du modèle.

 Cela permet d'économiser du temps de calcul et de conserver des algorithmes optimisés tout en traitant des problèmes complexes.

Concept	Explication
Définition	Le kernel transforme les données pour les rendre **séparables linéairement**
Pourquoi ?	Quand les données sont **non linéaires**, il permet d'utiliser un **modèle linéaire** dans un **espace transformé**
Exemples de Kernel	**Linéaire** (pas de transformation) **Polynomial** (tordre l'espace en puissance) **Radial (RBF)** (séparer des données circulaires)
Avantage	Évite de **calculer toutes les dimensions**, permet d'utiliser des **algorithmes linéaires** sur des **données complexes**

La régression linéaire

Définition

La régression est une méthode de prédiction utilisée en Machine Learning pour estimer une valeur numérique en fonction d'autres variables.

 Objectif : Trouver une relation entre une ou plusieurs variables d'entrée (features) et une variable cible.

Quand utiliser la régression linéaire ?

Cas d'utilisation	Explication
Prédire une valeur continue	Prix d'une maison, salaire, température.
Relation linéaire entre les variables	Si augmenter X augmente Y de manière proportionnelle.
Peu de données, besoin d'un modèle simple	Modèle rapide et interprétable.
Comprendre quelles variables influencent Y	Identifier les variables importantes avant d'utiliser un modèle plus complexe.

 Elle optimise les autres algorithmes en aidant à :

```
✔ Sélectionner les meilleures variables.
✔ Tester si une relation linéaire existe avant d'appliquer un modèle complexe.
✔ Servir de base mathématique pour d'autres modèles (logistique, SVM)
```

La fonction de coût

La fonction de coût est un outil utilisé en Machine Learning pour mesurer l'erreur d'un modèle.

 Elle sert à évaluer à quel point les prédictions du modèle sont proches ou éloignées des vraies valeurs.

```
1 Le modèle fait une prédiction.
2 On compare cette prédiction avec la valeur réelle.
3 La fonction de coût attribue une "pénalité" en fonction de l'écart entre les deux.
4 Le modèle ajuste ses paramètres pour réduire cette erreur et améliorer ses futures prédictions.
```

Concept	Explication
Définition	Mesure l'erreur entre la prédiction du modèle et la valeur réelle.

Concept	Explication
Utilité	Permet d’entraîner le modèle et d’améliorer ses performances.
Fonctionnement	Compare la prédiction avec la vraie valeur et attribue une pénalité (erreur).
Optimisation	Les algorithmes comme la **descente de gradient** minimisent cette erreur.
Types	Régression : MSE (erreur quadratique) / Classification : Log-Loss (entropie croisée).

La régression logistique

La régression logistique est un algorithme de classification qui permet de prédire une catégorie à partir de données. Contrairement à la régression linéaire (qui prédit une valeur continue), la régression logistique prédit une probabilité et attribue une classe (ex: "Oui" ou "Non", "Spam" ou "Non-Spam", etc.).

 Exemples concrets :

```
✔ Détecter un spam : Est-ce qu’un email est "Spam" ou "Non Spam" ?

✔ Diagnostic médical : Un patient est-il malade (1) ou en bonne santé (0) ?

✔ Prédiction d’achat : Un utilisateur va-t-il acheter un produit (Oui/Non) ?
```

 Avantages :

```
✔ Simple à comprendre et rapide à entraîner.
✔ Fournit une probabilité, utile pour ajuster les seuils.
✔ Fonctionne bien sur des petits jeux de données.
```

 Limites :

```
 Ne fonctionne que pour deux catégories (mais il existe des extensions pour plusieurs classes).
 Suppose que les données sont bien séparables (si ce n’est pas le cas, un SVM ou un réseau de neurones est plus adapté).
```

l'arbre de déscision

Cette algo control colone par colone la capacité de chaque colone à séparer le jeu de données vis à vis de la colone à définir (e.g. espèce).

On décrit jusqu'à quel niveau on va aller en terme de profondeur, je ne veut pas que mon arbre aille au dela de deux niveau de profondeur.

prb :

- il fait trop dé coupage à profondeur max pour ne trouver qu'un exemple à la fin
- A chaque étape de l'analyse l'algo ne prend qu'une colone à la fois, les informations de sont pas brasser à l'intérieur d'une seul colonne. En traitant variable après variable, il est problématique de transformer ses données en modalité pour les données qualitative.

Réduire le sur apprentissage :

- réduire la profondeur de l'arbre
- augmenter le nombre d'arbre
  - (tree bagging / random forest)
  - random forest crée des arbres en parallèle (c'est rapide)
  - boosting création des arbres en série => plus lent mais plus pertinent
  - vecteur machine

Présentation

2/3 min

- Expliquer les grandes ligne de la recherche
- Les étapes traversé
- Dire si on a validé / invalidé notre hypothèse

(avoir un plan de ce que l'on va présenter)

Ne pas montrer tout les graphiques, parler de l'analyse, les axes de recherche, nos conclusion. Montrer quelques graph intermédiaire.

Révision

typologie d'algorithme

Non supervisé

Quelle sont les ama d'individu qui vont ensemble.

- Il s'utilise quand on a des jeux de données que l'on veut libelé, mettre en cluster pour faire de l'apprentissage supervisé par la suite.
- Quand on veut connaître une variable qui n'est pas binaire (prix de l'appart)
- Je suis un opérateur internet, je regarde ce que mes clients achète, je propose des publicités ciblé en rapprochant deux individu.

Une fois que j'ai rapprocher / libélé ces informations je peux transporté ce jeu de données dans un algorythme supervisé et rapidement déterminé sa classe.

Supervisé

Il y a une colonne X ou Y a expliquer.

Il y deux type d'algo :

- **Régression** Y est *quantitatif*
- **Classification** binaire Y est *qualitatif*

Une modalité c'est la représentation de toute les valeur possible d'une valeur qualitative par Vrai / Faux. En somme on crée une colonne par valeur possible (vert / jaune / rouge) et chaque colonne répond par Vrai ou Faux. Chaque colonne est une modalité.