
# Apprentissage Supervisé

UQAC | Université du Québec à Chicoutimi

# Étape à suivre

- Nettoyage & transformation des données
- Division du jeu de données (Entraînement, Test)
- Développement de modèles
- Optimisation des hyperparamètres
- Mesure de la performance



UQAC

# Nettoyage & transformation des données

- Bon format des données (Catégorielle, Date, String etc)
- Valeurs manquantes
- Valeurs extrêmes
- Distribution des données
- Création/Supression de variable
- Réduction de la dimensionalité
- Standardiser les données



UQAC

# Nettoyage & transformation des données

Encodage des variables catégorielles

```
data = [
    {'price': 850000, 'rooms': 4, 'neighborhood': 'Queen Anne'},
    {'price': 700000, 'rooms': 3, 'neighborhood': 'Fremont'},
    {'price': 650000, 'rooms': 3, 'neighborhood': 'Wallingford'},
    {'price': 600000, 'rooms': 2, 'neighborhood': 'Fremont'}
]
```

```
{'Queen Anne': 1, 'Fremont': 2, 'Wallingford': 3};
```

*Queen Anne < Fremont < Wallingford ?*

*Wallingford - Queen Anne = Fremont ?*



UQAC

# Nettoyage & transformation des données

Encodage des variables catégorielles

```
data = [
    {'price': 850000, 'rooms': 4, 'neighborhood': 'Queen Anne'},
    {'price': 700000, 'rooms': 3, 'neighborhood': 'Fremont'},
    {'price': 650000, 'rooms': 3, 'neighborhood': 'Wallingford'},
    {'price': 600000, 'rooms': 2, 'neighborhood': 'Fremont'}
]
```

```
from sklearn.feature_extraction import DictVectorizer
vec = DictVectorizer(sparse=False, dtype=int)
vec.fit_transform(data)
```

```
array([[     0,      1,      0, 850000,      4],
       [     1,      0,      0, 700000,      3],
       [     0,      0,      1, 650000,      3],
       [     1,      0,      0, 600000,      2]], dtype=int64)
```



UQAC

# Nettoyage & transformation des données

Encodage des variables catégorielles

Avec cette méthode, on crée autant de nouvelles variables binaires qu'il y a de catégories dans chaque variable catégorielle. Chaque nouvelle variable créée indique par le chiffre 1 la présence de la catégorie et par le chiffre 0 l'absence de la catégorie. Il existe plusieurs manières de faire du One Hot Encoding.

Suivre ces liens pour plus de méthodes:
https://pandas.pydata.org/docs/reference/api/pandas.get_dummies.html
https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html)



UQAC

# Réduction de la dimensionalité

Au cours de l'apprentissage, un modèle d'apprentissage automatique peut être confronté à des milliers, voir des millions de variables. Le nombre de ces variables peuvent rendre l'apprentissage très lent, et même nous rendre la tâche ardue dans la recherche d'une bonne solution. Dans le jargon du *Machine Learning*, on appelle ce genre de problème : **la malédiction de la dimensionnalité** (*curse of dimensionality* en anglais).

Outre la diminution du temps d'apprentissage de l'algorithme, **réduire la dimensionnalité est également extrêmement utile pour la visualisation des données**. En effet, en réduisant le nombre de dimensions à deux ou trois, il est souvent possible de repérer visuellement des informations importantes, comme par exemple des données qui se regroupent en grappes.

**La visualisation des données est aussi essentielle pour communiquer vos conclusions à des personnes externes à la science des données**, comme par exemple des gestionnaires qui prendront des décisions sur vos résultats.

# Analyse par composante principale

- Crée des nouvelles variables qui sont des combinaisons linéaires des variables originales.
- Ces nouvelles variables sont non corrélées entre elles
- Maximise la variance



UQAC

# Analyse par composante principale

Fonctionnement:

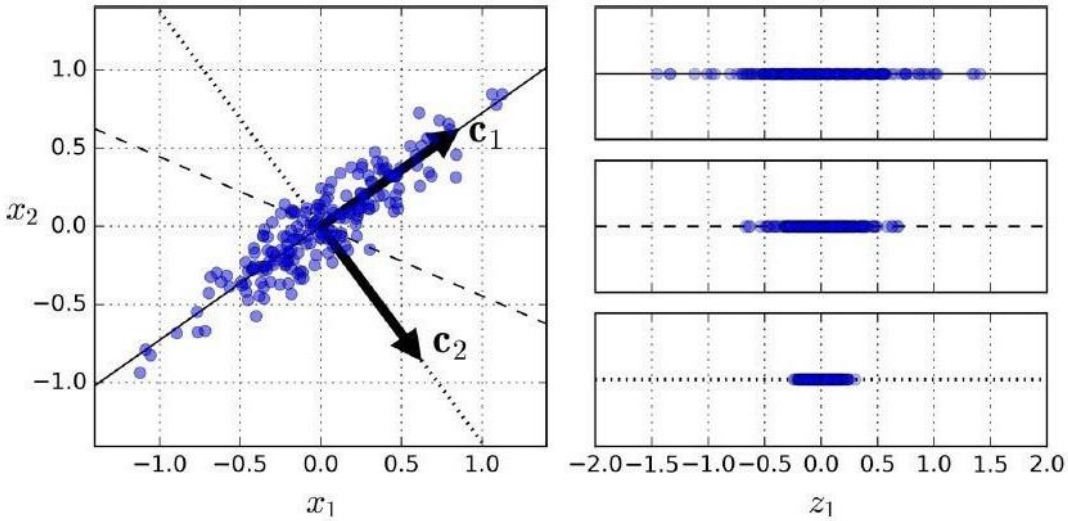
Avant de pouvoir **projeter** les données d'entrainements sur un hyperplan de dimension inférieure, on doit choisir l'hyperplan qui minimise la perte d'informations, c'est à dire qui maximise la variance.

L'ACP identifie l'axe qui représente la plus grande quantité de variance dans le jeu de données d'entrainement. Dans la figure ci-dessous, il s'agit de la **ligne continue**. Elle trouve également un deuxième axe, **orthogonal au premier**, qui représente la plus grande quantité de variance restante.

L'ACP trouve également un troisième axe, orthogonal aux deux axes précédents, puis un quatrième, un cinquième, etc jusqu’à ce que la variance totale du jeu de données soit capturée

# Analyse par composante principale

Exemple en 2D






UQAC

# Analyse par composante principale

**Interprétation**

Chaque composante est une combinaison linéaire des variables

$$Y_i = \alpha_{i,1} \times var_1 + \alpha_{i,2} \times var_2 + \ ...$$

Où:

- $Y_i$ est la iême composante
- $\alpha_{i,1}$ est le poids de la variable sur cette composante
- var est la variable



UQAC

# Analyse par composante principale

**Interprétation**

Chaque composante explique un certain pourcentage de la variation observée dans le jeu de donnée.

La proportion de variation expliquée la composante principale $Y_i$ est

$$\frac{\lambda_i}{\lambda_1 + \lambda_2 + \lambda_3 + \cdots}$$

Où $\lambda_i$ est la valeur propre du vecteur



UQAC

# Analyse par composante principale

**Interprétation**

Chaque observation du jeu de données contribue à la variabilité présente dans chaque composante. On peut mesurer la contribution de l'observation i à la composante k de la façon suivante:

$$C_{i,k}^{obs} = \frac{Y_{i,k}^2/n}{\lambda_k}$$

Où

- $Y_{i,k}^2$ est la coordonnée de l'observation i sur la composante k
- n est le nombre d'observation du jeu de données
- $\lambda_k$ est la valeur propre de la composante k



UQAC

# Analyse par composante principale

**Interprétation**

On peut mesurer la contribution de la variable j à la composante k de la façon suivante:

$$C_{j,k}^{var} = \frac{r_{j,k}^2}{\lambda_k}$$

Où

- $r_{j,k}^2$ est la coordonnée de la variable j sur l'axe de la composante k
  - C'est la corrélation entre la variable j et la composante k
- $\lambda_k$ est la valeur propre de la composante k

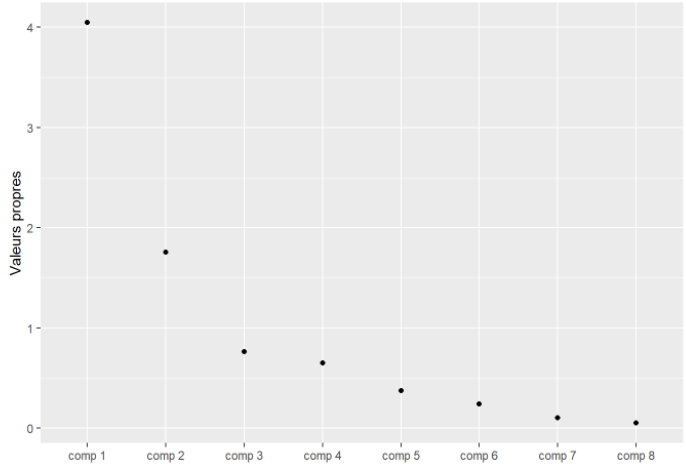


UQAC

# Analyse par composante principale

**Choix du nombre de composante**

- La règle du 80%: On garde les composantes qui représentent 80% de la variance
- La règle de Cattell






UQAC

# Division du jeu de données

On divise le jeu de données en un jeu d'entraînement et un jeu de test pour avoir une mesure honnête de la performance du modèle.

En effet, on entraîne notre modèle sur notre jeu de données d'entrainement et une fois que le modèle est prêt, ce dernier est utilisé sur le jeu de données test (qui n'a jamais été vu par le modèle) pour prédire la variable réponse et la comparer à la vraie valeur.

Le fait de tester notre modèle sur un jeu de données non-utilisé nous permet d'avoir une mesure réaliste de la performance.

**Généralement, on entraîne nos modèles sur 75% des données et on test sur 25% des données**

# Division du jeu de données

## Validation croisée

Plutôt que de séparer l'échantillon en un échantillon d'entraînement et un échantillon de test, on effectue cette procédure:

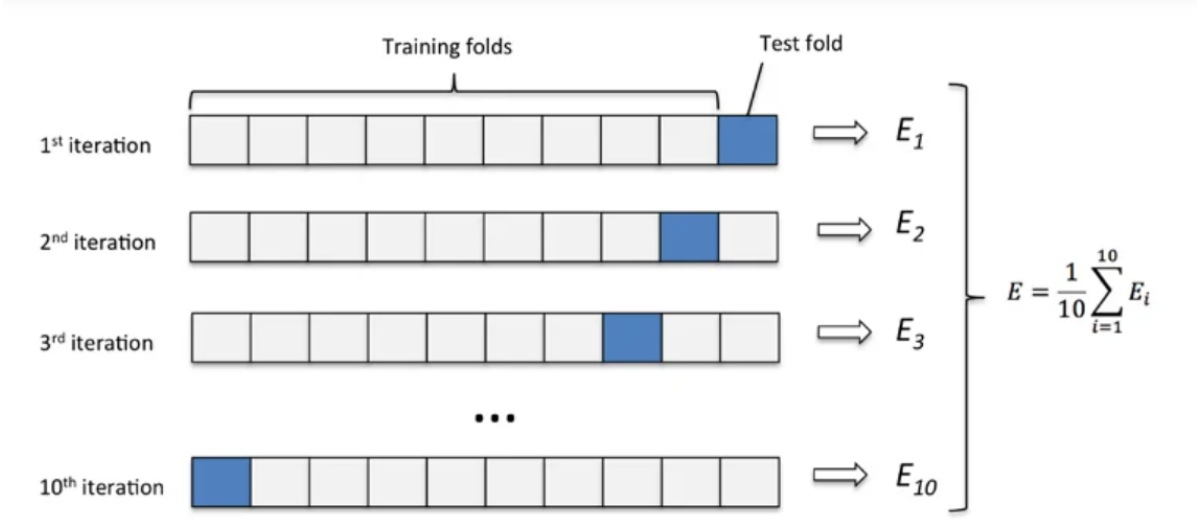
1. On sépare l'échantillon en K groupes de façon aléatoire
2. On entraîne un modèle en excluant le premier groupe
3. On calcule la performance du modèle sur le premier groupe

On répète les étapes 2 et 3 avec les K groupes.

On fait la moyenne des performances, on choisit le modèle le plus performant.

# Division du jeu de données

## Validation croisée






UQAC

# Division du jeu de données

**Compromis Biais-Variance**

L'inabilité d'un modèle prédictif a capturer la vraie relation dans les données est appellée le **biais**. Un modèle avec un fort biais aura une forte erreur lors de l'entraînement.

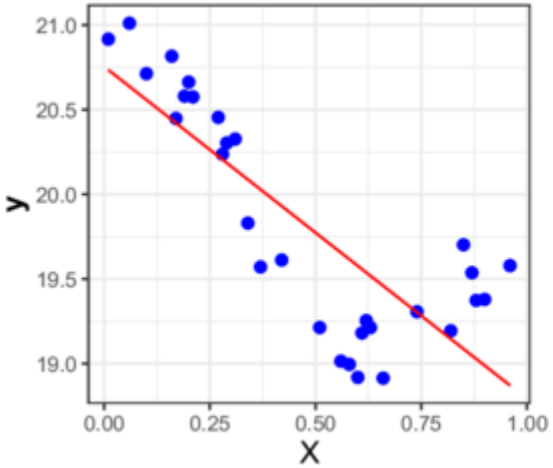
Cependant, si on minimise l'erreur d'entraînement on peut se retrouver dans le cas de surapprentissage.

L'utilisation d'un jeu de test nous permet de vérifier à quel point le modèle va être efficace lors de la généralisation de son apprentissage après entrainement. Le but est donc de minimiser l'erreur de généralisation.

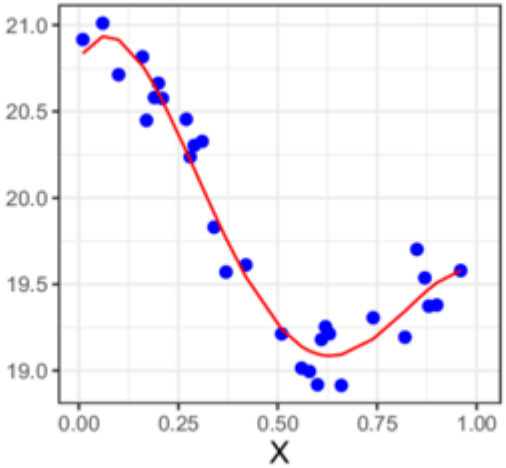
La différence d'adaptation entre les jeu de données est appelé **variance** (avant et après avoir testé la capacité de généralisation du modèle sur le jeu de test)

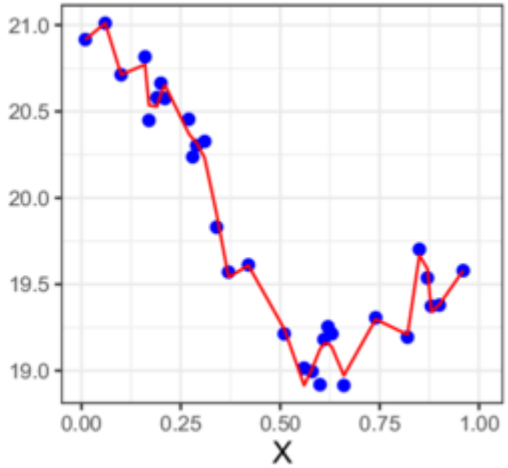
# Division du jeu de données

## Compromis Biais-Variance




Fort biais & variance faible




Faible biais & variance forte



UQAC

# Développement de modèles

- Une catégorisation utile des **algorithmes d'apprentissage automatique supervisé** est obtenue en faisant une distinction par rapport au type - **quantitatif ou qualitatif** - de la variable d'intérêt (y) que l'on cherche à prédire.
- Voyons ensemble quelques exemples :

| Type de la variable | Exemple | Doit être géré comme |
|---|---|---|
| Numérique continue | 10 secondes, 1 heure et 15 minutes, 2.54 années ... | Quantitatif |
| Numérique discret avec ordre naturel | 0 chambre, 1 chambre, 2 chambres ... | Quantitatif, Ordinal ou Qualitatif |
| Numérique discret sans ordre naturel | 1 = maison, 2 = duplex, 3 = maison de ville ... | Qualitatif |
| Texte non numérique | Biggin, Nelson, Greg, Collins ... | Qualitatif |



UQAC

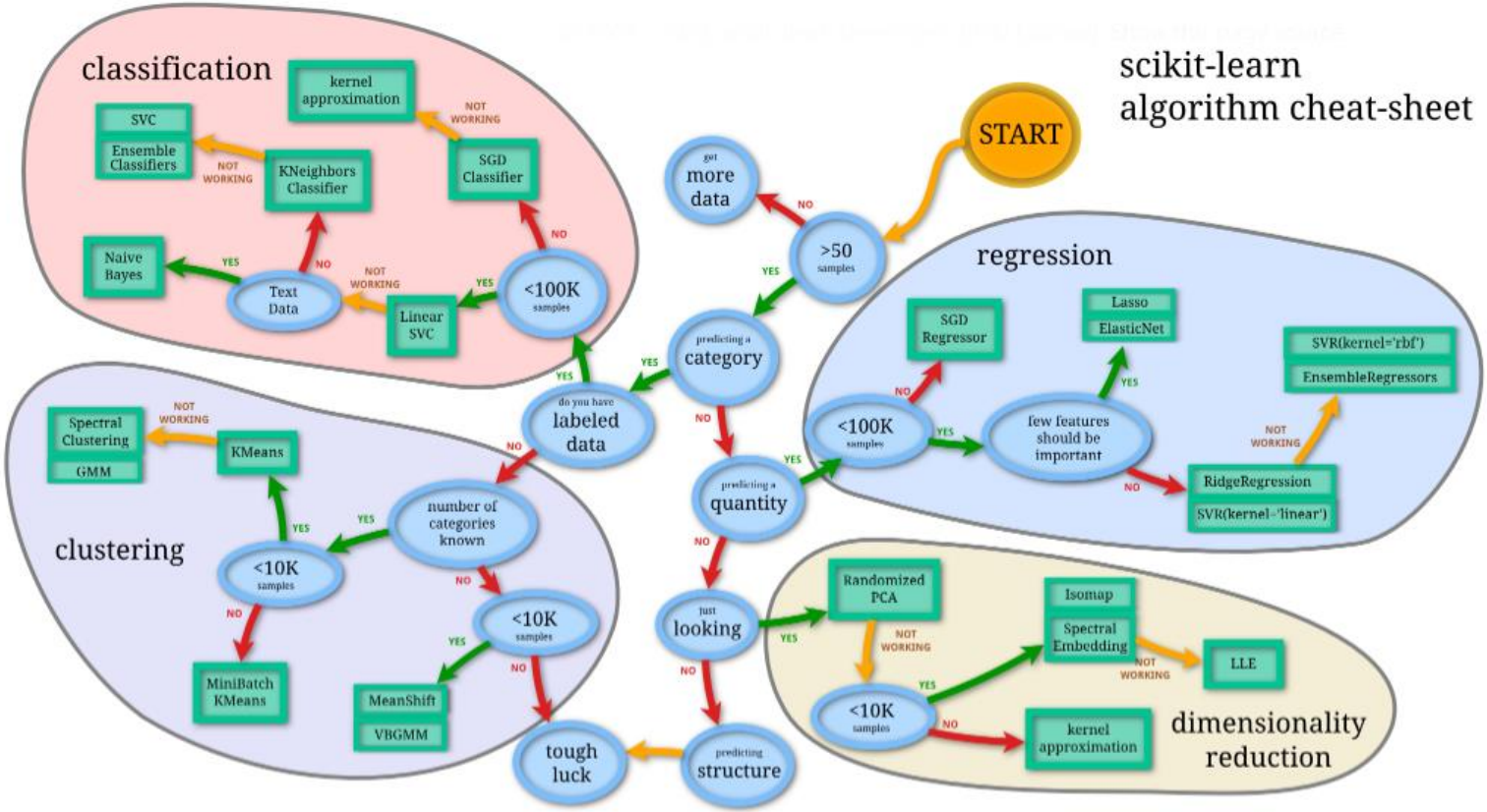
# Développement de modèles

- **La régression signifie qu'on prédit une valeur numérique alors que la classification signifie qu'on prédit une catégorie**
- Cependant, il faut être prudent. En effet, la distinction entre qualitatif et quantitatif est parfois arbitraire et il n'y a pas toujours de réponse claire.
- Par exemple, on pourrait modifier la variable qui compte le nombre de chambre (0,1,2,... : **Numérique discret avec ordre naturel**) en une variable qui indique si la maison possède oui ou non des chambres (0 pour non et 1 pour oui : **Numérique discret sans ordre naturel**). On transformerait ainsi le problème de régression en un problème de classification.



UQAC

# Développement de modèles




https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html



UQAC

# Développement de modèles

**Scikit-learn**

Un des avantages de cette librairies est la syntaxe commune aux différents modèles!

On compte 4 étapes principales:

1. On instancie le modèle
    - *ex : reg_LR = LinearRegression()*

2. On entraine le modèle grâce à la fonction "fit" qui permet au modèle de s'ajuster sur les données
    - *ex: reg_LR.fit(train_copie_X, y_train)*

3. On prédit sur l'ensemble de données test
    - *ex: reg_LR_preds= reg_LR.predict(test_copie_X)*

4. On vérifie la performance du modèle sur le jeu de données test en calculant une métrique
    - *ex: reg_LR_MAE = mean_absolute_error(reg_LR_preds, y_test)*

# Développement de modèles

**Forêt aléatoire (random forest)**

Cet algorithme est un exemple d'une méthode d'ensemble basée sur les arbres de décision.

L'idée fondamentale derrière les arbres de régression et de classification est de diviser progressivement l'ensemble des données d'entraînement en utilisant des variables discriminantes.

Voici comment l'algorithme procède:

1. Commencer avec toutes les observations à la racine de l'arbre.
2. Sélectionner une variable Xi et diviser les données en deux (ou plusieurs) nœuds en fonction de cette variable, de manière à rendre les valeurs de Y aussi homogènes que possible dans chaque nœud.
3. Répéter l'étape 2 pour chacun des nœuds créés.
4. Continuer ce processus jusqu'à ce qu'un critère d'arrêt prédéfini soit atteint.

Le modèle final est constitué d'une série de règles simples découlant de ces divisions successives.

# Arbre de décision

Exemple: On veut prédire la variable achat (oui=1 ou non=0)

| âge | revenu | propriétaire | achat |
|---|---|---|---|
| <=35 | élevé | non | non |
| <=35 | élevé | non | non |
| 36-45 | élevé | non | oui |
| >45 | moyen | non | oui |
| >45 | faible | oui | oui |
| >45 | faible | oui | non |
| 36-45 | faible | oui | oui |
| <=35 | moyen | non | non |
| <=35 | faible | oui | oui |
| >45 | moyen | oui | oui |

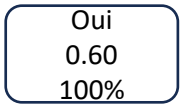
Noeud initial (toutes les données):
60% ont acheté et 40% non



UQAC

# Arbre de décision

Exemple: On veut prédire la variable achat (oui=1 ou non=0)




Ainsi, la valeur prédite à la racine de l'arbre serait «oui» et le taux d'erreur à la racine serait de 0,4 et l'exactitude serait de 0,6.



UQAC

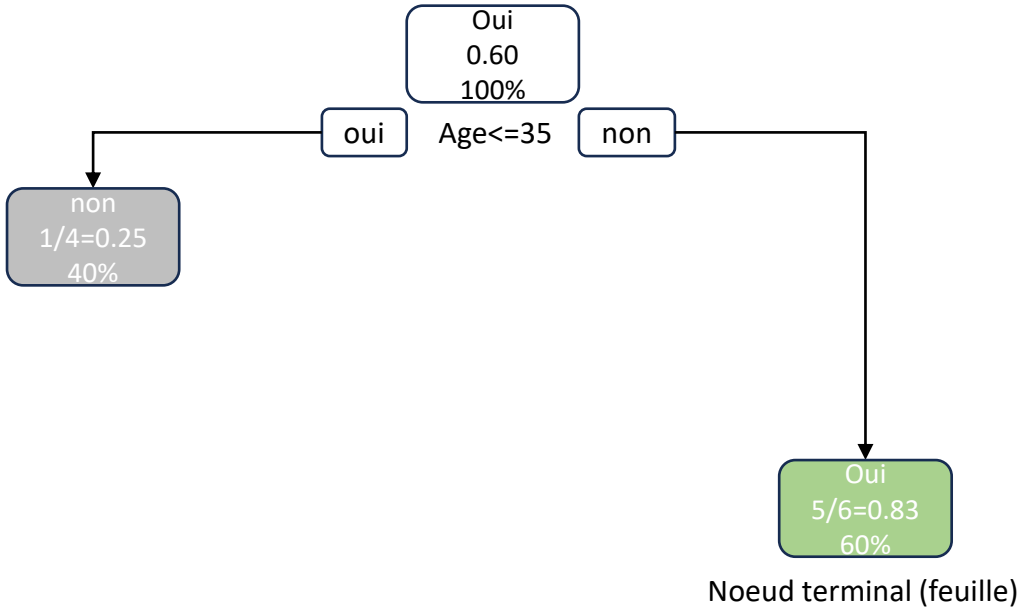
# Arbre de décision

L’algorithme choisi la variable âge pour créer les 2 prochains noeud

| âge | revenu | propriétaire | achat |
|---|---|---|---|
| <=35 | élevé | non | non |
| <=35 | élevé | non | non |
| 36-45 | élevé | non | oui |
| >45 | moyen | non | oui |
| >45 | faible | oui | oui |
| >45 | faible | oui | non |
| 36-45 | faible | oui | oui |
| <=35 | moyen | non | non |
| <=35 | faible | oui | oui |
| >45 | moyen | oui | oui |

# Arbre de décision

Exemple: On veut prédire la variable achat (oui=1 ou non=0)






UQAC

# Arbre de décision

L’algorithme choisi la variable revenu pour créer les 2 prochains noeud

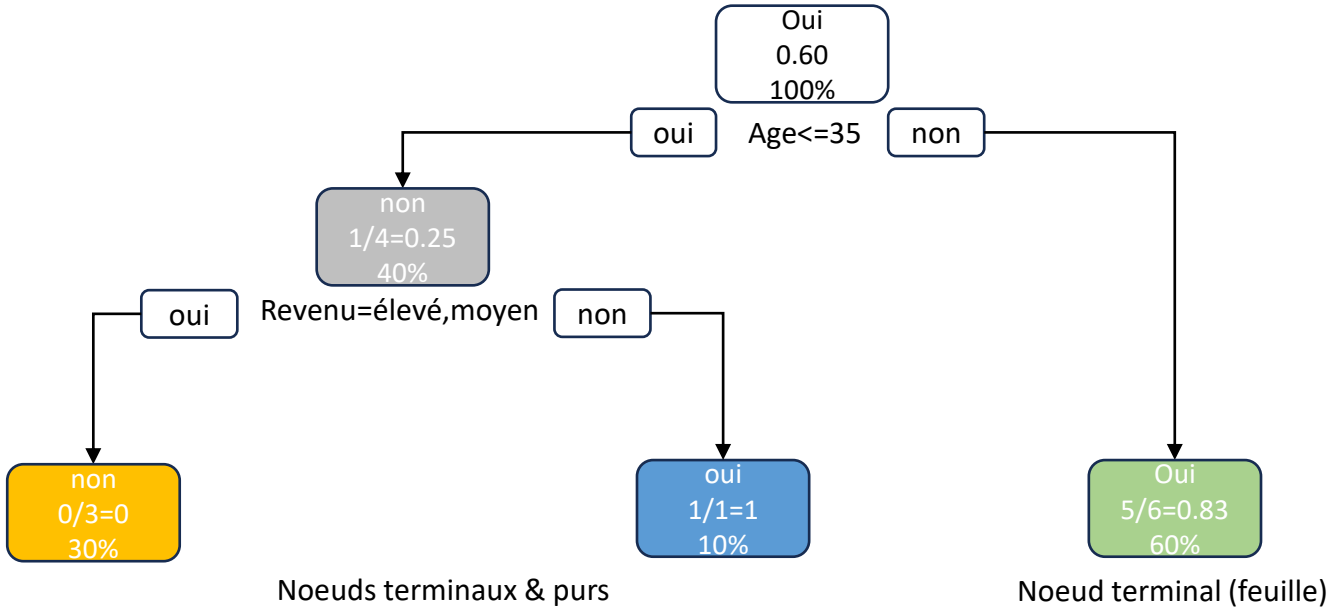
| âge | revenu | propriétaire | achat |
|---|---|---|---|
| <=35 | élevé | non | non |
| <=35 | élevé | non | non |
| 36-45 | élevé | non | oui |
| >45 | moyen | non | oui |
| >45 | faible | oui | oui |
| >45 | faible | oui | non |
| 36-45 | faible | oui | oui |
| <=35 | moyen | non | non |
| <=35 | faible | oui | oui |
| >45 | moyen | oui | oui |



UQAC

# Arbre de décision

Exemple: On veut prédire la variable achat (oui=1 ou non=0)






UQAC

# Arbre de décision

**Choix de la variable de séparation des noeuds**

Approche de type CART

Dans le cas binaire, un des critères de sélection le plus populaire à l'heure actuelle est l'indice de Gini.

On le décrit comme un indice d'impureté d'un noeud.

Dans le cas binaire, par exemple, l'indice de Gini sera maximal lorsqu'un noeud contient le même nombre d'observations dans les deux classes.

Le critère sera minimal si toutes les observations du noeud prennent la même valeur (obtenir un noeud pur).

# Arbre de décision

**Choix de la variable de séparation des noeuds**

Approche de type CART

Les algorithmes qui utilisent l'indice de Gini procèdent habituellement ainsi:

1. On commence avec toutes les observations
2. Pour chaque variable Xi, on sépare les observations en deux selon les différentes valeurs que peut prendre Xi.
3. On calcule l'indice de Gini des noeuds fils pour chaque séparation possible des Xi
4. On choisit la séparation qui fait le plus diminuer de l'indice de Gini
5. On répète les étapes 2 à 4 pour chaque noeud fils jusqu'à l'atteinte d'un critère d'arrêt.

# Arbre de décision

**Choix de la variable de séparation des noeuds**

Approche de type CART

Dans le cas où on cherche à prédire une variable continue, on utilise l'erreur quadratique moyenne sur les observations du noeud que l'on souhaite séparer.



UQAC

# Arbre de décision

**Choix de la variable de séparation des noeuds**

L'approche conditionnelle

Cela consiste à choisir d'abord la variable de séparation, puis à choisir dans un deuxième temps la façon de combiner les valeurs de  Xi pour obtenir la meilleure séparation possible.

Pour des variables catégorielle, on effectue un test d'indépendance du $\chi 2$ entre la variable à prédire Y et chacune des variable Xi et on choisit la variable pour laquelle la p-valeur est la plus faible. C'est ce que fait l'algorithme CHAID (Chi-squarred automatic interaction detection)

Dans le cas de deux variables continues, on pourrait utiliser la corrélation entre Xi et Y.

# Arbre de décision

**Critères d'arrêt**

Voici quelques critères d'arrêt:

1. L'effectif minimal dans chaque noeud terminal
2. L'effectif minimal pour séparer un noeud
3. La hauteur de l'arbre

# Arbre de décision

**Avantages et inconvénients des arbres**

Les principaux avantages des arbres de régression sont:

1.Robuste aux données extrêmes
2.Facile à interpréter
3.Sélectionne implicitement les variables importantes
4.Permet d'obtenir un modèle non linéaire

Les inconvénients sont les suivants:

1.L'approche n'est pas particulièrement performante en termes de prédiction
2.Les risques de surapprentissage sont élevés
3.Les algorithmes de construction des arbres nécessitent un temps d'exécution élevé

# Les méthodes d'ensemble

Une moyenne de valeurs prédites par différents modèles est souvent plus précise et plus stable que la valeur prédite à l'aide d'un seul modèle.

Une méthode d'ensemble est un ensemble de modèles dont les prédictions sont combinées d'une certaine manière afin d'obtenir une prédiction plus stable.

Voici différentes façons de combiner les données:

- Variable cible continue:
  - moyenne des valeurs prédites par chaque modèle.
- Variable cible de type nominale:
  - moyenne des probabilités a prédites de chaque modèle pour chaque classe de la variable cible
  - classification dans la classe de la variable cible pour laquelle le plus grand nombre de modèles classifie l'observation (vote le plus populaire!).

# Les forêts aléatoires

Voici la procédure que suit l'algorithme:

1. On sélectionne (généralement avec remise) un échantillon des données de taille c
2. On construit un arbre b de la façon suivante:
   i. On part, à la racine, avec toutes les observations de l'échantillon sélectionné en 1.
   ii. On sélectionne d variables parmi l'ensemble des variables disponibles.
   iii. Pour chacune des d variables sélectionnées en ii, on calcule un critère de séparation pour chacune des divisions (ou combinaisons de modalités) possibles.
   iv. On choisit la variable et la division qui fait le plus descendre le critère choisi et on crée deux nouveaux noeuds sur cette base.
   v. Pour chacun des deux noeuds créés, on répète les étapes ii à iv jusqu'à l'atteinte d'un certain critère d'arrêt.
3. On calcule une prévision pour chaque observation
4. On répète les étapes 1 à 3 B fois
5. On agrège les valeurs des B arbres

# Optimisation des hyperparamètres

Voici quelques hyperparamètres que l'on peut optimiser afin d'obtenir le meilleur modèle possible avec les forêts aléatoires:

- Nombre d'arbres
- Critère de séparation
- Hauteur des arbres
- Effectif minimal dans chaque noeud terminal
- Effectif minimal pour séparer un noeud

# Optimisation des hyperparamètres

**Recherche par cadrillage (Grid search)**

Il est possible d'essayer de trouver de meilleurs hyperparamètres manuellement, en les essayant successivement. Cependant, essayer de nombreuses combinaisons serait très vite chronophage.

Une solution s'offrant à nous est d'utiliser une grille de paramètres qui sera parcourue automatiquement.

La librairie scikit-learn propose des fonctions qui permettent de parcourir une grille de paramètres et de trouver la combinaison qui minimise l’erreur de généralisation.

# Mesure de performances

Les critères de performance doivent tous être calculés sur l'échantillon de validation pour éviter le problème de surajustement.

La mesure de performance dépend du type de la variable que l'on cherche à prédire:

- Continue
- Binaire



UQAC

# Mesure de performances

## Valeur continue

On choisit généralement l'erreur quadratique moyenne comme critère de performance.

$$EQM = \sum (y_i - y_i')^2$$

Où

- $y_i$ est la vraie valeur pour l'observation i des données de test
- $y_i'$ est la prédiction obtenue par le modèle pour cet observation



UQAC

# Mesure de performances

## Valeur binaire

Il existe beaucoup de critères pour évaluer la performance d'un modèle avec une variable réponse binaire.

Voici ceux que nous traiterons:

- Le taux de bonne classification (et le taux d'erreur)
- La précision
- La sensibilité (rappel, recall)
- Spécificité
- Score F1
- La courbe ROC et l'aire sous la courbe ROC

# Mesure de performances

## Valeur binaire

En général, les modèles donnent une probabilité pi que la réponse soit 1 pour chaque observation i. On obtient une prévision en fixant un seuil s et on prédit en fonction ce celui-ci.

$$y'_i = 1 \; si \; p_i \geq s$$
$$y'_i = 0 \; si \; p_i \leq s$$

Généralement, s est égal à 0.5.



UQAC

# Mesure de performances

## Matrice de confusion

| | $y'_i = \mathbf{0}$ | $y'_i = \mathbf{1}$ |
|---|---|---|
| $y_i = 0$ | Vrais négatifs (VN) | Faux positifs (FP) |
| $y_i = 1$ | Faux négatifs (FN) | Vrai positif (VP) |



UQAC

# Mesure de performances

## Le taux de bonne classification (accuracy)

Le taux de bonne classification est la proportion des observations qui sont bien classées.

$$\frac{VN+VP}{VP+VN+FP+FN}$$

## Le taux d'erreur

Le taux d'erreur est la proportion des observations mal classées.

$$1 - \frac{VN+VP}{VP+VN+FP+FN} \qquad \frac{FP+FN}{N}$$



UQAC

# Mesure de performances

## La précision

La précision est la proportion de prévisions positives qui sont réellement positives.

$$\frac{VP}{VP+FP}$$

**On veut éviter les faux positifs**.



UQAC

# Mesure de performances

**Sensibilité (rappel, recall)**
C’est la proportion d’observations positives détectées par le modèle.

$$\frac{VP}{VP+FN}$$

**On veut éviter les faux négatifs**.



UQAC

# Mesure de performances

**Spécificité**

C’est la proportion d’observations négatives détectées par le modèle.

$$\frac{VN}{VN+FP}$$

# Mesure de performances

## Score F1

Les valeurs de précision, de sensibilité et de spécificité ne peuvent servir de mesure pour choisir le meilleur modèle, car on obtient des valeurs minimales ou maximales avec des modèles inutiles. Par exemple, on peut maximiser la sensibilité en prédisant toutes les valeurs à 1. Dans ce cas toutefois, la précision et le rappel seront nuls.
L'une des mesures utilisées pour combiner précision et sensibilité est le score F1:

$$F_1 = 2 \times \frac{précision \times sensibilité}{précision + sensibilité}$$

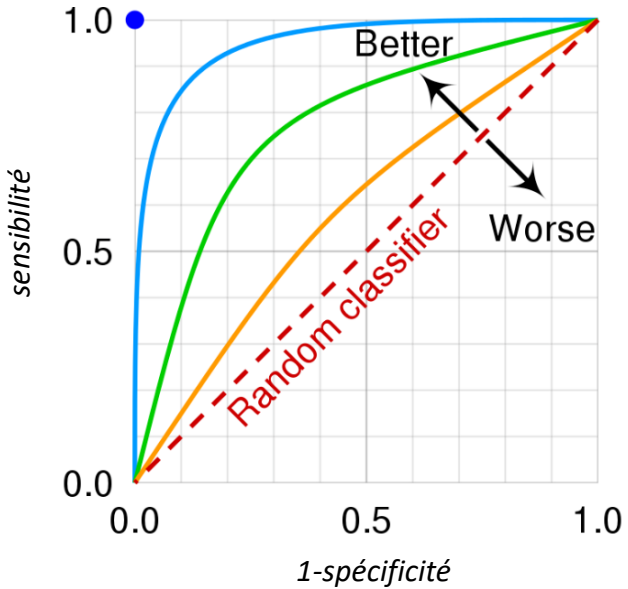


UQAC

# Mesure de performances

## La courbe ROC et l'aire sous la courbe ROC

On fait varier le seuil s et on calcule la sensibilité et la spécificité pour chacun des seuils.
Sur l'axe des x, nous avons 1-spécificité et sur l'axe des y nous avons la sensibilité.






UQAC

# Mesure de performances

**Comment améliorer sa performance prédictive**

- Essayer différents types de modèles est une bonne approche
- La recherche d'hyperparamètres est très importante également; on peut passer d'un modèle plus ou moins bon à un excellent modèle.
- Feature engineering:
  - décomposer la variable Date en année, mois et jour
  - Mettre les variables au carré ou encore à créer des intéractions entre les variables en les multipliant entre elles.
  - Choix du type de variable (continue, catégorielle)
  - La connaissance du domaine et de la problématique permet de créer des nouvelles variables pour mieux capter le signal dans les données.
- Réduire la dimensionalité grâce à des techniques comme l'Analyse en Composante Principale.



UQAC