Introduction à la Science des données
Forêt aléatoire

V. Bertret

[1] Université de Rennes 1/UFR Mathématiques

Outline

Contexte/Rappel sur l'apprentissage supervisé

Apprentissage supervisé : Apprendre à partir de données **labélisées/étiquetées**.

Jeu de données

- Données d'entrées (X), les régresseurs/features
- Données de sorties (Y), les réponses
- Un ensemble de paires (x, y), representant chacune un exemple

Objectif : Expliquer Y à partir de X (trouver les meilleurs paramètres θ tel que $Y = h_\theta(X)$ avec h_θ une certaine famille de fonction)

Exemples :

- Données
  - Y : spam/non spam, X : type d'adresse d'expéditeur, nombre d'adresses, fréquence des mots du titre, fréquence des mots du message, nombre de mots dans le message, etc.
  - Y : connection normale/connection attaque , X : temps de connection, heure de la connection, historique de cliques, etc.
- Modèle
  - Modèle linéaire : $f(x) = ax + b$. Deux paramètres à estimer (régression linéaire)
  - Modèle non-linéaire : $f(x) = ax + b\sin(cx) + d$. Quatre paramètres à estimer
  - K plus proches voisins.
  - ...

Observation arbre de décision

Principe : **partitionner l'espace en rectangles** dans lesquels la variable à prédire est **homogène** (faible variance) et d'ajuster un modèle **(très) simple** dans chaque région.

Avantages :

- Très rapide pour l'entraînement.
- Ne demande pas ou peu de préprocessing(prend en entrée tout type de données)

Problèmes

- Performance décevante.
- Gros problème de compromis biais/variance avec profondeur de l'arbre.
- Les **arbres profonds** ont une **forte variance** et les arbres **peu profonds** un **fort biais**.

Méthode d'ensemble

Comment pourrait-on agréger plusieurs méthodes ensembles pour diminuer la variance ?

- En faisant la **moyenne** ?
- En sélectionnant le **meilleur** ?
- . . .

Forêt aléatoire

- Entraînement de multiples arbres de décision d'une certaine manière afin de diminuer la variance de l'estimateur sans changer le biais.
- Modification des entrées des modèles afin d'avoir les **modèles les plus indépendants** possibles.
- Utilisation d'arbre de décision (CART) car rapide à construire.

L'idée de l' agrégation de modèles est de développer des procédures qui permettent de **combiner** les sorties de **classifier "faibles"** pour produire un **classifier performant**.

Outline

Bootstrap

- On utilise le bootstrap pour trouver le meilleur modèle parmi B modèles.

$$b^* = \arg\min_b \mathcal{L}_S(h^b_{\theta^*})$$

où L est une fonction de perte (Loss) [1].

- **Algorithme**
  Soit $S = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ un ensemble d'apprentissage.
  For $b = 1, \cdots, B$
  - Tirer aléatoirement avec remise $S^{(b)}$ de taille n_{app} dans S.
  - Ajuster le modèle h^b_θ.
  - Estimer l'erreur *out-of-bag* $\mathcal{E}^{(b)}_{oob}$.

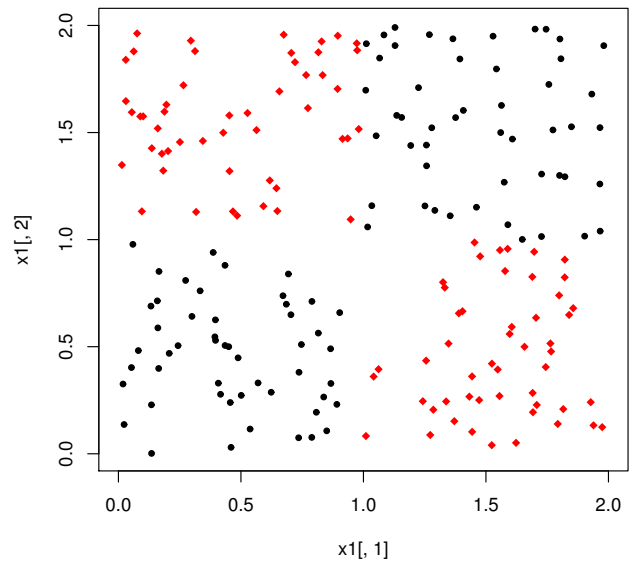
  Choisir le modèle $h^{b^*}_{\theta^*}$ tel que $b^* = \arg\min_{b\in\{1,\cdots,B\}} \mathcal{E}^{(b)}_{oob}$

- L'erreur *out-of-bag* $\mathcal{E}^{(b)}_{oob}$ est l'approximation empirique de la perte $\mathcal{L}$ estimée à partir des observations de l'ensemble d'apprentissage qui ne sont pas dans l'échantillon $S^{(b)}$.
- Cette approche par bootstrap permet de réduire le risque de sur-apprentissage ; En revanche la variance d'estimation reste forte : on ajuste B modèles mais on n'en conserve qu'un finalement.

1. Régression : $\mathcal{L}(f_\theta(X)) = E\left[Y - \hat{f}_\theta(X)\right]^2$; classification $\mathcal{L}(f_\theta(X)) = P(Y \neq \hat{f}_\theta(X))$

Exemple

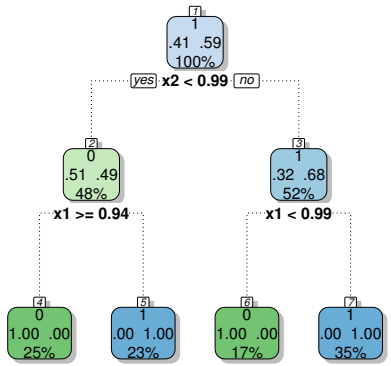
- On cherche un classifieur pour les données suivantes (n=200).



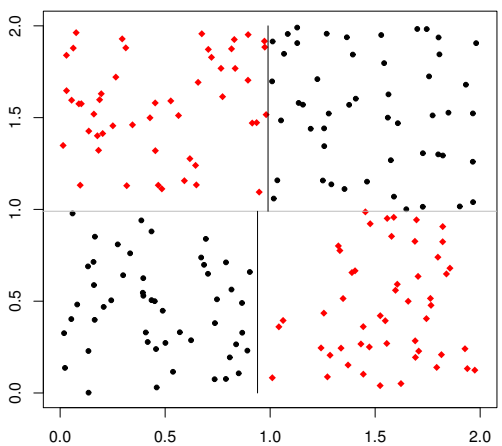
  Les classes sont matérialisées par la couleur des points.
- 500 arbres sont ajustés sur les sous échantillons $S^{(b)}$ de taille 150.
- L'erreur de classification est estimée sur l'échantillon *out-of-bag*.

Forte variabilité entre 2 modèles

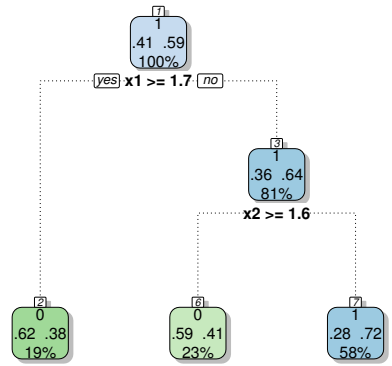
Meilleur modèle : Erreur = 0



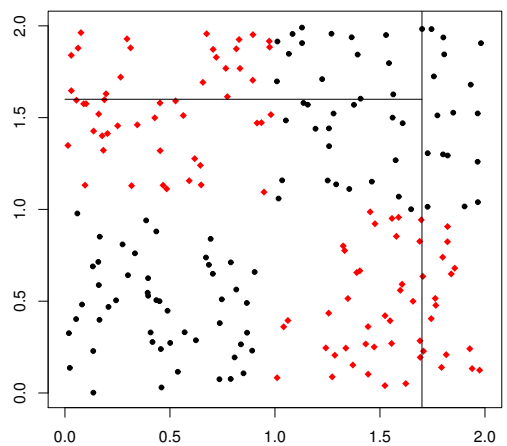
Rattle 2015−nov−26 18:29:37 valerie



Plus mauvais modèle : Erreur = 0.5



Rattle 2015−nov−26 18:29:38 valerie



- Beaucoup de variabilités entre les modèles.
- Pas d'exploitation de la variabilité.

Outline

Agregation de modèles indépendants

Supposons que nous avons un ensemble de B estimateurs $\{h^{(b)}_{\theta^*}(x)\}_{1,\dots,B}$.

Règle de décision :

$$h_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} h^{(b)}_{\theta^*}(x) \text{ ou } h_{bag}(x) = \arg \max_{k\in\{1,\cdots,K\}} \sum_{b=1}^{B} \delta_{h^{(b)}_{\theta^*}(x)=k}$$

- On remarque que si les estimateurs $h^{(b)}_{\theta^*}(x)$ sont indépendants et identiquement distribués on a

$$E(h_{bag}(x)) = E(h^{(1)}_{\theta^*}(x))$$

$$Var(h_{bag}(x)) = \frac{1}{B} Var(h^{(1)}_{\theta^*}(x))$$

Les espérances sont calculées sur la population des données $\mathcal{D}_n = \{(Y_1, X_1), (Y_2, X_2), \cdots, (Y_n, X_n)\}$.

- Ainsi, l'agrégation permet de **réduire la variance** mais **mais ne change pas le biais**.

Agregation de modèles non indépendants

Ainsi, l'agrégation permet de **réduire la variance** mais **mais ne change pas le biais**.

- Si les estimateurs $h^1_{\theta^*}(x), \cdots, h^B_{\theta^*}(x)$ sont i.i.d,
  - Biais : $E(\hat{h}_{bag}(x) - h(x)) = E(\hat{h}^1_{\theta^*}(x) - h(x))$
  - Variance : $Var(h_{bag}(x)) = \frac{1}{B} Var(\hat{h}^1_{\theta^*}(x))$

  Le choix de B n'est donc pas crucial pour la performance de l'estimateur, mais il est recommandé de le prendre **le plus grand possible** (en fonction du temps de calcul disponible).

- Les échantillons $S^{(b)}$ sont corrélés (ils contiennent des points communs).
  En notant $\rho(x) = corr(h^b_{\theta^*}(x), h^{b'}_{\theta^*}(x))$, on a

$$Var(h_{bag}) \simeq \rho(x) Var(h^1_{\theta^*})$$

  si B est grand.

C'est donc la **corrélation** $\rho(x)$ entre les estimateurs que l'on agrège qui quantifie le gain de la procédure d'agrégation : **la variance diminuera d'autant plus que les estimateurs que l'on agrège seront "différents" (décorrélés).**

Outline

Bagging

- **Bagging** est inspiré du **bootstrap** ; il est introduit par Léo Breiman en 1996. L'idée est de combiner des modèles entraîné sur des échantillons **bootstrap** en calculant la **moyenne** de leurs prédictions (ou vote à la majorité).

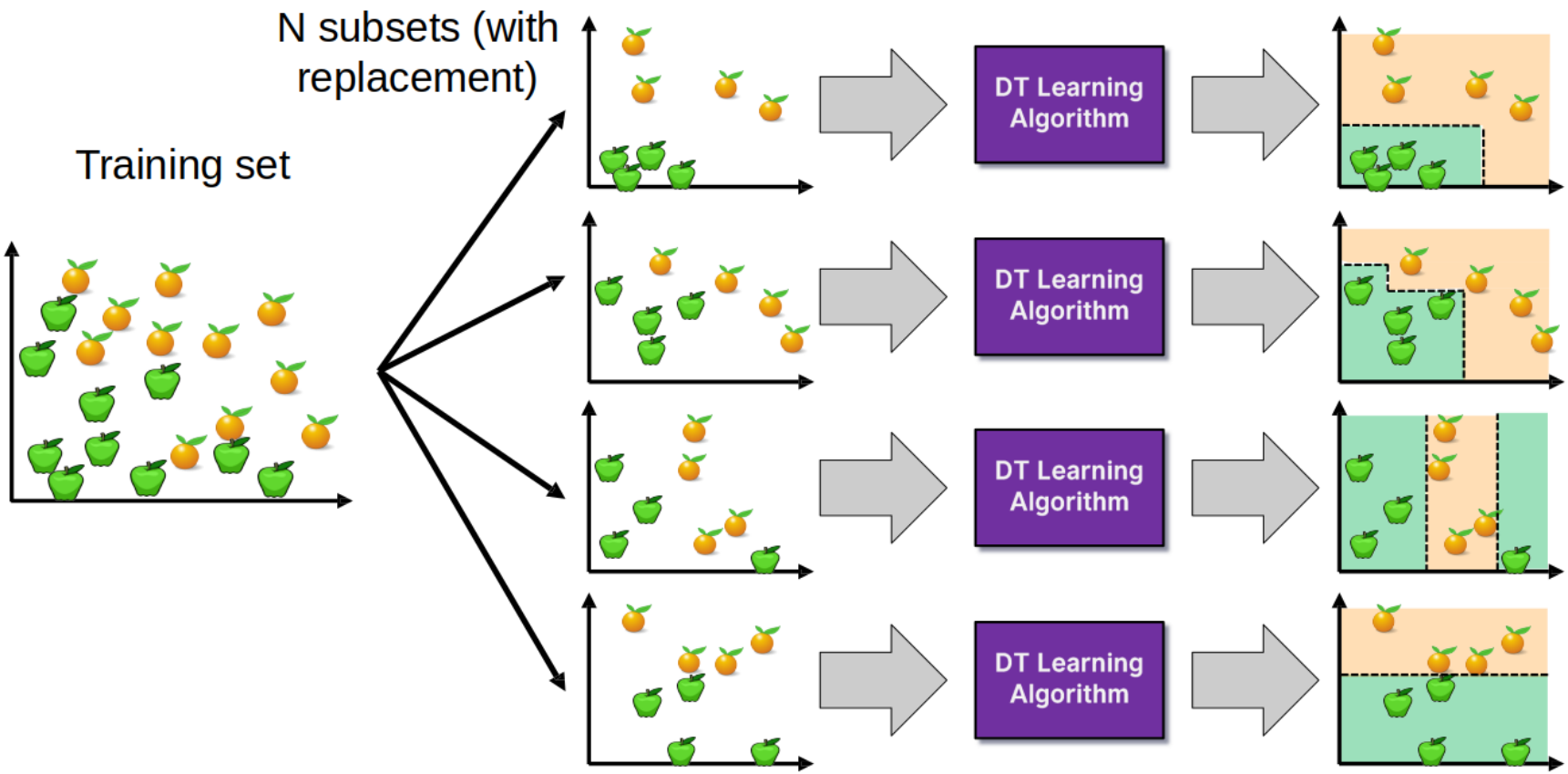
- **Algorithme**
  Soit $S = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ un ensemble d'apprentissage.
  For $b = 1, \cdots, B$
  - Tirer aléatoirement avec remise $S^{(b)}$ de taille n_{app} dans S.
  - Ajuster le modèle $h_\theta^{(b)}$.

  Règle de décision :

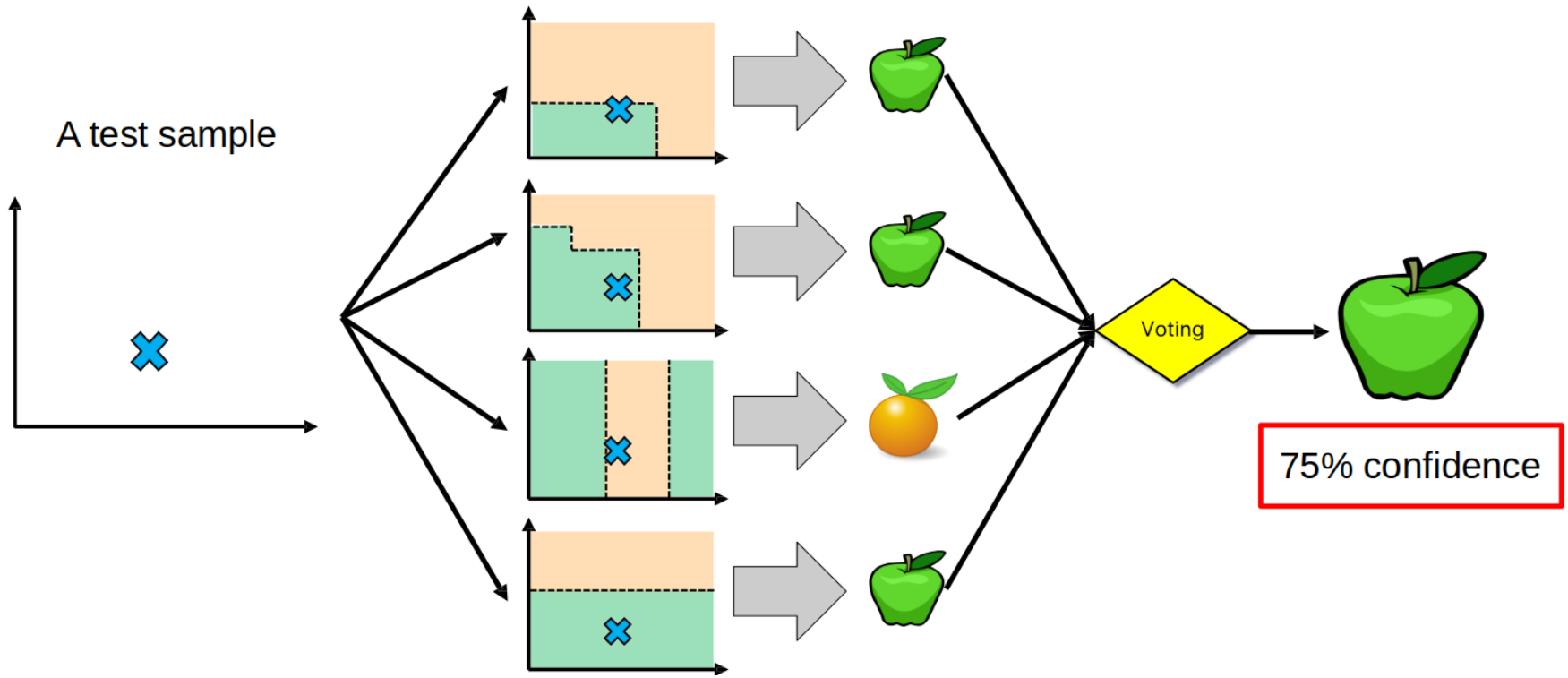
$$h_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} h_{\theta^*}^{(b)}(x) \text{ ou } h_{bag}(x) = \arg\max_{k\in\{1,\cdots,K\}} \sum_{b=1}^{B} \delta_{h_{\theta^*}^{(b)}(x)=k}$$

Les arbres de décision sont des classifieurs instables : une petite perturbation peut beaucoup changer l'arbre. **Ils sont donc de bons candidats pour le bagging !**

Entraînement des différents modèles avec le bagging.



Inférence des différents modèles avec le bagging.



Attention au choix des estimateurs faibles en classification

Pour un problème de classification à K classes, le bagging est basé sur un vote à la majorité

$$h_{\text{bag}}(x) = \arg \max_{k \in \{1, \cdots, K\}} \sum_{b=1}^{B} \delta_{h_{\theta^*}^{(b)}(x)=k}$$

En classification la réduction de variance n'est pas garantie. Bagguer un bon classifieur permet de l'améliorer, mais bagguer un mauvais classifieur peut l'empirer.

- Exemple d'un mauvais classifieur.
  Supposons que $Y = 1$ pour tout x, et que le classifieur $\hat{G}(x)$ predise
  $Y = 1$ avec une probabilité égale à 0.4
  $Y = 0$ avec une probabilité égale à 0.6.

  Alors la prédiction du bagging sera toujours 0 (qui a une plus grande probabilité d'occurrence).
  L'erreur de $G(x)$ est égale à 0.6 et celle du bagging est égale à 1.0.

Attention au choix des estimateurs faibles en classification

Pour un problème de classification à K classes, le bagging est basé sur un vote à la majorité

$$h_{\text{bag}}(x) = \arg \max_{k \in \{1, \cdots, K\}} \sum_{b=1}^{B} \delta_{h_{\theta^*}^{(b)}(x)=k}$$

En classification la réduction de variance n'est pas garantie. Bagger un bon classifieur permet de l'améliorer, mais bagguer un mauvais classifieur peut l'empirer.

- Exemple d'un bon classifieur.
  Notons $G(x) = 1$ la règle optimale pour x dans un problème à 2 classes.
  Supposons que chaque classifieur faible G_b admet un taux d'erreur $e_b = e < 0.5$, et définissons

$$S_1(x) = \frac{1}{B} \sum_{b=1}^{B} \mathbb{I}_{G_b^*(x)=1}$$

  le vote pour la classe 1.
  Si les classifieurs faibles sont independants [2],

$$S_1(x) \sim \mathcal{B}(1, 1-e)$$

  et $P(S_1(x) > 1/2) \to 1$ quand B tend vers l'infini.

2. mean of B independent $\{0, 1\}$ samples

Remarques supplémentaires Bagging

- L'**estimateur** obtenu par **bagging diffère** de l'estimateur "**faible**" seulement si l'estimateur faible est **non linéaire** ou **non paramétrique.**
- Possibilité d'utiliser l'échantillon **OOB** pour estimer la "vraie" erreur et peut-être utilisée pour faire de la **validation**.

Supposons qu'il existe un **prédicteur très fort** dans l'ensemble de données, ainsi qu'un certain nombre d'autres prédicteurs modérément forts.

Arbres entraînés par sur les échantillons Bootstrap

- **Sélectionnent tous** le prédicteur fort **au sommet de l'arbre.**
- **Les arbres se ressemblent tous.**

Comment éviter ce phénomène ?

Outline

Feature Agregating

Solution : Feature Agregating, c'est à dire ajuster N arbres de décision différents en contraignant chacun d'entre eux à opérer sur un sous-ensemble de caractéristiques.

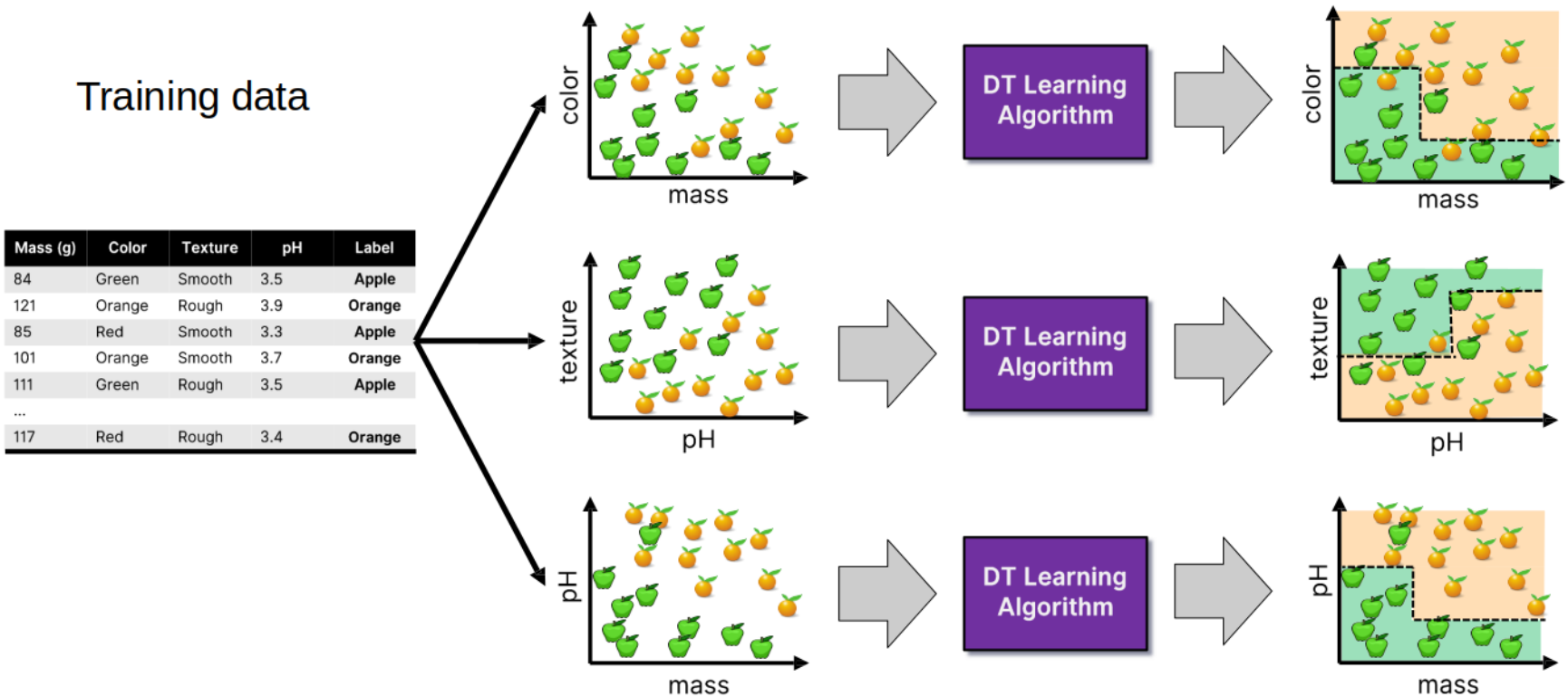
Comment choisir les features/caractéristiques ?

- **De manière aléatoire.**
  - Le prédicteur fort ne sera pas choisi dans tous les arbres.
  - Augmentation de l'indépendance entre les arbres.

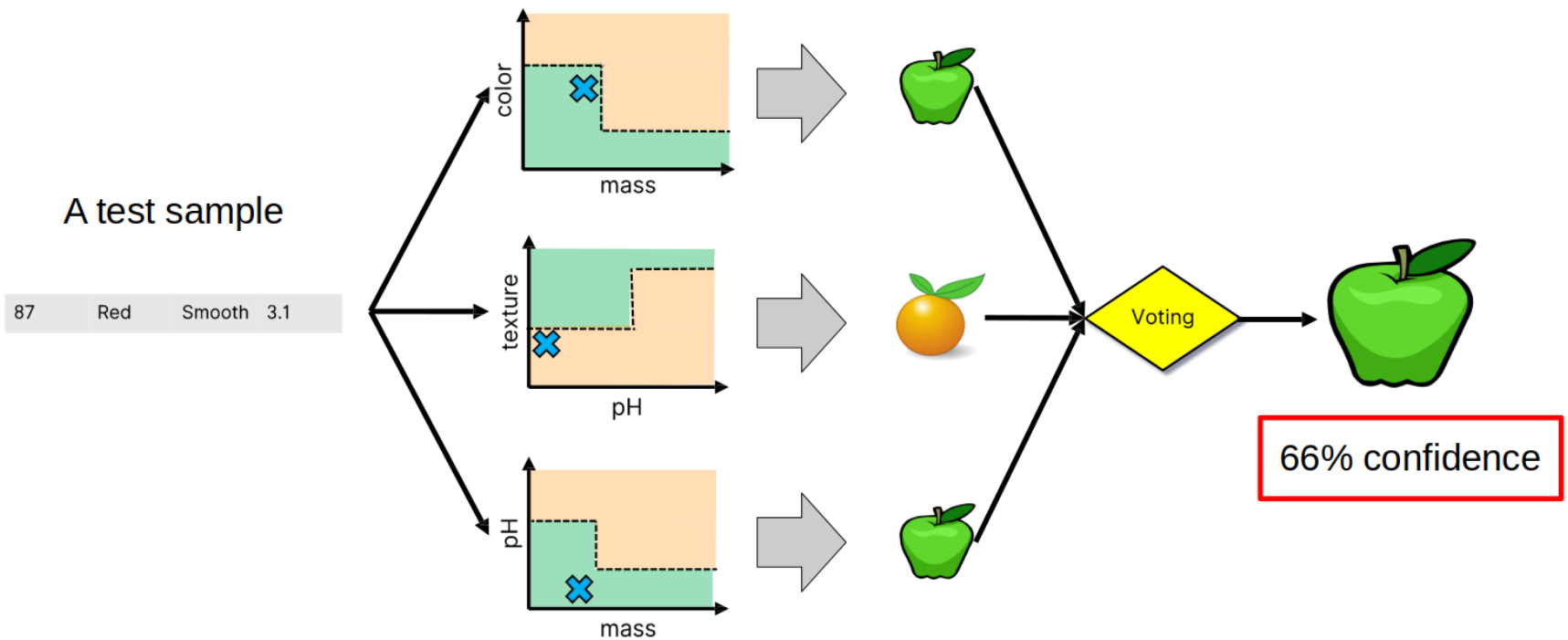
Un **seul** jeu de données nous permet d'obtenir **différents arbres**.

- Bootstrap Aggregating.
- Feature Bagging.

Entraînement des différents modèles avec le Feature Bagging.



Inférence des différents modèles avec le Feature Bagging.



Outline

Random Forest : Forêts

- Comme son nom l'indique, une **forêt aléatoire** consiste à **agréger** des **arbres** de discrimination ou de régression.
- Soit $T_b(x)$, $b = 1, \cdots, B$, B prédicteurs associés aux arbres $(T_b)_b$.

  Le prédicteur de la forêt aléatoire est obtenu en agrégeant les arbres de la collection :

$$\hat{T}_B(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x)$$

- Les forêts les plus connues sont dues à L. Breiman (2000) et sont construites à partir d'arbres CART (coupures dyadiques).

Random Forest : Forêts aléatoires

Comment obtenir de l'aléas entre les arbres ?

- Les arbres d'une forêt aléatoire sont ajustés sur des échantillons bootstrap comme dans le bagging.

  Particularité construction arbre :

- A **chaque noeud** d'un arbre donné, la coupure est basée sur la **"meilleure" variable** d'un sous ensemble de variables tiré aléatoirement : Feature bagging à chaque noeud.

Ces astuces permet de réduire la corrélation entre les arbres de la forêt.

- On a donc **deux** sources d'**aléa** dans les forêts :
  - L'échantillonnage des **individus**.
  - L'échantillonnage des **variables**.

Algorithme

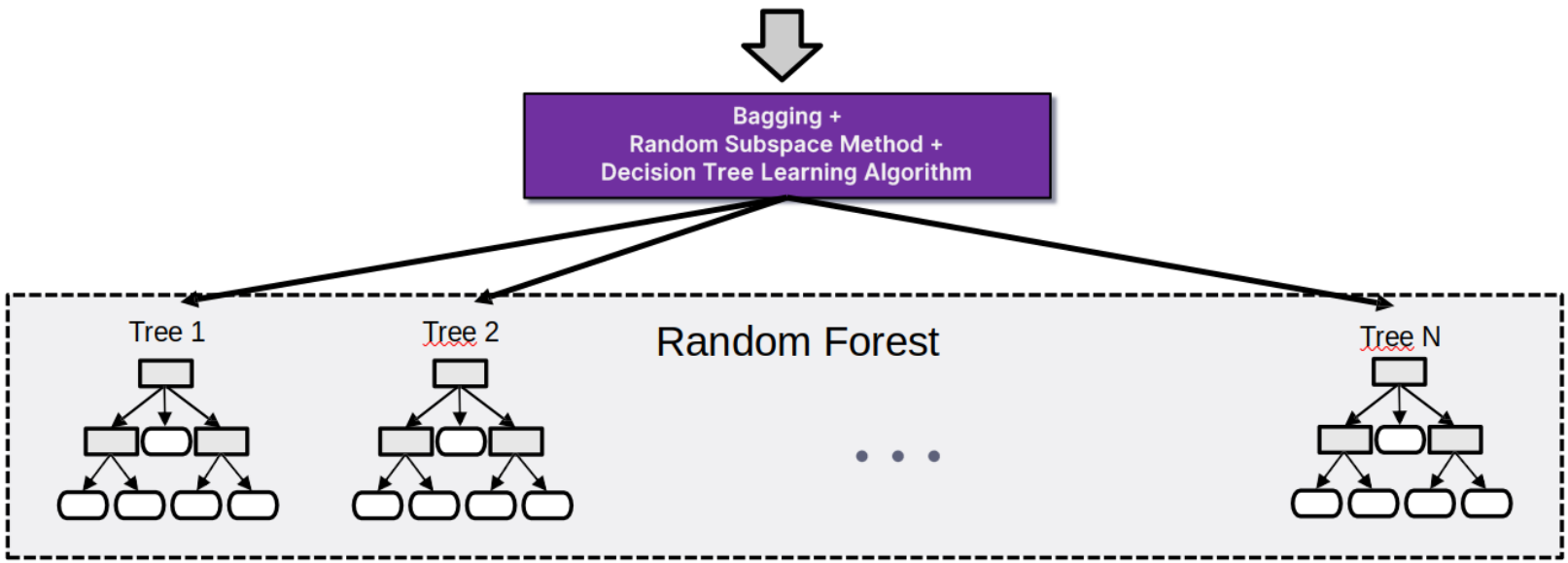
Forêt aléatoire

- $\mathcal{S} = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ a learning set.
- For $b = 1, \cdots, B$
  - (a) Draw a bootstrap sample $\mathcal{S}^{(b)}$ of size n among $\mathcal{S}$.
  - (b) Grow a tree $\mathcal{S}^{(b)}$ by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
    - i. Select m variables at random from the p variables.
    - ii. Pick the best variable/split-point among the m.
    - iii. Split the node into two daughter nodes.
  - (d) Estimate the generalization error using out-of-bag samples .
- For a new observation, $\mathbf{x}$, y is predicted by the mean or majority vote of the responses of the B regression trees.

Ref : Leo Breiman (2001), "Random Forests", Machine learning, vol. 45, p. 5-32.

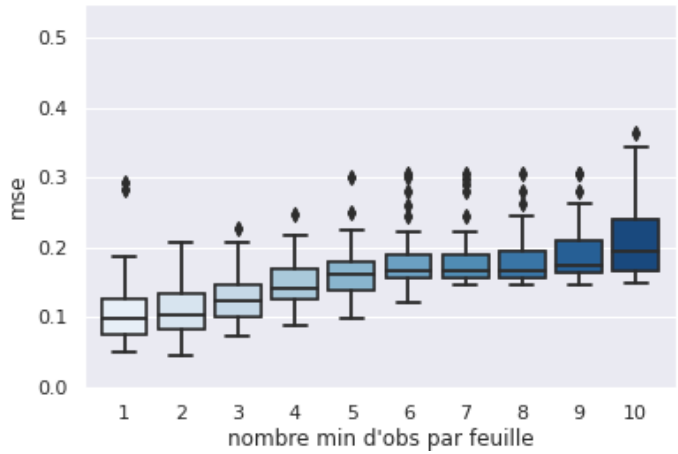
Illustration Forêt Aléatoire

Mass (g)	Color	Texture	pH	Label
84	Green	Smooth	3.5	**Apple**
121	Orange	Rough	3.9	**Orange**
85	Red	Smooth	3.3	**Apple**
101	Orange	Smooth	3.7	**Orange**
111	Green	Rough	3.5	**Apple**
...				
117	Red	Rough	3.4	**Orange**

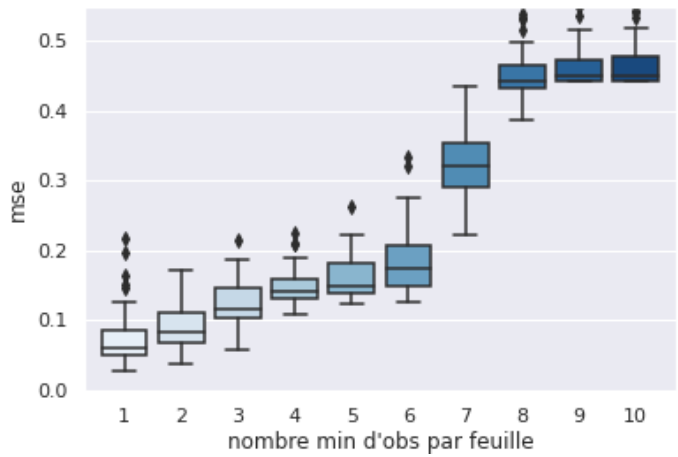


Attention au choix des classificateurs

$$y_i = (5x_i - 1)^2(x_i - 1) + u_i, \quad u_i \sim \mathcal{N}(0, 0.25)$$



On répète $S = 50$ fois l'estimation d'un arbre avec un nombre minimum d'observations par feuille variant de 1 à 10. Les boxplots représentent la distribution des $\widehat{MSE}$ sur l'entraînement.



Le graphique du bas est obtenu en construisant des forêts aléatoires.

Dans cet exemple à une **seule variable explicative**, les forêts permettent de réduire l'erreur en moyenne quadratique pour les estimateurs ayant un **biais faible**. Mais dans le cas des **"mauvais" estimateurs**, l'erreur **augmente beaucoup**.

Compromis biais-variance

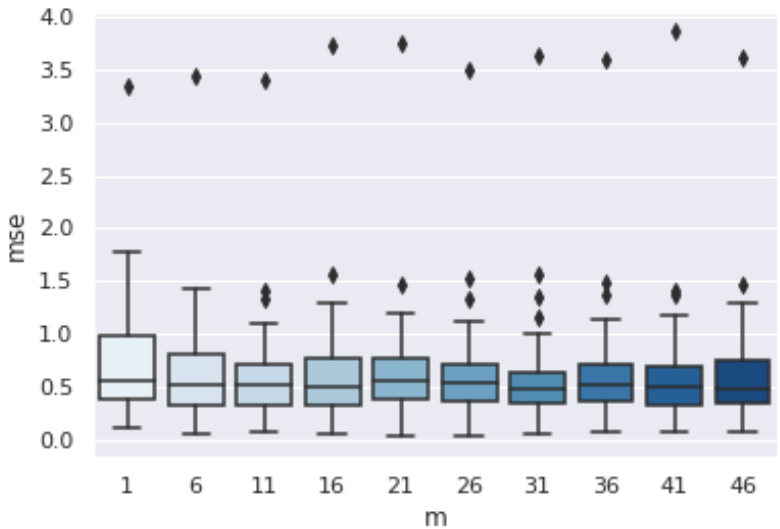
Nous retrouvons aussi un **compromis biais-variance** avec les **forêts aléatoires** :

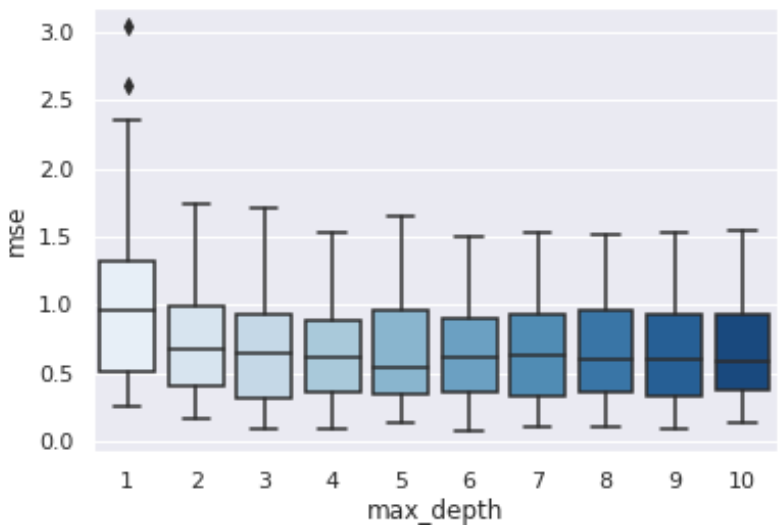
- avec les paramètres de l'arbre de décision (profondeur, complexity parameter)
- avec le choix du nombre de variables à sélectionner m.
  - lorsque m diminue, la corrélation entre les arbres va avoir tendance à diminuer également, ce qui entraînera une baisse de la variance de l'estimateur agrégé mais elle est associée à une augmentation du biais pour chaque arbre ainsi que pour l'estimateur agrégé.
  - lorsque m augmente, les phénomènes inverses se produisent.
  - L'algorithme "RandomForest" propose par défaut $m = p/3$ en régression et $m = \sqrt{p}$ en classification.
- avec le nombre d'estimateurs B.

Ces paramètres peuvent être **sélectionné** via des procédures de validation en utilisant différentes techniques de **validation croisée** ou bien en utilisant **l'erreur OOB**.

Compromis biais-variance en pratique : Prédiction du taux de graisse (biscuits).

Variables : spectroscopie NIR, 42 observations, environ 700 variables fortement corrélées
Critère de validation : MSE.





On répète 30 fois chaque expérience.

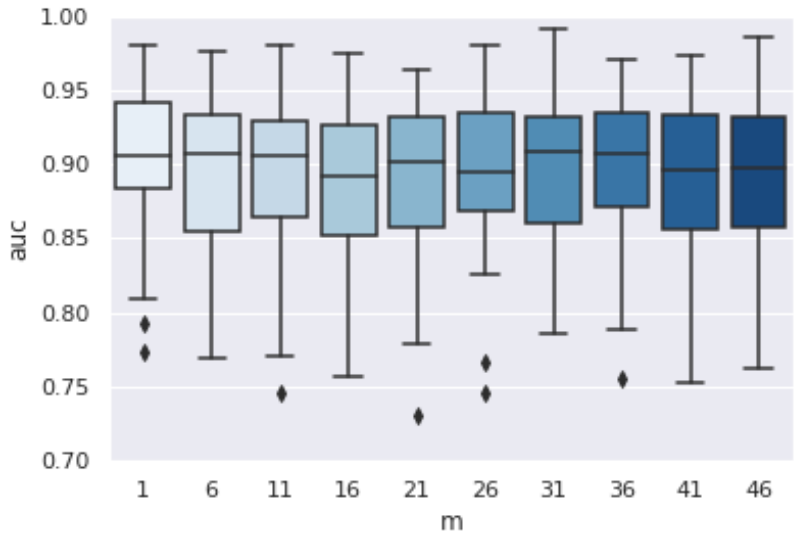
Le graphique du haut montre l'évolution de l'AUC en fonction de m pour une profondeur maximum de 5.

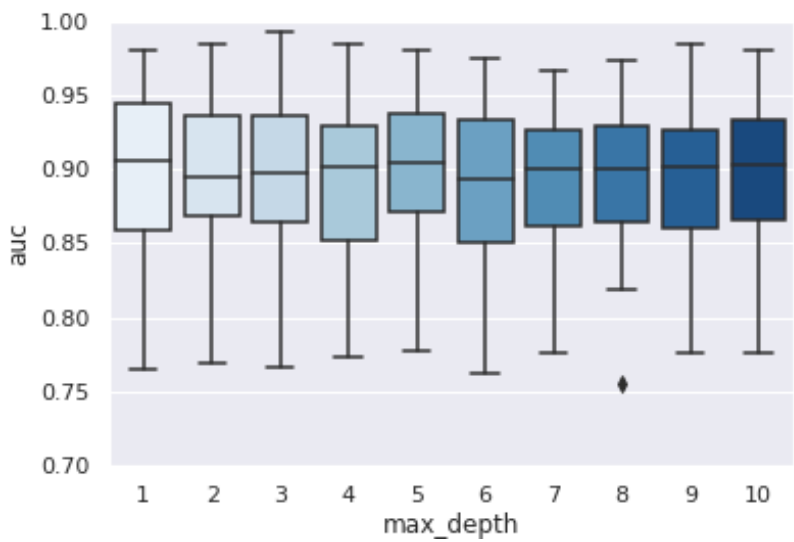
Le graphique du haut montre l'évolution de l'AUC en fonction de la profondeur maximum pour $m = 27$.

On observe que l'impact du choix des hyper paramètres est minime.

Compromis biais-variance en pratique : rechute après un traitement pour le cancer.

Variables : expressions de gène, 97 observations, 754 variables
Critère de validation : AUC.





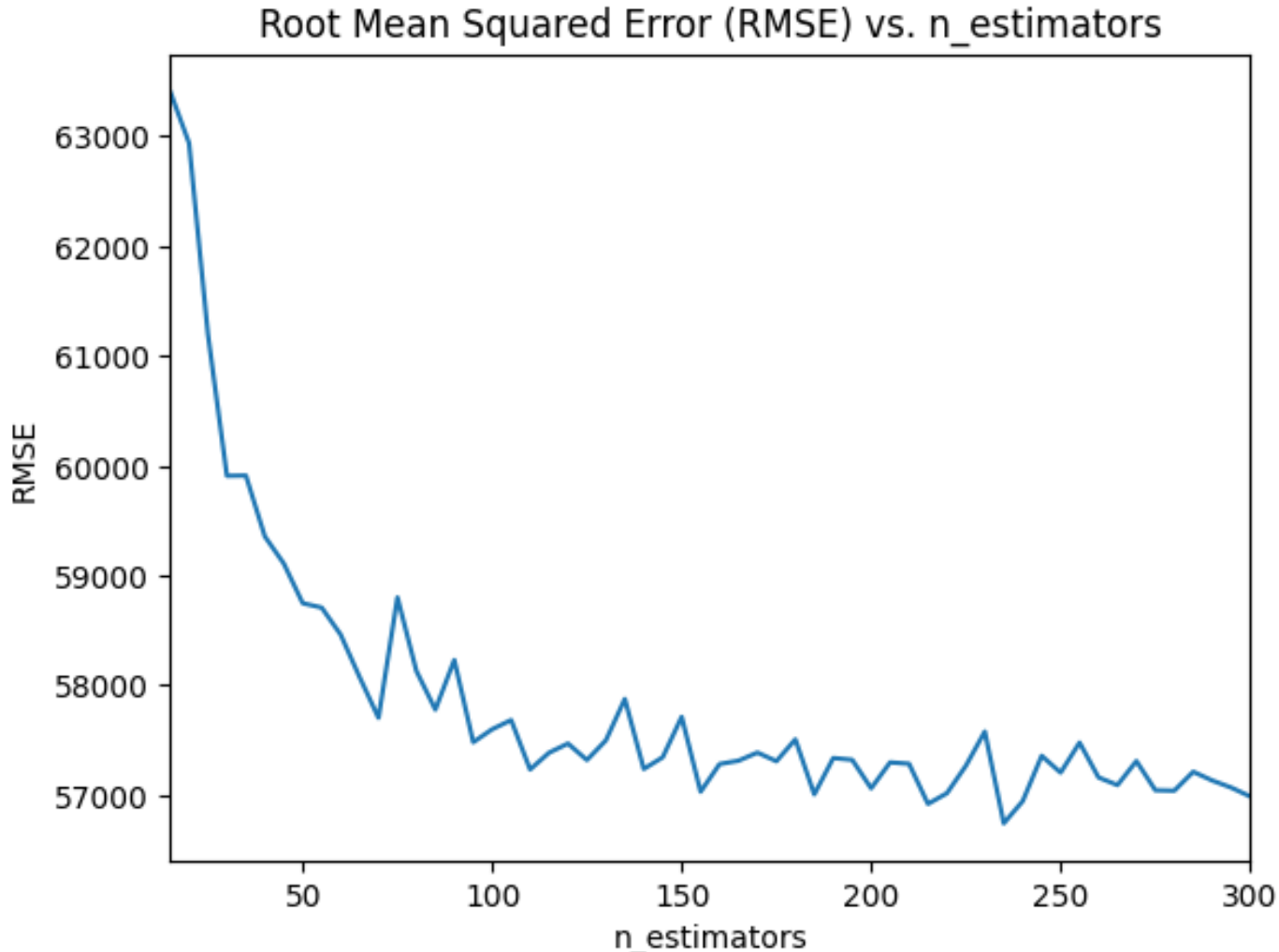
On répète 30 fois chaque expérience.

Le graphique du haut montre l'évolution de l'AUC en fonction de m pour une profondeur maximum de 5.

Le graphique du haut montre l'évolution de l'AUC en fonction de la profondeur maximum pour $m = 27$.

On observe que l'impact du choix des hyper paramètres est minime.

Compromis biais-variance en pratique : prédiction du prix médian des maisons en Californie



On observe qu'à partir d'un **nombre d'estimateur** l'erreur n'**évolue plus**. Il faut donc le choisir de manière à atteindre ce seuil au minimum.

Propriétés pratiques des forêts aléatoires

- Les **forêts aléatoires** sont facile à mettre en oeuvre.
- Elles conduisent à de **bons estimateurs** même quand la relation h est complexe et que les données sont de grande dimension.
- L'estimateur final est robuste dans le sens qu'il n'est **pas très sensible au choix des hyper-paramètres** (B, m, ...).
- Cependant, B doit être assez grand pour réduire la variance

$$Var(\hat{T}_B) = \rho(x)\, Var(T_k(x)) + \frac{1-\rho(x)}{B}\, Var(T_k(x))$$

- Le **biais** n'est pas impacté.
- Attention au choix des estimateurs (le biais ne doit pas être trop elevé).
- La théorie montre que m doit être proche de $p/3$ (quand n tend vers l'infini). Mais, Breiman montre par des exemples que choisir $m = 1$ ou $m = 2$ suffit souvent.
- Quand le nombre de variables est important mais la fraction de variables informatives est faible, les forêts aléatoires peuvent avoir des performances faibles si m est petit.
- La forêts aléatoires se stabilisent vite quand on augmente le nombre d'arbres.
- Un des principaux inconvénient de l'agrégation de modèle est qu'on perd l'interprétabilité. Pour les forêts aléatoires, il existe cependant des solutions.

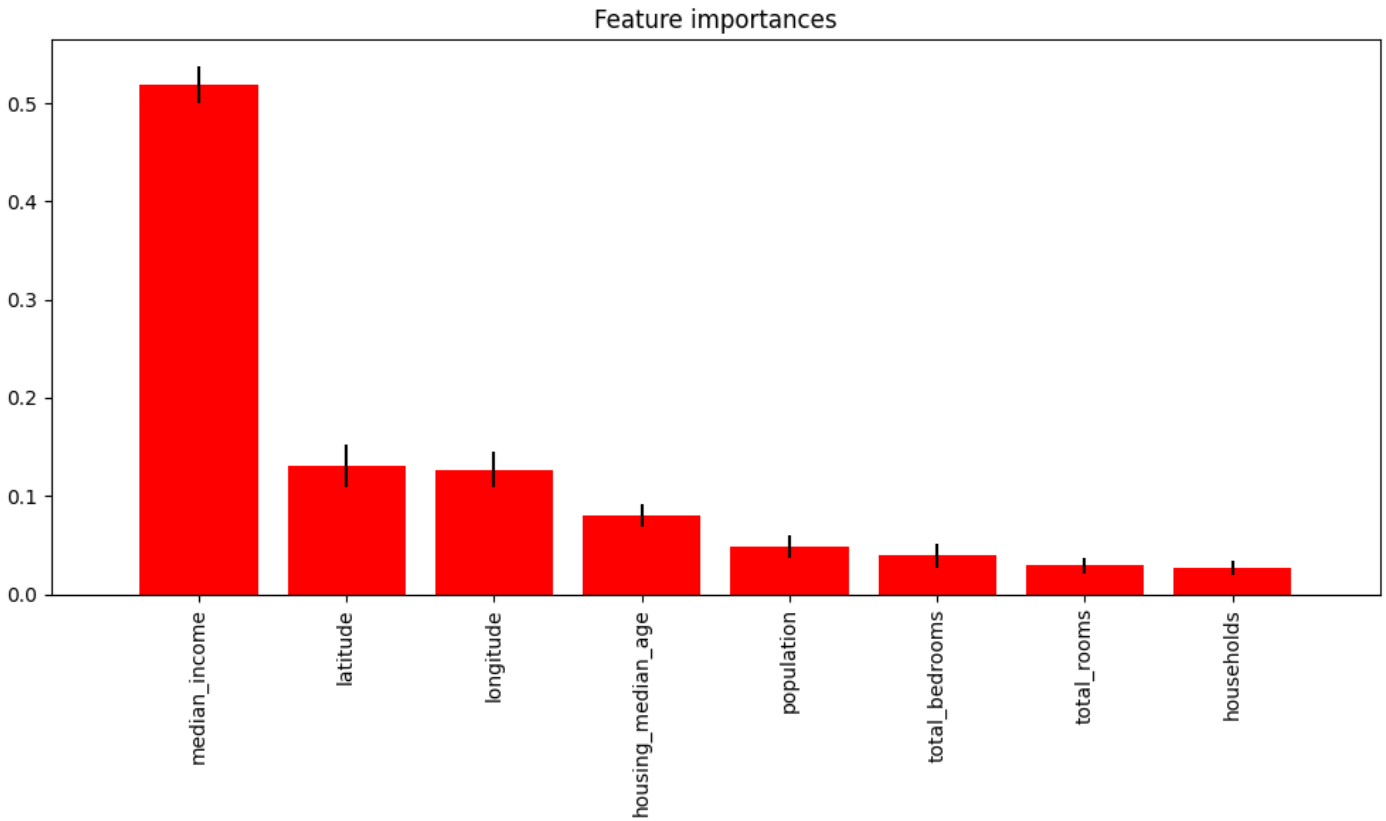
Interprétabilité

- On peut estimer **l'importance (relative) des variables** dans le cadre des forêts aléatoires.
- Une des mesures classiques est *permutation accuracy importance measure*.

- **Principe** : Test par permutation
  - Pour un prédicteur donné X_j, les observations sont permutées aléatoirement, et les observations de l'échantillon out-of-bag sont prédites.
  - La précision de la prédiction décroît significativement par rapport à l'original si X_j est associé à la réponse. En effet, la permutation casse l'association entre X_j et Y.
- Ainsi, on mesure l'**importance** de la variable par la **différence de précision** avant et après la permutation.

- L'**avantage** de cette méthode comparée à des **méthodes univariées** est qu'elle mesure l'impact du prédicteur dans un contexte multivarié (ie en interaction avec les autres variables.)

Importance sur la prédiction du prix médian des maisons en Californie



Le **revenu moyen** impacte de manière assez **importante la prédiction** alors que le **nombre de maisons** aux alentours **n'impacte pas beaucoup**.

Random Forest (Python)

```
from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier(n_estimators = 20)

clf.fit(X, y) # It can only handle numerical attributes !
# Categorical attributes need to be encoded, see LabelEncoder and OneHotEncoder

clf.predict([x]) # Predict class for x

clf.feature_importances_ # Importance of each feature
clf.tree_ # The underlying tree object

Clf.score(X_test, y_test) # Evaluate the performance of the model
```

Remarques de conclusion

- Le **bagging** est une technique d'**agrégation** qui aide à réduire la **variance**.
- La méthode de **bagging** la plus connue est la **forêt aléatoire**.
- Le principe est de **combiner** les prédictions de modèles élémentaires, appris sur des sous échantillons et si possible peu corrélés entre eux.
- Il est préférable que les sous échantillons soient petits (par exemple 1/3 de la base d'apprentissage).
- On peut prendre en compte **la performance des modèles élémentaires** en pondérant la prédiction par une mesure de **qualité des modèles**.
- **Interprétation** de l'**importance** de chaque variable pour la **prédiction**.

Références

Breiman, Leo. 1984. Classification and Regression Trees. Routledge.

Breiman, Leo, and Ross Ihaka. 1984. Nonlinear Discriminant Analysis via Scaling and Ace. Department of Statistics, University of California.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2001. The Elements of Statistical Learning. Vol. 1. Springer Series in Statistics New York, NY, USA :

Shalev-Shwartz, S., Ben-David, S. (2014). Understanding Machine Learning - From Theory to Algorithms.. Cambridge University Press. ISBN : 978-1-10-705713-5