


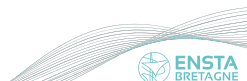
Evaluating machine-learning models

Machine learning basic concepts

Gilles Le Chenadec 



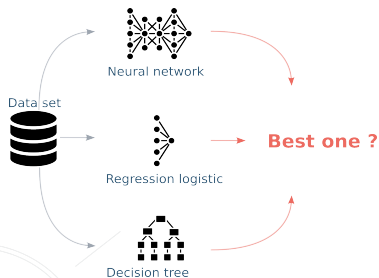
Introduction



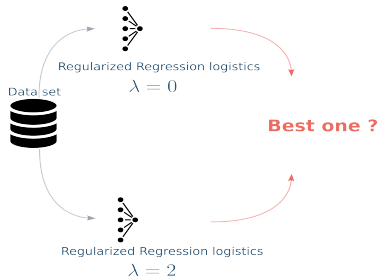
Motivations

Trois situations peuvent être rencontrées quand on veut évaluer des modèles de machine learning:

- Des algorithmes différents: SVM, random forest, régression linéaire, régression polynomiale, etc.
- Quel algorithme choisir ?



- Un seul algorithme avec des hyper-paramètres: régression polynomiale régularisée par exemple.
- Quelle valeur de ses hyper-paramètres choisir ?



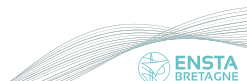
Qualité d'un modèle de machine learning

Qualité principale d'un modèle:

- capacité à apprendre sur les données d'apprentissage mais surtout à généraliser sur de nouvelles données
- i.e. de bonnes performances sur des données non “vues” au cours de l'apprentissage

Savoir quelle performance on peut attendre d'un ou plusieurs modèles

- Besoin d'une méthodologie pour comparer les modèles
- évaluer la validité de l'évaluation/la comparaison des performances



Contexte: en général, on navigue à l'aveugle!!!

Représentation des classes et de la frontière de décision sur les données d'apprentissage

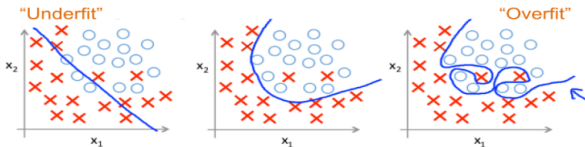


Image from Andrew Ng

En pratique, la représentation des points, classes et frontières de décision n'est possible que jusqu'à 3 dimensions!!!

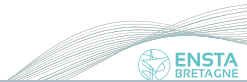
Qualités d'un modèle de machine learning (1/1)

Questions auxquelles il s'agit d'apporter une réponse

- **Trouver l(es) bonne(s) métrique(s) :**
 - ▶ Comment sait-on qu'on apprend bien?
 - ▶ Comment sait-on qu'on généralise bien?
- **Quelle stratégie d'apprentissage et d'évaluation adopter?**
 - ▶ Quelles données utiliser?
 - ▶ Si un ou plusieurs hyper-paramètres (par ex. λ de la régularisation) sont à fixer ? Comment fait-on efficacement?

Trouver l(es) bonne(s) métrique(s) de modèles de machine learning

- Métrique commune des modèles de machine learning
- Évaluation des modèles supervisés de régression
- Évaluation des modèles supervisés de classification



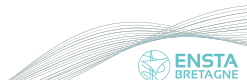
Trouver l(es) bonne(s) métrique(s) de performance de modèles de machine learning

Pour évaluer les performances des modèles de classification ou de régression sur une base de données, il faut définir des **métriques de performances**. Ces indicateurs servent à la fois à apprendre itérativement un modèle et à comparer les performances de différents modèles appris.

- Le coût (perte/loss) donné par la fonction de coût mesure la **qualité de la tâche qu'on a demandé à la machine d'apprendre**. Examiner son évolution au cours de l'apprentissage permet d'en apprendre (!) beaucoup sur la qualité de l'apprentissage.
- D'autres métriques sont utilisés pour évaluer les performances d'un modèle
 - ▶ spécifiques au modèle: par exemple accuracy pour la classification, mAP pour la détection d'objet
 - ▶ spécifiques à l'application: temps d'inférence (prédiction), gain financier généré
- En général, on utilise de multiples métriques pour évaluer ou sélectionner un modèle (comme quand vous êtes le choix de votre voiture!)

Métrique commune des modèles de machine learning

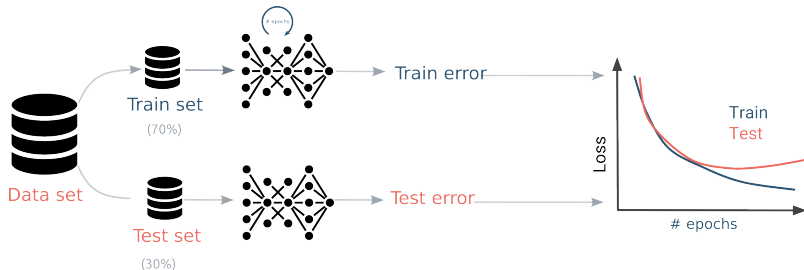
Trouver l(es) bonne(s) métrique(s) de modèles de machine learning



Évolution des fonctions de coût

Pour les modèles appris itérativement à partir d'une fonction de coût (p.e. les réseaux de neurones) pour la regression ou la classification, il est possible d'analyser **les courbes de coût en fonction du numéro d'itération** pour l'ensemble d'apprentissage et pour l'ensemble de test.

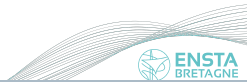
Un schéma d'évaluation typique peut être:



Valable si on a un grand nombre de données (explication après)

La qualité de l'apprentissage dépend de:

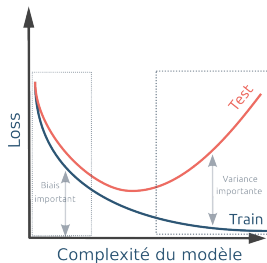
- la qualité et la complexité des données
- la complexité du modèle (linéaire, non linéaire, etc.)
- les hyper-paramètres (learning rate par exemple)



Complexité d'un modèle et compromis "biais / variance"

La **complexité d'un modèle** de machine learning est déterminée par la capacité du modèle à créer des fonctions de décisions complexes (# neurones d'un réseau de neurones par exemple).

Pour un jeu de données et pour un modèle, en général, alors que le nombre d'itérations augmente, la complexité du modèle augmente.

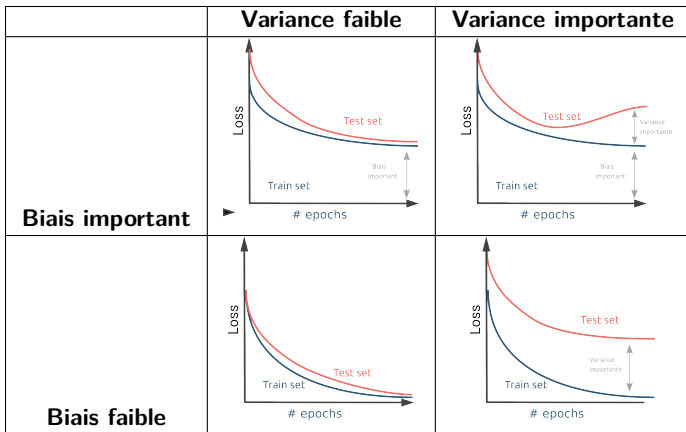


Un modèle est dit :

avec un **biais important** s'il n'est pas assez complexe pour un problème donné et a tendance à **sous-apprendre**,

avec une **variance importante** s'il est trop complexe pour un problème donné et a tendance à **sur-apprendre**.

Complexité d'un modèle et compromis "biais / variance"



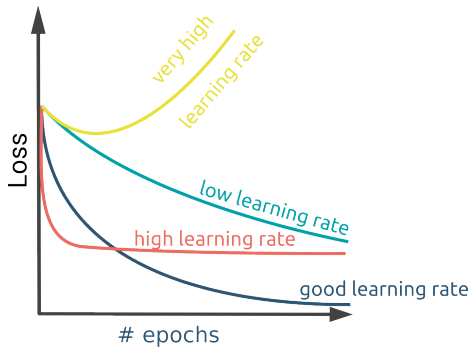
J'apprends bien (biais faible) et je généralise bien (variance faible).

Attention: le coût ne tend pas toujours vers 0 car cela dépend de la complexité du problème.

Influence du learning rate sur la fonction de coût

Tracer les deux fonctions de coût en fonction de:

- du numéro d'itération (epoch)
- pour plusieurs valeurs de learning rate



Remarques

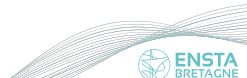
Que voulez vous analyser?

On peut analyser l'évolution de

- la fonction de coût (la qualité de la tâche demandée)
- n'importe quelle métrique (accuracy, etc.) (la performance finale)

en fonction:

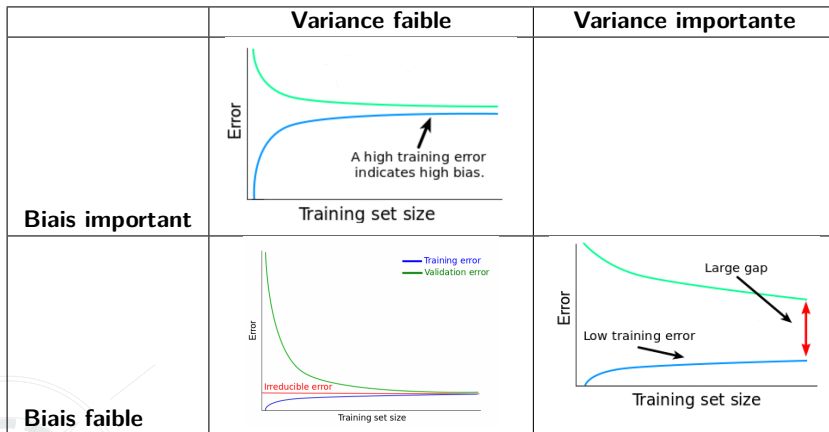
- du numéro de l'itération (epoch) (performance tout au long de l'apprentissage itératif)
- du nombre d'échantillons d'apprentissage (performance en fonction du nombre de données du data set)



Savoir si on a assez de données d'apprentissage, si le modèle est assez complexe

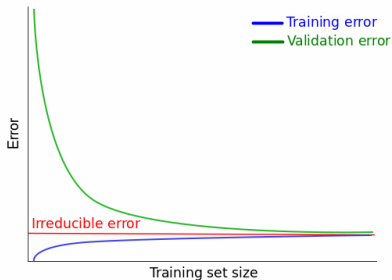
Tracer les deux fonctions de coût en fonction de:

- de la **taille de l'ensemble d'apprentissage**



Fonction de coût: savoir si on a assez de données d'apprentissage, si le modèle est assez complexe

Dans l'idéal, on veut



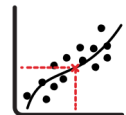
J'apprend bien (en bleu) et je généralise bien (en vert)

Évaluation des modèles supervisés de régression

Trouver l(es) bonne(s) métrique(s) de modèles de machine learning

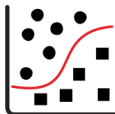


Évaluation des modèles supervisés de régression et de classification (1/1)



Regression

We try to predict a **quantity** (scalar, vector, ...)
→ Is my predicted value "good"?

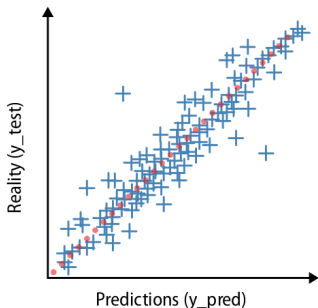


Classification

We try to predict a **quality** (class membership, ...)
→ Is my prediction "correct"?

Figure: Image extraite du cours FIDL.

Évaluation des modèles de régression



$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n [\hat{y}^{(i)} - y^{(i)}]^2$$

Mean Squared Error

Differentiable

Can be use as lost function

Increases very quickly

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n [\hat{y}^{(i)} - y^{(i)}]^2}$$

Root Mean Squared Error

Same unit as y

Robust to outliers

Humans understandable

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{y}^{(i)} - y^{(i)}|$$

Mean absolute error

Same unit as y

More robust to outliers

Humans understandable

$$\text{MAPE} = \frac{1}{n} \sum_{i=0}^n \frac{|y_i - \hat{y}_i|}{\max(\epsilon, |y_i|)}$$

Mean Absolute Percentage Error

Humans understandable (%)

Problem when y is null !

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

R² score, coefficient of determination

Result in [0, 1]

Measures a correlation between 2

series, nothing else..

Figure: Extrait de FIDL

Évaluation des modèles supervisés de classification

Trouver l(es) bonne(s) métrique(s) de modèles de machine learning



Matrice de confusion pour un modèle binaire

Comparaison des classes vraies (actual) et des classes prédites par le classifieur (prediction)

		Predictions		Total
		C^+	C^-	
Actual	C^+	TP	FN	N^+
	C^-	FP	TN	N^-
Total		\hat{N}^+	\hat{N}^-	N

- TP/TN : True positive/negative
- FP/FN : False positive/negative

Matrice de confusion pour un problème multi-classe

Pour la configuration multiclasse, on se ramène à un problème à deux classes (one-vs-all approach):

Pour la classe C_1 :

		Predictions					Total
		C_1	C_2	C_3	...	C_n	
Actual	C_1	TP_1	FN	FN	...	FN	N_1
	C_2	FP	TN	TN	...	TN	N_2
	C_3	FP	TN	TN	...	TN	N_3

	C_n	FP	TN	TN	...	TN	N_n
Total		\hat{N}_1	\hat{N}_2	\hat{N}_3	...	\hat{N}_n	N

- TP_i/TN_i : True positive/negative for Class $i \in \{1, \dots, n\}$
- FP_i/FN_i : False positive/negative for Class $i \in \{1, \dots, n\}$

Matrice de confusion pour un problème de classification multiclasse

Pour la configuration multiclasse, on se ramène à un problème à deux classes (one-vs-all approach):

Pour la classe C_2 :

		Predictions					
		C_1	C_2	C_3	...	C_n	Total
Actual	C_1	TN	FP	TN	...	TN	N_1
	C_2	FN	TP_2	FN	...	FN	N_2
	C_3	TN	FP	TN	...	TN	N_3

	C_n	TN	FP	TN	...	TN	N_n
Total		\hat{N}_1	\hat{N}_2	\hat{N}_3	...	\hat{N}_n	N

- TP_i/TN_i : True positive/negative for Class $i \in \{1, \dots, n\}$
- FP_i/FN_i : False positive/negative for Class $i \in \{1, \dots, n\}$

Matrice de confusion pour un problème de classification multiclasse

Pour la configuration multiclasse, on se ramène à un problème à deux classes (one-vs-all approach):

Pour la classe C_n :

		Predictions					
		C_1	C_2	C_3	...	C_n	Total
Actual	C_1	TN	TN	TN	...	FP	N_1
	C_2	TN	TN	TN	...	FP	N_2
	C_3	TN	TN	TN	...	FP	N_3

	C_n	FN	FN	FN	...	TP_n	N_n
Total		\hat{N}_1	\hat{N}_2	\hat{N}_3	...	\hat{N}_n	N

- TP_i/TN_i : True positive/negative for Class $i \in \{1, \dots, n\}$
- FP_i/FN_i : False positive/negative for Class $i \in \{1, \dots, n\}$

Évaluation des modèles de classification (1/1)

Real classes (y_test)	setosa	13	0	0
	versicolor	0	10	6
	virginica	0	0	9
	Predicted classes (y_pred)	setosa	versicolor	virginica

$$\text{accuracy} = \frac{\text{Total number of correct predictions}}{\text{Total number of prédictions}}$$

Ability to make correct predictions
« exactitude » en fr

$$\text{HammingLoss} = \frac{\text{Total number of wrong predictions}}{\text{Total number of prédictions}}$$

Ability to make wrong predictions

$$\text{precision}_{\text{class } i} = \frac{\text{Number of correct predictions for class } i}{\text{Total number of predictions for class } i}$$

Ability to identify without error, the elements of the class

$$\text{recall}_{\text{class } i} = \frac{\text{Number of correct predictions for class } i}{\text{Total number of real class } i}$$

Ability to identify all the elements of the class i

« sensibilité » en fr

$$F1_{\text{class } i} = 2 * \frac{\text{recall}_{\text{class } i} \cdot \text{precision}_{\text{class } i}}{\text{recall}_{\text{class } i} + \text{precision}_{\text{class } i}}$$

F1 is the harmonic mean of the model's precision and recall.

Figure: Extrait de FIDL

Métriques de performance

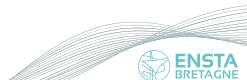
A minima,

- il faut examiner les matrices de confusion pour voir si une classe particulière n'est pas bien apprise.
- représenter le taux de précision ou d'erreur par classe

Pour aller plus loin, beaucoup d'autres mesures existent permettant de prendre en compte les types d'erreurs (FP, FN):

- Celles extraites de la matrice de confusion
 - ▶ Rappel (recall)
 - ▶ Specificité (specificity)
 - ▶ Precision (Precision)
 - ▶ F-score (f-score)
 - ▶ Matthews correlation coefficient (MCC) for binary classification
- Celles basées sur la courbe ROC (receiver operating characteristics): aire sous la courbe ROC (AUC)
- Jose A. Lozano, Guzmán Santafé, Iñaki Inza. "Classifier performance evaluation and comparison" International Conference on Machine Learning and Applications (ICMLA 2010) December 12-14, 2010

Stratégies d'apprentissage et d'évaluation



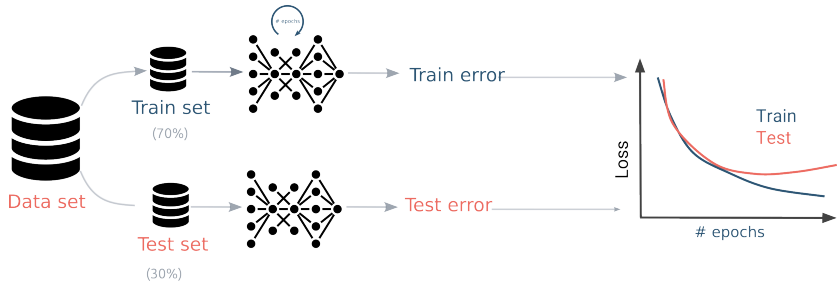
Comment utiliser les données disponibles?

Problème: On a un ensemble de données de **taille finie** qui doit être utilisé pour

- apprendre le modèle
 - ▶ plus il y a des données d'apprentissage, meilleure est la généralisation
- évaluer les performances
 - ▶ plus il y a des données d'évaluation, meilleure est l'estimation des performances
- choisir les hyper-paramètres
 - ▶ paramètres du modèle de machine learning; par exemple λ de régularisation

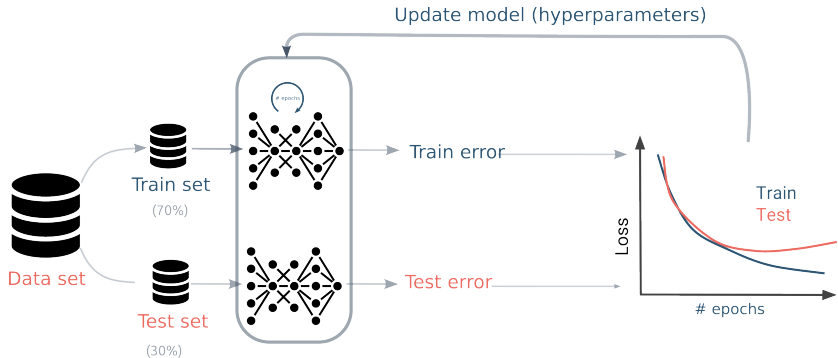
Stratégie basique et son biais (1/2)

Pour apprendre, on peut faire cela:



Stratégie basique et son biais (2/2)

Pour fixer les hyper-paramètres, il faut faire:

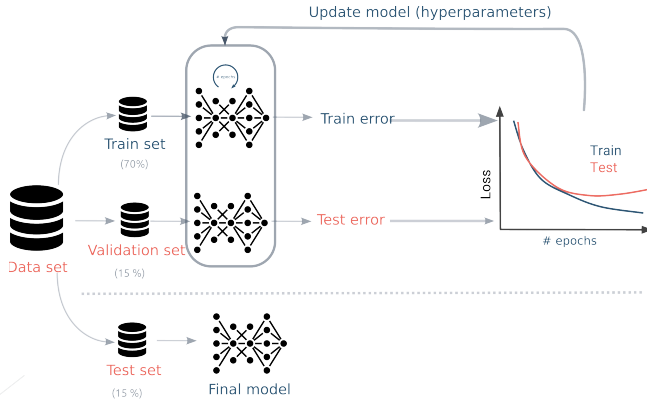


Dans cette stratégie, on apprend un modèle en fonction des données de test!
Attention, c'est un biais!

Stratégie pour les grands datasets (1/2)

Hold-out

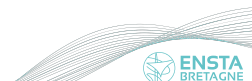
Pour les **grands ensembles de données**, pour une évaluation précise et une estimation des hyper-paramètres:



Stratégie pour les grands datasets (2/2)

Hold-out

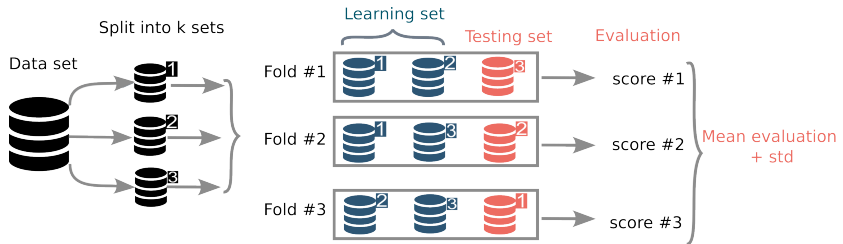
- Convient aux grands ensembles de données, ce qui permet de disposer de grands ensembles de validation et de test.
- Si les ensembles de validation et de test sont trop petits, l'évaluation finale sera statistiquement instable.



Stratégie pour les petits datasets (1/1)

k -fold cross validation

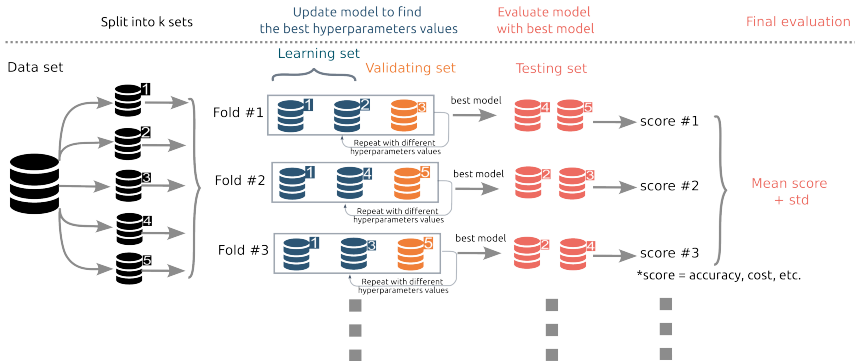
Stratégie très intéressante pour les petits datasets :



*score = accuracy, cost, etc.

- Ici le nombre de plis (fold), $k = 3$
- En pratique, nous préférons utiliser $k = 5$, $k = 8$, etc., $k = n$.
- Stratégie très intéressante pour les petits datasets.
- Cependant, si la quantité de données est faible, le résultat peut rester instable...

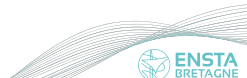
Nested k -fold cross validation (1/1)



- Stratégie à utiliser pour les petits datasets pour lesquels on doit fixer les hyper-paramètres

Quelques questions à garder à l'esprit (1/1)

- Mes sous-ensembles de données (train, test, etc.) sont-ils représentatifs de mes données ?
- Puis-je ou dois-je mélanger mes données ? (séquences temporelles, données ordonnées, etc.)
- Au sein de l'ensemble de données, quelle est la part et l'impact des valeurs aberrantes ?
- Mes résultats sont-ils significatifs ?
- Combien de plis, de combien d'itérations ai-je besoin ?
- De quelle quantité de données ai-je besoin ?
- Combien cela va-t-il coûter ?



Bibliographie

- Jose A. Lozano, Guzmán Santafé, Iñaki Inza. "Classifier performance evaluation and comparison" International Conference on Machine Learning and Applications (ICMLA 2010) December 12-14, 2010
- Dietterich, T. G., (1998). Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. Neural Computation, 10 (7) 1895-1924.
- Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research, 3, 1157-1182.
- Evaluating Learning Algorithms: A Classification Perspective Nathalie Japkowicz & Mohak Shah Cambridge University Press, 2011
- T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani, The elements of statistical learning, vol. 2. Springer, 2009.

