Machine Learning for Radiomics

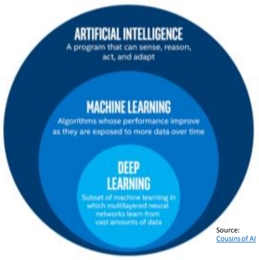
Carlos E. Cardenas, Ph.D.

Outline

- Introduction
- Data Curation
- Training, Validation, and Test datasets
- Linear and Logistic Regression
- Discriminant Analysis
- Penalized Regression (Ridge and Lasso)
- Support Vector Machines
- Decision Trees and Random Forests
- Clustering Methods
- Final Remarks

What is Machine Learning

- “Machine learning is a field of study that gives computers the ability to learn without being explicitly programmed”
  - Arthur Lee Samuel – 1959


ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act, and adapt
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amounts of data

Source: Cousins of AI

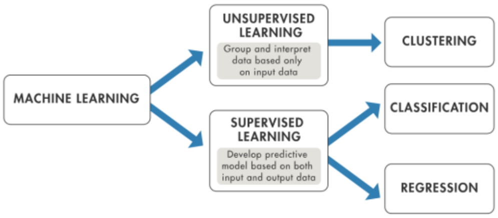
Why Machine Learning?

- Develop systems that can automatically adapt and customize themselves to individual users
- Discover new knowledge from large databases (data mining)
- Automate monotonous tasks (which may require some intelligence)
- Develop systems that are too difficult to hard-code because they require specific detailed skills or knowledge relevant to a specific task
  - Knowledge engineering bottleneck

Why now?

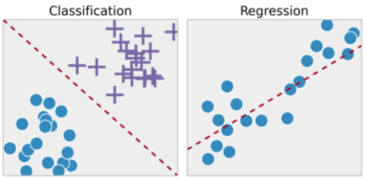
- Large amounts of clinical data
- Increasing computation power
- Growing progress in available algorithms and theory developed by researchers
- Increasing support from industries and funding agencies

Supervised vs Unsupervised Learning


MACHINE LEARNING
UNSUPERVISED LEARNING
Group and interpret data based only on input data
SUPERVISED LEARNING
Develop predictive model based on both input and output data
CLUSTERING
CLASSIFICATION
REGRESSION

Source: https://www.mathworks.com/discovery/machine-learning.html

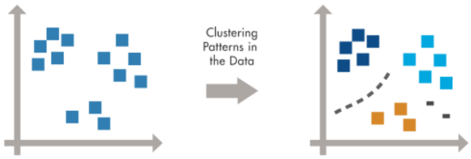
Supervised Learning



Source: KDnuggets

Goal: to find specific relationships or structure in the input data that allow us to effectively produce correct output data

Unsupervised Learning



Goal: to learn the inherent structure of our data without using explicitly-provided labels

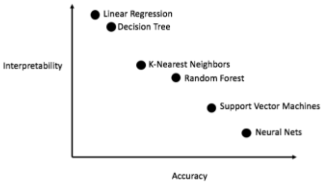
Supervised vs Unsupervised Learning

- Which one should I use??
  - Supervised Learning
    - if you need to train a model to make a prediction--for example, the future value of a continuous variable, such as patient weight or tumor size, or a classification—for example, a segmentation task or HPV status.
  - Unsupervised learning
    - if you need to explore your data and want to train a model to find a good internal representation, such as splitting data up into clusters.

The two “Trade-offs”

1. **Prediction Accuracy vs Model Interpretability**
   - Global Interpretability
   - Local Interpretability
   - Feature Selection

Source: Interpreting Machine Learning Models



The two “Trade-offs”

2. **Bias vs Variance**
   - Bias: error from erroneous assumptions in the learning algorithm
   - Variance: error from sensitivity to small fluctuations in the training set

Expected Error

$$\mathrm{E}\left[(y-\hat{f}(x))^2\right] = \left(\mathrm{Bias}\left[\hat{f}(x)\right]\right)^2 + \mathrm{Var}\left[\hat{f}(x)\right] + \sigma^2$$

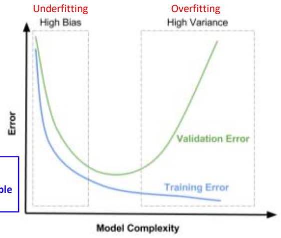
Irreducible Error

where

$$\mathrm{Bias}\left[\hat{f}(x)\right] = \mathrm{E}\left[\hat{f}(x) - f(x)\right]$$

and

$$\mathrm{Var}\left[\hat{f}(x)\right] = \mathrm{E}[\hat{f}(x)^2] - \left(\mathrm{E}[\hat{f}(x)]\right)^2$$



Source: Bias-Variance Tradeoff in Machine Learning

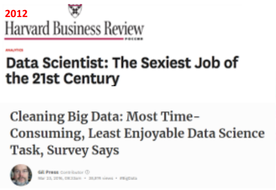
No Free Lunch in ML

Data Curation

Medical data mining

Linking diseases, drugs, and adverse reactions

Data Curation



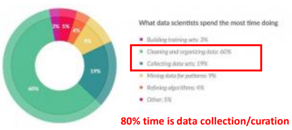
2012

Harvard Business Review

Data Scientist: The Sexiest Job of the 21st Century

Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says

Gil Press Contributor

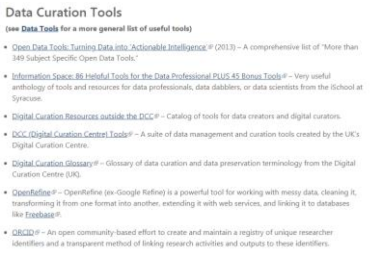


80% time is data collection/curation

Data Curation

NIH NLM NNLM
National Network of Libraries of Medicine

https://nnlm.gov/data/data-curation



Data Curation Tools

(see Data Tools for a more general list of useful tools)

- Open Data Tools: Turning Data into 'Actionable Intelligence' (2013) – A comprehensive list of "More than 349 Subject Specific Open Data Tools."
- Information Space: 86 Helpful Tools for the Data Professional PLUS 45 Bonus Tools – Very useful anthology of tools and resources for data professionals, data dabblers, or data scientists from the iSchool at Syracuse.
- Digital Curation Resources outside the DCC – Catalog of tools for data creators and digital curators.
- DCC (Digital Curation Centre) Tools – A suite of data management and curation tools created by the UK's Digital Curation Centre.
- Digital Curation Glossary – Glossary of data curation and data preservation terminology from the Digital Curation Centre (UK).
- OpenRefine – OpenRefine (ex-Google Refine) is a powerful tool for working with messy data, cleaning it, transforming it from one format into another, extending it with web services, and linking it to databases like Freebase.
- ORCID – An open community-based effort to create and maintain a registry of unique researcher identifiers and a transparent method of linking research activities and outputs to these identifiers.

Splitting Datasets

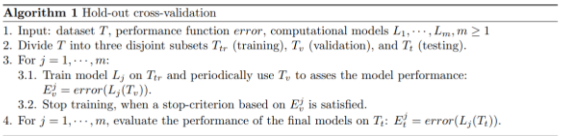
- Sampling Techniques
  - Simple Random Sampling (SRS)
  - Trial-and-error Methods
  - Systematic Sampling
  - Convenience Sampling
  - CADEX, DUPLEX
  - Stratified Sampling
- Typically in ML/DL we split our data set (of size n) into three subsets:
  - Training (train model)
  - Validation (evaluate model during hyper-parameter selection)
  - Test

Most commonly used

Resource: https://www.mff.cuni.cz/veda/konference/wds/proc/pdf10/WDS10_105_i1_Reitermanova.pdf

Cross-validation Techniques

- Hold-out cross-validation
  - Training, Validation, and Test mutually disjointed datasets
  - Advantage: proportion of subsets are not strictly restricted
  - Train models: Training Set
  - Fine-tune models: Validation Set



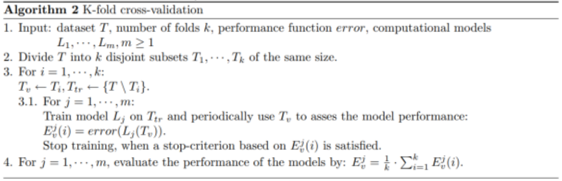
Algorithm 1 Hold-out cross-validation

```
1. Input: dataset $T$, performance function $error$, computational models $L_1, \cdots, L_m, m \geq 1$
2. Divide $T$ into three disjoint subsets $T_{tr}$ (training), $T_v$ (validation), and $T_t$ (testing).
3. For $j = 1, \cdots, m$:
   3.1. Train model $L_j$ on $T_{tr}$ and periodically use $T_v$ to asses the model performance:
        $E_v^j = error(L_j(T_v))$.
   3.2. Stop training, when a stop-criterion based on $E_v^j$ is satisfied.
4. For $j = 1, \cdots, m$, evaluate the performance of the final models on $T_t$: $E_t^j = error(L_j(T_t))$.
```

Resource: https://www.mff.cuni.cz/veda/konference/wds/proc/pdf10/WDS10_105_i1_Reitermanova.pdf

Cross-validation Techniques

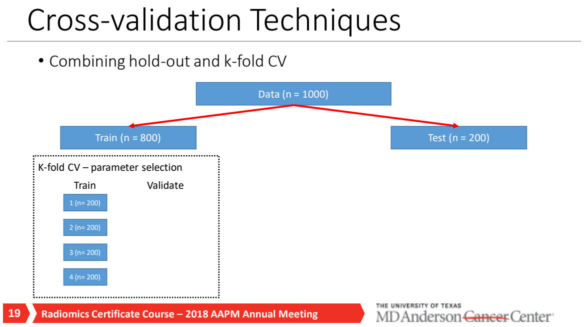
- k-fold cross-validation
  - More test → stable estimate of the model error
  - Useful when not enough data is available



Algorithm 2 K-fold cross-validation

```
1. Input: dataset $T$, number of folds $k$, performance function $error$, computational models
        $L_1, \cdots, L_m, m \geq 1$
2. Divide $T$ into $k$ disjoint subsets $T_1, \cdots, T_k$ of the same size.
3. For $i = 1, \cdots, k$:
   $T_v \leftarrow T_i, T_{tr} \leftarrow \{T \setminus T_i\}$.
   3.1. For $j = 1, \cdots, m$:
        Train model $L_j$ on $T_{tr}$ and periodically use $T_v$ to asses the model performance:
        $E_v^j(i) = error(L_j(T_v))$.
        Stop training, when a stop-criterion based on $E_v^j(i)$ is satisfied.
4. For $j = 1, \cdots, m$, evaluate the performance of the models by: $E_v^j = \frac{1}{k} \cdot \sum_{i=1}^{k} E_v^j(i)$.
```

Resource: https://www.mff.cuni.cz/veda/konference/wds/proc/pdf10/WDS10_105_i1_Reitermanova.pdf



Cross-validation Techniques

- Combining hold-out and k-fold CV

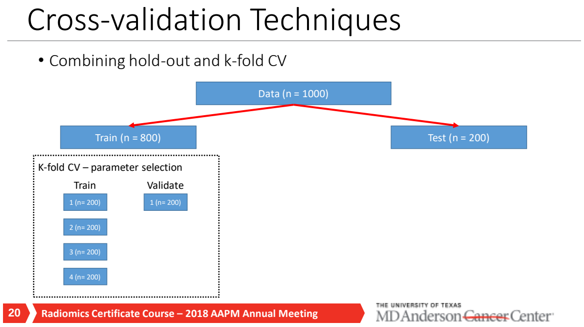
Data (n = 1000)

Train (n = 800)

Test (n = 200)

K-fold CV – parameter selection

Train	Validate
1 (n= 200)	
2 (n= 200)	
3 (n= 200)	
4 (n= 200)	



Cross-validation Techniques

- Combining hold-out and k-fold CV

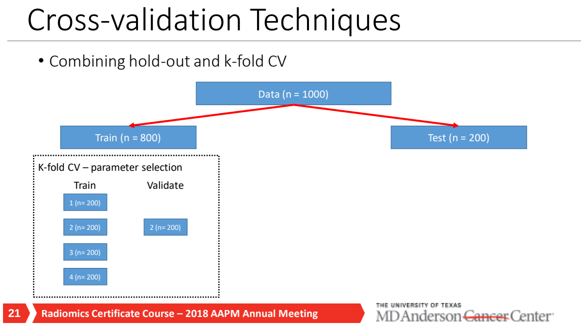
Data (n = 1000)

Train (n = 800)

Test (n = 200)

K-fold CV – parameter selection

Train	Validate
1 (n= 200)	1 (n= 200)
2 (n= 200)	
3 (n= 200)	
4 (n= 200)	



Cross-validation Techniques

- Combining hold-out and k-fold CV

Data (n = 1000)

Train (n = 800)

Test (n = 200)

K-fold CV – parameter selection

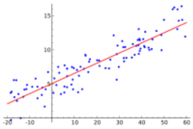
Train	Validate
1 (n= 200)	
2 (n= 200)	2 (n= 200)
3 (n= 200)	
4 (n= 200)	

Machine Learning Models

Linear Regression

$$Y \approx \beta_0 + \beta_1 X$$

Coefficients: intercept slope



Estimation of Parameters:

Residuals: $e_i = y_i - \hat{y}_i$

Residual Sum of Squares: $RSS = e_1^2 + e_2^2 + \cdots + e_n^2$

Or $\boxed{RSS = (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + \cdots + (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2}$ Minimize RSS

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \qquad \hat{\beta}_0 = \hat{y} - \hat{\beta}_1 \bar{x}$$

Linear Regression

- Assessing the accuracy of the model
  - R-squared or fraction of variance explained

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}$$

$$\text{TSS} = \textstyle\sum_{i=1}^{n}(y_i - \bar{y})^2 \qquad RSS = \textstyle\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

  - Residual Sum of Squares

$$\text{RSE} = \sqrt{\frac{1}{n-2}\text{RSS}}$$

Linear Regression

- Assumptions
  - Linear relationship
    - Residual plots
  - Multivariate normality (predictors are normally distributed)
    - Goodness of fit test (Kolmogorov-Smirnov test)
  - No or little multicollinearity
    - Correlation matrix (Pearson's), Tolerance, and variance inflation factor (VIF)
  - No auto-correlation
    - Time-series data
  - Homoscedasticity (residuals are equal across the regression line)
    - Scatter plots

Logistic Regression

- Less assumptions than linear regression
- Some still apply
  - Observations to be independent of each other
  - Little or no multicollinearity
  - Linearity of independent variables and log odds
  - Larger sample size is useful
- The only "real" limitation on logistic regression is that the outcome must be discrete

Logistic Regression

- Discrete Outcomes (Classification Problem)

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$

Logistic Function

Estimation of Parameters:
Maximum likelihood:

$$\boxed{\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1-p(x_{i'}))}$$

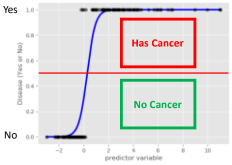
Likelihood Function

Logistic Regression

- Discrete Outcomes (Classification Problem)

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$

Logistic Function



Logistic Regression

- Limitations
  - Unstable with well separated classes
  - Unstable with few examples

Discriminant Analysis

- Classification algorithm that estimates Bayesian classification

Sample average

$$\hat{\mu}_k = \frac{1}{n_k}\sum_{i:y_i=k} x_i$$

Sample variance

$$\hat{\sigma}^2 = \frac{1}{n-K}\sum_{k=1}^{K}\sum_{i:y_i=k}(x_i-\hat{\mu}_k)^2$$

Prior probabilities

$$\hat{\pi}_k = n_k/n$$

LDA: $\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)$ Discriminant function

$$\delta_k(x) = x^T \mathbf{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k^T \mathbf{\Sigma}^{-1}\mu_k + \log \pi_k$$

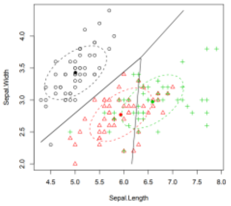
Multi-variable solution

$$\hat{\Sigma}_k := \frac{1}{\hat{n}_k}\sum_{i:y_i=k}(x_i-\hat{\mu}_k)(x_i-\hat{\mu}_k)^T$$

Discriminant Analysis

- Discriminant analysis predicts as follows:

$$\hat{Y}|X = x := \text{argmax}_k\, \pi_k p_k(x) = \text{argmax}_k\, \delta_k(x)$$



Discriminant Analysis

- Key Assumptions
  - Each class density is multivariate Gaussian

$$X|Y_j \sim N(\mu_j, \Sigma_j), \quad j = 0, 1$$

  - Equal covariance

$$\Sigma_j = \Sigma, \quad j = 0, 1$$

  - No outliers

Ridge Regression

- Like least squares linear regression but *shrinks* the estimated coefficients towards zero
- Very useful when multicollinearity (near-linear relationships among the independent variables) occurs
- Given a response vector $y \in \mathbb{R}^n$ and a predictor matrix $X \in \mathbb{R}^{n\times p}$, the ridge regression coefficients are defined as

$$\hat{\beta}^{\text{ridge}} = \underset{\beta\in\mathbb{R}^p}{\text{argmin}} \sum_{i=1}^{n}(y_i - x_i^T\beta)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

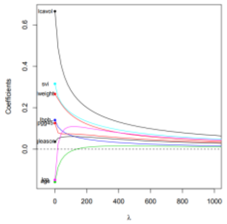
$$= \underset{\beta\in\mathbb{R}^p}{\text{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}} \longrightarrow \text{L}_2 \text{ norm}$$

- Where λ ≥ 0 is a tuning parameter that controls the strength of the penalty term

Ridge Regression

- Cannot perform variable selection
  - Coefficients are reduced close to zero, but not zero (unless λ = ∞, where all coefficients are zero)

  → Low interpretability



Ridge Regression

- **Variable standardization is the first step when using ridge regression!**

- Assumptions
  - Linearity
  - Constant variance (no outliers)
  - Independence

Lasso

- The Lasso combines some of the shrinking advantages of ridge regression with variable selection
- It is very competitive with the ridge regression in regards to prediction error
- The only difference between the two is that ridge regression uses ℓ_2 norm penalty where the lasso uses the ℓ_1 norm penalty
- While the ℓ_1 and ℓ_2 norm look very similar, the ridge and lasso solutions behave very differently

Lasso

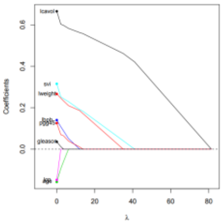
- The Lasso (Least Absolute Selection and Shrinkage Operator) is defined

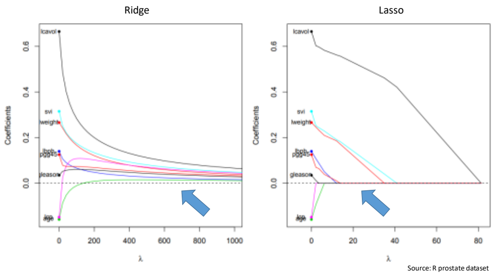
$$\hat{\beta}^{\text{lasso}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \|y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$
$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_1}_{\text{Penalty}} \longrightarrow L_1 \text{ norm}$$

- Again, we have a tuning parameter λ that controls the amount of regularization
- As usual, assume $X^{n \times p}$ is standardize and **y** is centered

Lasso

- Often, we believe that many of the β_j 's should be 0
- Therefore, we would like to have a set of **sparse solutions**
- Large enough λ will set some coefficients exactly equal to 0!
  - So the LASSO will perform variable selection for us!





Source: R prostate dataset

Lasso

- Limitation
  - Cases where p >> n the lasso selects at most n variables before it saturates
    - Convex optimization problem
    - Curse of dimensionality
- Solution
  - Elastic Nets (Ridge + Lasso)
  - Zou, Hui; Hastie, Trevor (2005). "Regularization and Variable Selection via the Elastic Net". *Journal of the Royal Statistical Society*. Series B. Wiley. **67**(2): 301–20. JSTOR 3647580

Classification w/ Ridge and Lasso

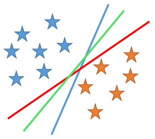
- Add a penalty to the logistic function

$$l_p(\beta_0;\beta;\lambda) = -l(\beta_0;\beta) + \lambda J(\beta)$$

- Where
  - $l(\beta_0;\beta)$ denotes the unrestricted log-likelihood function
  - λ is the regularization parameter controlling the amount of shrinkage
  - $J(\beta)$ is a penalty function on the coefficient parameter β
    - Either the lasso or ridge penalty functions

Support Vector Machines

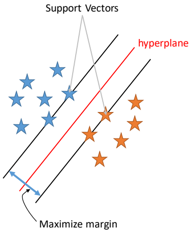
- Goal: find an optimal hyperplane
- Support Vectors
  - Data points that lie closest to the decision surface (or hyperplane)
  - Most difficult data points to classify
  - They have a direct relationship on the optimal location of the hyperplane



- In general, there are lots of possible solutions
- Support Vector Machine (SVM) finds an optimal solution

Support Vector Machines

- SVMs <u>maximize</u> the margin around the separating hyperplane
- The decision function is fully specified by a subset of training samples
  - Support Vectors
- Real-world data (non-separable)
  - Soft margin classifier
    - OK to misclassify a few training observations in order to do a better job in classifying the remaining observations



Support Vector Machines

$$\text{maximize } f(c_1 \ldots c_n) = \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i c_i (\varphi(\vec{x}_i)\cdot\varphi(\vec{x}_j)) y_j c_j$$

$$= \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i c_i \boxed{k(\vec{x}_i,\vec{x}_j)} y_j c_j$$

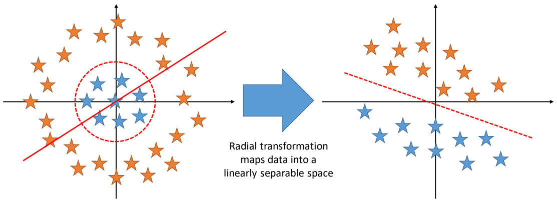
Kernel

$$\text{subject to } \sum_{i=1}^{n} c_i y_i = 0, \text{ and } 0 \le c_i \le \frac{1}{2n\lambda} \text{ for all } i.$$

- The Kernel Trick
  - Linear: $k(\vec{x_i},\vec{x_j}) = (\vec{x_i}\cdot\vec{x_j})$
  - Polynomial: $k(\vec{x_i},\vec{x_j}) = (\vec{x_i}\cdot\vec{x_j})^p$
  - Gaussian radial basis function: $k(\vec{x_i},\vec{x_j}) = e^{-\gamma\|\vec{x_i}-\vec{x_j}\|^2}$ for $\gamma > 0$
  - Hyperbolic tangent: $k(\vec{x_i},\vec{x_j}) = \tanh(\kappa\vec{x_i}\cdot\vec{x_j} + c)$ for some $\kappa > 0$ and $c < 0$

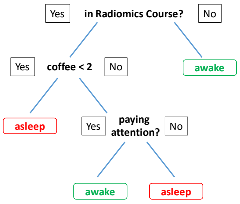
Support Vector Machines

- Kernel Trick
  - Useful when the decision function is not linear


Radial transformation maps data into a linearly separable space

Decision Trees

- CART (Classification and Regression Trees, *Breiman et al 1984*)
- A decision tree is drawn upside down with its root at the top
  - Node
  - Branches
  - Leaf (decisions)



in Radiomics Course?

Yes

No

coffee < 2

Yes

No

awake

asleep

paying attention?

Yes

No

awake

asleep

Decision Trees

How does the algorithm decide which features are more important (top of the tree) and where to create a split?

- Recursive binary splitting
  - All the features are considered and different split points are tried and tested using a cost function
  - The split with the lowest cost is selected
  - "Greedy algorithm"
- Cost functions
  - Regression: Mean Square Error
  - Classification: Gini, Entropy, etc.

Decision Trees

Improving prediction accuracy

- Assigning a maximum depth
  - Length of the longest path from the root to leaf
- Leaf size
  - Set a minimum number of training inputs to use on each leaf
- Pruning
  - Removal of branches that make use of features having low importance

Reduces complexity of the tree → increased predictive power by reducing overfitting

- In theory, the depth of the tree is limited by the number of training examples and can be extremely deep

Decision Trees

- Advantages
  - Simple to understand, interpret, visualize
  - Handles both numerical and categorical data
  - Nonlinear relationships between parameters do not affect tree performance
- Disadvantages
  - Decision tree learners can create over-complex trees → overfitting
  - Decision tree learners create biased trees if some classes dominate
    - Balance the data set prior to training
  - Greedy algorithms cannot guarantee to return the globally optimal decision tree → mitigated by ensemble learning (multiple trees)
    - Adaboost, Bagging, Random Forest, etc.

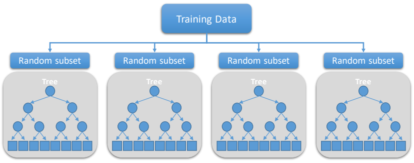
Random Forest

Just imagine…
a lot of decision trees

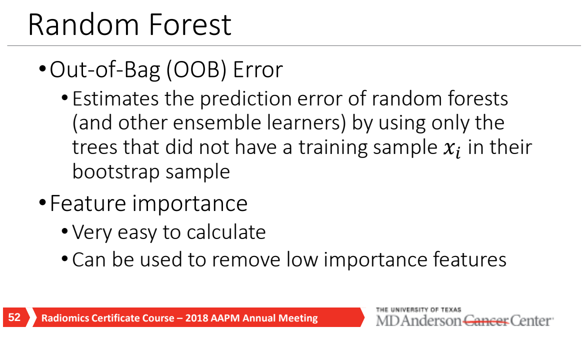
https://www.eustafor.eu/

Random Forest

- Adds additional randomness to the model

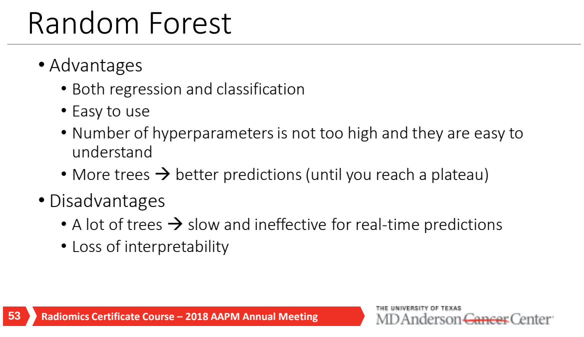


1. For each tree: randomly selects a subset of training data (~66%)
2. At each node: randomly selects a subset of predictor variables (~$\sqrt{p}$)



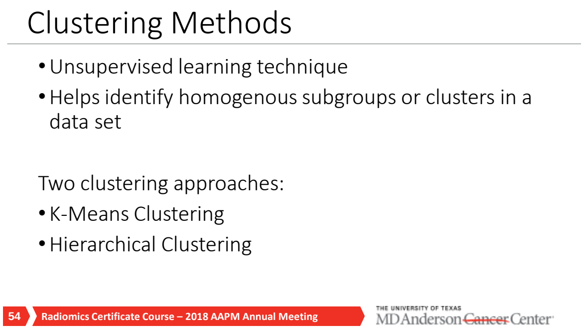
Random Forest

- Out-of-Bag (OOB) Error
  - Estimates the prediction error of random forests (and other ensemble learners) by using only the trees that did not have a training sample x_i in their bootstrap sample
- Feature importance
  - Very easy to calculate
  - Can be used to remove low importance features



Random Forest

- Advantages
  - Both regression and classification
  - Easy to use
  - Number of hyperparameters is not too high and they are easy to understand
  - More trees → better predictions (until you reach a plateau)
- Disadvantages
  - A lot of trees → slow and ineffective for real-time predictions
  - Loss of interpretability



Clustering Methods

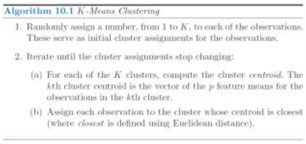
- Unsupervised learning technique
- Helps identify homogenous subgroups or clusters in a data set

Two clustering approaches:

- K-Means Clustering
- Hierarchical Clustering

K-Means Clustering

- How it works

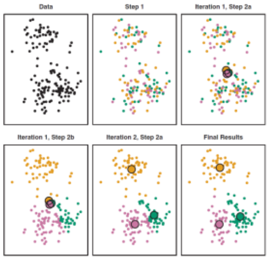


```
Algorithm 10.1 K-Means Clustering
1. Randomly assign a number, from 1 to $K$, to each of the observations.
   These serve as initial cluster assignments for the observations.
2. Iterate until the cluster assignments stop changing:
    (a) For each of the $K$ clusters, compute the cluster centroid. The
        $k$th cluster centroid is the vector of the $p$ feature means for the
        observations in the $k$th cluster.
    (b) Assign each observation to the cluster whose centroid is closest
        (where closest is defined using Euclidean distance).
```

Source: An Introduction to Statistical Learning, Witten et al, 2013

Objective Function

$$\underset{C_1,\ldots,C_K}{\text{minimize}}\left\{\sum_{k=1}^{K}\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2\right\}$$

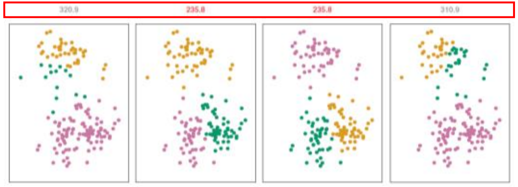


Source: An Introduction to Statistical Learning, Witten et al, 2013

K-Means Clustering

- Limitations
  - Random initialization & local optima. R

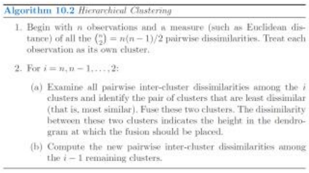
Objective Values



Source: An Introduction to Statistical Learning, Witten et al, 2013

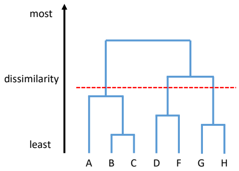
Hierarchical Clustering

- Creates a dendrogram



```
Algorithm 10.2 Hierarchical Clustering
1. Begin with $n$ observations and a measure (such as Euclidean dis-
   tance) of all the $\binom{n}{2} = n(n-1)/2$ pairwise dissimilarities. Treat each
   observation as its own cluster.
2. For $i = n, n-1, \ldots, 2$:
    (a) Examine all pairwise inter-cluster dissimilarities among the $i$
        clusters and identify the pair of clusters that are least dissimilar
        (that is, most similar). Fuse these two clusters. The dissimilarity
        between these two clusters indicates the height in the dendro-
        gram at which the fusion should be placed.
    (b) Compute the new pairwise inter-cluster dissimilarities among
        the $i-1$ remaining clusters.
```

Source: An Introduction to Statistical Learning, Witten et al, 2013



Clustering Methods

- Things to remember when using clustering algorithm:
  - Standardizing variables so that all are on the same scale. It is important when calculating distances
  - Treat data for outliers before forming clusters as it can influence the distance between the data points.

Honorable Mention List

- Principal Component Analysis
- Other Ensemble Methods
  - Bagging and Boosting
- Nearest Neighbor
- Naïve-Bayes Classifier
- Bayesian Networks

“Some food for thought”

- George Box (1919 – 2013)

“Essentially, all models are wrong, but some are useful”

Final Remarks

- Proper splitting of datasets leads to better generalization
  - No peaking at the test set!
- No one algorithm works best for every problem
  - "No free lunch in Machine Learning"
- Think about the inputs in your model
  - Intuition and knowledge about the data can prevent head-aches
- Develop a good understanding about mathematical principals behind your algorithm of choice!
  - Identify the strengths and weaknesses

Thank you!

Carlos E. Cardenas, PhD
cecardenas@mdanderson.org