Statistics and Machine Learning

- Author:
  - Yiwei Ang (Last edit 20th April 2021)
- This is to compile:
  - Traditional statistical testing
  - Modern Machine Learning Implication
  - Way to Choose Between Model
- Useful Cheatsheet:
  - Data Science Cheatsheet
  - Probability Cheatsheet

Statistics and Machine Learning

- The study of computer algorithms that improve automatically through **experience** and by the use of data.
- Involves computers discovering how they can perform tasks without being explicitly programmed to do so.

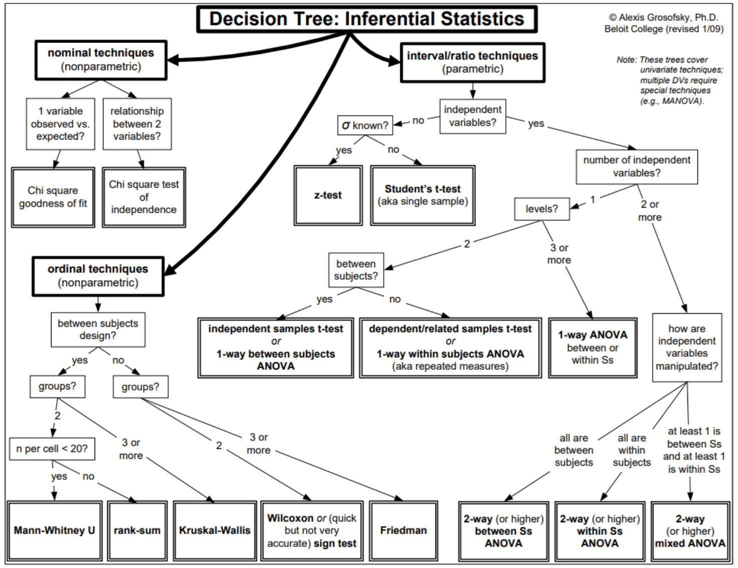
Machine learning approaches

- Machine learning approaches are traditionally divided into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system:

Types

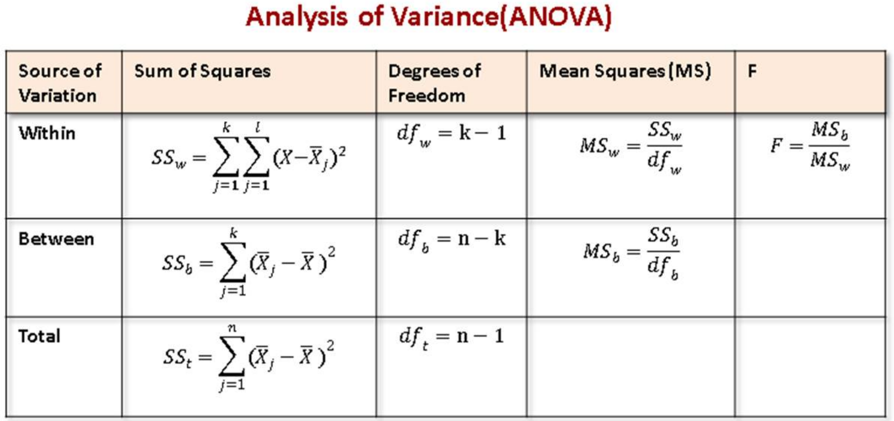
- **Supervised learning**: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.
- **Unsupervised learning**: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).
- **Reinforcement learning**: A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent). As it navigates its problem space, the program is provided feedback that's analogous to rewards.

Statistic


Decision Tree: Inferential Statistics
© Alexis Grosofsky, Ph.D.
Beloit College (revised 1/09)
Note: These trees cover univariate techniques; multiple DVs require special techniques (e.g., MANOVA).
nominal techniques (nonparametric)
1 variable observed vs. expected?
relationship between 2 variables?
Chi square goodness of fit
Chi square test of independence
interval/ratio techniques (parametric)
independent variables?
no
σ known?
yes
no
z-test
Student's t-test (aka single sample)
yes
number of independent variables?
1
levels?
2 or more
2
between subjects?
yes
no
independent samples t-test or 1-way between subjects ANOVA
dependent/related samples t-test or 1-way within subjects ANOVA (aka repeated measures)
3 or more
1-way ANOVA between or within Ss
how are independent variables manipulated?
all are between subjects
all are within subjects
at least 1 is between Ss and at least 1 is within Ss
2-way (or higher) between Ss ANOVA
2-way (or higher) within Ss ANOVA
2-way (or higher) mixed ANOVA
ordinal techniques (nonparametric)
between subjects design?
yes
no
groups?
2
n per cell &lt; 20?
yes
no
3 or more
groups?
2
3 or more
Mann-Whitney U
rank-sum
Kruskal-Wallis
Wilcoxon or (quick but not very accurate) sign test
Friedman

Commmon Statistical Test - ANOVA

- Analysis of Variance, where H_0 is that means of group are the same.



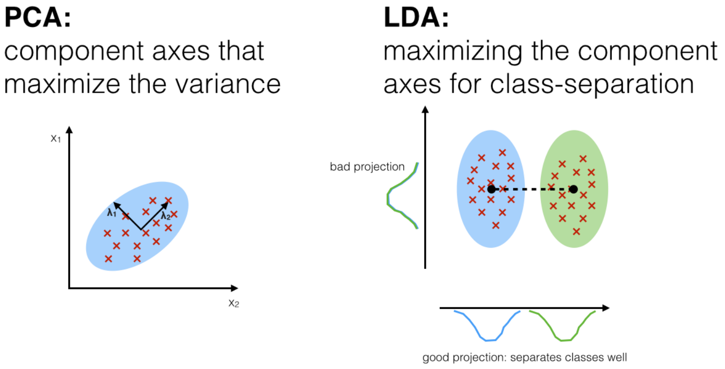
Analysis of Variance(ANOVA)

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Squares(MS)	F
Within	$SS_w = \sum_{j=1}^{k}\sum_{j=1}^{l}(X-\bar{X}_j)^2$	$df_w = \text{k} - 1$	$MS_w = \frac{SS_w}{df_w}$	$F = \frac{MS_b}{MS_w}$
Between	$SS_b = \sum_{j=1}^{k}(\bar{X}_j - \bar{X})^2$	$df_b = \text{n} - \text{k}$	$MS_b = \frac{SS_b}{df_b}$	
Total	$SS_t = \sum_{j=1}^{n}(\bar{X}_j - \bar{X})^2$	$df_t = \text{n} - 1$		

Suppervised Learning

1. Linear Discriminant Analysis

- Linear Discriminant Analysis, or LDA, uses the information from both features to create a new axis and projects the data on to the new axis in such a way as to:
  - Minimizes the variance
  - maximizes the distance between the **means** of the two classes.
- LDA is closely related to analysis of variance (ANOVA) and regression analysis, which also attempt to express one dependent variable as a linear combination of other features or measurements.
  - ANOVA uses **categorical independent variables** and **a continuous dependent variable**,
  - LDA has no restrictions.
- Closely related to principal component analysis (PCA) and factor analysis in that they both look for linear combinations of variables which best explain the data.
  - LDA explicitly attempts to model the difference between the classes of data.
  - PCA, in contrast, does not take into account any difference in class.
  - Factor analysis builds the feature combinations based on differences rather than similarities.


PCA:
component axes that maximize the variance
x_1
x_2
λ_1
λ_2
LDA:
maximizing the component axes for class-separation
bad projection
good projection: separates classes well

- Assumptions:
  - Independent variables are normal.
  - Homogeneity of variance.
  - Independence.
- Computation of eigenvectors and values (Higher the better)

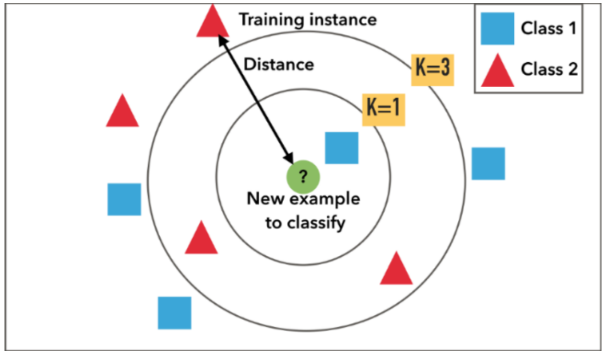
2. K-nearest neighbours

- One of the most simplest ways to classify data k in K-Nearest Neighbors is the number of neighbor.
- k in K-Nearest Neighbors is the number of neighbor.
- Assumption: Similar inputs have similar outputs.
- **Distance** is used: Euclidean mostly.
- For dimension > 10, usually dimension reduction is performed prior to applying knn to avoid the effct of curse of dimensionality.

  - As number of features grows, the amount of data we need to **generalize accurately** grows **exponentially**.

Classification

- Output is a class membership.
- Class is assigned to class where it is given most of k neighbours vote.
- Mode of k labels is returned.



Regression

- Output is the average of the values of k nearest neighbors.
- Mean of k labels is returned.

Advantage

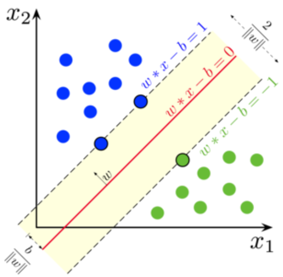
- Easy to interpret and implement
- Does not make any assumption.

Disadvantage

- When number of fetures increase/samples increase, it will be come slower.
- Sensitive to outliers.

3. Support Vector

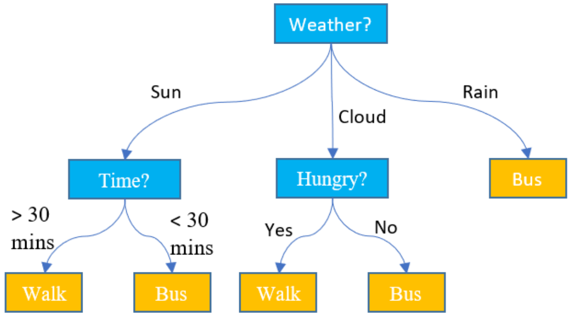
- Question: What is the best separating hyperplane?
  - The one that **maximizes the distance** to the closest data points from both classes.
  - We say it is the **hyperplane** with maximum **margin**.
- Can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. ### Classification



4. Decision Tree

- A decision tree is a decision support tool that uses a tree-like model.
- One of the **supervised** predictive modelling approaches
- It is one way to display an algorithm that only contains conditional control statements.

- Two main types:
  - Classification tree analysis is when the predicted outcome is the class (discrete) to which the data belongs.
  - Regression tree analysis is when the predicted outcome can be considered a real number (e.g. the price of a house, or a patient's length of stay in a hospital).



- Some techniques, often called ensemble methods, construct more than one decision tree:

- Boosted trees:
  - Incrementally building an ensemble by training each new instance to emphasize the training instances previously mis-modeled.
  - A typical example is AdaBoost.
  - These can be used for regression-type and classification-type problems.
- Bootstrap aggregated (or bagged) decision trees, an early ensemble method, builds multiple decision trees by repeatedly resampling training data with replacement, and voting the trees for a consensus prediction.[9]
  - A random forest classifier is a specific type of bootstrap aggregating
- Rotation forest – in which every decision tree is trained by first applying principal component analysis (PCA) on a random subset of the input features.[10]

Classification And Regression Tree (CART) - Gini Index

- Choose feature which has minimum Gini Index to split

$$Gini\ Index = 1 - \sum_{i=1}^{n} p_i^2$$

ID3, C4.5 and C5.0 tree-generation algorithms - Information Gain

- The best first split is the one that provides the most information gain

Entropy is defined as below

$$\mathrm{H}(T) = \mathrm{I}_E(p_1, p_2, \ldots, p_J) = -\sum_{i=1}^{J} p_i \log_2 p_i$$

where $p_1, p_2, \ldots$are fractions that add up to 1 and represent the percentage of each class present in the child node that results from a split in the tree.[20]

$$\overbrace{IG(T,a)}^{\text{Information Gain}} = \overbrace{\mathrm{H}(T)}^{\text{Entropy (parent)}} - \overbrace{\mathrm{H}(T|a)}^{\text{Sum of Entropy (Children)}}$$

$$= -\sum_{i=1}^{J} p_i \log_2 p_i - \sum_{i=1}^{J} -\Pr(i|a) \log_2 \Pr(i|a)$$

Averaging over the possible values of A,

$$\overbrace{E_A(IG(T,a))}^{\text{Expected Information Gain}} = \overbrace{I(T;A)}^{\text{Mutual Information between T and A}} = \overbrace{\mathrm{H}(T)}^{\text{Entropy (parent)}} - \overbrace{\mathrm{H}(T|A)}^{\text{Weighted Sum of Entropy (Children)}}$$

$$= -\sum_{i=1}^{J} p_i \log_2 p_i - \sum_{a} p(a) \sum_{i=1}^{J} -\Pr(i|a) \log_2 \Pr(i|a)$$

Advantages

- Simple to understand and interpret.
  - People are able to understand decision tree models after a brief explanation.
  - Trees can also be displayed graphically.
- Able to handle both **numerical and categorical** data.
- Requires little data preparation.
  - Other techniques often require data normalization.
  - Since trees can handle **qualitative** predictors, there is no need to create dummy variables.
- Uses a white box or open-box model. If a given situation is observable in a model the explanation for the condition is easily explained by boolean logic.
- Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model.

- Non-statistical approach that makes **no assumptions** of the training data or prediction residuals; e.g., no distributional, independence, or constant variance assumptions.
- Mirrors human decision making more closely than other approaches. This could be useful when modeling human decisions/behavior.
- Robust against co-linearity, particularly boosting
- In built feature selection. Additional irrelevant feature will be less used so that they can be removed on subsequent runs. The hierarchy of attributes in a decision tree reflects the importance of attributes.[23] It means that the features on top are the most informative.

Limitations

- Trees can be very non-robust. A small change in the training data can result in a large change in the tree and consequently the final predictions.
- The problem of learning an optimal decision tree is known to be NP-complete under several aspects of optimality and even for simple concepts.Consequently, practical decision-tree learning algorithms are based on heuristics such as the greedy algorithm where locally optimal decisions are made at each node. Such algorithms cannot guarantee to return the globally optimal decision tree. To reduce the greedy effect of local optimality, some methods such as the dual information distance (DID) tree were proposed.
- Can create over-complex trees that do not generalize well from the training data. (This is known as overfitting.
    - Mechanisms such as pruning are necessary to avoid this problem.
- The average depth of the tree that is defined by the number of nodes or tests till classification is not guaranteed to be minimal or small under various splitting criteria.

5. Adaptive Boosting Trees

- AdaBoost, short for Adaptive Boosting, is a machine learning meta-algorithm formulated by Yoav Freund and Robert Schapire, who won the 2003 Gödel Prize for their work.

Keywords

- Combination of weak learners such as stumps.
- Create an $\alpha \in R$ for a particular stump: Amount to say(vote).

$$\alpha = \frac{1}{2} log(\frac{1-\epsilon_t}{\epsilon_t})$$

$$0 \leq \epsilon_t \leq 1$$

    - Higher/lower error, α becomes more negatively/positively contributing to voting.
- Adjust the sample weights for correct (-) and incorrect (+) classification respectively:

$$weight_{new} = weight_{old} \times e^{\mp alpha}$$

- New collection of dataset using updated sample weights will be used for next stump creation.

Notes

- In SKLearn, learning rate is defined as:
    - learning rate shrinks the contribution of each classifier by learning_rate

6. Gradient Boosting Trees

- Ensemble of weak prediction models, which the models allow generalization by optimizing an arbitrary differentiable loss function.
- Weak learner are larger than stumps mostly, but limtation of leaves to be between 8-32.
- Building fixed sized tres based on previous tree's error.

Keywords

- Combination of weak learners such as stumps.
- Create an $\alpha \in R$ for a particular stump: Amount to say(vote).

$$\alpha = \frac{1}{2} log(\frac{1-\epsilon_t}{\epsilon_t})$$

$$0 \leq \epsilon_t \leq 1$$

    - Higher/lower error, α becomes more negatively/positively contributing to voting.
- Adjust the sample weights for correct (-) and incorrect (+) classification respectively:

$$weight_{new} = weight_{old} \times e^{\mp alpha}$$

- New collection of dataset using updated sample weights will be used for next stump creation.

Regression

- Starting with an average value.
- Add a tree based on **residuals**: observed - predicted values.

$$Final = Average + learning\_rate \times \sum_{i}^{n}(observed_i - predicted_i)$$

- a) Input:
    - Data: $(x_i, y_i)_{i=1}^{n}$
    - Differentiable Loss Function: $\frac{1}{2}(Observed - Predicted)^2$
- b) Compute

$$r_{im} = -[\frac{\delta(y_i, F(x_i)}{\delta F(x_i}]_{F(x)=F_{m-1}(x)}$$

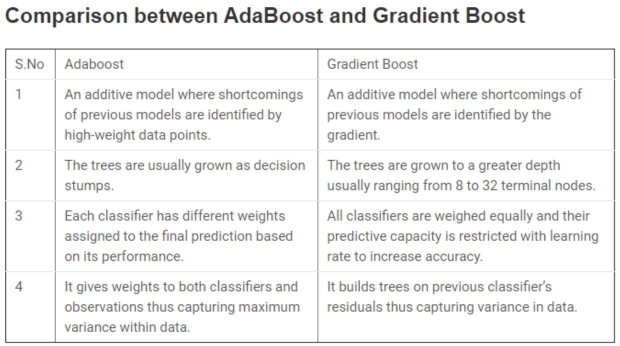
- c) Fit a regressioin tree to r_{im} values arnd create terminal regions R_{jm} for $j = 1, \ldots, J_m$
- d) Fr $j = 1, \ldots, J_m$, compute

$$\gamma_{jm} = argmin \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i + \gamma)$$

- e) Update $F_m(x) = F_{m-1}(x) + \upsilon \sum_{j=1}^{J_m} \gamma I(x \in R_{jm})$
- f) Output $F_M(x)$

Notes

- In SKLearn, learning rate is defined as:
    - learning rate shrinks the contribution of each classifier by learning_rate



Comparison between AdaBoost and Gradient Boost

S.No	Adaboost	Gradient Boost
1	An additive model where shortcomings of previous models are identified by high-weight data points.	An additive model where shortcomings of previous models are identified by the gradient.
2	The trees are usually grown as decision stumps.	The trees are grown to a greater depth usually ranging from 8 to 32 terminal nodes.
3	Each classifier has different weights assigned to the final prediction based on its performance.	All classifiers are weighed equally and their predictive capacity is restricted with learning rate to increase accuracy.
4	It gives weights to both classifiers and observations thus capturing maximum variance within data.	It builds trees on previous classifier's residuals thus capturing variance in data.

7. Random Forest

- One of the emsemble learning method: **bootsrap aggregating or bagging**, by random sampling with replacement.
- One of the **supervised** predictive modelling approaches
- Constructing a multitude of decision trees and:
    - Classification: Mode
    - Regression: Average

Unsupervised Learning

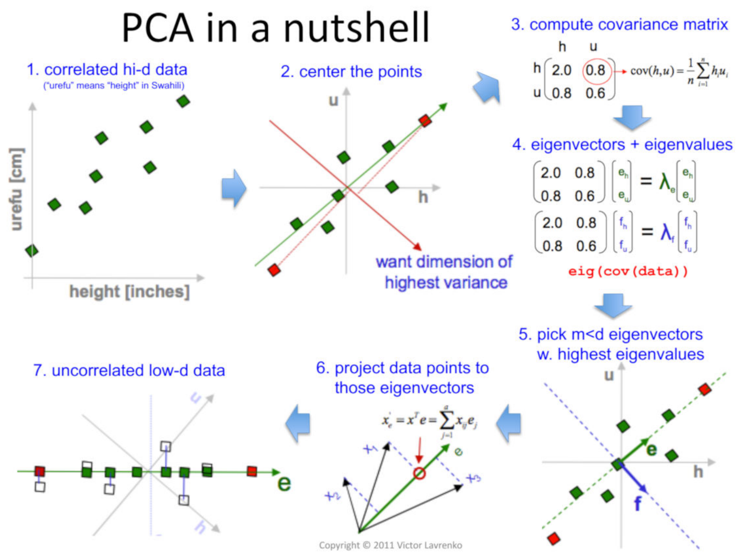
Principal Component Analysis (PCA)

- Projects data onto orthogonal vectors that maximize variance.
- Acts as a dimensionality reduction.
- Can be explained by linear combination of features which maximise variance.

Steps

- Start with the covariance matrix of standardized data
- Calculate eigenvalues and eigenvectors using SVD or eigendecomposition
- Rank the principal components by their proportion of explained variance.

Reference

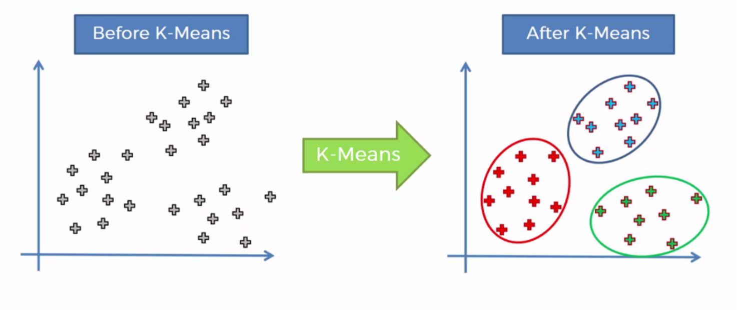

PCA in a nutshell
1. correlated hi-d data
("urefu" means "height" in Swahili)
urefu [cm]
height [inches]
2. center the points
u
h
want dimension of highest variance
3. compute covariance matrix
h u
h 2.0 0.8
u 0.8 0.6
$\mathrm{cov}(h,u) = \frac{1}{n}\sum_{i=1}^{n} h_i u_i$
4. eigenvectors + eigenvalues
$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix}\begin{pmatrix} e_h \\ e_u \end{pmatrix} = \lambda_e \begin{pmatrix} e_h \\ e_u \end{pmatrix}$
$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix}\begin{pmatrix} f_h \\ f_u \end{pmatrix} = \lambda_f \begin{pmatrix} f_h \\ f_u \end{pmatrix}$
eig(cov(data))
5. pick m<d eigenvectors w. highest eigenvalues
u
e
h
f
6. project data points to those eigenvectors
$x_e' = x^T e = \sum_{j=1}^{d} x_j e_j$
7. uncorrelated low-d data
e
Copyright © 2011 Victor Lavrenko

K-means Clustering

- Aims to partition n observations into k cluster as to minimize within-cluster sum of squares

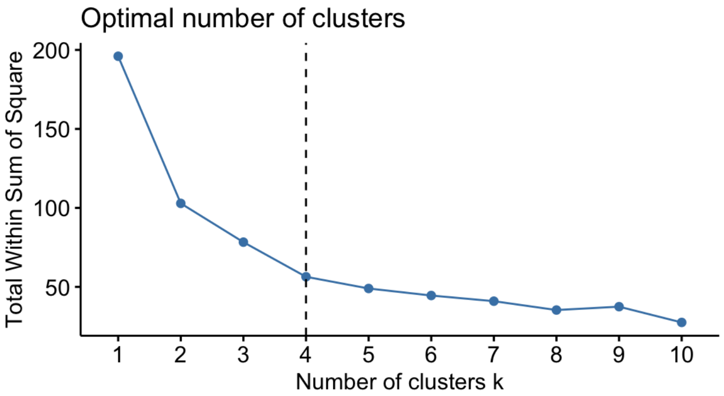
Steps

- Choose K
- Random select k data points
- Measure the distance between points and the k data points
- Assign points to nearest cluster
- Calculate the mean of each cluster and repeat



Pick K

- Use elbow plot: K is chosen where it gives drops in reduction of of variation.



Bias-Variance Trade-Off

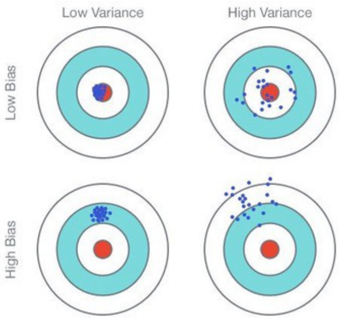


Fig. 1: Graphical Illustration of bias-variance trade-off , Source: Scott Fortmann-Roe., Understanding Bias-Variance Trade-off

- The goal of any supervised machine learning algorithm is to achieve **low bias** and **low variance**.
- In turn the algorithm should achieve good prediction performance

Bias

- **Low Bias**: Suggests less assumptions about the form of the target function.
- **High-Bias**: Suggests more assumptions about the form of the target function.

Variance

- Variance is the amount that the estimate of the target function will change if different training data was used.
- Strongly influenced by the specifics of the training data
- **Low Variance**: Suggests small changes to the estimate of the target function with changes to the training dataset.
- **High Variance**: Suggests large changes to the estimate of the target function with changes to the training dataset.

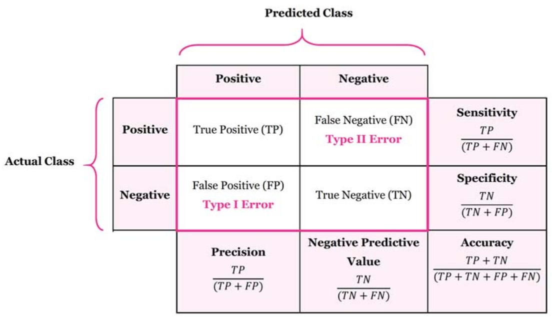
General Trend

- **Linear** machine learning algorithms often have a high bias but a low variance.
- **Nonlinear** machine learning algorithms often have a low bias but a high variance.

Configuring Trade Off

- The k-nearest neighbors algorithm has low bias and high variance, but the trade-off can be changed by increasing the value of k which increases the number of neighbors that contribute t the prediction and in turn increases the bias of the model.
- The support vector machine algorithm has low bias and high variance, but the trade-off can be changed by increasing the C parameter that influences the number of violations of the margin allowed in the training data which increases the bias but decreases the variance

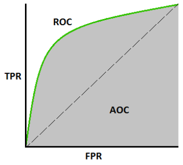
Confusion Matrix



		Predicted Class		
		Positive	**Negative**	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

AUC-ROC Curve

- Performance measurement for the classifcation problem.
- Receivor Operating Characteristics (ROC) is the probability curve.
- Area Under Curve (AUC) degree of seperability.



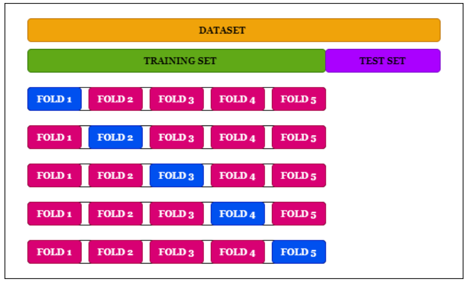
$$TPR\ /\ Recall\ /\ Sensitivity = \frac{TP}{TP+FN}$$

$$FPR = \frac{FP}{TN+FP} = 1 - Specificity = \frac{FP}{TN+FP}$$

Usage

- To visualize the threshold across accuaracies.
- Choose between models.

Cross Validation



- Compare different ML models and get some understanding how well they will work in practice.

Types

- Leave One Out Cross Validation: Extreme case where one sample is used for testing while others for training.
- 10 fold CV: Usual

```
In [ ]:
```