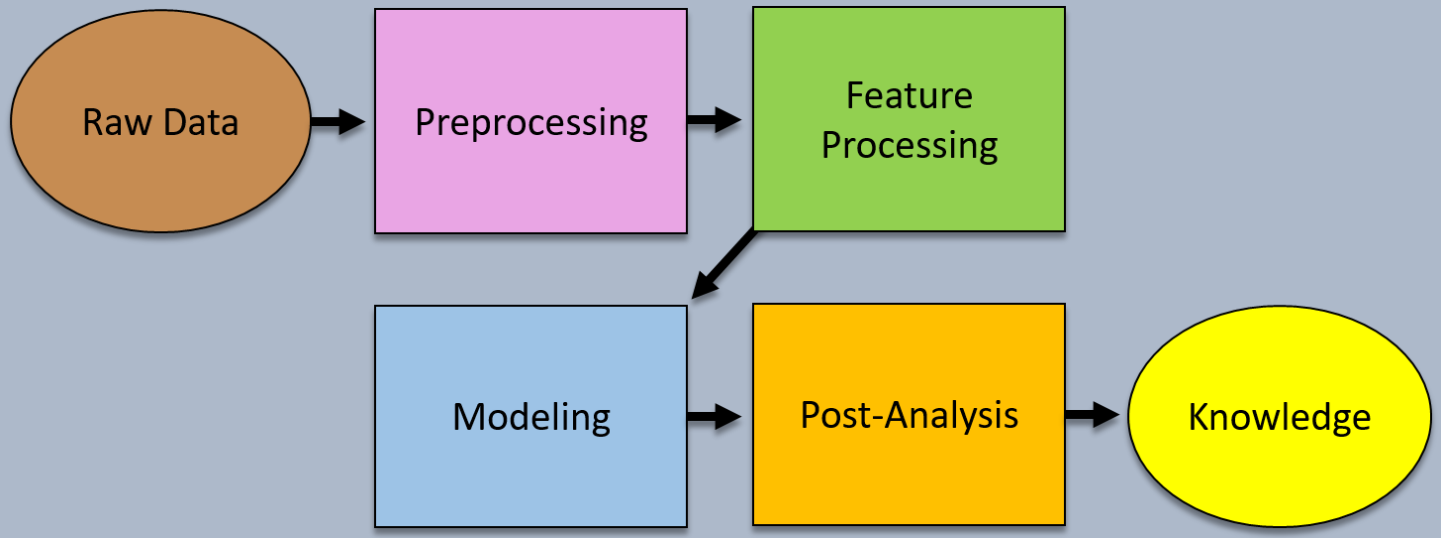
Machine Learning: Building an Analysis Pipeline



Ryan Urbanowicz, PhD

Perelman School of Medicine
University of Pennsylvania

Overview

- Brief Review of Machine Learning 101
- Stepping through a Machine Learning Analysis Pipeline
  - Biomedical Classification
  - Paired with Jupyter Notebook

- PHASE 1 (Preprocessing)
  - Exploratory Analysis
  - Cleaning
  - Partitioning

- PHASE 2 (Feature Processing)
  - Feature Transformation
  - Feature Construction
  - Feature Selection

- PHASE 3 (Modeling)
  - Algorithm Selection
  - Hyperparameter Sweep
  - Model Evaluation

- PHASE 4 (Post-Analysis)
  - Interpretation
  - Replication

Review

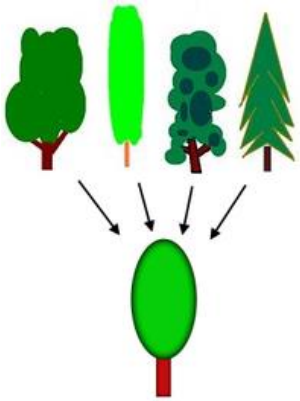
Terminology and Definitions

- Instance: an individual or example in data.
  - E.g. A subject/patient in a drug trial.
- Feature: one of the attributes describing an aspect of the instance. E.g. height, weight, age.
- Outcome: In supervised learning, this is endpoint value, a.k.a. the dependent variable, or the target being predicted.
  - Label/Class: Terms used for outcome in classification.
  - In regression, the outcome would be real-valued numbers.
- Model: A representation or simulation of reality. Typically a simplification based on a number of assumptions.

Machine Learning (ML)

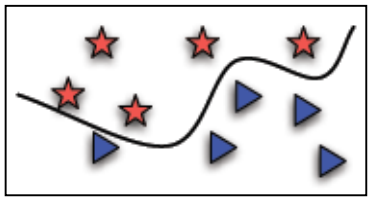
- Computational strategies that, given data, are designed to progressively improve performance on a specific task without being explicitly programmed.

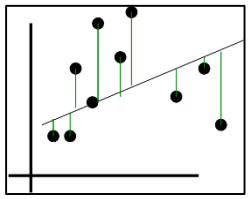
- ML includes many methods/algorithms.



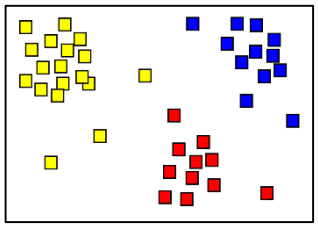
- Big Picture Goal: Learning useful generalizations.

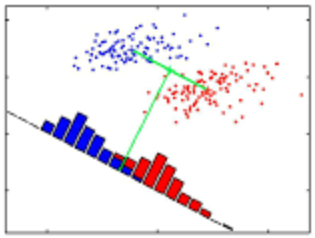
- Supervised Learning Labeled Data



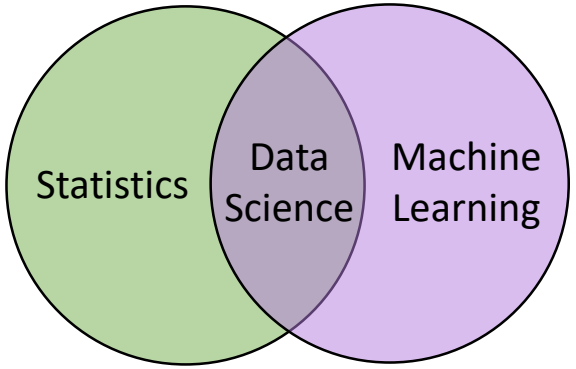


- Unsupervised Learning Unlabeled Data

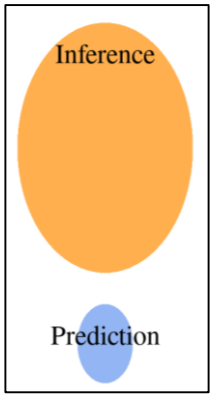




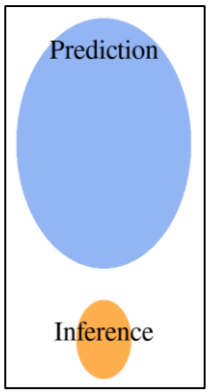
Statistics vs. Machine Learning



- Largely overlapping fields:
  - Both concerned with learning from data
  - Philosophical difference on 'focus' and 'approach'.
- Statistics:
  - Founded in mathematics
  - Drawing valid conclusions based on analyzing existing data.
    - Making inference about a 'population' based on a 'sample'
    - Tends to focus on fewer variables at once.
    - Precision and uncertainty are measures of model goodness.

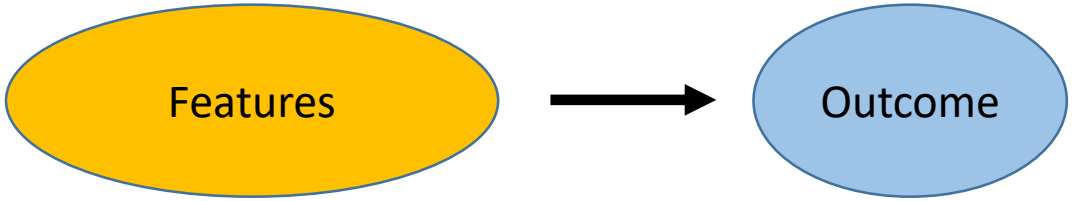


- Machine Learning:
  - Founded in computer science
  - Focused on making predictions or seeking patterns (generalization).
    - Often considers a large number of variables at once.
    - Prediction accuracy to measure model goodness.



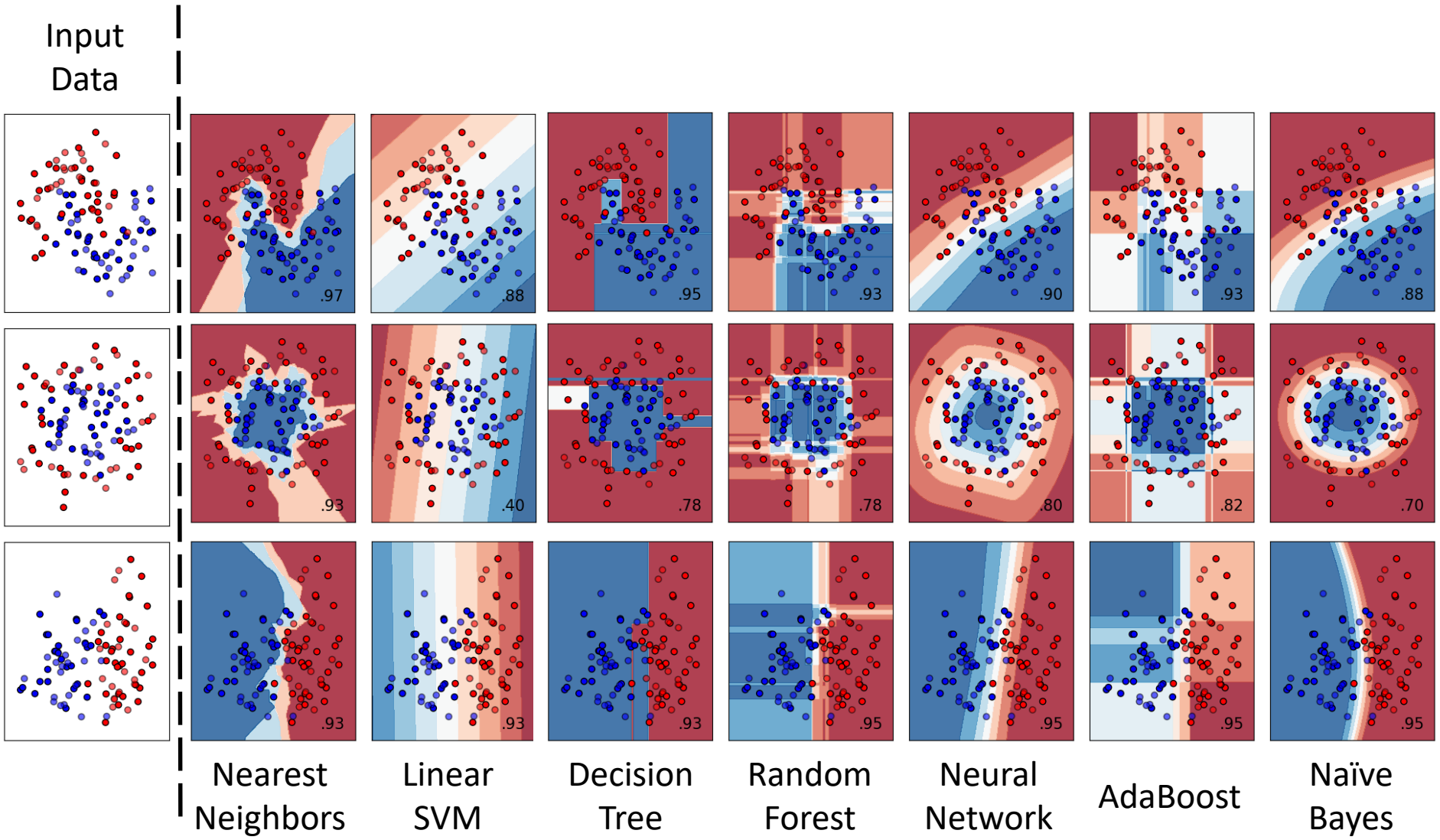
Inference vs. Prediction

- Both:
  - Supervised learning from data in order to find a model that describes the relationship between the features and the outcome.

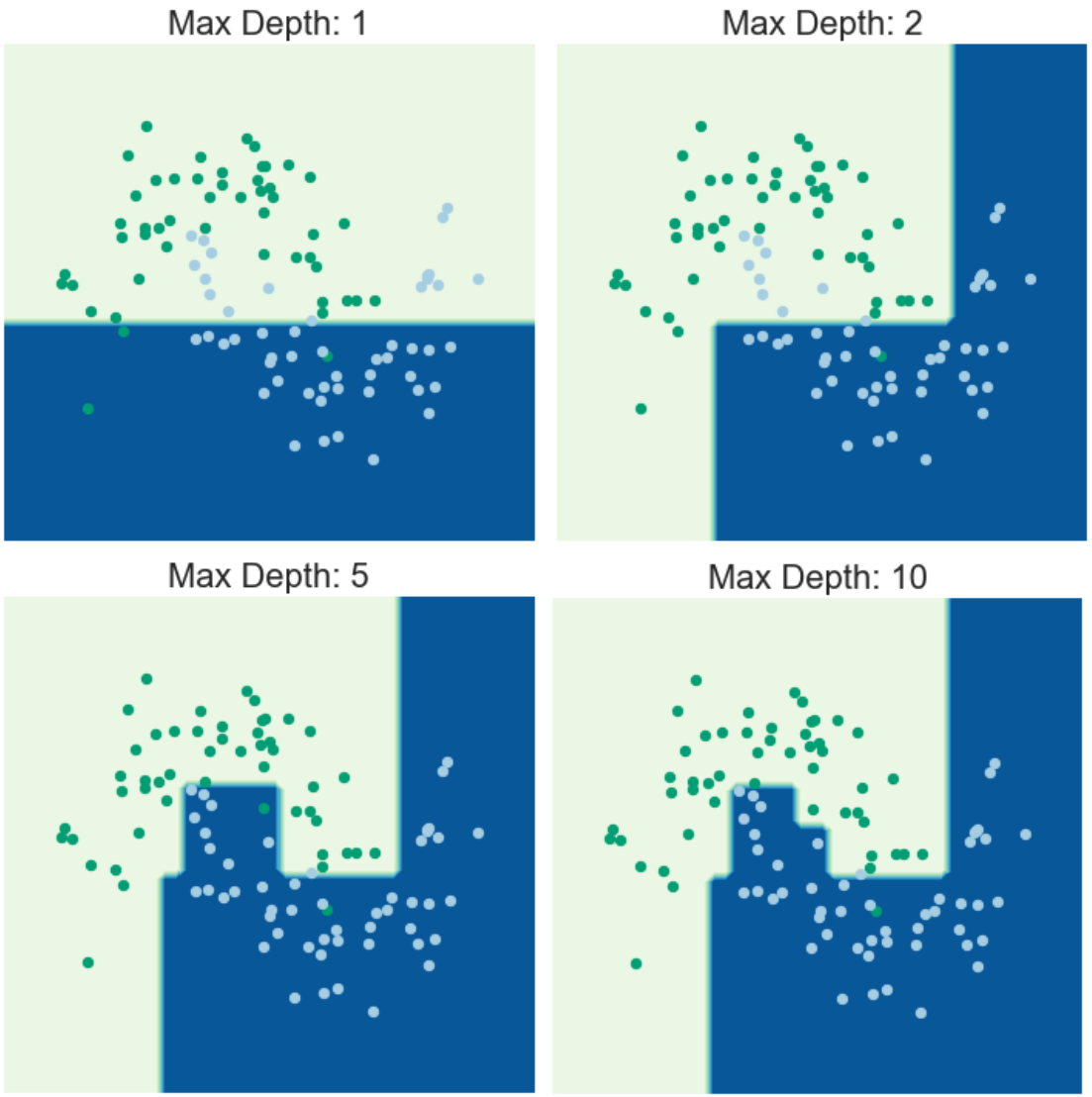


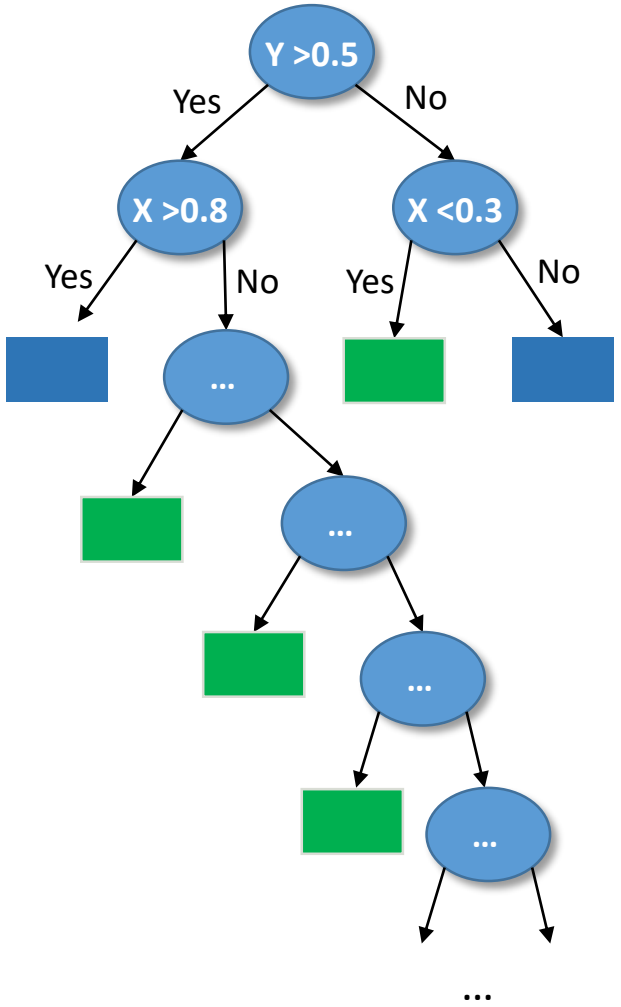
- Inference:
  - Use model to learn how the output is generated as a function of the data.
- Prediction:
  - Use model to predict outcomes for new data points
- Example:
  - Inference:
    - You want to find out what the effect of Age, Passenger Class and, Gender has on surviving the Titanic Disaster. You can put up a logistic regression and infer the effect each feature has on survival rates.
  - Prediction:
    - Given some information on Titanic passengers, you want to choose from the outcome (live or die) and be correct as often as possible.

Modeling with Machine Learning



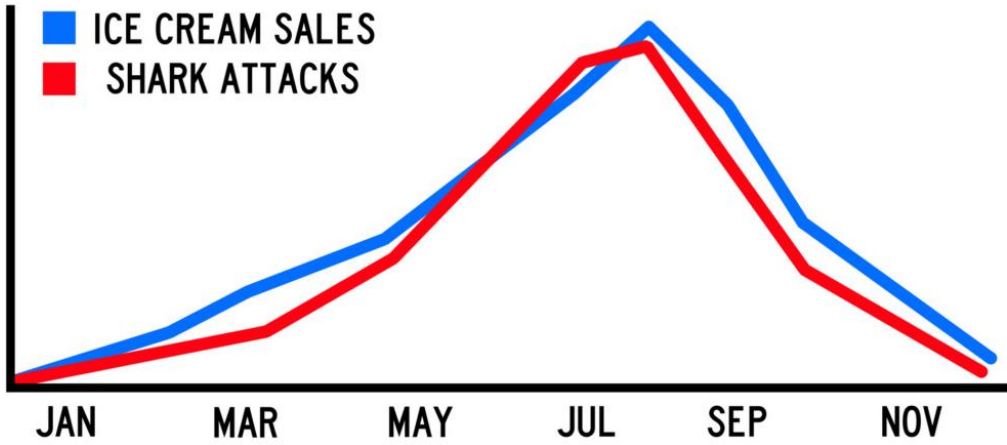
Decision Tree: Modeling





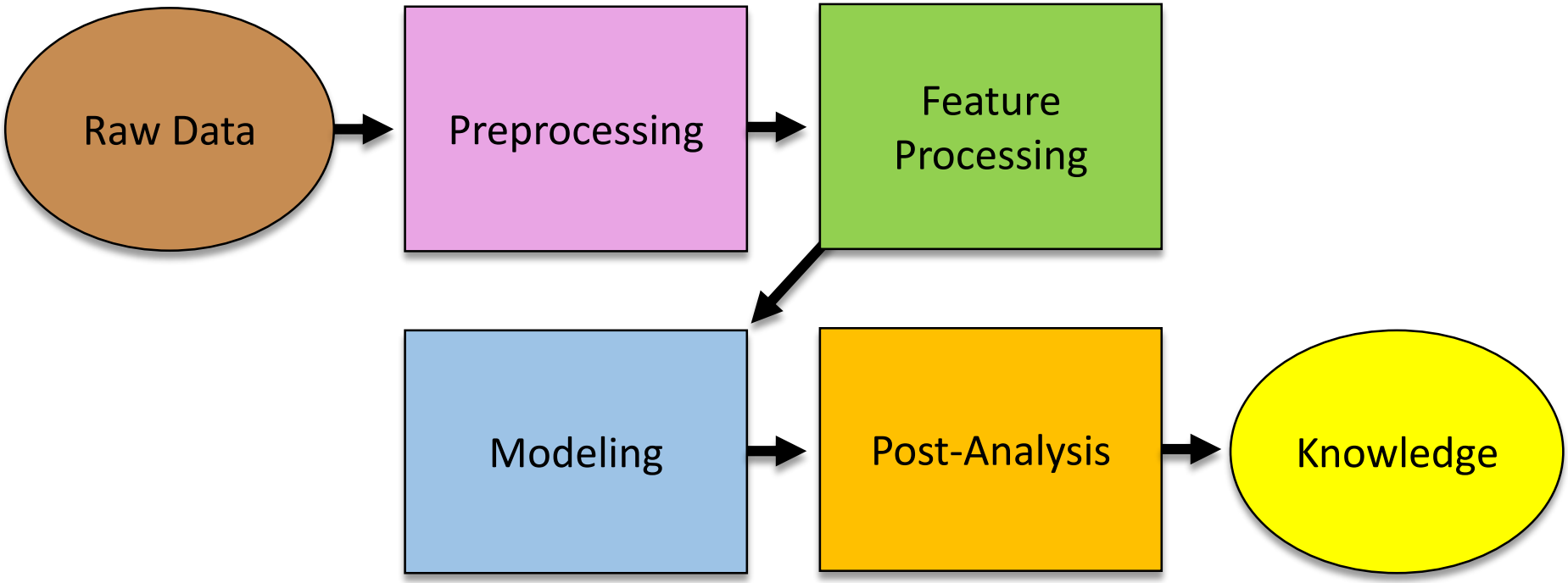
Common Machine Learning Pitfalls

- Working with bad data
- Data leakage
- Not defining the target problem/goals
- Ignoring exploratory analysis
- Handling missing data
- Ignoring assumptions
- Representable does not imply learnable
- Sampling bias
- Overfitting
- Simplicity does not imply better generalizability
- Using the default parameters
- Failing to use an appropriate evaluation metric
- Data dredging
- **Mistaking correlation for causation**
- Failing to consider confounding variables

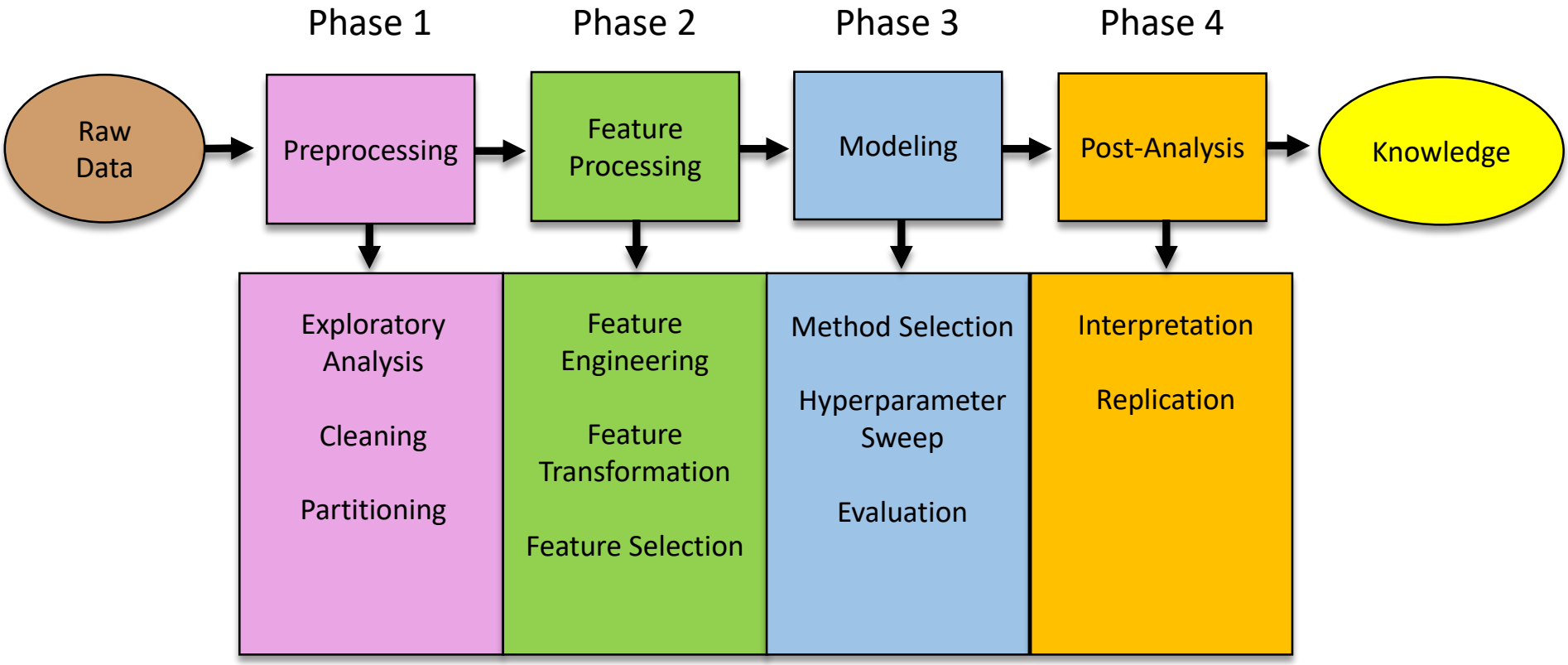


Machine Learning Analysis Pipeline

Analysis Pipeline Overview



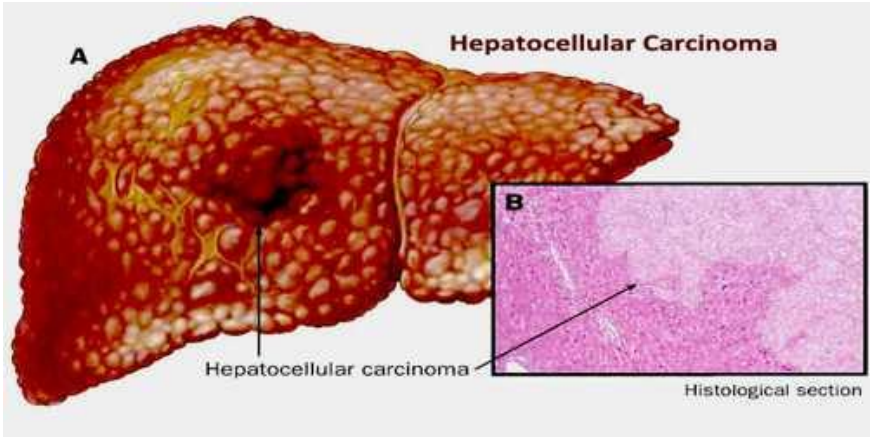
Specific Elements of a Pipeline



- In practice, pipelines are rarely this compartmentalized and linear.
- Phases 1 & 2 are fundamental data science (shared by statistics and ML alike)

Target Dataset

- Hepatocellular Carcinoma (HCC) Survival Data Set from UCI



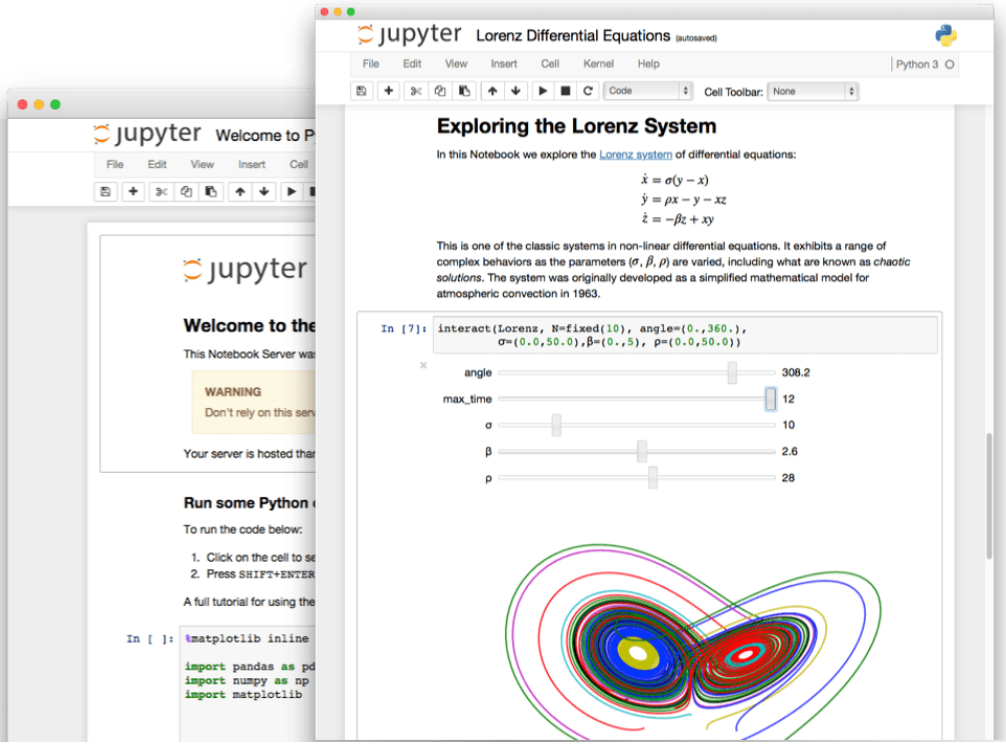
- Why?
  - Small Dataset
    - 165 Instances
    - 49 Features
  - Biomedical Classification Task (Died vs. Survival at 1 Year)
  - Representative of many data considerations/challenges
    - Mixed Feature Types (Binary, Real)
    - Missing Data (10%)
    - Class Imbalance (63 died while 102 survived)

  - Open Source: https://archive.ics.uci.edu/ml/datasets/HCC+Survival

Python and Jupyter Notebook

- Python:
  - Interpreted, object-oriented programming language
  - Simple, easy to learn, readable
  - Widely used in ML

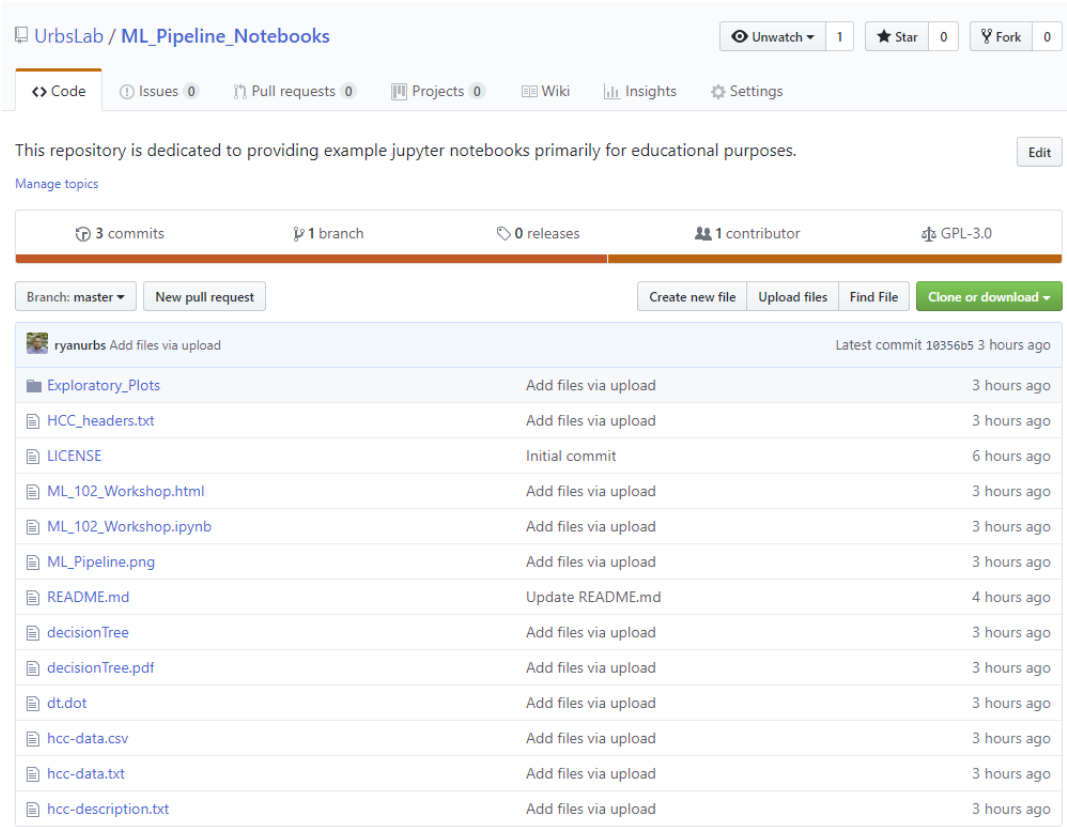
- Jupyter Notebook:
  - Open source web application to create and share documents with live code, equations, visualizations, and narrative text.
  - Very useful for developing reproducible analysis pipelines



An Example Analysis Pipeline

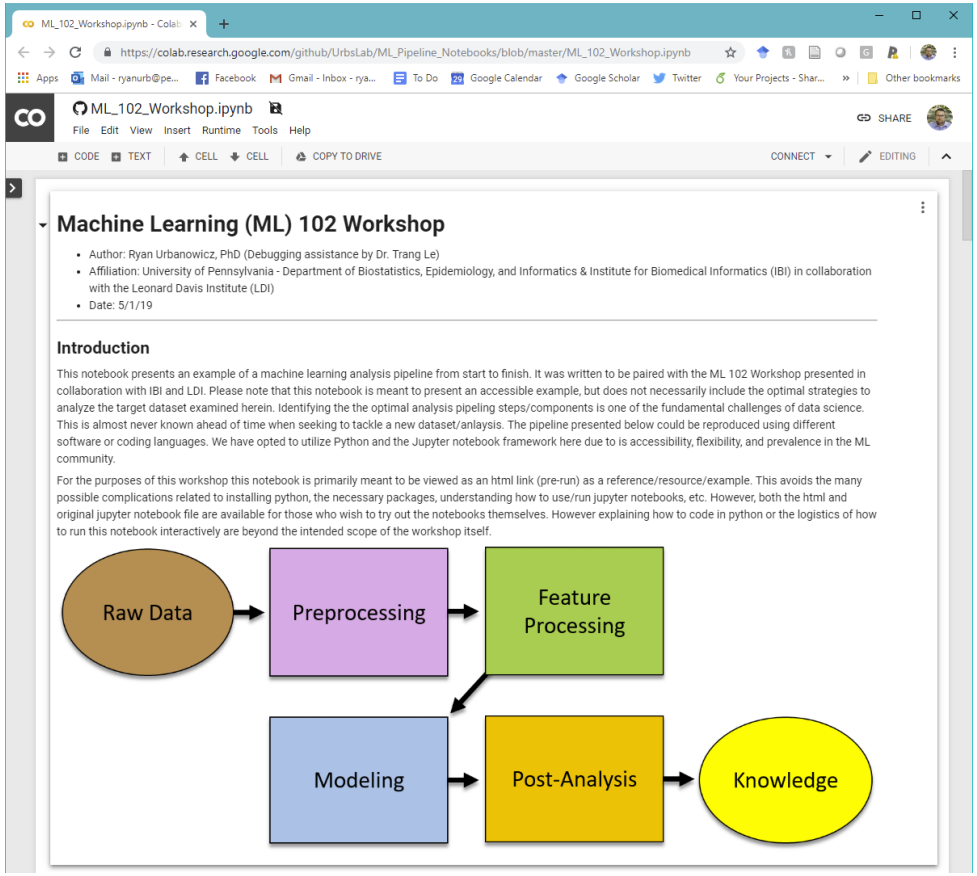
- https://github.com/UrbsLab/ML_Pipeline_Notebooks

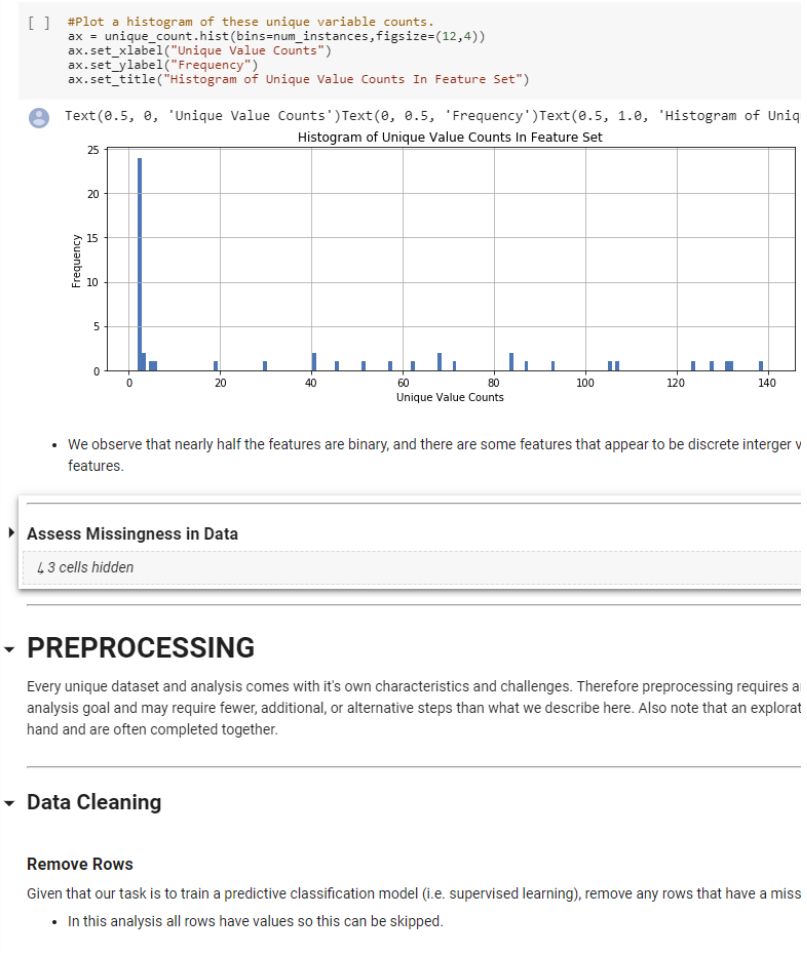
- **Includes**:
  - ML_102_Workshop.htlm
    (pre-run analysis notebook link)
  - ML_102_Workshop.ipynb
    (Jupyter notebook file)
  - Data files from UCI
    - hcc-data.txt
    - hcc-description.txt
  - /Exploratory_Plots/
    (Folder containing PDFs of exploratory analysis figures generated by notebook)



Viewing the Notebook (Google Colab)

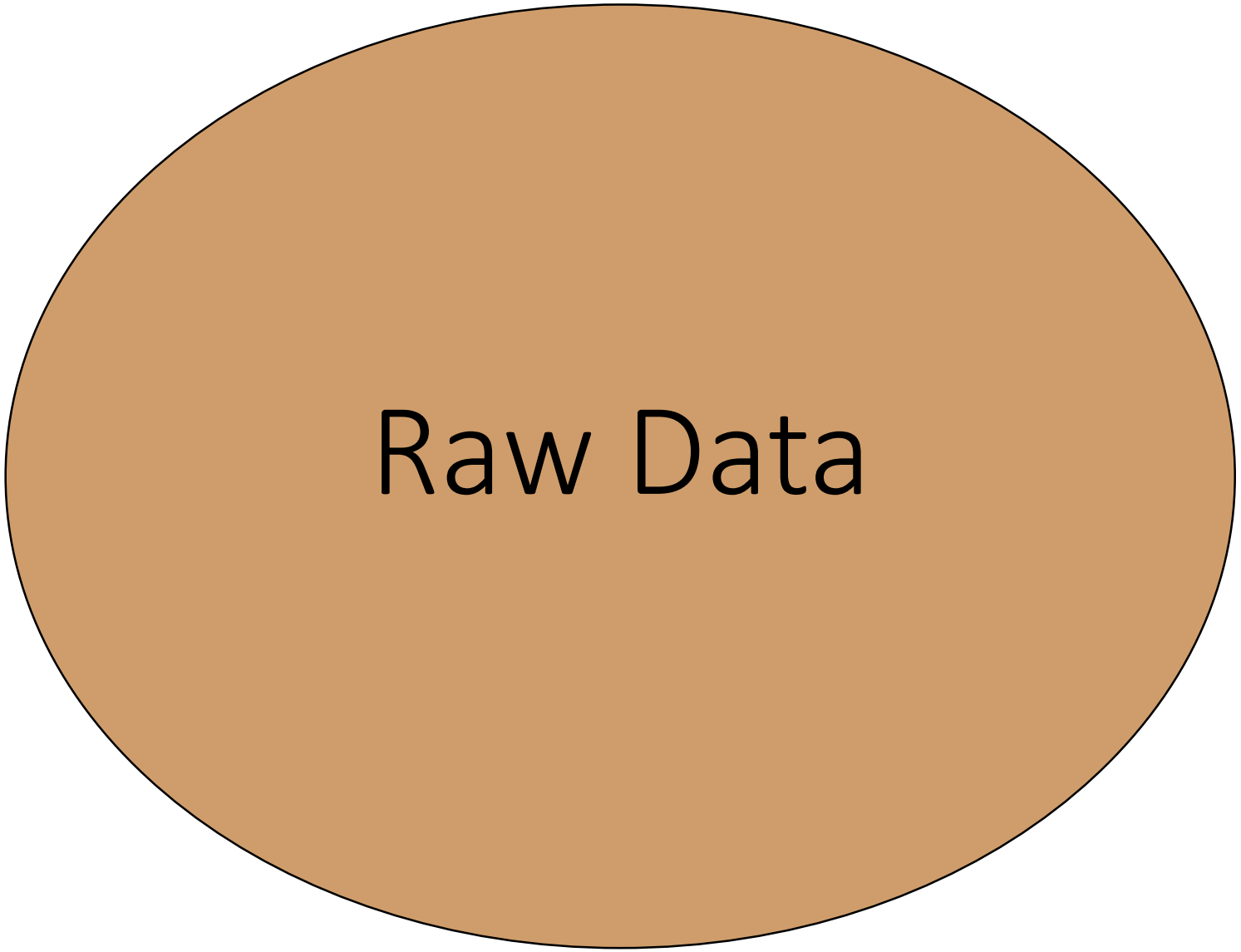
https://colab.research.google.com/github/UrbsLab/ML_Pipeline_Notebooks/blob/master/ML_102_Workshop.ipynb





Disclaimer

- This analysis pipeline is intended as an accessible example.
- There is no single definitive pipeline strategy out there.
- Many aspects of this pipeline can be cyclic or completed simultaneously.
- There are alternative methodologies available for most elements (with their own advantages and disadvantages).
- We will cover the majority of the essentials here, but this is not an exhaustive or ideal pipeline.


Raw Data

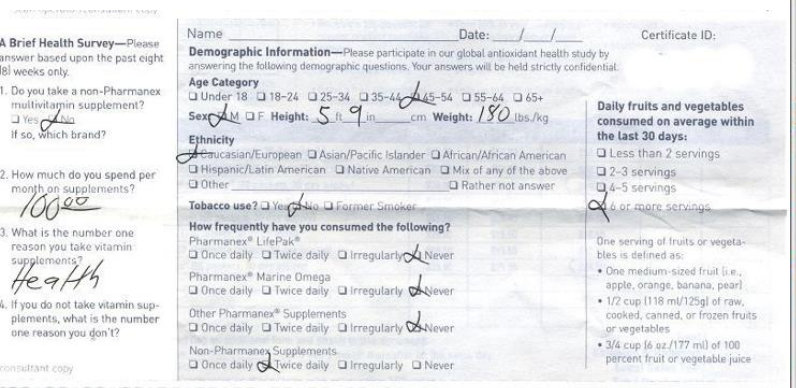
Raw Data

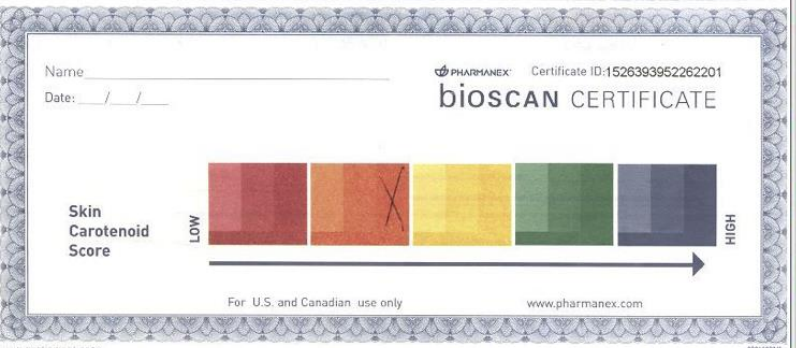
Exam Date: 04/05/2012 10:27 ORD #90003 Accession #9672187
History Number: 1636284
Age: 65Y Sex: M Race: W
Requester: ANDREW COSGAREA

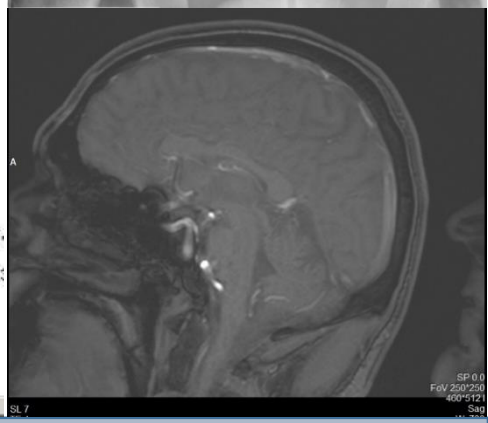
RESULT:
Bilateral knees, history of right knee pain. Four views right and 3 views left. Bilateral 3 compartment arthritis. Small osteophytes femoral notch on flexion view bilaterally. Bilateral patellofemoral arthritis left greater than right.
Moderate to marked Prepatellar soft tissue swelling right knee, suggesting bursitis. No evidence of right joint effusion.

COMPLETE BLOOD COUNT (CBC) WITHOUT DIFFERENTIAL (CBC) - Final result (02/09/2012 8:54 AM EST)

Component	Value
White Blood Cell Count	6040
Red Blood Cell Count	4.72
Hemoglobin	13.9
Hematocrit	40.0
Mean Corpuscular Volume	84.7
Mean Corpus HgB	29.4
Mean Corpus HgB Conc	34.8
RBC Distribution Width	12.7
Platelet Count	254
Mean Platelet Volume	10.1
Nucleated RBC Number	0

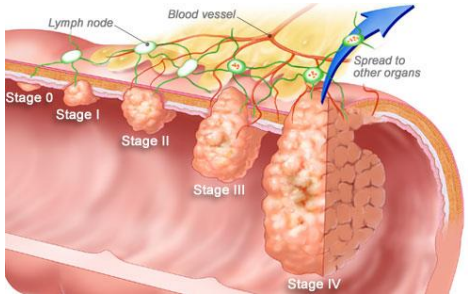






Basic Data (Feature) Characteristics

- Categorical
  - Discrete categories can be represented as integers.
  - No intrinsic ordering to categories.
- Ordinal
  - Discrete values, represented as integers, with a clear ordering of the variables (e.g. low, med, high)
- Continuous
  - Can have any value within a range of numbers
- Unstructured or Sparsely Structured Text
  - Very challenging to work with – see ‘Natural Language Processing’



HCC – Data Orientation [Notebook]

- The data has comma separated values (i.e. csv format).
- There is no header (i.e. column labels) in the data.
- A secondary data dictionary file is available that describes the features and includes the header values
- We have created a csv file of the header names in excel taken directly from this data dictionary.
- Missing values are denoted with '?'
- From the dictionary file we know that the class/outcome column is named 'Class Attribute'.
- Oddly the minority class is coded as 0 (patient died), and the majority class is coded as 1 (patient alive). This is because the 'target' event in this data is 'patient survived 1 year'.

- Target data → ['Pandas' data frame] → `td`

Common Machine Learning Pitfalls

- **Working with bad data**
- Data leakage
- Not understanding the target problem
- Ignoring exploratory analysis
- Handling missing data
- Ignoring assumptions
- Representable does not imply learnable
- Sampling bias
- Overfitting
- Simplicity does not imply better generalizability
- Using the default parameters
- Failing to use an appropriate evaluation metric
- Data dredging
- Mistaking correlation for causation
- Failing to consider confounding variables

Dirty

Noisy

Duplicate

Biased

Sparse

Preprocessing:

Exploratory analysis and data cleaning are, in many ways, intertwined.

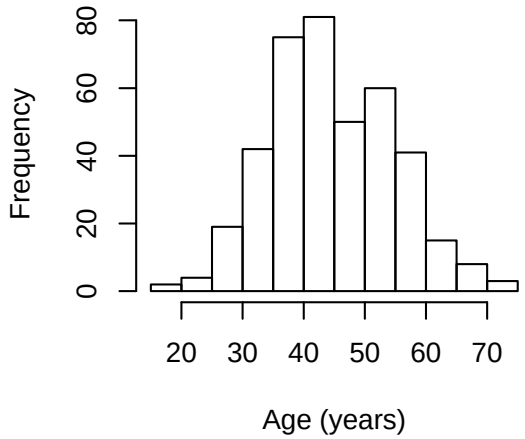
Preprocessing: Exploratory Analysis

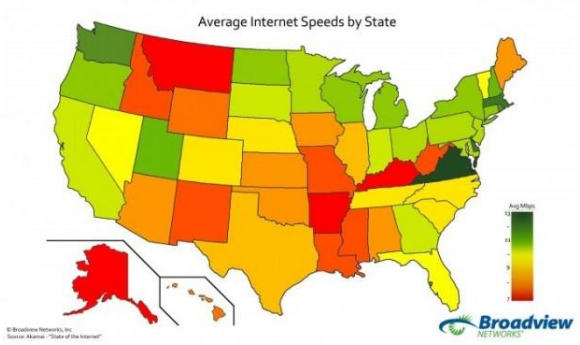
Exploratory Analysis

- AKA: Descriptive Analysis

- Why?
  - Easy to compute (summary statistics)
    - Measures of Center
      - Mean
      - Median
    - Measures of Spread
      - Range
      - Variance (Mean)
      - IQR (Median)
  - Easy to understand (graphically)
    - Frequency/Histograms, etc.
  - Identify outcome type and class imbalance
  - Identify feature types
  - Identify potential errors
  - Assess missingness
  - Identify outliers
  - Describe your sample (Table 1)

Frequency N (%)					
Sex	Race				Total
	White	Black	Asian	Other	
Female	272 (77%)	51 (15%)	25 (7%)	4 (1%)	352 (100%)
Male	262 (78%)	46 (14%)	19 (6%)	8 (2%)	335 (100%)
Total	534 (78%)	97 (14%)	44 (6%)	12 (2%)	687 (100%)



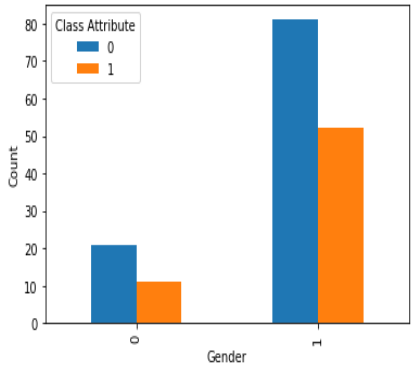


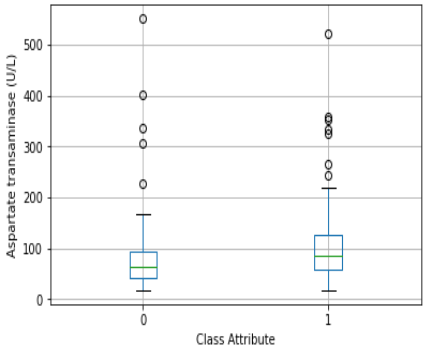
Univariate analyses (Graphs and Statistics)

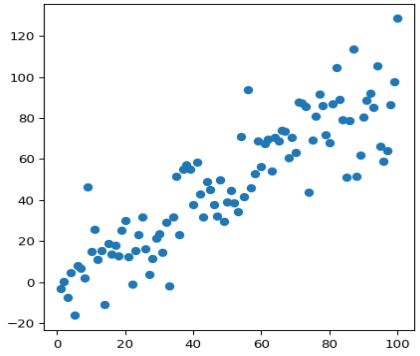
- Choose appropriate approach based on variable types.

- Graph:
  - Outcome = categorical/discrete:
    - Feature = categorical/discrete → contingency table count bar plot
    - Feature = real/continuous → boxplot
  - Outcome = real/continuous:
    - Feature = categorical/discrete → boxplot
    - Feature = real/continuous → scatterplot

- Statistical Test (non-parametric):
  - Outcome = categorical/discrete:
    - Feature = categorical/discrete → Chi Square Test
    - Feature = real/continuous → Mann-Whitney Test
  - Outcome = real/continuous:
    - Feature = categorical/discrete → Mann-Whitney Test
    - Feature = real/continuous → Spearman Correlation







(HCC) Orientation [Notebook]

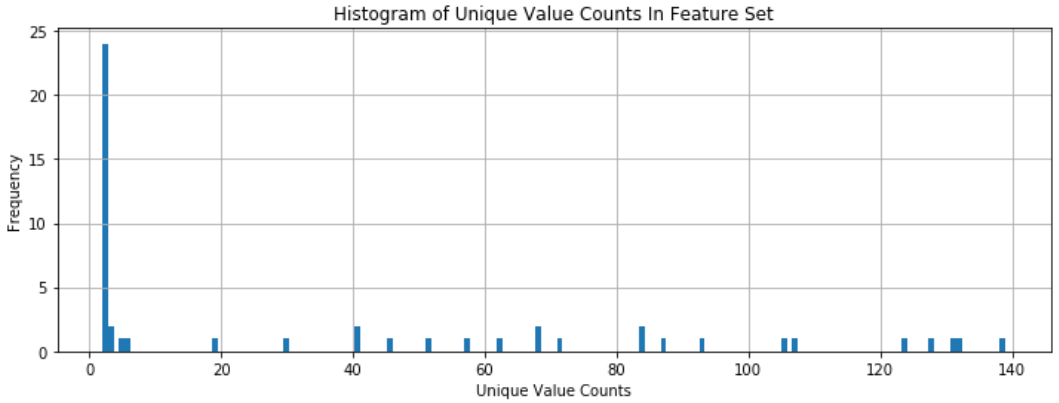
Unique Values

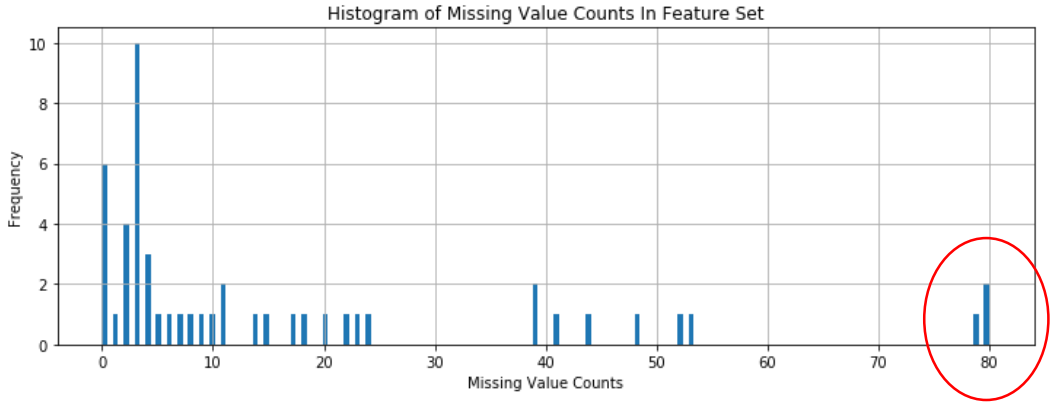
Gender	2
Symptoms	2
Alcohol	2
Hepatitis B Surface Antigen	2
Hepatitis B e Antigen	2
Hepatitis B Core Antibody	2
Hepatitis C Virus Antibody	2
Cirrhosis	2
Endemic Countries	2
Smoking	2
Diabetes	2
Obesity	2
Hemochromatosis	2
Arterial Hypertension	2
Chronic Renal Insufficiency	2
Human Immunodeficiency Virus	2
Nonalcoholic Steatohepatitis	2
Esophageal Varices	2
Splenomegaly	2
Portal Hypertension	2
Portal Vein Thrombosis	2
Liver Metastasis	2
Radiological Hallmark	2
Age at diagnosis	51
Grams of Alcohol per day	19
Packs of cigarets per year	30
Performance Status*	5
Encephalopathy degree*	3
Ascites degree*	3
International Normalised Ratio*	87
Alpha-Fetoprotein (ng/mL)	132
Haemoglobin (g/dL)	71
Mean Corpuscular Volume	128
Leukocytes(G/L)	105
Platelets	131
Albumin (mg/dL)	41
Total Bilirubin(mg/dL)	62
Alanine transaminase (U/L)	93
Aspartate transaminase (U/L)	107
Gamma glutamyl transferase (U/L)	139
Alkaline phosphatase (U/L)	124
Total Proteins (g/dL)	46
Creatinine (mg/dL)	84
Number of Nodules	6
Major dimension of nodule (cm)	68
Direct Bilirubin (mg/dL)	41
Iron	68
Oxygen Saturation (%)	57
Ferritin (ng/mL)	84
Class Attribute	2

Missing Values

Gender	0
Symptoms	18
Alcohol	0
Hepatitis B Surface Antigen	17
Hepatitis B e Antigen	39
Hepatitis B Core Antibody	24
Hepatitis C Virus Antibody	9
Cirrhosis	0
Endemic Countries	39
Smoking	41
Diabetes	3
Obesity	10
Hemochromatosis	23
Arterial Hypertension	3
Chronic Renal Insufficiency	2
Human Immunodeficiency Virus	14
Nonalcoholic Steatohepatitis	22
Esophageal Varices	52
Splenomegaly	15
Portal Hypertension	11
Portal Vein Thrombosis	3
Liver Metastasis	4
Radiological Hallmark	2
Age at diagnosis	0
Grams of Alcohol per day	48
Packs of cigarets per year	53
Performance Status*	0
Encephalopathy degree*	1
Ascites degree*	2
International Normalised Ratio*	4
Alpha-Fetoprotein (ng/mL)	8
Haemoglobin (g/dL)	3
Mean Corpuscular Volume	3
Leukocytes(G/L)	3
Platelets	3
Albumin (mg/dL)	6
Total Bilirubin(mg/dL)	5
Alanine transaminase (U/L)	4
Aspartate transaminase (U/L)	3
Gamma glutamyl transferase (U/L)	3
Alkaline phosphatase (U/L)	3
Total Proteins (g/dL)	11
Creatinine (mg/dL)	7
Number of Nodules	2
Major dimension of nodule (cm)	20
Direct Bilirubin (mg/dL)	44
Iron	79
Oxygen Saturation (%)	80
Ferritin (ng/mL)	80
Class Attribute	0

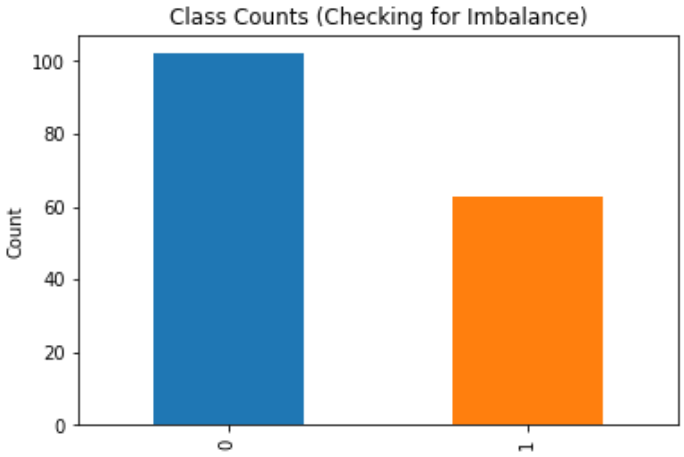
Features: 23 binary, 4 int, 22 continuous
Class: binary





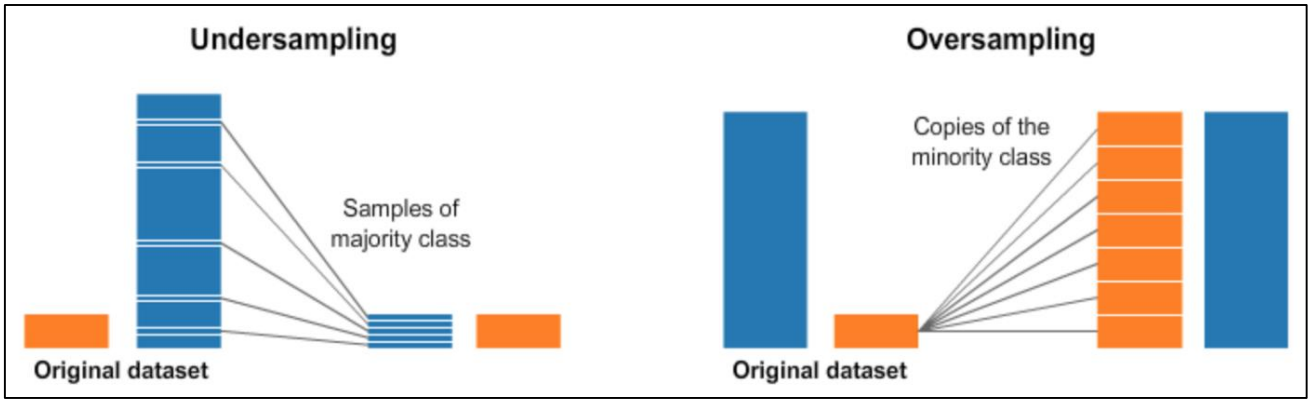
Class Imbalance [Notebook]

- Note that we switched original class encoding (died within 1 year = class 1)
- 'Balanced' classification data → equal number of instances with each class.
- Potential Decisions:
  - Leave imbalance:
    - Use appropriate metrics downstream
  - Undersampling
  - Oversampling



Counts of each class

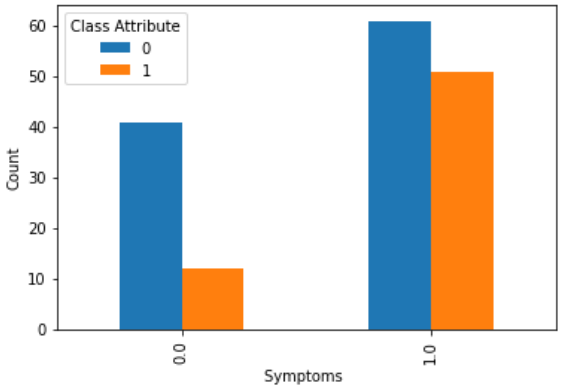
0	102
1	63



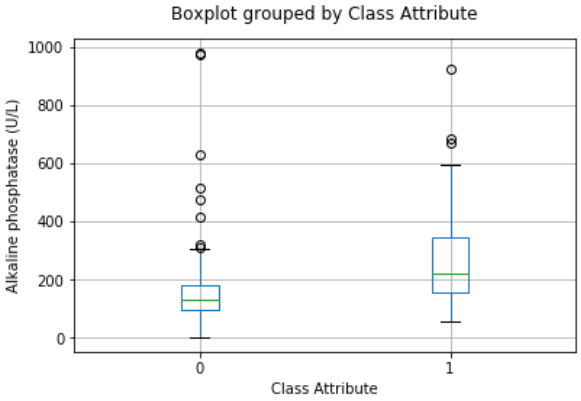
https://towardsdatascience.com/having-an-imbalanced-dataset-here-is-how-you-can-solve-it-1640568947eb

HCC Data Univariate Analysis [Notebook]

- Code to automatically select an appropriate graph and univariate association test based on data type (feature/output).
- PDFs Saved to … [Exploratory_Plots]



##################
Class Attribute 0 1
Symptoms
0.0 41 12
1.0 61 51
Chi Square P-value = 0.00793



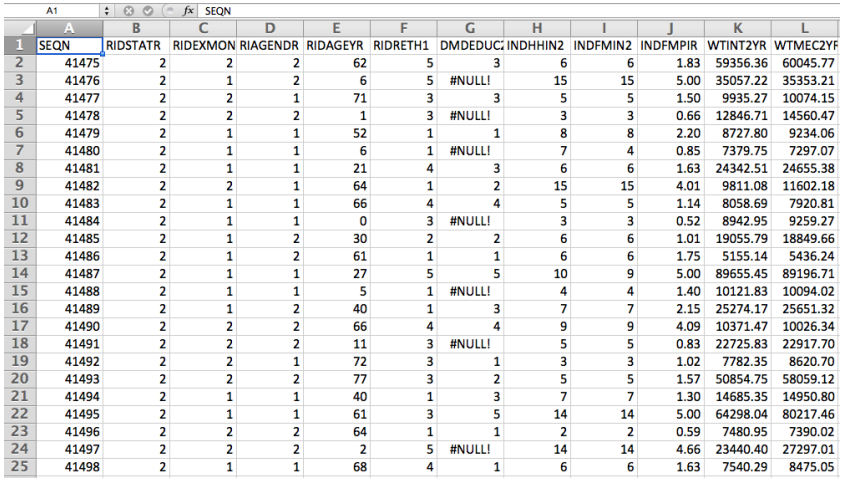
##################
Alkaline phosphatase (U/L)
Mann-Whitney P-value = 3.4492e-07
Mann-Whitney U-Statistic = 1732.5

Significant Univariate Associations:
Alkaline phosphatase (U/L): (p-val = 3.4492742612067905e-07)
Total Bilirubin(mg/dL): (p-val = 0.018476329971748702)
Haemoglobin (g/dL): (p-val = 2.9840465874556433e-05)
Portal Vein Thrombosis: (p-val = 0.008776766144086954)
Aspartate transaminase (U/L): (p-val = 0.0007939456295873922)
Major dimension of nodule (cm): (p-val = 0.019017078947742534)
Age at diagnosis: (p-val = 0.01784161875604351)
Liver Metastasis: (p-val = 0.002624463646548994)
Alpha-Fetoprotein (ng/mL): (p-val = 1.4155839980840113e-06)
Performance Status*: (p-val = 9.393521645117886e-07)
Direct Bilirubin (mg/dL): (p-val = 0.0016708952949634813)
Ferritin (ng/mL): (p-val = 0.003653424065493246)
Ascites degree*: (p-val = 0.0003951810056896238)
Iron: (p-val = 0.0019041108808599487)
Symptoms : (p-val = 0.00793357794444806)
Gamma glutamyl transferase (U/L): (p-val = 0.009313664559528074)
Albumin (mg/dL): (p-val = 0.00011176248489123741)
Creatinine (mg/dL): (p-val = 0.04920378880256977)
International Normalised Ratio*: (p-val = 0.01570018645708362)

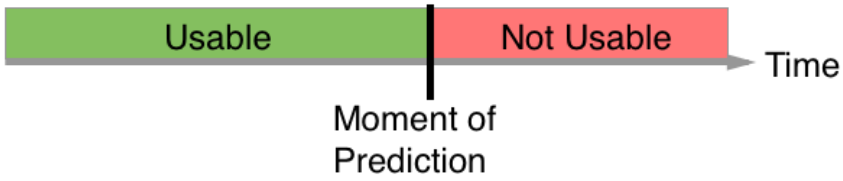
Preprocessing: Data Cleaning

Basic Data Cleaning

- Remove Rows
  - Supervised Learning → Remove rows with missing outcome value.
- Remove Columns
  - Clearly irrelevant features (e.g. instance ID)
  - Prevent Data Leakage:
    - Remove precursor features (used to build outcome variable in study)
    - Remove features that would be unavailable when prediction made.
- None removed in [notebook]



	A	B	C	D	E	F	G	H	I	J	K	L
1	SEQN	RIDSTATR	RIDEXMON	RIAGENDR	RIDAGEYR	RIDRETH1	DMDEDUC2	INDHHIN2	INDFMIN2	INDFMPIR	WTINT2YR	WTMEC2YR
2	41475	2	2	2	62	5	3	6	6	1.83	59356.36	60045.77
3	41476	2	1	2	6	5	#NULL!	15	15	5.00	35057.22	35353.21
4	41477	2	2	1	71	3	3	5	5	1.50	9935.27	10074.15
5	41478	2	2	2	1	3	#NULL!	3	3	0.66	12846.71	14560.47
6	41479	2	1	1	52	1	1	8	8	2.20	8727.80	9234.06
7	41480	2	1	1	6	1	#NULL!	7	4	0.85	7379.75	7297.07
8	41481	2	1	1	21	4	3	6	6	1.63	24342.51	24655.38
9	41482	2	2	1	64	1	2	15	15	4.01	9811.08	11602.18
10	41483	2	1	1	66	4	4	5	5	1.14	8058.69	7920.81
11	41484	2	1	1	0	3	#NULL!	3	3	0.52	8942.95	9259.27
12	41485	2	1	2	30	2	2	6	6	1.01	19055.79	18849.66
13	41486	2	1	2	61	1	1	6	6	1.75	5155.14	5436.24
14	41487	2	1	1	27	5	5	10	9	5.00	89655.45	89196.71
15	41488	2	1	1	5	1	#NULL!	4	4	1.40	10121.83	10094.02
16	41489	2	1	2	40	1	3	7	7	2.15	25274.17	25651.32
17	41490	2	2	2	66	4	4	9	9	4.09	10371.47	10026.34
18	41491	2	2	2	11	3	#NULL!	5	5	0.83	22725.83	22917.70
19	41492	2	2	1	72	3	1	3	3	1.02	7782.35	8620.70
20	41493	2	2	2	77	3	2	5	5	1.57	50854.75	58059.12
21	41494	2	1	1	40	1	3	7	7	1.30	14685.35	14950.80
22	41495	2	1	1	61	3	5	14	14	5.00	64298.04	80217.46
23	41496	2	2	2	64	1	1	2	2	0.59	7480.95	7390.02
24	41497	2	2	2	2	5	#NULL!	14	14	4.66	23440.40	27297.01
25	41498	2	1	1	68	4	1	6	6	1.63	7540.29	8475.05



HCC Data Handle Missingness [Notebook]

- Scikit-learn does not allow for missing values
- Continuous features → Median imputation
- Discrete features → Mode imputation
- Re-check uniqueness and missingness

Unique Values

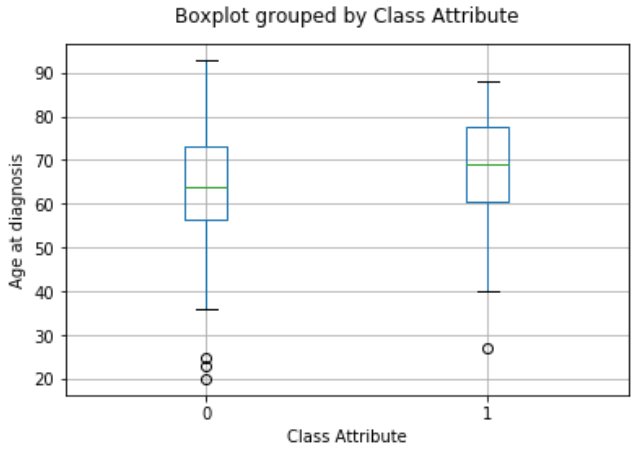
Gender	2
Symptoms	2
Alcohol	2
Hepatitis B Surface Antigen	2
Hepatitis B e Antigen	2
Hepatitis B Core Antibody	2
Hepatitis C Virus Antibody	2
Cirrhosis	2
Endemic Countries	2
Smoking	2
Diabetes	2
Obesity	2
Hemochromatosis	2
Arterial Hypertension	2
Chronic Renal Insufficiency	2
Human Immunodeficiency Virus	2
Nonalcoholic Steatohepatitis	2
Esophageal Varices	2
Splenomegaly	2
Portal Hypertension	2
Portal Vein Thrombosis	2
Liver Metastasis	2
Radiological Hallmark	2
Age at diagnosis	51
Grams of Alcohol per day	19
Packs of cigarets per year	30
Performance Status*	5
Encephalopathy degree*	3
Ascites degree*	3
International Normalised Ratio*	87
Alpha-Fetoprotein (ng/mL)	132
Haemoglobin (g/dL)	72
Mean Corpuscular Volume	129
Leukocytes(G/L)	105
Platelets	131
Albumin (mg/dL)	41
Total Bilirubin(mg/dL)	62
Alanine transaminase (U/L)	93
Aspartate transaminase (U/L)	107
Gamma glutamyl transferase (U/L)	140
Alkaline phosphatase (U/L)	125
Total Proteins (g/dL)	47
Creatinine (mg/dL)	84
Number of Nodules	6
Major dimension of nodule (cm)	68
Direct Bilirubin (mg/dL)	41
Iron	69
Oxygen Saturation (%)	57
Ferritin (ng/mL)	84
Class Attribute	2

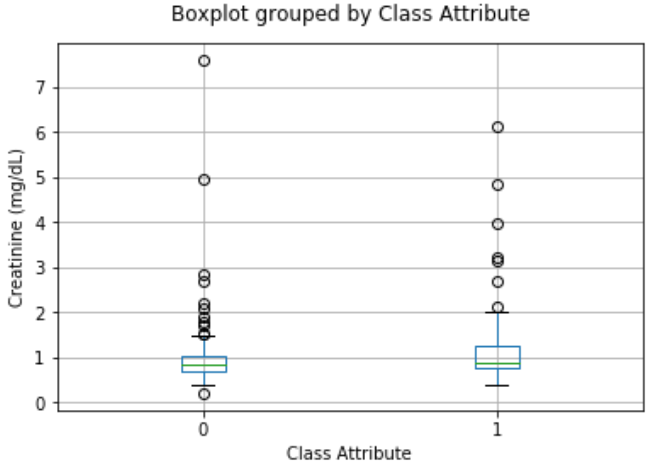
Missing Values

Gender	0
Symptoms	0
Alcohol	0
Hepatitis B Surface Antigen	0
Hepatitis B e Antigen	0
Hepatitis B Core Antibody	0
Hepatitis C Virus Antibody	0
Cirrhosis	0
Endemic Countries	0
Smoking	0
Diabetes	0
Obesity	0
Hemochromatosis	0
Arterial Hypertension	0
Chronic Renal Insufficiency	0
Human Immunodeficiency Virus	0
Nonalcoholic Steatohepatitis	0
Esophageal Varices	0
Splenomegaly	0
Portal Hypertension	0
Portal Vein Thrombosis	0
Liver Metastasis	0
Radiological Hallmark	0
Age at diagnosis	0
Grams of Alcohol per day	0
Packs of cigarets per year	0
Performance Status*	0
Encephalopathy degree*	0
Ascites degree*	0
International Normalised Ratio*	0
Alpha-Fetoprotein (ng/mL)	0
Haemoglobin (g/dL)	0
Mean Corpuscular Volume	0
Leukocytes(G/L)	0
Platelets	0
Albumin (mg/dL)	0
Total Bilirubin(mg/dL)	0
Alanine transaminase (U/L)	0
Aspartate transaminase (U/L)	0
Gamma glutamyl transferase (U/L)	0
Alkaline phosphatase (U/L)	0
Total Proteins (g/dL)	0
Creatinine (mg/dL)	0
Number of Nodules	0
Major dimension of nodule (cm)	0
Direct Bilirubin (mg/dL)	0
Iron	0
Oxygen Saturation (%)	0
Ferritin (ng/mL)	0
Class Attribute	0

Outliers

- Use logic and understanding of variables to check for impossible outliers.
- Target data: Age?
- Look for and consider strategies to deal with outliers.
  - Distinguish valid outliers from invalid outliers.
  - Consider your problem/goal to decide whether to:
    - Leave outliers
    - Remove outliers
    - Apply transformation function

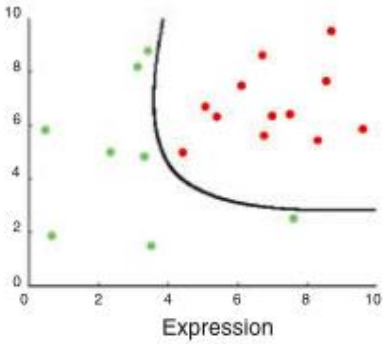


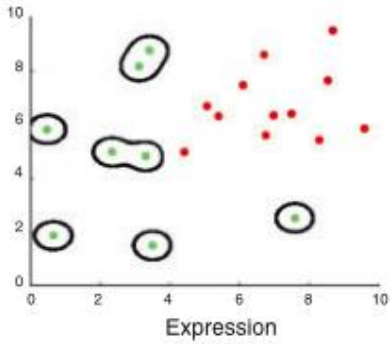


Preprocessing: Data Partitioning

Common Machine Learning Pitfalls

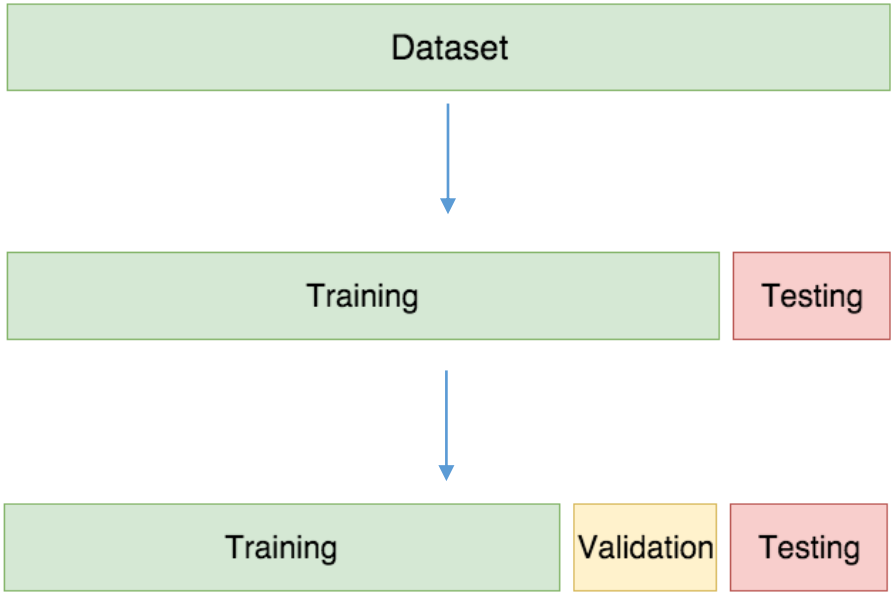
- Working with bad data
- Data leakage
- Not defining the target problem/goals
- Ignoring exploratory analysis
- Handling missing data
- Ignoring assumptions
- Representable does not imply learnable
- Sampling bias
- **Overfitting**
- Simplicity does not imply better generalizability
- Using the default parameters
- Failing to use an appropriate evaluation metric
- Data dredging
- Mistaking correlation for causation
- Failing to consider confounding variables





Data Partitions: Training, Validation, Testing

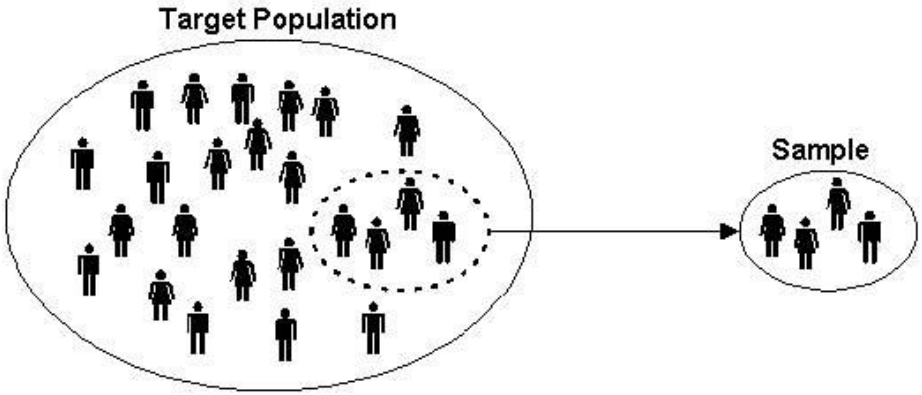
- Training set:
  - Train the model
- Validation set:
  - Evaluate model generalization during hyperparameter optimization
  - Indirectly impacts modeling
- Test set:
  - Hold out data for unbiased evaluation of final model fit/generalization
- Take steps to ensure representative partitions:
  - Randomly selected?
  - Roughly equal sample sizes?
  - Roughly equal sample variance?
  - Roughly the same class balance in each partition?



https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6

Common Machine Learning Pitfalls

- Working with bad data
- Data leakage
- Not defining the target problem/goals
- Ignoring exploratory analysis
- Handling missing data
- Ignoring assumptions
- Representable does not imply learnable
- **Sampling bias**
- Overfitting
- Simplicity does not imply better generalizability
- Using the default parameters
- Failing to use an appropriate evaluation metric
- Data dredging
- Mistaking correlation for causation
- Failing to consider confounding variables



- Modeling can be biased by training on a non-representative dataset – results in poorly generalizable model.
  - Can be a problem founded in data collection
  - Can be a problem in data partitioning for ML modeling

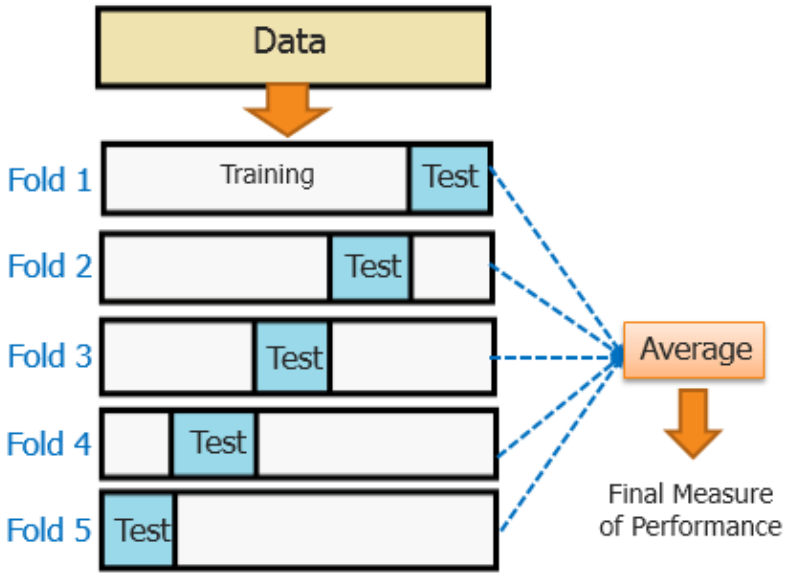
K-fold Cross-validation (CV)

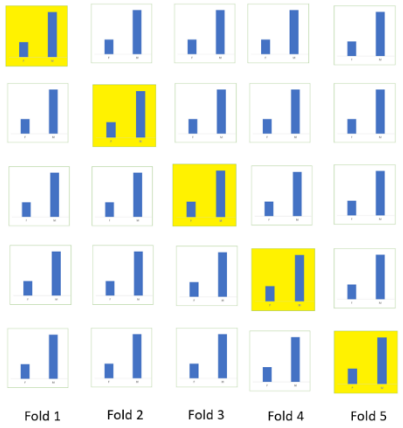
- Widely used for hyperparameter optimization and evaluating model predictive ability.

- General idea:
  - split data into groups
  - use a portion for creating model
  - use the rest for testing model
  - combine results obtained for each test

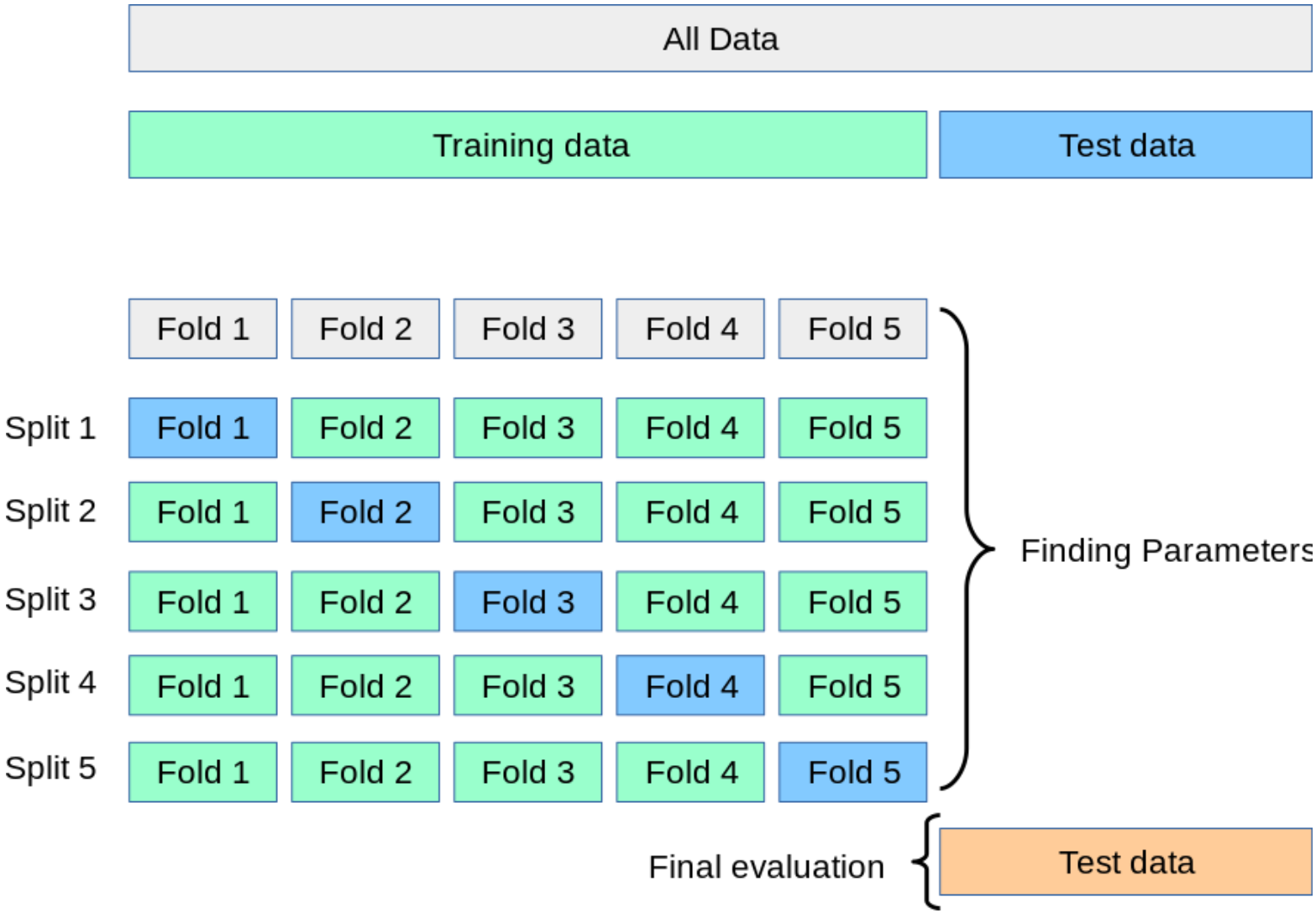
- K-fold: Split data into K randomly selected sub-samples and use one portion for testing and K-1 for model training

- Variations of this concept:
  - Leave-one-out: (Logical extreme) (related to jackknife in statistics but on left out rather than kept values) Keep single observations out while using remaining data for training
  - Stratified k-fold CV – key covariates, imbalanced data, or matched datasets
  - Nested CV – time series data



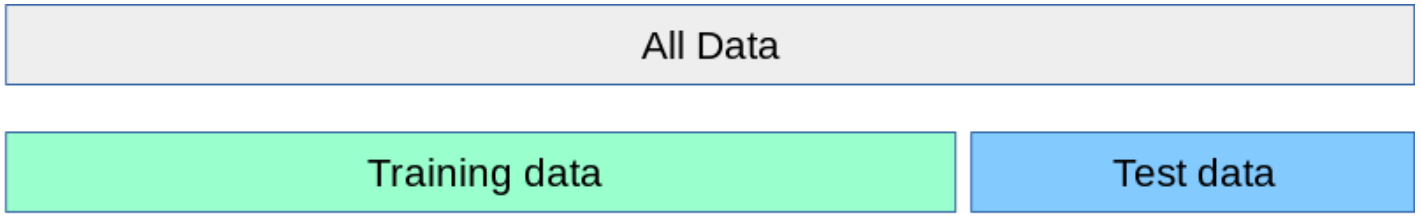


CV Design for Prediction Modeling

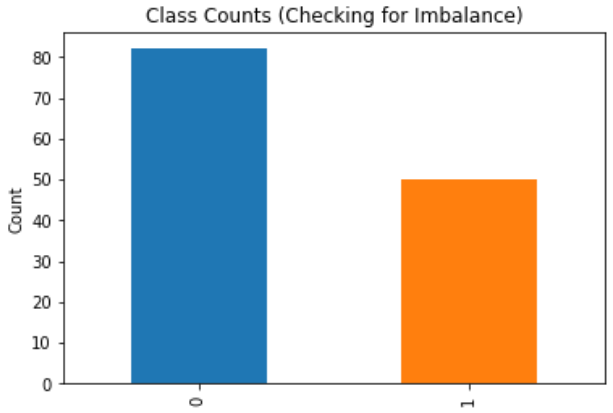


https://scikit-learn.org/stable/modules/cross_validation.html

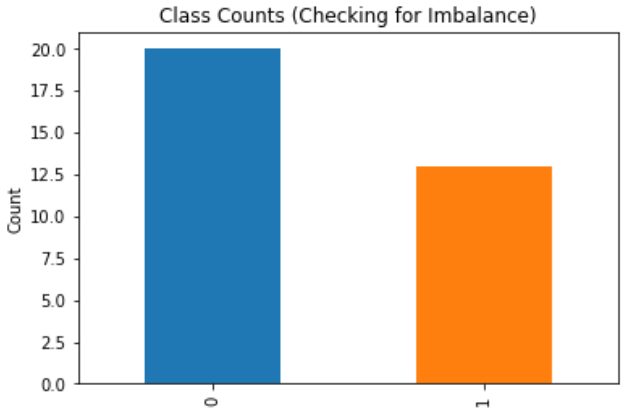
HCC Data Partitioning [Notebook]



- Stratified CV (by outcome)
- 80% training 20% testing split
- Training Data
  - 132 instances
- Testing Data
  - 33 instances



Counts of each class in training data
0 82
1 50



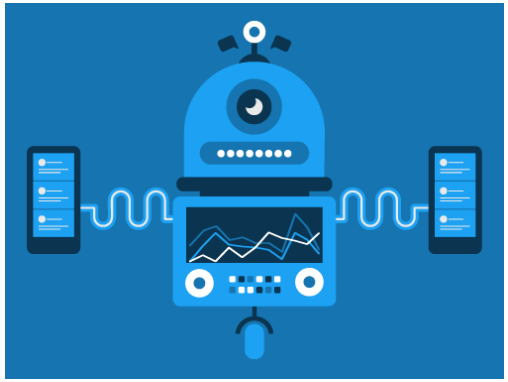
Counts of each class in testing data
0 20
1 13

Feature Processing

Feature Processing

- Improving the representation of data so that downstream machine learning can achieve higher performance

- Feature Engineering (manual)
  - Custom build new features from existing ones using domain knowledge
  - E.g. Dates (start and stop) → treatment duration



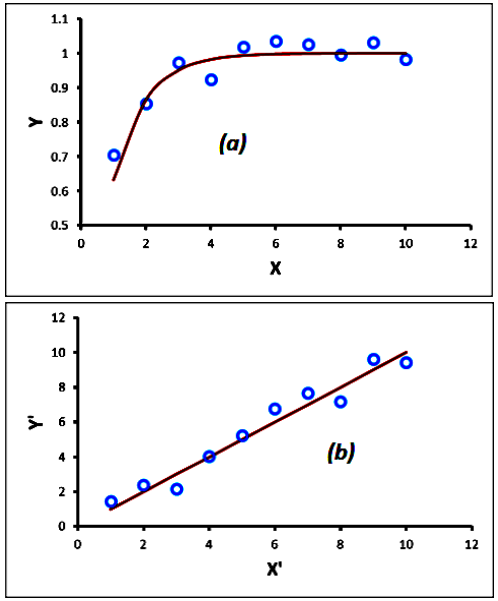
- Feature Transformation (‘automated’ by function or algorithm)
  - Scaling
  - Binning
  - Dimensionality Reduction
  - Feature Construction

- Feature Selection (manual and/or ‘automated’ by algorithm)

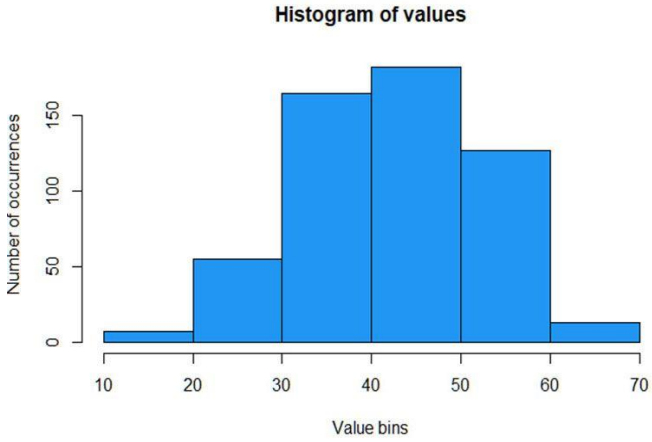
Feature Processing: Feature Transformation

Feature Transformation [1 of 2]

- Scaling Features:
  - Mathematically normalizing features to fall within a given range (e.g. 0-1)
    - Can be important to interpretation of statistical modeling (e.g. linear regression)
    - Typically less/not important in ML
  - Transformed by a given function (e.g. log transformation).
    - e.g. attempt to linearize association with outcome

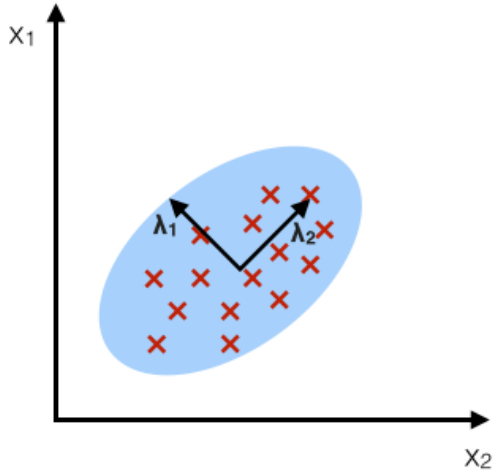


- Binning:
  - Converting continuous/real-valued features to discrete or categorical features
  - A useful simplification so that certain analyses/methods can be applied.
  - Information loss is possible/likely.

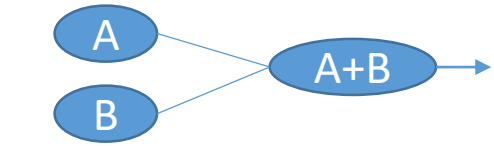


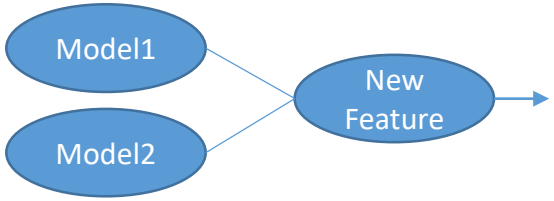
Feature Transformation [2 of 2]

- Dimensionality Reduction:
  - Also known as feature extraction or feature projection
  - Reduce the number of random variables under consideration by obtaining a set of principal variables, typically via some projection strategy (e.g. PCA)



- Feature Construction:
  - Essentially an 'automated' version of feature engineering
  - Computational method combines 2 or more:
    - Existing features
    - ML model output
  - …to 'construct' a new feature.



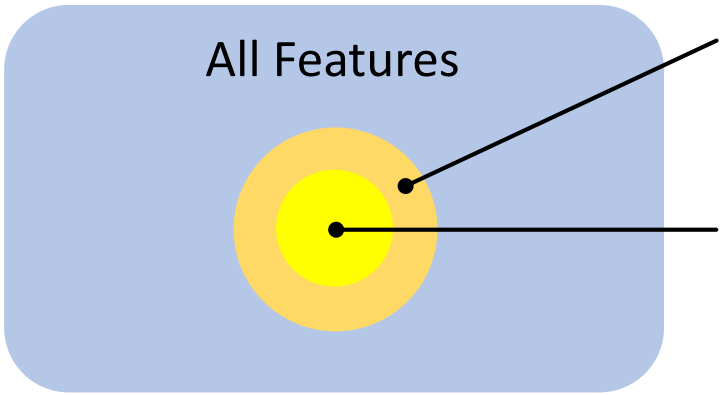


Feature Processing: Feature Selection

Relevant and Non-Redundant Features

- Feature Selection:
  - Process of identifying and selecting the subset of relevant (i.e. predictive), and (sometimes) non-redundant features, for downstream analysis.

- Why is Feature selection important?
  - Computational burden
  - Noise of non-relevant features slows or can prevent successful modeling.



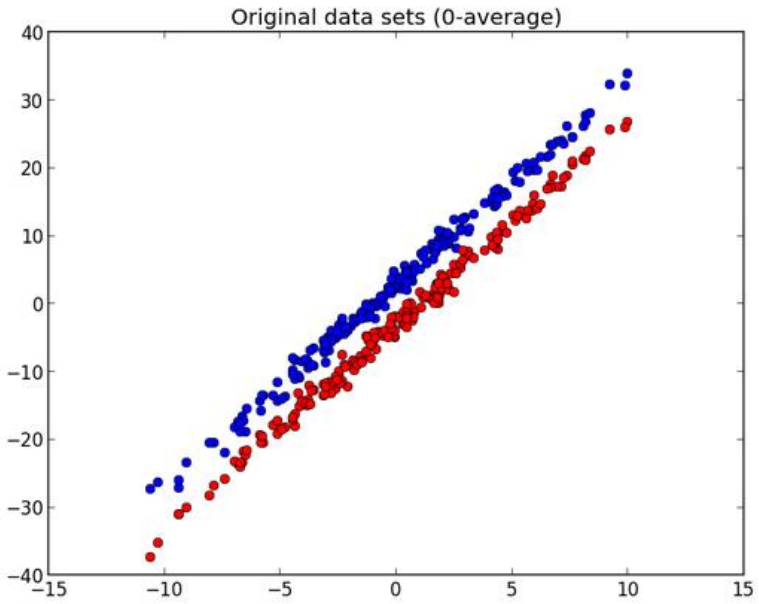
Relevant Features

Relevant and Non-redundant Features

Correlated Features (Redundant?)

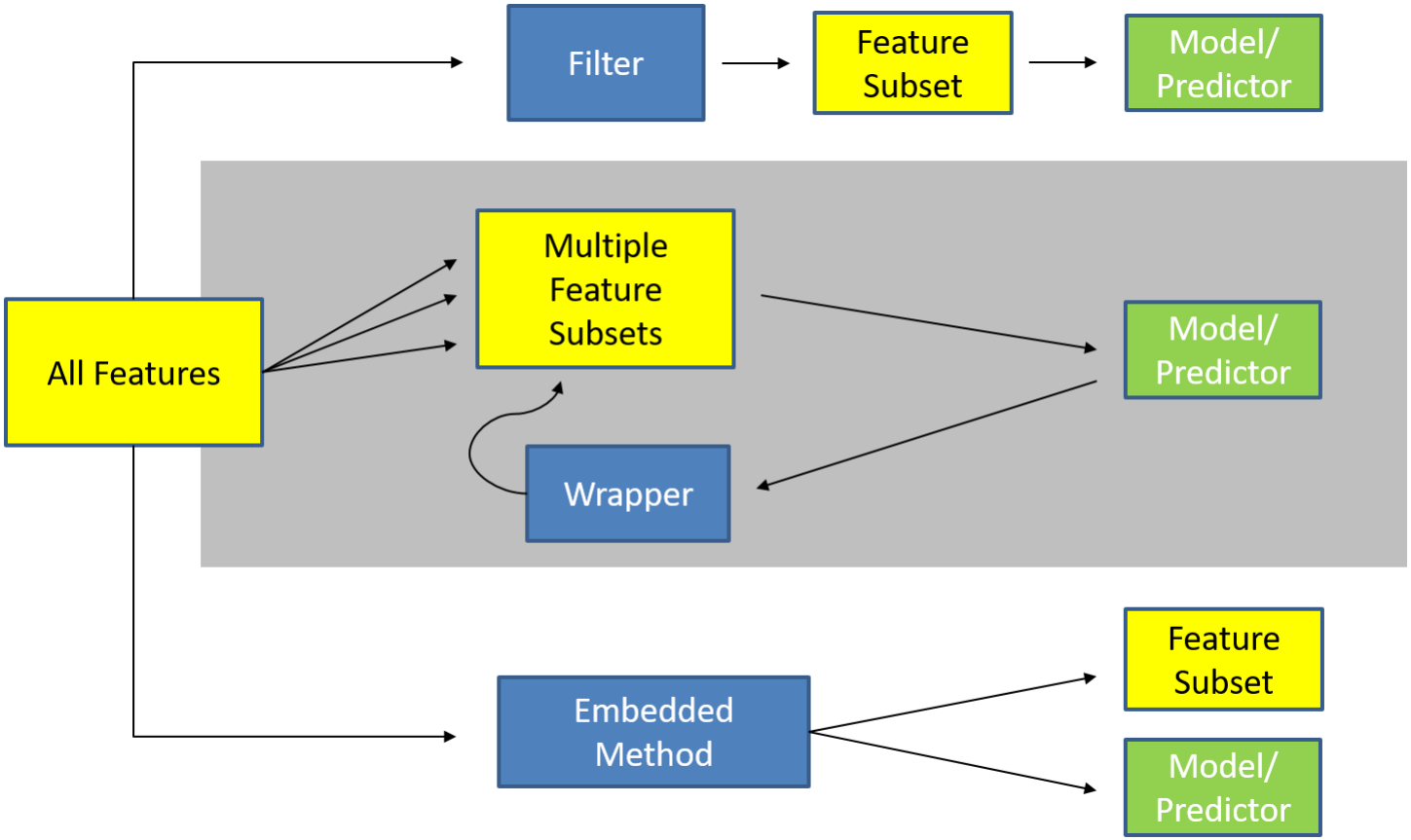
- Problem: Should one feature of a correlated pair be removed or not?

- Solution:
  - First check for duplicate/copied data variables. Can be removed safely.
  - Highly correlated features can be removed, however critical information could still be lost.
  - Perfectly correlated features can be removed.



https://www.quora.com/In-feature-selection-should-we-always-remove-all-the-features-that-have-high-covariance-between-them

Feature Selection Method Families



- Filter Methods: Proxy measure for ranking. Fast! Independent of modeling method. (e.g. univariate tests of association)
- Wrapper Methods: Feedback from modeling. Computationally expensive! Not scalable. Provides best performing feature set for a particular type of model. (e.g. recursive feature elimination, genetic algorithms)
- Embedded Methods: Integrated with modeling. Less expensive than wrappers. Less prone to overfitting. Specific to learning machine selected. (e.g. L1 LASSO regularization, decision tree)

Filter-based Feature Selection

- Example methods:
  - Mutual Information
  - Gini index
  - Pearson correlation
  - Chi-square
  - Information gain

- Above methods are not effective in dealing with complex multivariate, feature interactions.

- Feature interactions – association between multiple features and outcome is more than "the sum of their parts"

- Relief-based feature selection:
  - Sensitive to feature interactions without explicitly testing feature combinations.

Relief-Algorithm Intuition [1 of 2]

- A predictive attribute is more likely to have a different state between instances from different classes and should have the same state for instances from the same class.

Similar Instances (Neighbors)

+1

A A B C C A C B B A B B A C C – Healthy

A A B C C A C B B B B B A C C – Sick

This feature is more likely to be informative.

Relief-Algorithm Intuition [2 of 2]

- A predictive attribute is more likely to have a different state between instances from different classes and should have the same state for instances from the same class.

Similar Instances (Neighbors)

-1

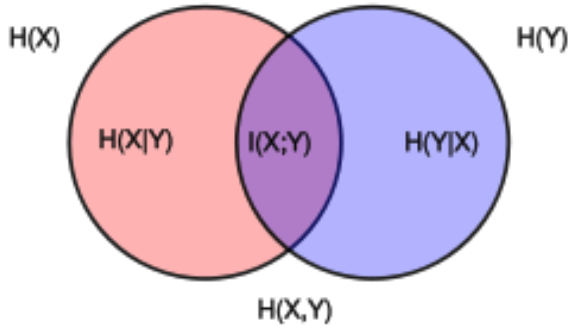
A A B C C A C B B C B B A C C – Healthy

A A B C A A C B B C B B A C C – Healthy

This feature is less likely to be informative.

HCC – Feature Selection [Notebook]

- Applied:
  - Mutual Information:
    - Measure of mutual dependence
    - i.e. the amount of information obtained about one variable by observing the other.



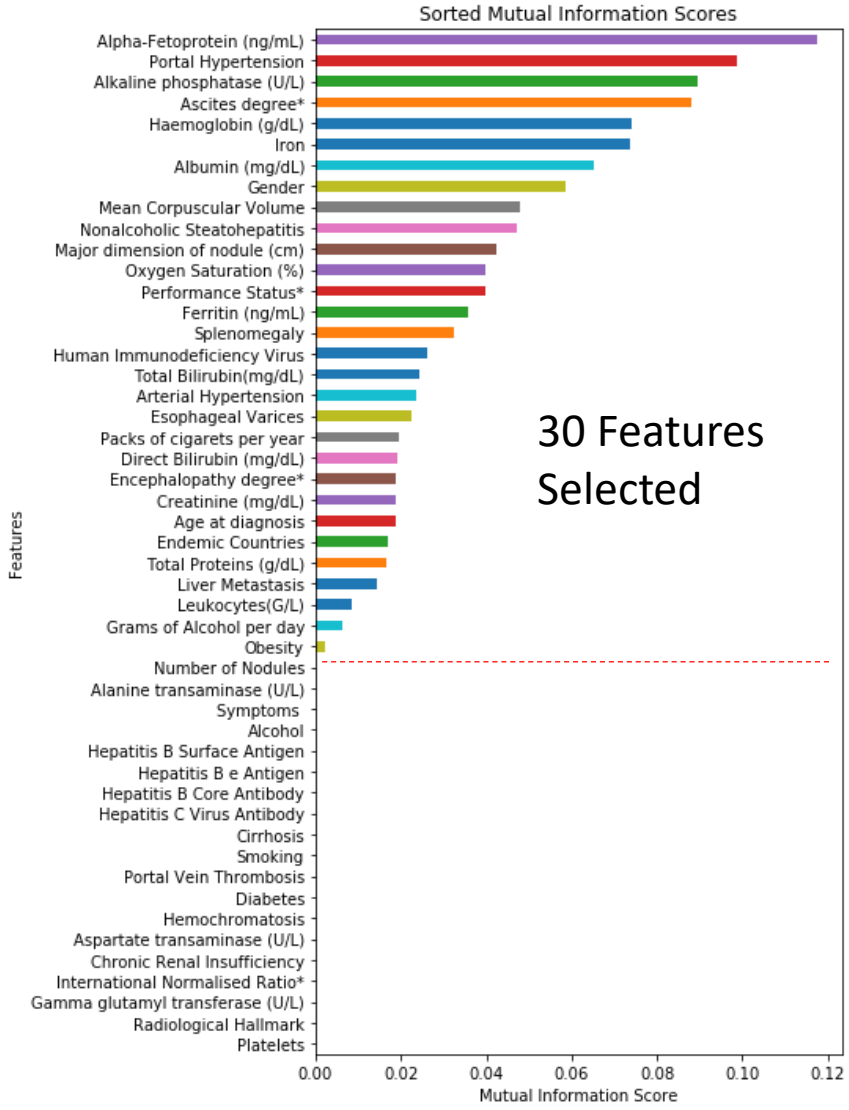
  - ReliefF
    - Relief-based feature selection algorithm
    - k = 100

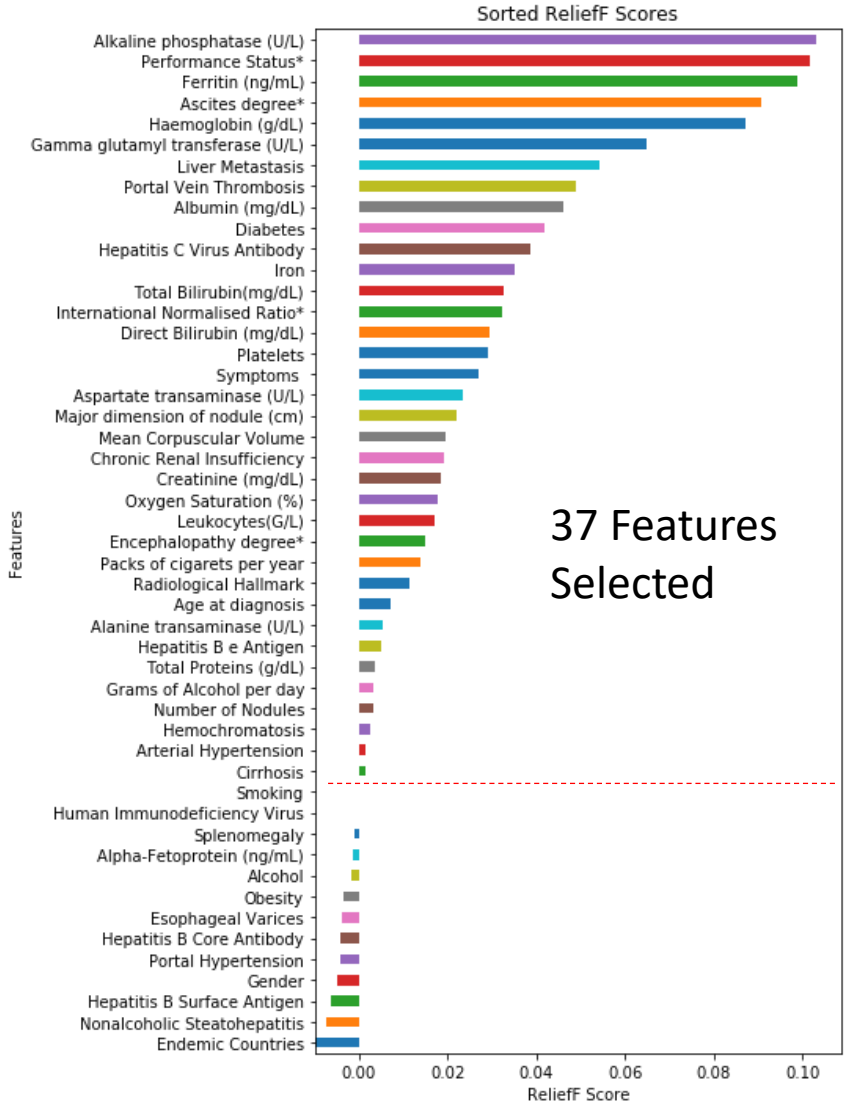
github.com/EpistasisLab/scikit-rebate

github.com/EpistasisLab/ReBATE

  - Selection Strategy:
    - Relatively small feature space (i.e. 49 features) …
    - So, opted for a very liberal approach…
    - Selected any features yielding a positive value by either method.
    - i.e. union of the two sets

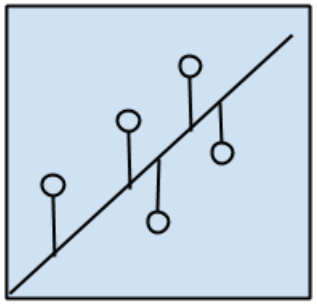
HCC – Feature Selection [Notebook]



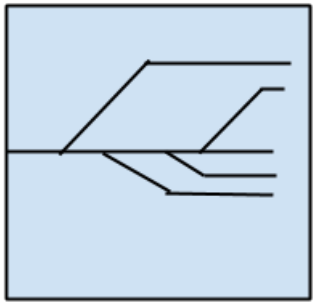


Modeling:
Method Selection

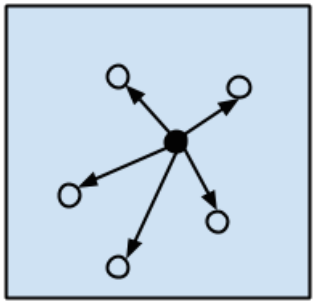
Machine Learning Algorithm Families



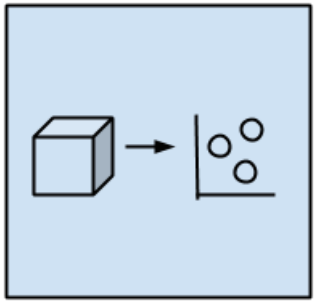
Regression Algorithms



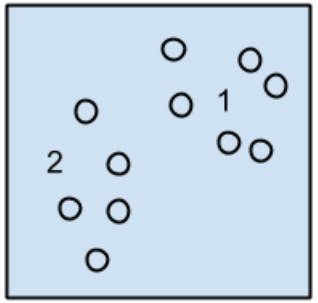
Regularization Algorithms



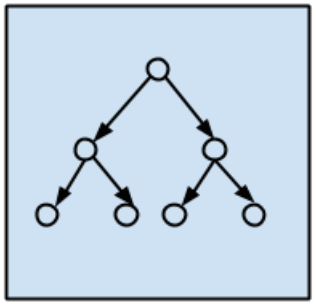
Instance-based Algorithms



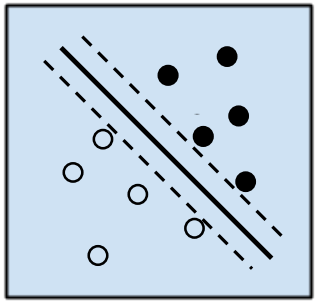
Dimensional Reduction Algorithms



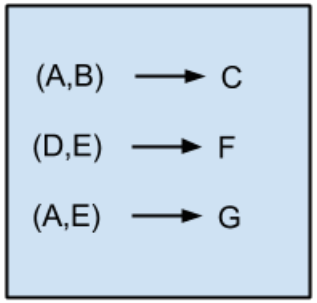
Clustering Algorithms



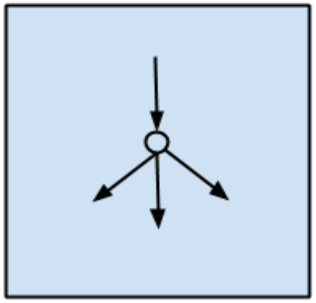
Decision Tree Algorithms



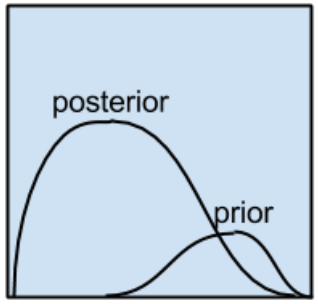
Support Vector Machines



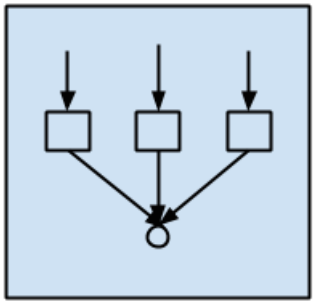
Association Rule Learning Algorithms



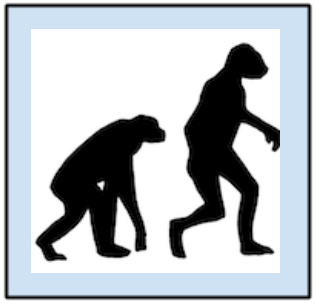
Artificial Neural Network Algorithms



Bayesian Algorithms

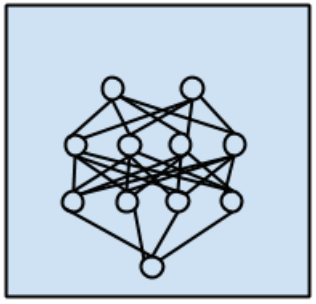


Ensemble Algorithms

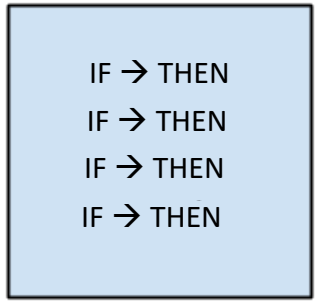


Evolutionary Algorithms

Non-exhaustive list of ML families

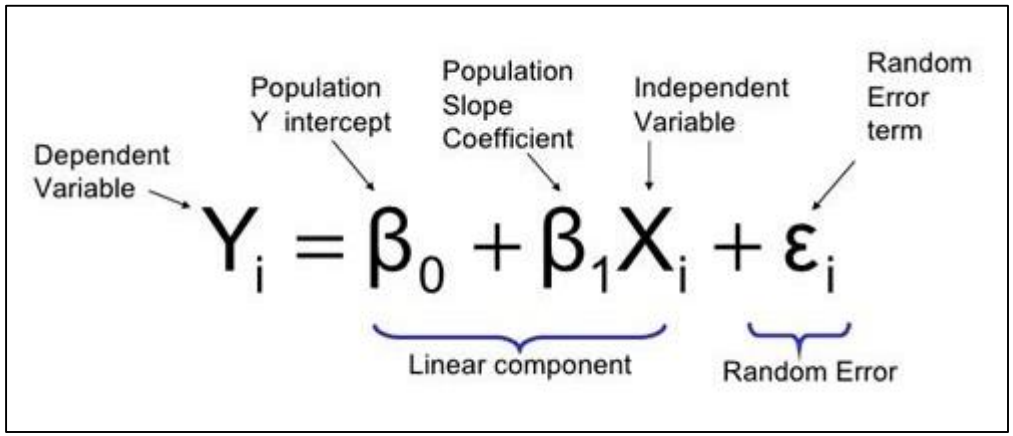


Deep Learning Algorithms

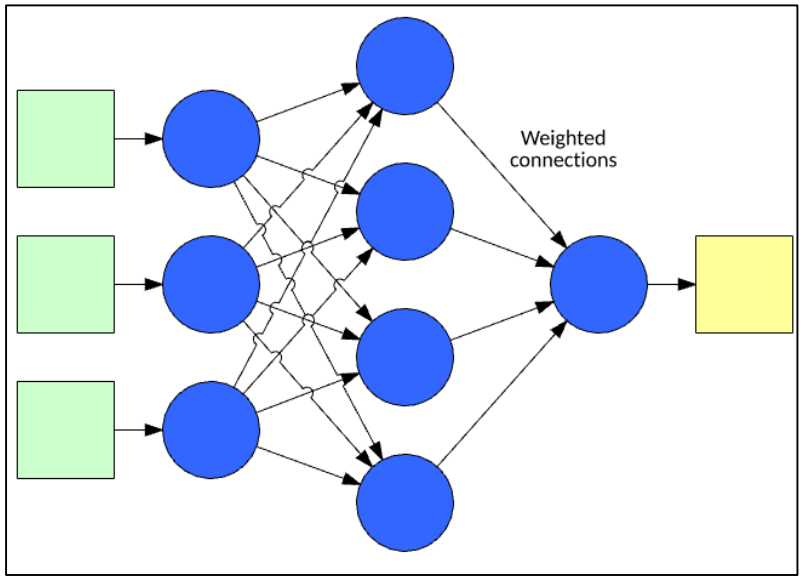


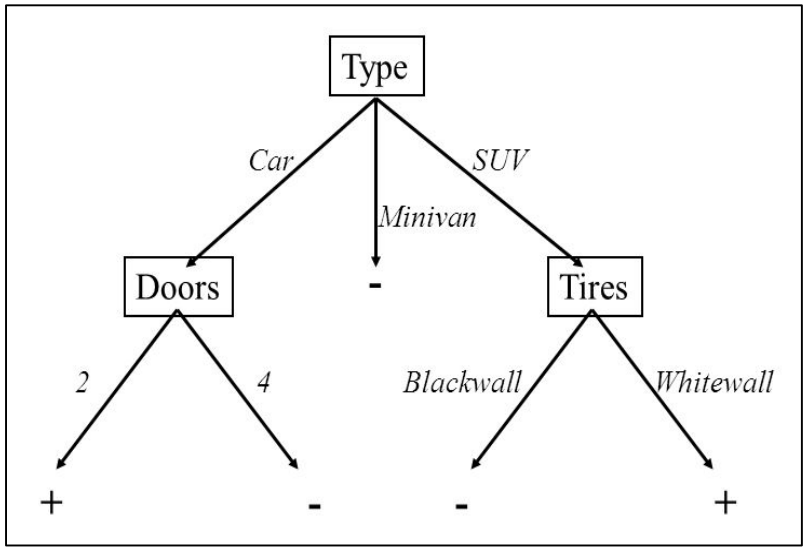
Learning Classifier Systems

Models/ML: Representation



- R1: **IF** the animal has hair
  THEN it is a mammal
- R2: **IF** the animal gives milk
  THEN it is a mammal
- R3: **IF** the animal has feathers
  THEN it is a bird
- R4: **IF** the animal flies
  the animal lays eggs
  THEN it is a bird
- R5: **IF** the animal is a mammal
  the animal eats meat
  THEN it is a carnivore





Models and the NFL

"All models are wrong, but some models are useful" – George Box

Our model is a simplification of reality

⇩

Simplification is based on assumptions (model bias)

⇩

Assumptions fail in certain situations

- Assumptions that work well in one domain may fail in another.
- No Free Lunch Theorem (NFL):
  - No single algorithm/model can perform optimally across all problems.
- Try:
  - More than one modeling approach
  - Different run parameters
    - "The knobs a data scientist gets to turn when setting up an algorithm to run"
  - Ensemble methods.

Where to start in selecting a method?

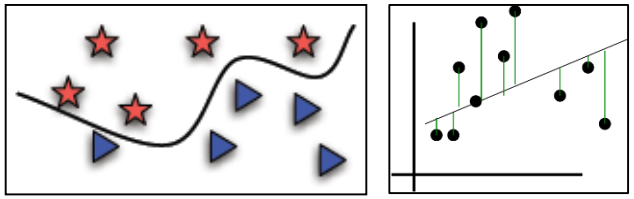
- If there is a strong, simple relationship among variables, most methods will find it.

- Generally start with simpler methods if you know nothing about the problem.

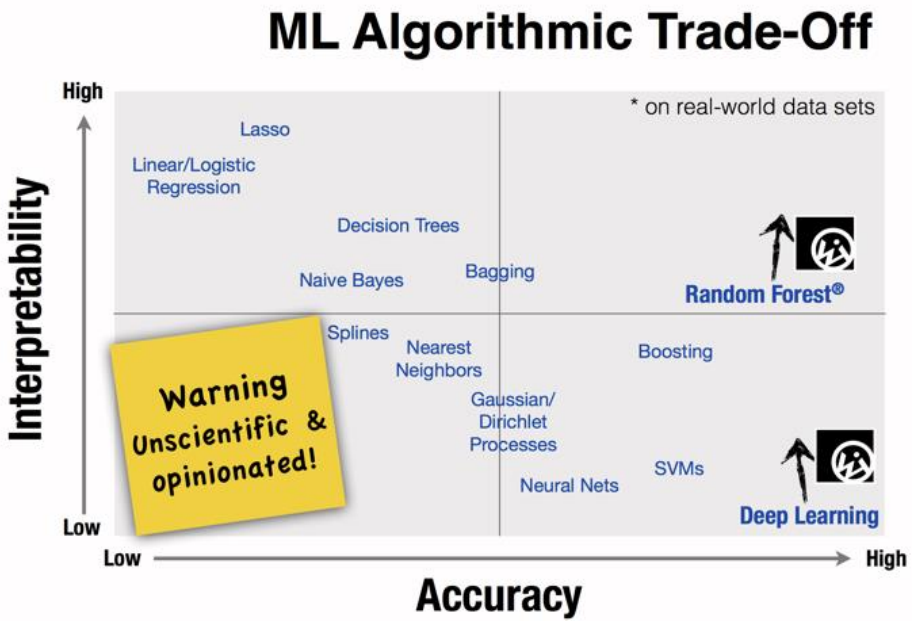
- When possible, limit the search space with knowledge/assumptions about the problem.
  - E.g. If we want to know if there are linear patterns, use linear regression.

- Incorrect assumptions will limit or invalidate what can be found.

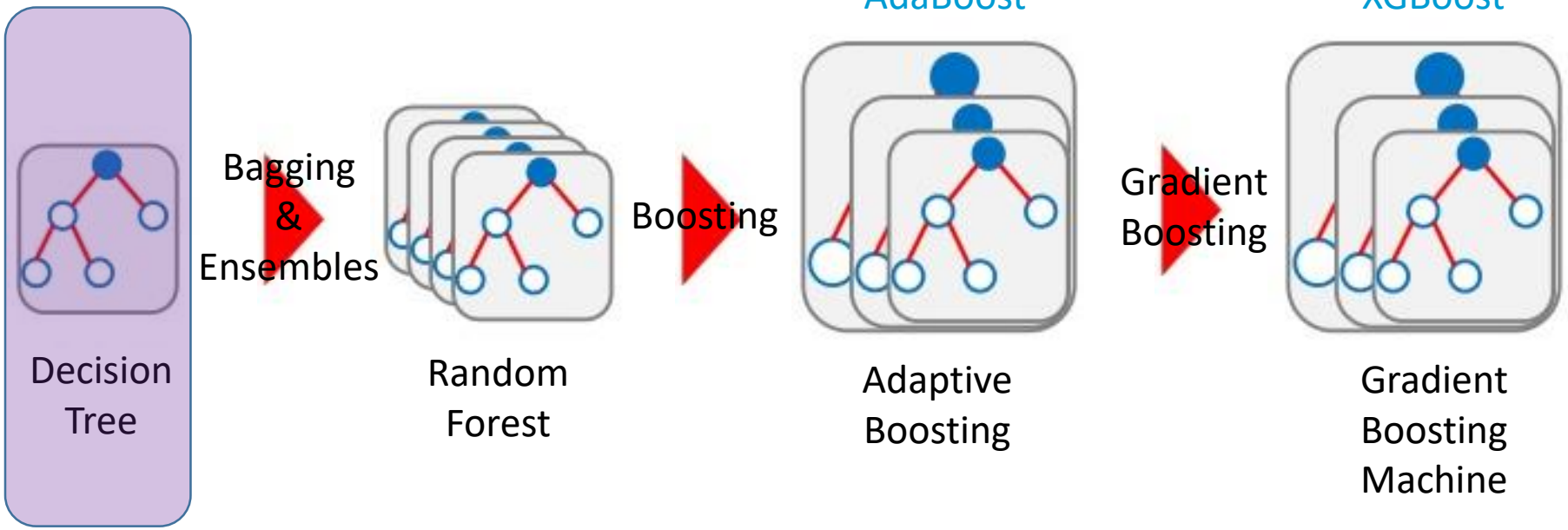
Considerations When Choosing an ML Algorithm



- Data – Labeled?, Endpoint?
- Training Time / Run Speed
- Number of Hyperparameters
- Data Size – Features, Instances
- Interpretability
- Assumptions
  - Linearity
  - Clean vs. Noisy Signal (potential to overfit)
  - Number of predictive vs. non-predictive features
  - Epistatic interactions and dimensionality of interactions
  - Heterogeneous associations

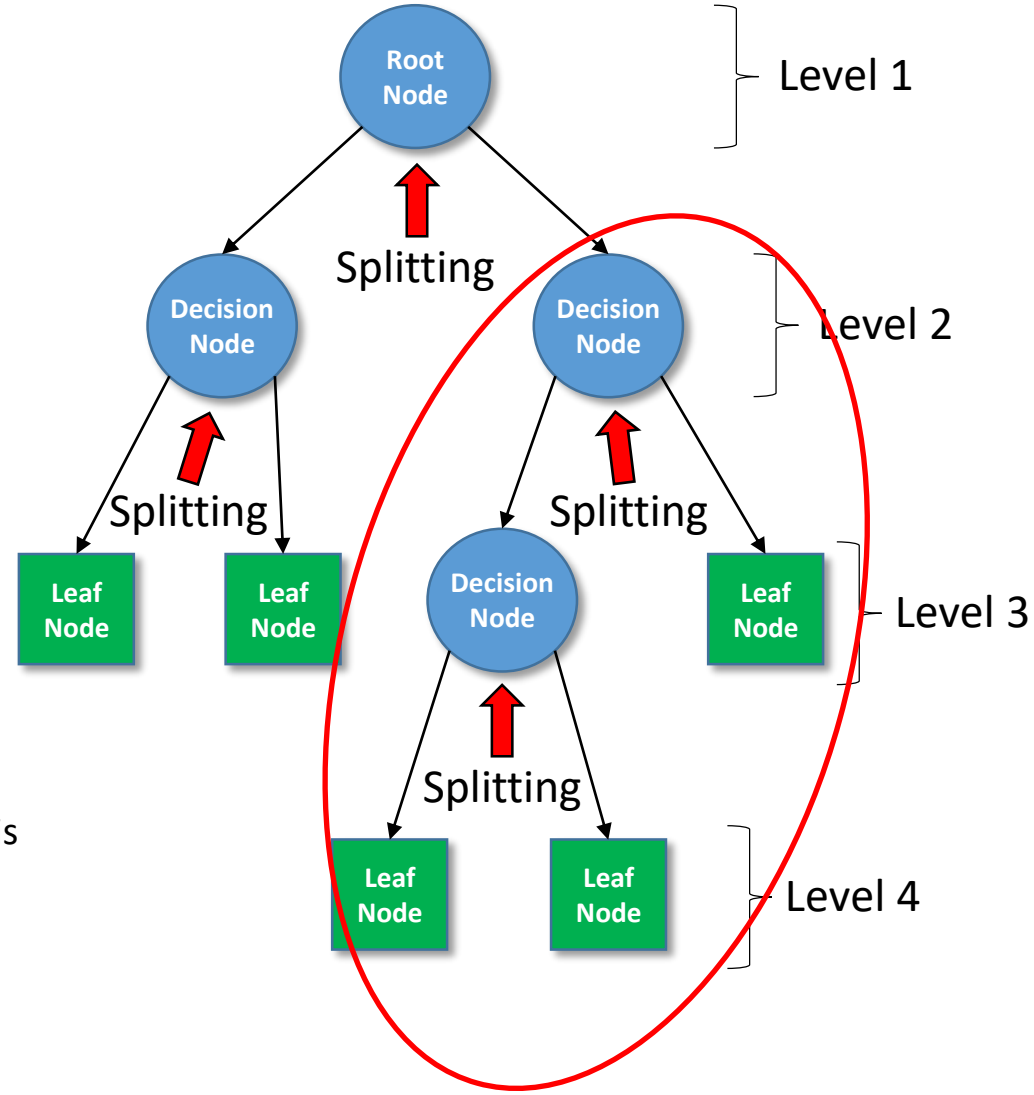


Down the Tree-based Modeling Rabbit Hole



Decision Tree: Terminology

- **Nodes:**
  - **Root:** It represents entire population or sample. Will get divided into two or more homogeneous sets.
  - **Decision:** When a sub-node splits into further sub-nodes, then it is called decision node.
    - (AKA: Sub, internal, split, or chance node)
  - **Leaf:** Nodes that don't split. Gives class or average value.
    - (AKA: Terminal, or outcome node)
  - **Parent and Child:** Parent node splits into offspring nodes.
- **Splitting:** It is a process of dividing a node into two or more sub-nodes.
- **Branch / Sub-Tree:** A sub section of entire tree is called branch or sub-tree.
- **Levels/Depth:** The number of splits through a given path down the three.

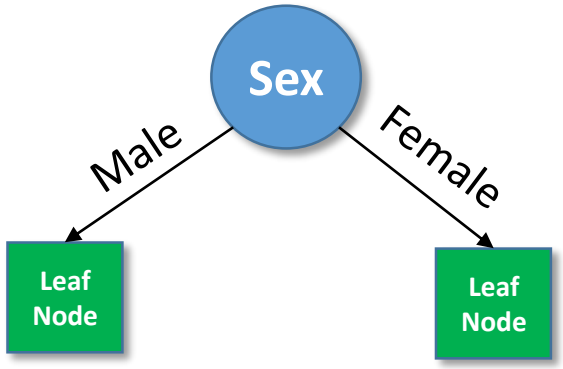


Decision Tree: Training the Model

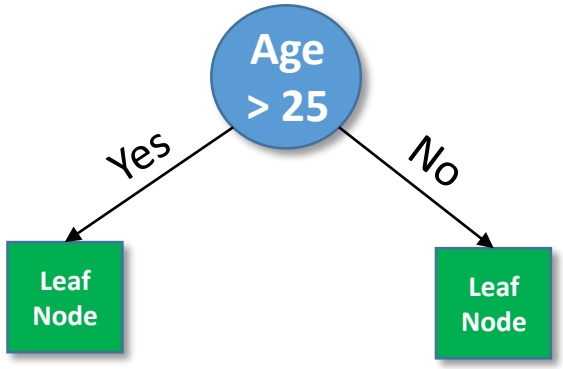
- Begin at 'root' node
- Recursively finds a variable that best divides data into outcomes. Hunt's Algorithm
- 'best' variable is determined heuristically
  - Gini Index (e.g. CART)
  - Information Gain (e.g. ID3, C4.5)
  - Chi Square
  - Variance reduction (e.g. CART regression)
- Heuristics: produce spits as homogeneous (pure) as possible in terms of outcome labels
- Stop criterion: Max depth, purity of labels in each leaf

Decision Tree: Splitting by…

- Categorical Features
  - E.g. sex (male/female)
  - Split defined by discrete value

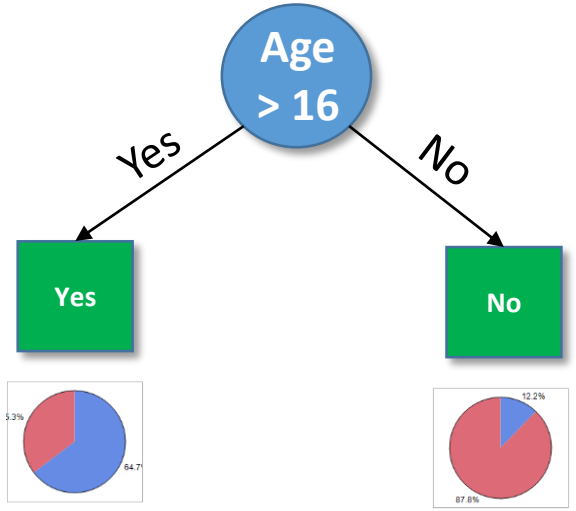


- Continuous-Valued Features
  - E.g. age (> 25)
  - Split defined by threshold

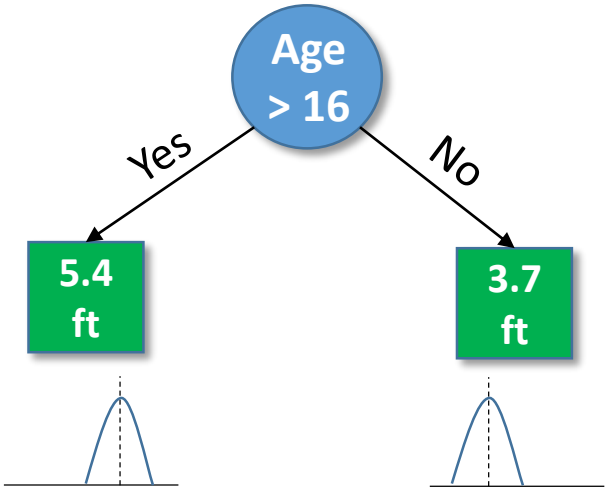


Decision Trees: By Outcome

- Classification Tree:
  - Categorical Outcome
  - E.g. Can a person drive?
  - Class proportions indicates class probability.
  - Prediction typically the class majority
  - Can be ‘multi-class’.



- Regression Tree
  - Continuous Outcome
  - E.g. Height (ft)
  - Average height of individuals in given leaf node.

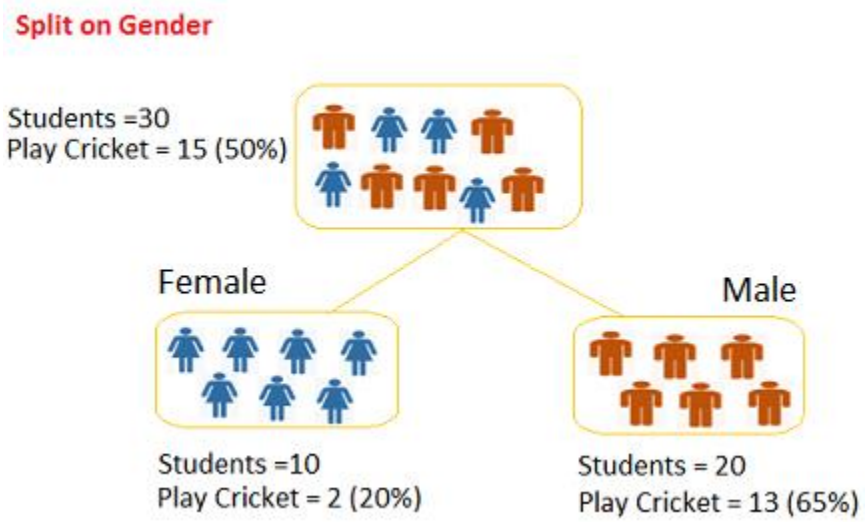


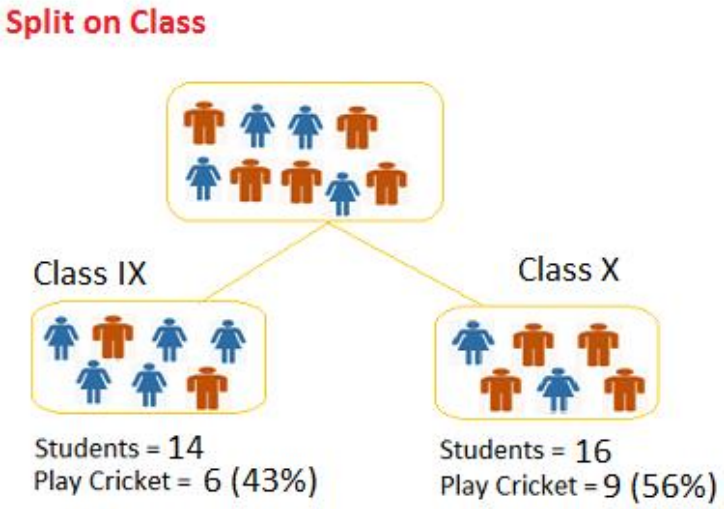
Decision Tree: Split by - Gini Index

- Gini impurity Metric calculated for each sub-node
- Used by CART
- Zero is best score
- Minimal Impurity = Homogeneous split

- **Split on Gender:**
- Calculate, Gini for sub-node Female = (0.2)*(0.2)+(0.8)*(0.8)=0.68
- Gini for sub-node Male = (0.65)*(0.65)+(0.35)*(0.35)=0.55
- Calculate weighted Gini for Split Gender = (10/30)*0.68+(20/30)*0.55 = **0.59**

- **Similar for Split on Class:**
- Gini for sub-node Class IX = (0.43)*(0.43)+(0.57)*(0.57)=0.51
- Gini for sub-node Class X = (0.56)*(0.56)+(0.44)*(0.44)=0.51
- Calculate weighted Gini for Split Class = (14/30)*0.51+(16/30)*0.51 = **0.51**





Decision Tree: Regression Split

- Regression Trees
  - Mean Squared Error
    - Sum of squared errors from average

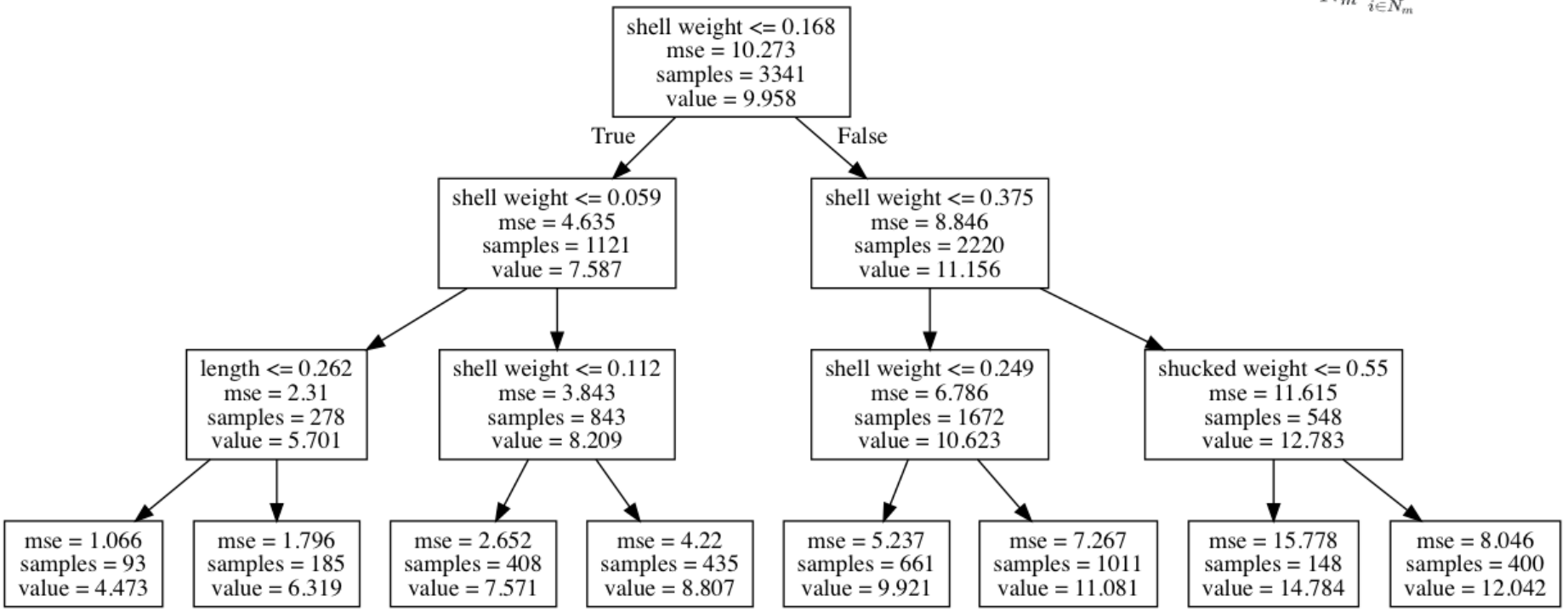
$$\bar{y}_m = \frac{1}{N_m} \sum_{i \in N_m} y_i$$

$$H(X_m) = \frac{1}{N_m} \sum_{i \in N_m} (y_i - \bar{y}_m)^2$$

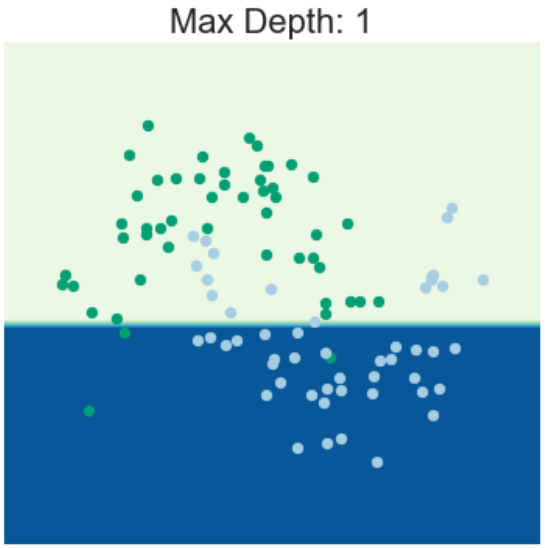
  - Mean Absolute Error
    - Sum of errors from average

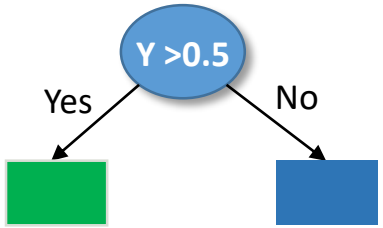
$$\bar{y}_m = \frac{1}{N_m} \sum_{i \in N_m} y_i$$

$$H(X_m) = \frac{1}{N_m} \sum_{i \in N_m} |y_i - \bar{y}_m|$$

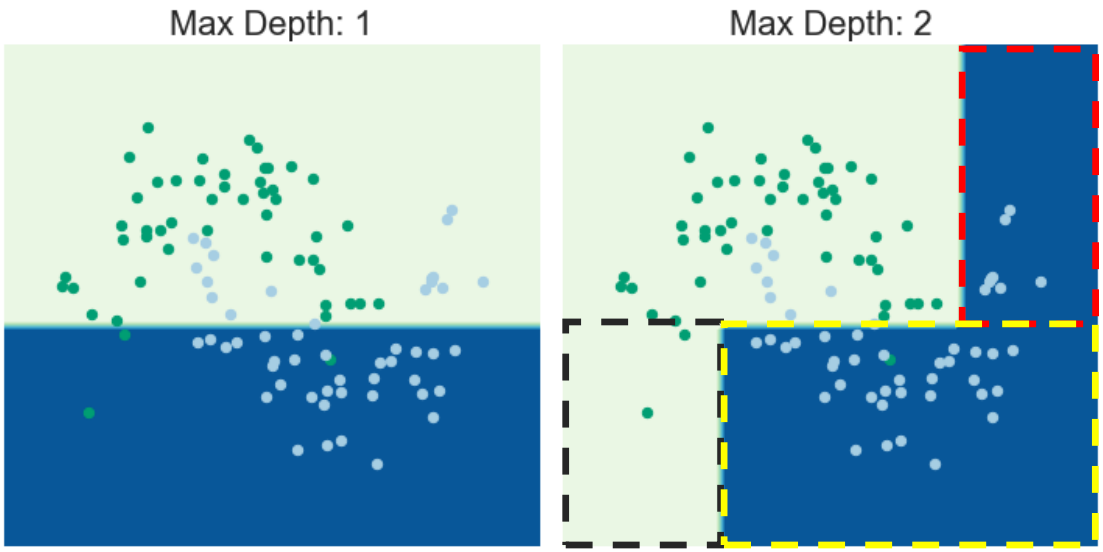


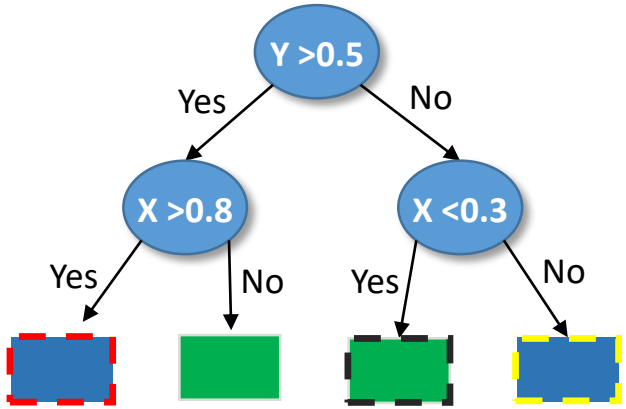
Decision Tree: Fitting with Splits



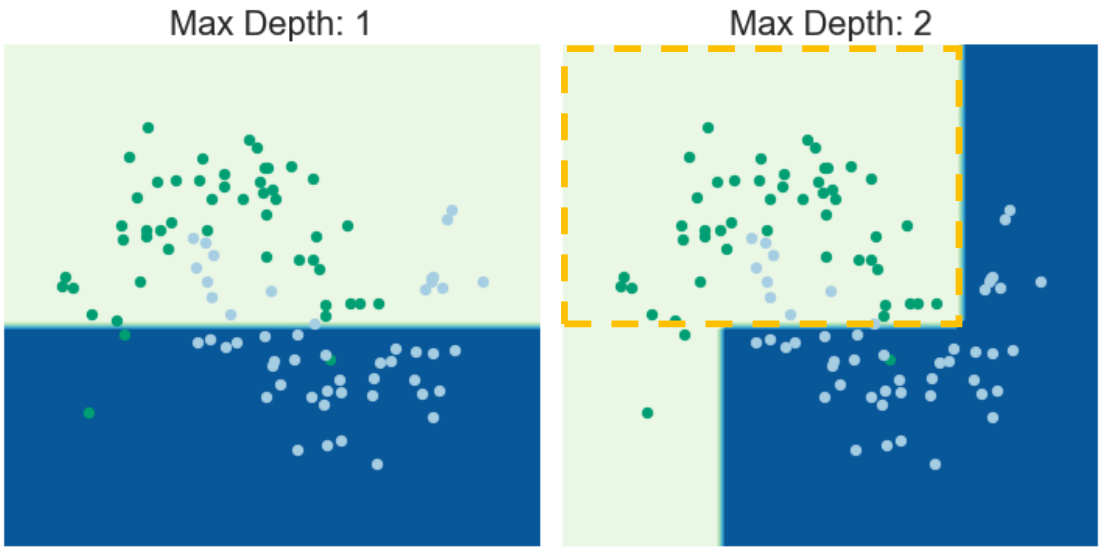


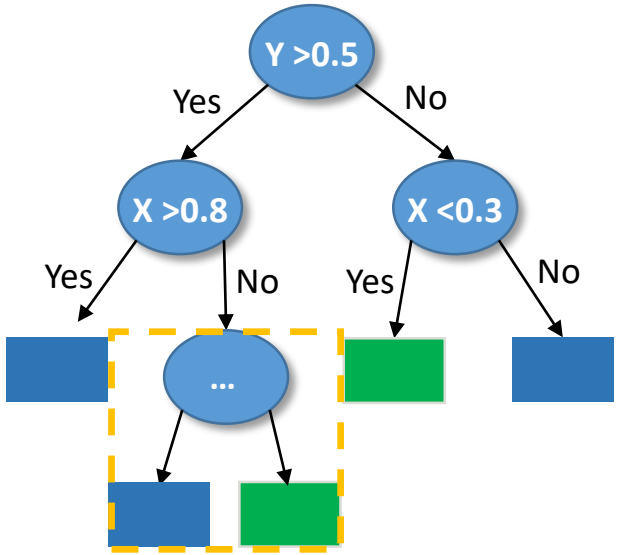
Decision Tree: Fitting with Splits





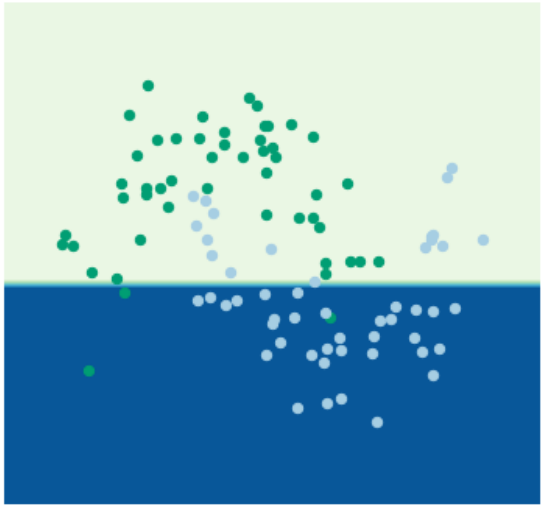
Decision Tree: Fitting with Splits



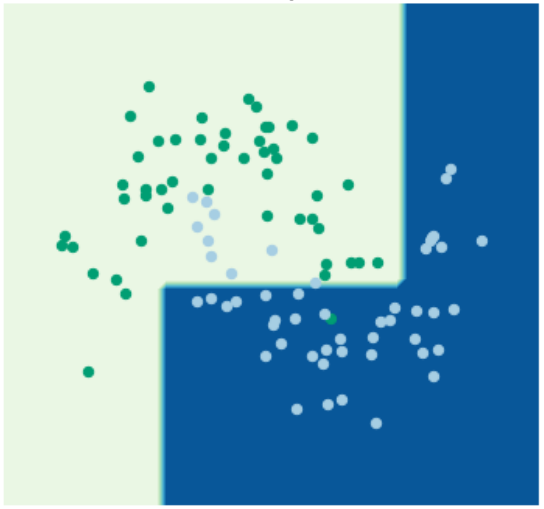


Decision Tree: Fitting with Splits

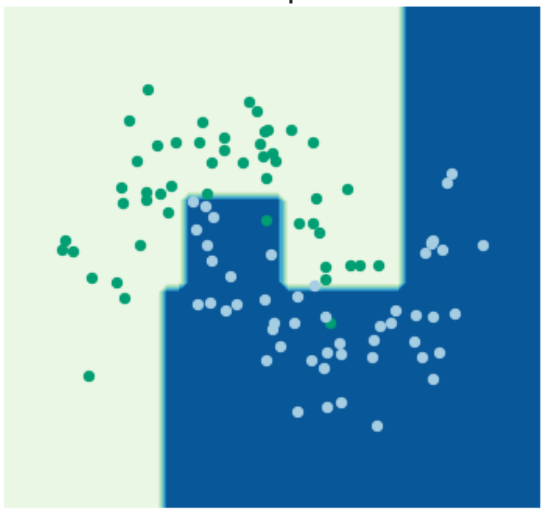
Max Depth: 1

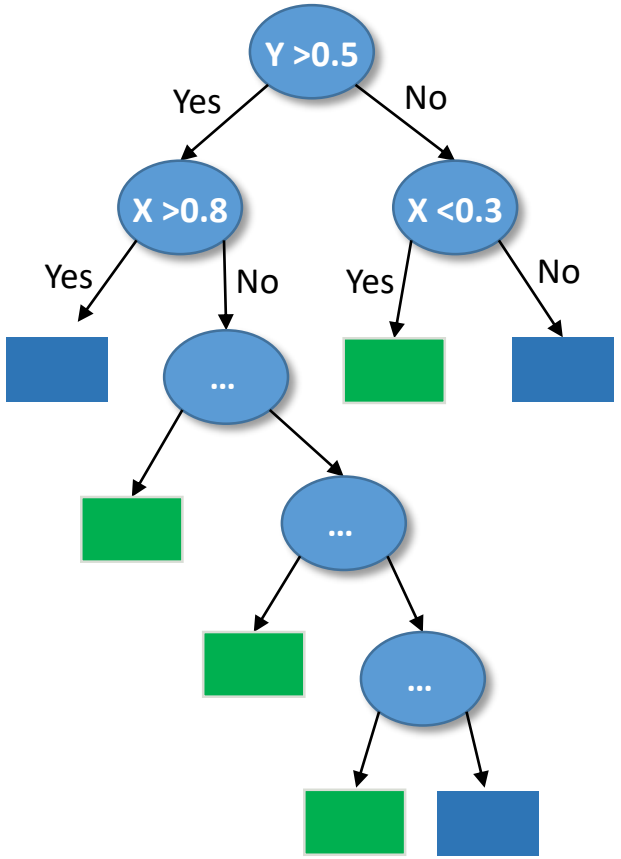


Max Depth: 2

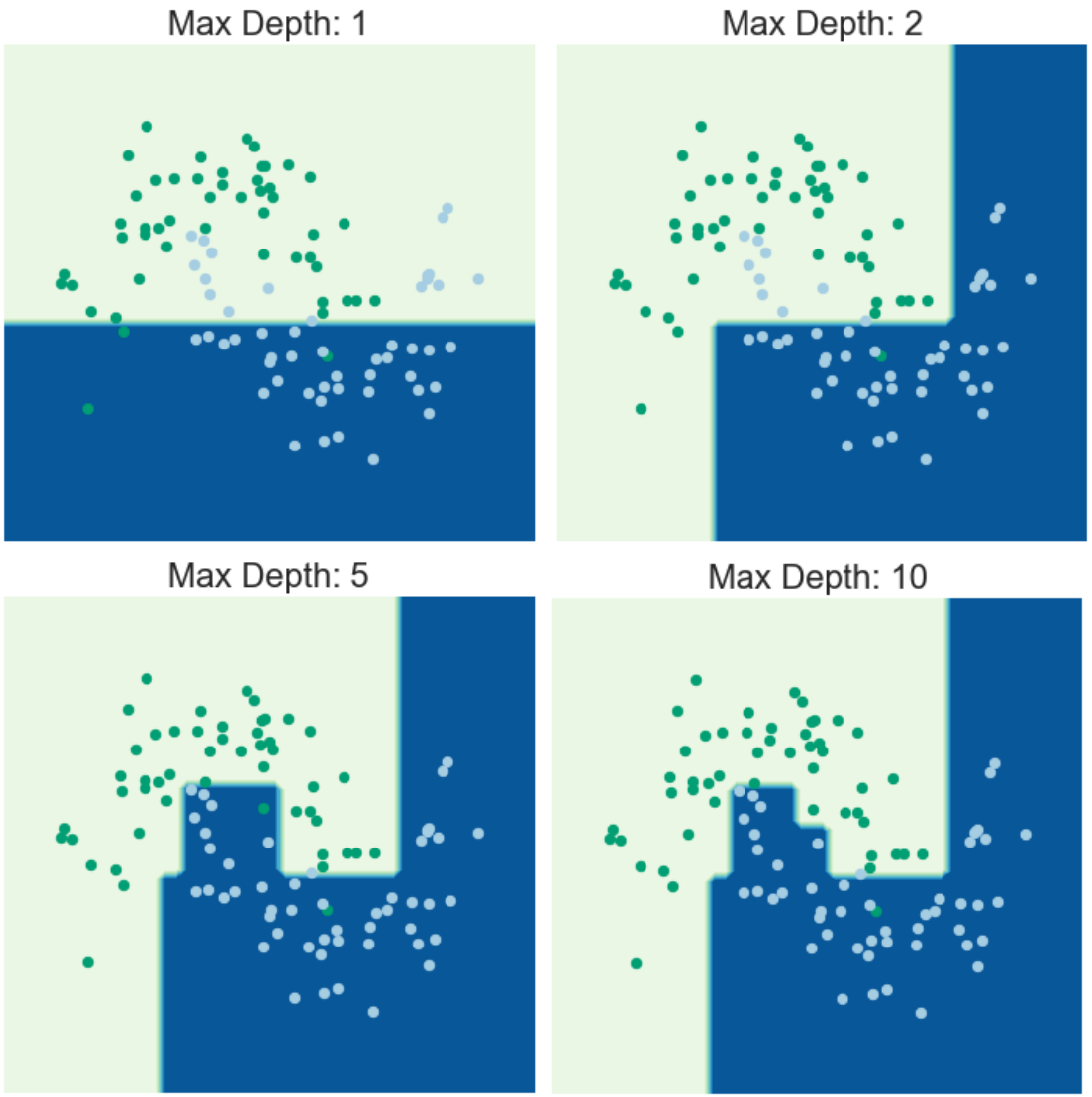


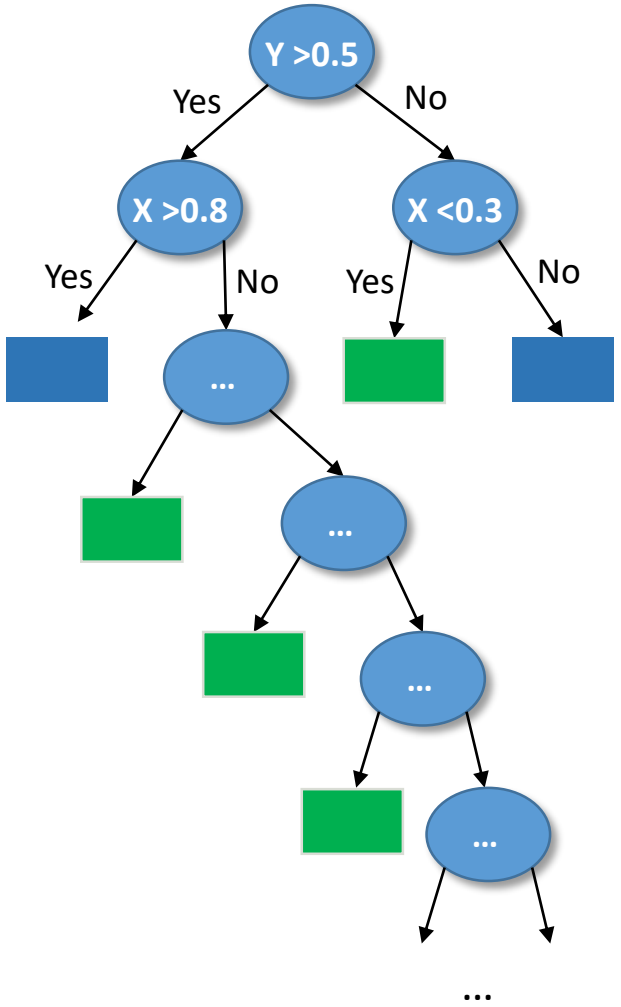
Max Depth: 5



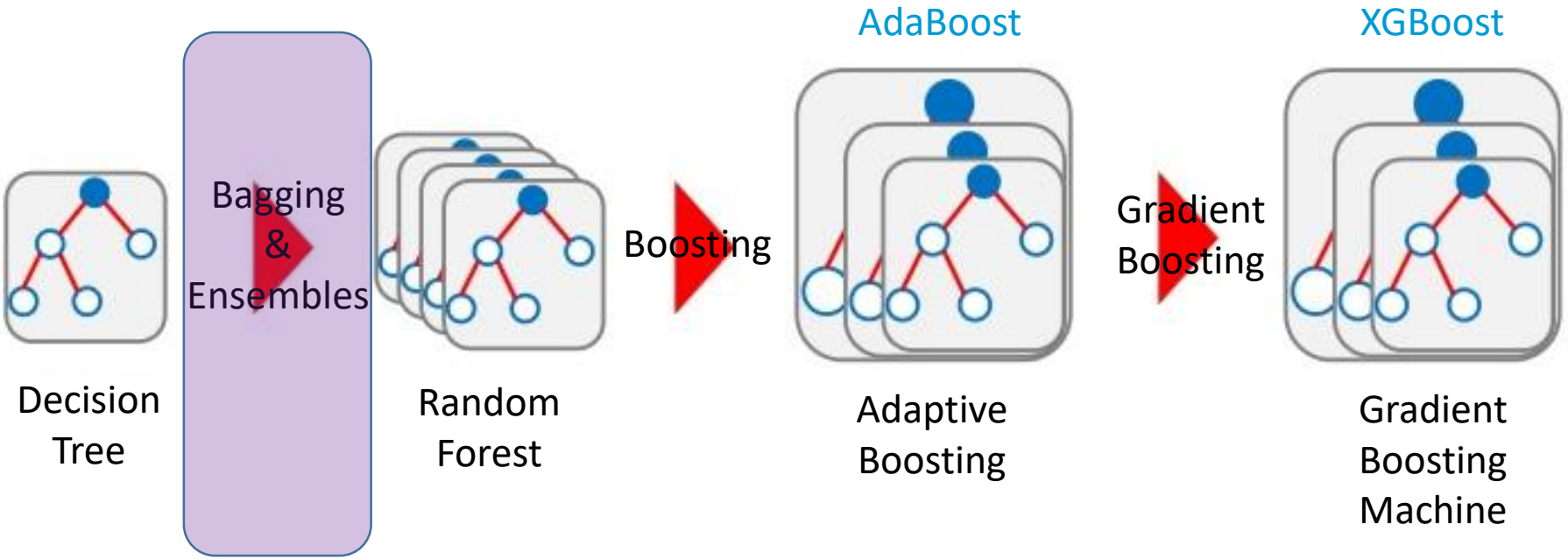


Decision Tree: Fitting with Splits





Down the Tree-based Modeling Rabbit Hole



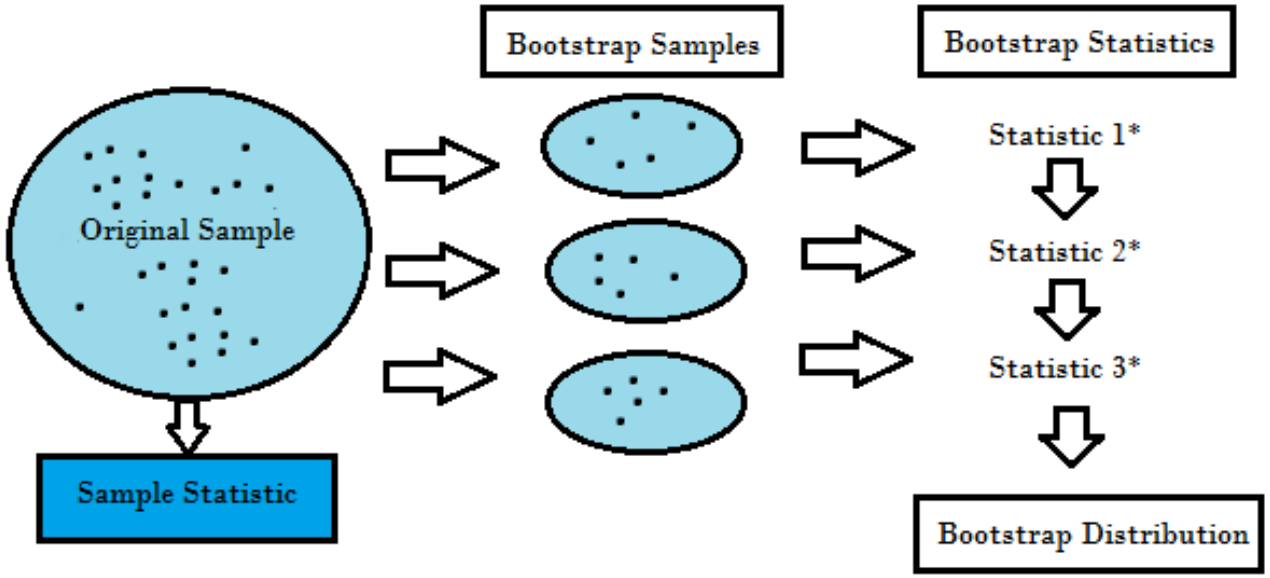
Ensembles

- An ensemble is simply a collection (i.e. ‘group’) of models trained on the same task
- Many versions of same model or different types of models
- Final output: weighted average or vote
- Ensemble of different models that achieve similar generalization performance often outperforms any individual models. How?...

Ensembles – Why they work?

- Suppose we have a set of binary classifiers
- Each classifier has the same average error that is better than randomly guessing
- Assume the errors they make are independent
- Intuition: the majority of the classifiers will be correct on many examples where any individual classifier makes a mistake
- A simple majority vote can improve classification performance by decreasing variance in this setting
- How do we train such an ensemble?

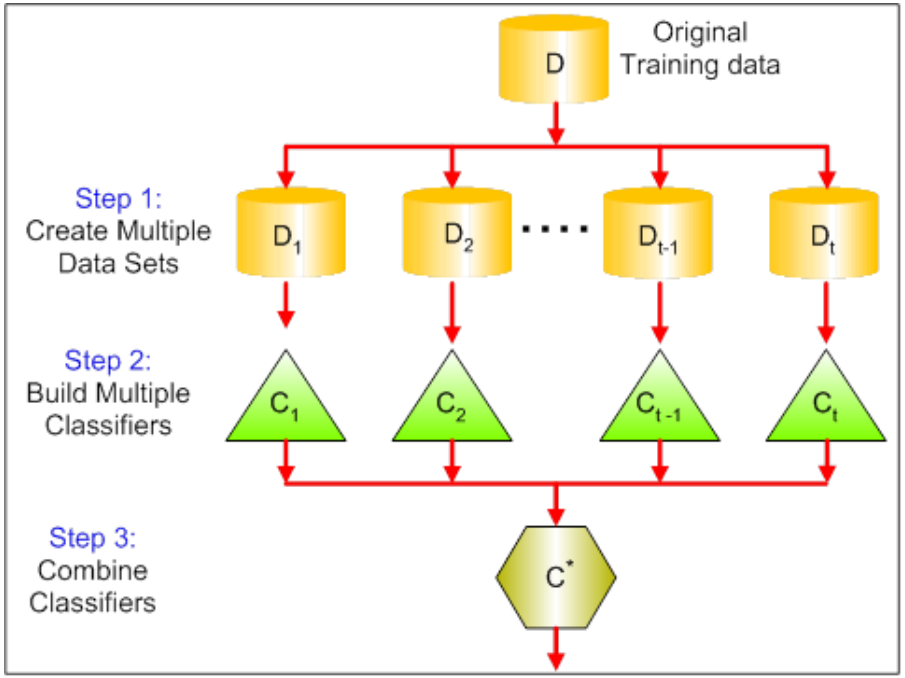
Precursor: Bootstrapping



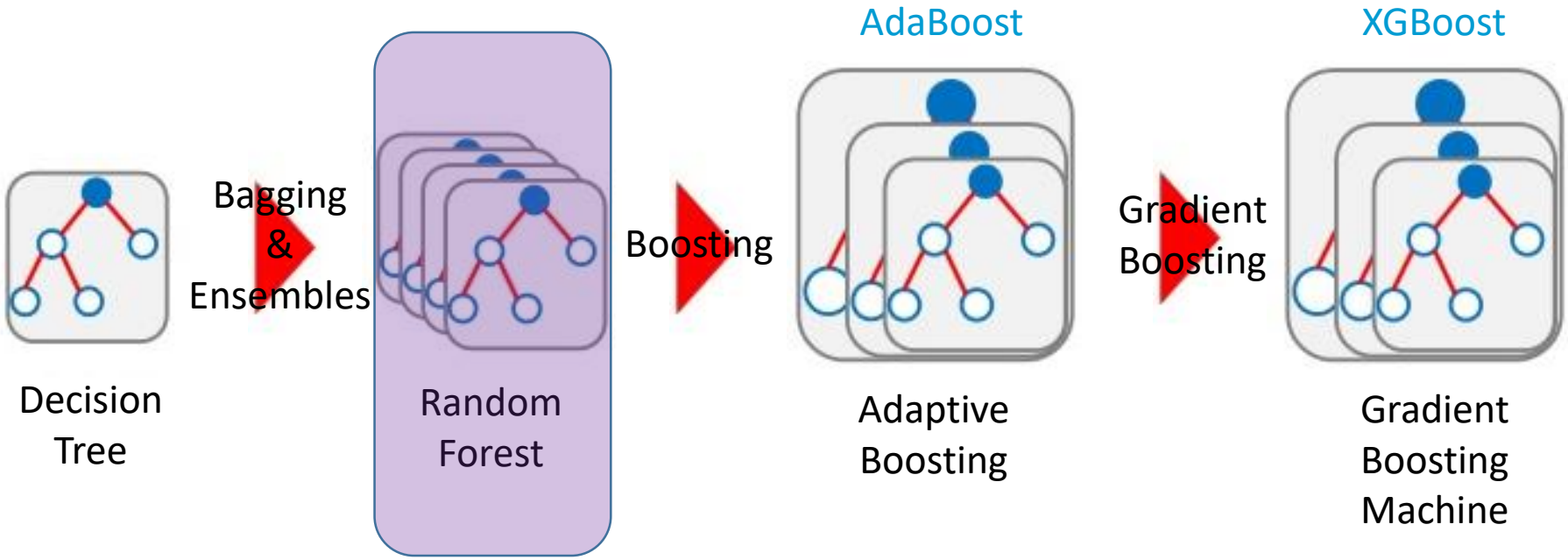
- Widely used to quantify uncertainty associated with model
  - Standard error
  - Confidence interval for coefficient
- Obtain many datasets of the same size as original by sampling with replacement
- Obtain statistics of interest and look at their distribution to estimate variability

Bagging / Bagged Trees

- AKA: Bootstrap aggregation
- Goal: Reduce variance
- Attempts to train independent classifiers by sampling the training set
- Sample k times with replacement
- Train k classifiers on subsets
- Useful for high-variance, high-capacity models (i.e. decision trees)
- Higher number of models always better (trade off in run time)
- Internal error estimate: use the portion of data that wasn't built to create model as test set (out of bag data)



Down the Tree-based Modeling Rabbit Hole



Random Forests

- Random forests are a successful extension of bagged trees
- Considered to be panacea of all ML problems.
  - Go-to algorithm
  - Generally works pretty well across most problems.
- Extension from Bagging: Considers random sub-set of features when deciding which variables to split on.
- When given new data, pass it to all trees in forest and estimate class based on most popular outcome.
- Feature Importance: estimate of single variable contribution to classification.

Random Forest - Illustration

Bootstrap sampling

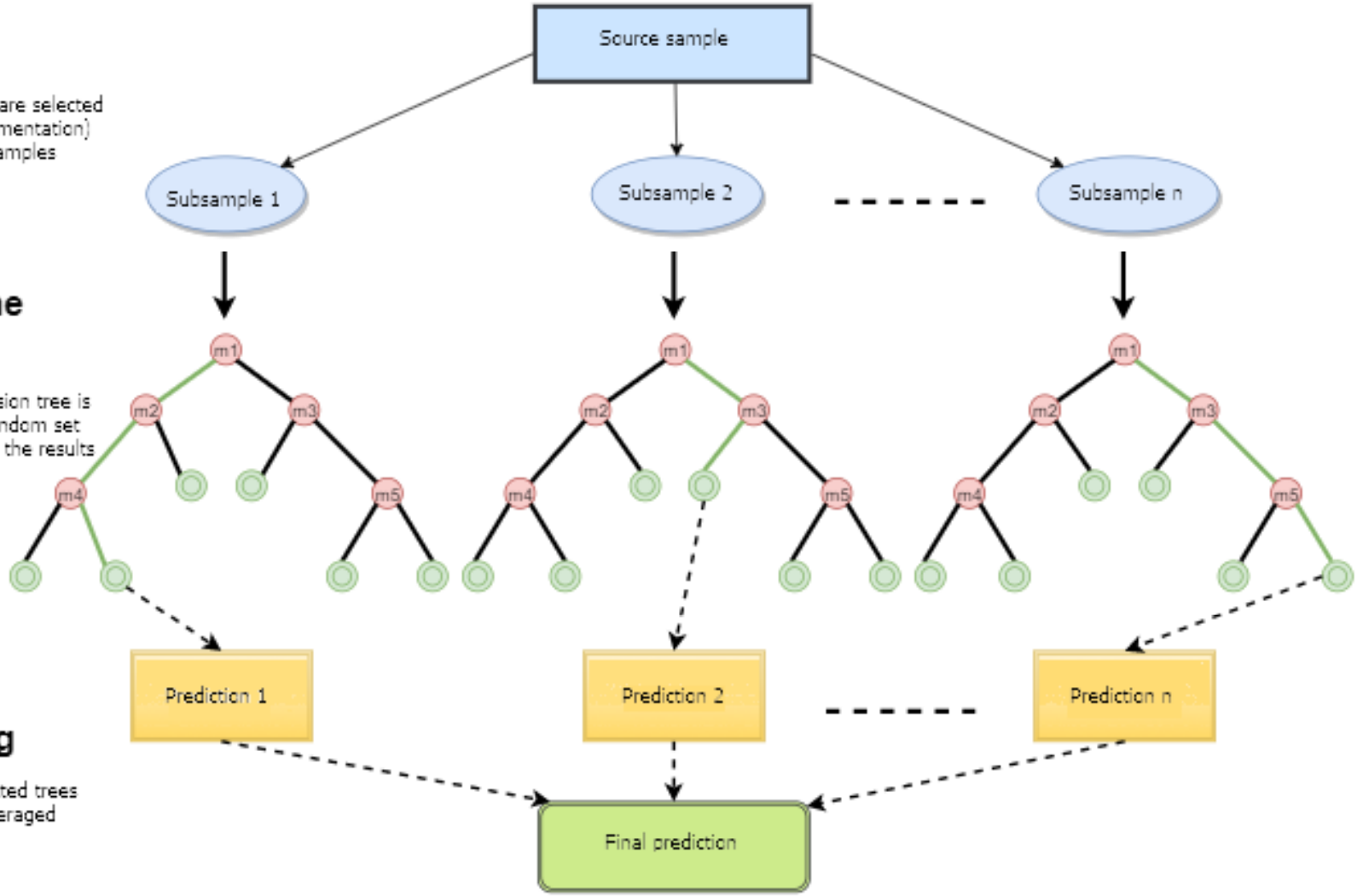
r (percentage) examples are selected (0.63 in classical implementation) in n random subsamples

Building the models

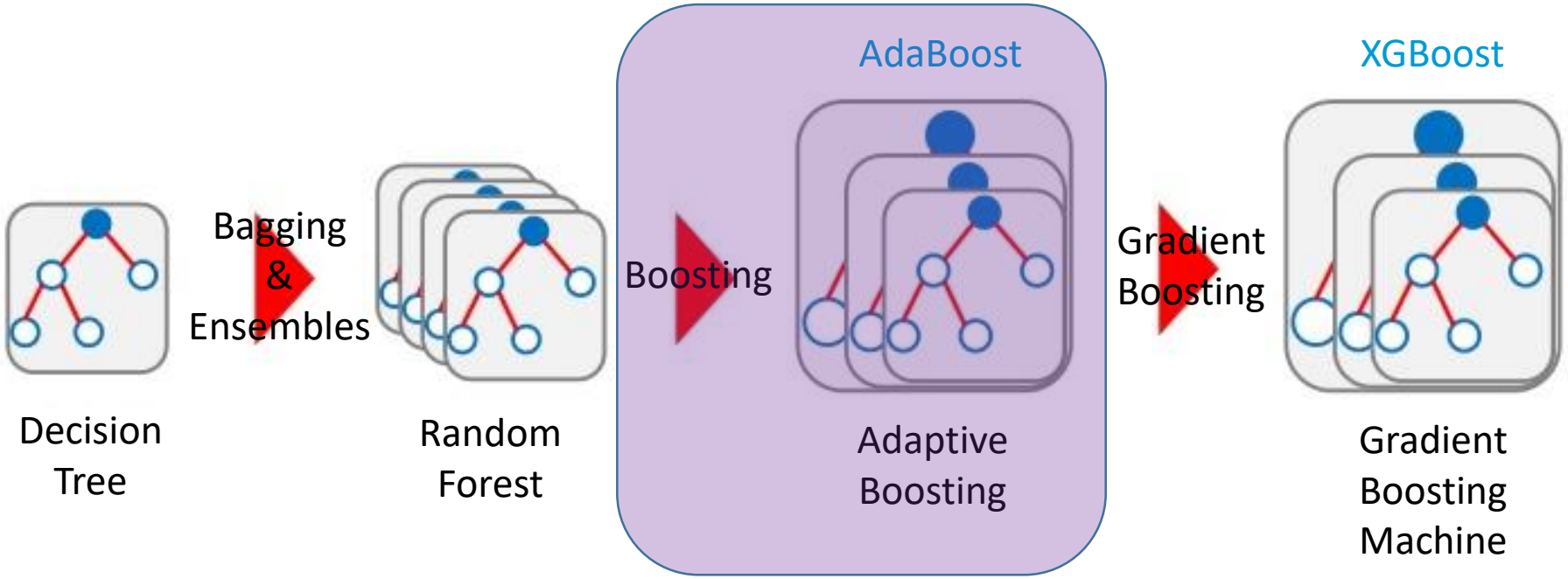
for each subsample, a decision tree is constructed based on a random set of m features (covariants), the results fall into leaves

Bootstrap aggregating

results from all constructed trees are gathered and averaged

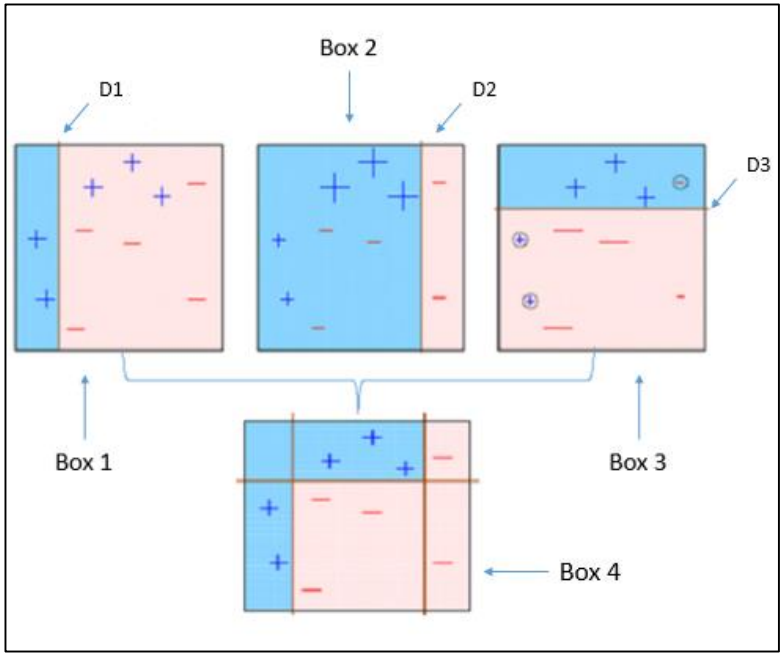


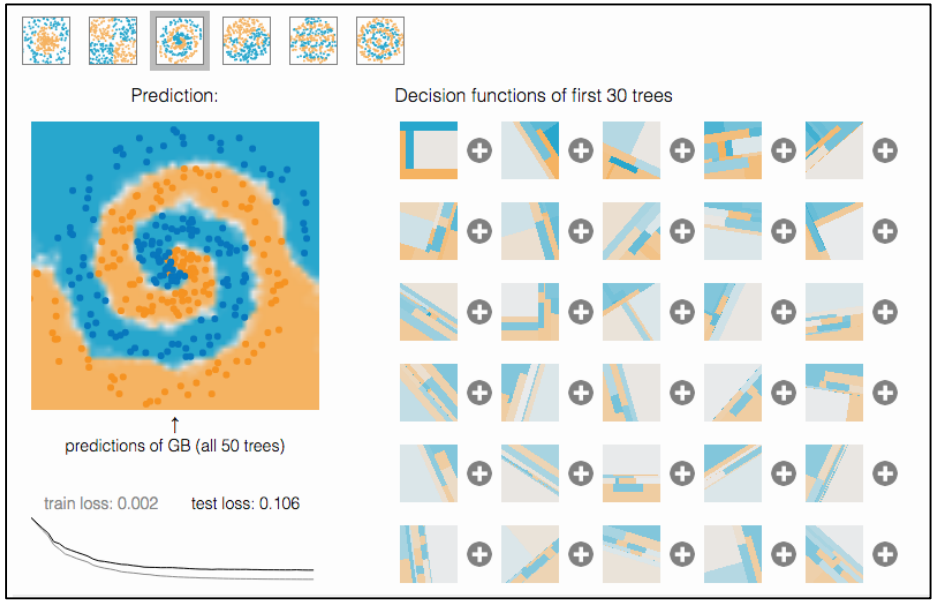
Down the Tree-based Modeling Rabbit Hole



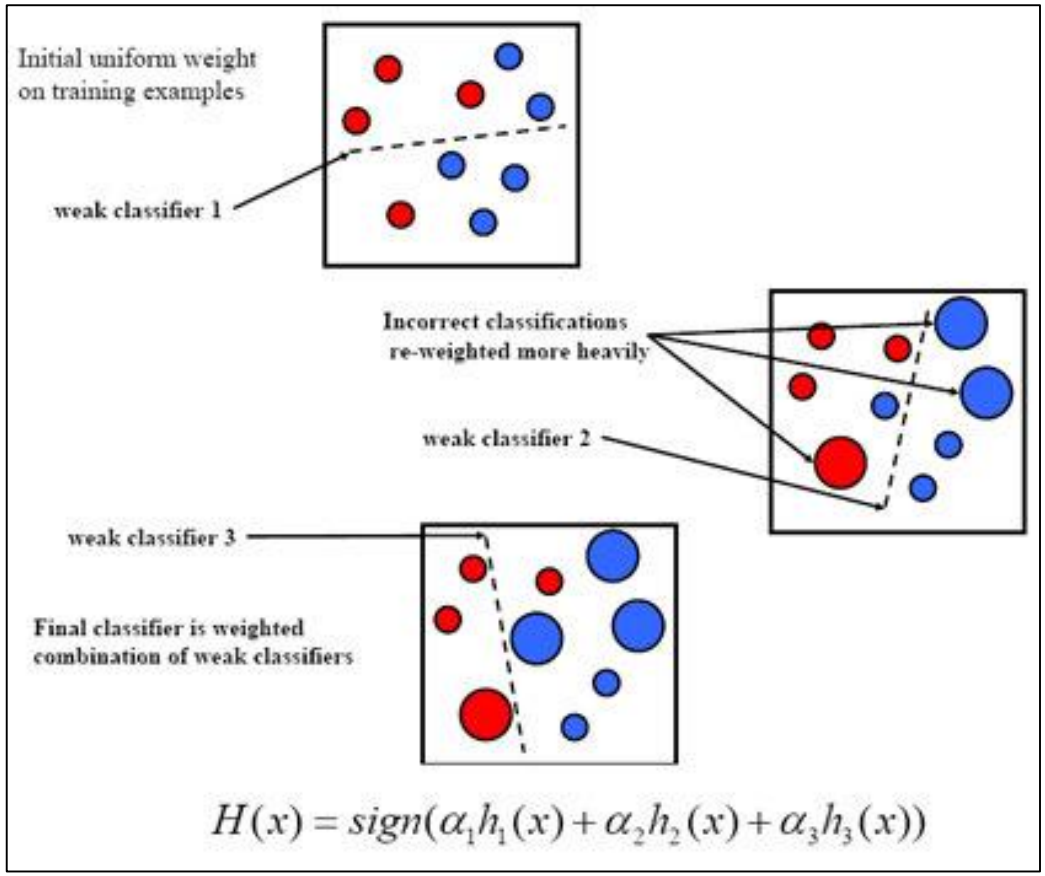
Boosting

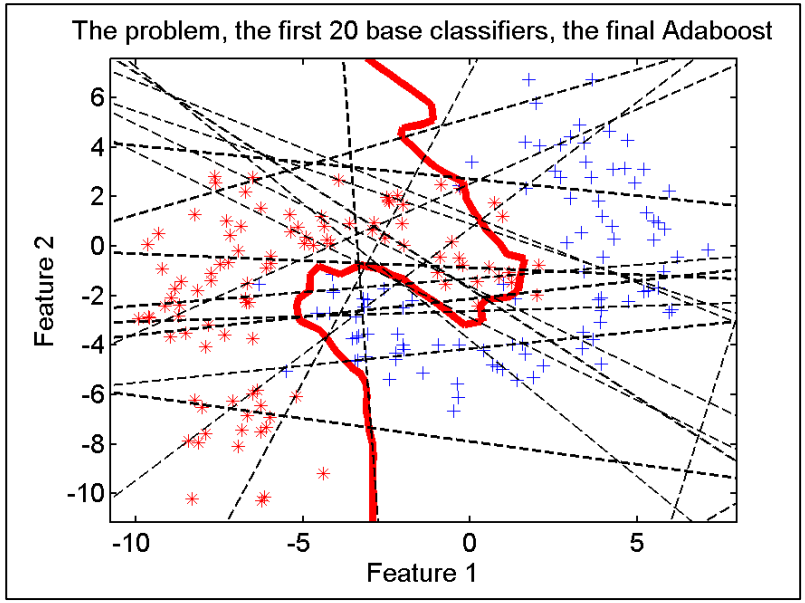
- Ensemble meta-algorithm for combining weak learners into a strong one.
- Iteratively learning weak learners (e.g. 1 level decision tree), sequentially
- Each iteration, data weights are updated to focus more on misclassified instances.
- Easily defeated by noisy data and outliers



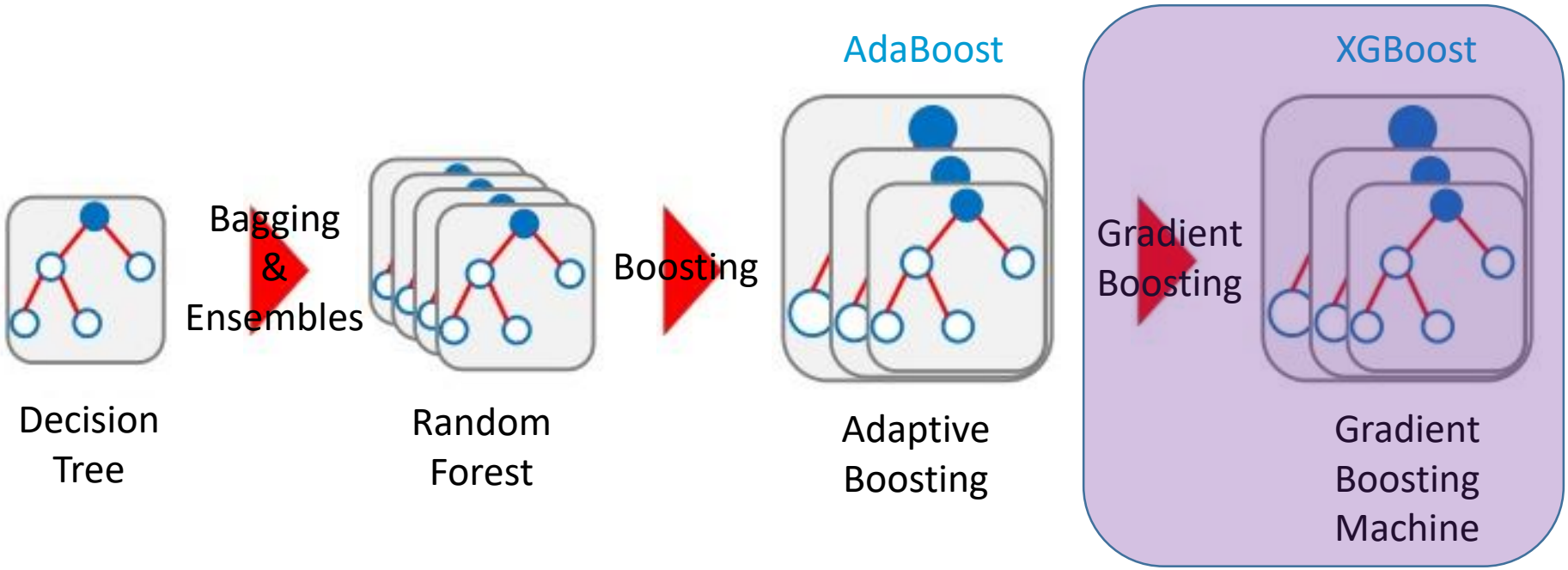


ADABoost – Adaptive Boosting





Down the Tree-based Modeling Rabbit Hole



GBM - Gradient Boosting Machine

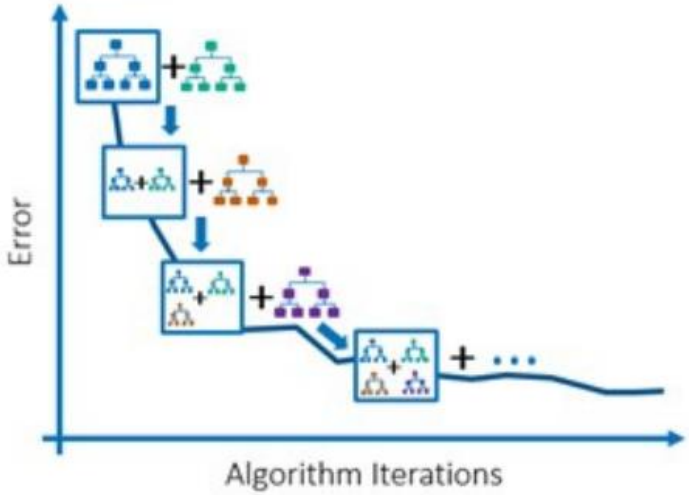
- E.g.
  - MART (Multiple Additive Regression trees)
  - GBRT (Gradient Boosted Regression Trees)
  - XGBoost

- Instead of training on weighted instances, the weak learner trains on the remaining errors (so-called pseudo-residuals) of the strong, meta, learner.

- Another way of giving importance to the difficult instances.

- Contribution of weak learner to the strong one isn't computed according to its performance on the new distribution sample, but using a gradient descent optimization process.

- The computed contribution is the one minimizing the overall error of the strong learner.



XGBoost – Extreme Gradient Boosting

- A regularized Gradient boosting approach.
  - (L1 and L2 regularization) – prevent overfitting
- Parallel computing (10x faster than gradient boosting
- Faster

HCC Modeling [Notebook]

- Applied:
  - Decision Tree
  - Random Forest
- Scikit-learn
- Default Hyperparameters
- Evaluate:
  - Basic Accuracy Metric
  - Training and Testing Accuracy

Decision Tree
Training Accuracy: 1.0
Testing Accuracy: 0.696969696969697

Random Forest
Training Accuracy: 0.9848484848484849
Testing Accuracy: 0.7272727272727273

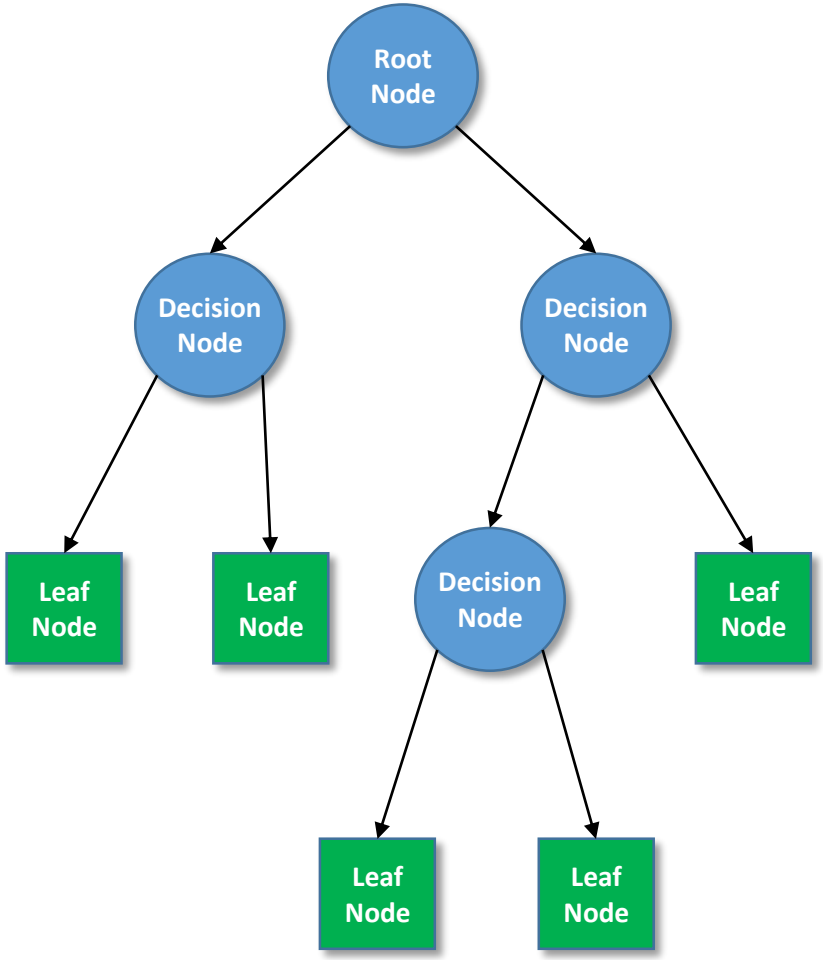
Modeling: Hyperparameter Sweep

Decision Tree Challenges

- How do we decide best feature or value to split on (criterion)?
- When should we stop splitting?
- What do we do if we can't achieve perfect classification?
- What if the tree is too large, complex, and/or overfit?
- With sufficient features: we could generate a tree with one path to a leaf for each instance/subject.
  - But this is just memorization.
- We want to generalize to new examples (avoid overfitting).

Decision Tree Hyperparameters [Notebook]

- Scikit-learn: Classification Decision Tree
  - `model = tree.DecisionTreeClassifier()`
- Hyperparameters:
  - max_depth = The maximum depth of the tree. (default = None)
  - min_samples_split = The minimum number of samples required to split an internal node: (default = 2)
  - min_samples_leaf = The minimum number of samples required to be at a leaf node. (default = 1)
  - criterion = The function to measure the quality of a split. (Default = 'gini')

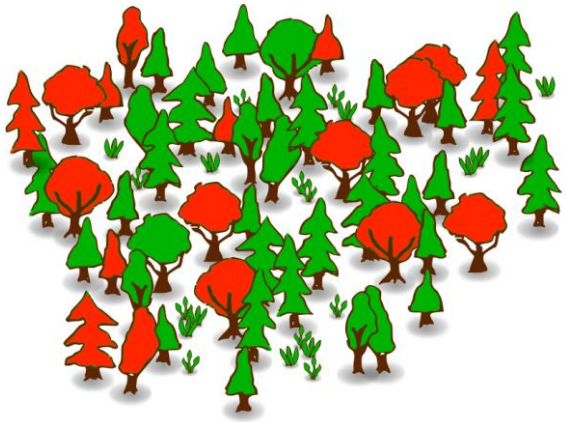


RandomForest Hyperparameter [Notebook]

- Scikit-learn: Classification Random Forest
  - `model = RandomForestClassifier()`

- Hyperparameters (Same as decision trees):
  - max_depth = The maximum depth of the tree. (default = None)
  - min_samples_split = The minimum number of samples required to split an internal node: (default = 2)
  - min_samples_leaf = The minimum number of samples required to be at a leaf node. (default = 1)
  - criterion = The function to measure the quality of a split. (Default = 'gini')

  - n_estimators = The number of trees in the forest. (default = 10)

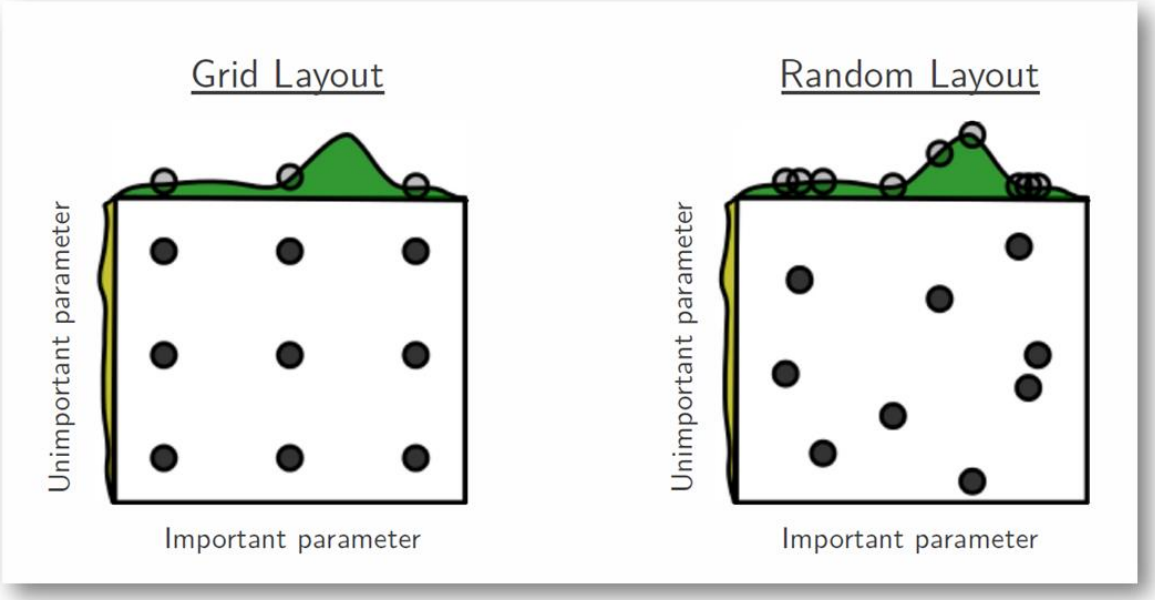


Default Hyperparameters

- Problem:
  - Default hyperparameters are intended to offer a starting point, often selected to perform well on a simple toy/example dataset.
- Solution:
  - Use existing ML theory to identify more appropriate hyperparameters for the given problem.
  - Perform a hyperparameter sweep! (i.e. a search of the hyperparameter combination space.

Hyperparameter Sweep

- Strategies:
  - Grid Search
  - Random Search
  - Evolutionary Search

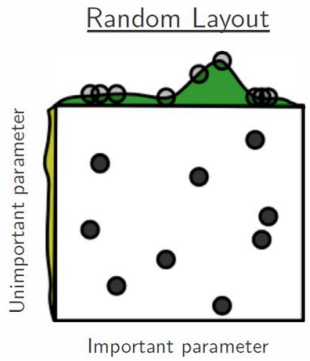


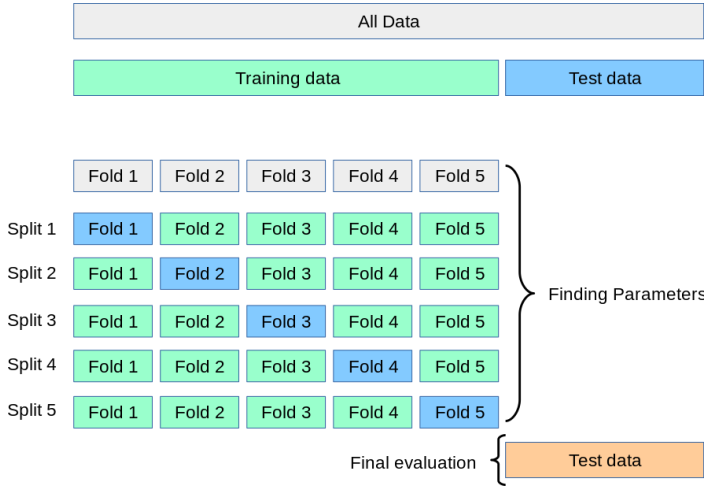
https://github.com/EpistasisLab/tpot

Hyperparameter Sweep in Modeling

1. Identify key hyperparameters to search/optimize
2. Identify hyperparameter values/ranges
3. Select sweep strategy
4. Apply sweep using k-fold cross validation to evaluate
5. Identify hyperparameter combination with best validation accuracy.
6. Train model using entire training dataset applying ‘optimized’ hyperparameter values.

Random Hyperparameter Sweep [Notebook]





`RandomizedSearchCV(n_iter=100)`

3-fold CV used

- Decision Trees Parameter Grid:
  - max_depth: [3, None]
  - min_samples_split: randint(2, 10)
  - min_samples_leaf: randint(1, 10)
  - criterion: ["gini", "entropy"]

- Random Forest Parameter Grid (adds…)
  - n_estimators: randint(2, 1000)

DT Average validation accuracy = 0.6818

- max_depth: 3
- min_samples_split: 5
- min_samples_leaf: 3
- criterion: 'gini'

RF Average validation accuracy = 0.6818

- n_estimators: 451
- max_depth: 3
- min_samples_split: 9
- min_samples_leaf: 4
- criterion: 'entropy'

‘Optimized’ Modeling [Notebook]

- Run both ML algorithms using the identified ‘optimized’ hyperparameter settings.

- Decision Tree
  - Training Accuracy: 0.818
  - Testing Accuracy: 0.757

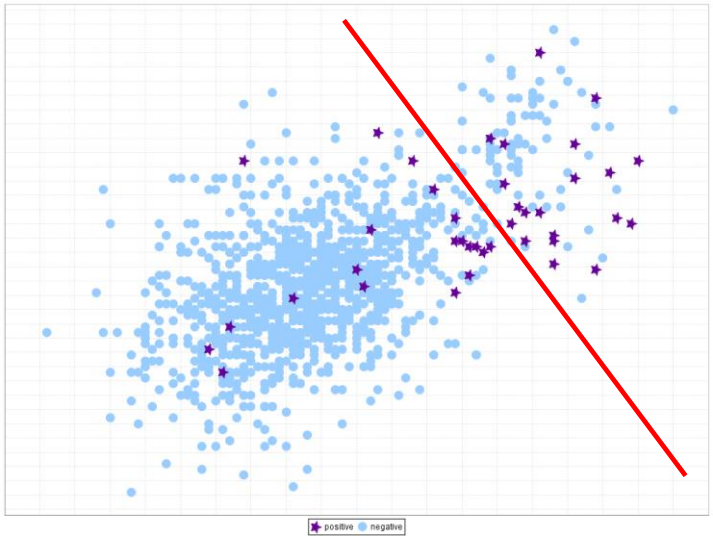
- Random Forest
  - Training Accuracy: 0.864
  - Testing Accuracy: 0.757

Modeling: Evaluation

Evaluation Considerations

- Problem:
  - There are many evaluation/performance metrics available with which to assign value to, and compare your models.
    - Which to utilize?
  - Consider: Using basic accuracy metric for an imbalanced dataset
  - Your problem puts more weight on making accurate predictions for a given class.

$$accuracy = \frac{\sum true\ positive + \sum true\ negative}{\sum total\ population}$$

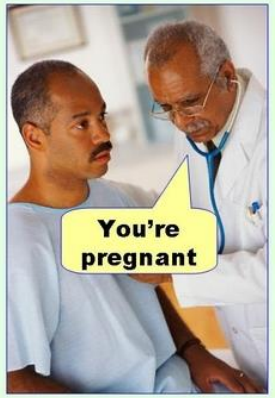


- Solution:
  - Chose evaluation metric(s) that suits…
    - The goals of your problem
    - The nature of your dataset
      - Is the endpoint discrete or continuous?
      - Is the dataset balanced or imbalanced?
      - Is it more important to make accurate predictions on one class vs. another?

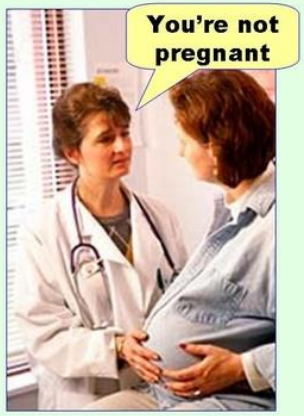
TP, TN, FP, FN

		Actual Class: Positive	Actual Class: Negative
Predicted Class	Positive	**Hit** True Positive TP	**False Alarm** False Positive Type I Error FP
	Negative	**Miss** False Negative Type II Error FN	**Correct Rejection** True Negative TN

Type I error
(false positive)

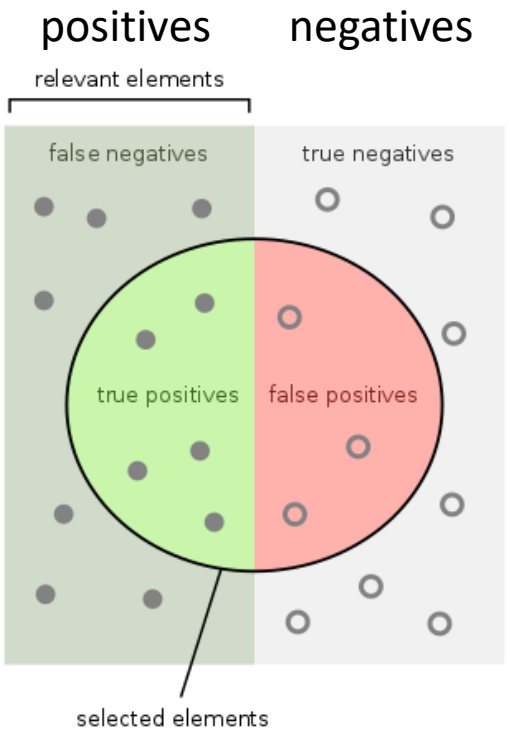


Type II error
(false negative)

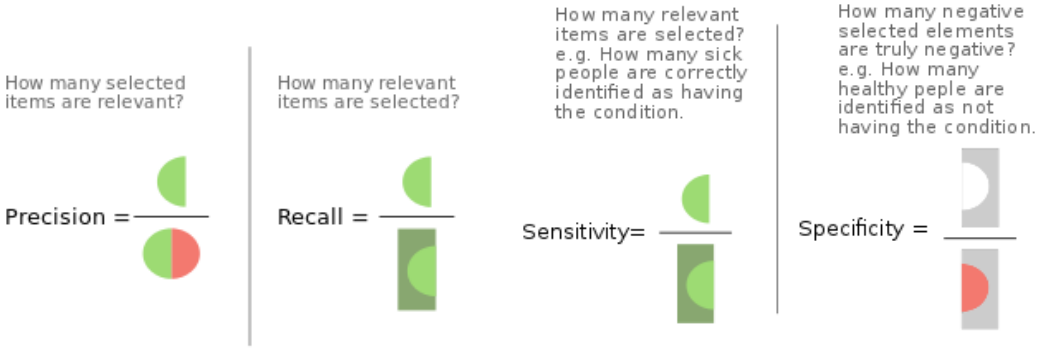


Classification Metrics

- Accuracy = (TP+TN)/(TP+TN+FP+FN)
- Sensitivity (a.k.a Recall, True Positive Rate) = TP/(TP+FN)
- Specificity (a.k.a. True Negative Rate) = TN/(TN+FP)
- Precision (a.k.a. Positive Predictive Value) = TP/(TP+FP)
- False Positive Rate = FP/(FP+TN) → want to minimize.
- F1 Score = 2(Precision*Recall)/(Precision + Recall)
- Balanced Accuracy = (Sensitivity + Specificity)/2



Rectangle = Actual
Circle = Predictions



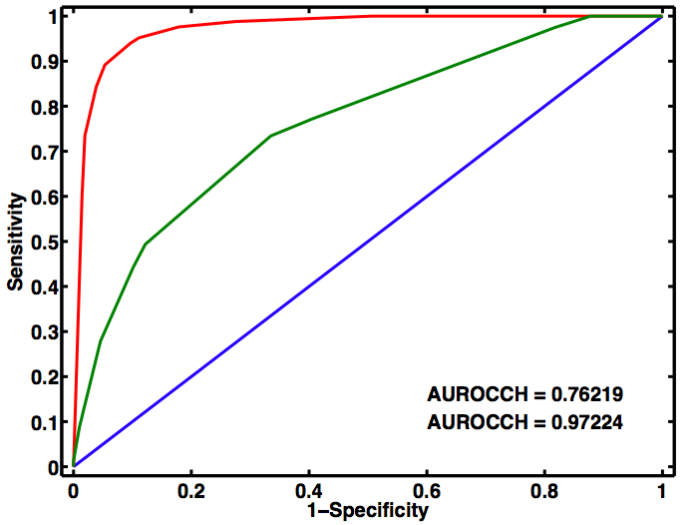
HCC Evaluation of Decision Tree [Notebook]

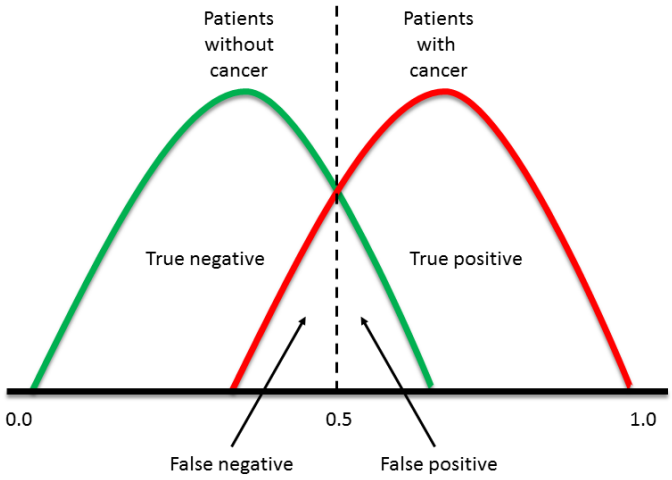
- Classifications:
  - TP = 8
  - FP = 3
  - FN = 5
  - TN = 17

- Accuracy = 0.757
- Sensitivity = 0.76
- Specificity = 0.85
- Precision = 0.75
- False Positive Rate = 0.15
- F1 Score = 0.75
- Balanced Accuracy = 0.7886

ROC Curve and AUC

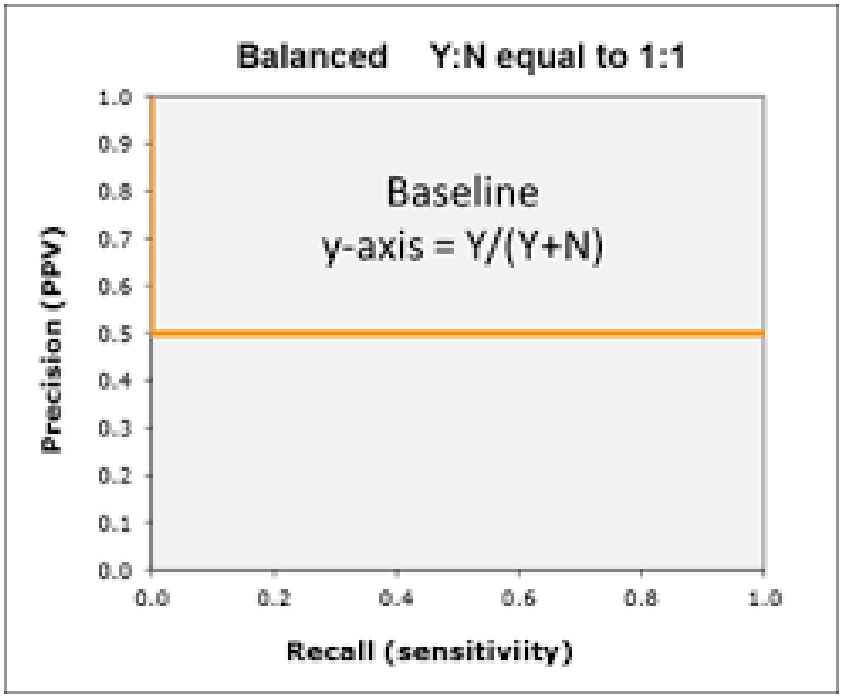
- Receiver Operating Characteristic (ROC):
  - Graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.
  - Axes:
    - True Positive Rate (Sensitivity)
    - False Positive Rate (1-Specificity)
  - Consider threshold extremes:
    - If we never want to make a FN, we always choose positive
    - If we never want to make a FP, we always choose negative
- Area Under the Curve (AUC)
  - Summary statistic for ROC
  - = 0.5 no predictive ability
  - = 1.0 perfect predictive ability



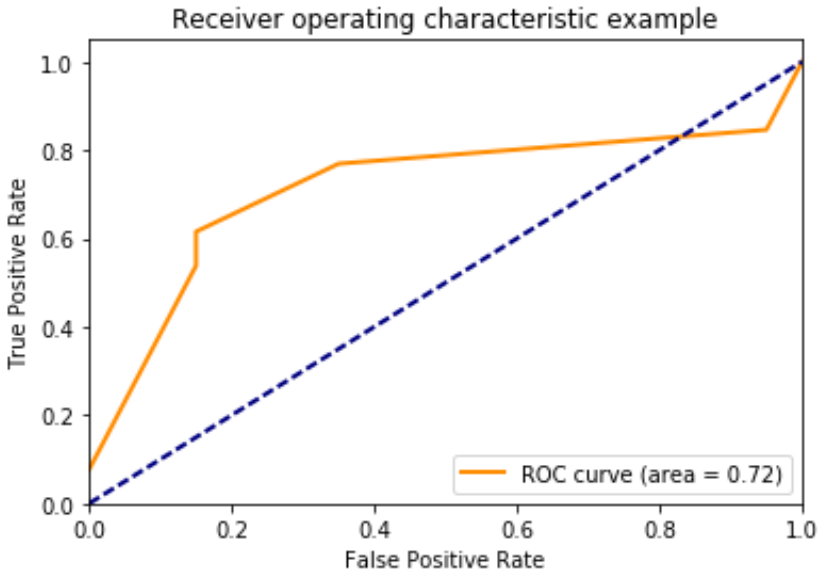


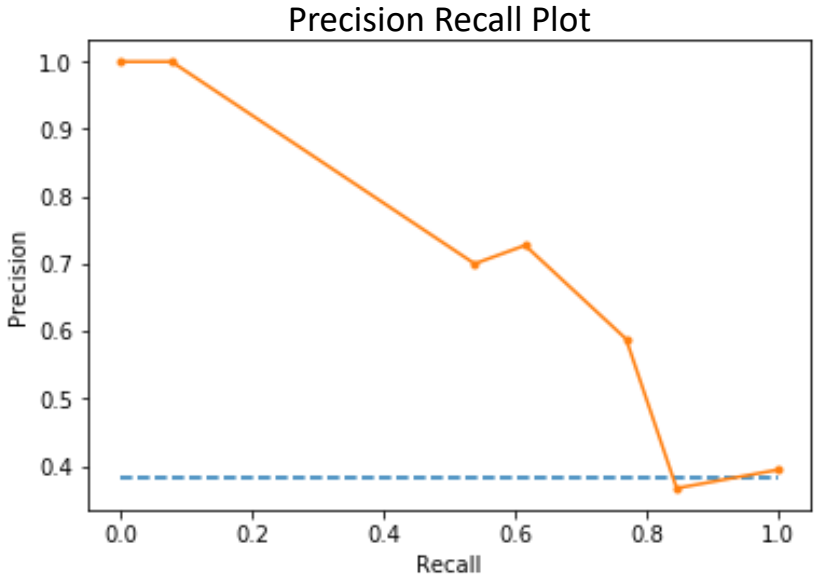
Precision-Recall Plot

- Precision-Recall Plot:
  - An alternative to ROC when there is class imbalance
  - Axes:
    - Precision
    - Recall
- Area Under the Curve (AUC)
  - Can set the 'no-skill-line' based on class imbalance ratio



ROC & Precision/Recall Plots [Notebook]





Our class imbalance ratio is 1:1.61 which translates to a no skill line of about 0.383

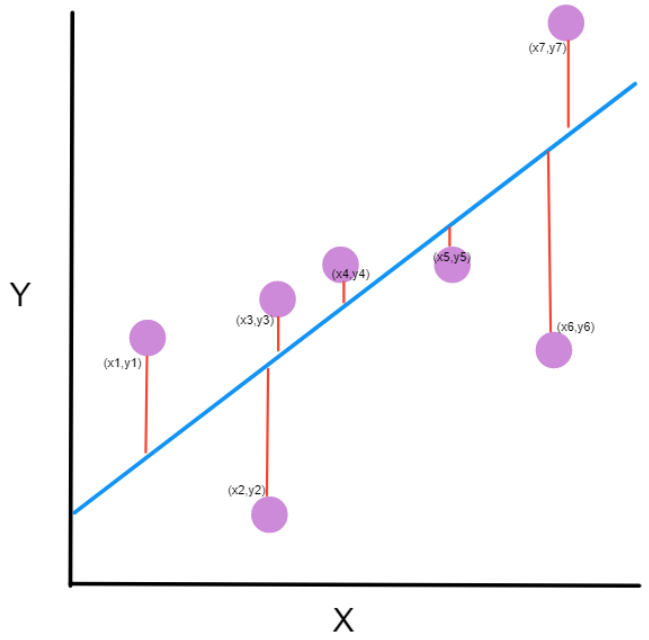
AUC (ROC): 0.72

AUC (P/R): 0.721

Regression Metrics

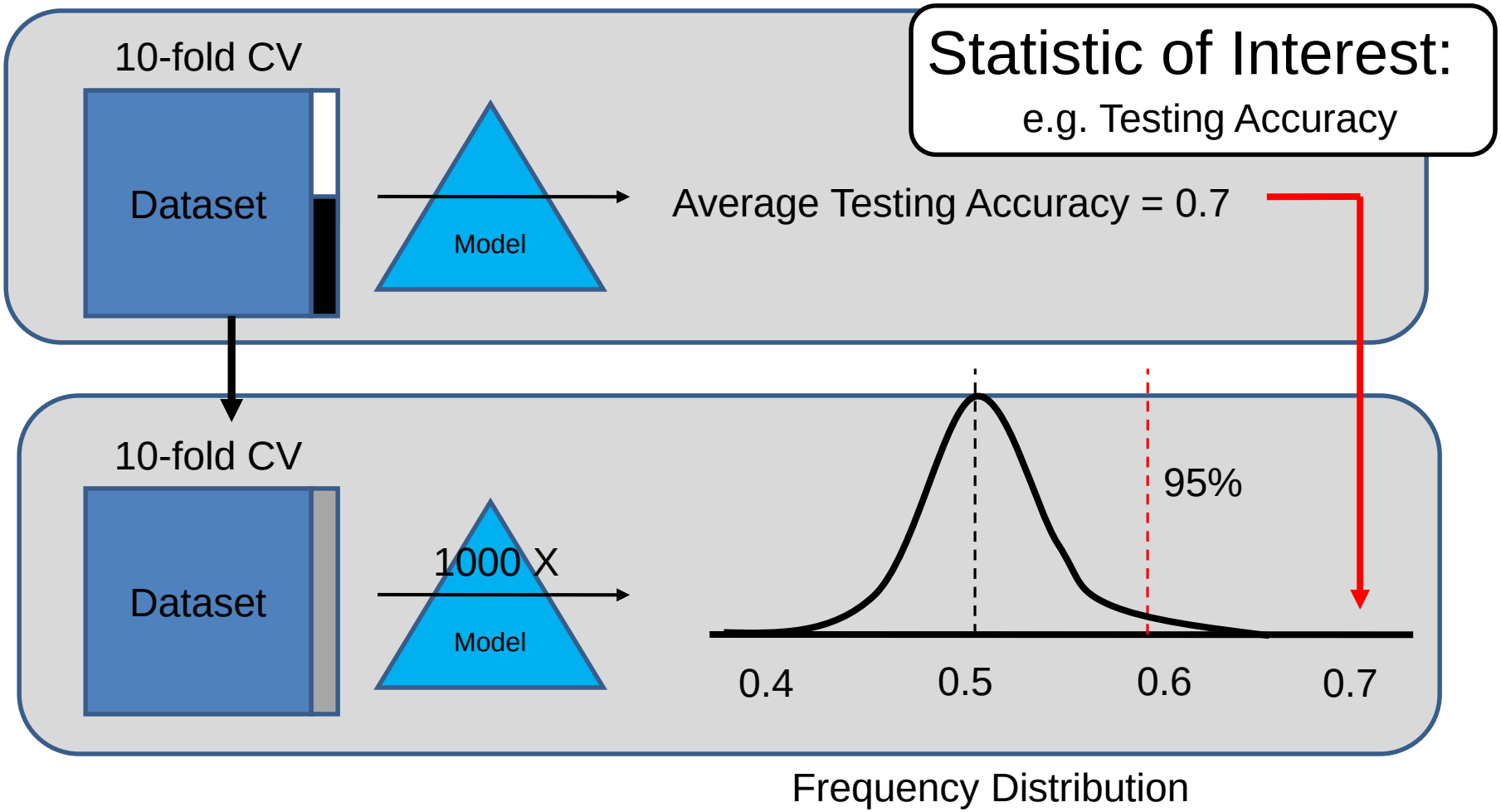
$$MeanAbsoluteError = \frac{1}{N}\sum_{j=1}^{N}|y_j - \hat{y}_j|$$

$$MeanSquaredError = \frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2$$



Permutation Testing

- How do we assess whether our evaluation statistics are likely to be meaningful? (i.e. better than expected by random chance)
- Permutation-Based Significance Testing (obtain p-values for each statistic/metric of interest).



Post-Analysis: Interpretation

Avenues of Interpretation

- Not all ML modeling strategies yield an 'interpretable' solution.
  - Representation
  - Model size/complexity
  - Ensemble vs. single model

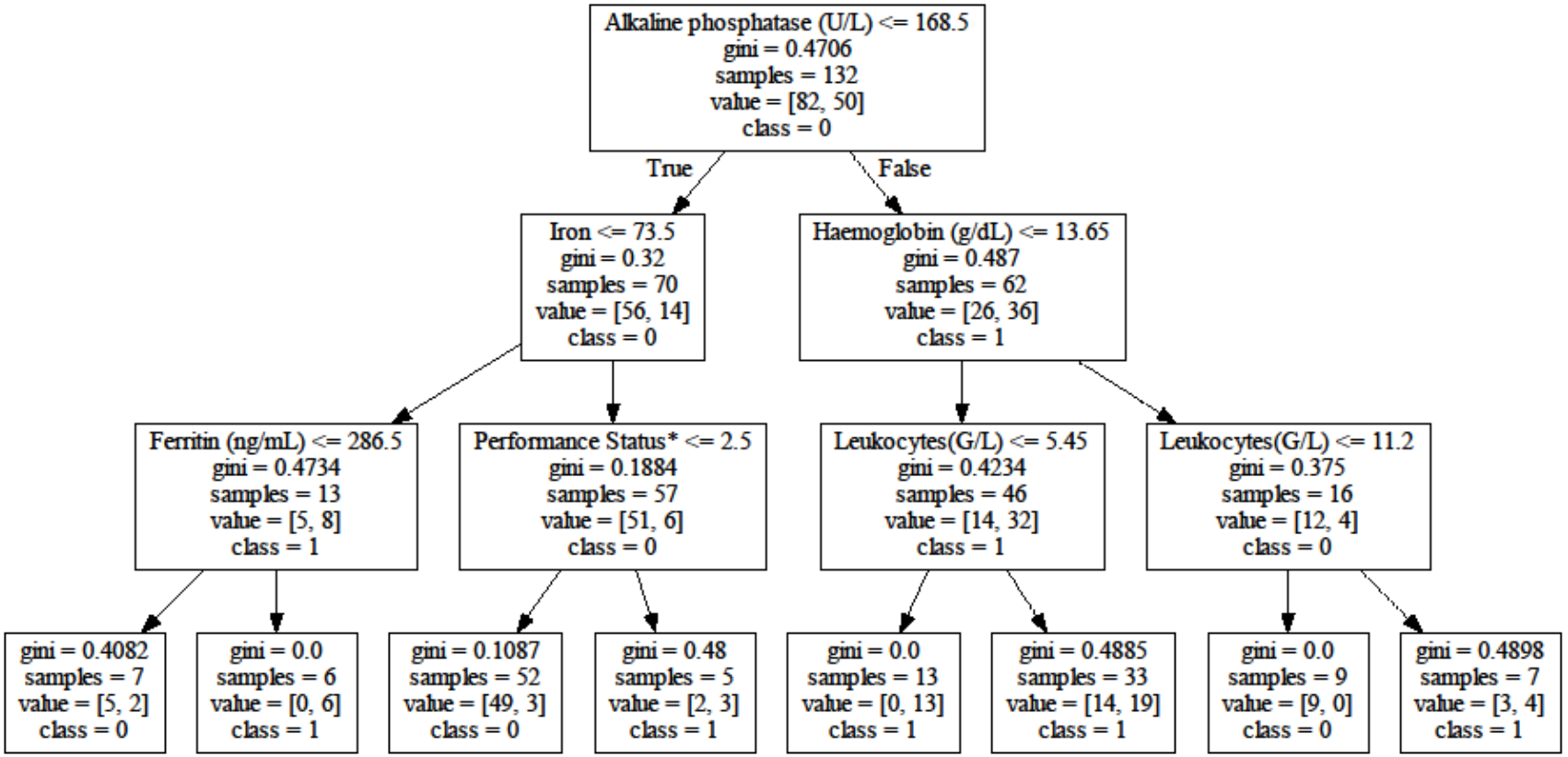
- Interpretability is largely a subjective quality.

- Modes of interpretability:
  - Identifying relevant features, quantifying feature importance
  - Can the relationship between features in making predictions be characterized?
    - Additive Multivariate
    - Feature interactions
    - Heterogeneous associations
  - Can prediction decisions be described in human comprehensible terms?
  - Can the model be described as a set of conditional rule expressions?

- Different ML modeling approaches can have different opportunities for interpretation.

Decision Tree Interpretation [Notebook]

- Visualization of our 'best' trained decision tree.



Rules from Decision Trees [Notebook]

Decision Tree as nested if/else expression

```
if Alkaline phosphatase (U/L) <= 168.5:
  if Iron <= 73.5:
    if Ferritin (ng/mL) <= 286.5:
      return [[5. 2.]]
    else:  # if Ferritin (ng/mL) > 286.5
      return [[0. 6.]]
  else:  # if Iron > 73.5
    if Performance Status* <= 2.5:
      return [[49.  3.]]
    else:  # if Performance Status* > 2.5
      return [[2. 3.]]
else:  # if Alkaline phosphatase (U/L) > 168.5
  if Haemoglobin (g/dL) <= 13.649999618530273:
    if Leukocytes(G/L) <= 5.449999809265137:
      return [[ 0. 13.]]
    else:  # if Leukocytes(G/L) > 5.449999809265137
      return [[14. 19.]]
  else:  # if Haemoglobin (g/dL) > 13.649999618530273
    if Leukocytes(G/L) <= 11.199999809265137:
      return [[9. 0.]]
    else:  # if Leukocytes(G/L) > 11.199999809265137
      return [[3. 4.]]
```

Example Derived Rule

if Alkaline phosphatase (U/L) <= 168.5:

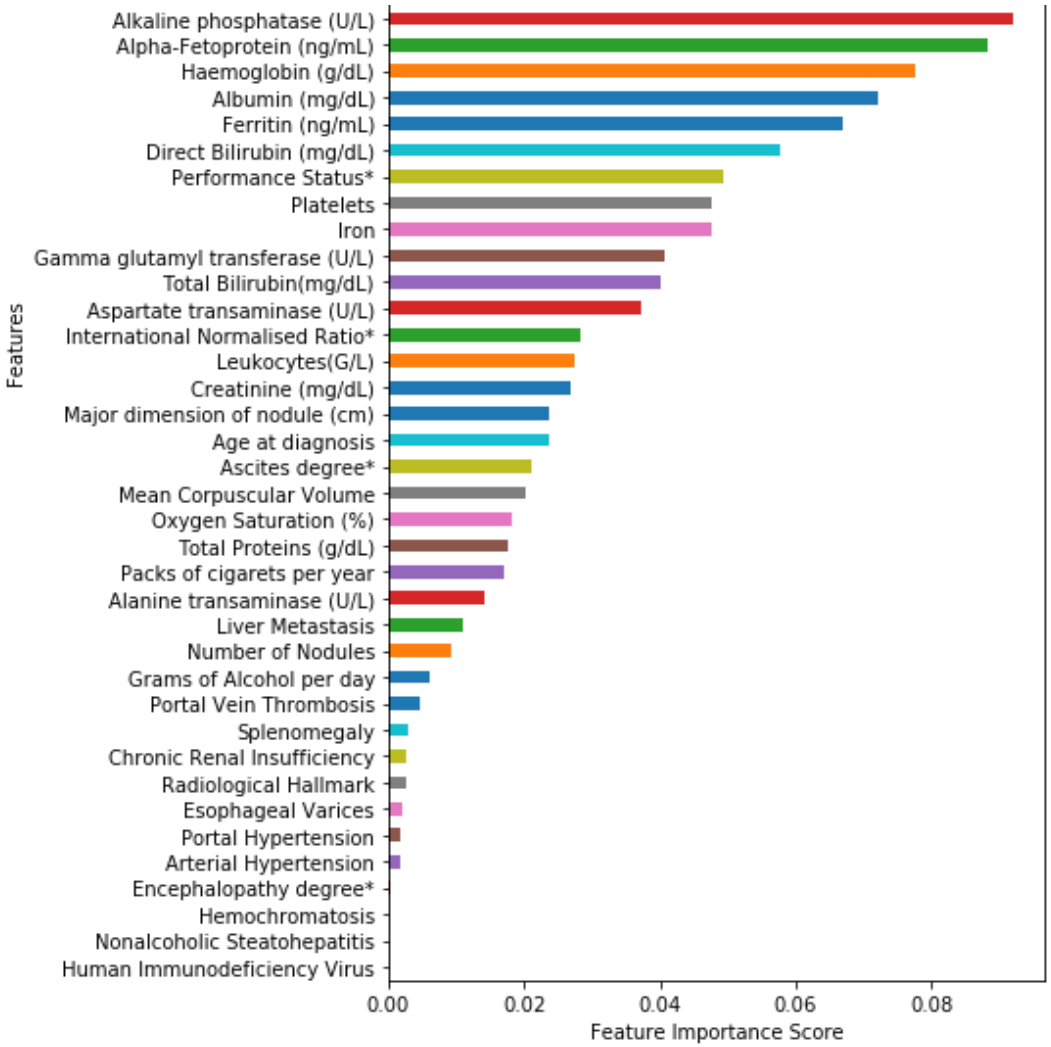
AND if Iron <= 73.5:

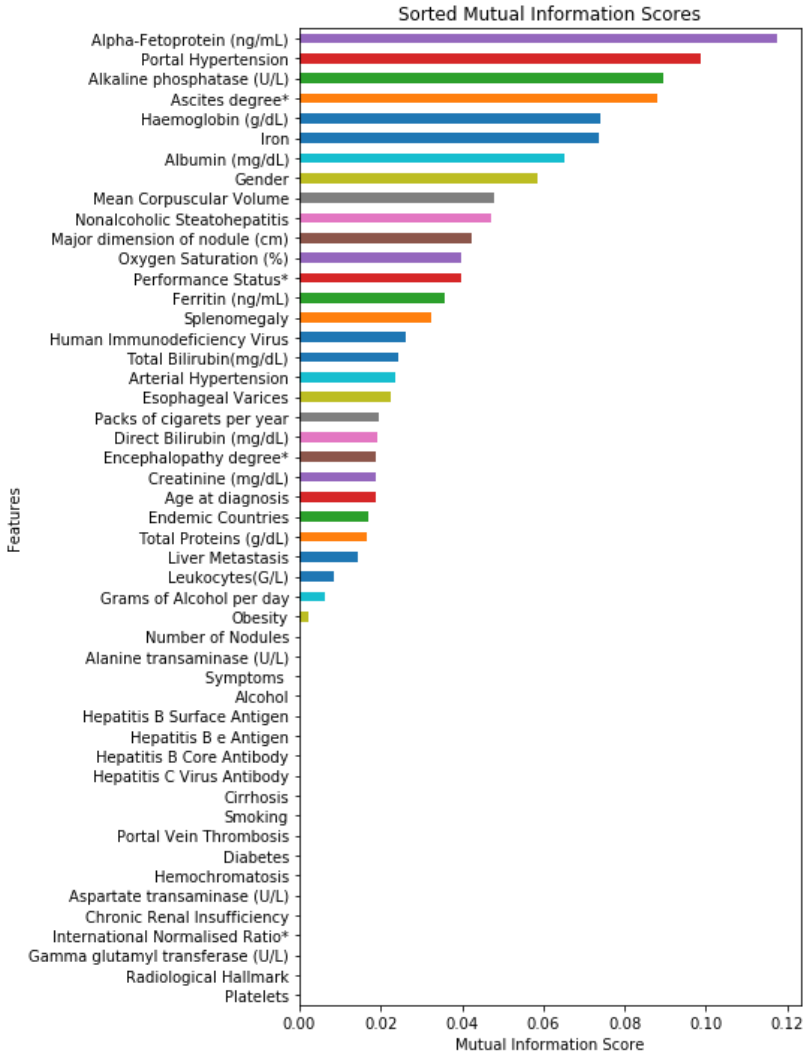
AND if Ferritin (ng/mL) <= 286.5:

THEN: [[5. 2.]] → Predict class 0

RF Feature Importance [Notebook]

Random Forest Feature Importance Scores



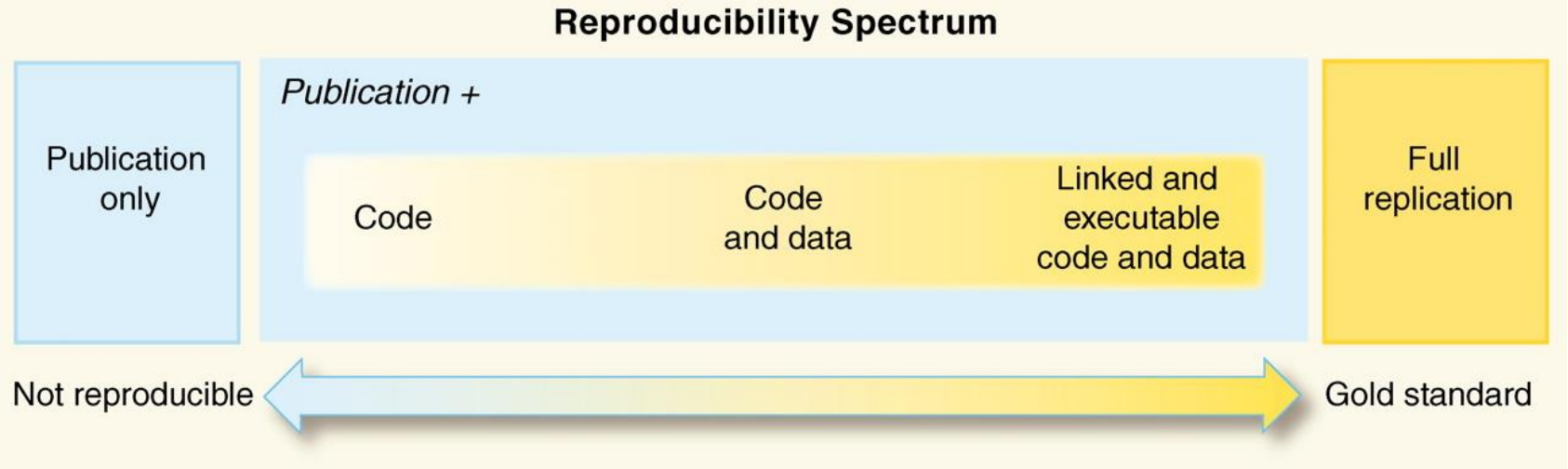


Post-Analysis: Replication

Replication and Reproducibility

- Replicate same or similar findings in an independently sampled dataset.

- Reproducibility of Code, analysis pipeline.



Knowledge

What 'Knowledge' Might Be Obtained?

- Newly generated hypotheses:
  - Novel candidate risk factors, biomarkers, causal candidates.
  - Characterized patterns of association
    - Feature interactions
    - Heterogeneous associations
  - Novel subject subgroups (Future targets for personalized medicine)

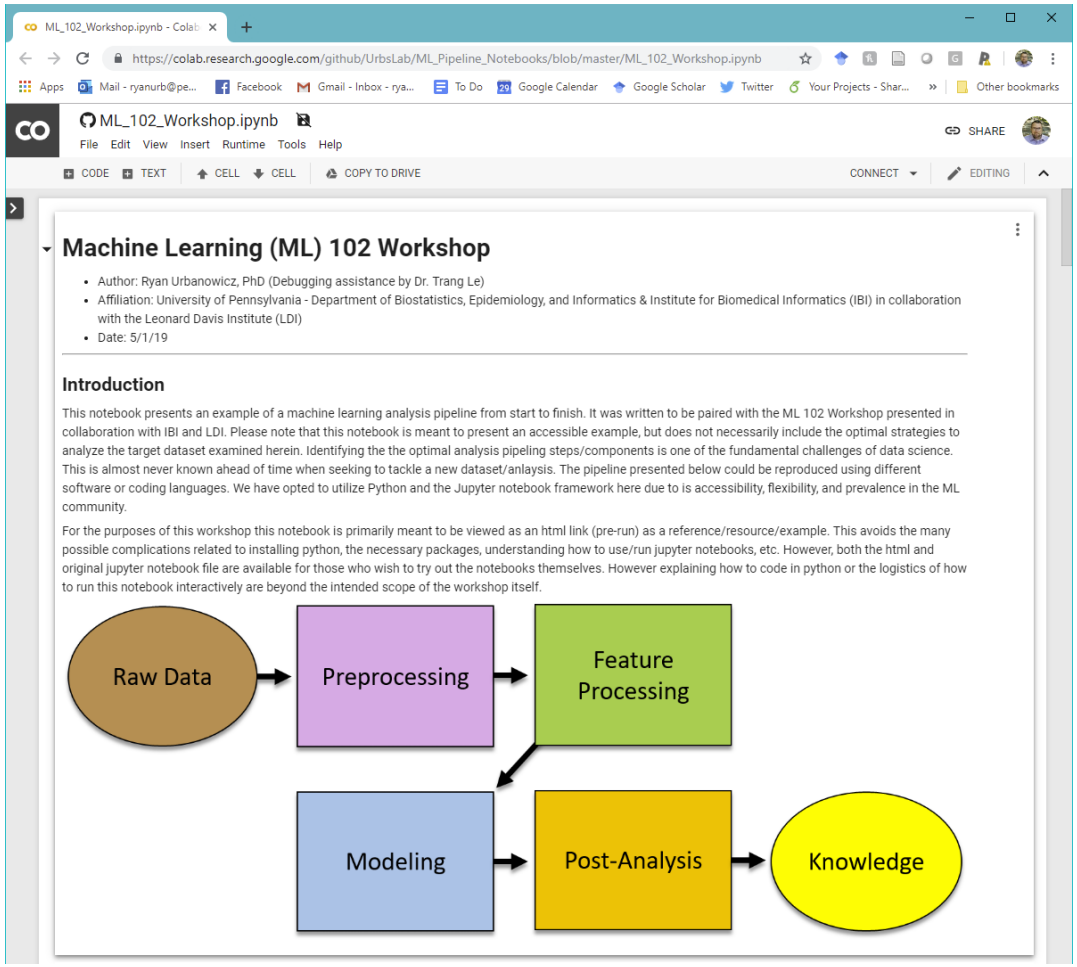
- An 'optimally' accurate prediction model:
  - Compare to other standards in the field
  - Consider for deployment

What's next?

- Many alternative approaches to every element described in this pipeline
  - Each with own trade-offs
- Machine learning is not only about applying existing methods but advancing new/better ones and strategies to combine them as part of an analysis pipeline.
- Important to keep thinking about the intersection of statistics and machine learning → how can they best reinforce each other?
- Lots more to learn, explore, and invent …

[Jupyter Notebook]

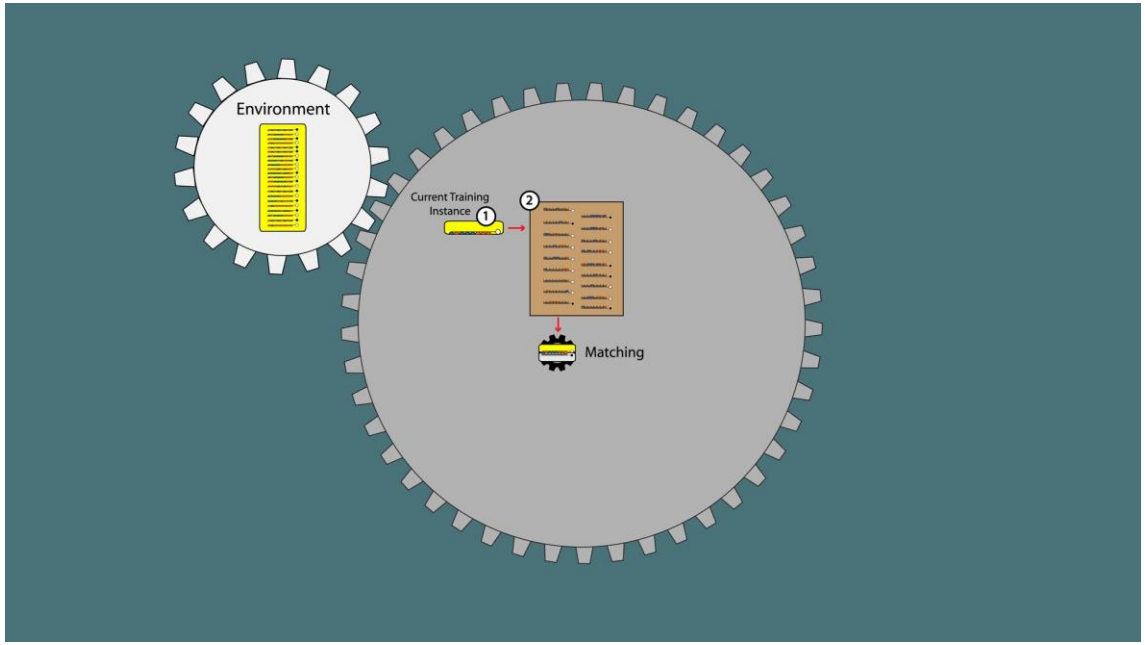
- https://github.com/UrbsLab/ML_Pipeline_Notebooks

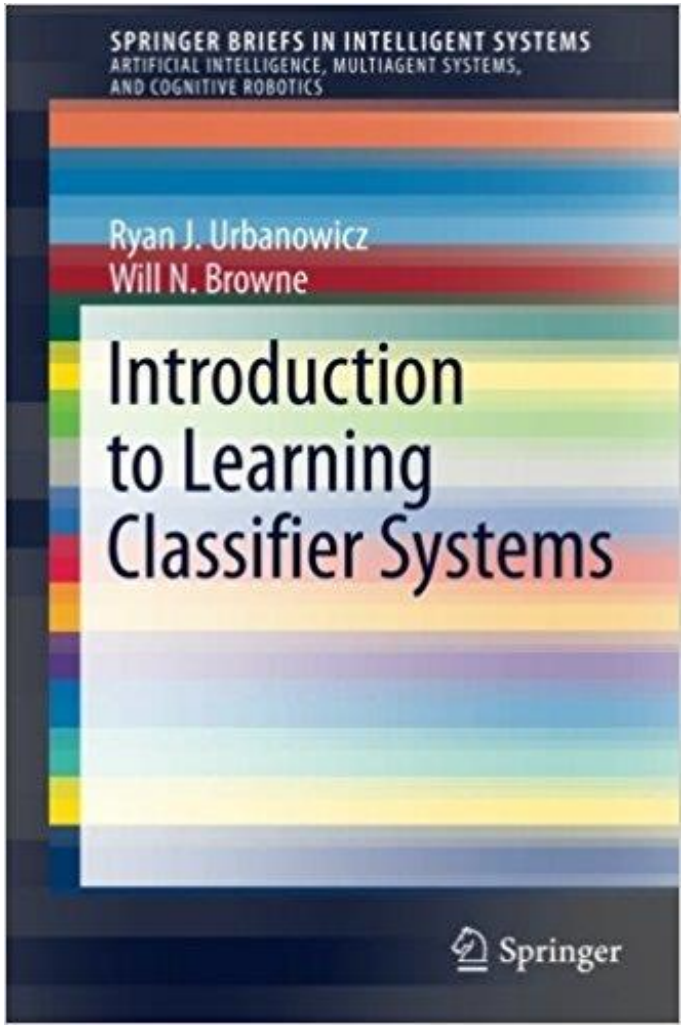


Advanced ML – A Shameless Plug

- YouTube Video on LCS:

https://www.youtube.com/watch?v=CRge_cZ2cJc

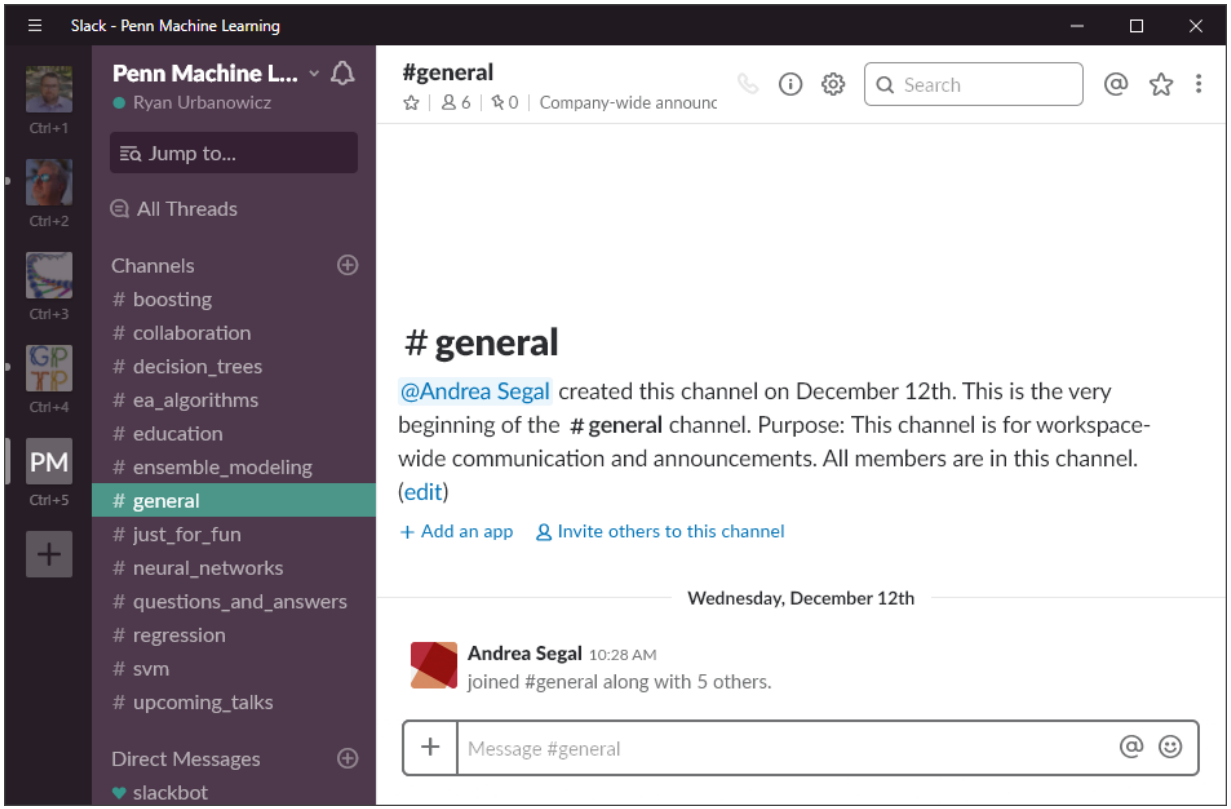




- Textbook: Introduction to Learning Classifier Systems [Urbanowicz & Brown, 2017]. Springer. 2017. (Available on Amazon)

UPenn ML slack

- Penn Machine Learning – Slack Workspace
- pennmachinelearning.slack.com



Acknowledgements and Funding

- Pennsylvania Commonwealth Universal Research Enhancement Program (CURE)