+

3099704: AI for Digital Health

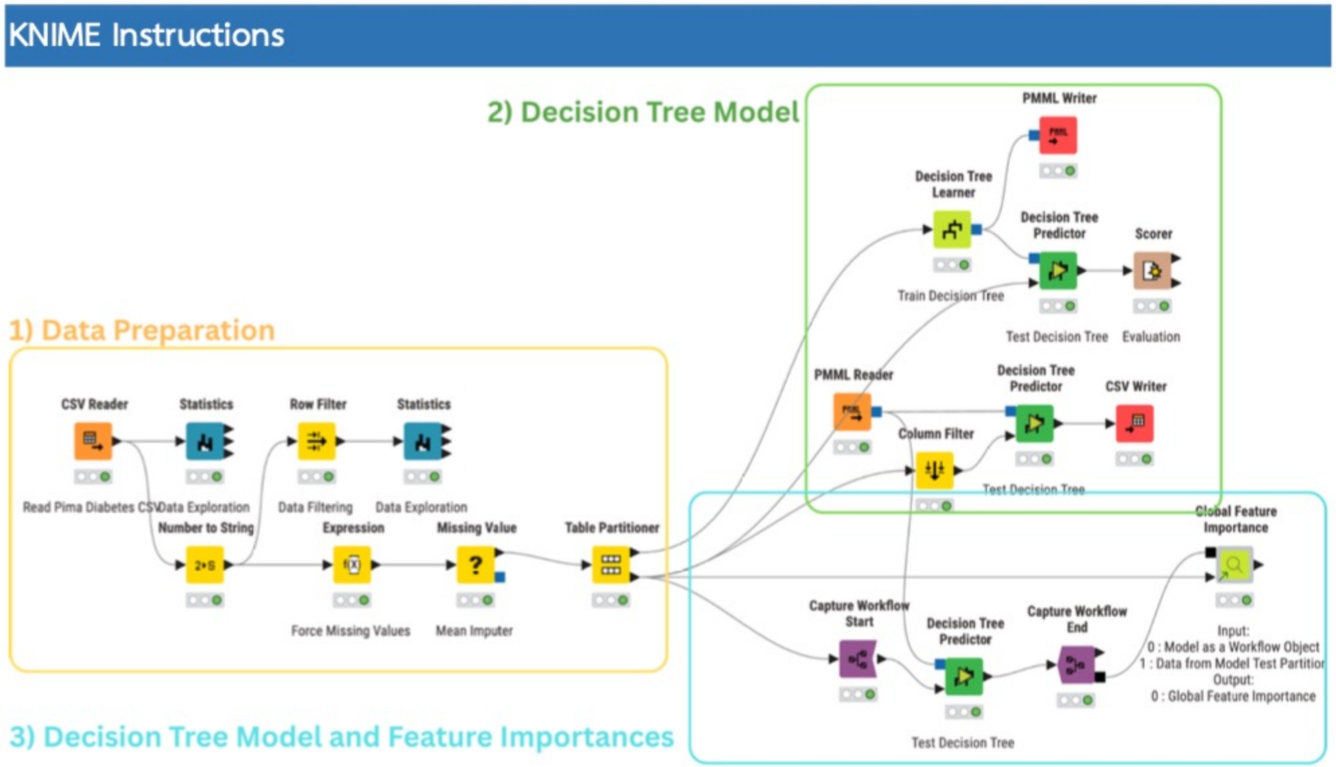
Classification with Tree-Based Models

Prof. Peerapon Vateekul, Ph.D.

Peerapon.v@chula.ac.th

Outlines

- Supervised Learning
- Decision Tree
- Random Forest
- Classification Performance

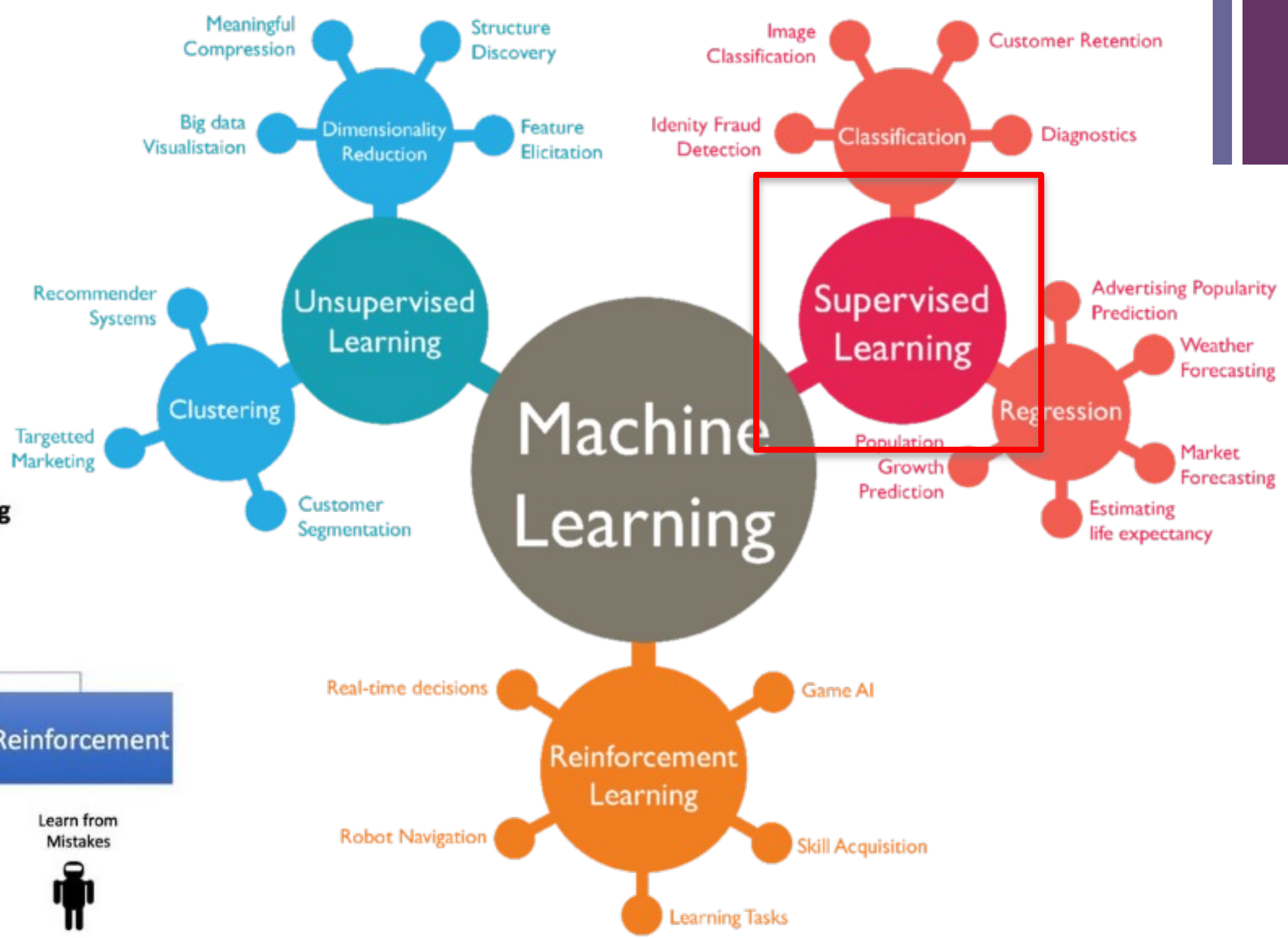


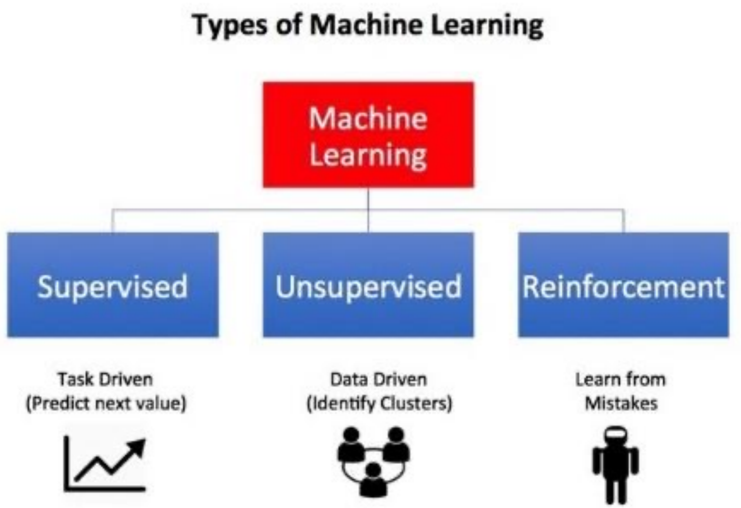
CHULA ΣNGINEERING
Foundation toward Innovation
COMPUTER

+

Supervised Learning (Predictive Task)

+ Machine Learning



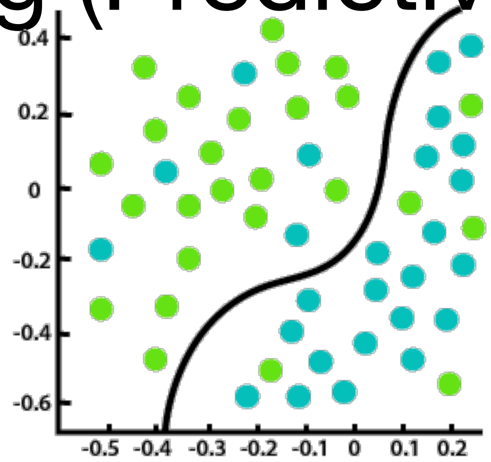


Supervised Learning (Predictive Task)

inputs				target
Age	**Temp**	**Gender**	**Smell**	**Covid**
25	39.0	Female	No	Yes
35	38.9	Female	No	Yes
32	36.5	Male	Yes	No

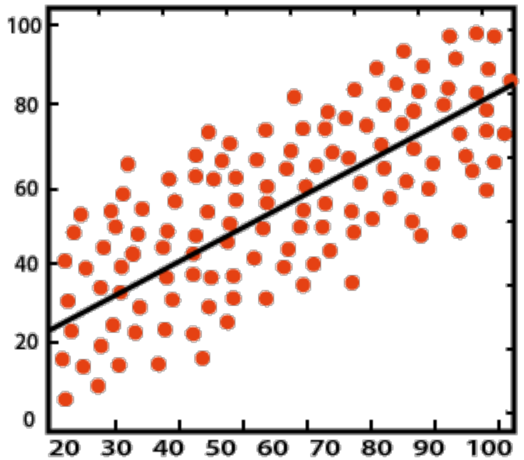
- Goal: To learn a prediction model mapping from inputs to output.

- Data without label (answer) is meaningless!
- Label should be provided by experts!



Classification

- Target is categorical variable.
- <u>Example</u>
- Covid diagnosis (yes/no)
- Disease diagnosis from gait information:
  - 1) Normal,
  - 2) Sick/Knee OA
  - 3) Sick/Parkinson



Regression

- Target is numeric variable.
- <u>Example</u>
- PD's state diagnosis from movement data.
- Glucose level prediction from breath particles.

There are two main processes: Train/Test

1) Training Phase: Model Construction

Training Data

inputs

target

Age	Income	Gender	Province	Purchase
25	25,000	Female	Bangkok	Yes
35	50,000	Female	Nontaburi	Yes
32	35,000	Male	Bangkok	No

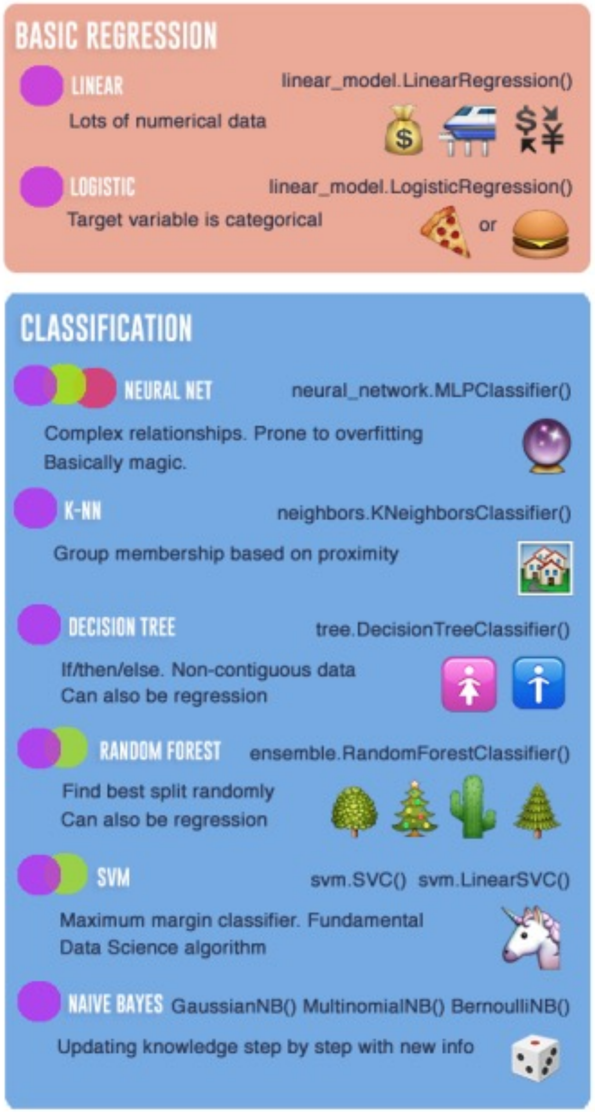
2) Testing Phase: Model Evaluation, Model Assessment
Also called "prediction, inference, scoring"

Testing Data

Age	Income	Gender	Province	Purchase
25	25,000	Female	Bangkok	?

+ Prediction Algorithms

- Decision Tree
- (Logistic) Regression
- Neural Networks (NN)
- kNN
- Support Vector Machine
- Deep Learning



Scikit-learn: Machine learning library in Python

- Provides many machine learning tools with a common Estimator interface
- Built in helpers for common ML tasks (e.g., metrics, preprocessing)
- Easily combine algorithms to make a complex pipeline
- Relies heavily on numpy and scipy, often used with pandas

How do you pronounce the project name?

sy-kit learn. sci stands for science!

Why scikit?

There are multiple scikits, which are scientific toolboxes built around SciPy. You can find a list at https://scikits.appspot.com/scikits. Apart from scikit-learn, another popular one is scikit-image.

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction. — Examples

http://scikit-learn.org/stable/index.html

CHULA ΣNGINEERING COMPUTER
Foundation toward Innovation

Estimator Interface

Decision Trees

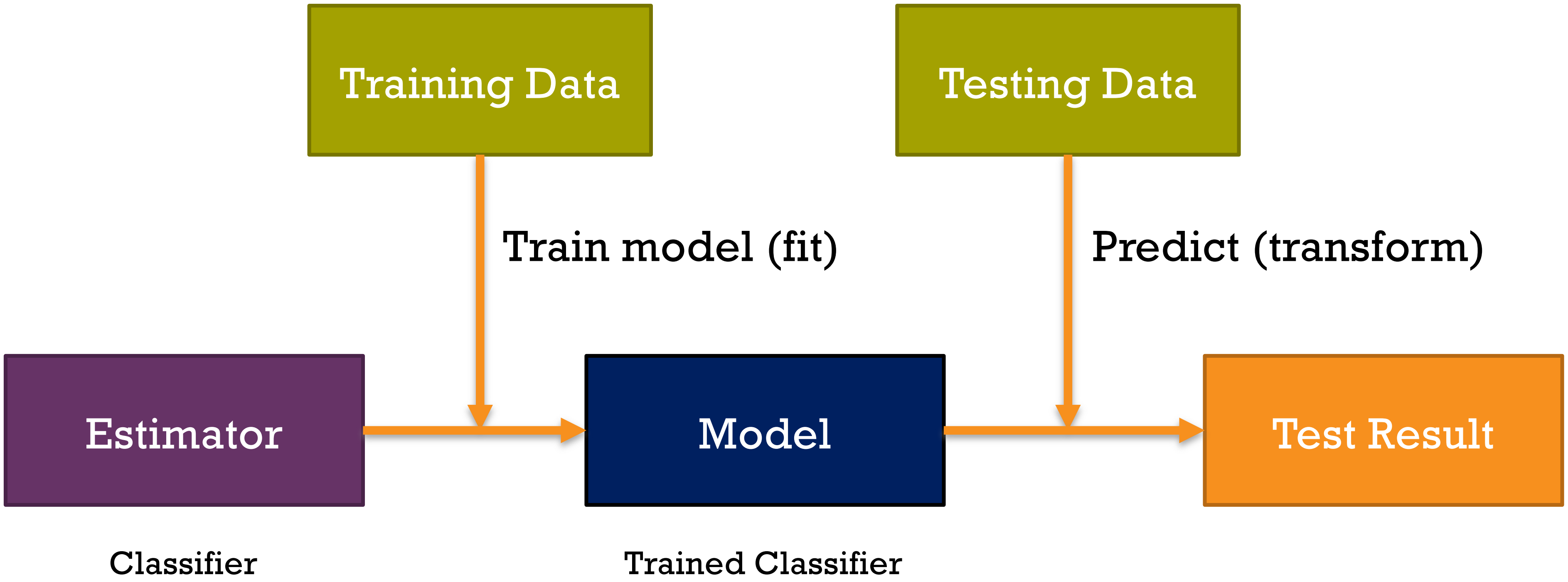
We'll start just by training a single decision tree.

```
In [8]: from sklearn.tree import DecisionTreeClassifier

In [9]: dtree = DecisionTreeClassifier(min_samples_leaf=10, criterion='entropy')

In [10]: dtree.fit(X_train,y_train)

Out[10]: DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=None,
                    max_features=None, max_leaf_nodes=None,
                    min_impurity_decrease=0.0, min_impurity_split=None,
                    min_samples_leaf=10, min_samples_split=2,
                    min_weight_fraction_leaf=0.0, presort=False, random_state=None,
                    splitter='best')
```

Prediction and Evaluation

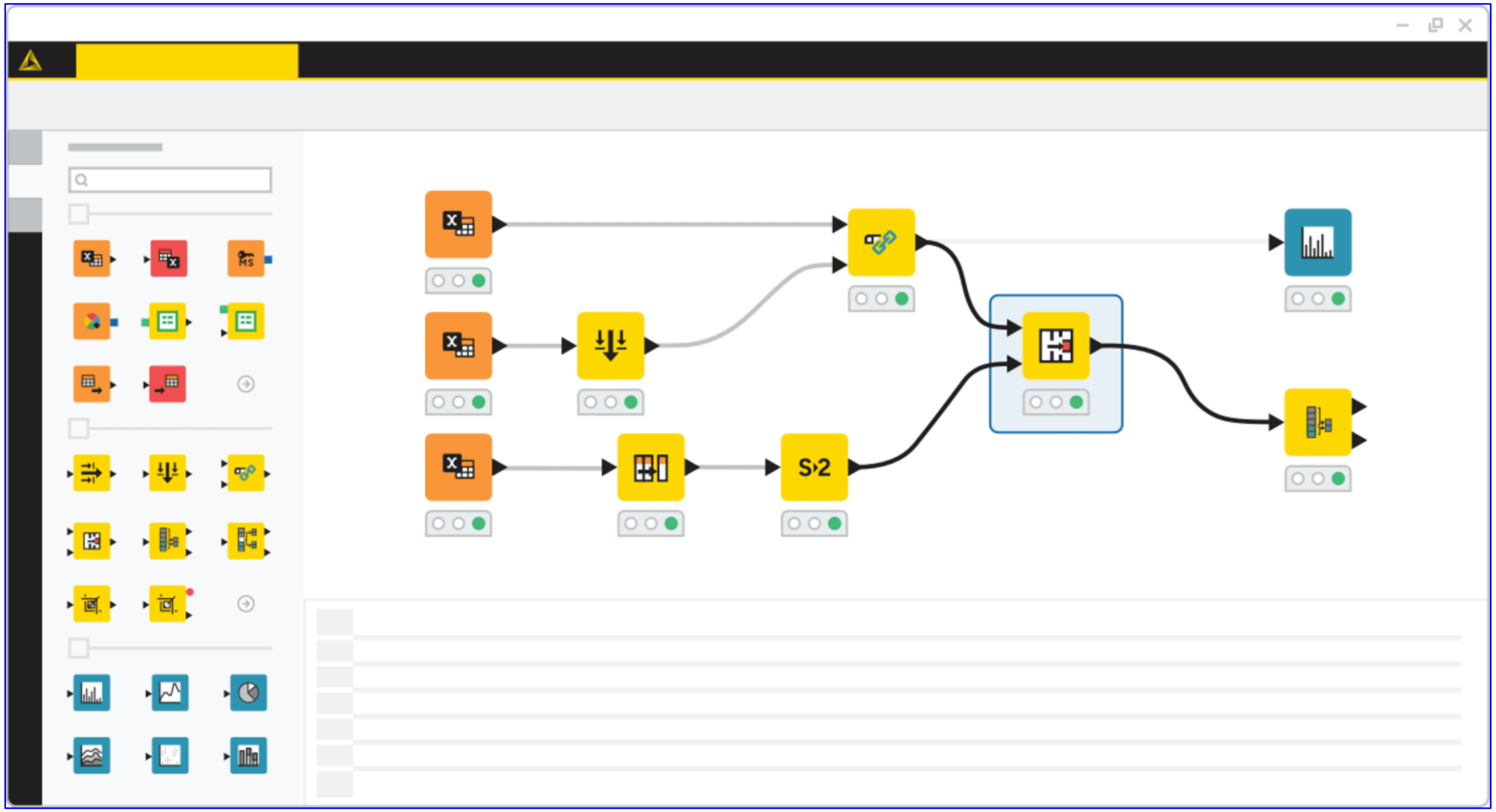
Let's evaluate our decision tree.

```
[11]: predictions = dtree.predict(X_test)

[12]: from sklearn.metrics import classification_report,confusion_matrix

[13]: print(classification_report(y_test,predictions))

             precision    recall  f1-score   support

     absent       0.85      0.85      0.85        20
    present       0.40      0.40      0.40         5

avg / total       0.76      0.76      0.76        25
```



Low-Code/No-Code Software

Open for Innovation KNIME

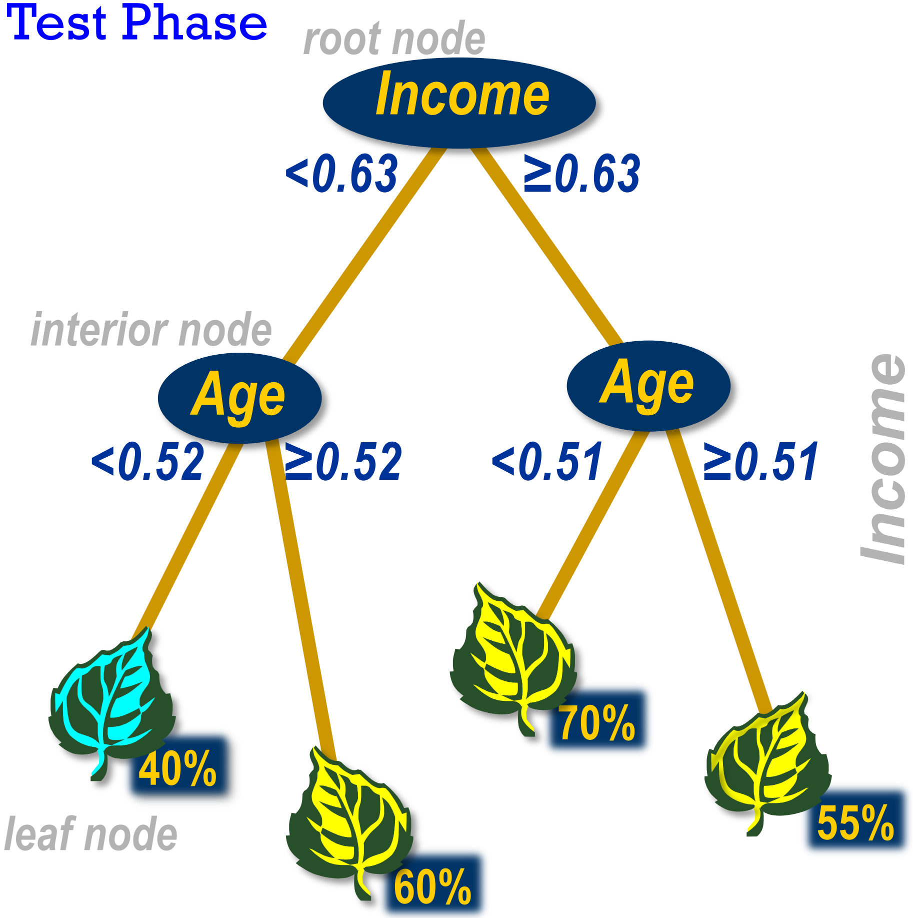


CHULA ΣNGINEERING
Foundation toward Innovation
COMPUTER

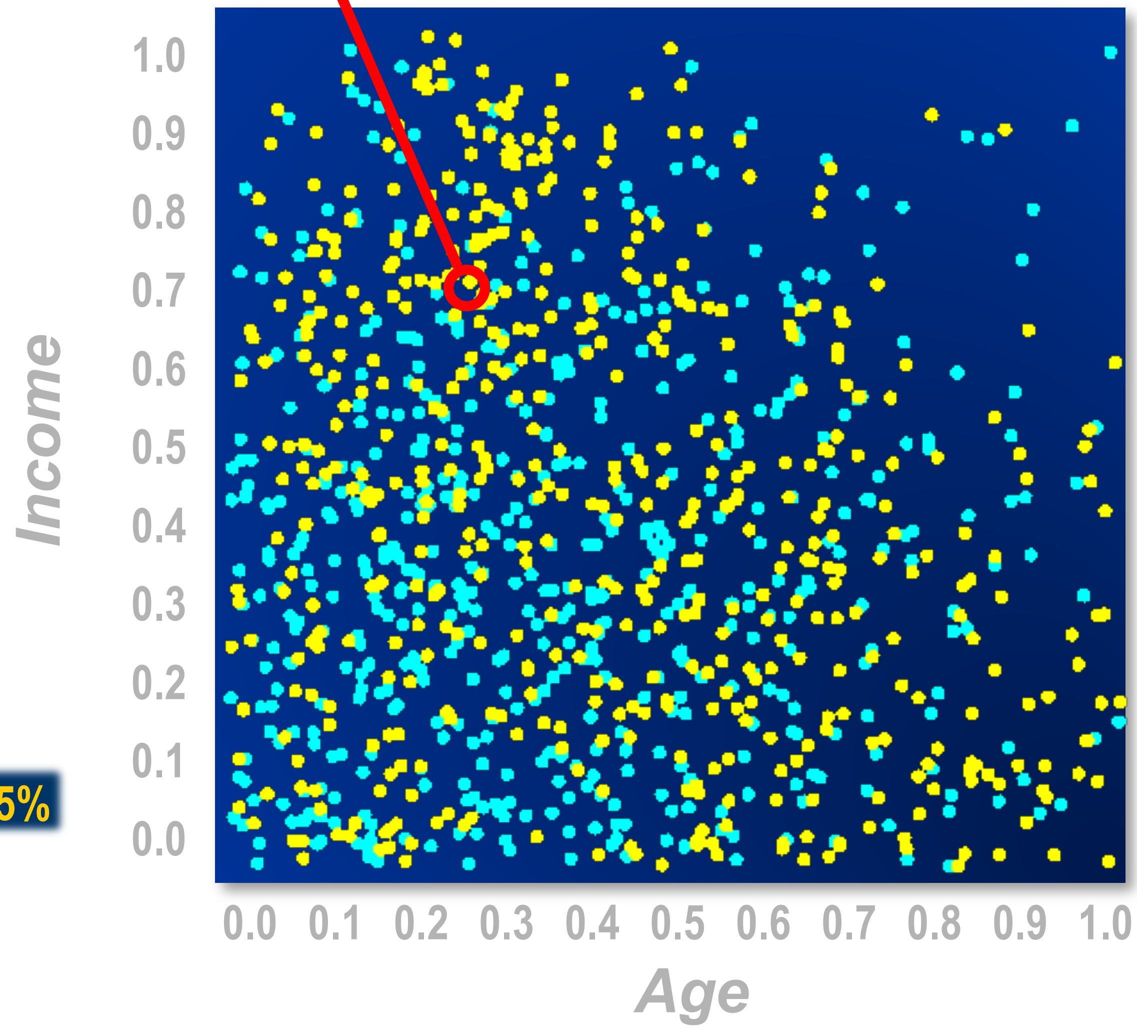
Decision Tree

1) Decision Tree Prediction Rules

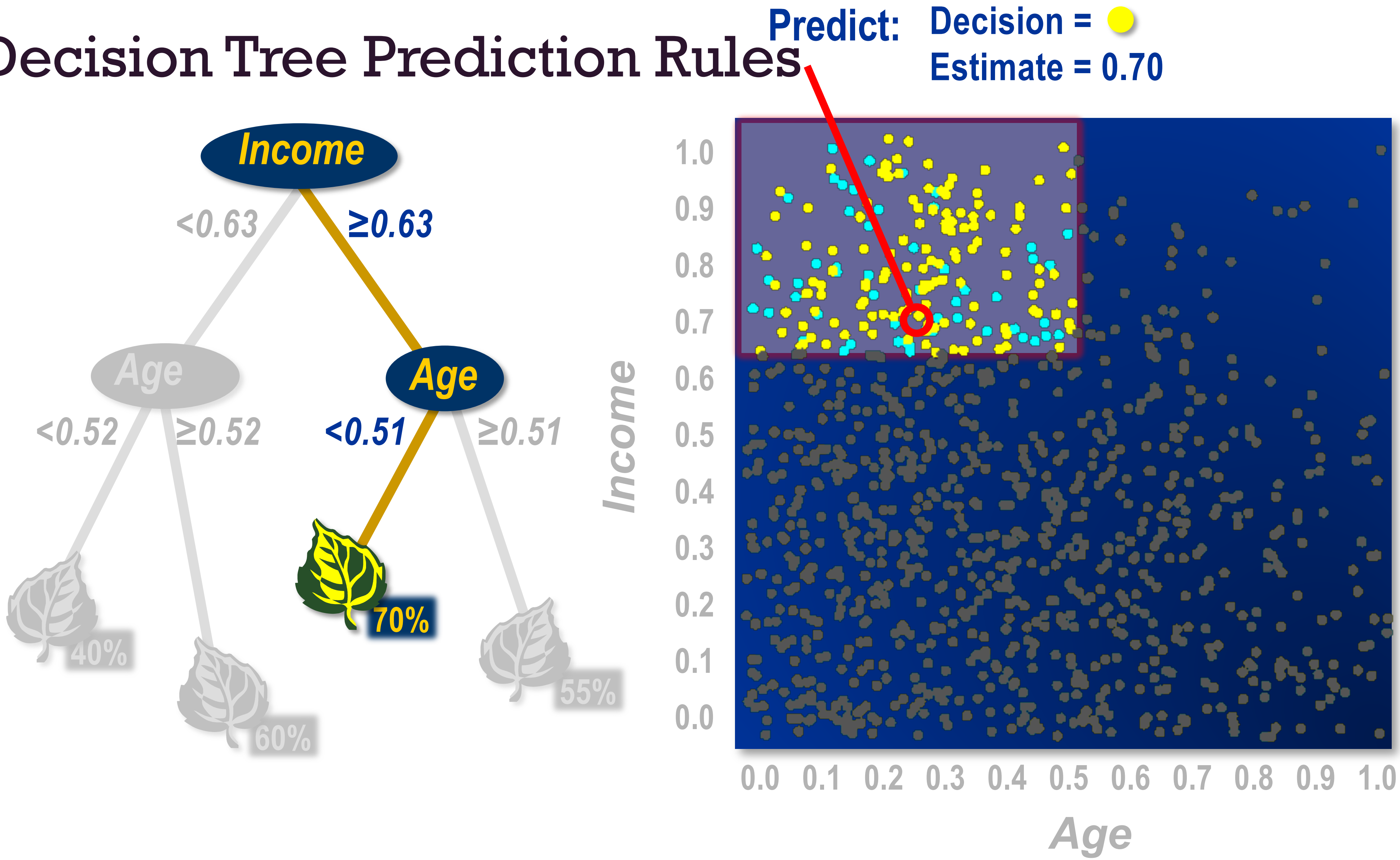
Test Phase



Predict:



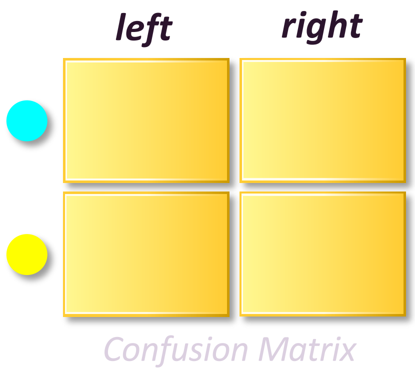
Decision Tree Prediction Rules

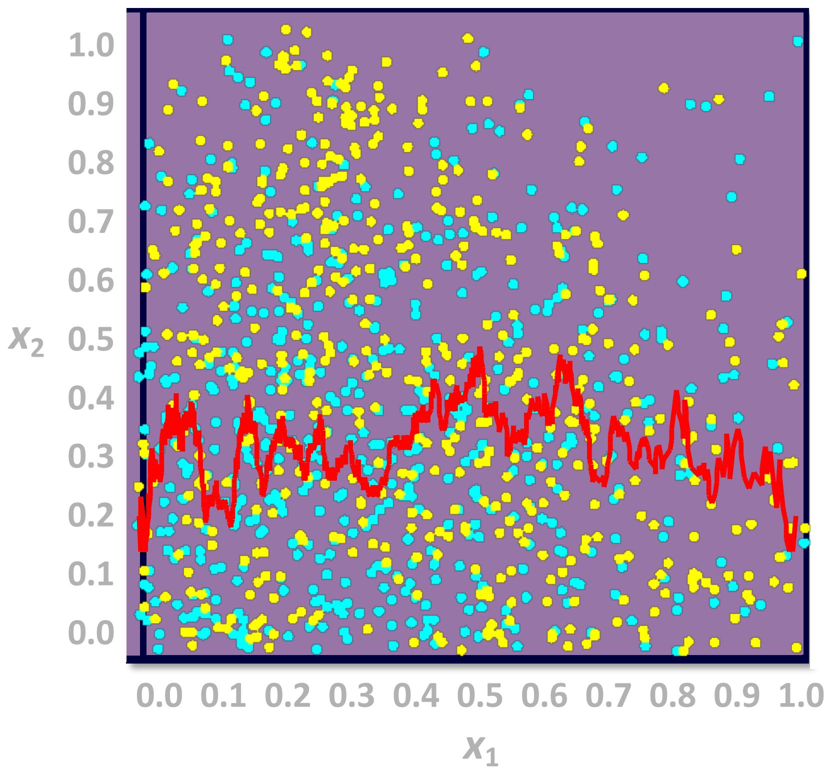


Model Essentials: Decision Trees

- Predict cases. — **Prediction rules**
- Select useful predictors. — **Split search**

Decision Tree Split Search

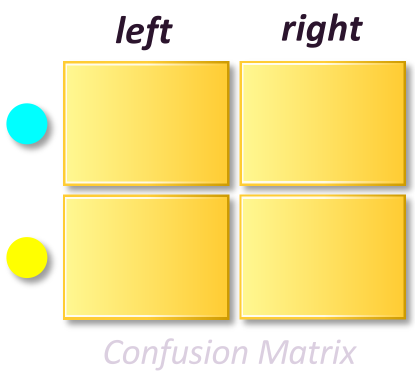




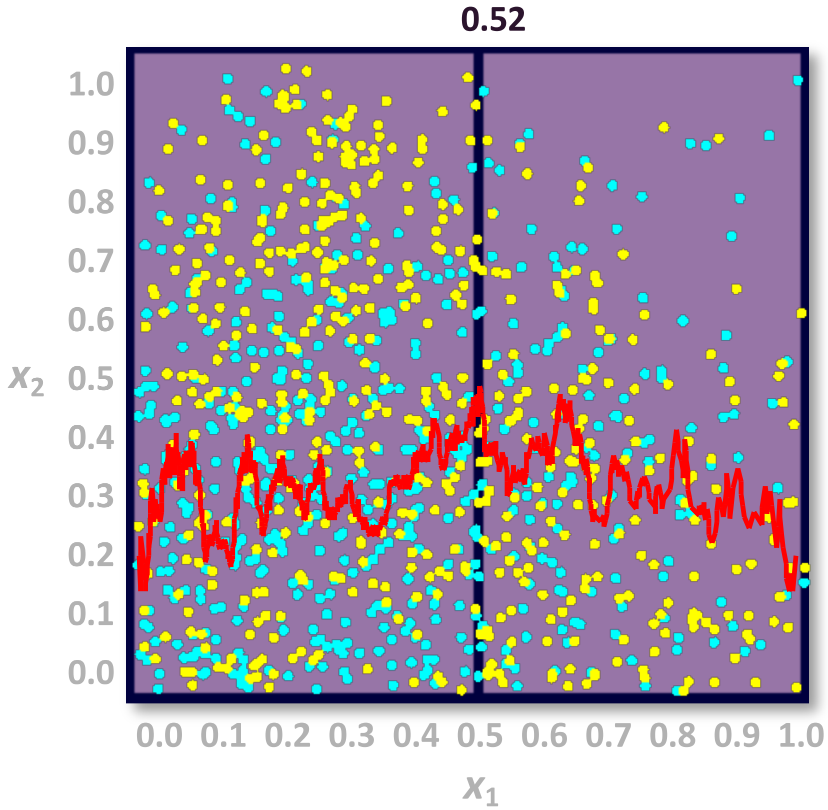
Calculate information gain on partitions on input x_1.

...

Decision Tree Split Search



Calculate the gain of every partition on input x_1.



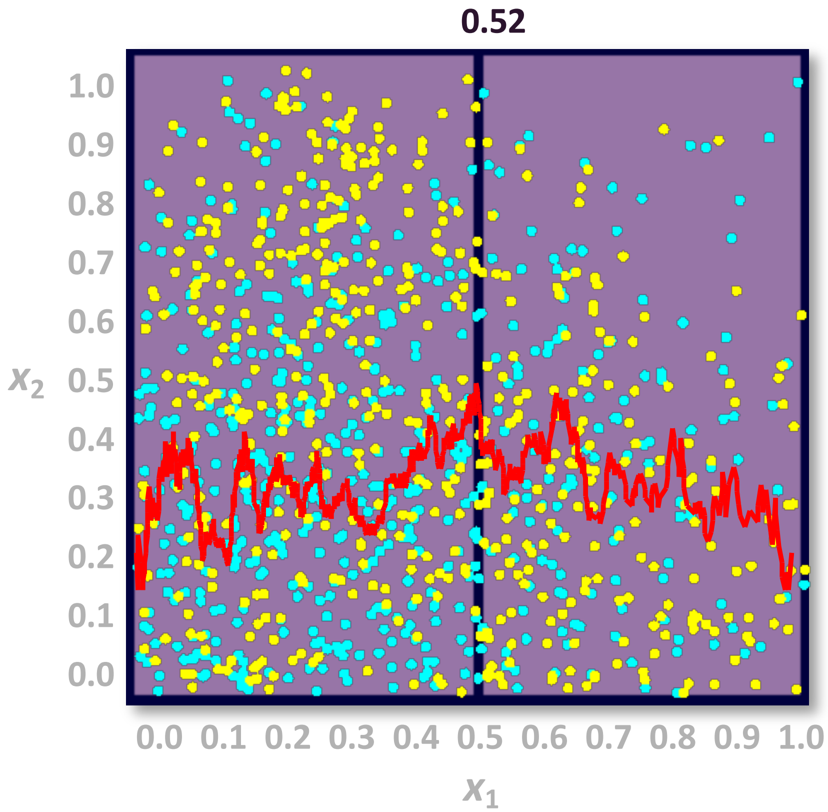
...

Decision Tree Split Search

	left	right
	53%	42%
	47%	58%

max
gain(x_1)
0.0112

Select the partition with the maximum gain.



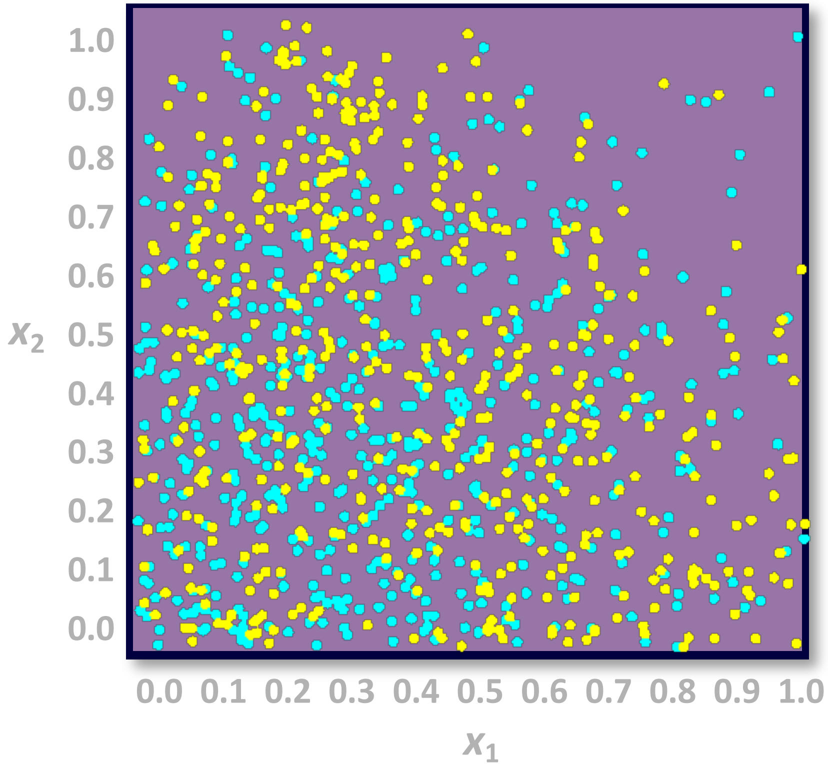
...

Decision Tree Split Search

	left	right
	53%	42%
	47%	58%

max Gain(x_1) 0.0112

Repeat for input x_2.



...

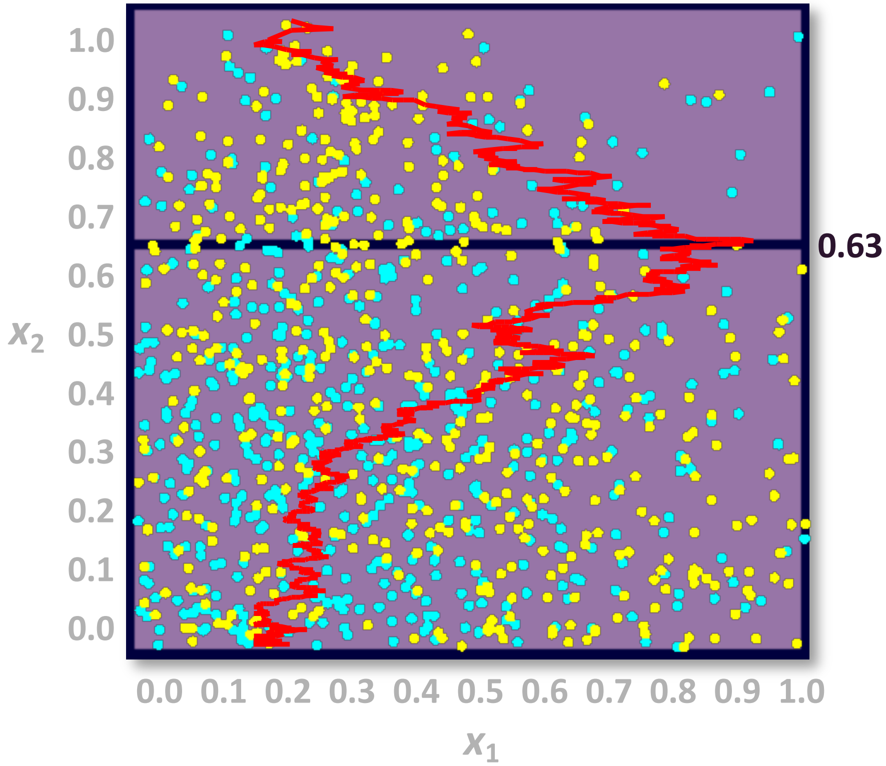
Decision Tree Split Search

	left	right
	53%	42%
	47%	58%

max gain(x_1) 0.0112

	bottom	top
	54%	35%
	46%	65%

max gain(x_2) 0.0273



...

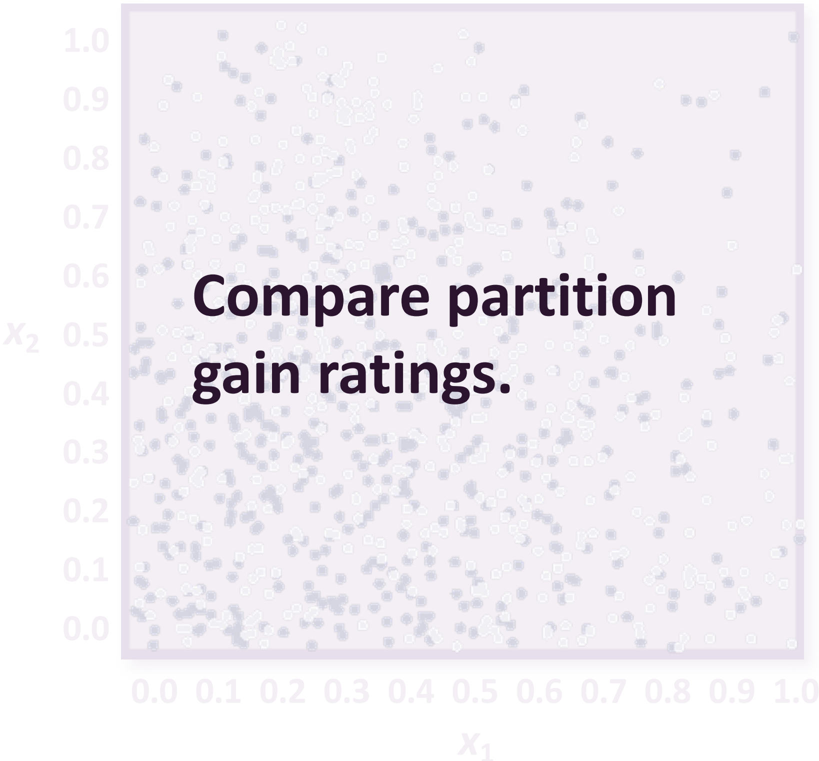
Decision Tree Split Search

	left	*right*
(cyan)	53%	42%
(yellow)	47%	58%

max
gain(x_1)
0.0112

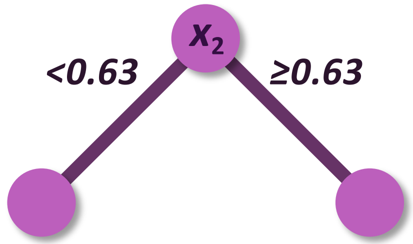
	bottom	*top*
(cyan)	54%	35%
(yellow)	46%	65%

max
gain(x_2)
0.0273

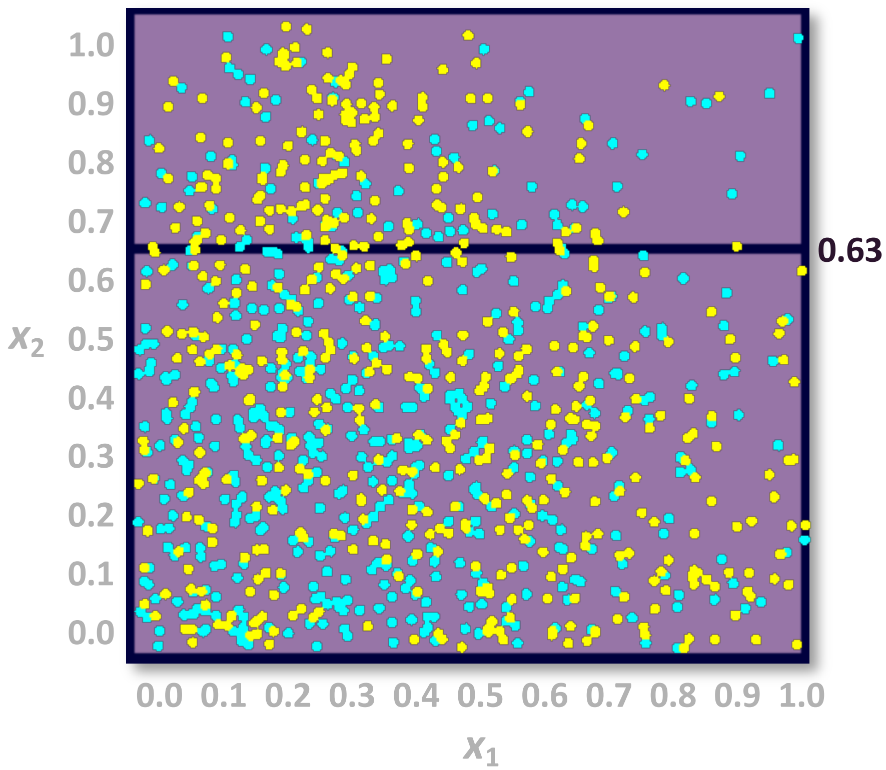


...

Decision Tree Split Search

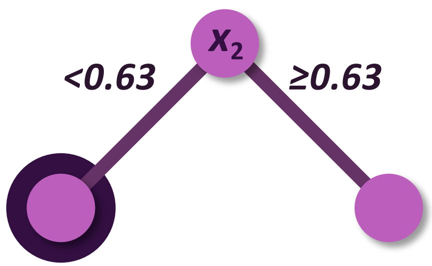


Create a partition rule from the best partition across all inputs.

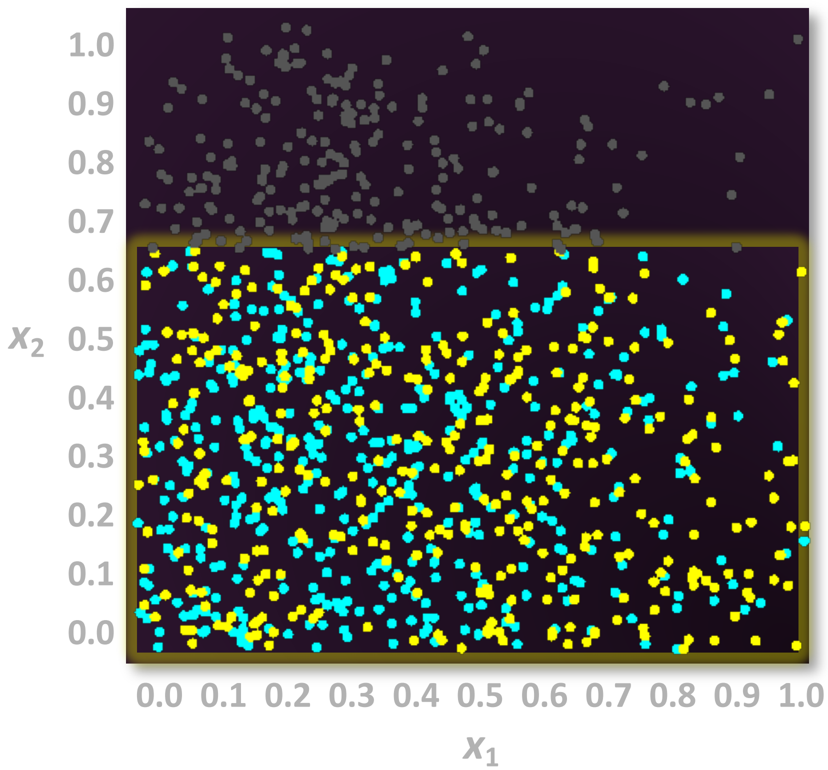


...

Decision Tree Split Search



Repeat the process in each subset.

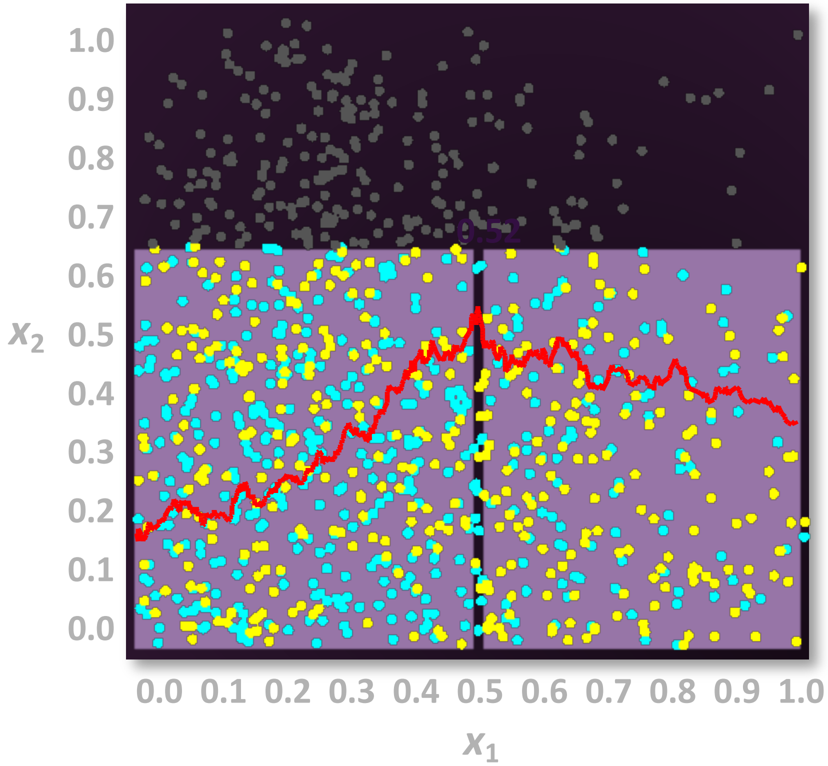


...

Decision Tree Split Search

	left	*right*
	61%	55%
	39%	45%

max gain(x_1)
0.0203



...

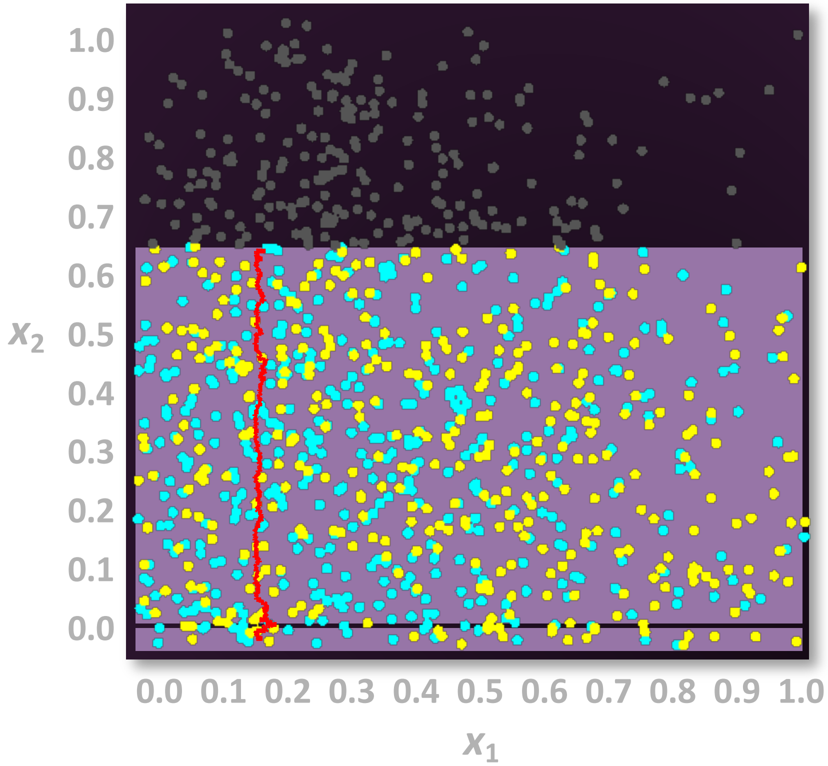
Decision Tree Split Search

	left	*right*
	61%	55%
	39%	45%

max gain(x_1)
0.0203

	bottom	*top*
	38%	55%
	62%	45%

max gain(x_2)
0.019



...

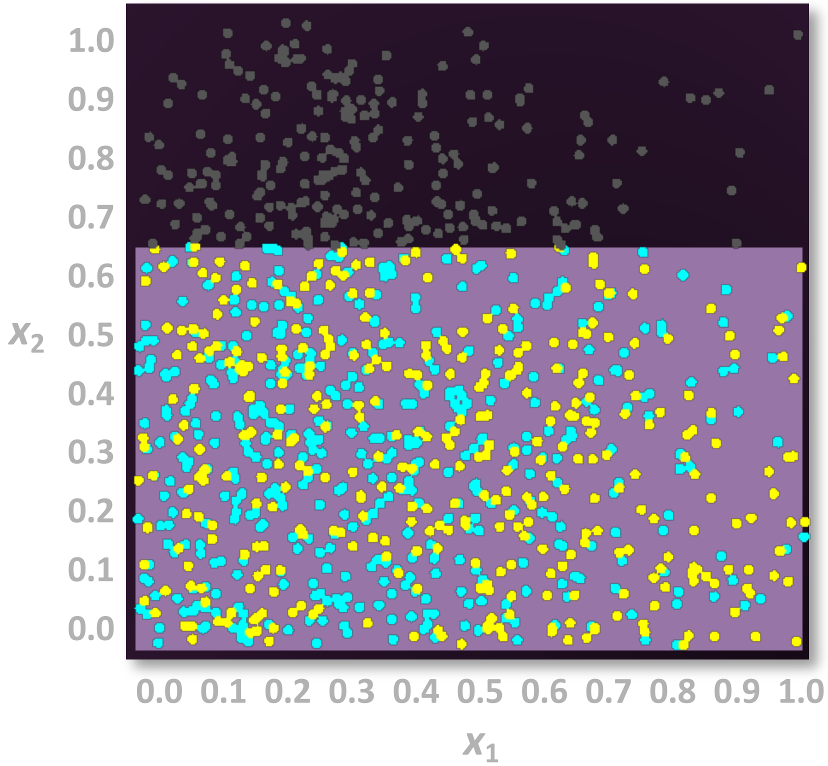
Decision Tree Split Search

	left	*right*
cyan	61%	55%
yellow	39%	45%

max gain(x_1) **0.0203**

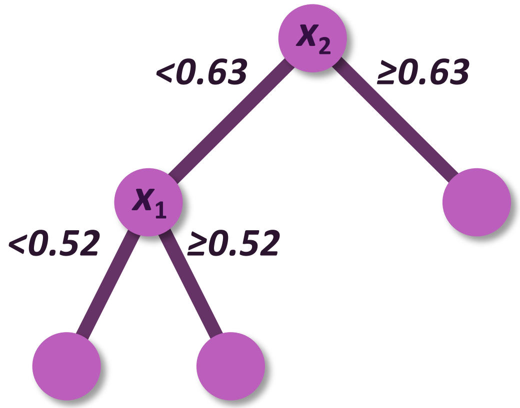
	bottom	*top*
cyan	38%	55%
yellow	62%	45%

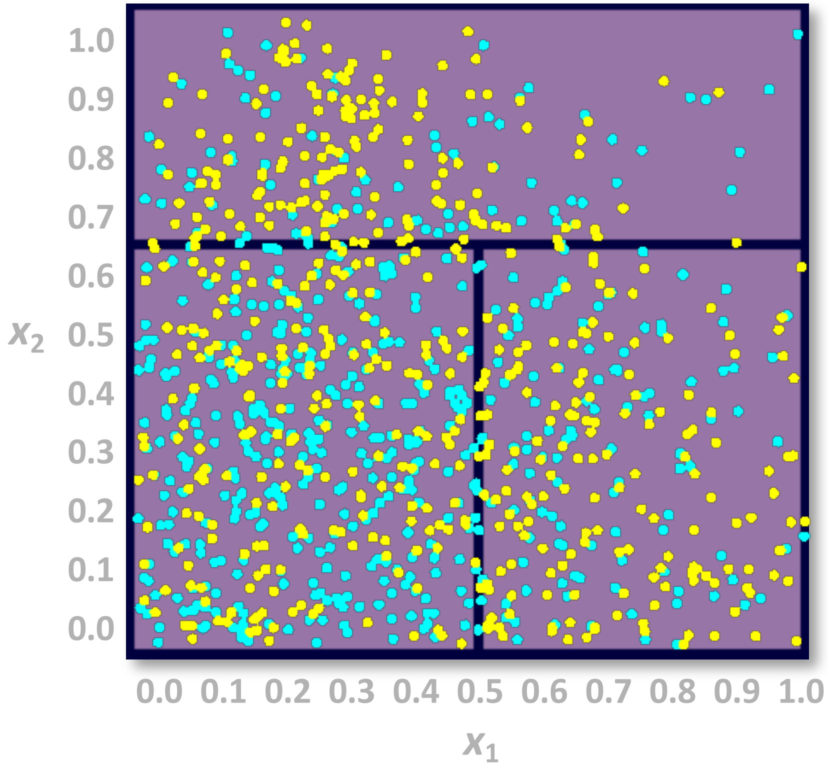
max gain(x_2) **0.019**



...

Decision Tree Split Search

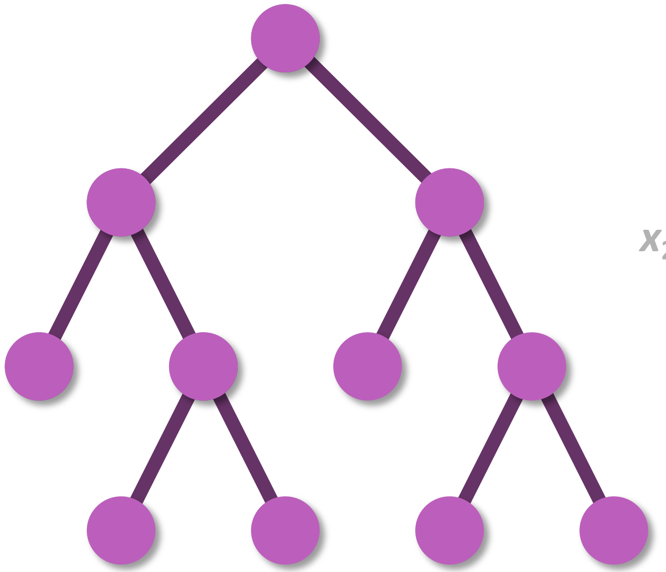


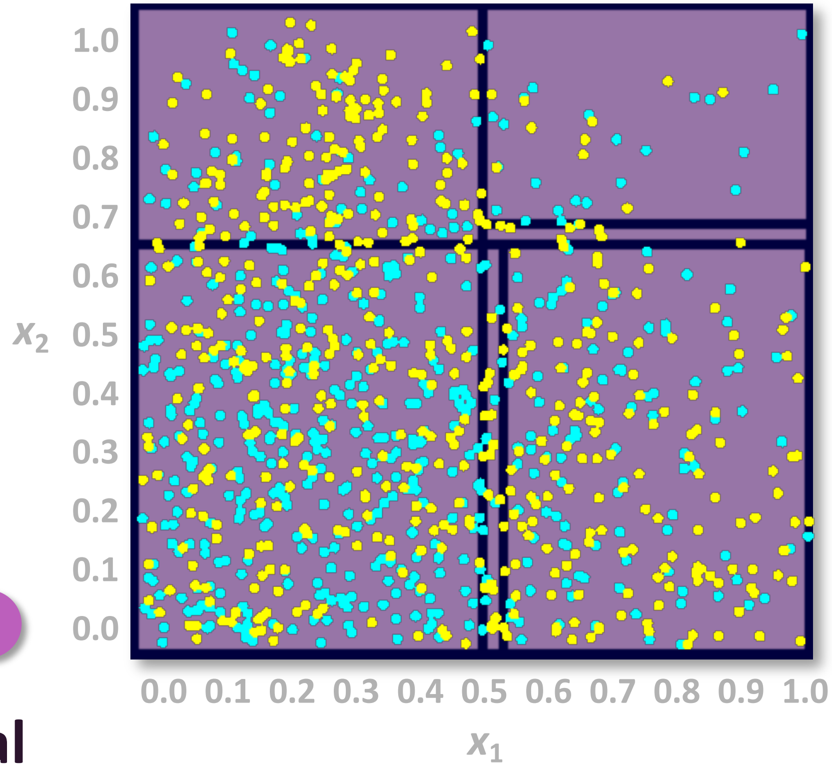


Create a second partition rule.

...

Decision Tree Split Search





Repeat to form a maximal tree.

1) Entropy (impurity)

- **Entropy** is a measure of disorder or uncertainty and the goal of machine learning models and general is to reduce uncertainty.

$$Entropy = \sum_{i=1}^{n} -p_i log_2 p_i$$

scipy.stats.entropy

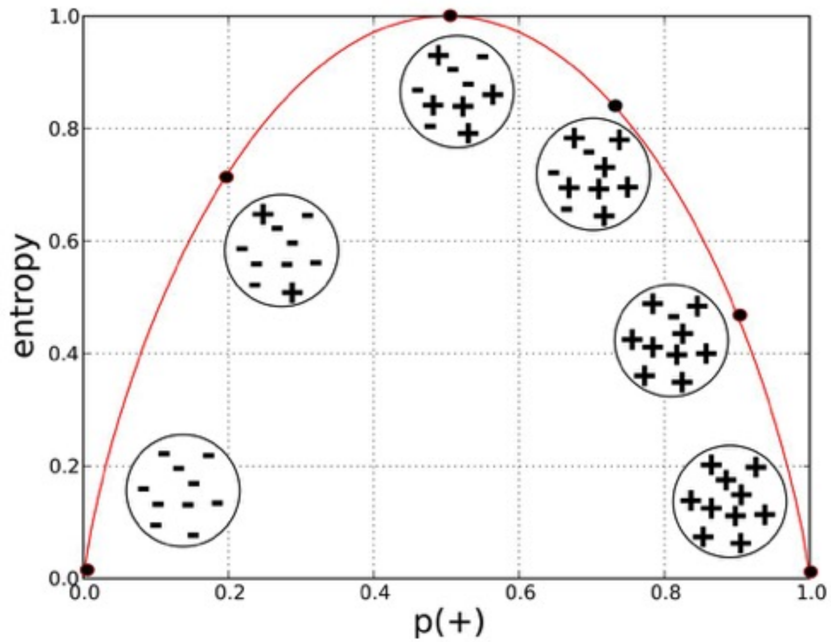
scipy.stats.**entropy**(*pk*, *qk=None*, *base=None*, *axis=0*)

Calculate the entropy of a distribution for given probability values.

If only probabilities *pk* are given, the entropy is calculated as `S = -sum(pk * log(pk), axis=axis)`.

If *qk* is not None, then compute the Kullback-Leibler divergence `S = sum(pk * log(pk / qk), axis=axis)`.

This routine will normalize *pk* and *qk* if they don't sum to 1.



Source: Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking

Information Gain
Before - After

- Which one is Better ?

Split w/ Age: 70
∑ Entropy: 0.350

Split w/ Age: 50
∑ Entropy: 0.348

- Information Gain: measure the reduction of this disorder in our target variable/class given additional information

$$InformationGain = Entropy(before) - \sum Entropy(after)$$

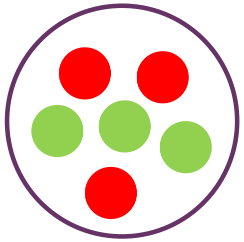
- "Before" = Entropy of Parent Node
  "After" = Entropy of Child Nodes

2) Gini Impurity
Gini Reduction = Before - After

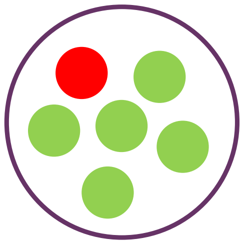
- Another way to measure how well a splitting feature.

$$Gini = 1 - \sum_{i=1}^{n} (P_i)^2$$

When P is the probability of class i in data-set.



$$Gini = 1 - ((3/6)^2 + (3/6)^2) \\ = 0.5$$



$$Gini = 1 - ((5/6)^2 + (1/6)^2) \\ = 0.28$$

- Easy to calculation, may take <u>less time</u> to build in large dataset.

Source: https://towardsdatascience.com/understanding-decision-tree-classification-with-scikit-learn-2ddf272731bd

Types of Decision Tree

Algorithm	Splitting Measure
ID3	Entropy
C4.5	Gain Ratio
CART	Gini index
CHAID	Chi-squared test

Important Parameters in Decision Tree

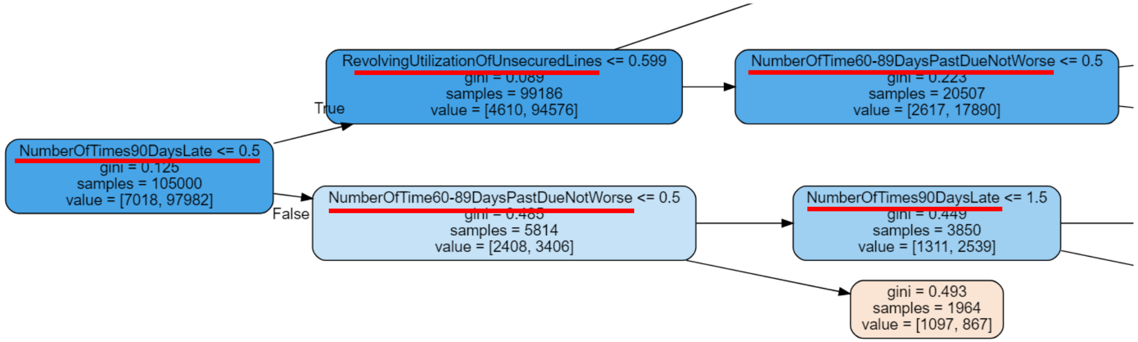
- Splitting measure (criterion) : gini / entropy
- Maximum depth : ~5-10 (depend on number of feature)
- Maximum leaf nodes : depend on number of class (target) and feature
- Minimum sample split : 5 – 20% (depend on number of data)

Which features are important ?

- Check how important by . feature_importances_
- The importance of a feature is computed as the (normalized) total reduction of the criterion brought by that feature. It is also known as the gini importance.

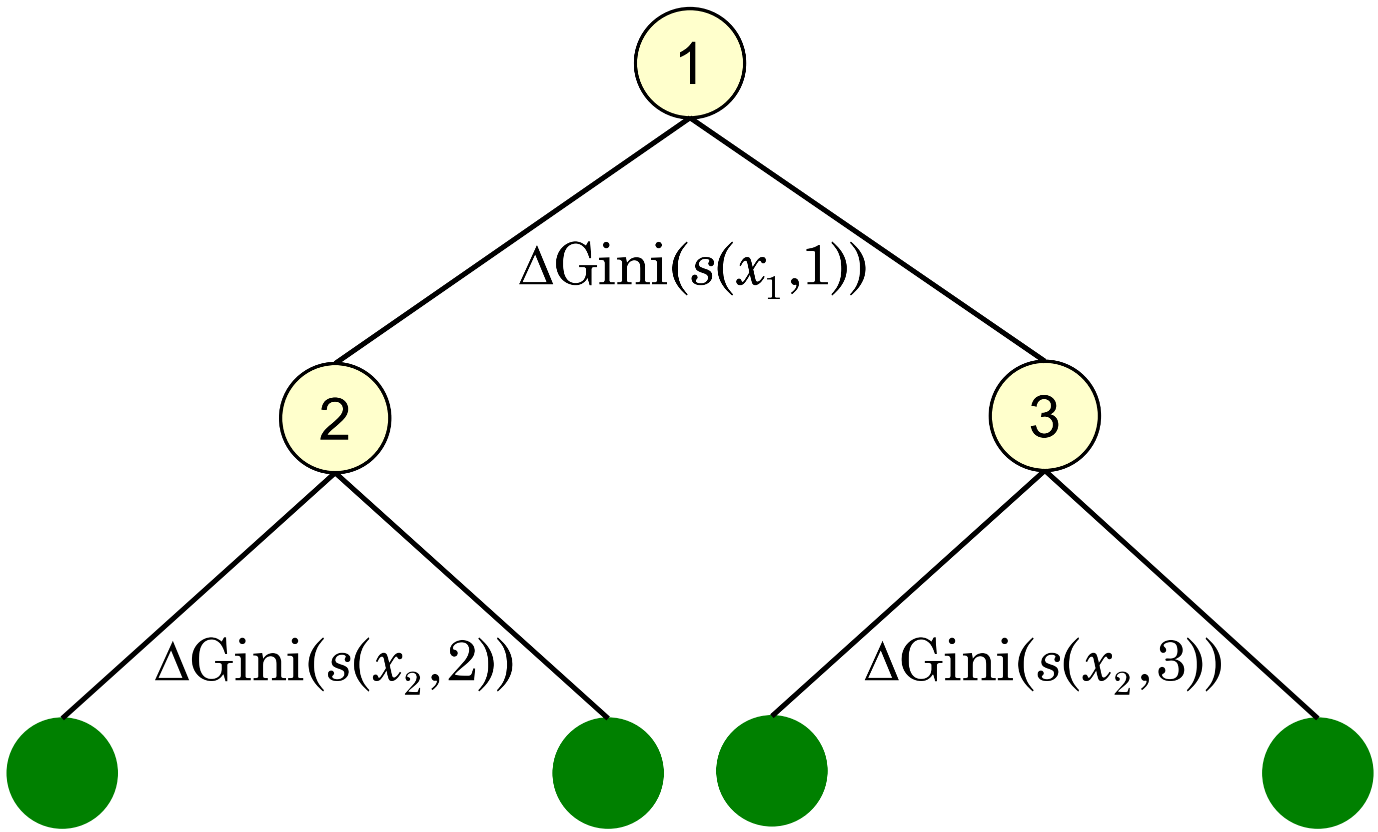
```
dtree.feature_importances_
```

```
array([0.15330291, 0.00371739, 0.07500437, 0.        , 0.        ,
       0.        , 0.61403332, 0.        , 0.15394201, 0.        ])
```




```
======= Features Important =======
0.614 : Numberoftimes90dayslate
0.154 : Numberoftime60-89dayspastduenotworse
0.153 : Revolvingutilizationofunsecuredlines
0.075 : Numberoftime30-59dayspastduenotworse
0.004 : Age
0.000 : Debtratio
0.000 : Monthlyincome
0.000 : Numberofopencreditlinesandloans
0.000 : Numberrealestateloansorlines
0.000 : Numberofdependents
```

+ Variable Importance

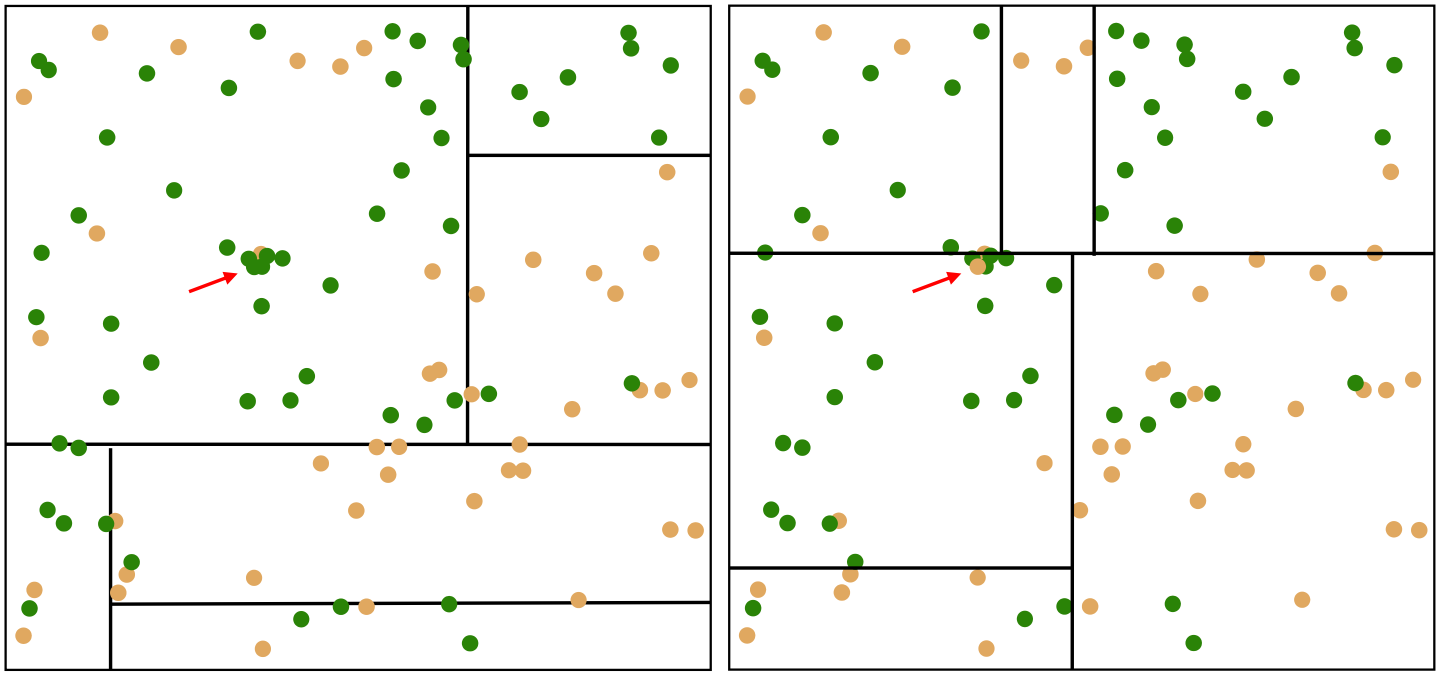


Random Forest

CHULA ΣNGINEERING COMPUTER
Foundation toward Innovation

+ Instability

One reversal



Accuracy = 81%

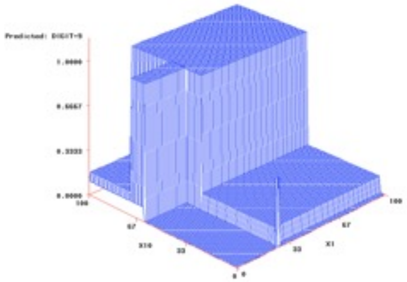
Accuracy = 80%

...

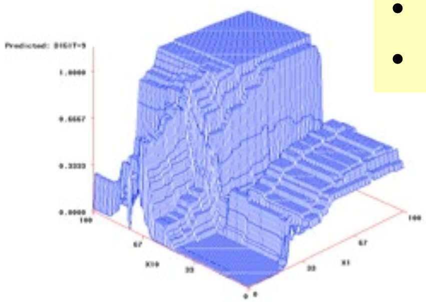
Random Forest Algorithm (RF)

- Forest algorithm samples the rows and the columns at each step.
- Forest takes bootstrap samples of the rows in training data (sampling with replacement)
- At each step, a set of variables (columns) is sampled.
- This increases variation among trees in the **ensemble** often leads to improved prediction accuracy.

Important Params:
- #Tree
- #Columns (variables)
- #Rows (examples)

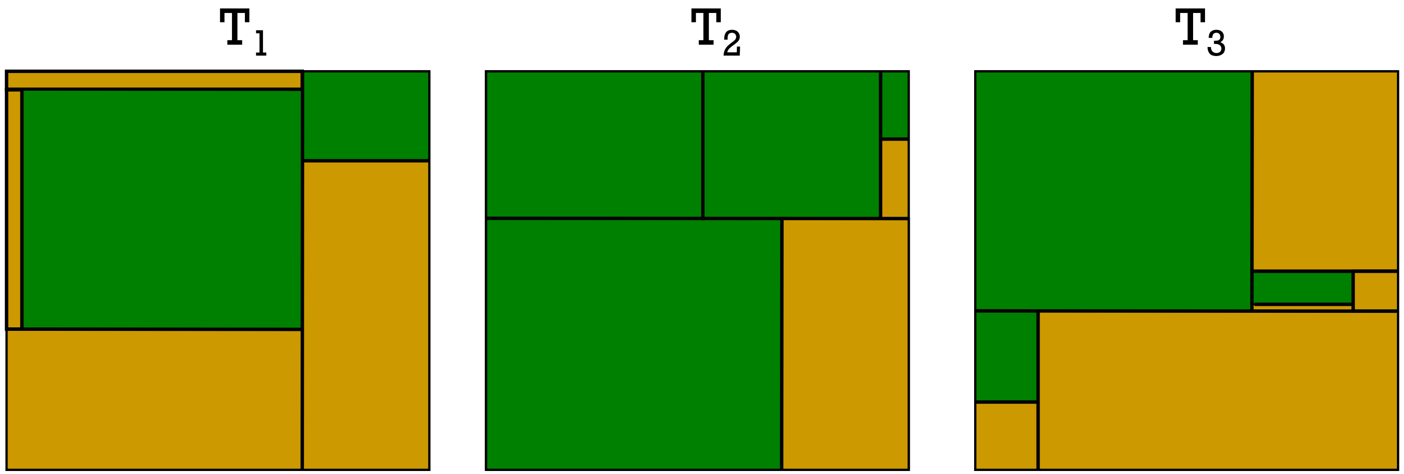


Single Tree

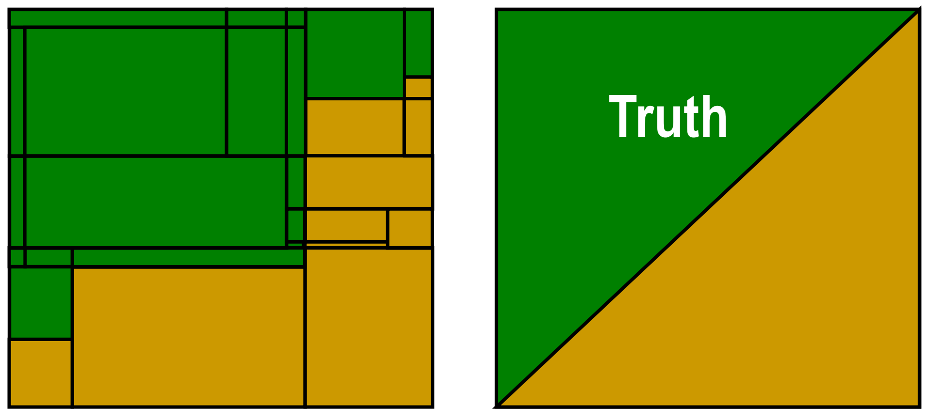


Random Forest

Combine



$\text{avg}(T_1, T_2, T_3) =$



...

CHULA ΣNGINEERING
Foundation toward Innovation
COMPUTER

Classification Performance

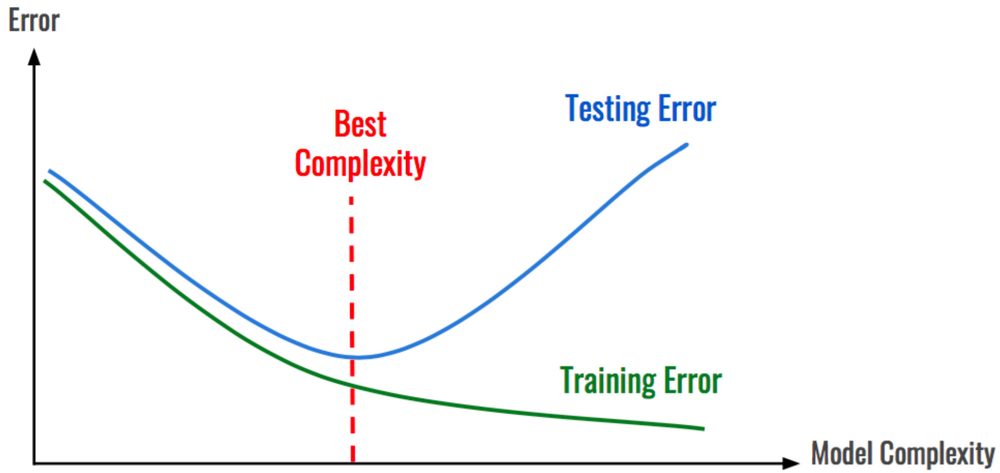
Evaluation (Train/Test Split)

Training Data

Age	Income	Purchase
25	25,000	Yes
35	50,000	Yes
32	35,000	No

Testing Data

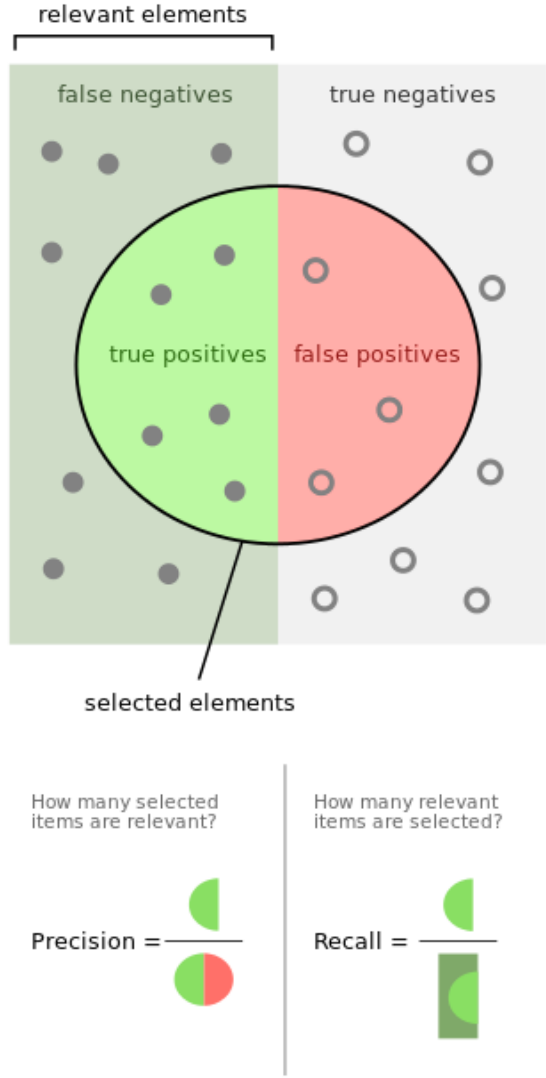
Age	Income	Purchase
27	35,000	Yes
23	20,000	No
45	34,000	No



Confusion Matrix
Precision, Recall, F1

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

- Precision = correctly predict = TP / (TP + FP)
- Recall = coverage = TP / (TP + FN)
- F1 = (2*pre*rec) / (pre + rec)



+

Thank you & any questions