# Decision Trees

Jayanth Rasamsetti
Founder & Chief Scientist
www.sgmoid.com
Columbia University (MS)
IIT Madras (B.Tech & M.Tech)
jr3281@columbia.edu
https://www.linkedin.com/in/jayanthr/

# Supervised Machine Learning

**What is a decision tree?**

**Lets Play a Game: Guess the Animal**

1) I am thinking of an Animal. You can ask a set of questions (on features of the animal)
2) Can you guess the animal based on my answers?
3) Conditions:

   Only Yes/No questions or questions on a single attribute
   No questions based on the animal name itself

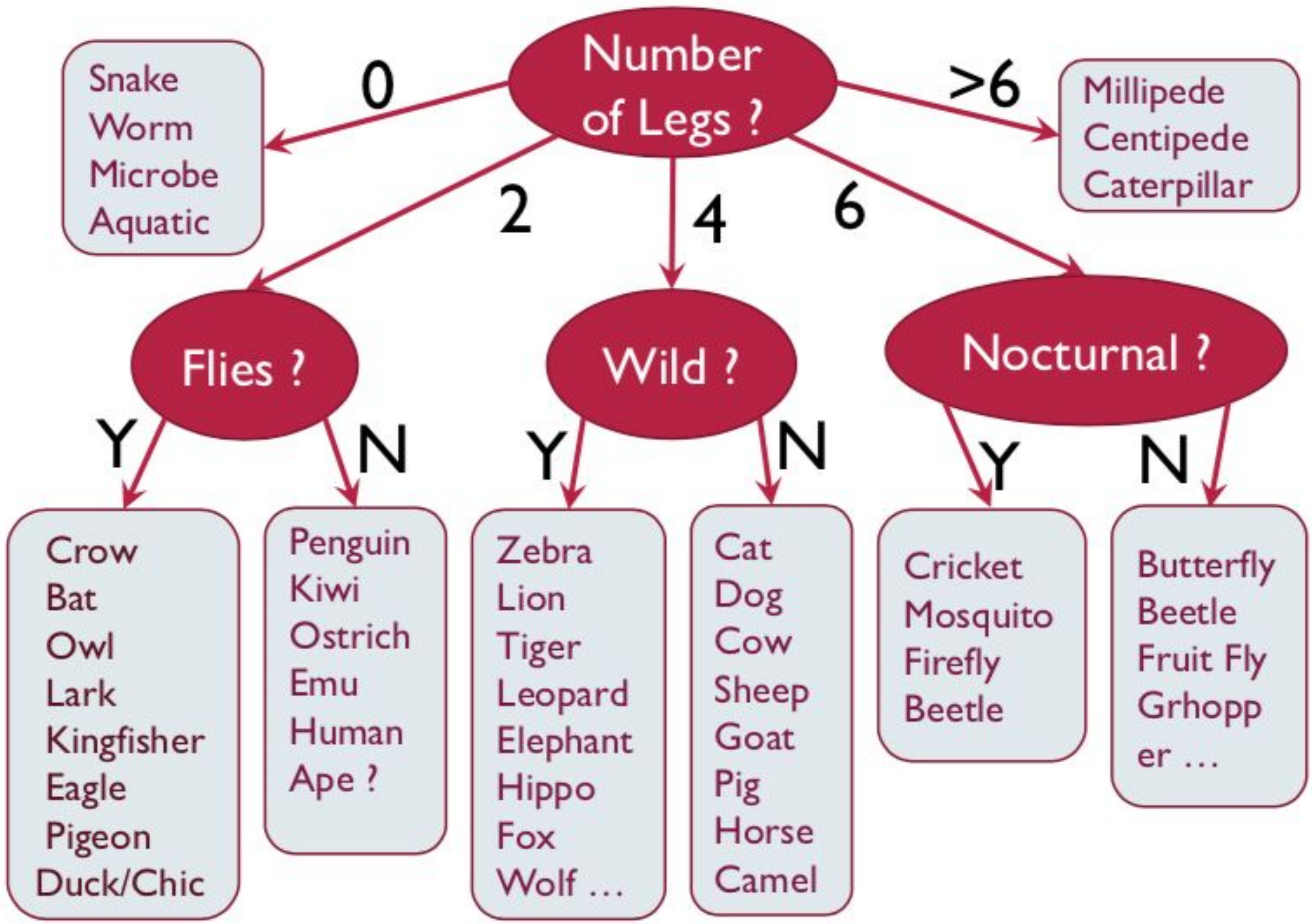
Let’s begin!

# Supervised Machine Learning

**Guess the animal**

**Questions**

1) How many legs?
2) Does it fly?
3) Wild animal?
4) Nocturnal?
5) Fur/Feather?
6) Farm Animal?



# Supervised Machine Learning

**Larger Picture**

1) We have lots of possible animals

2) Each has a set of attributes

3) Look at 1 attribute at a time

4) Narrow down the class label

| Animal | Legs | Wild | Flies | Noct | Fur/Feather | Farm |
|---|---|---|---|---|---|---|
| Zebra | 4 | Y | N | N | N | N |
| Horse | 4 | Y/N | N | N | N | Y |
| Cow | 4 | N | N | N | N | Y |
| Cat | 4 | Y/N | N | Y/N | Y | N |
| Penguin | 2 | Y | N | N | N | N |
| Owl | 2 | Y | Y | Y | Y | N |
| Fish | 0 | Y | N | Y/N | N | N |
| Snake | 0 | Y | N | Y/N | N | N |
| Millipede | 1000 | Y | N | Y | N | N |
| Firefly | 6 | Y | Y | Y | N | N |
| Butterfly | 6 | Y | Y | N | N | N |

# Supervised Machine Learning

**What are we doing?**
**(Larger Picture)**

**Goal**

Arrive at a <u>single class</u> label

Can we learn the *tree*?

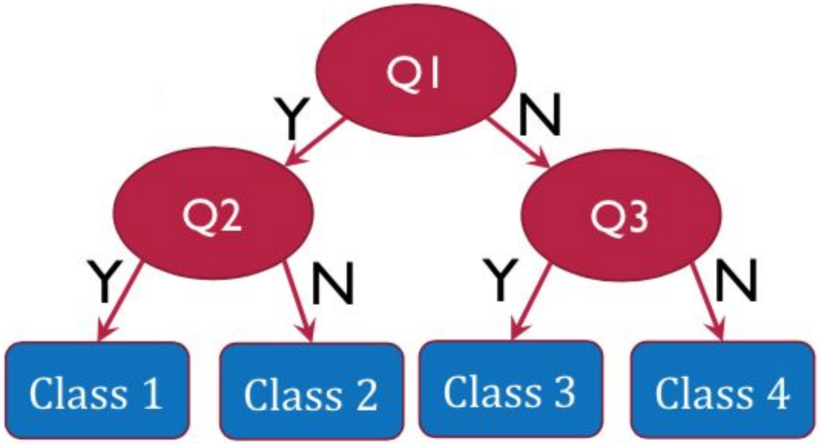
*“Which question to ask at any point?”*

| Animal | Legs | Wild | Flies | Noct | Fur/Feather | Farm |
|---|---|---|---|---|---|---|
| Zebra | 4 | Y | N | N | N | N |
| Horse | 4 | Y/N | N | N | N | Y |
| Cow | 4 | N | N | N | N | Y |
| Cat | 4 | Y/N | N | Y/N | Y | N |
| Penguin | 2 | Y | N | N | N | N |
| Owl | 2 | Y | Y | Y | Y | N |
| Fish | 0 | Y | N | Y/N | N | N |
| Snake | 0 | Y | N | Y/N | N | N |
| Millipede | 1000 | Y | N | Y | N | N |
| Firefly | 6 | Y | Y | Y | N | N |
| Butterfly | 6 | Y | Y | N | N | N |

# Supervised Machine Learning

**Decision Trees: Summary. Steps**

1) At each step, ask the question that minimizes uncertainty (entropy)
2) Once the set contains only a single class, label it
3) The sequence of questions/decisions can be represented as a tree: The Decision Tree
4) Each question is on a single feature
5) Tree Terms: Node, Edge, Root, Leaf, Depth, Height, Path, Parent, Child



# Supervised Machine Learning

**Decision Trees**

What is the best ways to split? (Mathematically?)

*Entropy*

# Supervised Machine Learning

**Decision Trees**

What is the best ways to split? (Mathematically?)

| | Gender | Smokes |
|---|---|---|
| Healthy | 5 | 6 |
| Cancer | 5 | 4 |

# Supervised Machine Learning

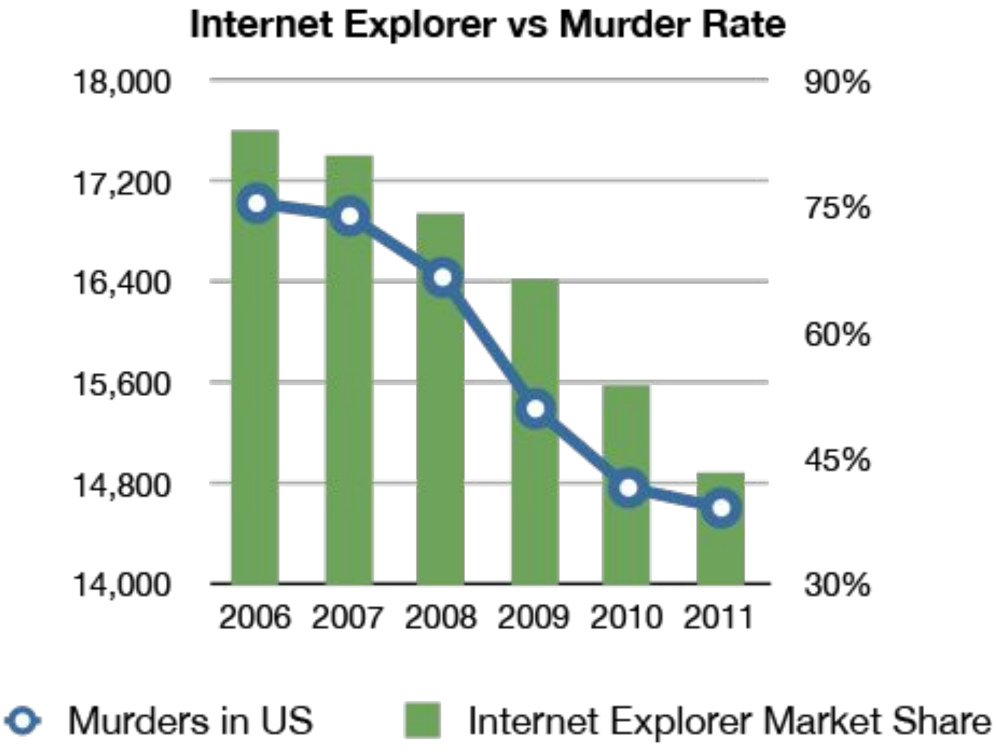
**Decision Trees**

What is the best ways to split? (Mathematically?)

| | Gender | Smokes |
|---|---|---|
| Healthy | 9 | 6 |
| Cancer | 1 | 4 |

# Supervised Machine Learning

Caution: Correlation need not imply causation!



# Supervised Machine Learning

**What is Entropy?**

1. Is a measure of Uncertainty

$$H(X) = -\sum_i P(i)\, log_2\, P(i)$$

2. Assume a set contains two classes

$$H = -P_1\, log_2\, P_1 - P_2\, log_2\, P_2$$

3. Information Gain / Loss function

Information Gain is the Entropy/Gini of the parent node minus the entropy/gini of the child nodes

$$IG(D_p, x_i) = I(D_p) - \frac{N_{left}}{N_p} I(D_{left}) - \frac{N_{right}}{N_p} I(D_{right})$$

# Supervised Machine Learning

**Common measures of purity**

$$Entropy = \sum_{i=1}^{C} -p_i * \log_2(p_i)$$

**Entropy:**

Favors splits with small counts but many unique values

Weights probability of class by log(base=2) of the class probability

A smaller value of Entropy is better. That makes the difference between the parent node's entropy larger

**Gini:**

Subtract the sum of the squared probabilities of each class from one

Uses squared proportion of classes

Perfectly classified, Gini Index would be zero

Evenly distributed would be 1 – (1/# Classes)

You want a variable split that has a low Gini Index

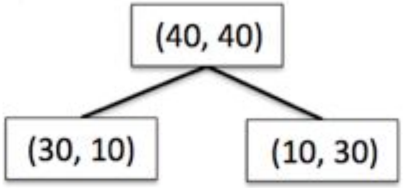
Used in CART algorithm

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

# Supervised Machine Learning

Information Gain using Entropy

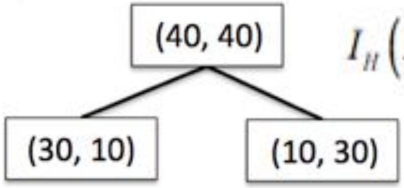
$$H(X) = -\sum_{i=0}^{N-1} p_i \log_2 p_i$$



# Supervised Machine Learning

Information Gain using Entropy

$$H(X) = -\sum_{i=0}^{N-1} p_i \log_2 p_i$$

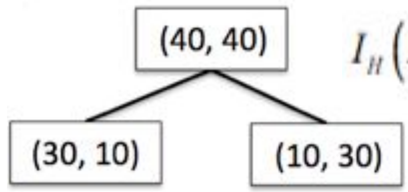



$$I_H(D_p) = -\left(0.5 \log_2(0.5) + 0.5 \log_2(0.5)\right) = 1$$

# Supervised Machine Learning

Information Gain using Entropy

$$H(X) = -\sum_{i=0}^{N-1} p_i \log_2 p_i$$




$$I_H(D_p) = -(0.5 \log_2(0.5) + 0.5 \log_2(0.5)) = 1$$

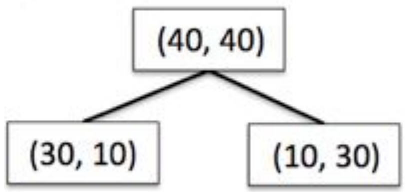
$$-\left(\frac{3}{4}\log_2\left(\frac{3}{4}\right) + \frac{1}{4}\log_2\left(\frac{1}{4}\right)\right) = 0.81 \qquad -\left(\frac{1}{4}\log_2\left(\frac{1}{4}\right) + \frac{3}{4}\log_2\left(\frac{3}{4}\right)\right) = 0.81$$

Information Gain = reduction in entropy = $1 - \frac{4}{8}0.81 - \frac{4}{8}0.81 = 0.19$

# Supervised Machine Learning

Information Gain using Gini index

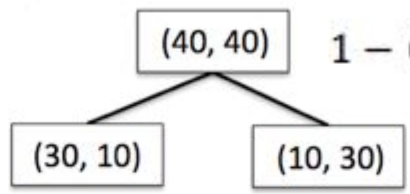
$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$



# Supervised Machine Learning

Information Gain using Gini index

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

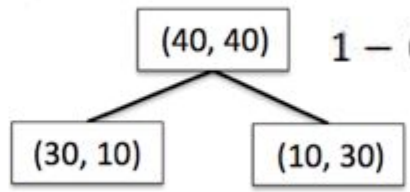



$$1 - (0.5^2 + 0.5^2) = 0.5$$

# Supervised Machine Learning

Information Gain using Gini index

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$




$$1 - (0.5^2 + 0.5^2) = 0.5$$

$$1 - \left(\left(\frac{3}{4}\right)^2 + \left(\frac{1}{4}\right)^2\right) = \frac{3}{8} = 0.375 \qquad 1 - \left(\left(\frac{1}{4}\right)^2 + \left(\frac{3}{4}\right)^2\right) = \frac{3}{8} = 0.375$$

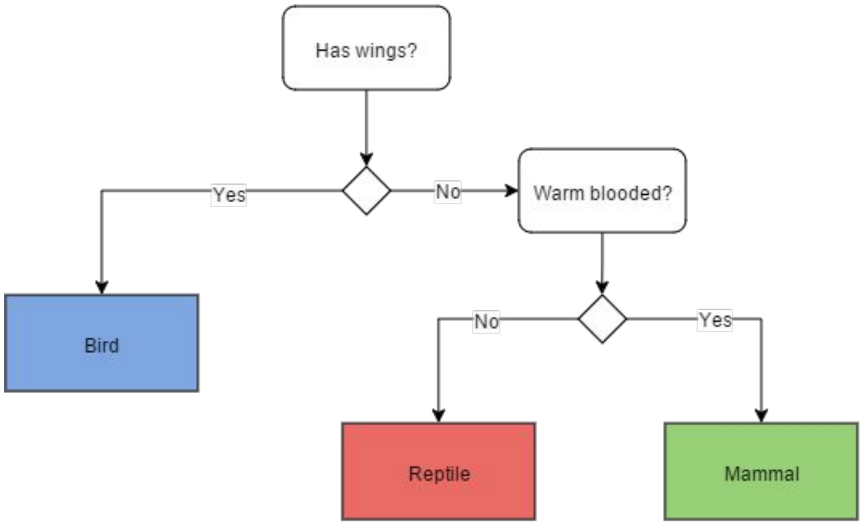
Information Gain = reduction in Gini index = $0.5 - \frac{4}{8}0.375 - \frac{4}{8}0.375 = 0.125$

# Supervised Machine Learning

**Decision Trees (Classification)**

1) For classification problem, the posterior probability of all the classes is reflected in the leaf node and the Leaf Node belongs to the majority class

2) After executing all the functions from Root Node to Leaf Node, the class of a data point is decided by the leaf node to which it reaches

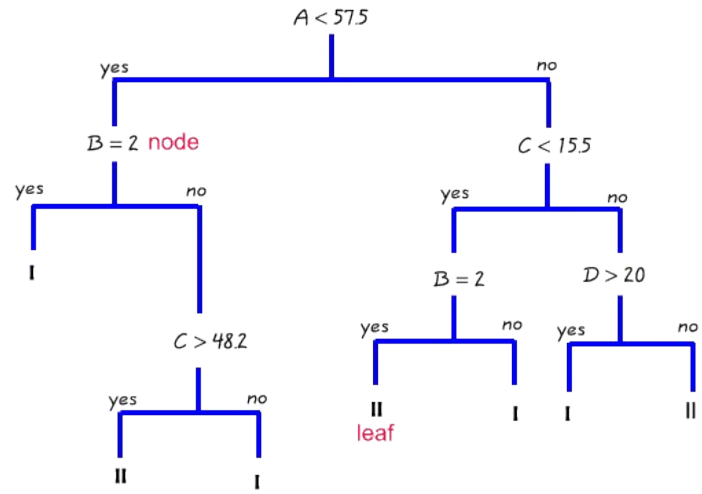


# Supervised Machine Learning

## Decision Trees (Regression)

1) For regression, the average/ median value of the target attribute is assigned to the query variable

2) Tree creation splits data into subsets and subsets into further smaller subsets. The algorithm stops splitting data when data within the subsets are sufficiently homogenous or some other stopping criterion is met



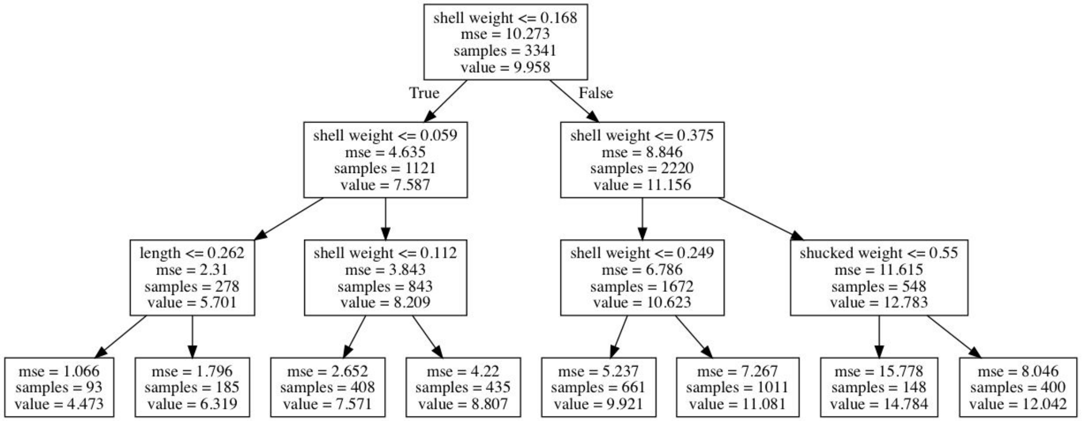
# Supervised Machine Learning

**Loss Function (Different from our SSE!)**

1. The decision tree algorithm learns (i.e. creates the decision tree from the data set) through optimization of a loss function

2. The loss function represents the loss of impurity in the target column. The requirement here is to minimize the impurity as much as possible at the leaf nodes

3. Purity of a node is a measure of homogeneity in the target column at that node



**Decision Trees: Recently Asked Interview Question**

Decision tree only works for classification algorithms

a) True
b) False

UBER

# Supervised Machine Learning

**Decision Trees: Recently Asked Interview Question**

Decision tree only works for classification algorithms

a) True
b) False

Ans) False

# Supervised Machine Learning

**Common** Algorithms (Applicable to both R and Python)

1. ID3 (Iterative Dichotomiser 3) (Author: Quinian) Creates a multi branch tree at each node using greedy algorithm. Trees grow to maximum size before pruning

2. C4.5 succeeded ID3 by overcoming limitation of features required to be categorical. It dynamically defines discrete attribute for numerical attributes. It converts the trained trees into a set of if-then rules. Accuracy of each rule is evaluated to determine the order in which they should be applied

3. C5.0 is Quinlan's latest version and it uses less memory and builds smaller rulesets than C4.5 while being more accurate

# Supervised Machine Learning

CART (Classification & Regression Trees)

Similar to C4.5 but it supports numerical target variables and does not compute rule sets. Creates binary tree

**Scikit uses CART**

```
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor(random_state=0, max_depth=3)
```

# Supervised Machine Learning

Decision Trees

Advantages:

1. Simple, Fast in processing and effective
2. Can deal with missing data, handles numeric and categorical variables
3. Interpretation of results does not require mathematical or statistical knowledge

Disadvantages

1. Often biased towards splits or features have large number of levels
2. Small changes in training data can result in large changes to the logic
3. Large trees can be difficult to interpret

# Supervised Machine Learning

Preventing overfitting through tuning (hyper parameters)

1. Decision trees do not assume a particular form of relationship between the independent and dependent variables unlike linear models

2. DT is a non-parameterized algorithm unlike linear models where we supply the input parameters

3. If left unconstrained, they can build tree structures to adapt to the training data leading to overfitting

4. To avoid overfitting, we need to restrict the DT’s freedom during the tree creation. This is called regularization

5. The regularization hyperparameters depend on the algorithms used

# Supervised Machine Learning

Decision trees are notorious for overfitting. Solutions:

1. Early Stopping
2. Do not split a not beyond a point (of number of items or purity)
3. Pruning Once the tree is formed, remove weakest branches. Use validation set to decide when to stop
4. Both approaches also reduces the depth and improves classification speed

# Supervised Machine Learning

**Decision Tree Hyper parameters**

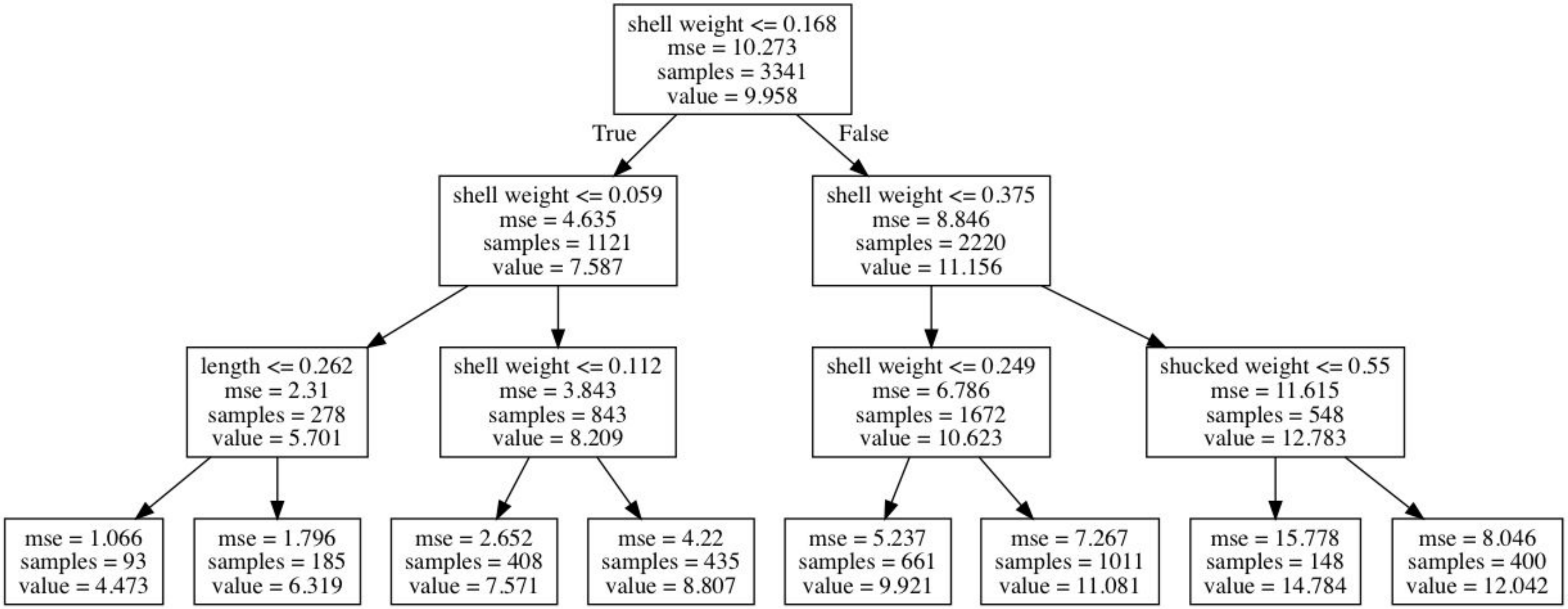
1. max_depth – Is the maximum length of a path from root to leaf (in terms of number of decision points. The leaf node is not split further. It could lead to a tree with leaf node containing many observations on one side of the tree, whereas on the other side, nodes containing much less observations get further split

2. min_sample_split - A limit to stop further splitting of nodes when the number of observations in the node is lower than this value

3. min_sample_leaf – Minimum number of samples a leaf node must have. When a leaf contains too few observations, further splitting will result in overfitting (modeling of noise in the data).

# Supervised Machine Learning

4. min_weight_fraction_leaf – Same as min_sample_leaf but expressed in fraction of total number of weighted instances
5. max_leaf_nodes – maximum number of leaf nodes in a tree
6. max_feature_size - max number of features that are evaluated for splitting each node



# Supervised Machine Learning



Case Study: Approximately ***1 in 8*** women will be diagnosed with Breast Cancer. You are hired as a Data Scientist by the Wisconsin Hospital Association to help screen patients in rural areas. How do we go about building the classifier?

# Supervised Machine Learning

569 rows and 32 columns

| | id | diagnosis | radius_mean | texture_mean | perimeter_mean | area_mean | smoothness_mean | compactness_mean | conc |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 87139402 | B | 12.32 | 12.39 | 78.85 | 464.1 | 0.10280 | 0.06981 | |
| **1** | 8910251 | B | 10.60 | 18.95 | 69.28 | 346.4 | 0.09688 | 0.11470 | |
| **2** | 905520 | B | 11.04 | 16.83 | 70.92 | 373.2 | 0.10770 | 0.07804 | |
| **3** | 868871 | B | 11.28 | 13.39 | 73.00 | 384.8 | 0.11640 | 0.11360 | |
| **4** | 9012568 | B | 15.19 | 13.21 | 97.65 | 711.8 | 0.07963 | 0.06934 | |
| **5** | 906539 | B | 11.57 | 19.04 | 74.20 | 409.7 | 0.08546 | 0.07722 | |
| **6** | 925291 | B | 11.51 | 23.93 | 74.52 | 403.5 | 0.09261 | 0.10210 | |
| **7** | 87880 | M | 13.81 | 23.75 | 91.56 | 597.8 | 0.13230 | 0.17680 | |
| **8** | 862989 | B | 10.49 | 19.29 | 67.41 | 336.1 | 0.09989 | 0.08578 | |
| **9** | 89827 | B | 11.06 | 14.96 | 71.49 | 373.9 | 0.10330 | 0.09097 | |

Benign (Healthy)
Malignant (Cancer)

# Supervised Machine Learning

569 rows and 32 columns

| | id | diagnosis | radius_mean | texture_mean | perimeter_mean | area_mean | smoothness_mean | compactness_mean | conc |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 87139402 | B | 12.32 | 12.39 | 78.85 | 464.1 | 0.10280 | 0.06981 | |
| **1** | 8910251 | B | 10.60 | 18.95 | 69.28 | 346.4 | 0.09688 | 0.11470 | |
| **2** | 905520 | B | 11.04 | 16.83 | 70.92 | 373.2 | 0.10770 | 0.07804 | |
| **3** | 868871 | B | 11.28 | 13.39 | 73.00 | 384.8 | 0.11640 | 0.11360 | |
| **4** | 9012568 | B | 15.19 | 13.21 | 97.65 | 711.8 | 0.07963 | 0.06934 | |
| **5** | 906539 | B | 11.57 | 19.04 | 74.20 | 409.7 | 0.08546 | 0.07722 | |
| **6** | 925291 | B | 11.51 | 23.93 | 74.52 | 403.5 | 0.09261 | 0.10210 | |
| **7** | 87880 | M | 13.81 | 23.75 | 91.56 | 597.8 | 0.13230 | 0.17680 | |
| **8** | 862989 | B | 10.49 | 19.29 | 67.41 | 336.1 | 0.09989 | 0.08578 | |
| **9** | 89827 | B | 11.06 | 14.96 | 71.49 | 373.9 | 0.10330 | 0.09097 | |

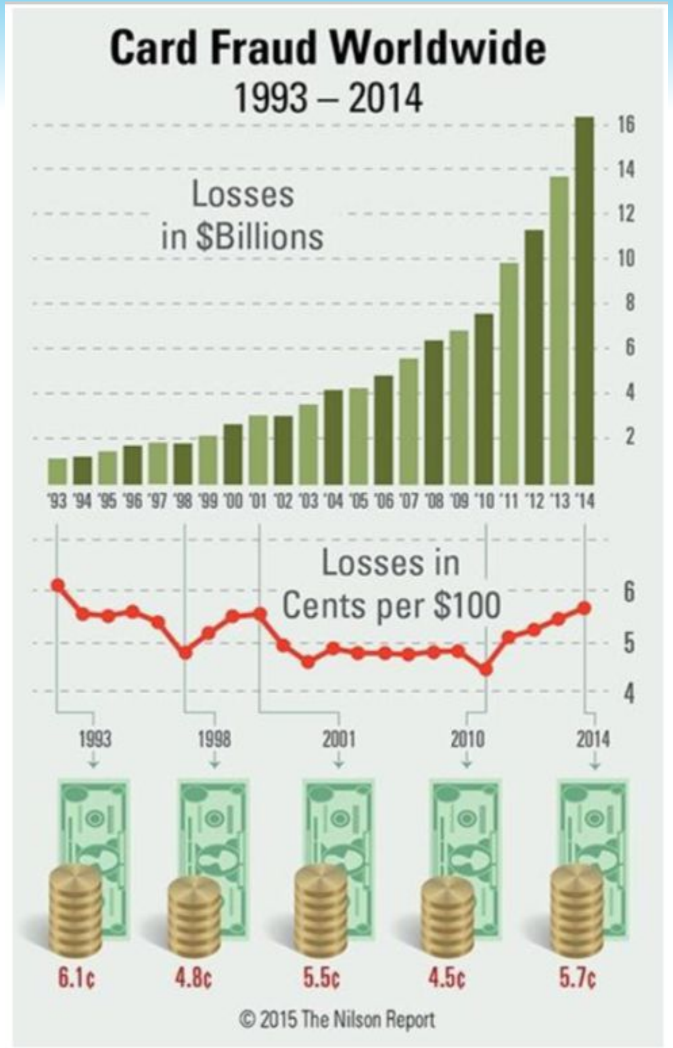
Decision Tree: 87.5 (max_depth: 1)

Decision Tree: ? (Change max_depth: 3)

# Supervised Machine Learning

**Credit Card Fraud Detection**

An e-commerce site has been offered a fraud detection system promising high accuracy. It correctly catches 99% of fraudulent transactions and incorrectly flags 1% of honest transactions as fraudulent.




Source:
http://www.nilsonreport.com/publication_chart_and_graphs_archive.php?1=1&year=2015
Last accessed: October 16, 2015

# Supervised Machine Learning

**Credit Card Fraud Detection**

1) What will be the implications to the retailer of mistaking honest transactions as fraudulent?

2) What will be the implications to the retailer of mistaking fraudulent transactions as honest?

3) What types of errors are the above two? Which one is more critical to the business?

| | |
|---|---|
| True Positive | True Negative |
| False Positive | False Negative |

1 2

**Credit Card Fraud Detection**

What conditional probabilities are given? What others are important for business?

Given: P(Flagged fraud | Fraud) = 0.99

P(Flagged fraud | Honest) = 0.01

Other important conditional probabilities:

P(Fraud | Flagged fraud)

P(Honest | Not flagged fraud)

**Credit Card Fraud Detection**

Suppose 1% of the transactions at the e-commerce site are fraudulent. What are the chances that the transaction is honest given the system has incorrectly flagged it as fraudulent?

To calculate:

P(Honest | Flagged fraud) when P(Fraud) = 0.01

$$P(Honest|Flagged\ fraud) = \frac{P(Honest) * P(Flagged\ fraud|Honest)}{P(Honest) * P(Flagged\ fraud|Honest) + P(Fraud) * P(Flagged\ fraud|Fraud)}$$

$$= \frac{0.99 * 0.01}{0.99 * 0.01 + 0.01 * 0.99} = \frac{1}{2} = 0.50$$

# Supervised Machine Learning

**Credit Card Fraud Detection**

Suppose 5% of the transactions at the e-commerce site are fraudulent. What are the chances that the transaction is honest given the system has incorrectly flagged it as fraudulent?

To calculate:

P(Honest | Flagged fraud) when P(Fraud) = 0.05

$$P(Honest|Flagged\ fraud)$$
$$= \frac{P(Honest) * P(Flagged\ fraud|Honest)}{P(Honest) * P(Flagged\ fraud|Honest) + P(Fraud) * P(Flagged\ fraud|Fraud)}$$
$$= \frac{0.95 * 0.01}{0.95 * 0.01 + 0.05 * 0.99} = \frac{0.0095}{0.059} = 0.16$$

# Supervised Machine Learning

**Credit Card Fraud Detection**

What is your evaluation of the system? Is it adequate for the e- commerce site's needs?

If fraud is rare (1% or less), too many honest transactions are labeled as fraud. This will be disastrous for the company.

At higher fraud rates (5% or more), the system's performance may be acceptable, depending on size of transactions and costs involved in contacting annoyed customers and taking measures to retain them.

**How good is your classification matrix?**

**Buyer or non-buyer**

A retail store's marketing team uses analytics to predict who is likely to buy a newly introduced high-end (read "expensive") product. Indicate which measure is more important for the business to track and explain why. Calculate other measures also.

| Buyer or Not | | Actual | | Total |
|---|---|---|---|---|
| | | Negative | Positive | |
| Predicted | Negative | 725 | 158 | 883 |
| | Positive | 75 | 302 | 377 |
| Total | | 800 | 460 | 1260 |

# Supervised Machine Learning

| Buyer or Not | | Actual | | Total |
|---|---|---|---|---|
| | | Negative | Positive | |
| Predicted | Negative | 725 | 158 | 883 |
| | Positive | 75 | 302 | 377 |
| Total | | 800 | 460 | 1260 |

**Confusion Matrix**

Buyer or Non-Buyer

State/Calculate:

TP = 302 TN = 725 FP = 75 FN = 158

Should the business be more worried about FP or FN or equally worried about both of them? Why?

FN. If the model predicts that the person will not buy, the product will not be marketed to him/her, and the business will lose...er, business.

FP is not such a big worry since only the cost of a phone call, SMS or sending a catalog will be lost.

## Confusion Matrix

| Buyer or Not | | Actual: Negative | Actual: Positive | Total |
|---|---|---|---|---|
| Predicted | Negative | 725 | 158 | 883 |
| | Positive | 75 | 302 | 377 |
| Total | | 800 | 460 | 1260 |

What is more important: Recall, Precision or Accuracy?

| | | Predicted: Positive | Predicted: Negative | |
|---|---|---|---|---|
| Actual | Positive | True +ve | False –ve | Recall/Sensitivity/True Positive Rate (Minimize False –ve) |
| | Negative | False +ve | True –ve | Specificity/True Negative Rate (Minimize False +ve) |
| | | Precision | | Accuracy, $F_1$ score |

## Confusion Matrix

| Buyer or Not | | Actual | | Total |
|---|---|---|---|---|
| | | Negative | Positive | |
| Predicted | Negative | 725 | 158 | 883 |
| | Positive | 75 | 302 | 377 |
| Total | | 800 | 460 | 1260 |

What is more important: Recall, Precision or Accuracy?

$$Recall\ (or\ Sensitivity) = \frac{TP}{TP + FN} = \frac{302}{460} = 65.6\%$$

$$Precision = \frac{TP}{TP + FP} = \frac{302}{377} = 80.1\%$$

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} = \frac{1027}{1260} = 81.5\%$$

$$Specificity = \frac{TN}{TN + FP} = \frac{725}{800} = 90.6\%$$

$$F_1 = \frac{2 * Recall * Precision}{Recall + Precision} = \frac{2 * 0.656 * 0.801}{0.656 + 0.801} = 72.1\%$$

# Supervised Machine Learning

## Sensitivity - Specificity

| TEST NAME | TECHNOLOGY | VALUE | UNITS |
|---|---|---|---|
| **ANTI CCP (ACCP)** | E.L.I.S.A | 0.48 | OD Ratio |

**Reference Range :**

Negative : < 0.80
Equivocal: 0.80 - 1.20
Positive : > 1.20

Clinical Significance :

Anti-Cyclic-Citrullinated-Peptide (Anti-CCP) Antibodies hold promise for early and more accurate detection of Rheumatoid Arthritis before the disease proceeds into an irreversible damage.

Analytical Specifications :

Anti-Cyclic-Citrullinated-Peptide (Anti-CCP) antibodies are detected using a solid phase enzyme immuno assay having an analytical sensitivity of 1.0 U/ml. No cross reactivity to other auto antigen is found. Sensitivity of the method is 68% and specificity is 92%.

**Method :** SOLID PHASE CAPTURE ENZYME IMMUNOASSAY

| | | | |
|---|---|---|---|
| **ANTI NUCLEAR ANTIBODIES (ANA)** | E.L.I.S.A | 0.29 | OD Ratio |

**Reference Range :**

Negative < 0.80
Equivocal 0.8 – 1.20
Positive > 1.20

# Supervised Machine Learning

**Framingham Heart Study**

A Project of the National Heart, Lung, and Blood Institute and Boston University

Case – Framingham Heart Study

Committed to identifying common factors contributing to cardiovascular disease (CVD).

Setup in the town of Framingham, MA in 1948

Random sample consisting of 2/3rds of adult population in the town

| AGE-SEX DISTRIBUTION AT ENTRY (1948) | | | | |
|---|---|---|---|---|
| **Age** | **29-39** | **40-49** | **50-62** | **Totals** |
| Men | 835 | 779 | 722 | 2,336 |
| Women | 1,042 | 962 | 869 | 2,873 |
| Totals | 1,877 | 1,741 | 1,591 | 5,209 |

# Supervised Machine Learning

**Case Study: Data (framinghamheartstudy.org & MITx)**

5209 men and women participated

Age range: 30-62

People who had not yet developed overt symptoms of CVD or suffered a heart attack or stroke

Careful monitoring of Framingham Study population has led to identification of major CVD risk factors

Led to development of Framingham Risk Score, a gender specific algorithm used to estimate the 10-year cardiovascular risk of an individual

http://cvdrisk.nhlbi.nih.gov/

**Case Study** – Predicting Coronary Heart Disease (CHD)

Data description 4240 observations; 15 predictor and 1 predicted variables

TenYearCHD – To be predicted. Risk of having a heart attack or stroke in the next 10 years.

Predictors

Demographic Risk Factors

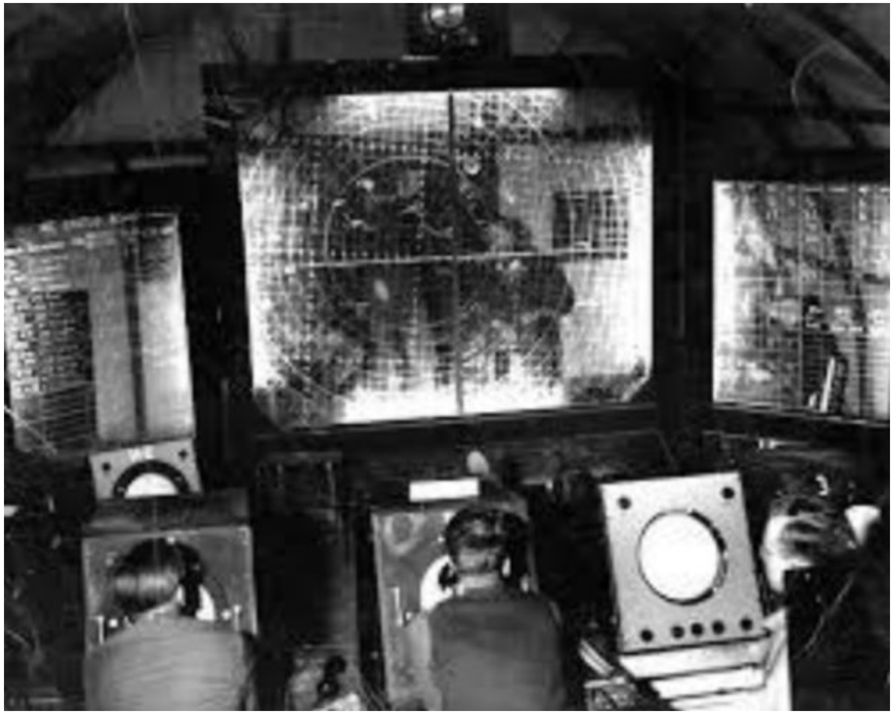
– male: Gender of subject – Yes or No

– age: Age of subject at first examination

– education: some high school (1), high school (2), some college/vocational college (3), college (4)

**ROC Curves and AUC**

ROC – Receiver Operating Characteristics
AUC – Area Under the ROC Curve



**ROC Curves and AUC**

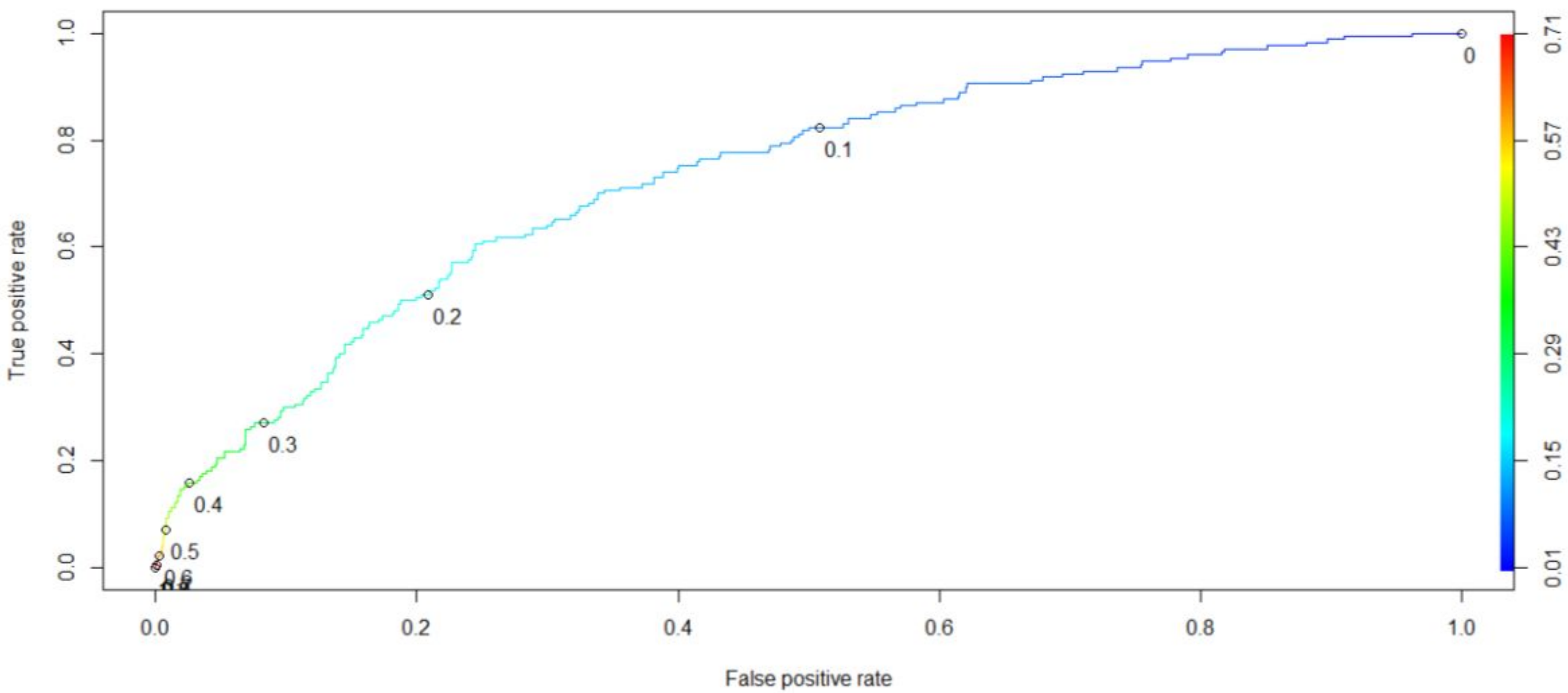
Logistic regression gives Probability forecasts for the given data point to be in a given bucket. • A threshold needs to be chosen to finally translate this probability to a bucket allocation

At a given threshold, we can evaluate the classification accuracy (accuracy, sensitivity, recall, kappa etc)

ROC curve tries to evaluate how well the regression has achieved the separation between the classes at all threshold values

**ROC Curves and AUC**

ROC – Plot of True Positive Rate vs False Positive Rate, i.e., Sensitivity vs 1-Specificity



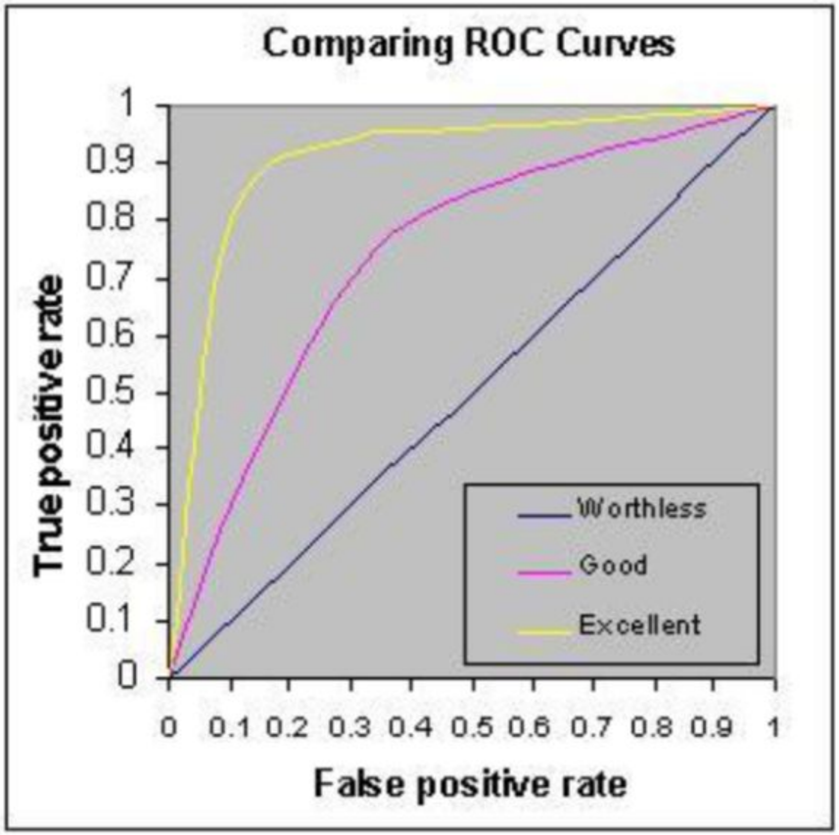
**ROC Curves and AUC**

AUC – Measures discrimination, i.e., ability to correctly classify those with and without CHD

If you randomly pick one person who HAS CHD and one who DOESN'T and run the model, the one with the higher probability should be from the high risk group

AUC is the percentage of randomly drawn such pairs for which the classification is done correctly

## ROC Curves and AUC




Rough rule of thumb:

- 0.90 -1.0 = Excellent
- 0.80 – 0.90 = Good
- 0.70 – 0.80 = Fair
- 0.60 – 0.70 = Poor
- 0.50 – 0.60 = Fail

- <0.50 – You are better off doing a coin toss than working hard to build a model ☺

**ROC Curve Demo**

http://www.navan.name/roc/

https://youtu.be/OAl6eAyP-yo

**Kappa Metric**

Accuracy can often be a misleading metric, when one category occurs more often than other in the given data-set

e.g: Occurrence of cancer in general population is 0.4%

If a prediction system blindly marks everyone as “No cancer”, it will 99.6% accurate

**Kappa Metric**

Quantifies how accurate the prediction algorithm is when compared to a random prediction

$$kappa = \frac{totalAccuracy - randomAccuracy}{1 - randomAccuracy}$$

$$\text{totalAccuracy} = \frac{CorrectPredictions}{Total}$$

$$\text{randomAccuracy} = \frac{ActualFalse}{Total} * \frac{PredictedFalse}{Total} + \frac{ActualTrue}{Total} * \frac{PredictedTrue}{Total}$$

| Kappa Value | |
|---|---|
| <0 | No agreement |
| 0-0.2 | Slight |
| 0.21 to 0.4 | Fair |
| 0.4 to 0.6 | Moderate |
| 0.6 to 0.8 | Substantial |
| 0.8 to 1 | Almost Perfect |

**Kappa Metric**

| 10-year CHD risk | | Predicted | |
|---|---|---|---|
| | | True | False |
| Actual | True | 30 | 357 |
| | False | 9 | 2170 |

Total= 30+357+9+2170=2566

TotalAccuracy=(30+2170)/2566=0.857

PercTrue=(30+357)/2566 = 0.15; PercFalse=(9+2170)/2566 = 0.85

PredTrue=(30+9)/2566=0.015; PredFalse=(357+2170)/2566 = 0.985

randomAccuracy= 0.15*0.015 + 0.85*0.985 = 0.84

$$\text{Kappa} = \frac{TotalAccur - randomAccur}{1 - randomAccur} = \frac{0.857 - 0.84}{1 - 0.84} = 0.10$$

Slightly better than random!

# Supervised Machine Learning

Actuarial Statistics is a huge industry with market potential of over ~200 Billion/year. LIC has released a new product (“Yuva Jeevan”) and has hired you as Machine Learning Engineer to predict the average claim per year. We have a single attribute (age). How do we go about this?

# Supervised Machine Learning

**Demo**

**1000 rows X 2 columns**
**No missing values**

| | Age | Average Claims per Year (Rupees) |
|---|---|---|
| **0** | 25 | 4500 |
| **1** | 30 | 5000 |
| **2** | 35 | 6000 |
| **3** | 40 | 8000 |
| **4** | 45 | 11000 |
| **5** | 50 | 15000 |
| **6** | 55 | 20000 |
| **7** | 60 | 30000 |
| **8** | 65 | 50000 |
| **9** | 70 | 100000 |

**http://localhost:8888/notebooks/Downloads/DT_RF_Regression.ipynb**

# Supervised Machine Learning

**Decision Tree (Excercises):**

Lab_1  Decision Tree based model to predict class of wines

**Sol:** DecisionTree_Wine.ipynb

Lab_2  Build a decision tree model for Pima Diabetes dataset to predict diabetic cases

**Notes:**

1. Its loved by the industry! (Because?)
2. An ensemble method of Decision Trees called Random Forest can model most tabular data incredibly well so much so that some courses only teach Random Forest
   Source: [www.fast.ai](www.fast.ai)

# Supervised Machine Learning

## Thank you! Questions?

Jayanth Rasamsetti
Founder & Chief Scientist
www.sgmoid.com
Columbia University (MS)
IIT Madras (B.Tech & M.Tech)
jr3281@columbia.edu
https://www.linkedin.com/in/jayanthr/