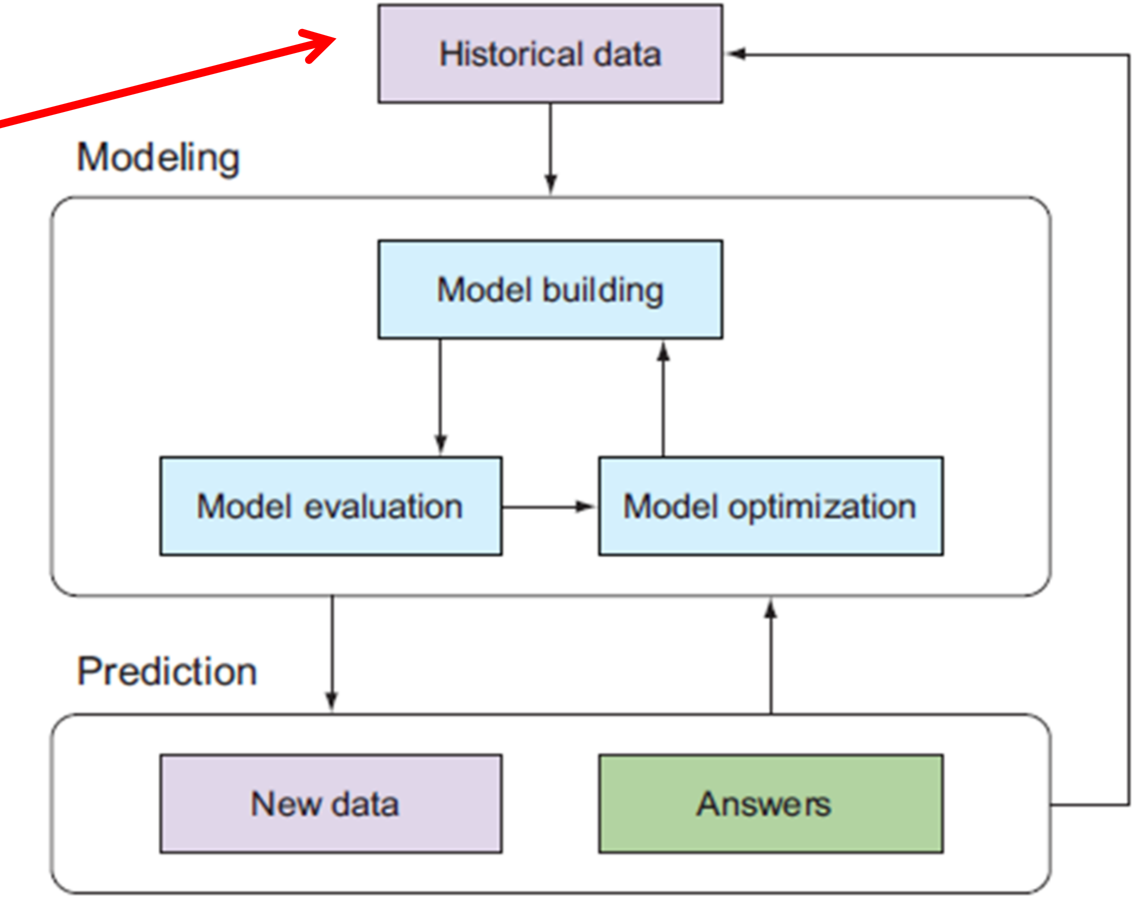
# GETTING STARTED WITH MACHINE LEARNING

# RECAP ABOUT MACHINE LEARNING

**QUIZ?? OR NOT?**

- What it is?
- How does ML works?
- What are the ML workflows?
- What are the ML framework?

# MACHINE LEARNING BASIC WORK FLOW



# GETTING STARTED: DATA COLLECTION

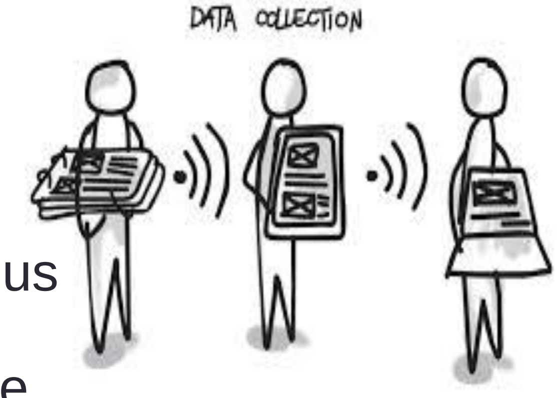
How?

1. Identify (assess) its instances of interest (one or several)

   Example: instances can be; people (which of my customer will churn this month), events (is the sentiment of this tweet will be positive, negative or neutral), or even periods of time (what will be my demand of my product next month?).

2. Identify its target of interest

   Example: binary (churn versus not churn, fraud versus not fraud), multiple classes (negative versus positive versus neutral), or even hundreds or thousands of classes (picking a song out of a large library) or numerical values (product demand).



# GETTING STARTED: DATA COLLECTION

3. Identify its historical data in which the target is known

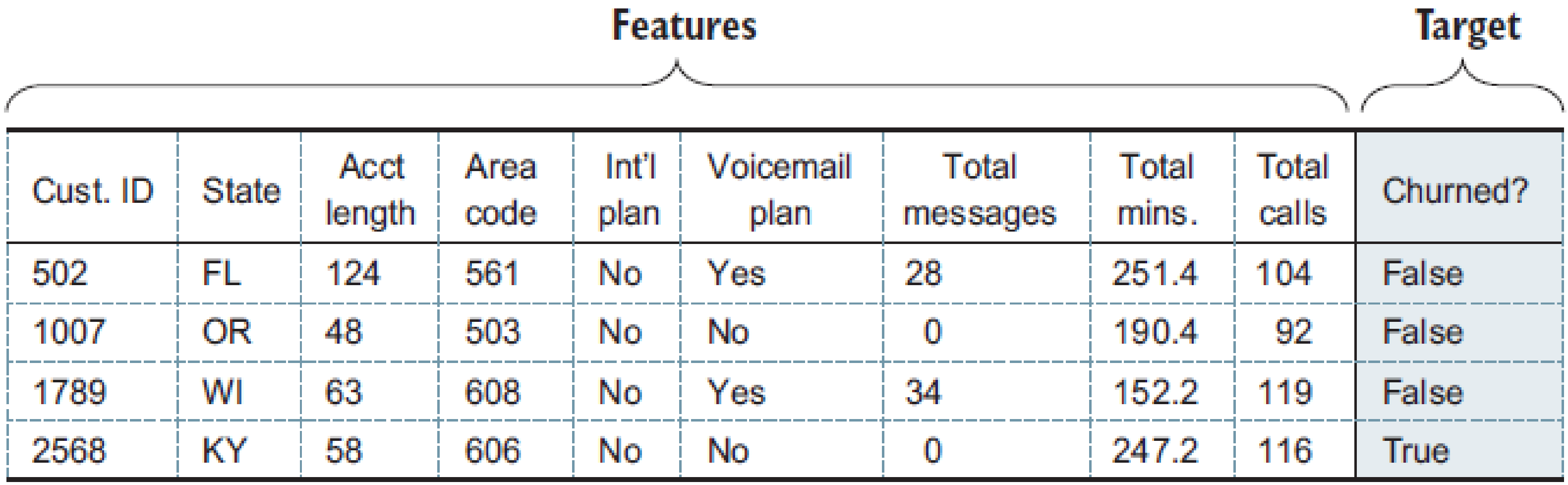
   Example: over weeks or months of data collection, determine which of your subscribers churned and which people clicked your ads.

   Historical data files will contain information about each instance that’s knowable at the time of prediction.

   These are **input features** (also commonly referred to as the *explanatory* or *independent variables*).

4. Identify its implied action if the target were knowable

   Example: if you knew that a user would click your ad, you would bid on that user and serve the user an ad.



| Features | | | | | | | | | Target |
|---|---|---|---|---|---|---|---|---|---|
| Cust. ID | State | Acct length | Area code | Int'l plan | Voicemail plan | Total messages | Total mins. | Total calls | Churned? |
| 502 | FL | 124 | 561 | No | Yes | 28 | 251.4 | 104 | False |
| 1007 | OR | 48 | 503 | No | No | 0 | 190.4 | 92 | False |
| 1789 | WI | 63 | 608 | No | Yes | 34 | 152.2 | 119 | False |
| 2568 | KY | 58 | 606 | No | No | 0 | 247.2 | 116 | True |

Figure: Training data with four instances for the telecom churn problem

## YOU MIGHT BE ASKING:

- "Which input features should I include?"
- "How do I obtain known values of my target variable?"
- "How much training data do I need?"
- "How do I know if my training data is good enough?"

# “Which input features should I include?”

RULE # 1:

“features should be included only if they’re suspected to be related to the target variable.”

Why? (RULE # 2):

- The more uninformative features are present, the lower the signal-to-noise ratio and thus the less accurate (on average) the ML model will be.

The practical steps:

1. Include all the features that you suspect to be predictive of the target variable. Fit an ML model. If the accuracy of the model is sufficient, stop.
2. Otherwise, expand the feature set by including other features that are less obviously related to the target. Fit another model and assess the accuracy. If performance is sufficient, stop.
3. Otherwise, starting from the expanded feature set, run an ML *feature selection algorithm* to choose the best, most predictive subset of your expanded feature set.

# “How do I obtain known values of my target variable?”

“collection of instances of known target variables can be painful (both in terms of money and time)”

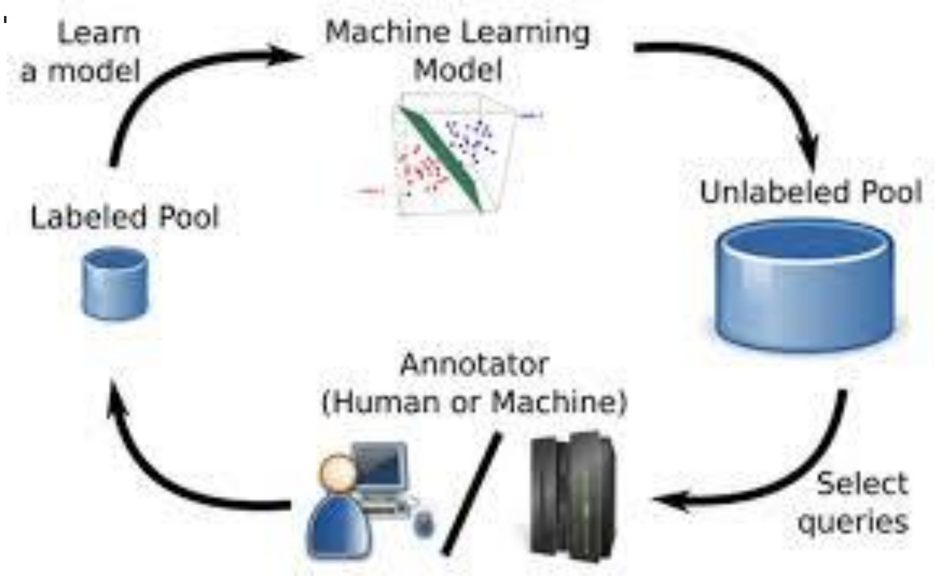
Ways of obtaining ground-truth values of the target variable: → Labor intensive

1. Dedicating analysts to manually look through past or current data to determine or estimate the ground-truth values of the target
2. Using crowdsourcing to use the “wisdom of crowds” in order to attain estimates of the target
3. Conducting follow-up interviews or other hands-on experiments with customers
4. Running controlled experiments (for example, A/B tests) and monitoring the responses

# “How do I obtain known values of my target variable?”

Other possible way: Active Learning (by Dasgupta. See http://videolectures.net/icml09_dasgupta_langford_actl/ ).

Active learning will identifies the subset of instances from the latter set whose inclusion in the training set would yield the most accurate ML model.



# “How much training data do I need?”

Factors determine the amount of training data needed:

- The complexity of the problem.

  Does the relationship between the input features and target variable follow a simple pattern, or is it complex and nonlinear?

- The requirements for accuracy.

  If you require only a 60% success rate for your problem, less training data is required than if you need to achieve a 95% success rate.

- The dimensionality of the feature space.

  If only two input features are available, less training data will be required than if there were 2,000 features.

# “How much training data do I need?”

Guiding principle:

“as the training set grows, the models will (on average) get more accurate.”

Why?:

“ The relationship between the features and target is learned entirely from the training data, the more you have, the higher the model’s ability to recognize and capture more-subtle patterns and relationships.”

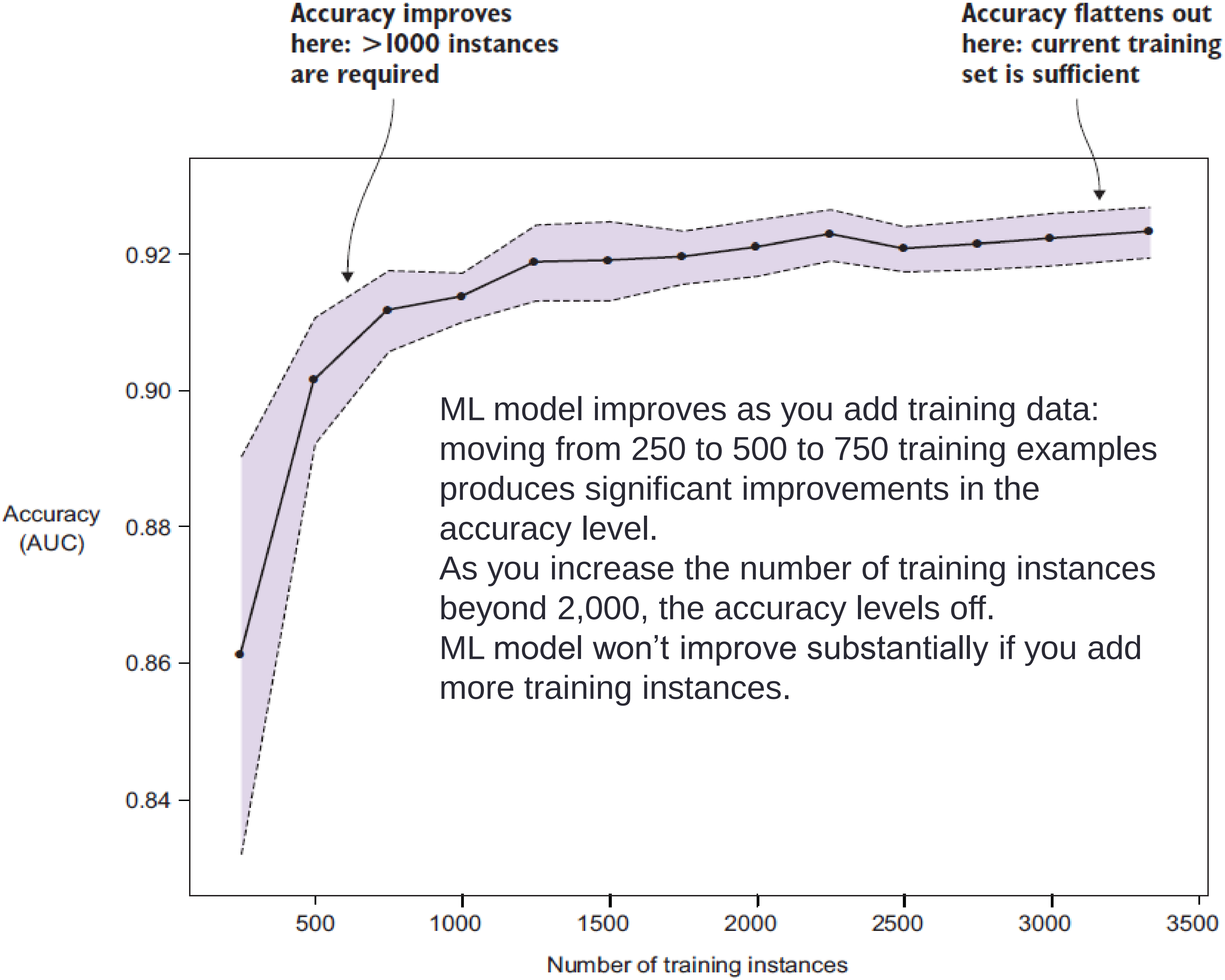
EXAMPLE:
Problem → Telco Customer Churn Prediction Analysis
Data Set → 3,333 instances, each containing 19 features plus the binary outcome of unsubscribed versus renewed.

**POSSIBLE TRAINING DATA STRATEGY:**

1. Using the current training set, choose a grid of subsample sizes to try. For example, with 3,333 instances of training data, your grid could be 500; 1,000; 1,500; 2,000; 2,500; 3,000.
2. For each sample size, randomly draw that many instances (without replacement) from the training set.
3. With each subsample of training data, build an ML model and assess the accuracy of that model.
4. Assess how the accuracy changes as a function of sample size. If it seems to level off at the higher sample sizes, the existing training set is probably sufficient. But if the accuracy continues to rise for the larger samples, the inclusion of more training instances would likely boost accuracy.

OTHER TRAINING DATA STRATEGY:

Have a clear accuracy target:
➔ to assess whether that target has been fulfilled by the current ML model built on the existing training data



Accuracy improves here: >1000 instances are required

Accuracy flattens out here: current training set is sufficient

ML model improves as you add training data: moving from 250 to 500 to 750 training examples produces significant improvements in the accuracy level.

As you increase the number of training instances beyond 2,000, the accuracy levels off.

ML model won’t improve substantially if you add more training instances.

# “How do I know if my training data is good enough?”

Similar question:

“How similar are the instances in the training set to the instances that will be collected in the future?”

➔ your training sample should be representative of the target

A training set that consists of a non representative sample of what future data will look like is called *sample selection bias* or *covariate shift*.

# “How do I know if my training data is good enough?”

Some reasons which may cause your training sample non representative:

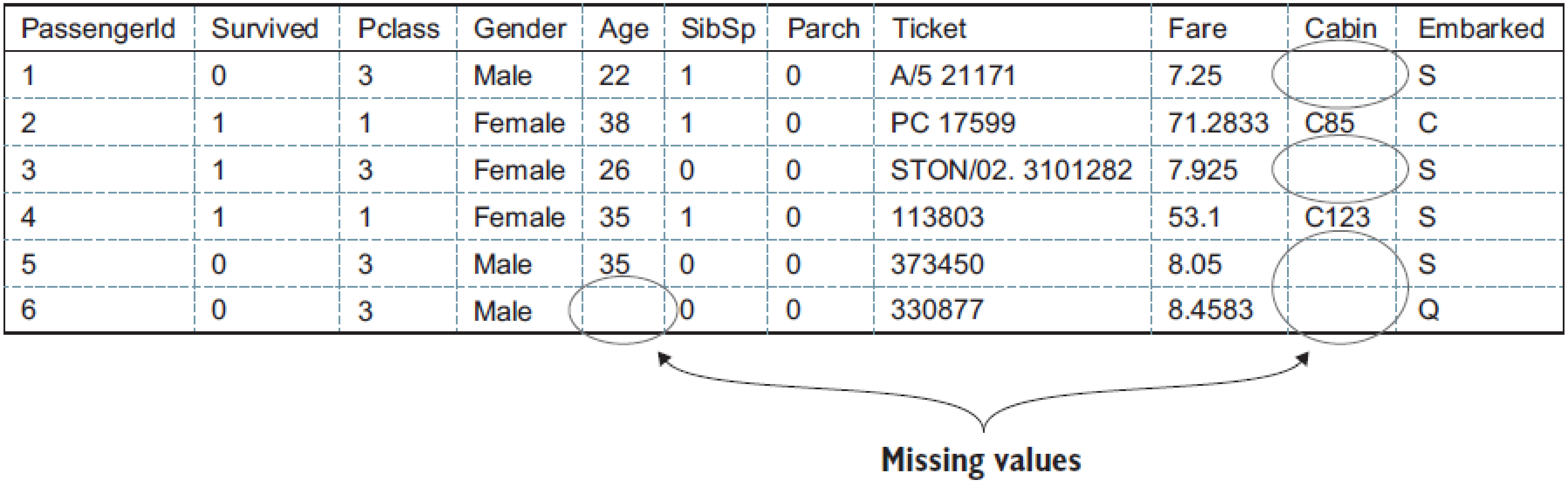
- It was possible to obtain ground truth for the target variable for only a certain.
- The properties of the instances have changed over time.
- The input feature set has changed over time. This change may require to modify the feature set used for the model and potentially discard old data from the training set.

# NEXT: PREPROCESSING DATA FOR MODELLING

Some cases require to:

1. rescale the numeric features to make them comparable or to bring them into line with a frequency distribution (for example, grading on the normal curve).
2. dealing with non-numeric features
3. dealing with missing values

   concept of replacing the missing value = imputation
4. data to be *normalized* ➔ meaning that each individual feature has been manipulated to reside on the same numeric scale.

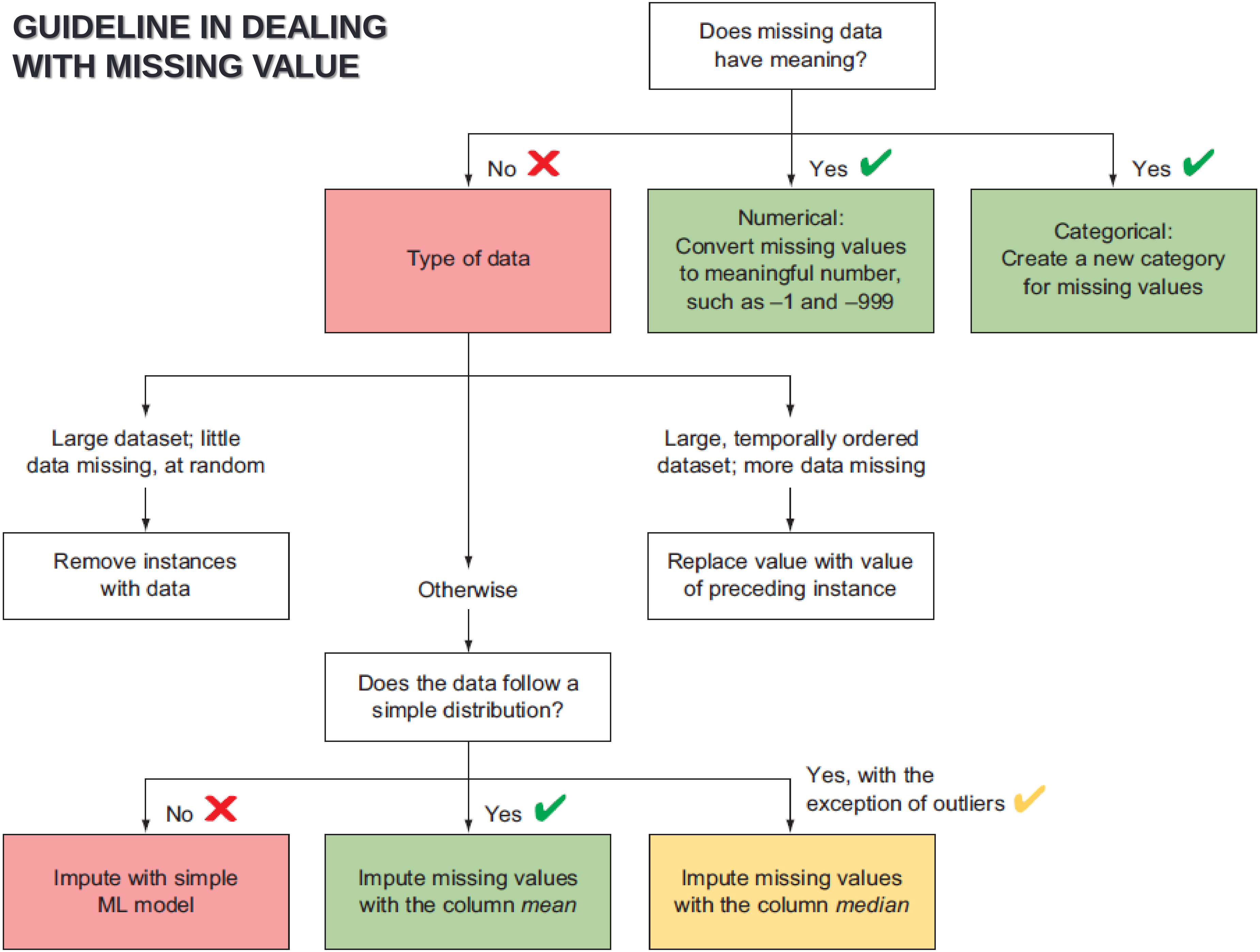


| PassengerId | Survived | Pclass | Gender | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 3 | Male | 22 | 1 | 0 | A/5 21171 | 7.25 |  | S |
| 2 | 1 | 1 | Female | 38 | 1 | 0 | PC 17599 | 71.2833 | C85 | C |
| 3 | 1 | 3 | Female | 26 | 0 | 0 | STON/02. 3101282 | 7.925 |  | S |
| 4 | 1 | 1 | Female | 35 | 1 | 0 | 113803 | 53.1 | C123 | S |
| 5 | 0 | 3 | Male | 35 | 0 | 0 | 373450 | 8.05 |  | S |
| 6 | 0 | 3 | Male |  | 0 | 0 | 330877 | 8.4583 |  | Q |

Missing values

## EXAMPLE OF TRAINING DATA SET WITH MISSING VALUE

# GUIDELINE IN DEALING WITH MISSING VALUE



Does missing data have meaning?

No ✗

Yes ✓

Yes ✓

Type of data

Numerical: Convert missing values to meaningful number, such as –1 and –999

Categorical: Create a new category for missing values

Large dataset; little data missing, at random

Large, temporally ordered dataset; more data missing

Remove instances with data

Otherwise

Replace value with value of preceding instance

Does the data follow a simple distribution?

No ✗

Yes ✓

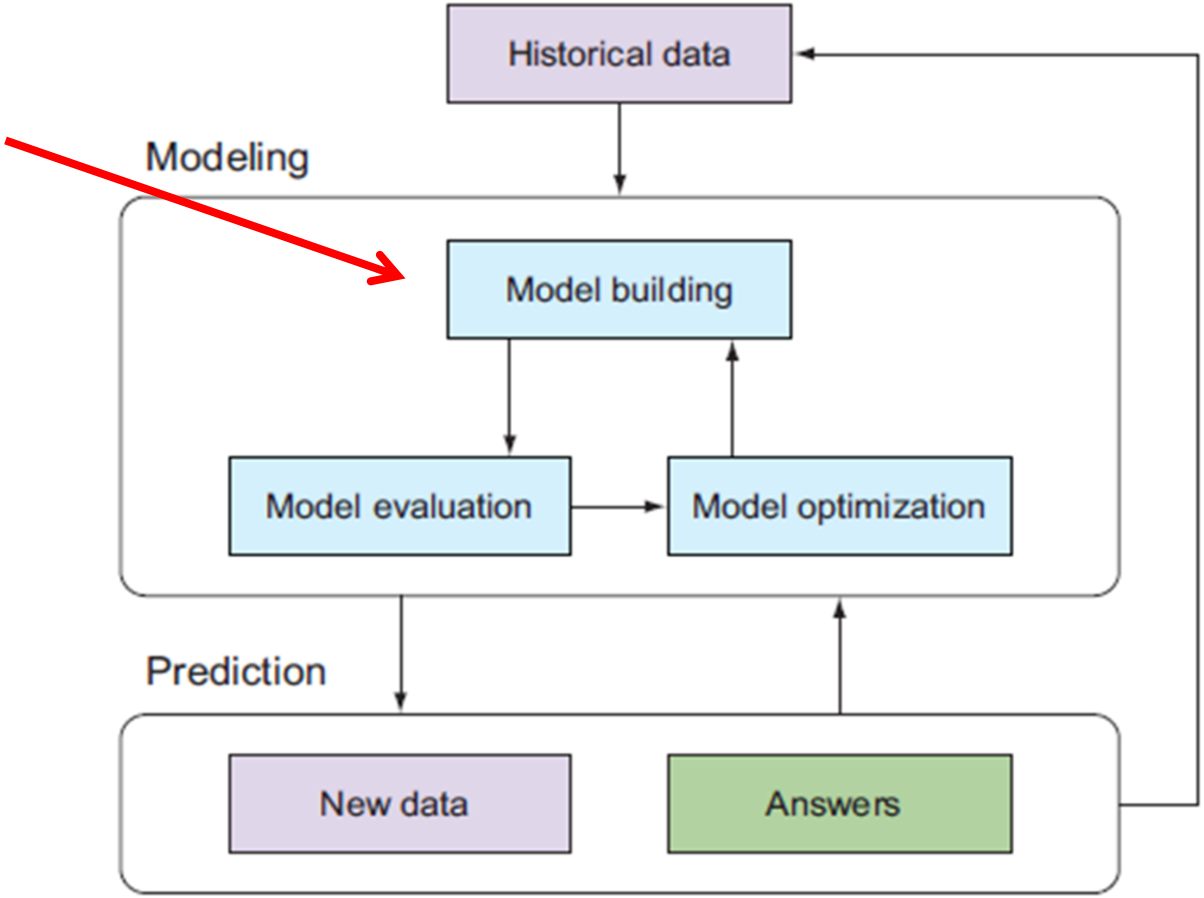
Yes, with the exception of outliers ✓

Impute with simple ML model

Impute missing values with the column *mean*

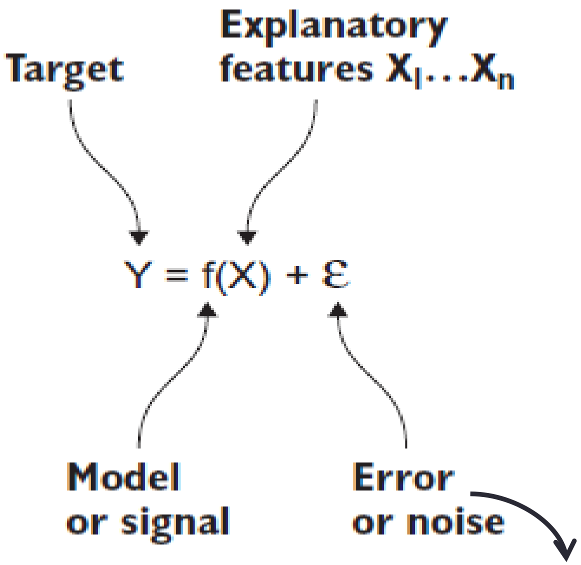
Impute missing values with the column *median*

# MACHINE LEARNING BASIC WORK FLOW



# MACHINE LEARNING MODELING

Remember the ML framework:

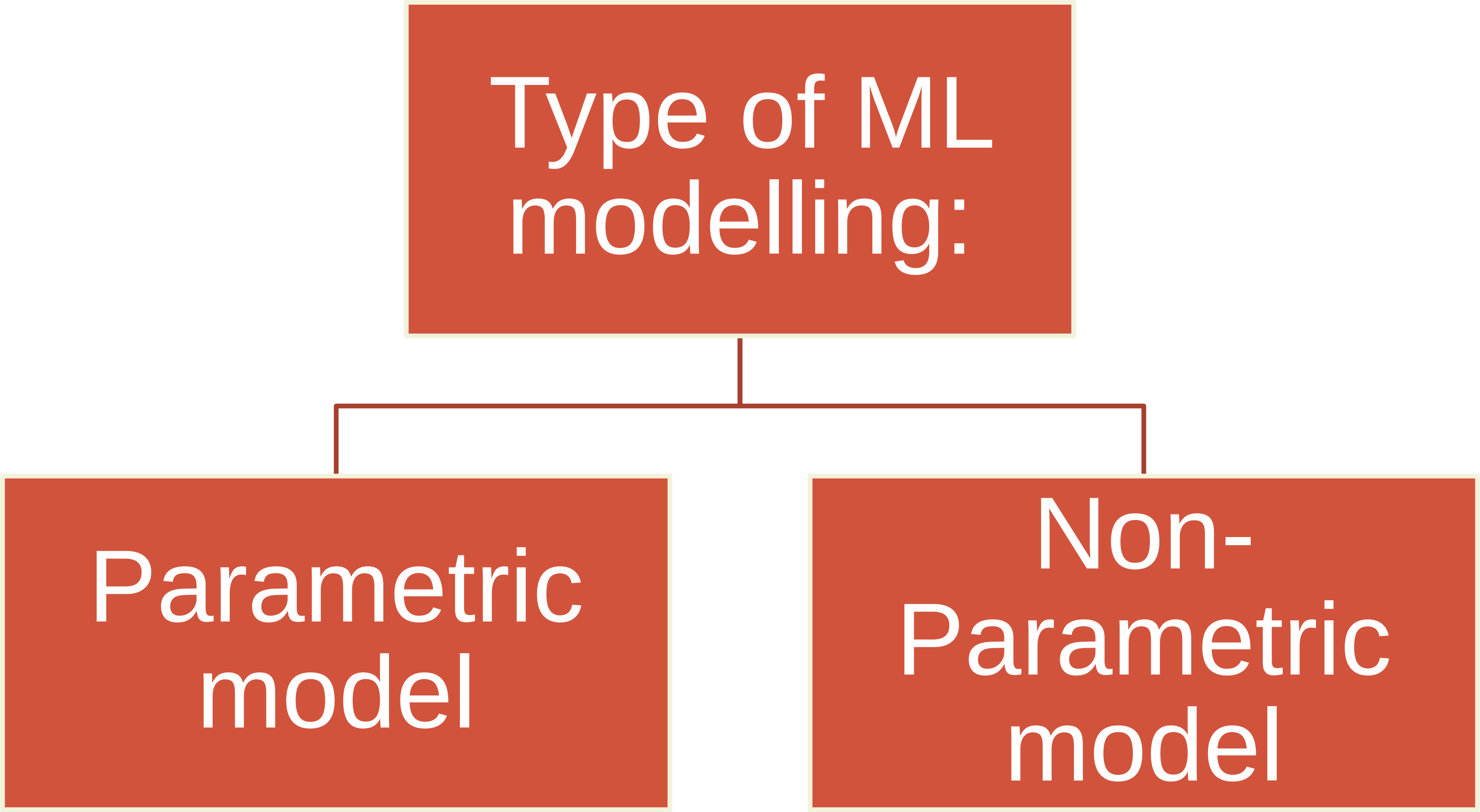


The goal of ML modeling is to accurately estimate *f* by *using data.*

*random noise in the data that's unrelated to the function f.*

The *f* function can be as prediction or inference

# MACHINE LEARNING MODELING



Type of ML modelling:

Parametric model

Non-Parametric model

# MACHINE LEARNING MODELING

- Parametric model: assume that *f* takes a specific functional form
  - tend to be simple and interpretable
  - simplest example: linear regression
  - However, real-world problems, *f* doesn’t assume such a simple form, especially when there are many input variables (X).
  - will fit the data poorly, leading to inaccurate predictions.
- Non-Parametric model: the form and complexity of *f* adapts to the complexity of the data
  - *f* doesn’t take a simple, fixed function
  - simplest example: classification, decision tree
  - others: k-nearest neighbors, splines, basis expansion methods, kernel smoothing, generalized additive models, neural nets, bagging, boosting, random forests, and support vector machines.

# GETTING STARTED WITH MACHINE LEARNING

SUPERVISED LEARNING

# RECAP ABOUT SUPERVISED LEARNING

**QUIZ?? OR NOT?**

- What it is?
- How does your algorithm (your ML) learns based on supervised learning?
- What example of problems (business use case) can implement supervised learning?

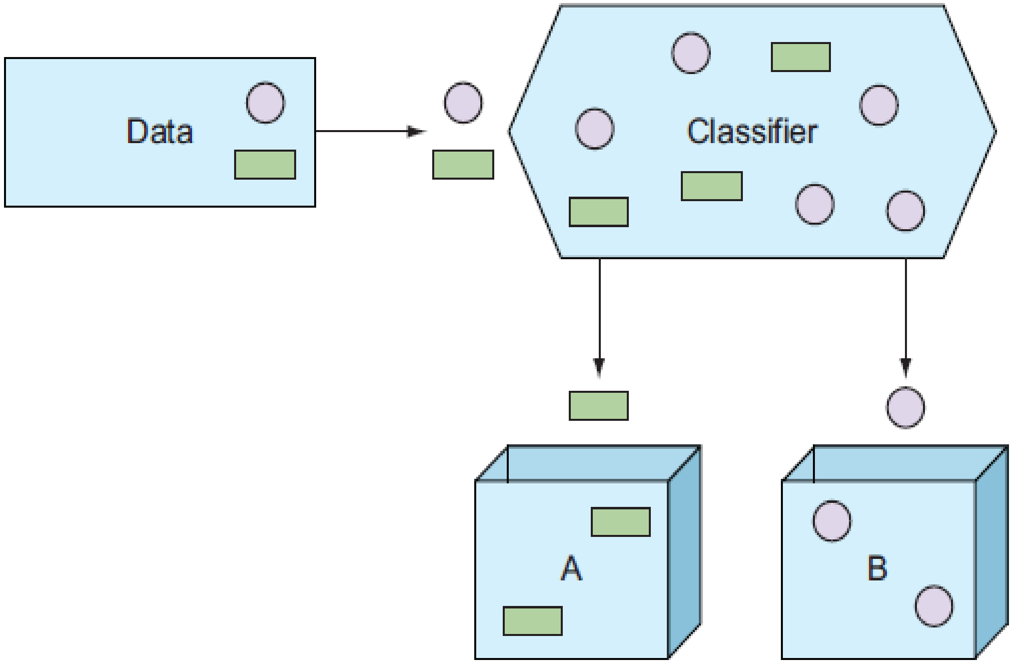
# CLASSIFICATION

“Many algorithms are available, and each has pros and cons for different data and deployment requirements.”

# CLASSIFICATION IN ML

**What it is?**

It describes the prediction of new data into buckets (classes) by using a classifier built by the machine-learning algorithm.



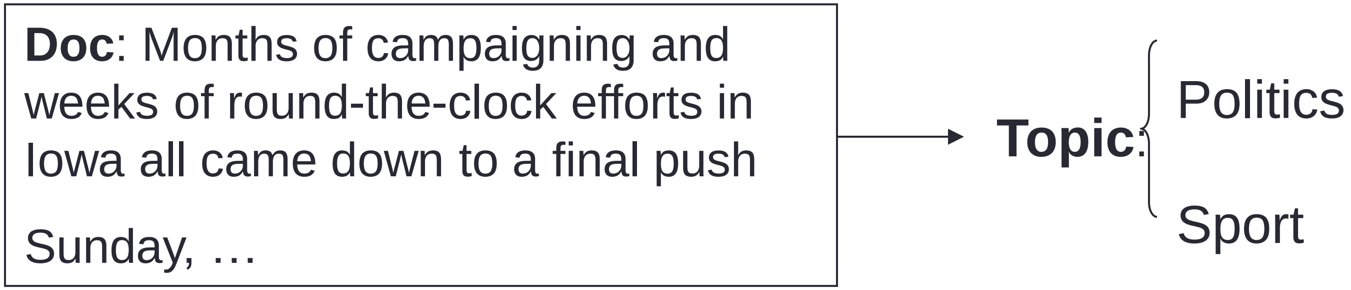
This is a case of binary classification with only two classes.

# CLASSIFICATION IN ML

- Given input: $\mathbf{x} = (x_1, x_2, \ldots, x_d)$
- Predict the output (class label) $y \in \mathcal{Y}$
  - Binary classification: $\mathcal{Y} = \{-1, +1\}$
  - Multi-class classification: $\mathcal{Y} = \{1, 2, \ldots, C\}$
- Learn a classification function: $f(\mathbf{x}) : \mathbb{R}^d \mapsto \mathcal{Y}$

# Examples of Classification Problem

- Text categorization:



**Doc**: Months of campaigning and weeks of round-the-clock efforts in Iowa all came down to a final push

Sunday, …

**Topic**:

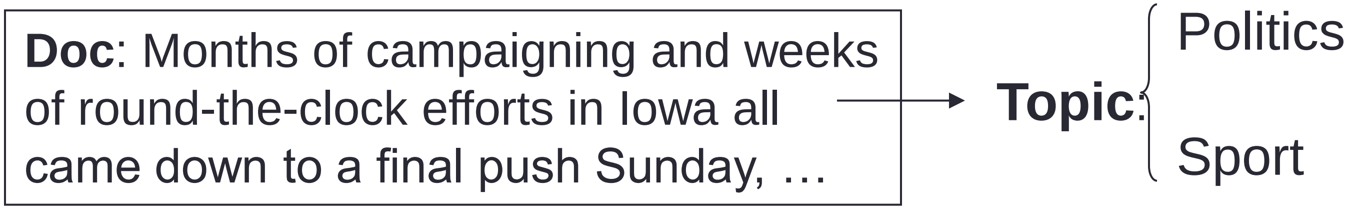
Politics

Sport

- Input features : ??
- Class label: ??

# Examples of Classification Problem

- Text categorization:



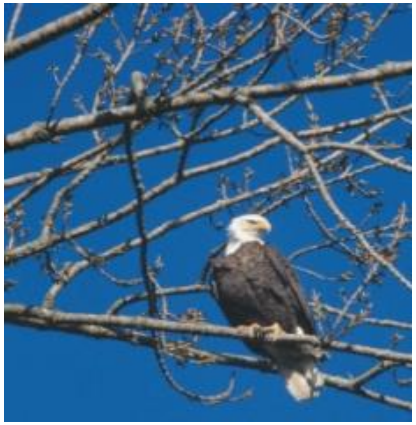
**Doc**: Months of campaigning and weeks of round-the-clock efforts in Iowa all came down to a final push Sunday, …

**Topic**: Politics / Sport

- Input features : $\mathbf{x}$
  - Word frequency
  - {(campaigning, 1), (democrats, 2), (basketball, 0), …}
- Class label: $y$
  - ‘Politics’: $y = +1$
  - ‘Sport’: $y = -1$

# Examples of Classification Problem

- Image Classification:



Which images have birds,
which one does not?

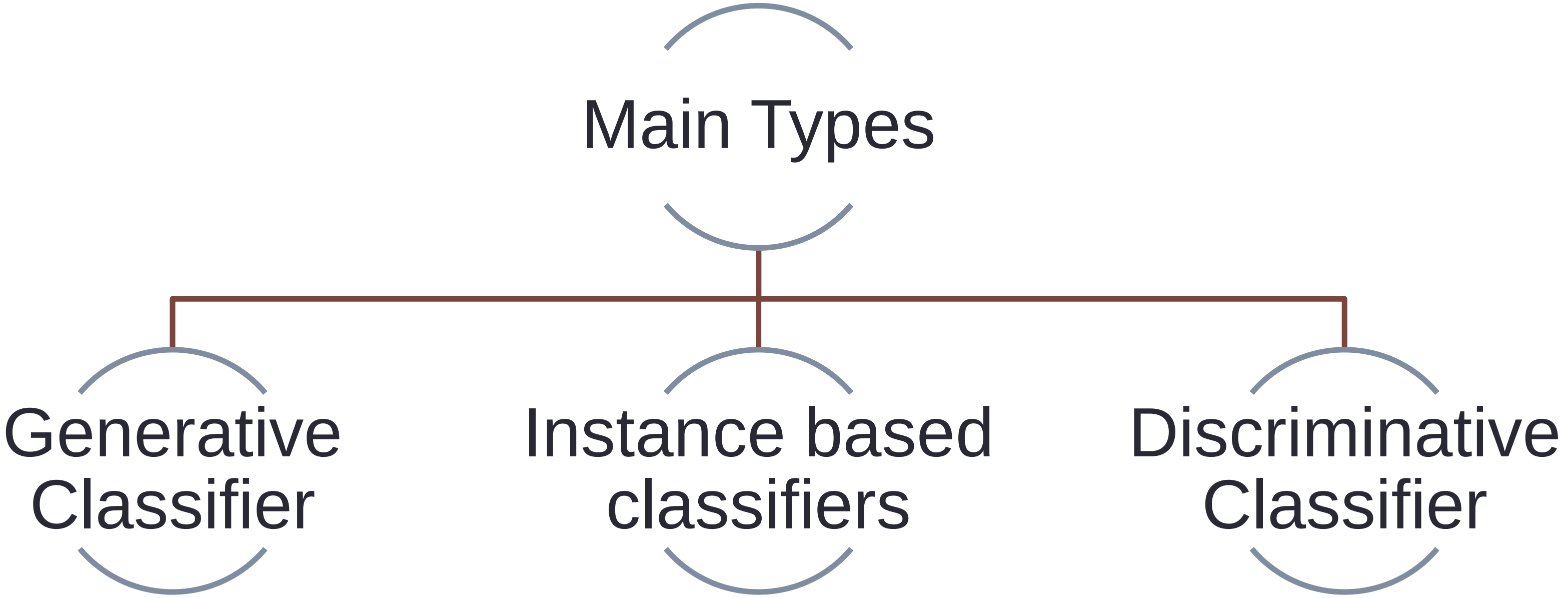
# Examples of Classification Problem

- Image Classification:

Which images are birds, which are not?

- Input features : $\mathbf{x}$
  - Color histogram
  - {(red, 1004), (blue, 23000), …}
- Class label : $y$
  - 'bird image': $y = +1$
  - 'non-bird image': $y = -1$

# Classification



Main Types

Generative Classifier

Instance based classifiers

Discriminative Classifier

- Build a generative statistical model
- e.g., Naïve Bayes

- Use observation directly (no models)
- e.g. K nearest neighbors (KNN)

- directly estimate a decision rule/boundary
- e.g., decision tree, Artificial Neural Network, Perceptron

# Generative Classifiers

- Tries to learn the model that generates the data behind the scenes by **estimating** the assumptions and distributions of the model.
- It then uses this to predict unseen data, because it assumes the model that was learned captures the real model.
- **Examples**
  - Naïve Bayes classifier
  - Bayesian network

# Generative Classifiers

How generative classifier works:

Problem – football players will play or not based on weather condition

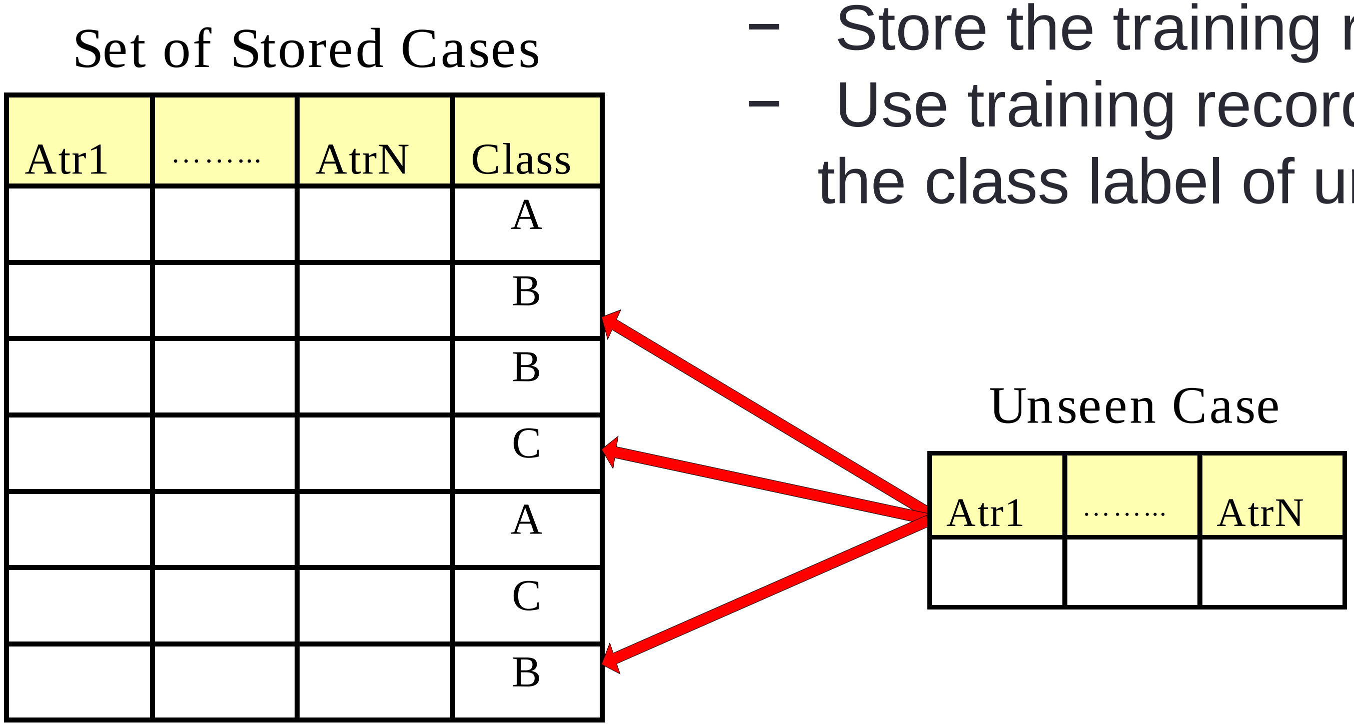
- Estimate from the data what is the probability of to play or not play football (X) given the weather is sunny (Y).
- Then, it estimates how many players will play football when the weather is sunny. **P(Y).

# Discriminative Classifier

- Tries to model by just depending on the observed data. It makes fewer assumptions on the distributions but depends heavily on the quality of the data
- **Examples**
  - Artificial Neural Network
  - Logistic regression
  - SVM
  - Boosted decision trees
- Discriminative classifiers outperform generative classifiers, if you have a lot of data.

# Instance-Based Classifiers (Lazy Learning)

- No model is learned
- Why??
  - Store the training records
  - Use training records to predict the class label of unseen cases



Set of Stored Cases

| Atr1 | ……… | AtrN | Class |
|---|---|---|---|
| | | | A |
| | | | B |
| | | | B |
| | | | C |
| | | | A |
| | | | C |
| | | | B |

Unseen Case

| Atr1 | ……… | AtrN |
|---|---|---|
| | | |

# Pros and Cons of Instances Based Learning

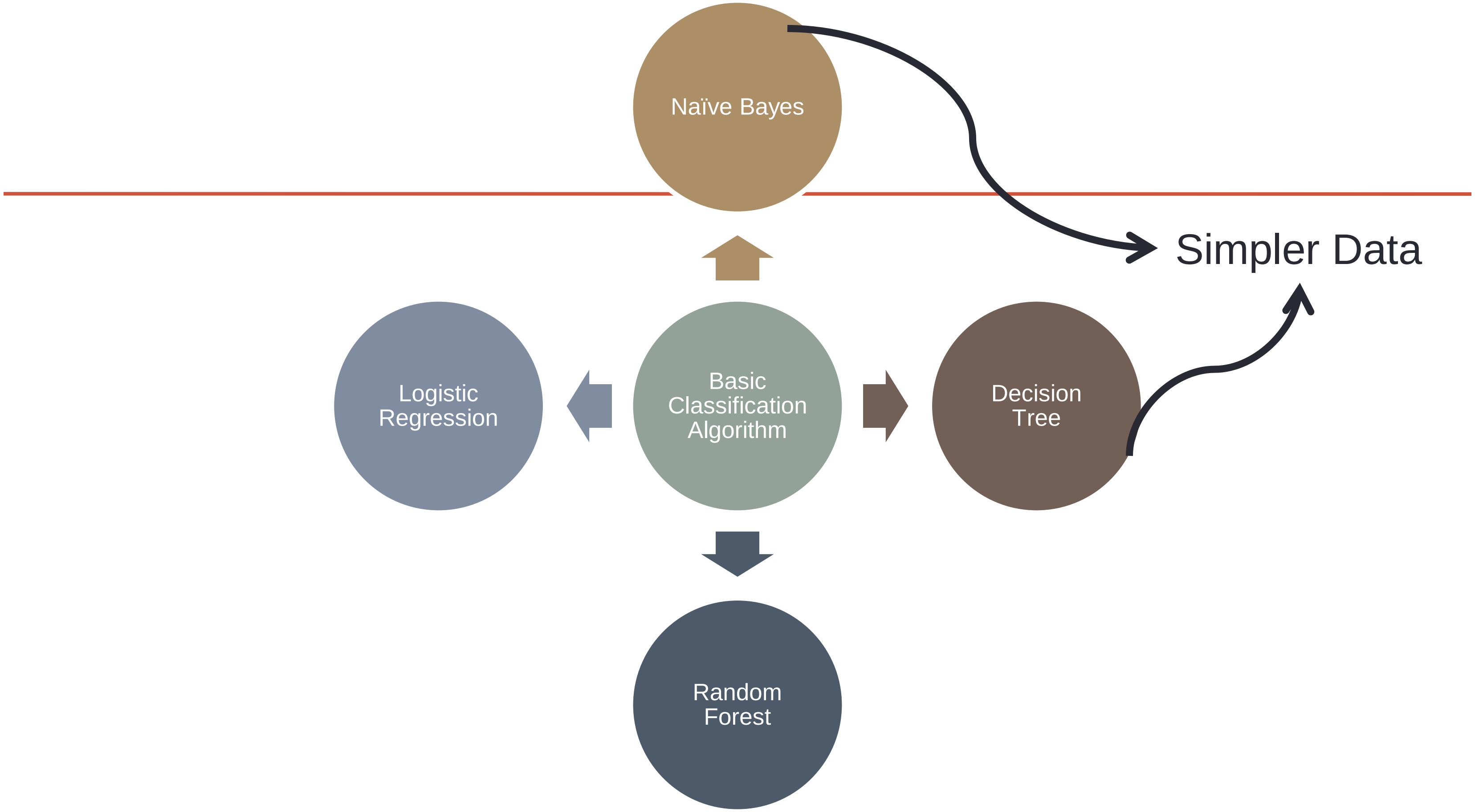
| Pros | Cons |
|---|---|
| – Can construct a different approximation to the target function for each distinct query instance to be classified.<br>– Can use more complex, symbolic representations | – Cost of classification can be high<br>– Uses all attributes (do not learn which are the most important) |

# Instance Based Classifiers

- Examples:
  - Rote-learner
    - Memorizes entire training data and performs classification only if attributes of record match one of the training examples exactly.
    - **Drawback:** Cannot classify a new  instance if it does match any training  example
  - Nearest neighbor
    - Uses k “closest” points (nearest neighbors) for performing classification

# BUILDING CLASSIFIER AND MAKING PREDICTION

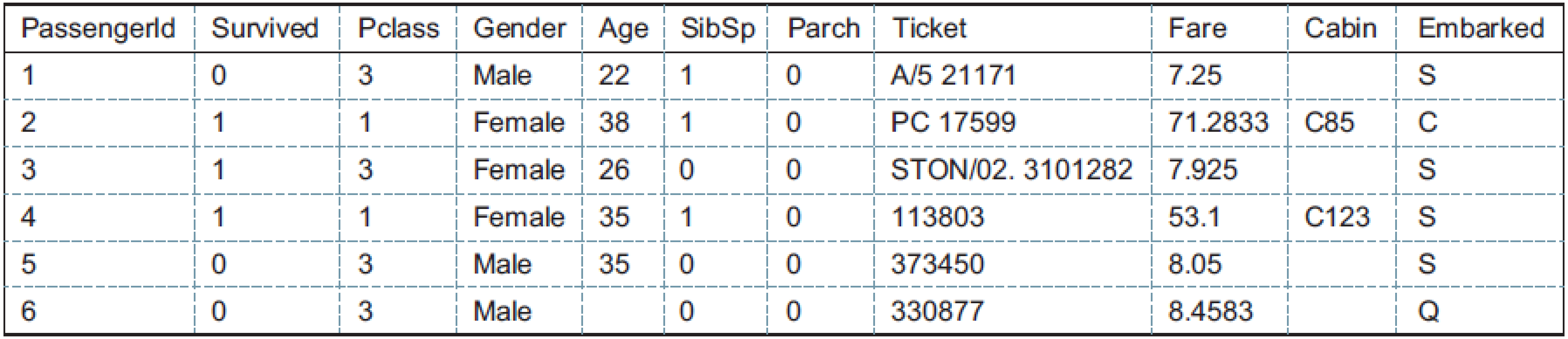
# “MANY ALGORITHMS ARE AVAILABLE, AND EACH HAS PROS AND CONS FOR DIFFERENT DATA AND DEPLOYMENT REQUIREMENTS.”



# BUILDING CLASSIFIER AND MAKING PREDICTION

## Example: Titanic Survival Model

- Classify between passenger which is not survive and survive
- Data set



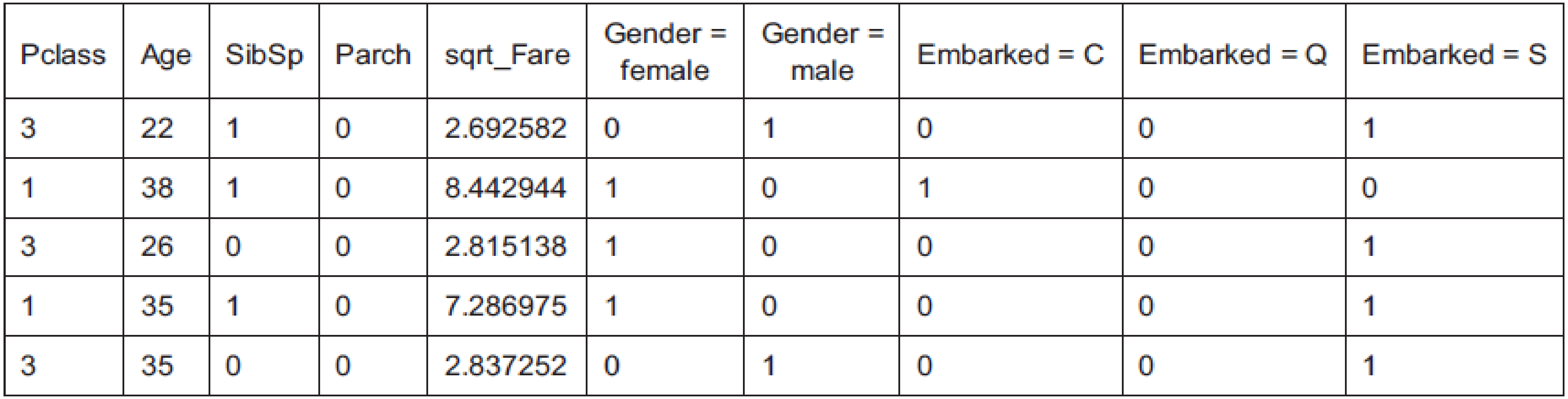
| PassengerId | Survived | Pclass | Gender | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 3 | Male | 22 | 1 | 0 | A/5 21171 | 7.25 |  | S |
| 2 | 1 | 1 | Female | 38 | 1 | 0 | PC 17599 | 71.2833 | C85 | C |
| 3 | 1 | 3 | Female | 26 | 0 | 0 | STON/02. 3101282 | 7.925 |  | S |
| 4 | 1 | 1 | Female | 35 | 1 | 0 | 113803 | 53.1 | C123 | S |
| 5 | 0 | 3 | Male | 35 | 0 | 0 | 373450 | 8.05 |  | S |
| 6 | 0 | 3 | Male |  | 0 | 0 | 330877 | 8.4583 |  | Q |

- Algorithm: Logistic Regression for Titanic Survival Model

# BUILDING CLASSIFIER AND MAKING PREDICTION

- Data Preprocessing: Missing value, Categorical variable and transform Fare by taking the square root (because of heavily skewed) to reduce the potentially harmful impact of outliers.

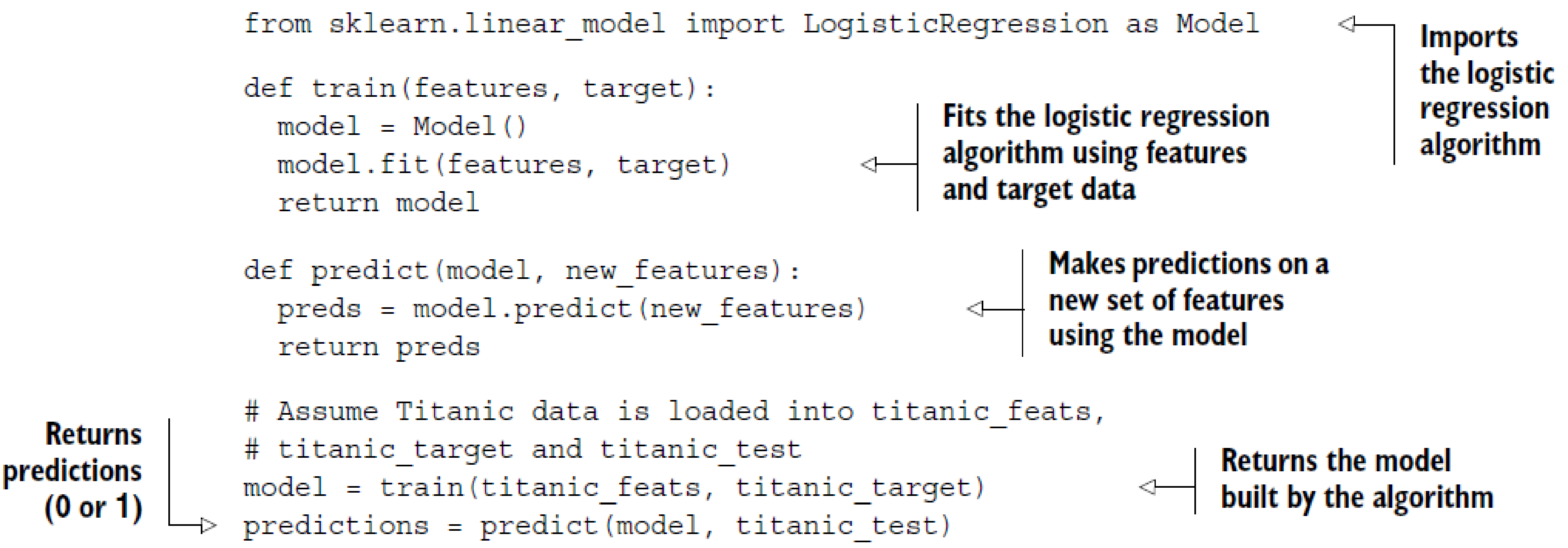
After data Pre-Processing:



| Pclass | Age | SibSp | Parch | sqrt_Fare | Gender = female | Gender = male | Embarked = C | Embarked = Q | Embarked = S |
|---|---|---|---|---|---|---|---|---|---|
| 3 | 22 | 1 | 0 | 2.692582 | 0 | 1 | 0 | 0 | 1 |
| 1 | 38 | 1 | 0 | 8.442944 | 1 | 0 | 1 | 0 | 0 |
| 3 | 26 | 0 | 0 | 2.815138 | 1 | 0 | 0 | 0 | 1 |
| 1 | 35 | 1 | 0 | 7.286975 | 1 | 0 | 0 | 0 | 1 |
| 3 | 35 | 0 | 0 | 2.837252 | 0 | 1 | 0 | 0 | 1 |

# BUILDING CLASSIFIER AND MAKING PREDICTION

Model building and prediction: Logistic Regression via Python



```
from sklearn.linear_model import LogisticRegression as Model

def train(features, target):
  model = Model()
  model.fit(features, target)
  return model

def predict(model, new_features):
  preds = model.predict(new_features)
  return preds

# Assume Titanic data is loaded into titanic_feats,
# titanic_target and titanic_test
model = train(titanic_feats, titanic_target)
predictions = predict(model, titanic_test)
```

Imports the logistic regression algorithm

Fits the logistic regression algorithm using features and target data

Makes predictions on a new set of features using the model

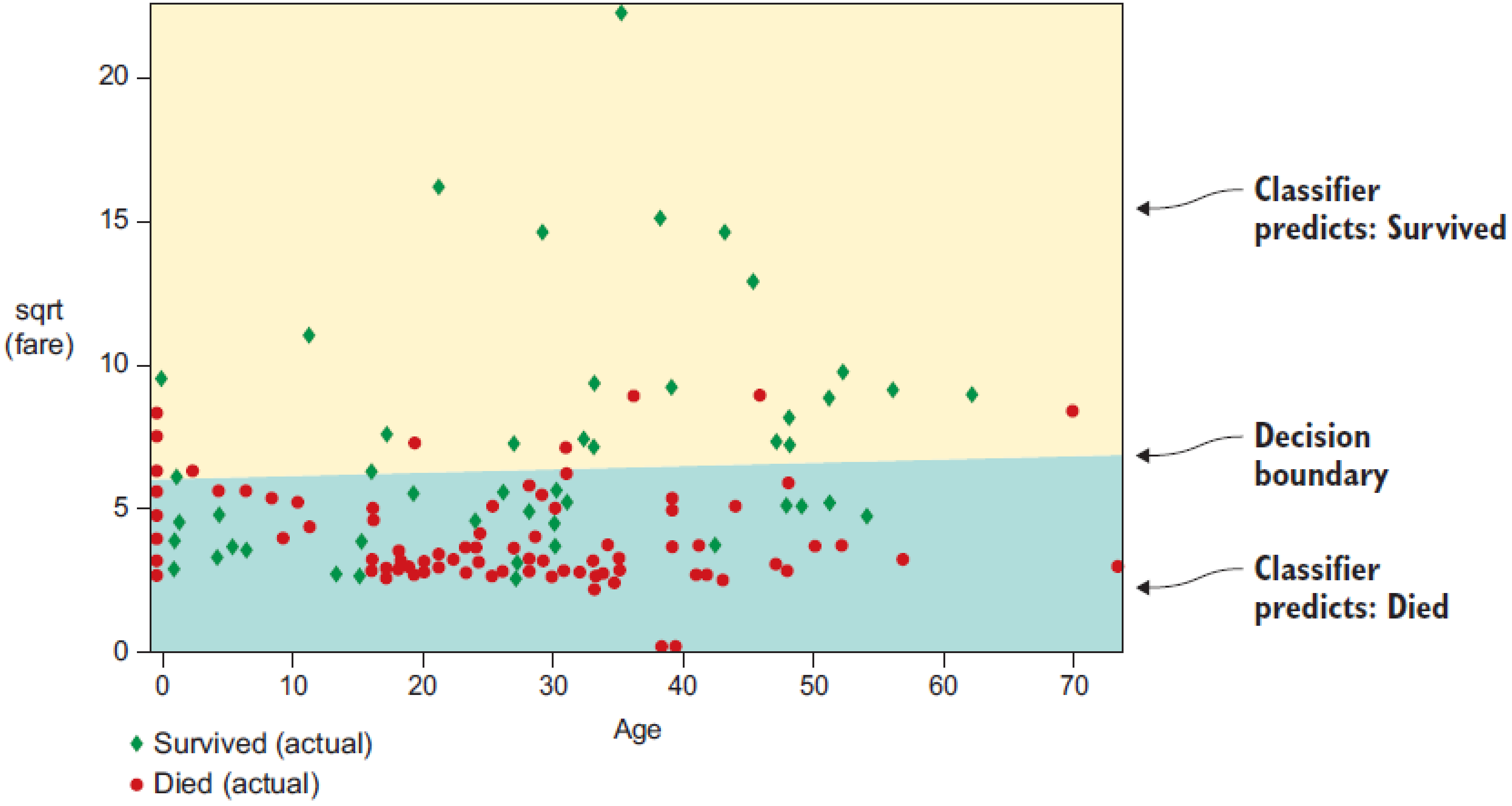
Returns the model built by the algorithm

Returns predictions (0 or 1)

The output of the predict function will be 1 if the passenger is predicted to survive, and 0 otherwise.

# BUILDING CLASSIFIER AND MAKING PREDICTION

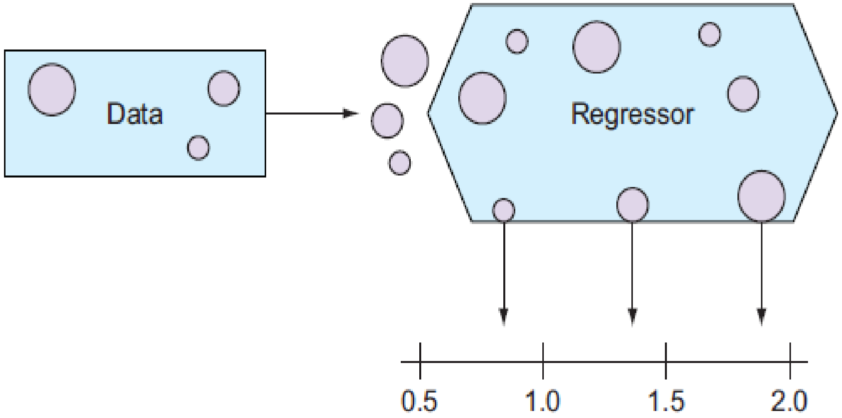
It's useful to visualize the classifier by plotting the decision boundary.
Two of the features in the dataset (age and sqrt fare), plot the boundary that separates surviving passengers from the dead, according to the model.



# OTHER ML SUPERVISED PROBLEM - REGRESSION

Purpose: Predicting Numerical Value

# REGRESSION



The regressor is predicting the numerical value of a record.

$$y = \beta_0 + \beta_1 x + \varepsilon$$

where:

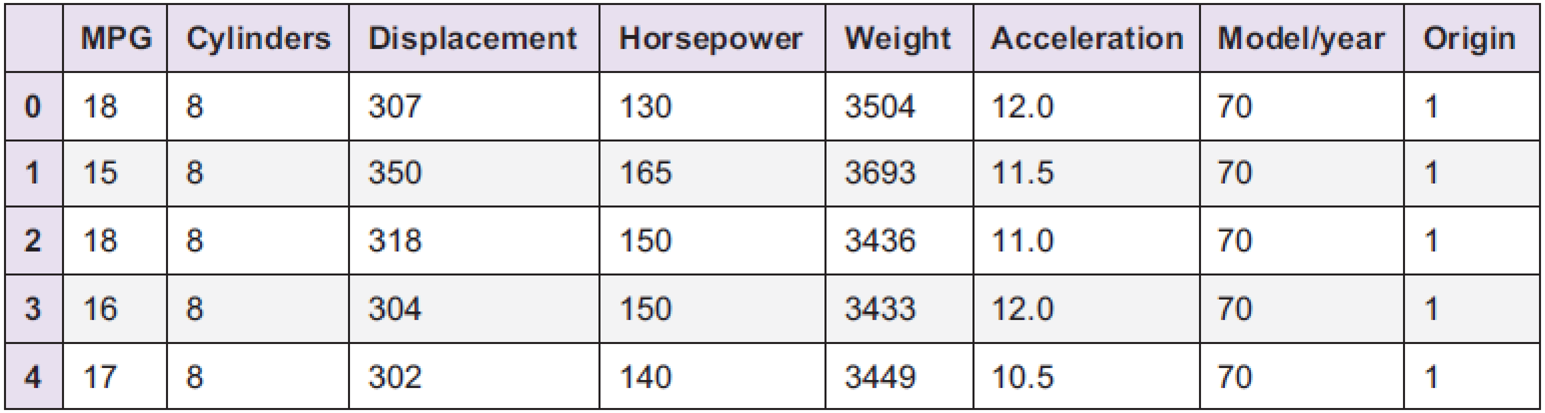
$y$ = Value of the dependent variable
$x$ = Value of the independent variable
$\beta_0$ = Population's y intercept
$\beta_1$ = Slope of the population regression line
$\varepsilon$ = Random error term

Linear Component

Random error component - maybe positive, zero or negative

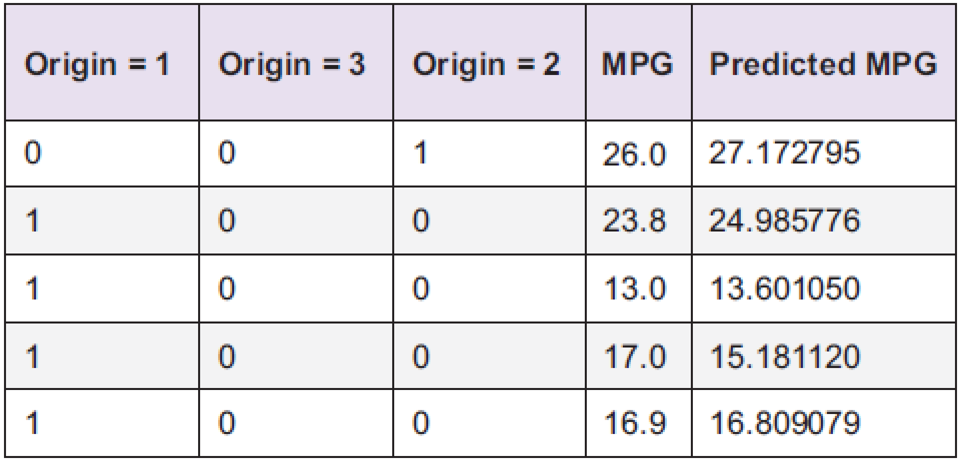
# REGRESSION

MPG Data Set



| | MPG | Cylinders | Displacement | Horsepower | Weight | Acceleration | Model/year | Origin |
|---|---|---|---|---|---|---|---|---|
| **0** | 18 | 8 | 307 | 130 | 3504 | 12.0 | 70 | 1 |
| **1** | 15 | 8 | 350 | 165 | 3693 | 11.5 | 70 | 1 |
| **2** | 18 | 8 | 318 | 150 | 3436 | 11.0 | 70 | 1 |
| **3** | 16 | 8 | 304 | 150 | 3433 | 12.0 | 70 | 1 |
| **4** | 17 | 8 | 302 | 140 | 3449 | 10.5 | 70 | 1 |

after model building and prediction



| Origin = 1 | Origin = 3 | Origin = 2 | MPG | Predicted MPG |
|---|---|---|---|---|
| 0 | 0 | 1 | 26.0 | 27.172795 |
| 1 | 0 | 0 | 23.8 | 24.985776 |
| 1 | 0 | 0 | 13.0 | 13.601050 |
| 1 | 0 | 0 | 17.0 | 15.181120 |
| 1 | 0 | 0 | 16.9 | 16.809079 |

# CLASSIFICATION & REGRESSION

HOW TO DIFFERENTIATE THEM?

# THE DIFFERENCE

| Property | Regression | Classification |
|---|---|---|
| Output Type | Discrete (Class label) | Continuous (number) |
| What are you trying to find? | Decision Boundary | Best fit Line |
| Evaluation | Confusion Matrix:<br>Accuracy<br>Specificity<br>Sensitivity | Error Rate:<br>SSE or Coefficient of determination, $R^2$, or MSE or MAD |

# CHECK YOUR UNDERSTANDING!!

Identify the individuals, variables, and discrete and continuous data in Table 1.

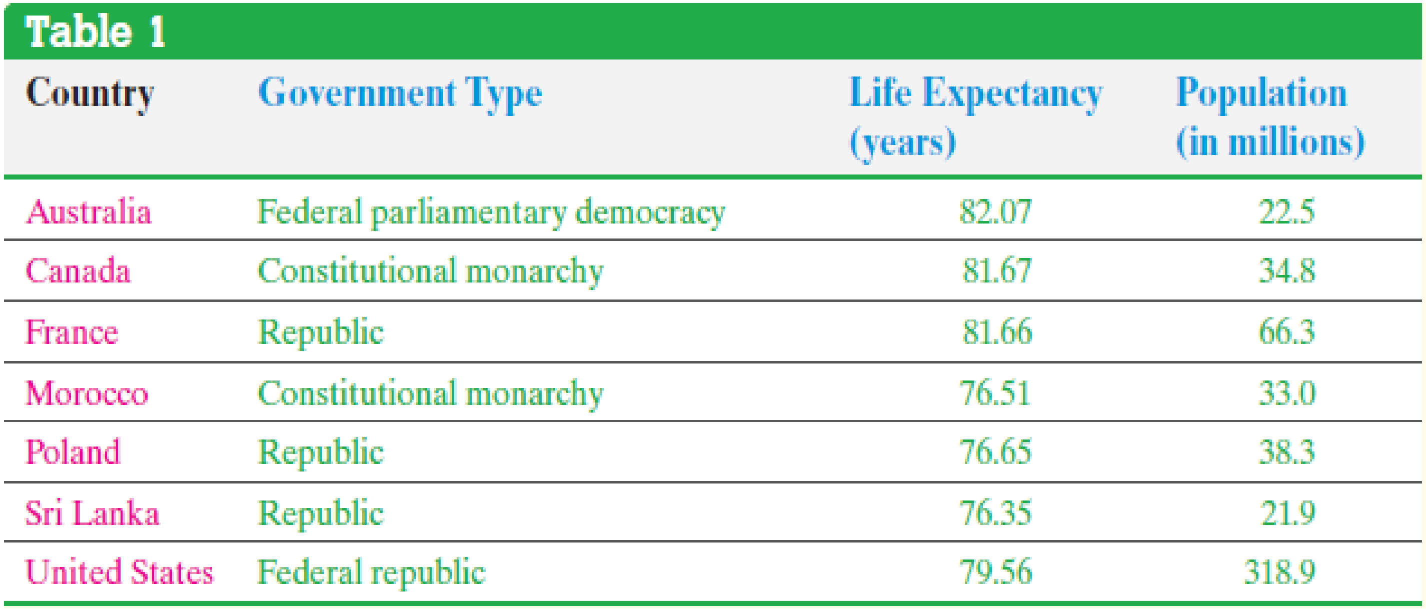
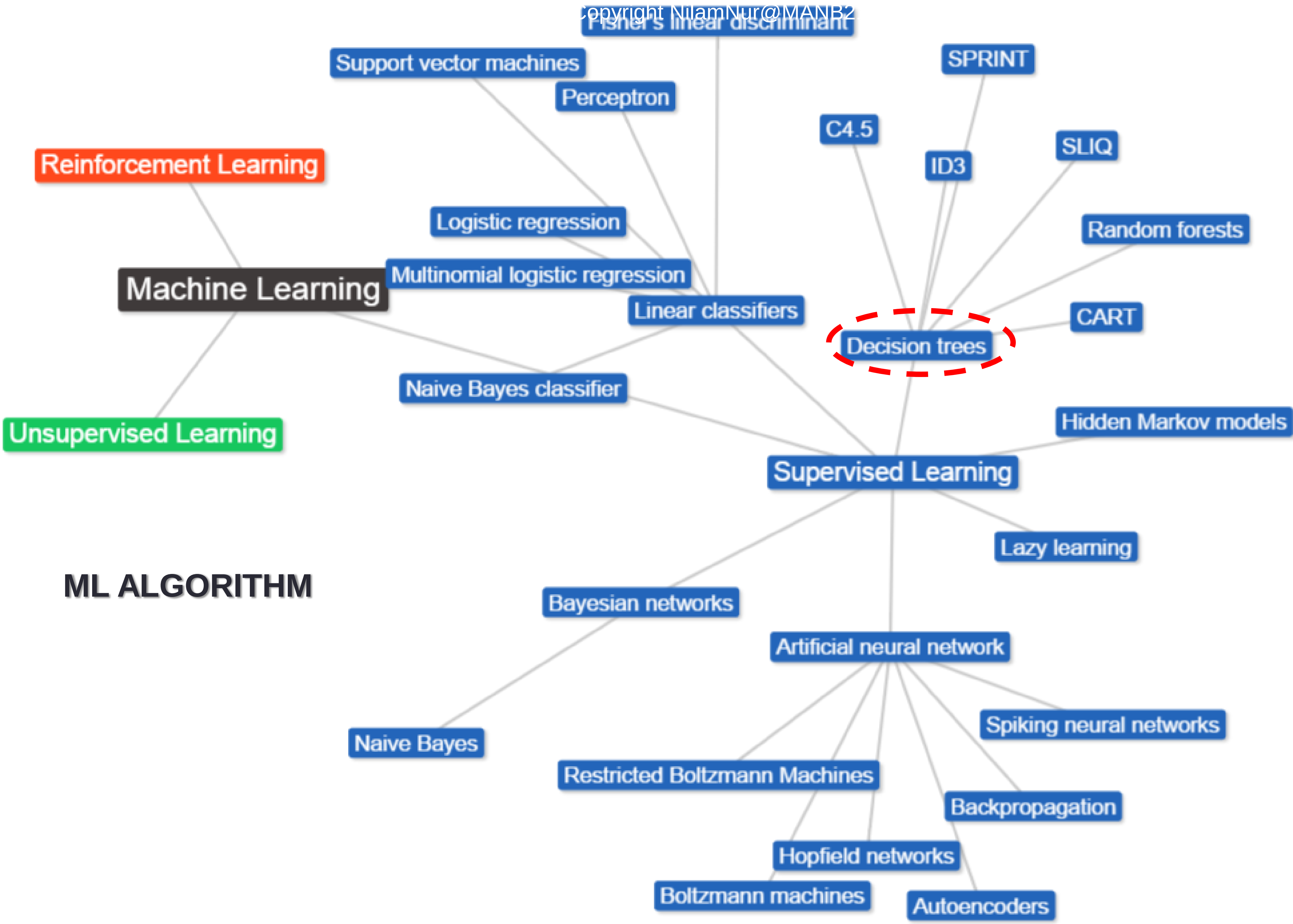


Table 1

| Country | Government Type | Life Expectancy (years) | Population (in millions) |
|---|---|---|---|
| Australia | Federal parliamentary democracy | 82.07 | 22.5 |
| Canada | Constitutional monarchy | 81.67 | 34.8 |
| France | Republic | 81.66 | 66.3 |
| Morocco | Constitutional monarchy | 76.51 | 33.0 |
| Poland | Republic | 76.65 | 38.3 |
| Sri Lanka | Republic | 76.35 | 21.9 |
| United States | Federal republic | 79.56 | 318.9 |

# OTHER ML ALGORITHM: DECISION TREE

## ML ALGORITHM



Machine Learning

Reinforcement Learning

Unsupervised Learning

Supervised Learning

Fisher's linear discriminant

Support vector machines

Perceptron

Logistic regression

Multinomial logistic regression

Linear classifiers

Naive Bayes classifier

Decision trees

C4.5

SPRINT

ID3

SLIQ

Random forests

CART

Hidden Markov models

Lazy learning

Bayesian networks

Naive Bayes

Artificial neural network

Spiking neural networks

Restricted Boltzmann Machines

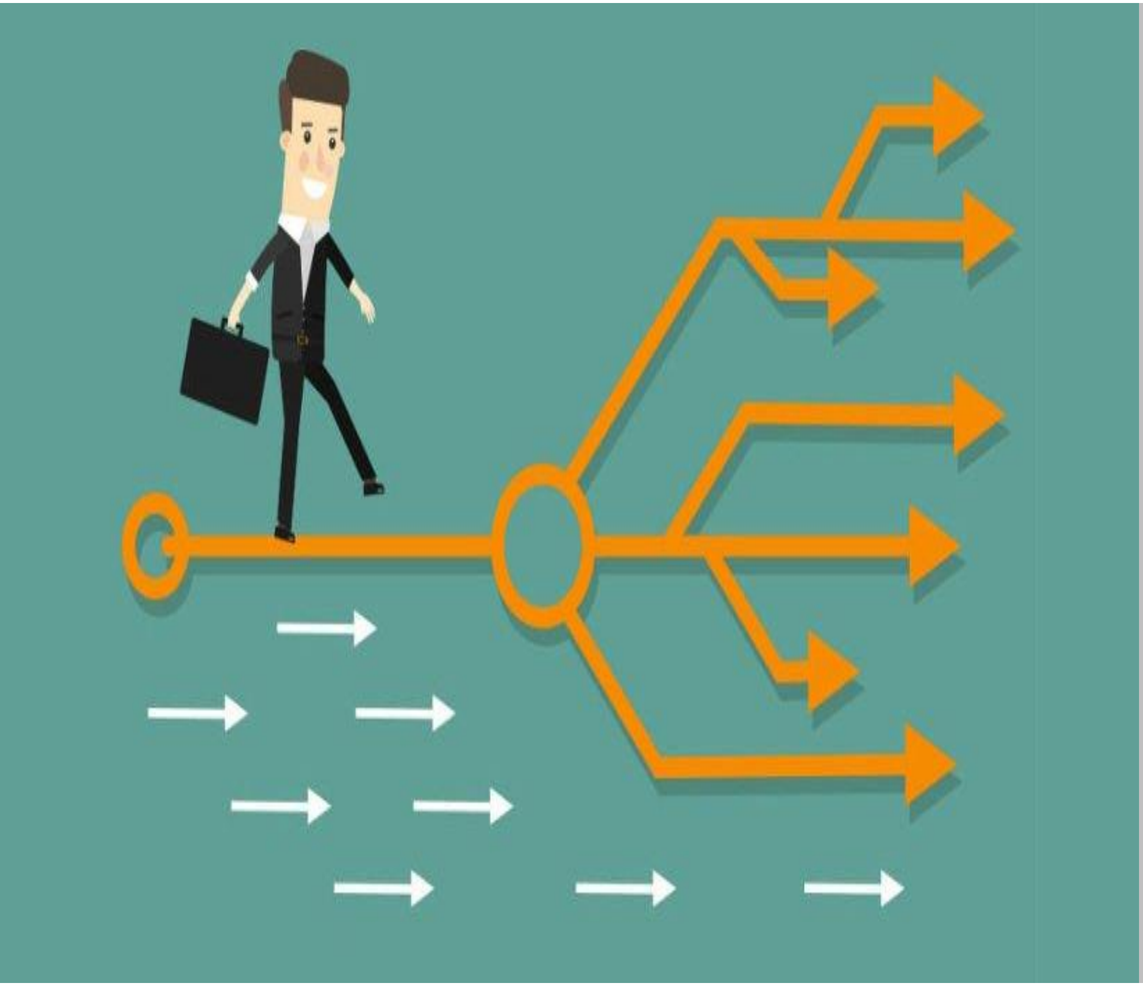
Backpropagation

Hopfield networks

Boltzmann machines

Autoencoders

# Decision Trees are excellent tools for helping you to choose between several courses of action.



- They provide a **highly effective structure** within which you can lay out options and investigate the possible outcomes of choosing those options.

- They also **help you to form a balanced picture of the risks and rewards** associated with each possible course of action.

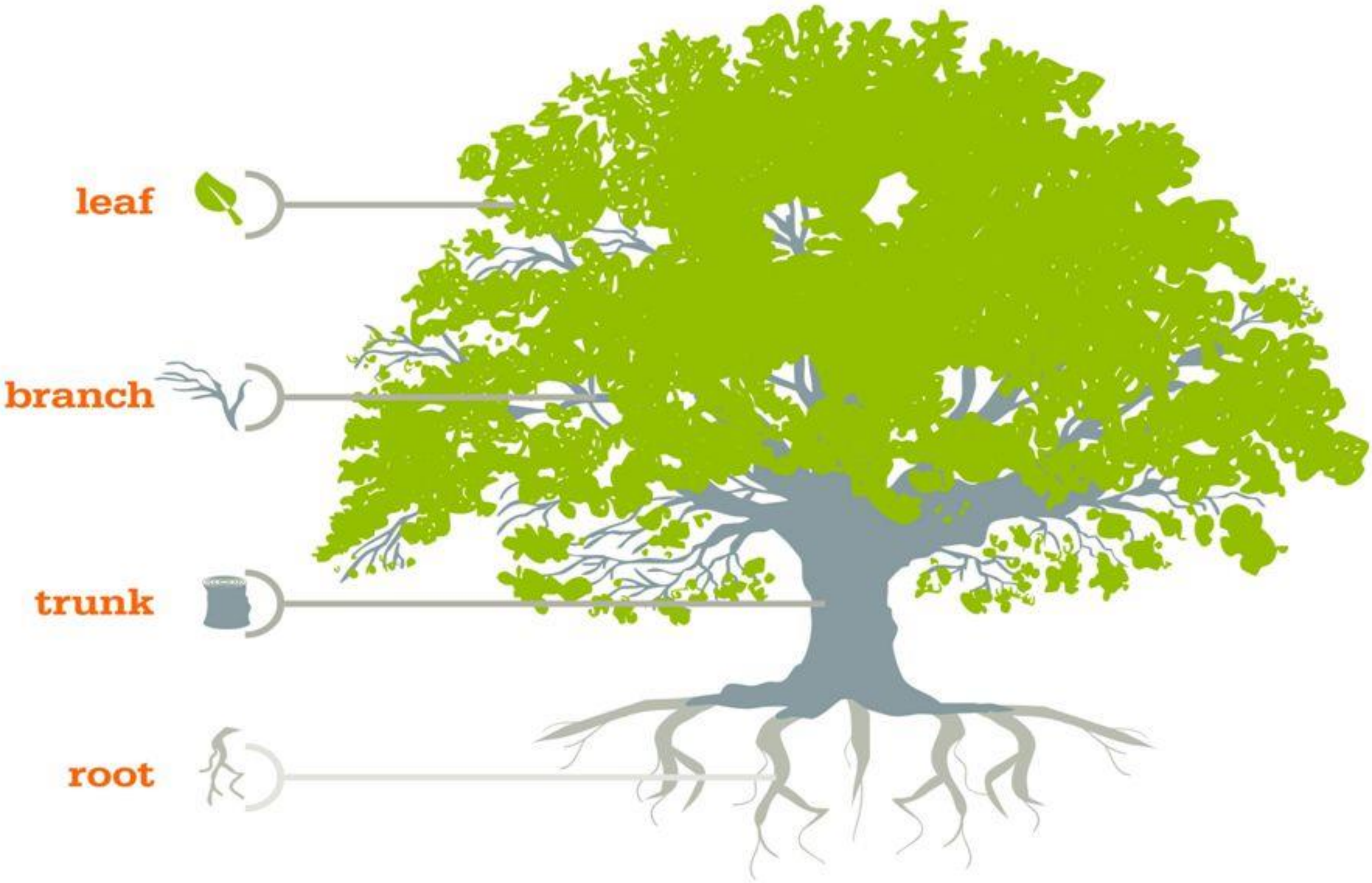
# Decision Tree : Concept

- Supervised algorithm (needs dataset for creating a tree)
- Also known as Greedy algorithm (favorite attribute first)
- Rules for classifying data using attributes.
- This algorithm makes classification decision for a test sample with the help of tree like structure.
  - **Nodes** in the tree are attribute names of the given data.
  - **Branches** in the tree are attributes values
  - **Leaf** nodes are the class labels.

# Decision Tree : Concept

- A decision tree is an example of a nonparametric ML algorithm, because its functional form isn’t fixed.
- The tree model can grow in complexity with larger amounts of data to capture more complicated patterns.
- In each terminal node of the tree, the ratio represents the number of training instances in that node that died versus lived.
- Variables which is the most important is put in the top of the tree, and then gradually use less-important variables.

fierce. CONVERSATIONS® Delegate



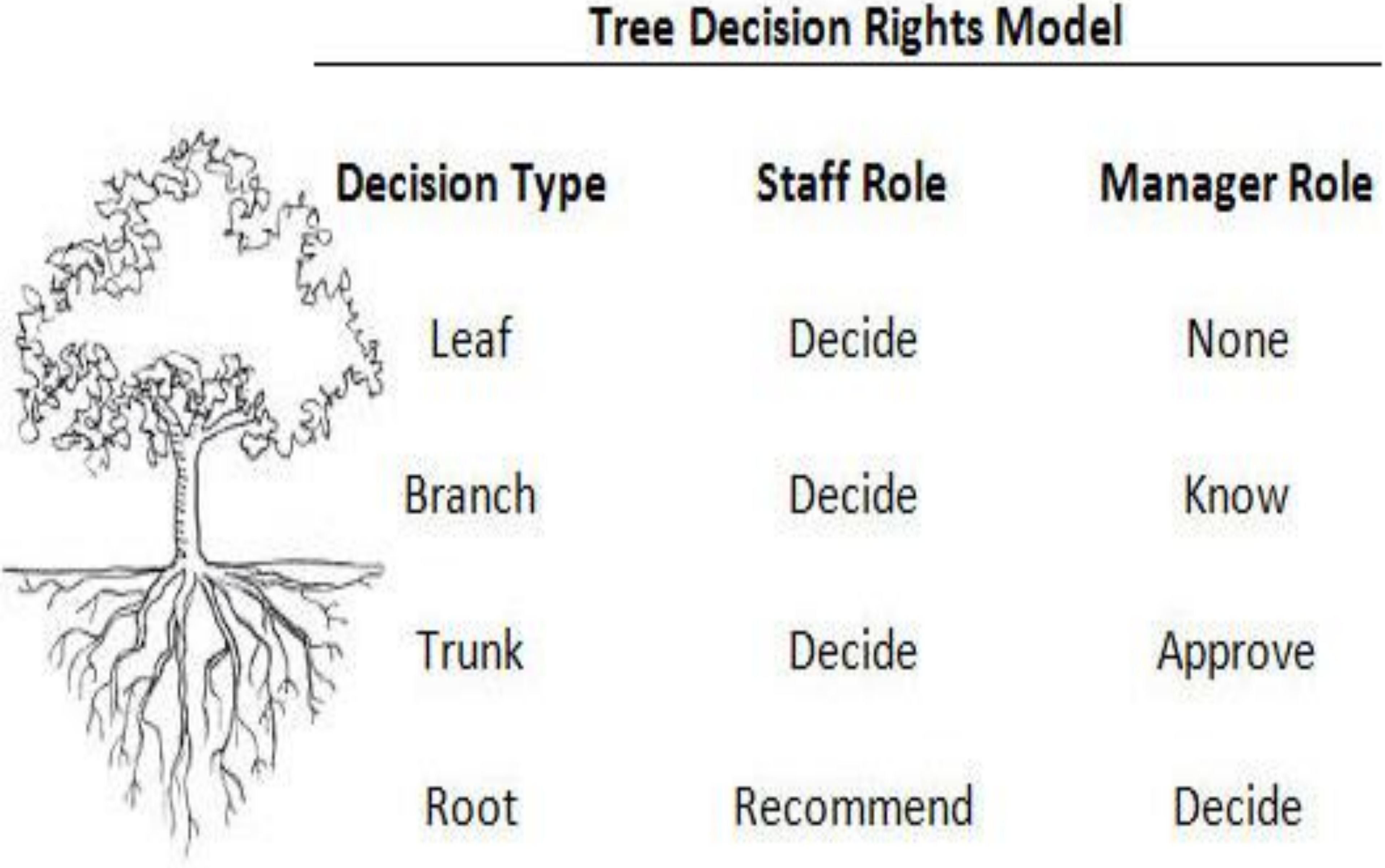
leaf

branch

trunk

root

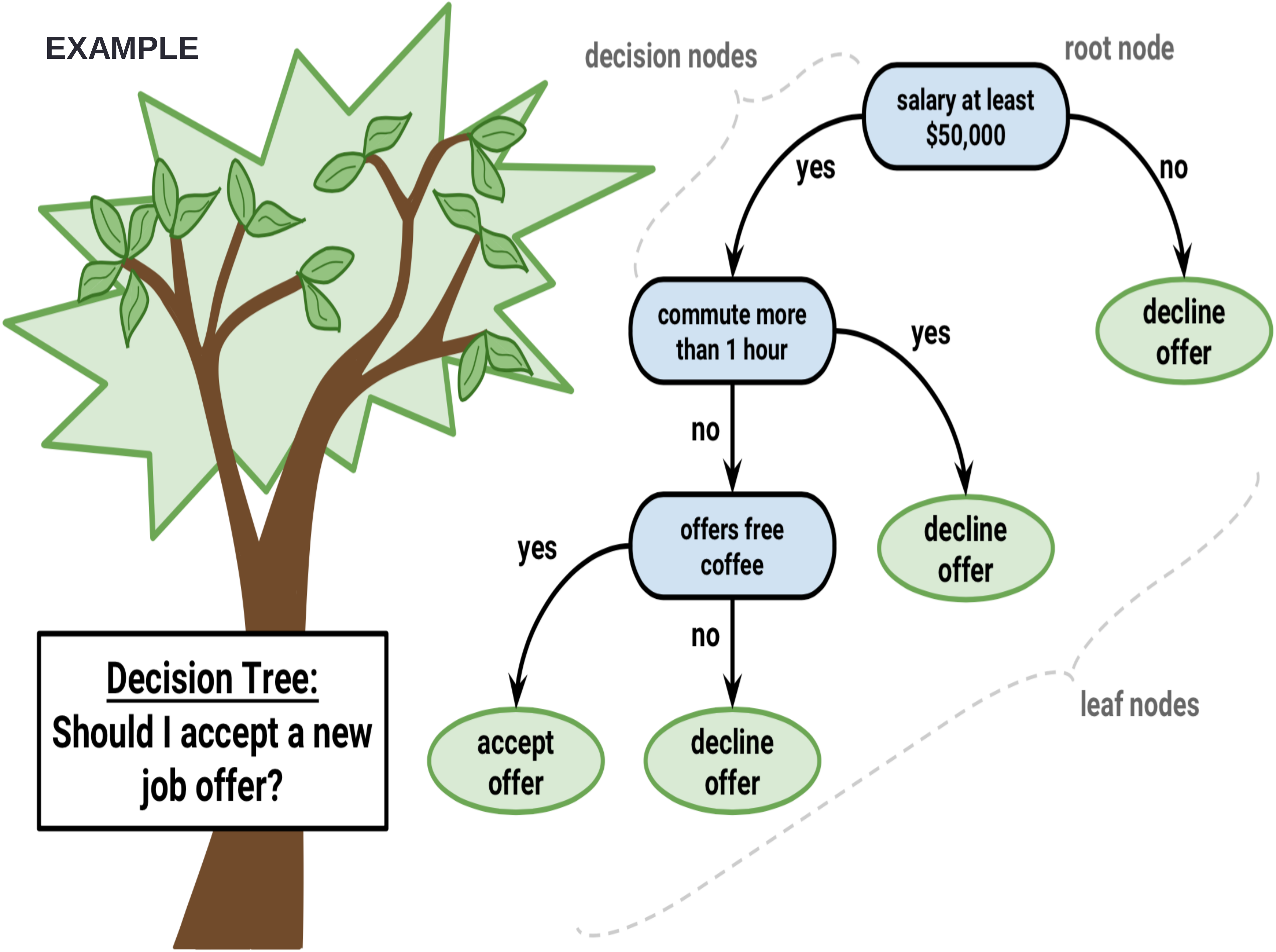
## THE METAPHOR



### Tree Decision Rights Model

| Decision Type | Staff Role | Manager Role |
|---|---|---|
| Leaf | Decide | None |
| Branch | Decide | Know |
| Trunk | Decide | Approve |
| Root | Recommend | Decide |

## EXAMPLE



**Decision Tree:**

Should I accept a new job offer?

decision nodes

root node

salary at least $50,000

yes

no

commute more than 1 hour

yes

no

decline offer

offers free coffee

yes

no

decline offer

accept offer

decline offer

leaf nodes

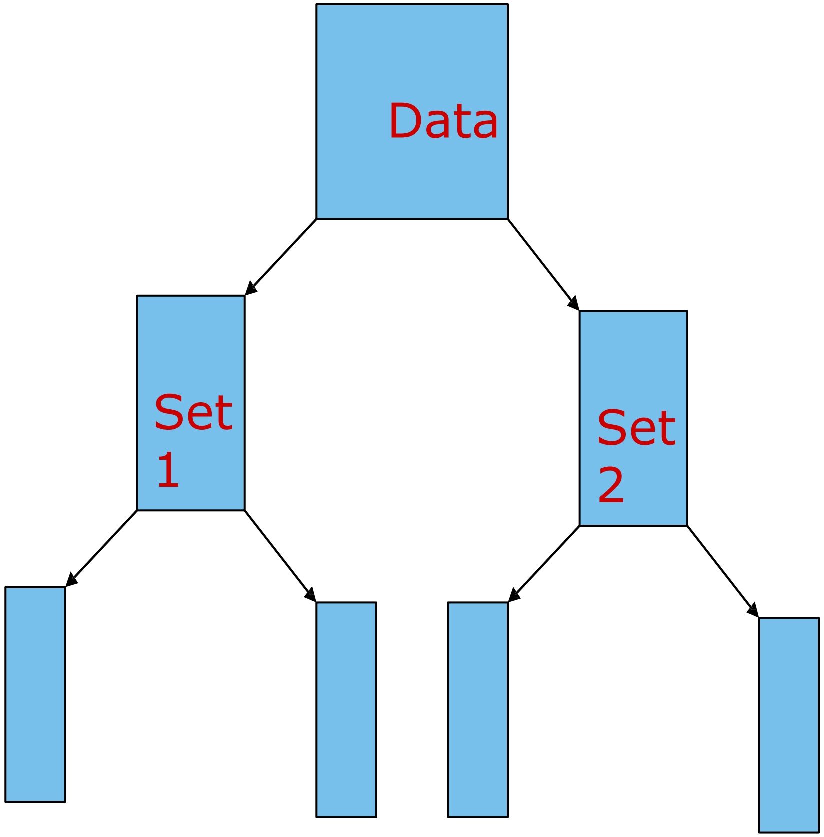
# Building Decision Tree

- Decision tree generation consists of two step methods:
  - Tree construction
    - At start, all the training examples are at the root
    - Partition examples recursively based on selected attributes.
    - Attributes are categorical if continuous-valued, they are discretized in advanced
    - Test attributes are selected on the basis of  a heuristic or statistical measure (e.g.,  information gain)
    - How??
      1. Pick an attribute for division of given data.
      2. Divide the given data into sets on the basis of this attribute
      3. For every set created above-repeat 1 and 2 until you find leaf nodes in all the branches of the tree-terminate.
  - Tree pruning (Optimization)
    - Identify and remove branches that in Decision Tree that are not useful for classification
      1. Pre-Pruning
      2. Post-Pruning

# Condition for Stopping Partitioning

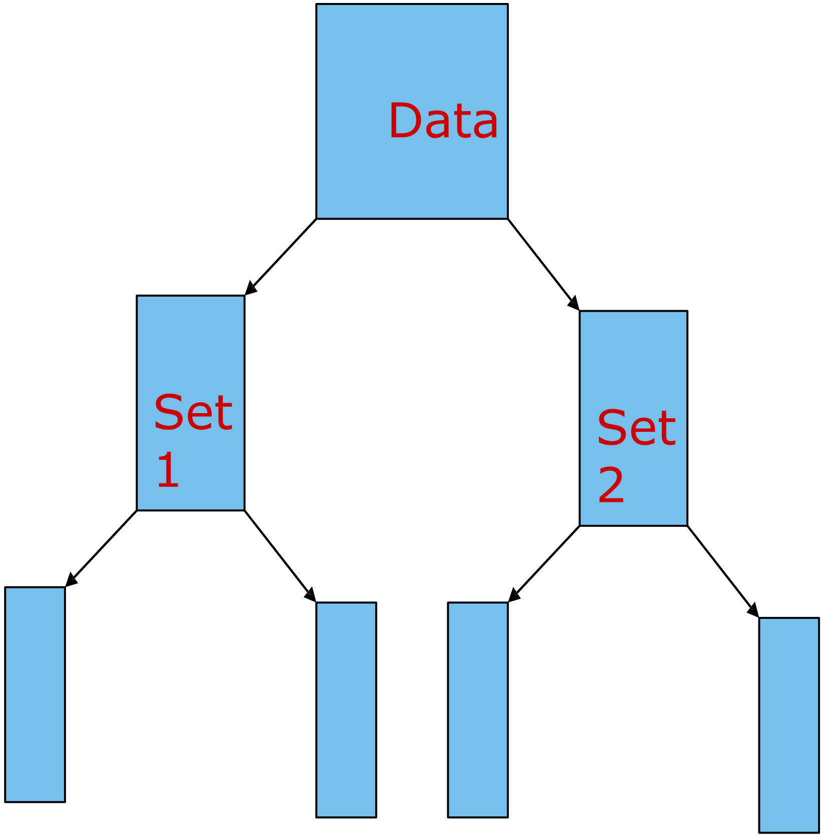
- All samples for a given node belong to the same class
- There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
- There are no samples left

# Algorithm in Action…



- For prediction, in decision tree, $h: \mathcal{X} \rightarrow Y$, that predict the label associated with an instance $x$ by travelling from a root node of a tree to a leaf.
- For simplicity, we focus on binary classification setting, namely $\mathrm{Y} = \{0,1\}$, but decision tree can be applied for other prediction problems as well

# Algorithm in Action…



- At each node on the root-to-leaf path, the successor child is chosen on the basis of a splitting of the input space.
- The splitting is based on the features of x or on a predefined set of splitting rules.
- A leaf contain a specific label.

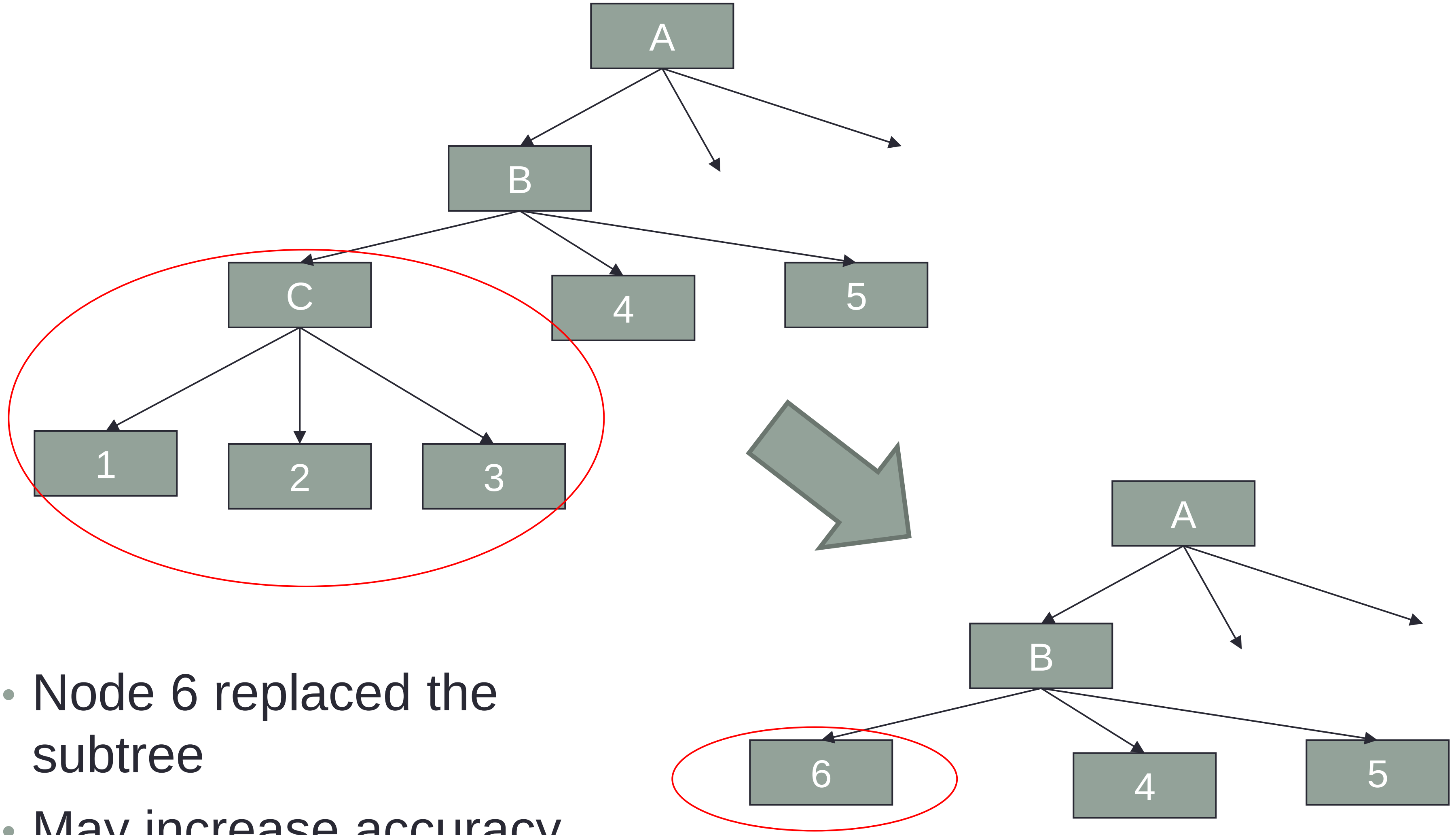
# Pre-Pruning

- We decide during the building process, when to stop adding attributes possibly based on their information gain.
- However, this may be problematic – why?
  - Sometimes, attribute individually do not compute much to a decision, but combined they may have significant impact.

# Post-pruning

- Post-pruning waits until full decision tree has been built and then prunes the attributes.
- Two techniques:
    - Subtree replacement
    - Subtree raising

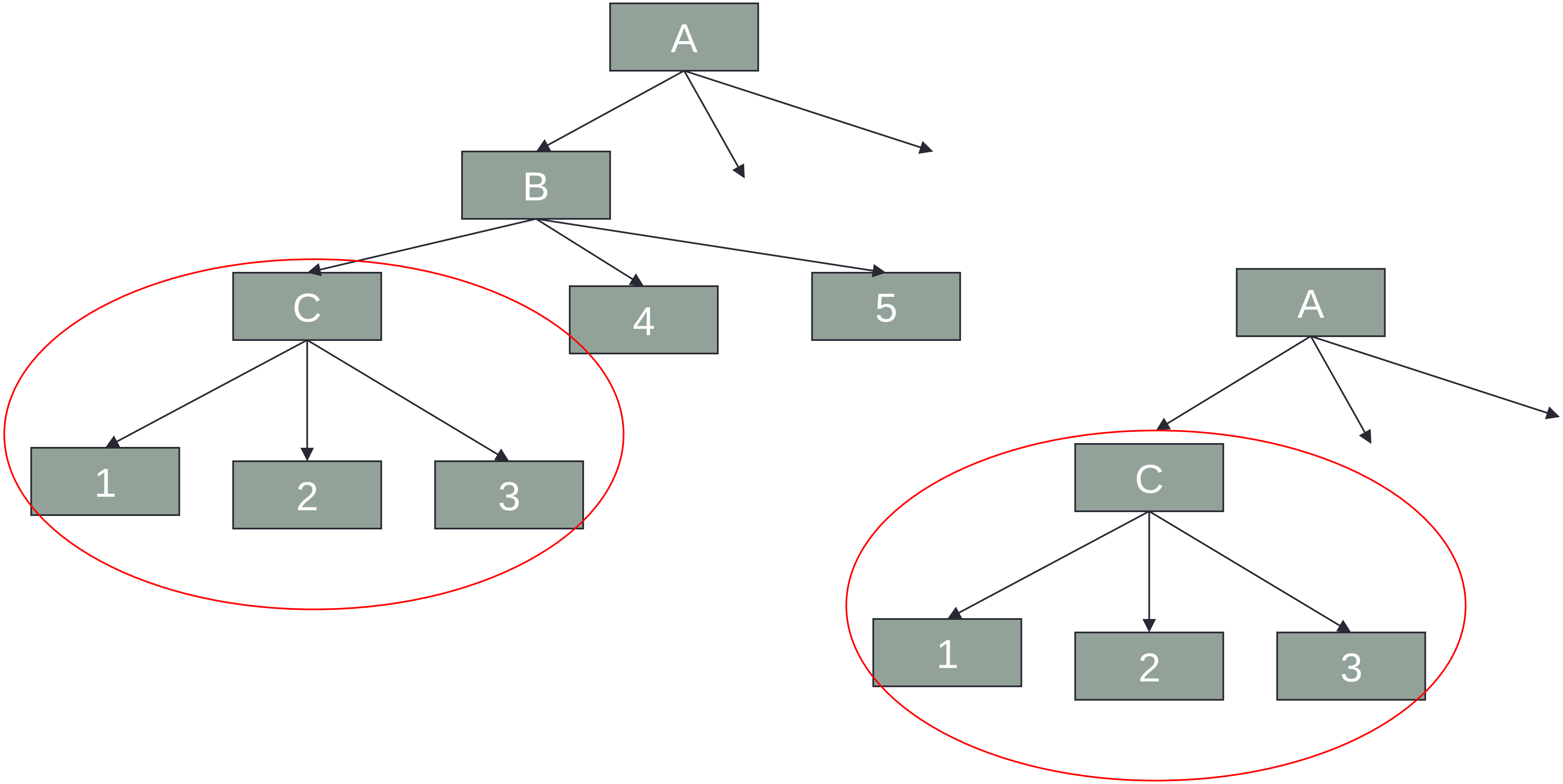
# Subtree Replacement



- Node 6 replaced the subtree
- May increase accuracy

# Subtree Raising

- Entire subtree is raised onto another node



# Algorithm at work
# Tree-Construction (Step 1)

**Given data**

| age | income | student | credit_rating | buys_computer |
|---|---|---|---|---|
| <=30 | high | no | fair | no |
| <=30 | high | no | excellent | no |
| 31…40 | high | no | fair | yes |
| >40 | medium | no | fair | yes |
| >40 | low | yes | fair | yes |
| >40 | low | yes | excellent | no |
| 31…40 | low | yes | excellent | yes |
| <=30 | medium | no | fair | no |
| <=30 | low | yes | fair | yes |
| >40 | medium | yes | fair | yes |
| <=30 | medium | yes | excellent | yes |
| 31…40 | medium | no | excellent | yes |
| 31…40 | high | yes | fair | yes |
| >40 | medium | no | excellent | no |

# Algorithm at work
# Tree-Construction (Step 2)

Three Data Sets formed (colour) after division at root node on the basis of “age” attribute

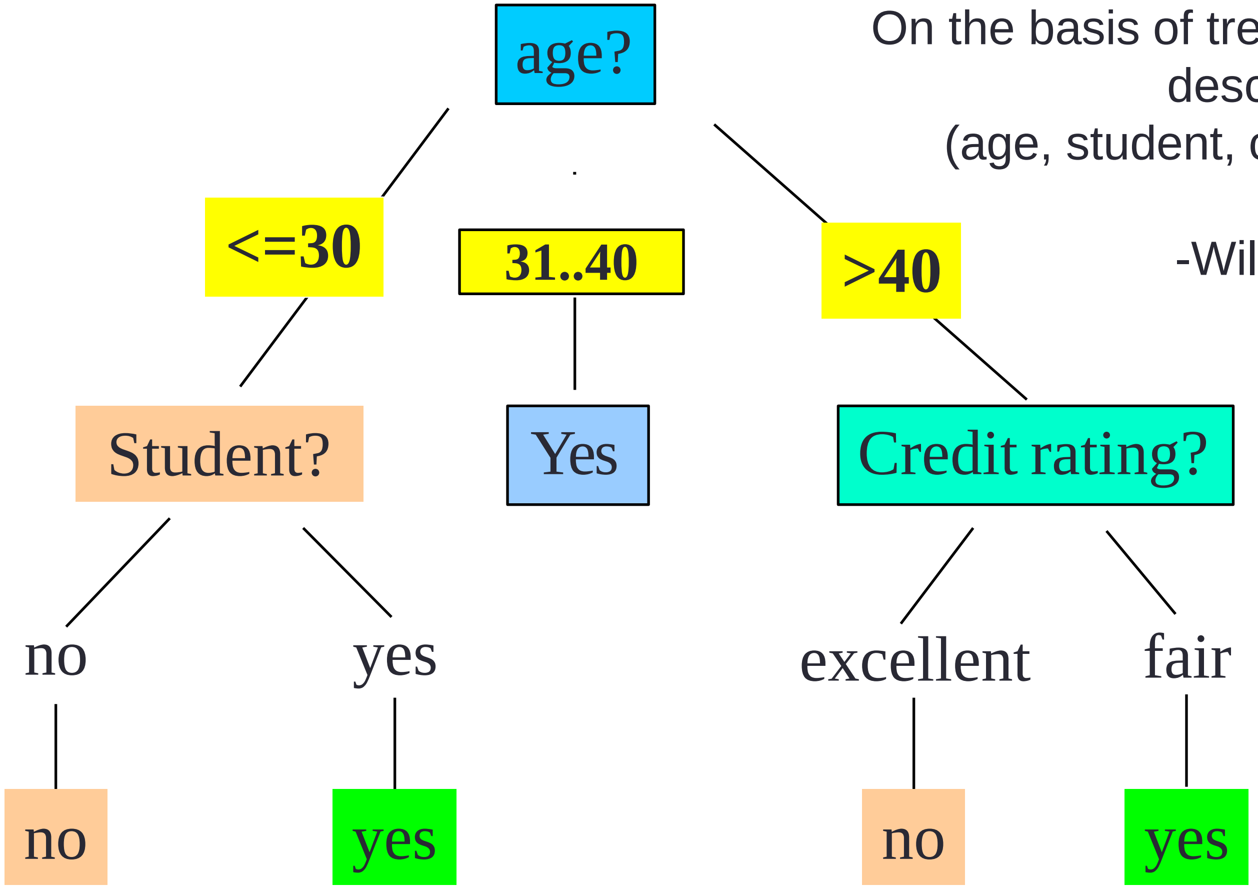
| age | income | student | credit_rating | buys_computer |
|---|---|---|---|---|
| <=30 | high | no | fair | no |
| <=30 | high | no | excellent | no |
| 31…40 | high | no | fair | yes |
| >40 | medium | no | fair | yes |
| >40 | low | yes | fair | yes |
| >40 | low | yes | excellent | no |
| 31…40 | low | yes | excellent | yes |
| <=30 | medium | no | fair | no |
| <=30 | low | yes | fair | yes |
| >40 | medium | yes | fair | yes |
| <=30 | medium | yes | excellent | yes |
| 31…40 | medium | no | excellent | yes |
| 31…40 | high | yes | fair | yes |
| >40 | medium | no | excellent | no |

- On the basis of tree constructed in the manner described, classify a test sample (age, student?, credit rating, buys_computer)

  **(<=30, yes, excellent, ?**)

- Will this student buy computer?

# Final Decision Tree



On the basis of tree constructed in the manner described, classify a test sample (age, student, creditrating, buys_computer) **(<=30, yes, excellent, ?**)

-Will this student buy computer?

# CHILL OUT & WATCH THIS

https://youtu.be/RmajweUFKvM

# No classifier is inherently better than any other: you need to make assumptions to generalize

## No Free Lunch Theorem

Slide credit: D. Hoiem

# READY FOR ASSIGNMENT #2?