MENG 3065 - MODULE 4

Artificial Intelligence: A Modern Approach
Chapter 19 Learning From Examples - Supervised

These slides has been extracted, modified and updated from original slides of :
Artificial Intelligence: A Modern Approach, Fourth Edition.© 2022 Pearson Education, Inc.

HUMBER

WE ARE HUMBER

Outline

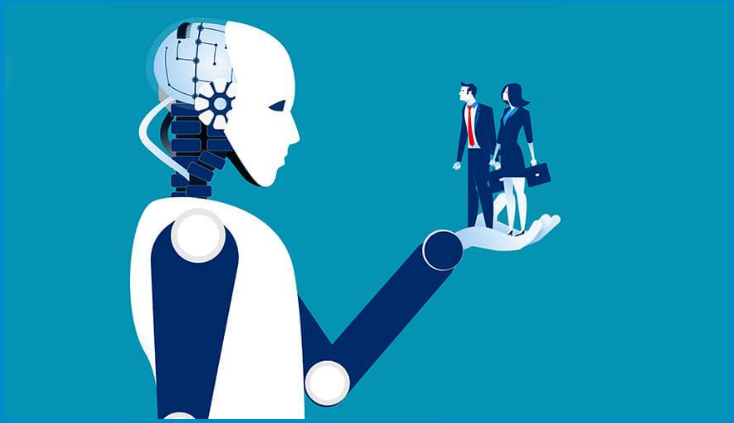
- Introduction: Learning
- Forms of Learning:
  - Supervised
  - Unsupervised
  - Reinforcement learning
- Supervised Learning Examples:
  - Decision Trees
  - Linear Regression
  - Logistic regression
- Evaluation Metrics in Machine Learning
- Supervised Learning Workflow

Remember: AI vs ML vs DL

- Artificial Intelligence: Create intelligent machines that work and act like humans.
- Machine Learning: Find an algorithm that automatically learns from example data.
- Deep Learning: Using deep neural networks to automatically learn from example data.

Introduction

- Machine learning is making great strides
  - Large, good data sets
  - Compute power
  - Progress in algorithms
- Many interesting applications
  - Commercial
  - Scientific



Forms of Learning

- There are three types of feedback that can accompany the inputs, and that determine the three main types of learning:
- Supervised Learning
  - agent observes input-output pairs
  - learns a function that maps from input to output
- Unsupervised Learning
  - agent learns patterns in the input without any explicit feedback
  - clustering
- Reinforcement Learning
  - agent learns from a series of reinforcements: rewards & punishments

Machine learning tasks

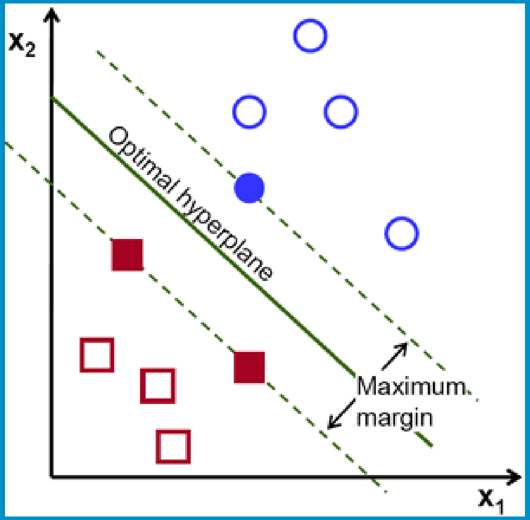
- Supervised learning
  - classification: predict categorical values, i.e., labels
  - regression: predict numerical values
- Unsupervised learning
  - clustering: group data according to "distance"
  - association: find frequent co-occurrences
  - link prediction: discover relationships in data
  - data reduction: project features to fewer features

Machine learning algorithms

- Classification:
  Naive Base, Support Vector Machines,
  Random Forest, Multilayer Neural Networks,
  Deep Neural Networks, ...

- Regression:
  Regression, Support Vector Machines, Random Forest,
  Multilayer Neural Networks, Deep Neural Networks, ...

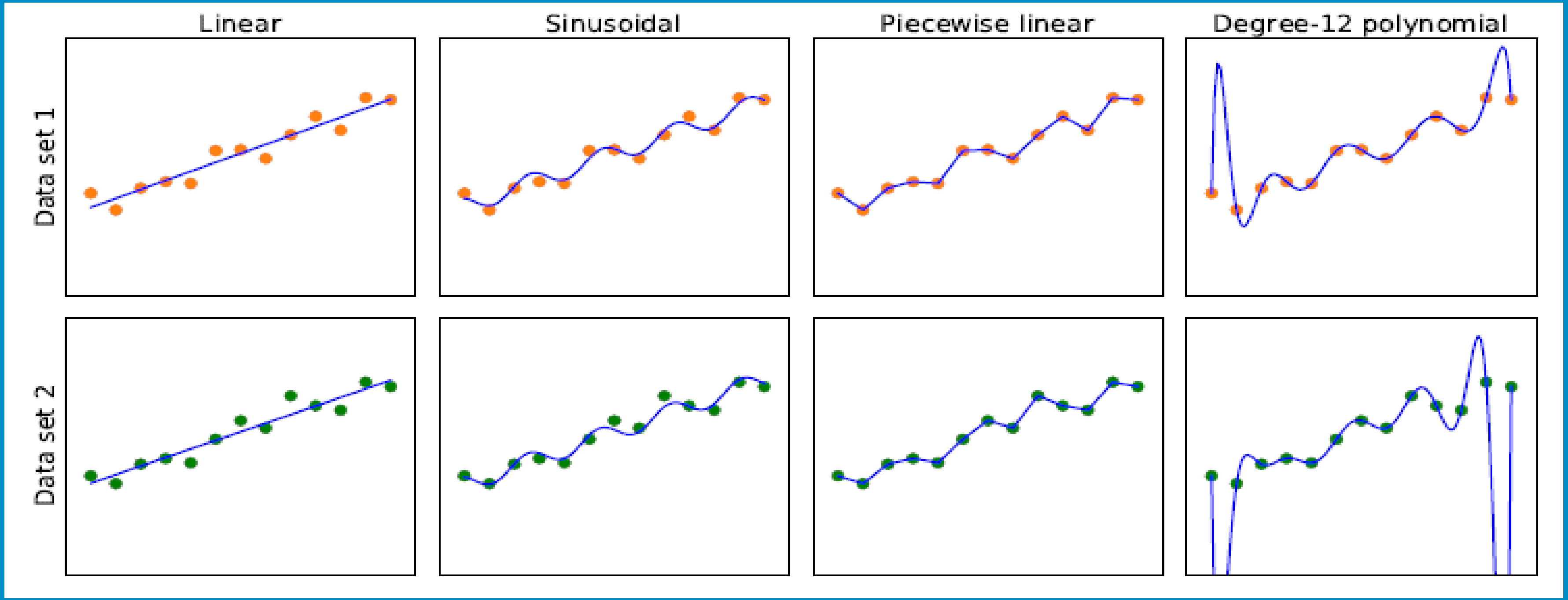
- Clustering:
  k-Means, Hierarchical Clustering, ...



Supervised Learning

- Training set of examples of input output (N)
  - (x1, y1), (x2, y2), . . . (xN, yN) ,
  - y = f (x),
- Function *h* is hypothesis about the world, approximates the true function f
  - drawn from a hypothesis space *H* of possible functions
  - *h* Model of the data, drawn from a model class *H*
- Consistent hypothesis: an *h* such that each x_i in the training set has $h(x_i) = y_i$.
  - look for a best-fit function for which each h(x_i) is close to y_i
- The true measure of a hypothesis, depends on how well it handles inputs it has not yet seen. E.g.: a second sample of (x_i, y_i)
- *h* generalizes well if it accurately predicts the outputs of the test set

Supervised Learning



Finding hypotheses to fit data.

Top row: four plots of best-fit functions from four different hypothesis spaces trained on data set 1.

Bottom row: the same four functions, but trained on a slightly different data set (sampled from the same *f* (*x*) function).

Supervised Learning

- Use **bias** to analyze hypothesis space
  - the tendency of a predictive hypothesis to deviate from the expected value when averaged over different training set
- **Underfitting**: fails to find a pattern in the data
- **Variance:** the amount of change in the hypothesis due to fluctuation in the training data.
- **Overfitting:** when it pays too much attention to the particular data set it is trained on, causing it to perform poorly on unseen data.
- **Bias–variance tradeoff**: a choice between more complex, low-bias hypotheses that fit the training data well and simpler, low-variance hypotheses that may generalize better.

Supervised Learning - Classification

HUMBER

WE ARE HUMBER

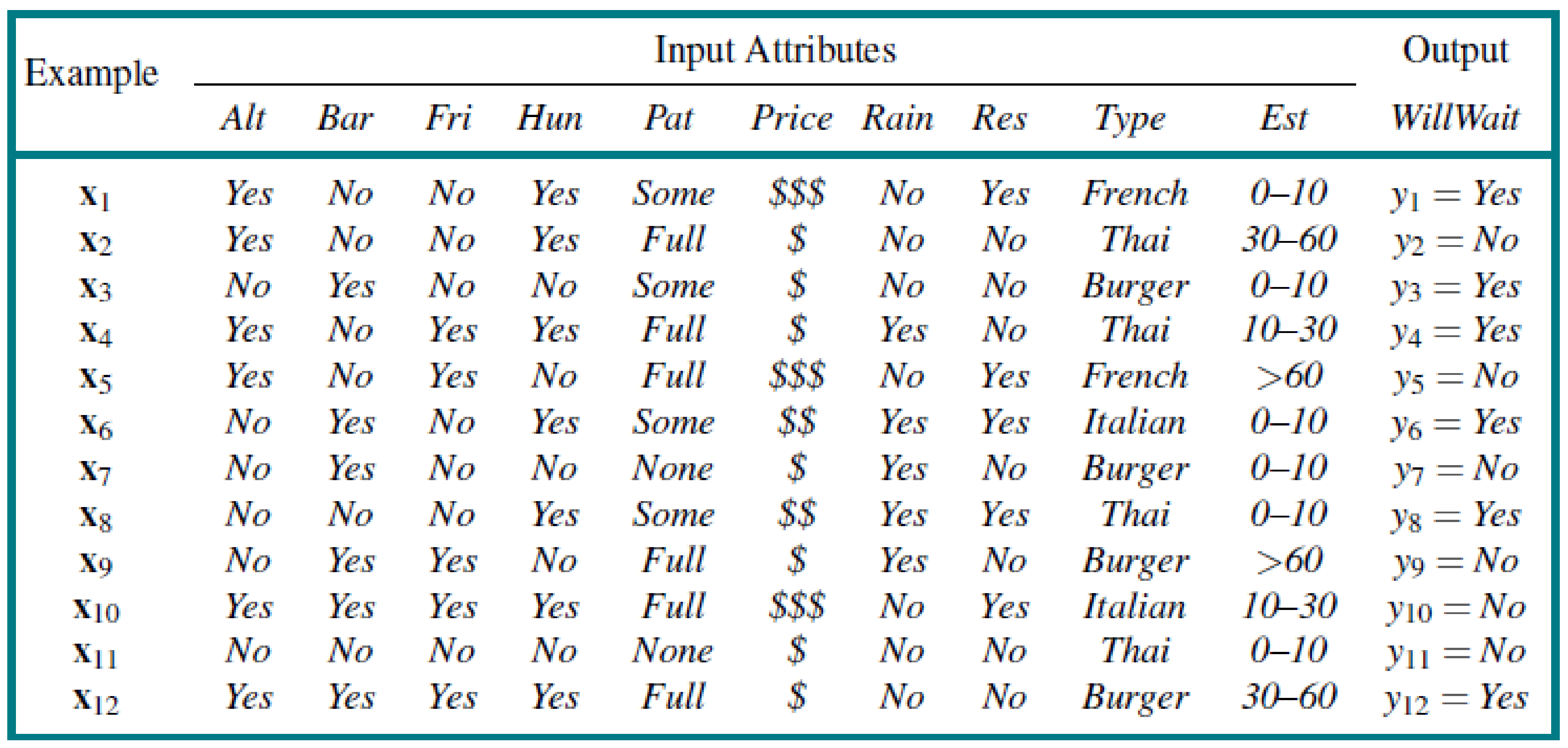
Supervised Learning - Example

Example problem: Restaurant waiting

- the problem of deciding whether to wait for a table at a restaurant.
- For this problem the **output, *y*,** is a Boolean variable that we will call ***WillWait***.
- **The input, *x*,** is a vector of ten attribute values, each of which has discrete values:
  1. *Alternate*: whether there is a suitable alternative restaurant nearby.
  2. *Bar*: whether the restaurant has a comfortable bar area to wait in.
  3. *Fri/Sat*: true on Fridays and Saturdays.
  4. *Hungry*: whether we are hungry right now.
  5. *Patrons*: how many people are in the restaurant (values are *None*, *Some*, and *Full*).
  6. *Price*: the restaurant’s price range ($, $$, $$$).
  7. *Raining*: whether it is raining outside.
  8. *Reservation*: whether we made a reservation.
  9. *Type*: the kind of restaurant (French, Italian, Thai, or burger).
  10. *WaitEstimate*: host’s wait estimate: 0–10, 10–30, 30–60, or >60minutes

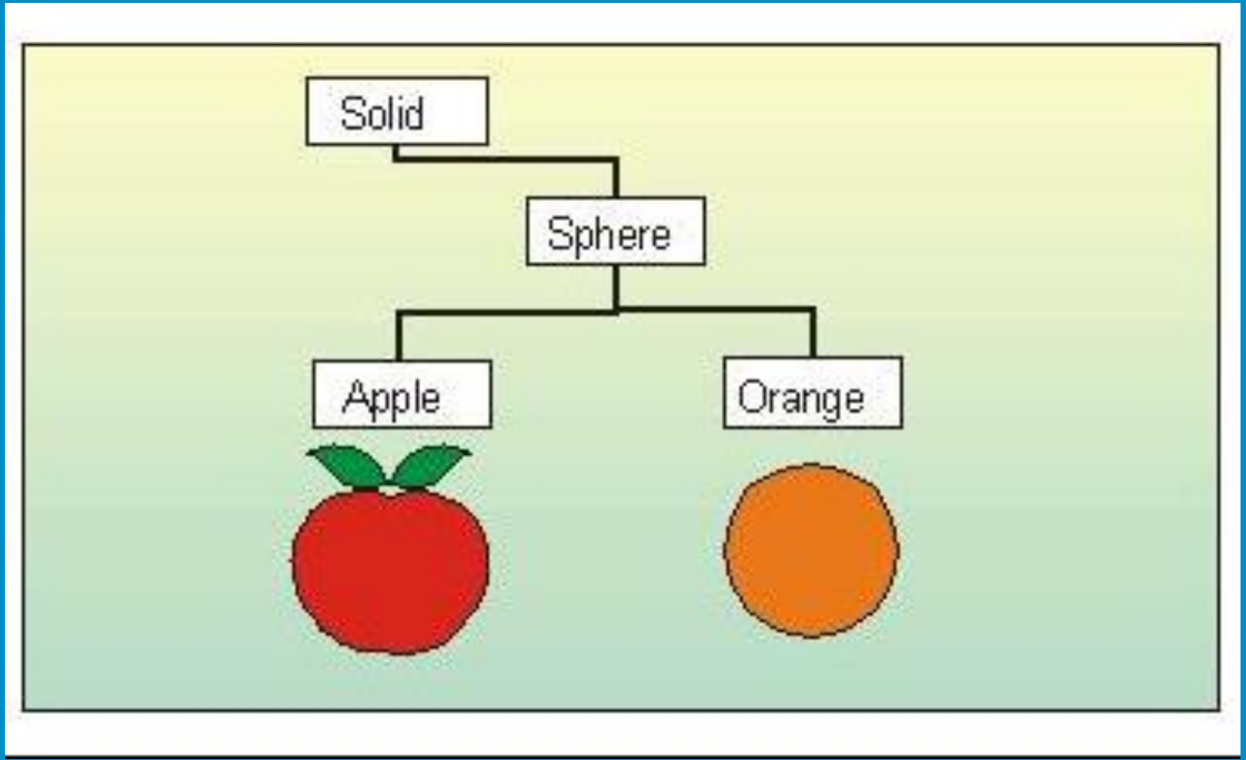
Supervised Learning – Example (Cont'd)

- Examples for the restaurant domain.

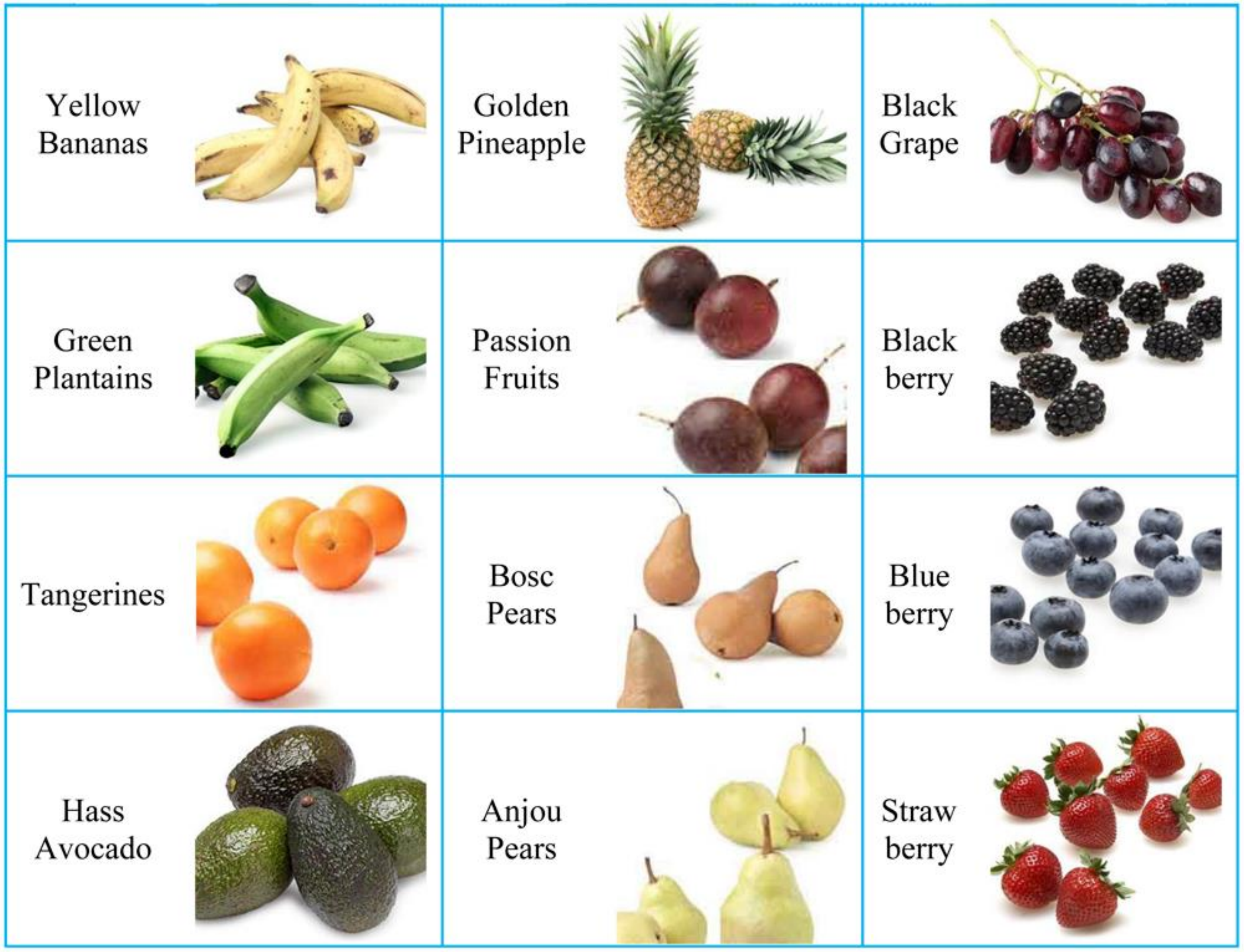


Example	Input Attributes										Output
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	$y_1 = Yes$
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	$y_2 = No$
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	$y_3 = Yes$
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10–30*	$y_4 = Yes$
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 = No$
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	$y_6 = Yes$
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	$y_7 = No$
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	$y_8 = Yes$
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 = No$
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	$y_{10} = No$
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	$y_{11} = No$
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	$y_{12} = Yes$

Fruit Categories

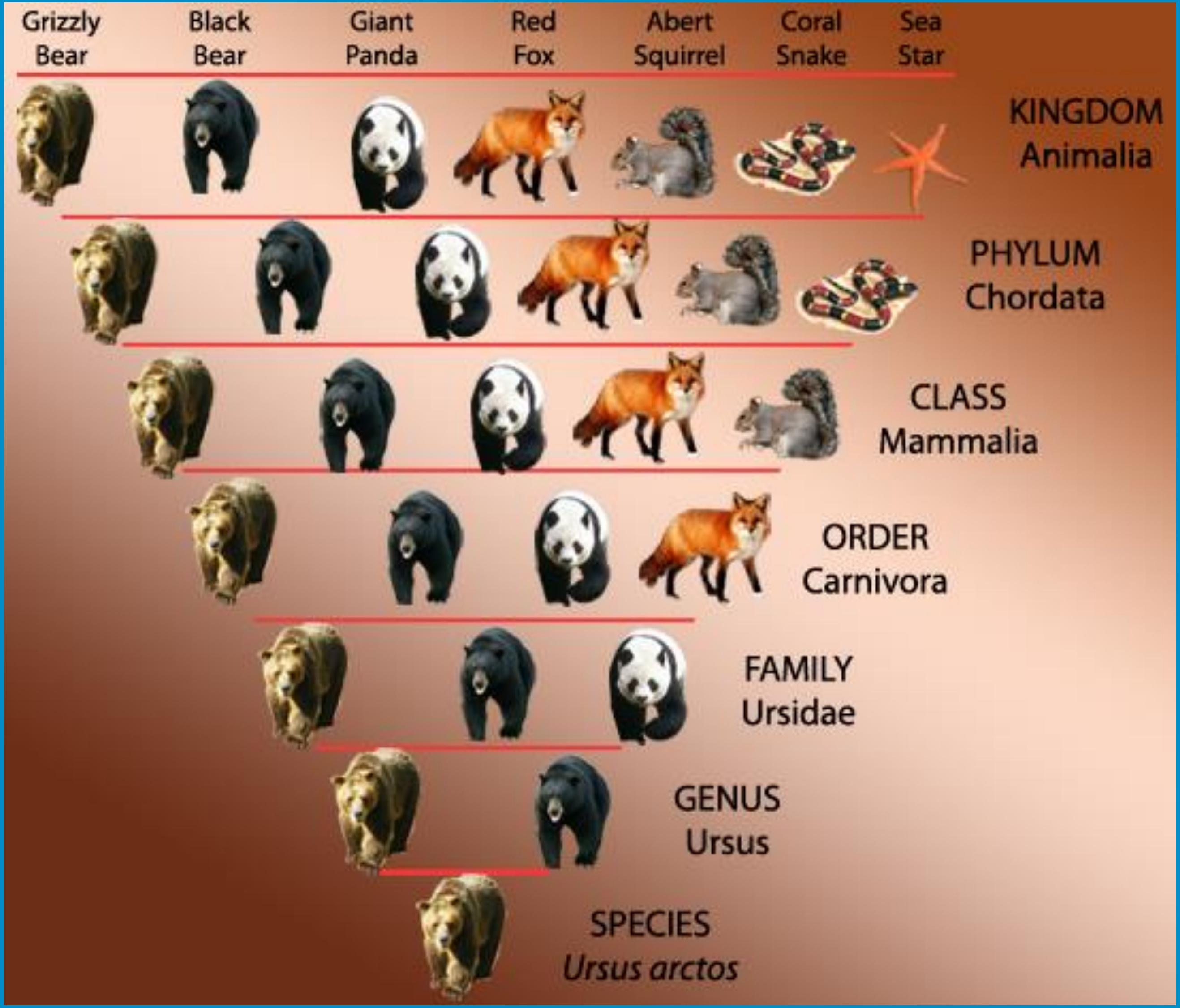


Fruit Classification



Yellow Bananas	Golden Pineapple	Black Grape
Green Plantains	Passion Fruits	Black berry
Tangerines	Bosc Pears	Blue berry
Hass Avocado	Anjou Pears	Straw berry

Animal Classification/Categorization



WE ARE HUMBER

Definition of Classification

- The method of arranging the organisms (objects or events) into groups is called classification. When we classify things, we put them into groups based on their characteristics/similarities.

- Is another fundamental supervised learning method for **prediction**,

- In classification learning, a classifier is presented with a set of examples (training set) that are already classified, and, from this training set, the classifier learns to **predict** where to assign the unseen example.

- Unlike clustering, classification uses predetermined labels to make classifications.

- Example classification method: Decision Trees

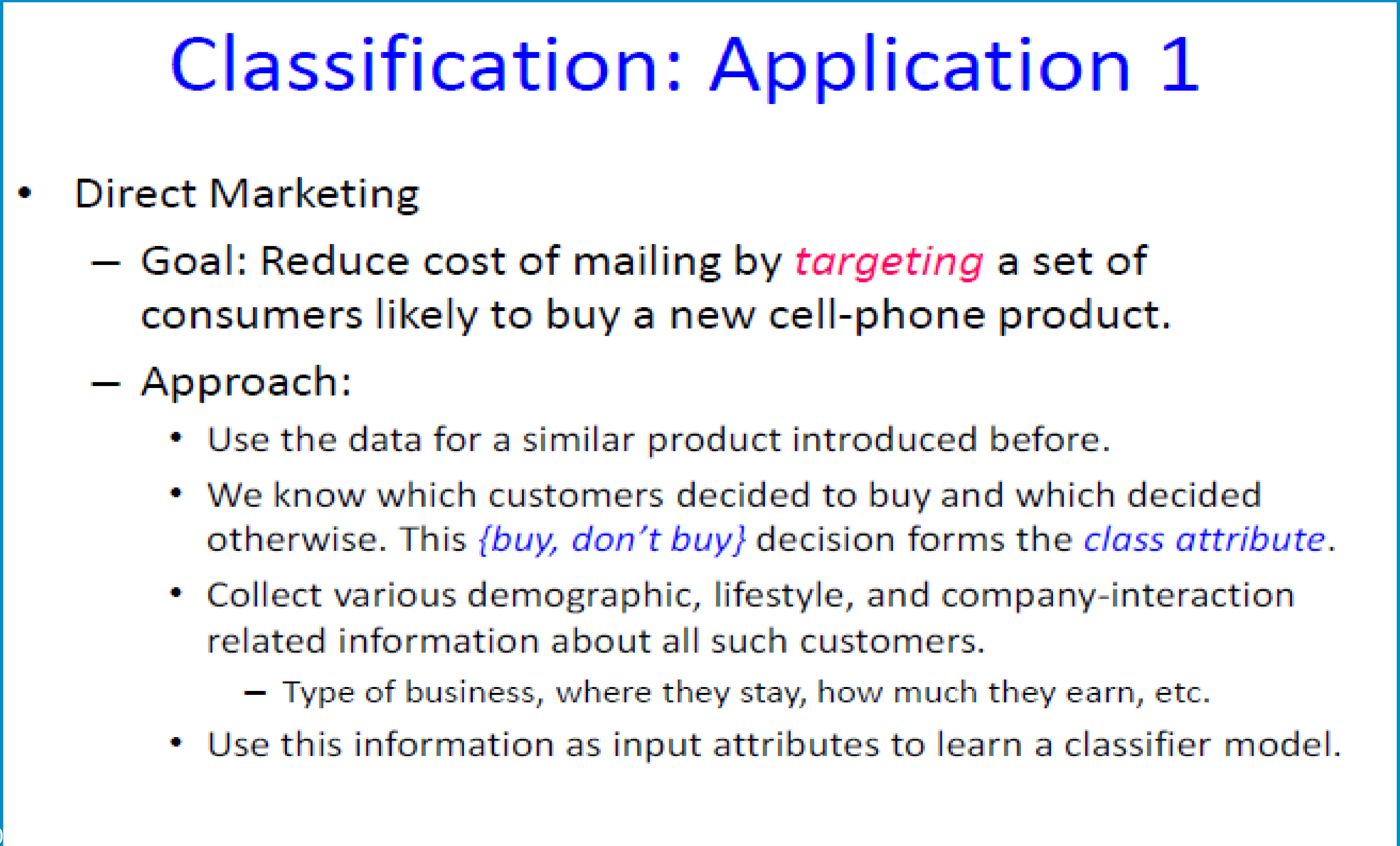
Classification

- Classification is a supervised learning approach
- Data set may be bi-class or multi-class.
- Practical examples:
  - Speech recognition
  - Handwriting recognition
  - Bio-metric identification
  - Document classification

Example of Classification Application

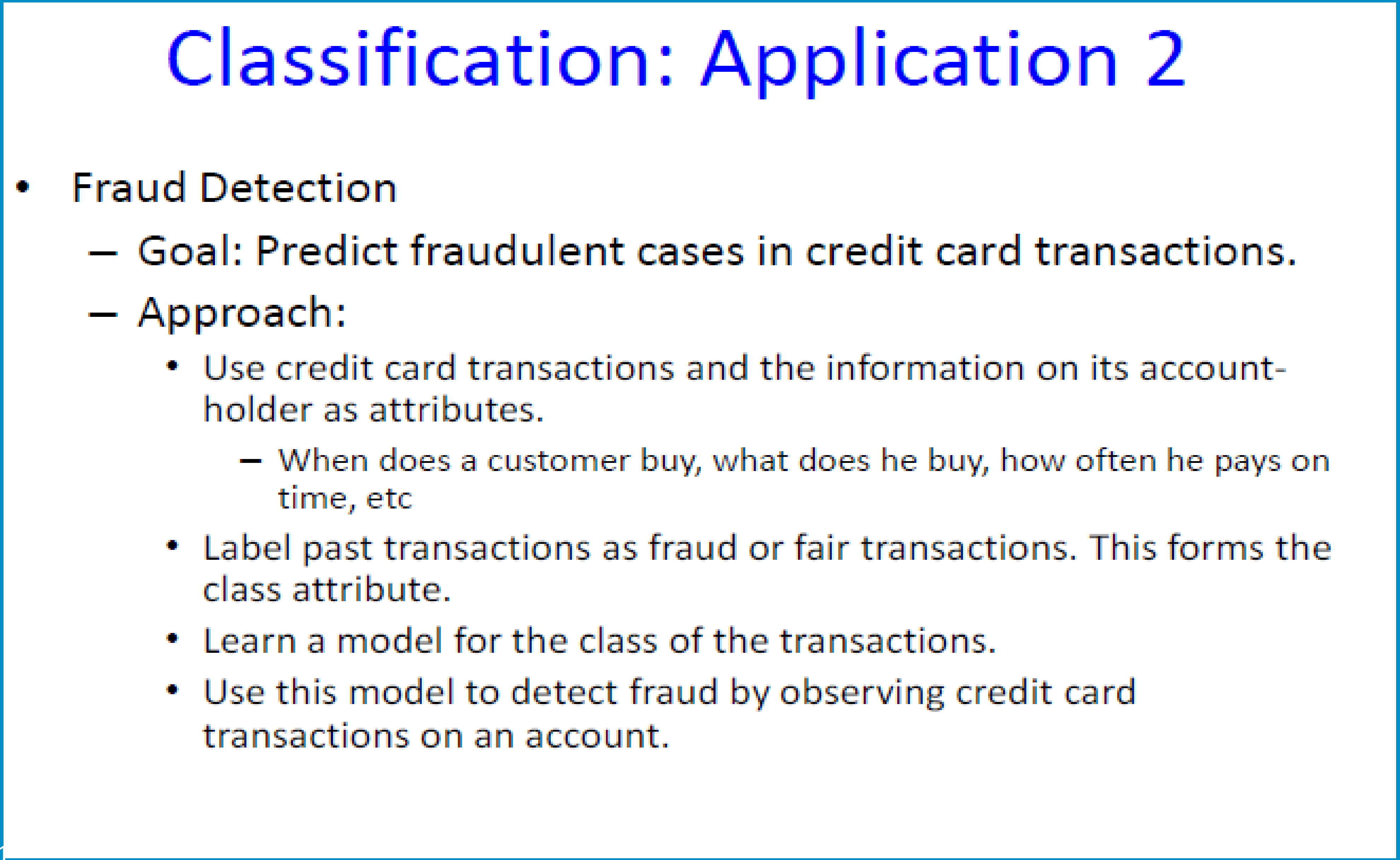
Following are the examples of cases where classification is used:

- A bank loan officer wants to analyze the data in order to know which customer (loan applicant) are risky or safe.
- A marketing manager at a company needs to analyze to guess if a customer with a given profile will buy a new computer.
- In both examples a model or classifier is constructed to predict categorical labels. These labels are risky or safe for loan application data and yes or no for marketing data.



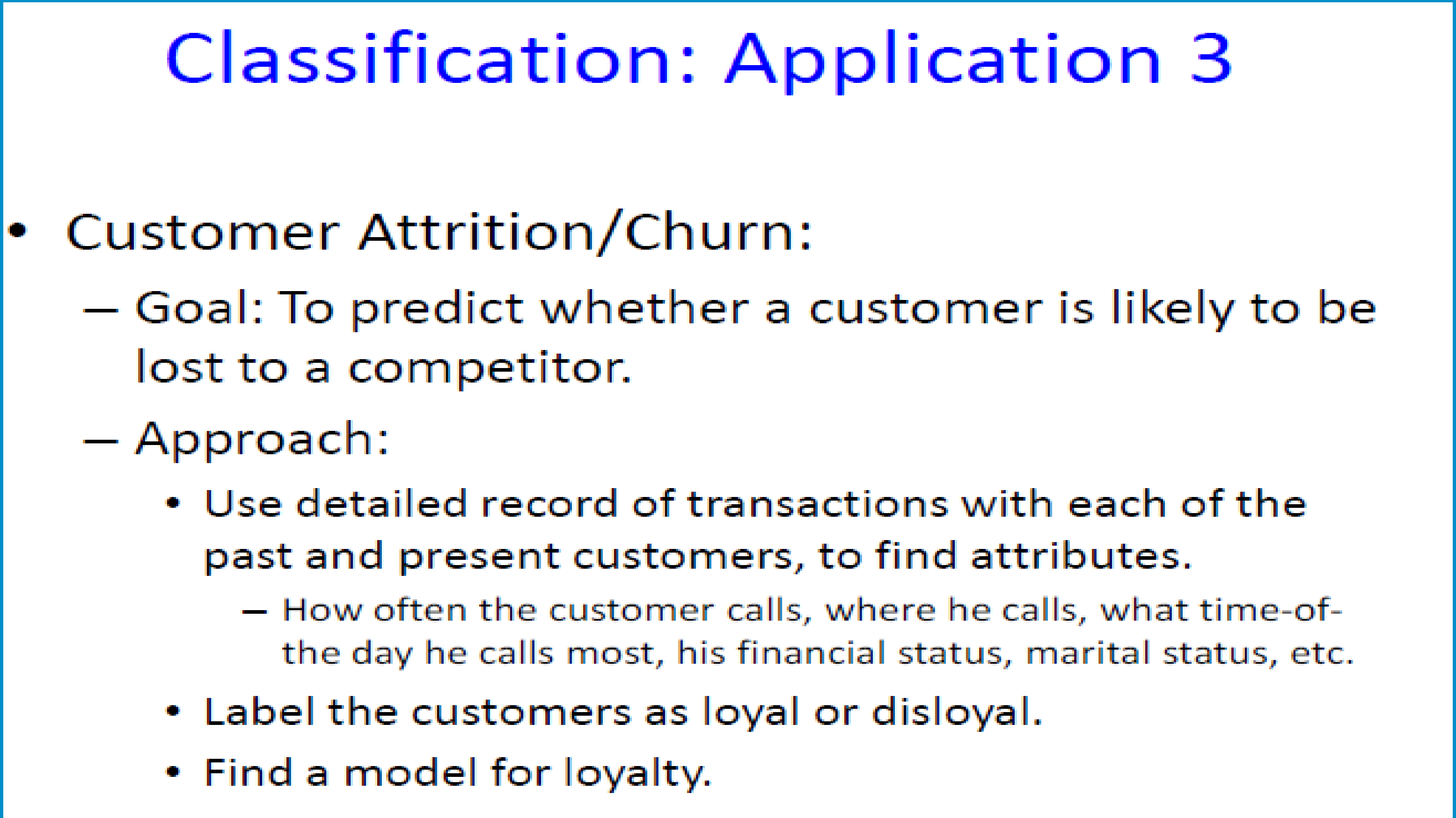
Classification: Application 1

- Direct Marketing
  - Goal: Reduce cost of mailing by *targeting* a set of consumers likely to buy a new cell-phone product.
  - Approach:
    - Use the data for a similar product introduced before.
    - We know which customers decided to buy and which decided otherwise. This *{buy, don't buy}* decision forms the *class attribute*.
    - Collect various demographic, lifestyle, and company-interaction related information about all such customers.
      - Type of business, where they stay, how much they earn, etc.
    - Use this information as input attributes to learn a classifier model.



Classification: Application 2

- Fraud Detection
  - Goal: Predict fraudulent cases in credit card transactions.
  - Approach:
    - Use credit card transactions and the information on its account-holder as attributes.
      - When does a customer buy, what does he buy, how often he pays on time, etc
    - Label past transactions as fraud or fair transactions. This forms the class attribute.
    - Learn a model for the class of the transactions.
    - Use this model to detect fraud by observing credit card transactions on an account.



Classification: Application 3

- Customer Attrition/Churn:
  - Goal: To predict whether a customer is likely to be lost to a competitor.
  - Approach:
    - Use detailed record of transactions with each of the past and present customers, to find attributes.
      - How often the customer calls, where he calls, what time-of-the day he calls most, his financial status, marital status, etc.
    - Label the customers as loyal or disloyal.
    - Find a model for loyalty.

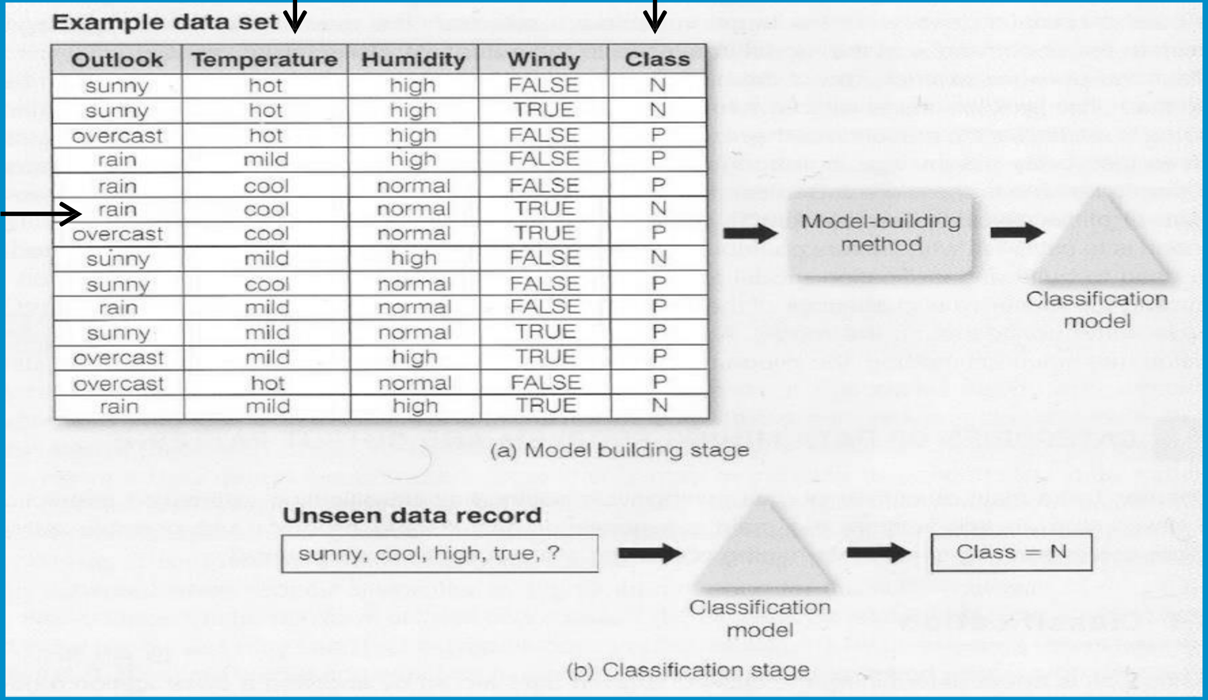
Classification

- Determining the class of a given data record by applying a classification model

Attribute/ Descriptive Feature
Measuring aspects of an instance

Instance
Individual, independent examples of a concept

Class/ Category
(kinds of things that can be learned)



Example data set

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	FALSE	N
sunny	hot	high	TRUE	N
overcast	hot	high	FALSE	P
rain	mild	high	FALSE	P
rain	cool	normal	FALSE	P
rain	cool	normal	TRUE	N
overcast	cool	normal	TRUE	P
sunny	mild	high	FALSE	N
sunny	cool	normal	FALSE	P
rain	mild	normal	FALSE	P
sunny	mild	normal	TRUE	P
overcast	mild	high	TRUE	P
overcast	hot	normal	FALSE	P
rain	mild	high	TRUE	N

(a) Model building stage

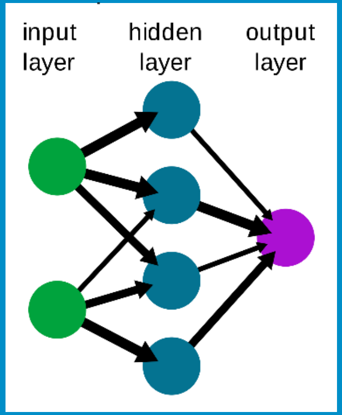
Unseen data record

sunny, cool, high, true, ?

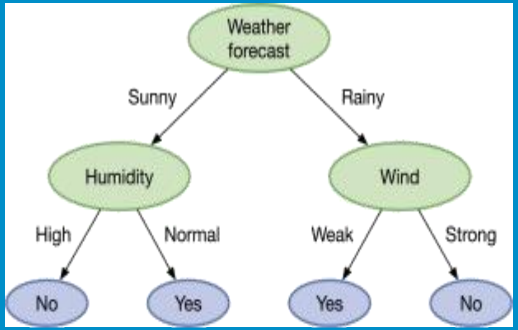
(b) Classification stage

Classification Models

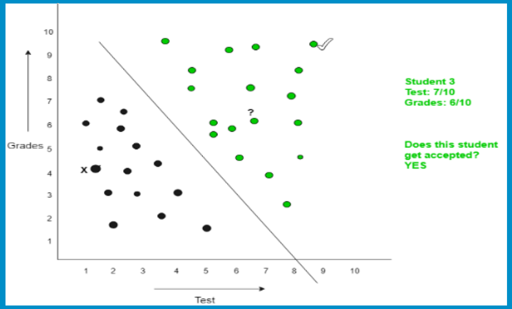
- **Various forms of classification models**



Neural Network



Decision Tree



Logistic Regression

Many more ...

Supervised Learning – Classification

- Decision Trees

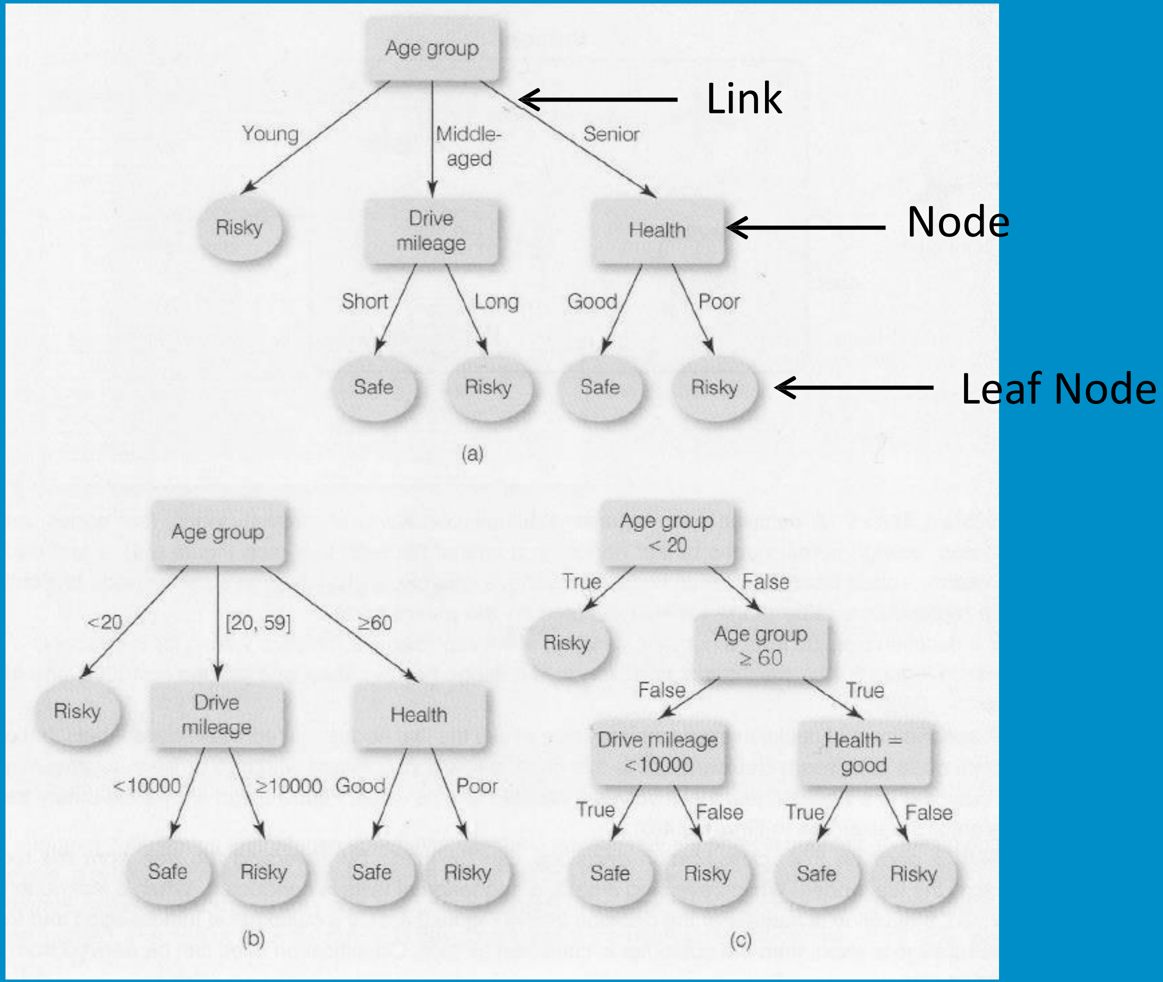
Decision Tree Classifier

- A good automatics rule discovery technique is the Decision Tree
- Produces a set of branching decisions that end in a classification.
- Works best on nominal attributes – numeric ones need to be split into bins

Decision Trees

- A decision tree is a representation of a function that maps a vector of attribute values to a single output value—a “decision.”
  - reaches its decision by performing a sequence of tests, starting at the **root** and following the appropriate branch until a leaf is reached.
  - each **internal node** in the tree corresponds to a test of the value of one of the input attributes
  - the **branches** from the node are labeled with the possible values of the attribute
  - the **leaf** nodes specify what value is to be returned by the function.
- Boolean decision tree is equivalent to a logical statement of the form:
  - $Output \Leftrightarrow (Path_1 \vee Path_2 \vee \cdots)$

Classification -Decision Trees



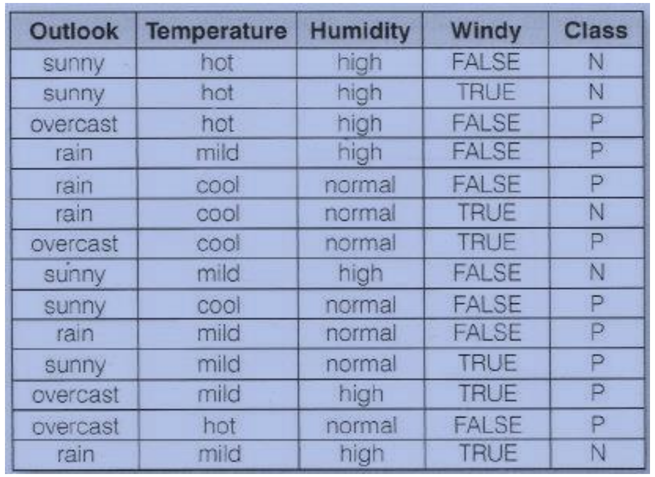
Link

Node

Leaf Node

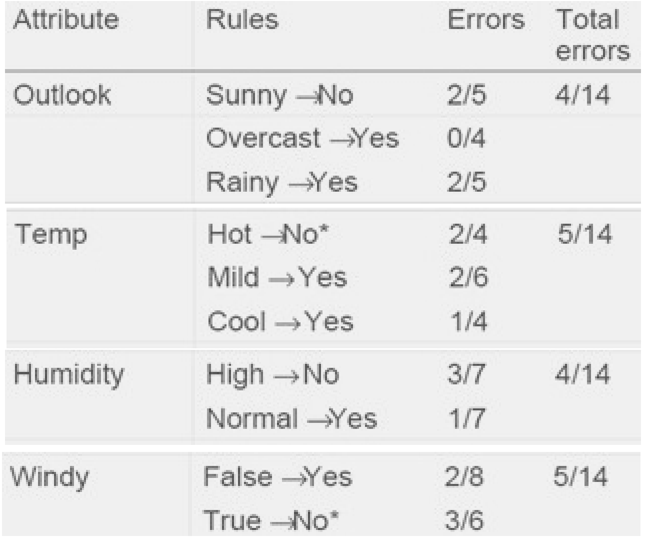
Classification -Decision Trees

Input Training Examples

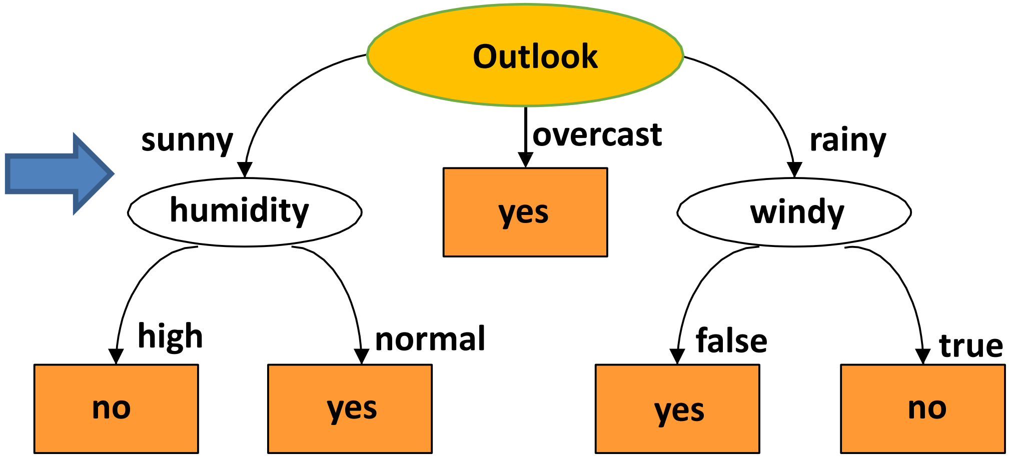


Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	FALSE	N
sunny	hot	high	TRUE	N
overcast	hot	high	FALSE	P
rain	mild	high	FALSE	P
rain	cool	normal	FALSE	P
rain	cool	normal	TRUE	N
overcast	cool	normal	TRUE	P
sunny	mild	high	FALSE	N
sunny	cool	normal	FALSE	P
rain	mild	normal	FALSE	P
sunny	mild	normal	TRUE	P
overcast	mild	high	TRUE	P
overcast	hot	normal	FALSE	P
rain	mild	high	TRUE	N

Which is the best attribute?

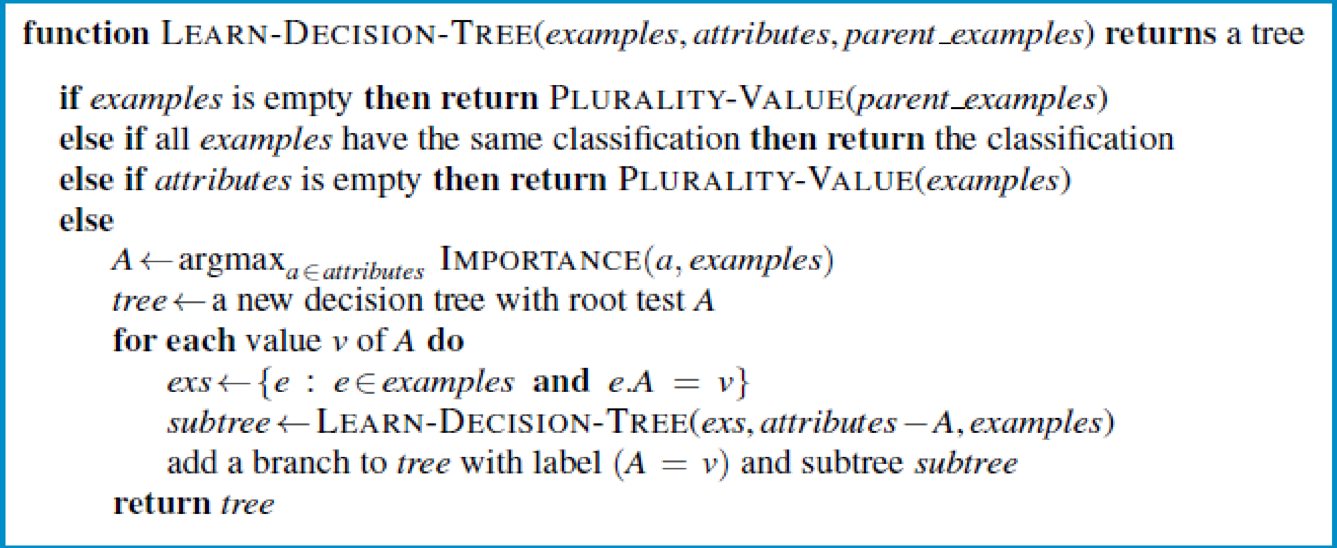


Attribute	Rules	Errors	Total errors
Outlook	Sunny →No	2/5	4/14
	Overcast →Yes	0/4	
	Rainy →Yes	2/5	
Temp	Hot →No*	2/4	5/14
	Mild →Yes	2/6	
	Cool →Yes	1/4	
Humidity	High →No	3/7	4/14
	Normal →Yes	1/7	
Windy	False →Yes	2/8	5/14
	True →No*	3/6	



Internal Nodes: **Attributes**
Links: **Attribute values**
Leaf Nodes: **Class labels**

Decision Trees



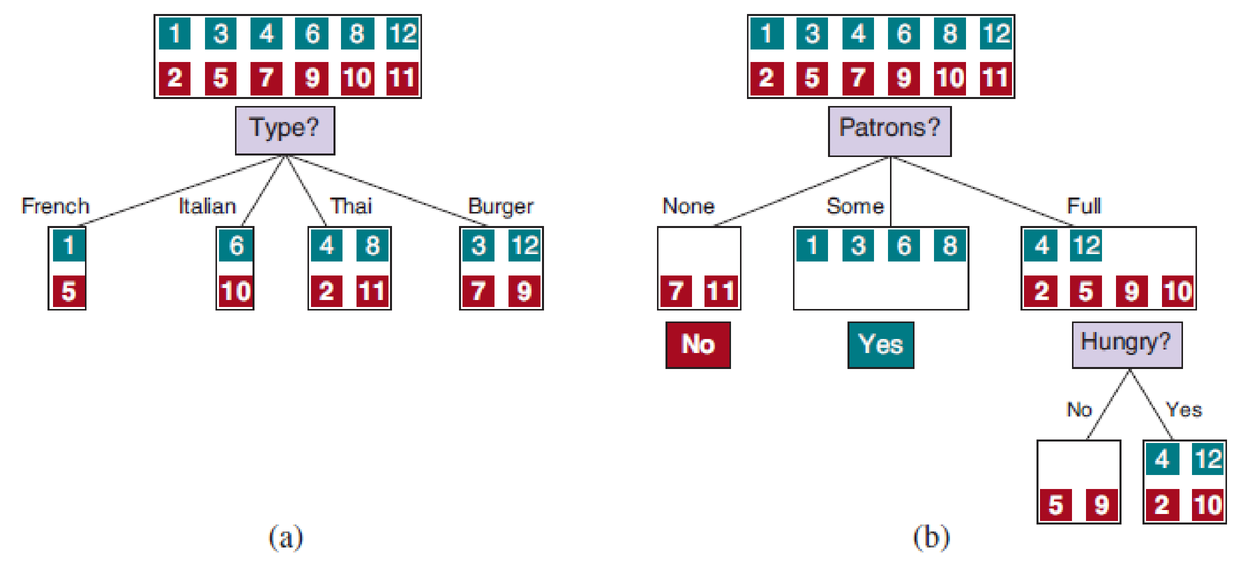
```
function LEARN-DECISION-TREE(examples, attributes, parent_examples) returns a tree

  if examples is empty then return PLURALITY-VALUE(parent_examples)
  else if all examples have the same classification then return the classification
  else if attributes is empty then return PLURALITY-VALUE(examples)
  else
     $A \leftarrow \text{argmax}_{a \in attributes}$ IMPORTANCE($a$, examples)
     tree ← a new decision tree with root test $A$
     for each value $v$ of $A$ do
        $exs \leftarrow \{e : e \in examples$ and $e.A = v\}$
        subtree ← LEARN-DECISION-TREE(exs, attributes − A, examples)
        add a branch to tree with label ($A = v$) and subtree subtree
     return tree
```

The decision tree learning algorithm. The function PLURALITY-VALUE selects the most common output value among a set of examples, breaking ties randomly.

Aim: find a small tree consistent with the training examples

Idea: (recursively) choose “most significant” attribute as root of (sub)tree

Decision Trees



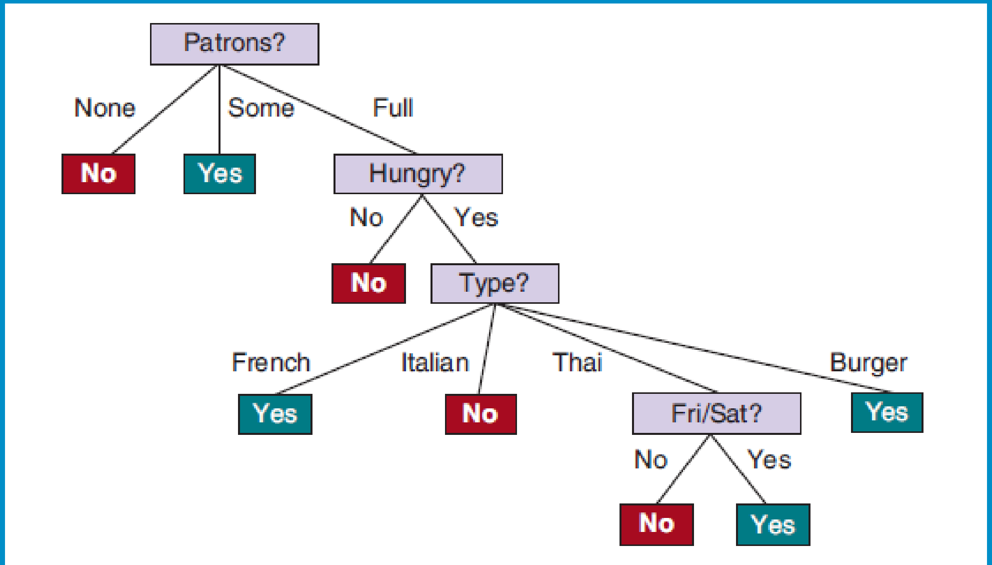
Splitting the examples by testing on attributes. At each node we show the positive (light boxes) and negative (dark boxes) examples remaining.

(a) Splitting on *Type* brings us no nearer to distinguishing between positive and negative examples. (four possible outcomes, each of which has the same number of positive as negative examples)

(b) Splitting on *Patrons* does a good job of separating positive and negative examples. After splitting on *Patrons*, *Hungry* is a fairly good second test

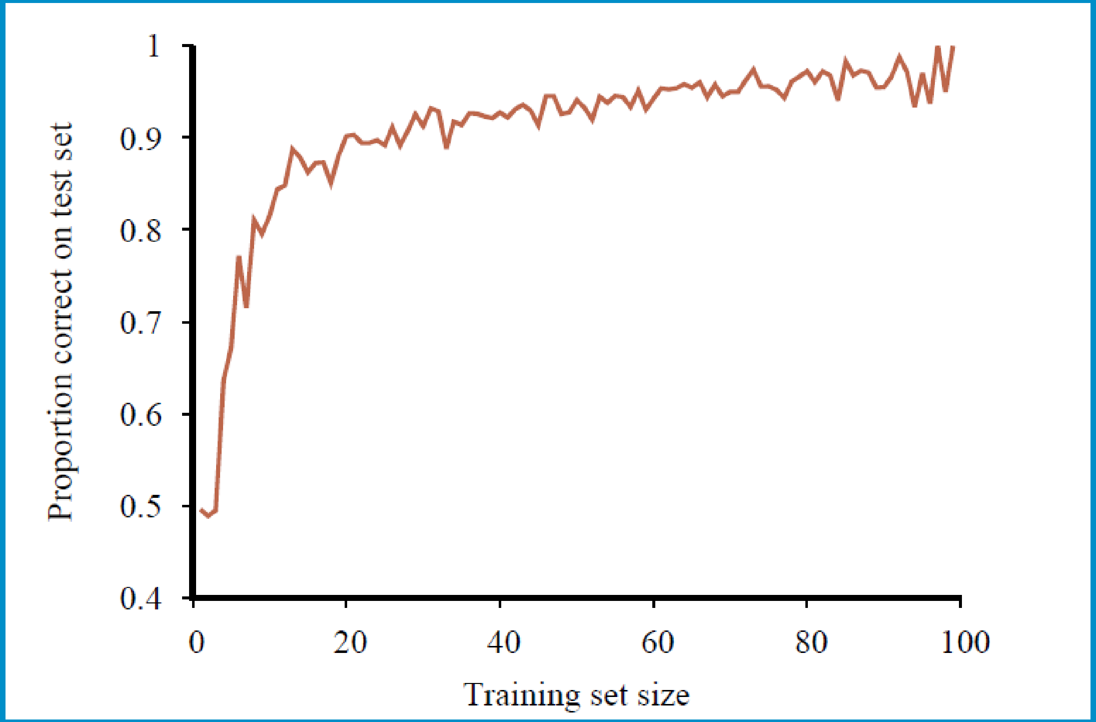
Decision Trees

- The decision tree induced from the 12-example training set.



Decision Trees

- The learning curve for the decision tree learning algorithm on 100 randomly generated examples in the restaurant domain. Each data point is the average of 20 trials.



Decision Trees

Choosing attribute tests

- **Entropy**: measure of the uncertainty of a random variable;
  - the more information, the less entropy
  - fundamental quantity in information theory
- In general, the entropy of a random variable *V* with values v_k having probability P(v_k)

$$\text{Entropy:} \quad H(V) = \sum_k P(v_k) \log_2 \frac{1}{P(v_k)} = -\sum_k P(v_k) \log_2 P(v_k)\,.$$

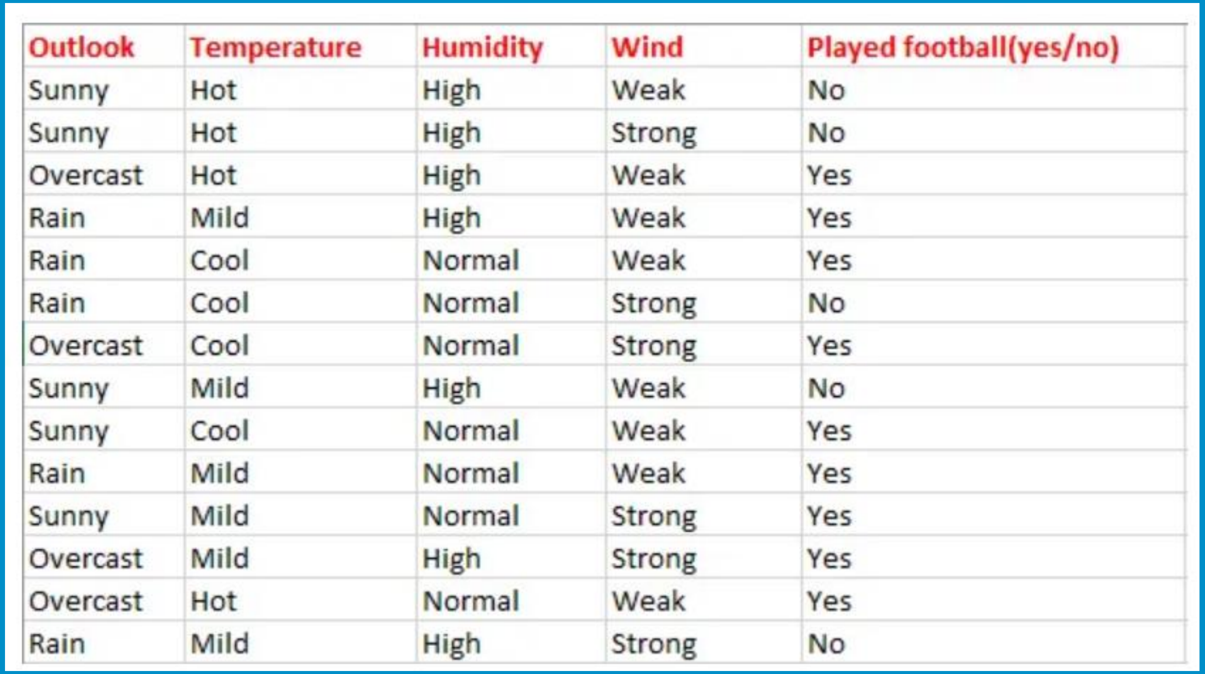
- If a training set contains *p* positive examples and *n* negative examples, then the entropy of the goal attribute on the whole set is $H(Goal) = B\left(\frac{p}{p+n}\right)$.
- The expected entropy remaining after testing attribute *A is Remainder (A)*
- The **information gain** from the attribute test on *A* is the expected reduction in entropy:

$$Gain(A) = B\left(\tfrac{p}{p+n}\right) - Remainder(A)\,.$$

- The information that you gain, by knowing the value of an attribute
- So, the "most informative" attribute is the attribute with the highest InfoGain

Example

- 4 independent variables to determine the dependent variable



Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

- Dependent variable:
  - Play football or not.
- Independent variables:
  - Outlook, Temperature, Humidity, and Wind

Construct Decision Tree, Example

- Find the parent node for our decision tree:

1. Find the entropy of the class variable.

   Total 14 yes/no. Out of which 9 yes and 5 no

   E(S) = -[(9/14)log(9/14) + (5/14)log(5/14)] = 0.94

Construct Decision Tree, Example

- Find the parent node for our decision tree:

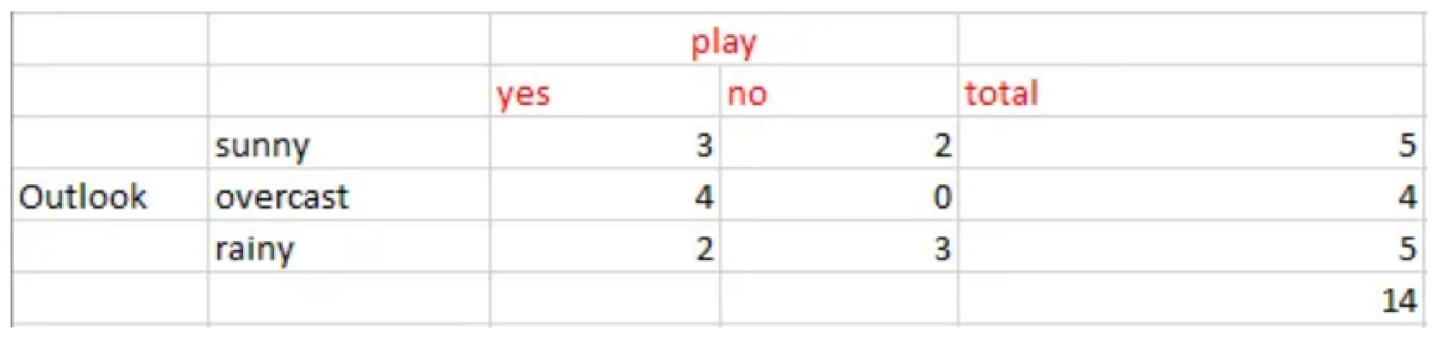
2. Calculate average weighted entropy.

Total of weights of each feature multiplied by probabilities

E(S, outlook) = (5/14)*E(3,2) + (4/14)*E(4,0) + (5/14)*E(2,3)

= (5/14)(-(3/5)log(3/5)-(2/5)log(2/5))+ (4/14)(0) + (5/14)((2/5)log(2/5)-(3/5)log(3/5))

= 0.693



		play		
		yes	no	total
	sunny	3	2	5
Outlook	overcast	4	0	4
	rainy	2	3	5
				14

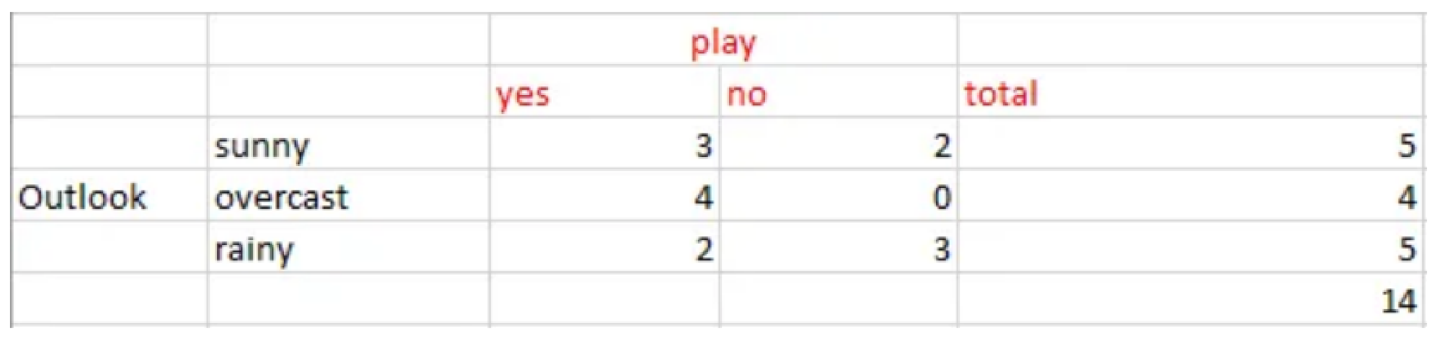
Construct Decision Tree, Example

- Find the parent node for our decision tree:

3. Find the information gain

Difference between parent entropy and average weighted entropy

IG(S, outlook) = 0.94 - 0.693 = 0.247



		play		
		yes	no	total
	sunny	3	2	5
Outlook	overcast	4	0	4
	rainy	2	3	5
				14

Construct Decision Tree, Example

- Find the parent node for our decision tree:

4. Similarly find Information gain for Temperature, Humidity, and Windy.

IG(S, Temperature) = 0.940 - 0.911 = 0.029

IG(S, Humidity) = 0.940 - 0.788 = 0.152

IG(S, Windy) = 0.940 - 0.8932 = 0.048

Construct Decision Tree, Example

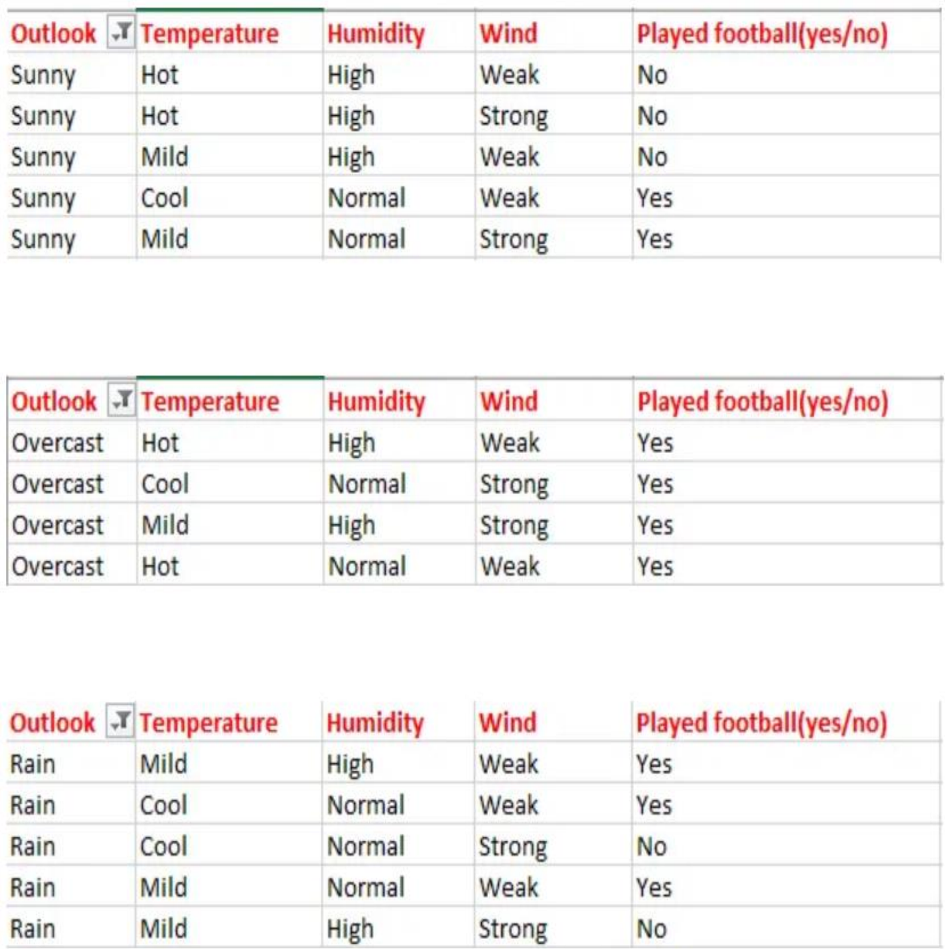
- Find the parent node for our decision tree:

5. Select the feature having the largest entropy gain.

→ **Outlook**: selected as the root node of our decision tree.

Construct Decision Tree, Example

→ Now our data look as follows



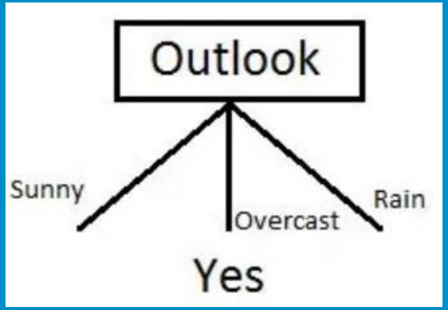
Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes

Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Overcast	Hot	High	Weak	Yes
Overcast	Cool	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes

Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Rain	Mild	Normal	Weak	Yes
Rain	Mild	High	Strong	No

- overcast contains only rows of class 'Yes'.
  - – If outlook is overcast then decide to play.

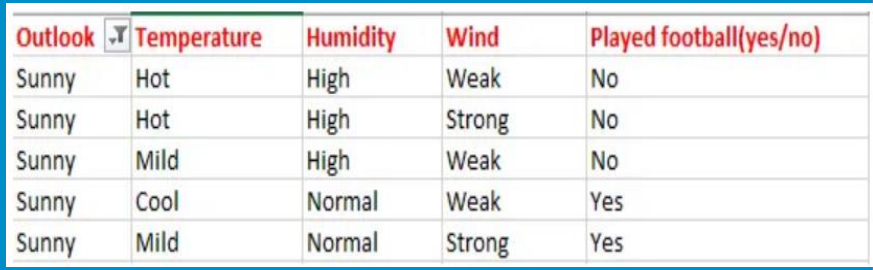
→ Now the decision tree



Construct Decision Tree, Example

- Find the next node for our decision tree:
- Find node under sunny.
  - Determine which of the following Temperature, Humidity or Wind has higher information gain
  - Select the feature having the largest entropy gain.

Construct Decision Tree, Example



Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes

- Find the next node for our decision tree:

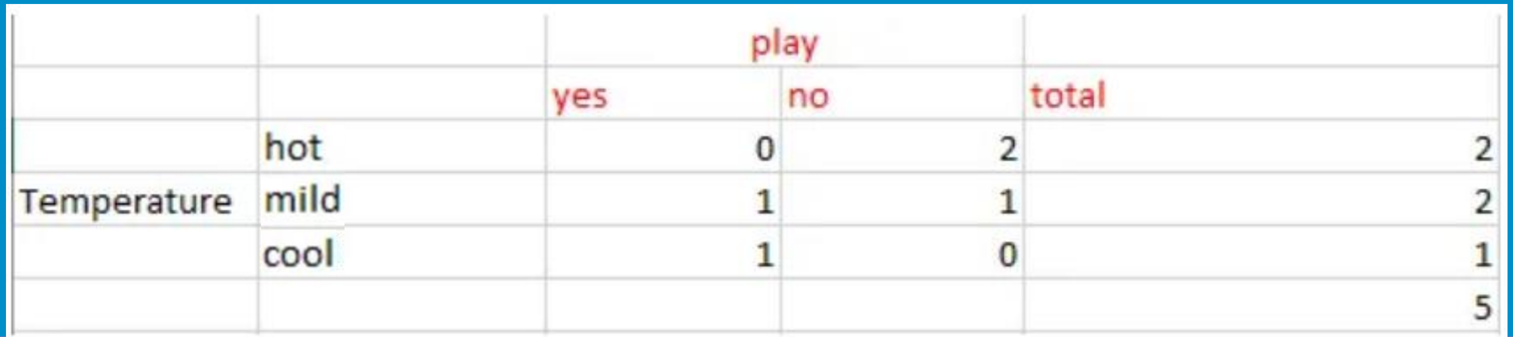
1. Calculate parent entropy E(sunny)

E(sunny) = (-(3/5)log(3/5)-(2/5)log(2/5)) = 0.971.

2. Calculate the information gain of Temperature IG(sunny, Temperature)

E(sunny, Temperature) = (2/5)*E(0,2) + (2/5)*E(1,1) + (1/5)*E(1,0)=2/5=0.4

IG(sunny, Temperature) = 0.971–0.4 =0.571



		play		
		yes	no	total
	hot	0	2	2
Temperature	mild	1	1	2
	cool	1	0	1
				5

Construct Decision Tree, Example

- Find the next node for our decision tree

3. Similarly we get information gain for:

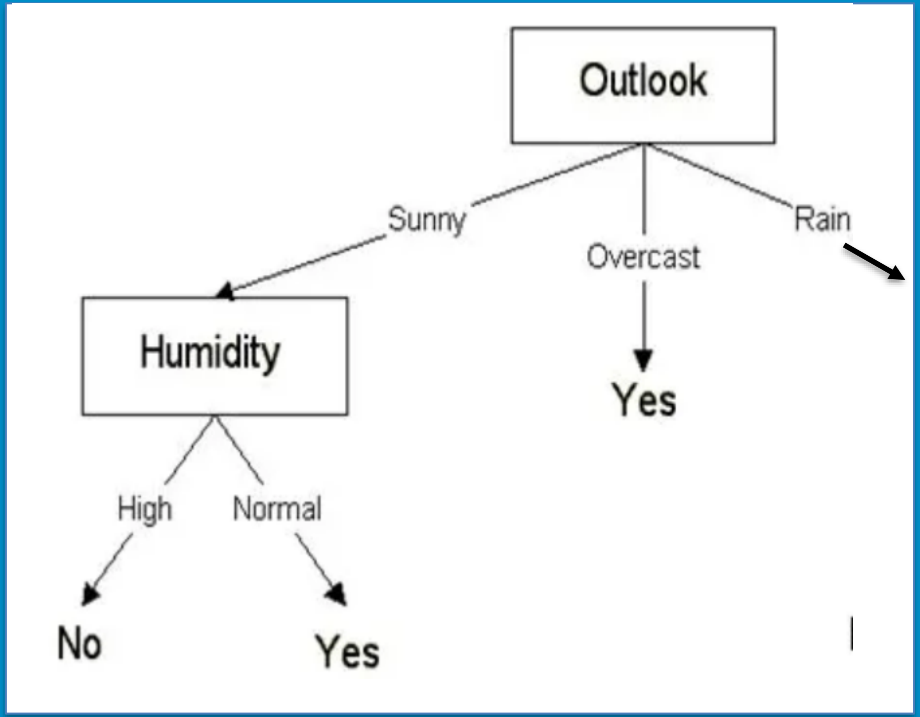
   IG(sunny, Humidity) = 0.971

   IG(sunny, Windy) = 0.020

4. Find the largest information gain

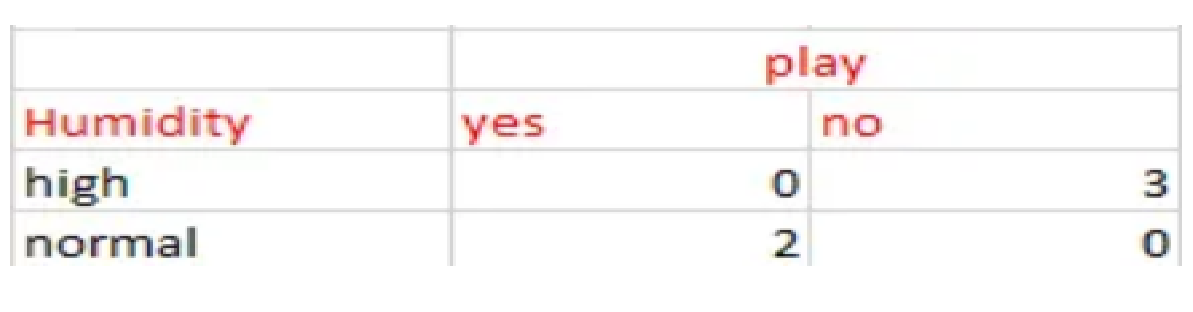
   → IG(sunny, Humidity) is the largest value

   → Humidity is the node that comes under outlook



Construct Decision Tree, Example

- For humidity:
  - play is *yes* if humidity is normal and *no* if it is high.

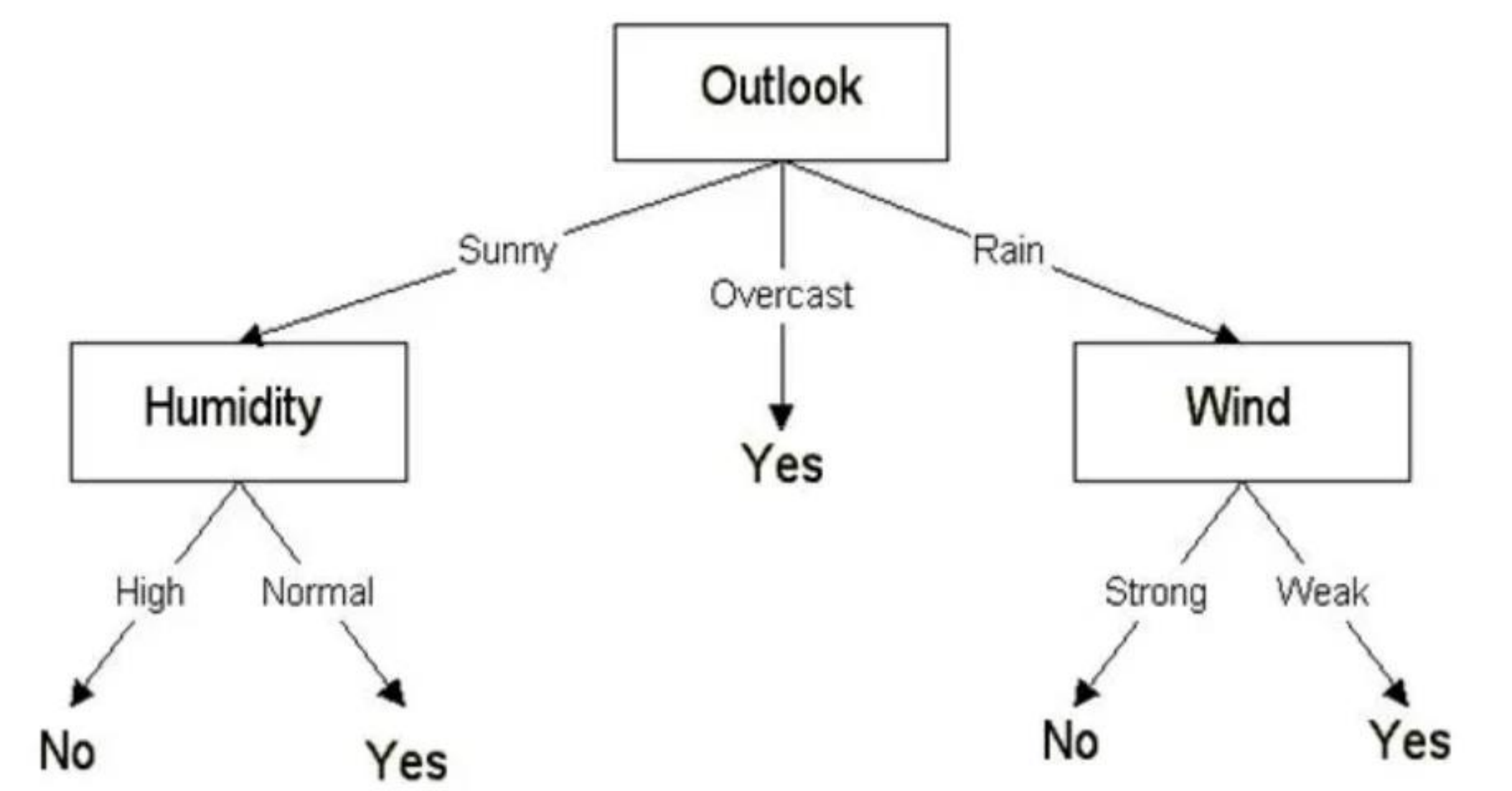


	play	
Humidity	yes	no
high	0	3
normal	2	0

- Similarly, find the nodes under rainy

Construct Decision Tree, Example

- Finally, the decision tree looks as follows:



Decision Trees

Broadening the applicability of decision trees

- Decision trees can be made more widely useful by handling the following complications:
  - Missing data
  - Continuous and multivalued input attributes
  - Continuous-valued output attribute
- Decision trees are also unstable in that adding just one new example can change the test at the root, which changes the entire tree

Decision Tree Classifier - Use Cases

- When a series of questions (yes/no) are answered to arrive at a classification
  - Biological species classification
  - Checklist of symptoms during a doctor’s evaluation of a patient
- When “if-then” conditions are preferred to linear models.
  - Customer segmentation to predict response rates to marketing and promotions.
  - Financial decisions such as loan approval
  - Fraud detection
- Short Decision Trees are the most popular "weak learner" or “base learners” in ensemble learning techniques

Supervised Learning - Regression

- Linear Regression
- Logistic Regression

Regression

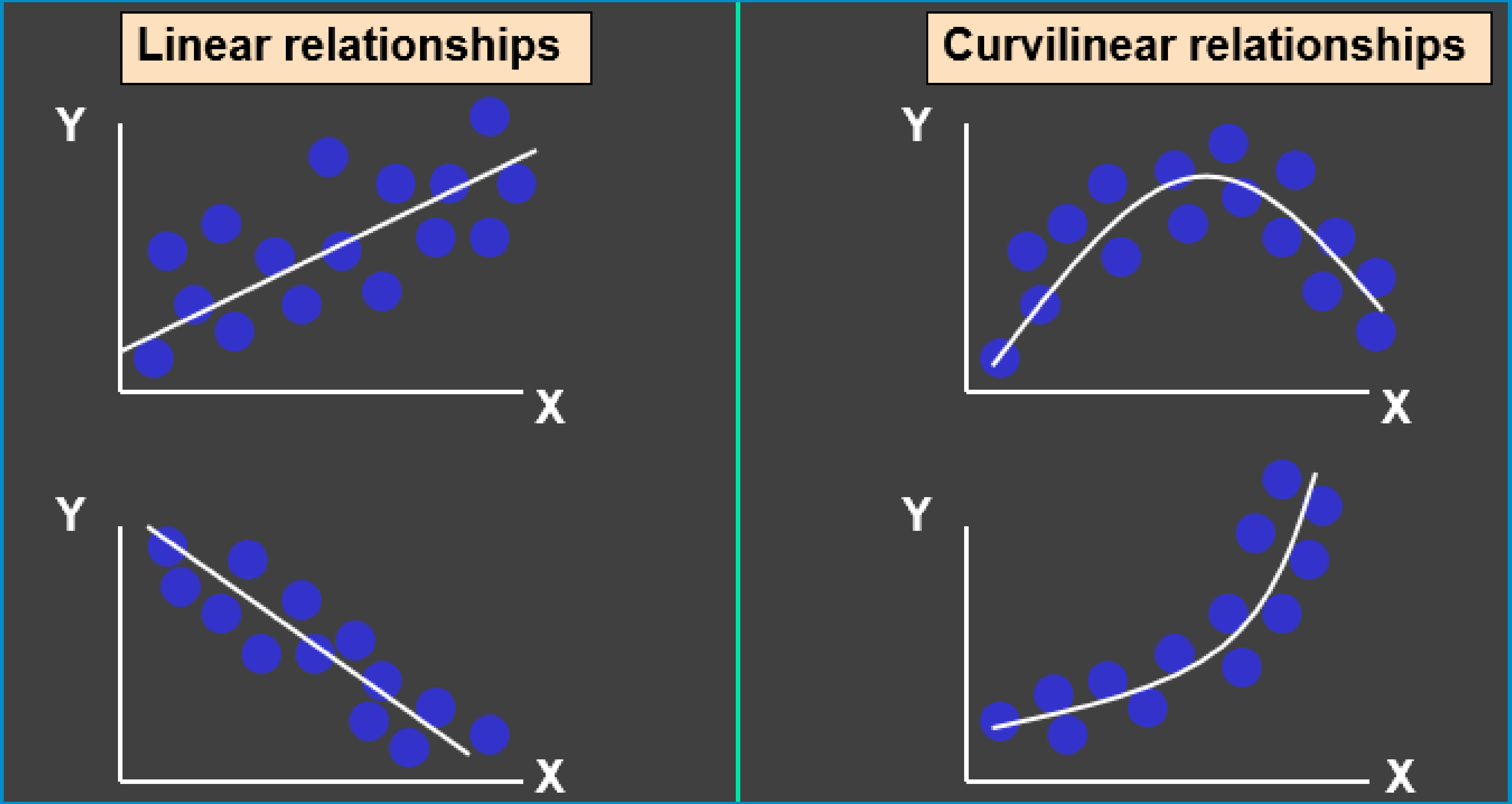
- While decision trees are excellent for categorical outcomes and classification, we'll now shift our focus to addressing continuous numerical outcomes
- Decision trees are suitable for classifying data into discrete categories, whereas regression analysis deals with predicting continuous values

Correlation vs. Regression

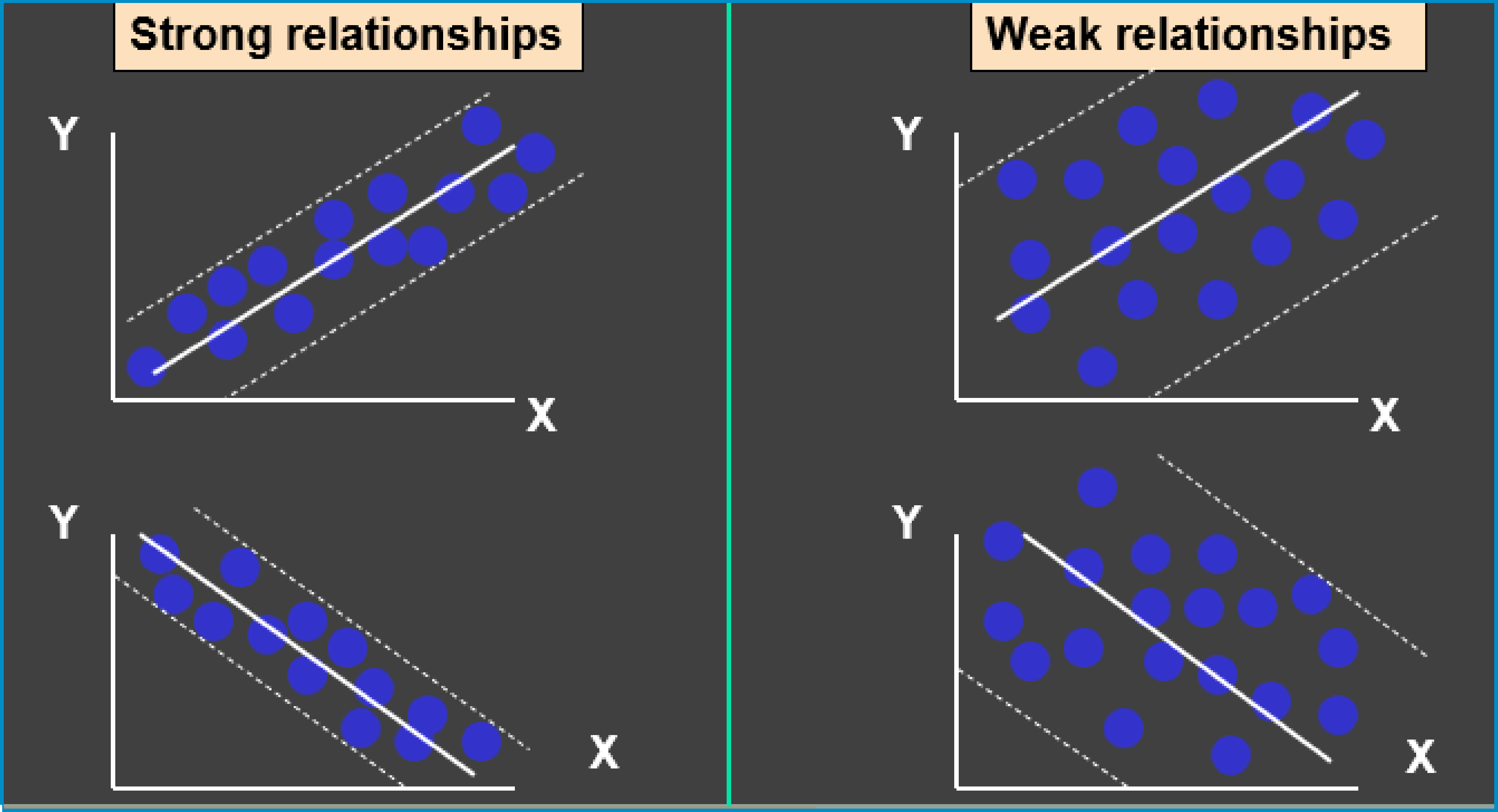
- A scatter plot can be used to show the relationship between two variables

- Correlation analysis is used to measure the strength of the association (linear relationship) between two variables
  - Correlation is only concerned with strength of the relationship
  - No causal effect is implied with correlation

Regression Models



Types of Relationships



Introduction to Regression Analysis

- Regression analysis is used to:
  - Predict the value of a dependent variable based on the value of at least one independent variable
  - Explain the impact of changes in an independent variable on the dependent variable
- Dependent variable: the variable we wish to predict or explain
- Independent variable: the variable used to predict or explain the dependent variable

Regression

- Regression focuses on the relationship between an outcome and its input variables.
  - In other words, we don't just predict the outcome, we also have a sense of how changes in individual drivers affect the outcome.
- The outcome can be continuous or discrete.
  - When it's discrete, we are predicting the probability that the outcome will occur.

Example Questions:

  - I want to predict the lifetime value (LTV) of this customer (and understand what drives LTV).
  - I want to predict the probability that this loan will default (and understand what drives default).

- **Examples: Linear Regression, Logistic Regression**

Linear Regression -What is it?

- Used to estimate a continuous value as a linear (additive) function of other variables
  - Income (outcome) as a function of years of education, age, gender
  - House price (outcome) as function of median home price in neighborhood, square footage, number of bedrooms/bathrooms
  - Neighborhood house sales in the past year based on economic indicators (unemployment, stock price, etc.)
- Input variables can be continuous or discrete.
- Output:
  - A set of coefficients that indicate the relative impact of each driver.
  - A linear expression for predicting outcome as a function of drivers.

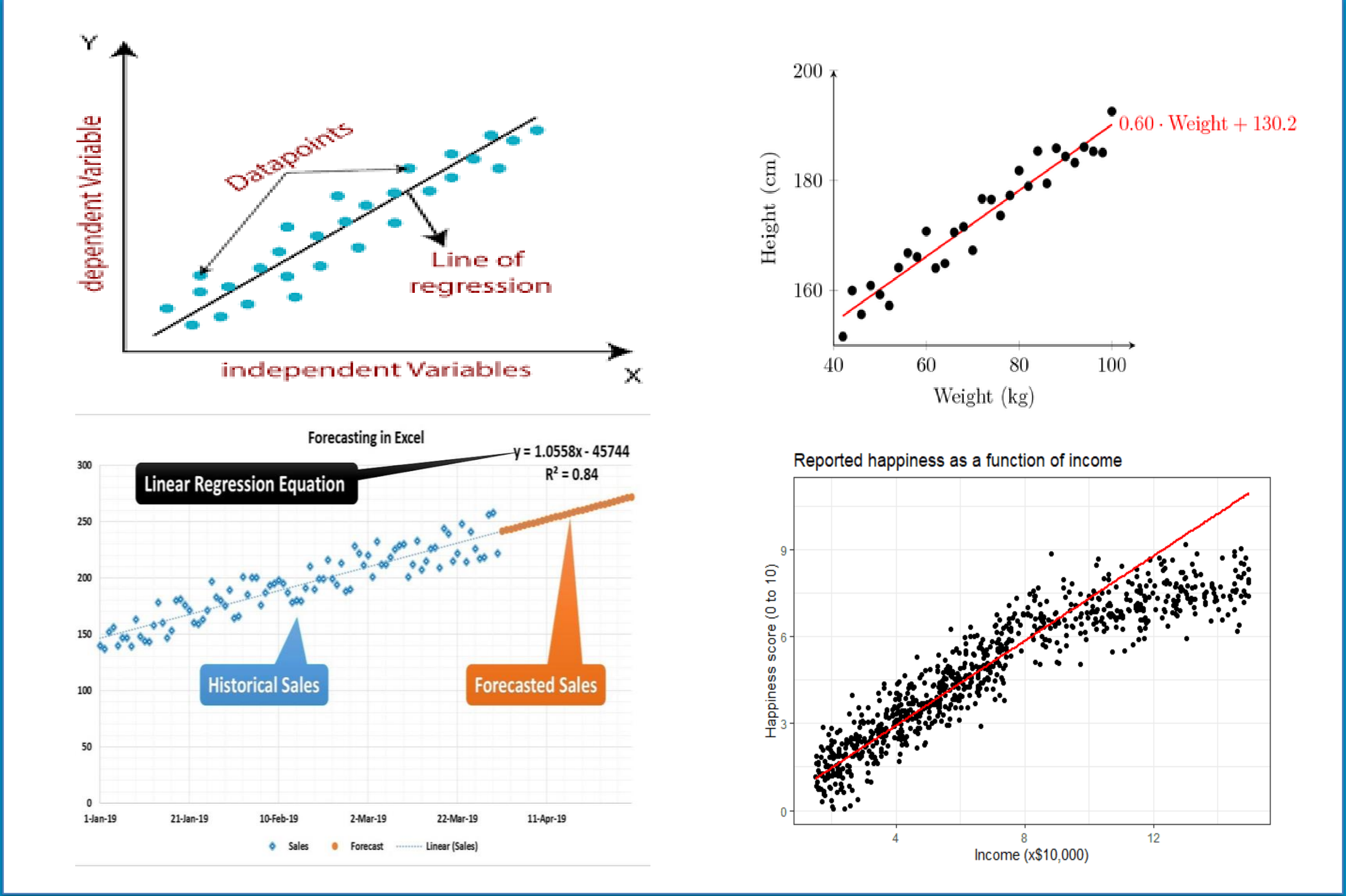
Linear Regression - Use Cases

- The preferred method for almost any problem where we are predicting a continuous outcome
  - Try this first; if it fails, then try something more **complicated**
- Examples:
  - Customer lifetime value
  - Home value
  - Loss given default on loan
  - Income as a function of demographics

Other examples:

- Look at past years' sales orders and advertising campaigns to decide where and how you will spend this year's advertising budget
- Identify the relationship between important variables that affect your business or organization

Examples:



Y
dependent Variable
Datapoints
Line of regression
independent Variables
X
200
180
160
Height (cm)
0.60 · Weight + 130.2
40
60
80
100
Weight (kg)
Forecasting in Excel
Linear Regression Equation
y = 1.0558x - 45744
R^2 = 0.84
300
250
200
150
100
50
0
Historical Sales
Forecasted Sales
1-Jan-19
21-Jan-19
10-Feb-19
2-Mar-19
22-Mar-19
11-Apr-19
Sales
Forecast
Linear (Sales)
Reported happiness as a function of income
Happiness score (0 to 10)
9
6
3
0
4
8
12
Income (x$10,000)

Simple Linear Regression Model

- Only **one** independent variable, X
- Relationship between X and Y is described by a linear function
- Changes in Y are assumed to be related to changes in X

Simple Linear Regression Model

Population Y intercept

Population Slope Coefficient

Independent Variable

Random Error term

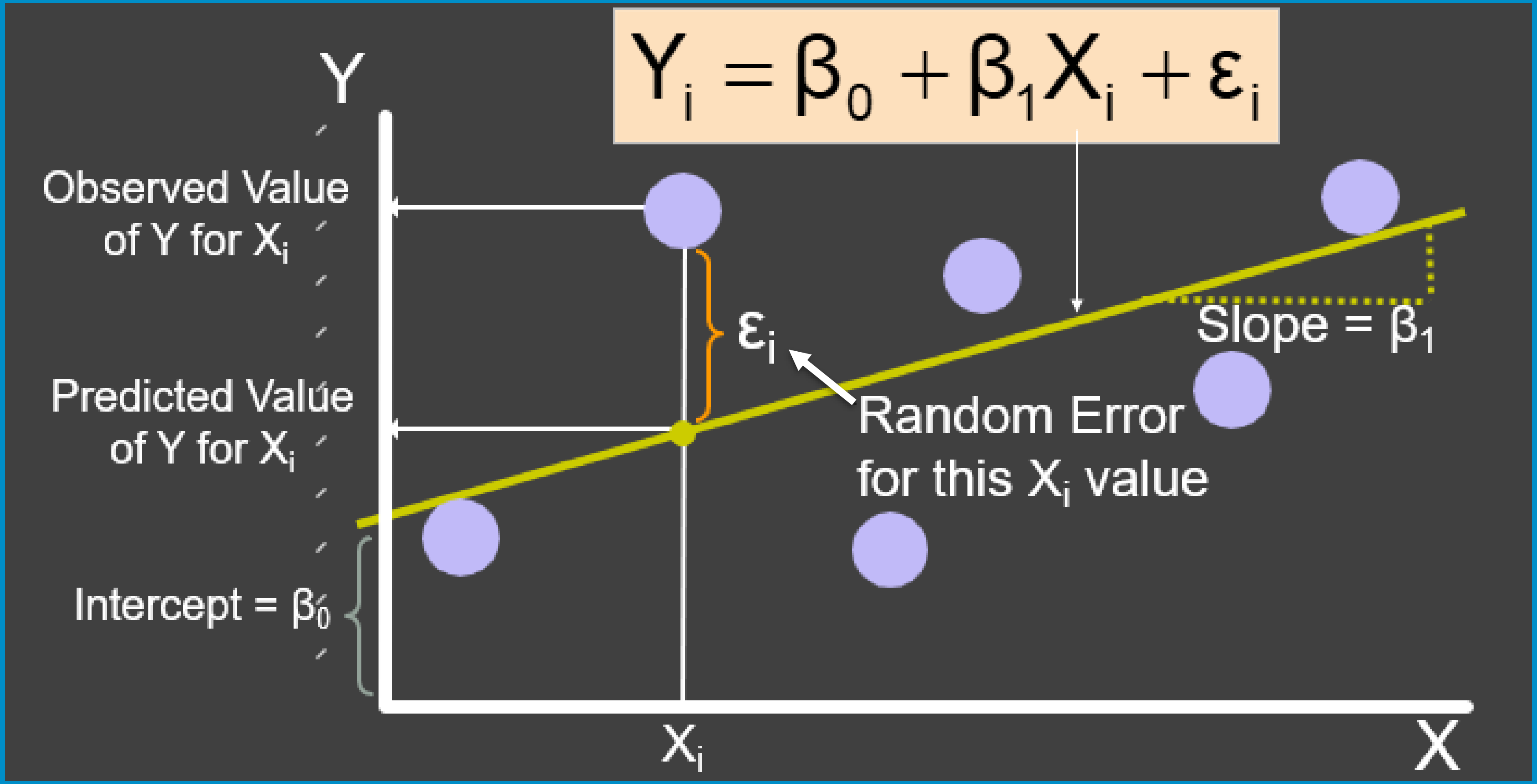
Dependent Variable

$$Y_i = \underbrace{\beta_0 + \beta_1 X_i}_{\text{Linear component}} + \underbrace{\varepsilon_i}_{\text{Random Error component}}$$

What do the Coefficients b_i Mean?

- Change in $\boldsymbol{y}$ as a function of unit change in $\boldsymbol{x}_i$
  - all other things being equal
- Example: income in units of \$10K, years in age, b_{age}= 2
  - For the same gender, years of education, and state of residence, a person's income increases by 2 units (20K) for every year older
- Standard packages also report the significance of the b_i: probability that, in reality, $b_i = 0$
  - b_i "significant" if P($b_i = 0$) is small

Simple Linear Regression Model



$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

Y
Observed Value of Y for X_i
Predicted Value of Y for X_i
Intercept = β_0
ε_i
Random Error for this X_i value
Slope = β_1
X_i
X

Simple Linear Regression Equation (Prediction Line)

The simple linear regression equation provides an estimate of the population regression line

Estimated (or predicted) Y value for observation i

Estimate of the regression intercept

Estimate of the regression slope

$$\hat{Y}_i = b_0 + b_1 X_i$$

Value of X for observation i

Simple Linear Regression Example

A real estate agent wishes to examine the relationship between the selling price of a home and its size (measured in square feet)

- A random sample of 10 houses is selected
  - Dependent variable (Y) = house price in $1000s
  - Independent variable (X) = square feet

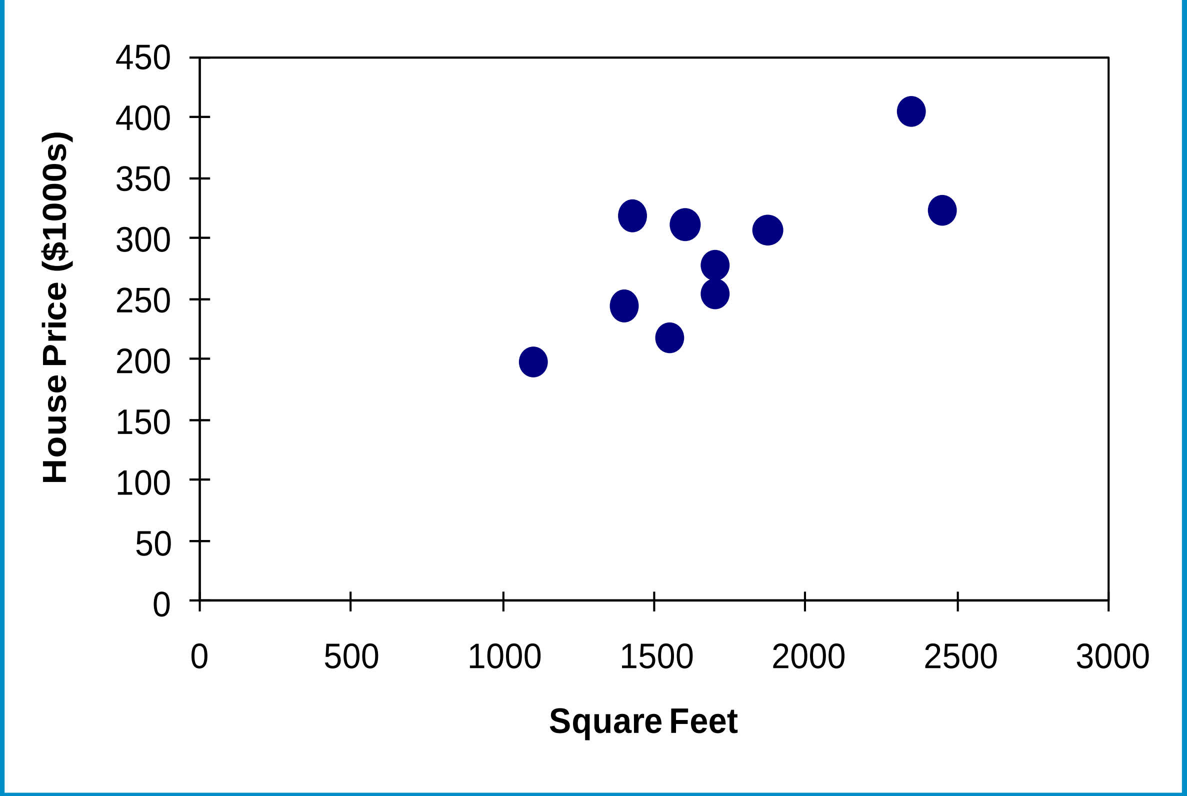
WE ARE HUMBER

Simple Linear Regression Example: Data

House Price in $1000s (Y)	Square Feet (X)
245	1400
312	1600
279	1700
308	1875
199	1100
219	1550
405	2350
324	2450
319	1425
255	1700

Simple Linear Regression Example: Scatter Plot

House price model: Scatter Plot



Linear model assumptions

- Linear regression analysis is based on six fundamental assumptions:

1. **Linearity:** The dependent and independent variables show a linear relationship between the slope and the intercept.
2. **The independent variable is not random.**
3. **No Endogeneity**: Endogeneity refers to situations where the independent variable is correlated with the error term.
4. **Constant Variance:** The variance of the errors should be constant across all levels of the independent variables.
5. **Independence of Errors:** The value of the residual (error) is not correlated across all observations.
6. **Normality of Errors:** The residual (error) values follow the normal distribution. This means that the distribution of residuals should follow a bell-shaped curve with a mean of zero

Multiple linear regression

- In Multiple linear regression multiple independent variables are used in the model. The mathematical representation of multiple linear regression is:

$$Y = a + bX_1 + cX_2 + dX_3 + \epsilon$$

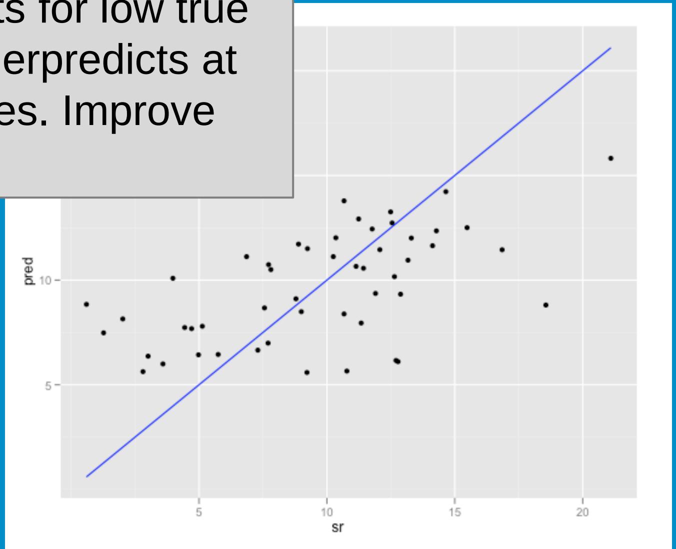
Simple and Multiple linear regression

– The difference between simple linear regression and multiple linear regression is that, multiple linear regression has (>1) independent variables, whereas simple linear regression has only 1 independent variable.

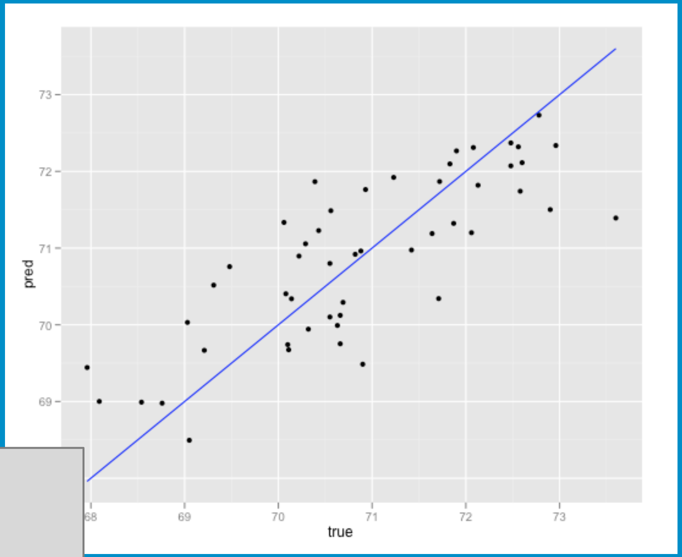
– Now, the question is “How do we obtain best fit line?”.

Diagnostics

- **Plot it!**
  - Prediction vs. true outcome
- Look for:
  - Systematic over/under prediction
  - Non-consistent variance
    - The data cloud should be symmetric about the line of true prediction
  - Obvious outliers



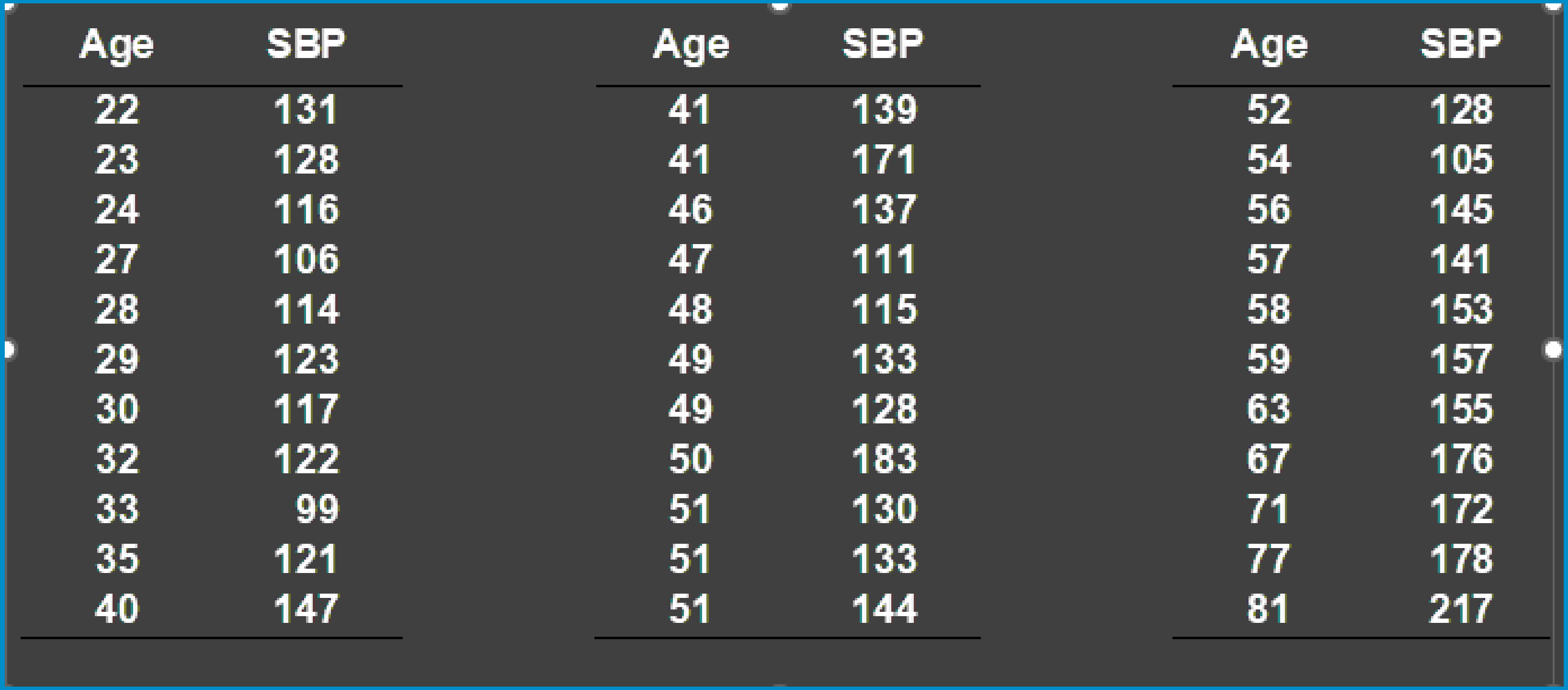
Overpredicts for low true values, underpredicts at higher values. Improve the model.



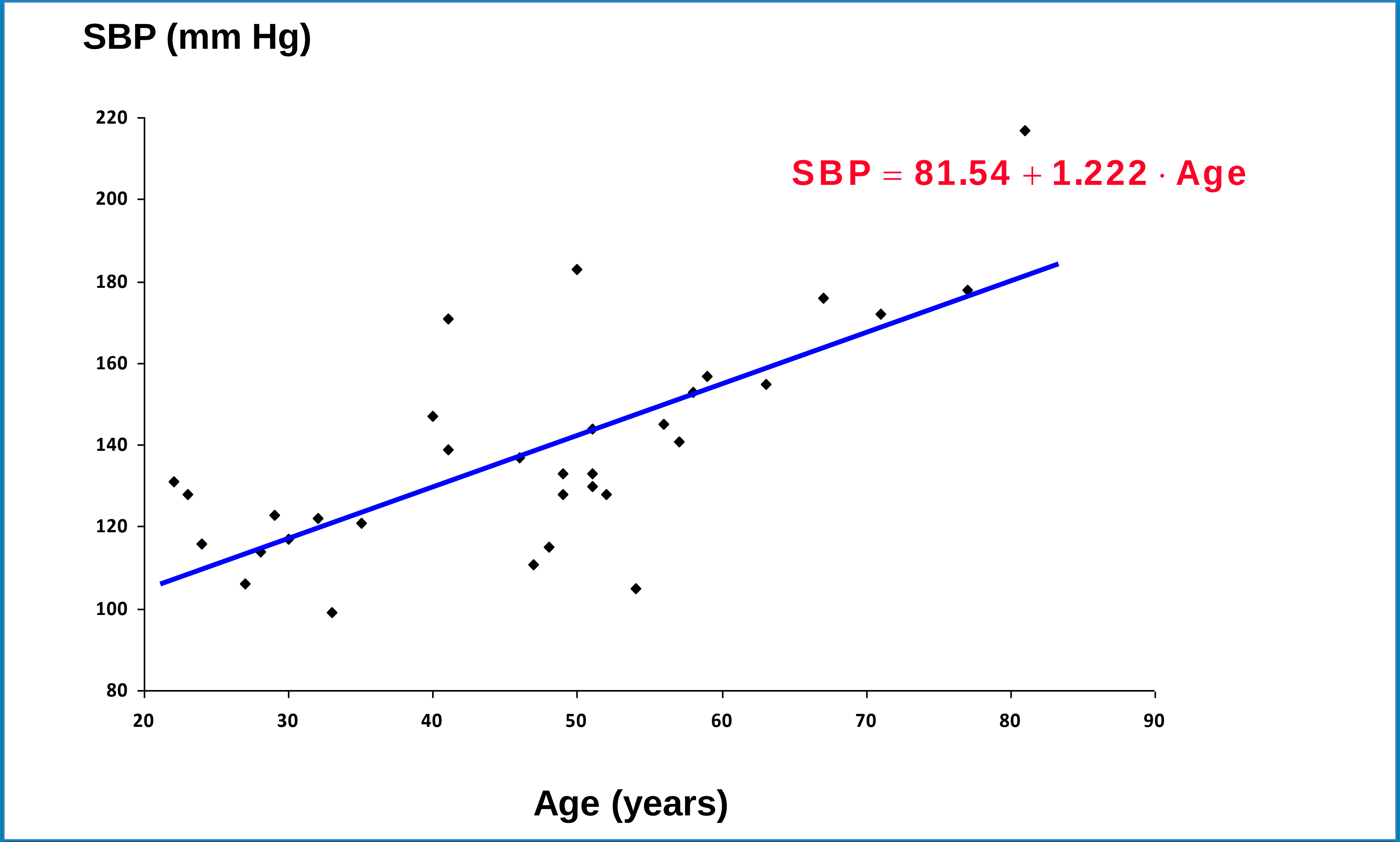
Not quite consistent variance, but much better.

Simple linear regression: Example

- Table 1: Age and systolic blood pressure (SBP) among 33 adult women



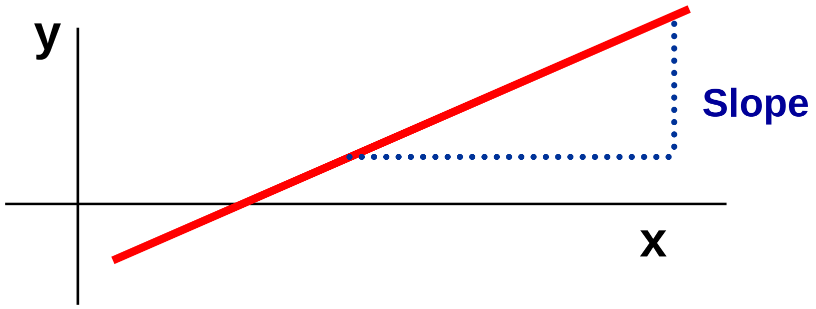
Age	SBP	Age	SBP	Age	SBP
22	131	41	139	52	128
23	128	41	171	54	105
24	116	46	137	56	145
27	106	47	111	57	141
28	114	48	115	58	153
29	123	49	133	59	157
30	117	49	128	63	155
32	122	50	183	67	176
33	99	51	130	71	172
35	121	51	133	77	178
40	147	51	144	81	217



adapted from Colton T. Statistics in Medicine. Boston: Little Brown, 1974

Simple linear regression

- Relation between 2 variables (SBP and age)



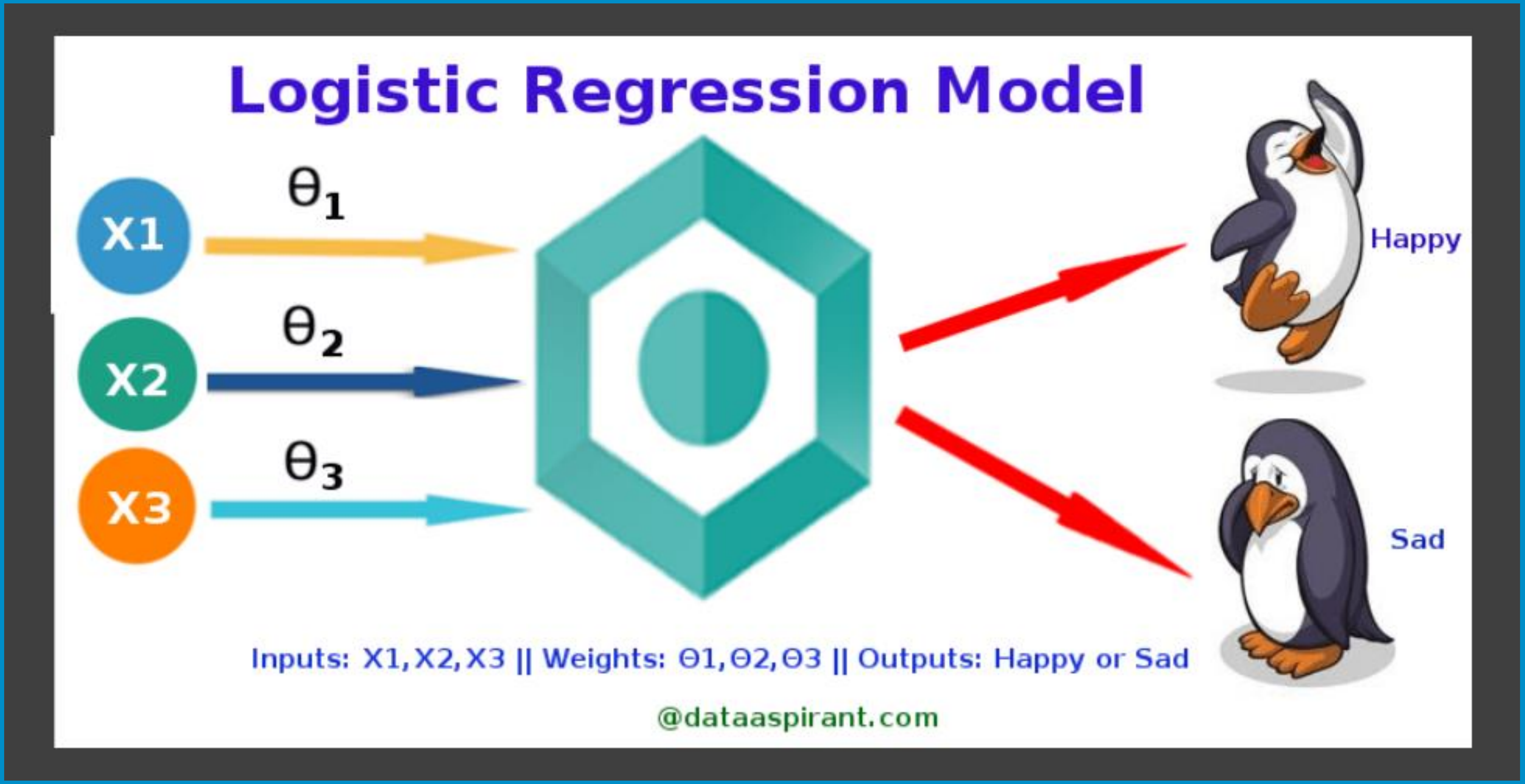
$$y = \alpha + \beta_1 x_1$$

- Regression coefficient b1
- Measures association between y and x
- Amount by which y changes on average when x changes by one unit
- Least squares method

Logistic Regression

- Used to estimate the probability that an event will occur as a function of other variables
  - The probability that a borrower will default as a function of his credit score, income, the size of the loan, and his existing debts
- Can be considered a classifier, as well
  - Assign the class label with the highest probability
- Input variables can be continuous or discrete
- Output:
  - Log-odds ratio A set of coefficients that indicate the relative impact of each driver
  - A linear expression for predicting the log-odds ratio of outcome as a function of drivers. (Binary classification case)
    - easily converted to the probability of the outcome

Logistic regression


Logistic Regression Model
X1
X2
X3
θ_1
θ_2
θ_3
Happy
Sad
Inputs: X1,X2,X3 || Weights: Θ1,Θ2,Θ3 || Outputs: Happy or Sad
@dataaspirant.com

Logistic Regression Use Cases

The preferred method for many binary classification problems:

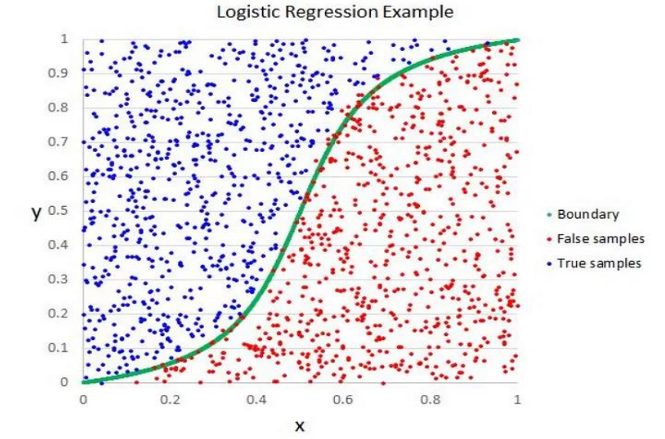
- Especially if you are interested in the probability of an event, not just predicting the "yes or no“
- Try this first; if it fails, then try something more complicated

Binary Classification examples:

- The probability that a borrower will default
- The probability that a customer will churn

Multi-class example

- The probability that a politician will vote yes/vote no/not show up to vote on a given bill



WE ARE HUMBER

Logistic Regression Model - Example

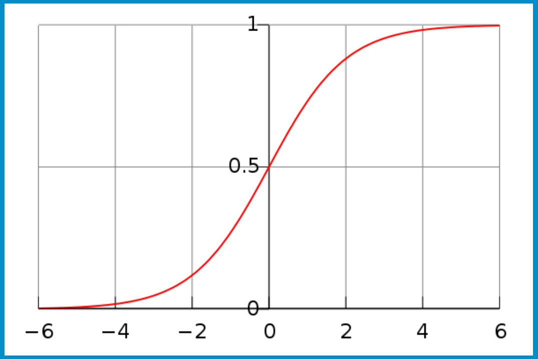
$$\text{default} = f(\text{creditScore}, \text{income}, \text{loanAmt}, \text{existingDebt})$$

- Training data: default is 0/1
  - default=1 if loan defaulted
- The model will return the probability that a loan with given characteristics will default
- If you only want a "yes/no" answer, you need a threshold
  - The standard threshold is 0.5

Technical Description (Binary Case)

$$\ln \frac{P(y=1)}{1-P(y=1)} = b_0 + b_1 x_1 + b_2 x_2 \ldots$$

- y=1 is the case of interest: 'TRUE'
- LHS is called logit(P(y=1))
  - hence, "logistic regression"
- logit(P(y=1)) is inverted by the sigmoid function
  - standard packages can return probability for you
- Categorical variables are expanded as with linear regression
- Iterative, not closed form solution
  - "Iteratively re-weighted least squares"



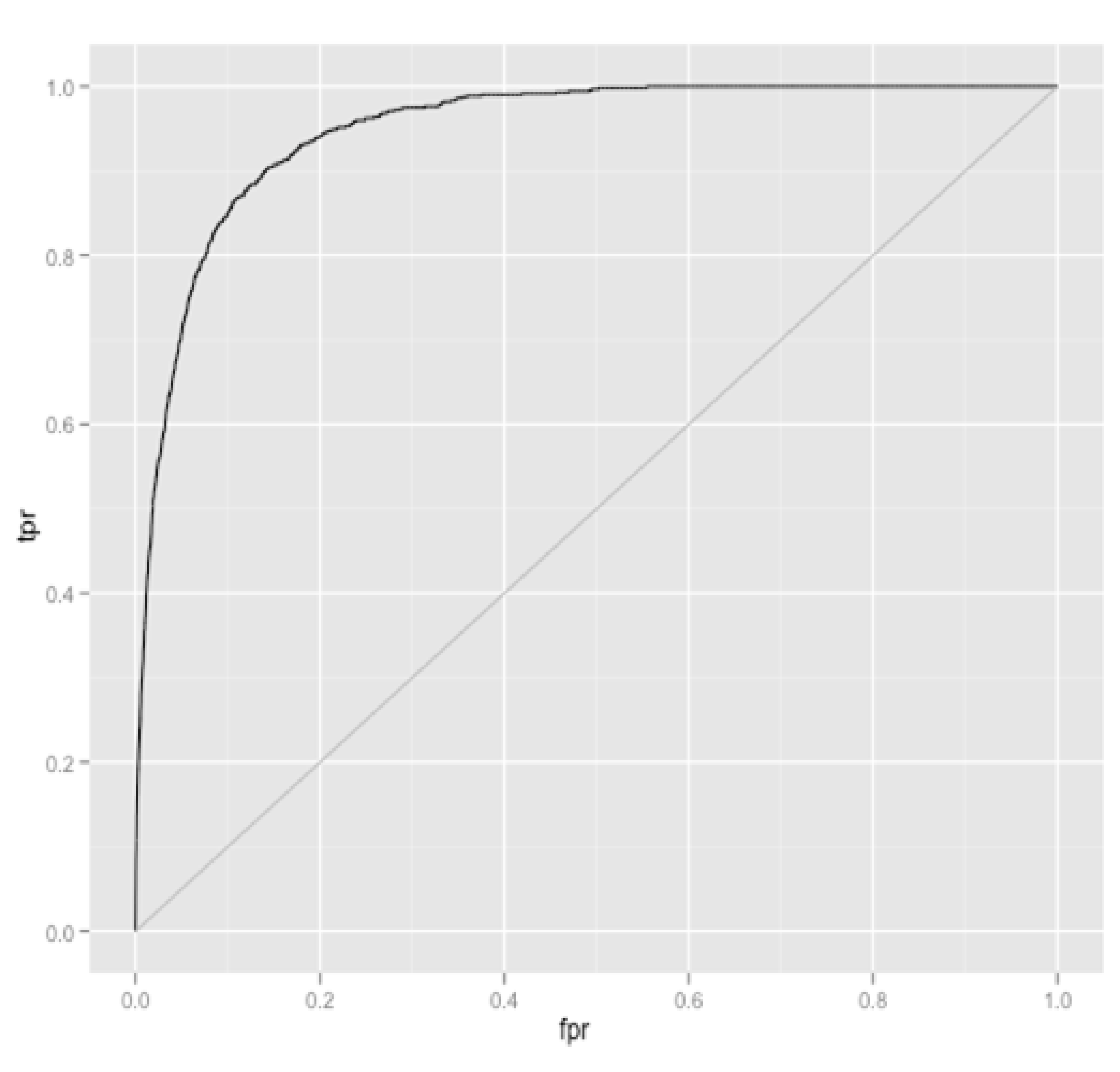
Diagnostics: ROC Curve

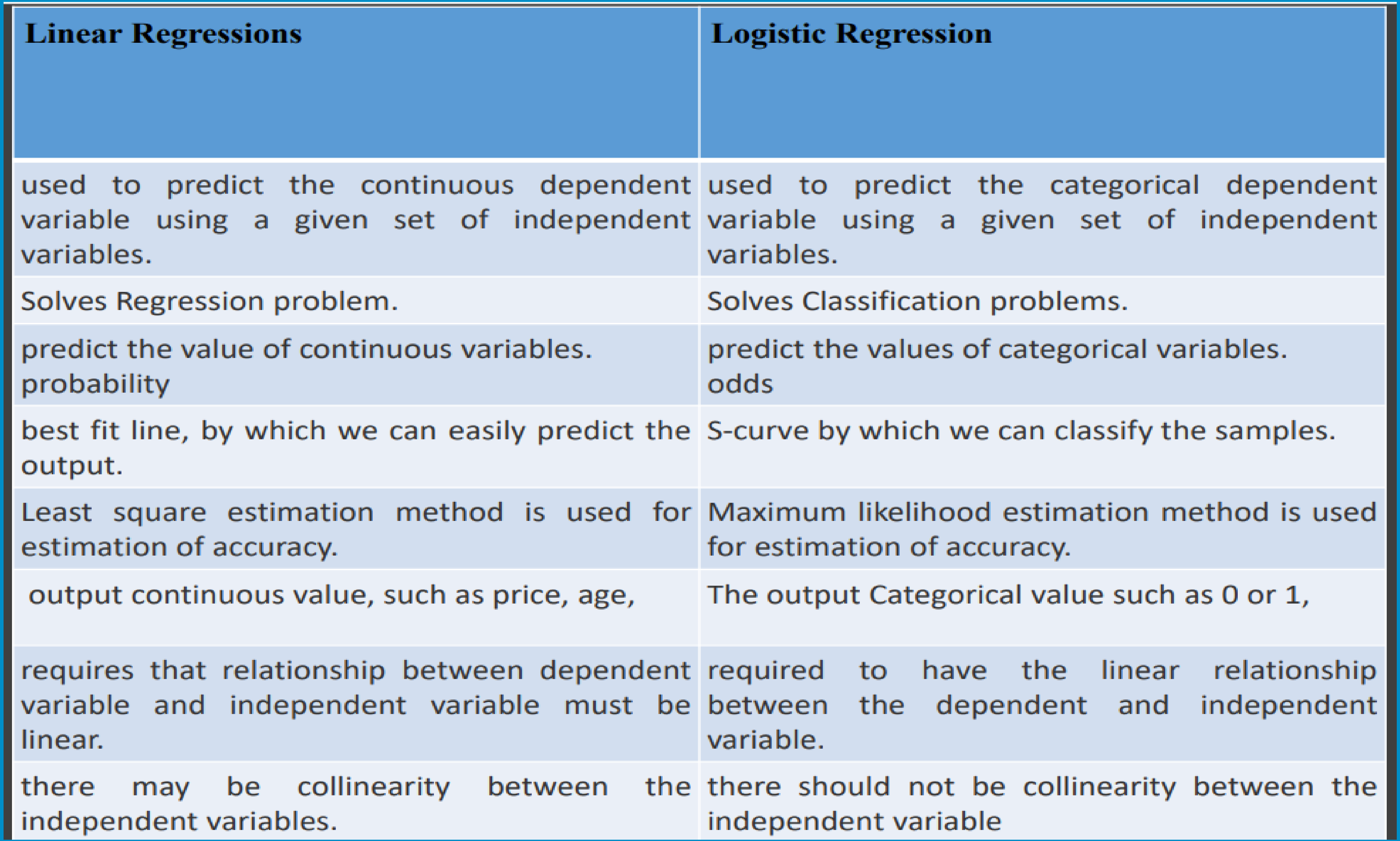
$$\text{FPR} = \frac{\#\ \text{false positives}}{\text{all negatives}}$$

$$\text{TPR} = \frac{\#\ \text{true positives}}{\text{all positives}}$$

Area under the curve (AUC) tells you how well the model predicts. (Ideal AUC = 1)

For logistic regression, ROC curve can help set classifier threshold.

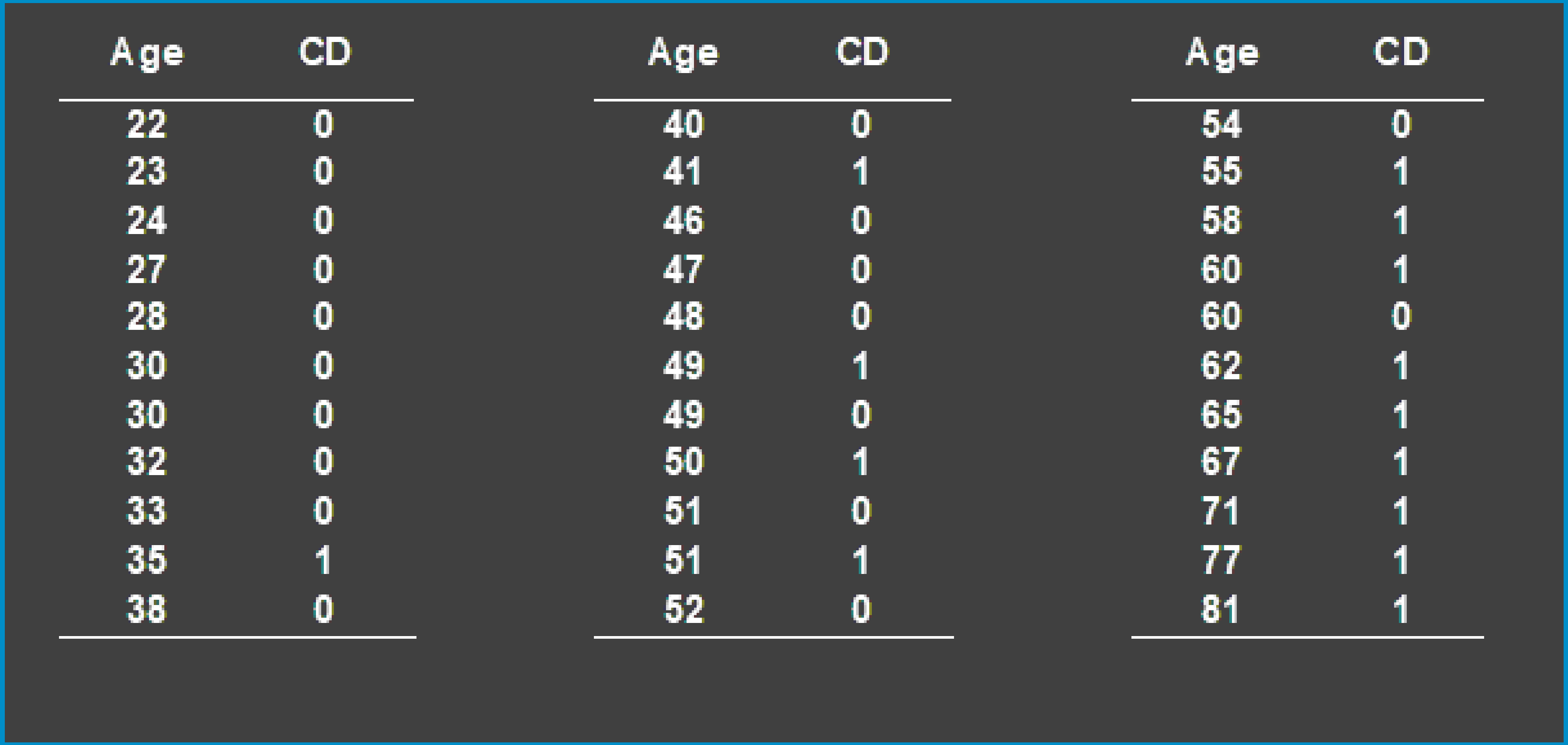




Linear Regressions	Logistic Regression
used to predict the continuous dependent variable using a given set of independent variables.	used to predict the categorical dependent variable using a given set of independent variables.
Solves Regression problem.	Solves Classification problems.
predict the value of continuous variables. probability	predict the values of categorical variables. odds
best fit line, by which we can easily predict the output.	S-curve by which we can classify the samples.
Least square estimation method is used for estimation of accuracy.	Maximum likelihood estimation method is used for estimation of accuracy.
output continuous value, such as price, age,	The output Categorical value such as 0 or 1,
requires that relationship between dependent variable and independent variable must be linear.	required to have the linear relationship between the dependent and independent variable.
there may be collinearity between the independent variables.	there should not be collinearity between the independent variable

Logistic regression

- Table 2: Age and signs of coronary heart disease (CD)

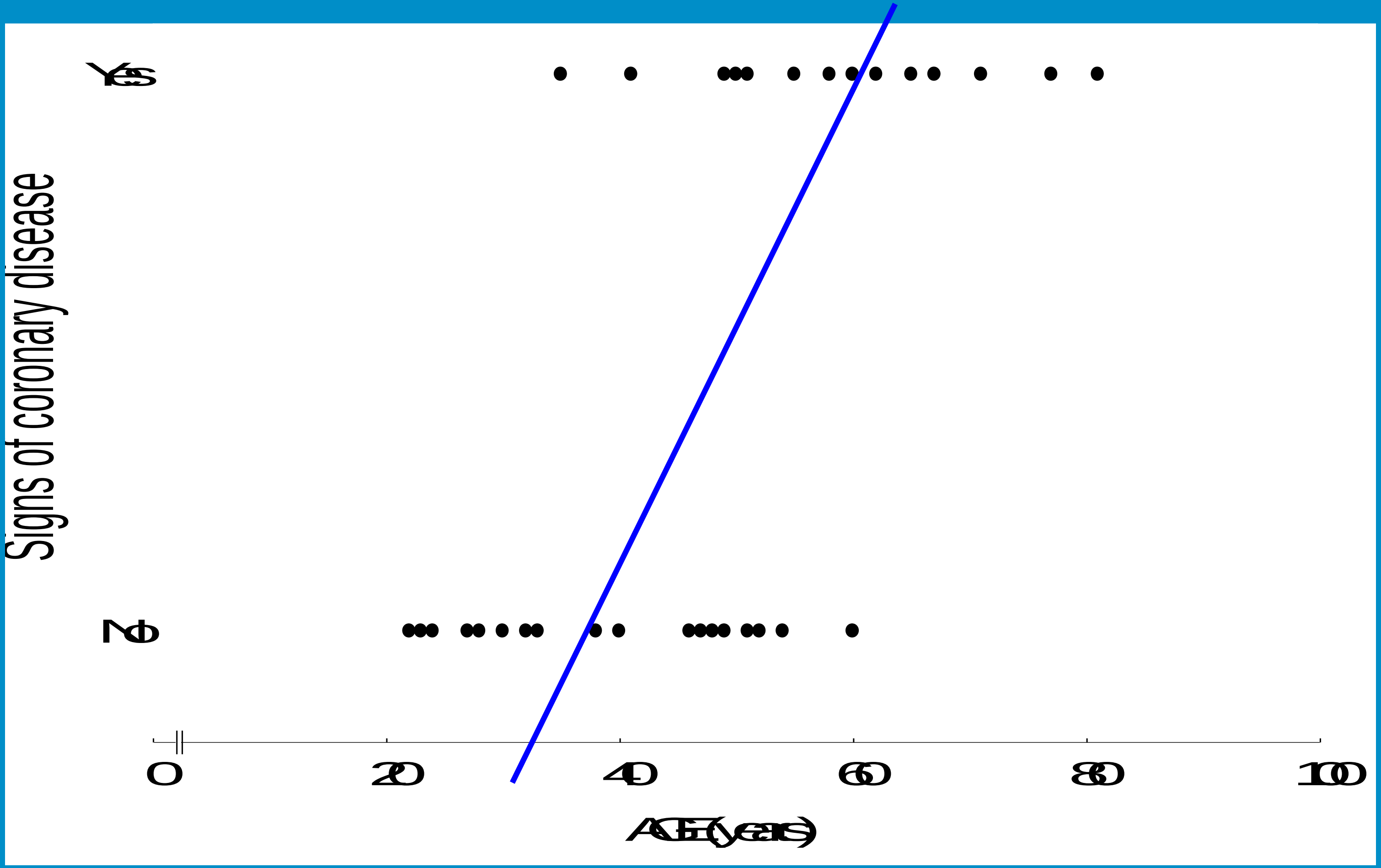


Age	CD	Age	CD	Age	CD
22	0	40	0	54	0
23	0	41	1	55	1
24	0	46	0	58	1
27	0	47	0	60	1
28	0	48	0	60	0
30	0	49	1	62	1
30	0	49	0	65	1
32	0	50	1	67	1
33	0	51	0	71	1
35	1	51	1	77	1
38	0	52	0	81	1

How can we analyze these data?

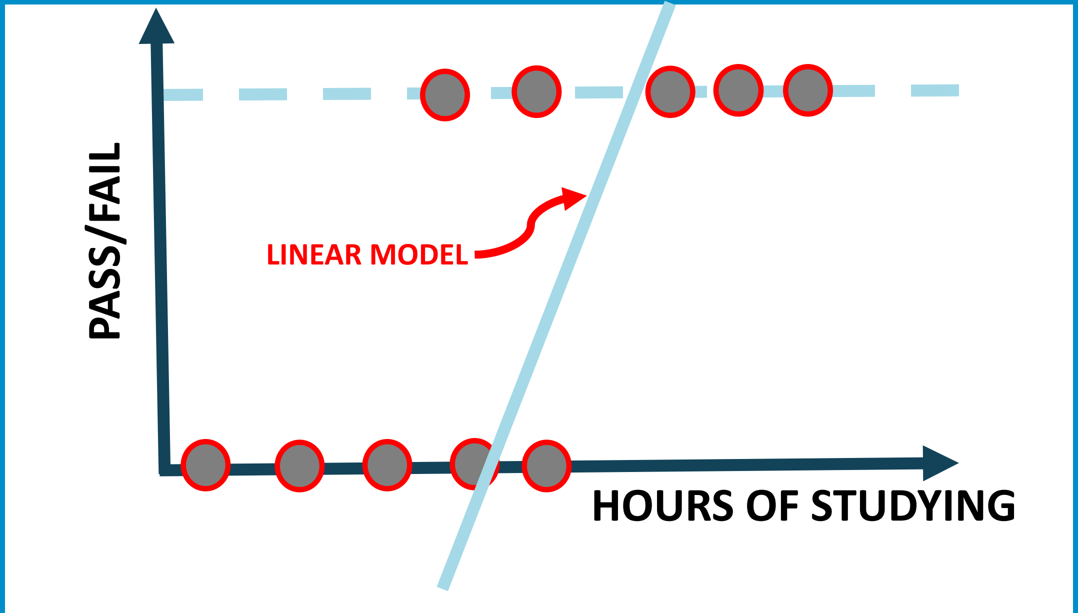
- Compare mean age of diseased and non-diseased
  - Non-diseased: 38.6 years
  - Diseased: 58.7 years (p<0.0001)
- Linear regression?

Dot-plot: Data from Table 2



Logistic regression

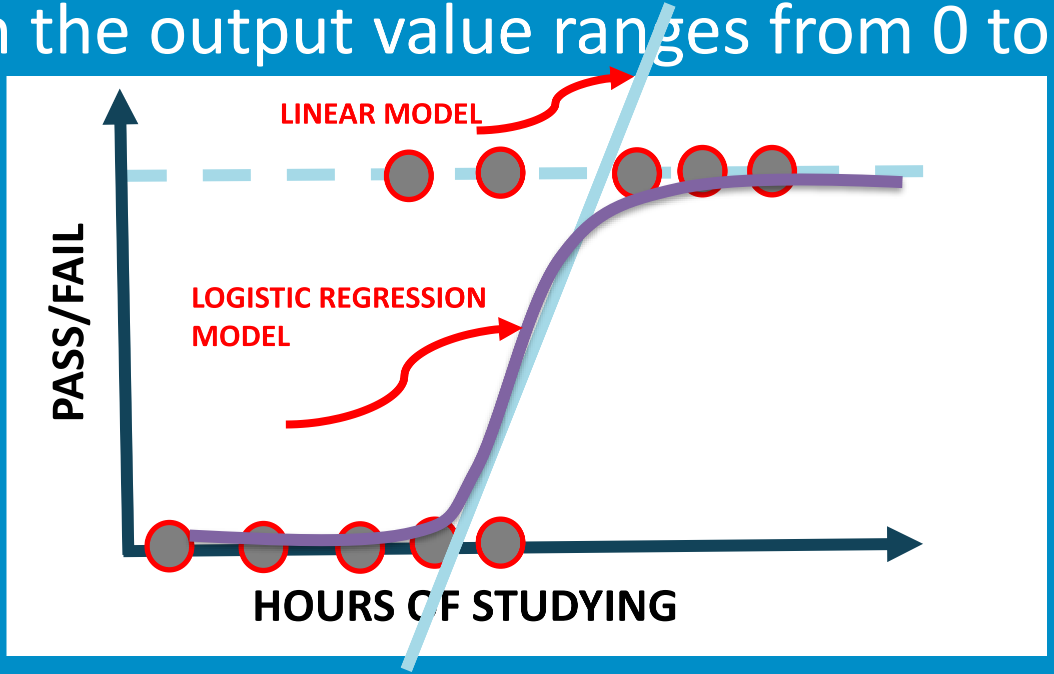
- **Linear regression** is used to predict outputs on a continuous spectrum.
    - Example: predicting revenue based on the outside air temperature.
- **Logistic regression is used to predict binary outputs** with two possible values labeled "0" or "1"
    - Logistic model output can be one of two classes: pass/fail, win/lose, healthy/sick



Hours Studying	Pass/Fail
1	0
1.5	0
2	0
3	1
3.25	0
4	1
5	1
6	1

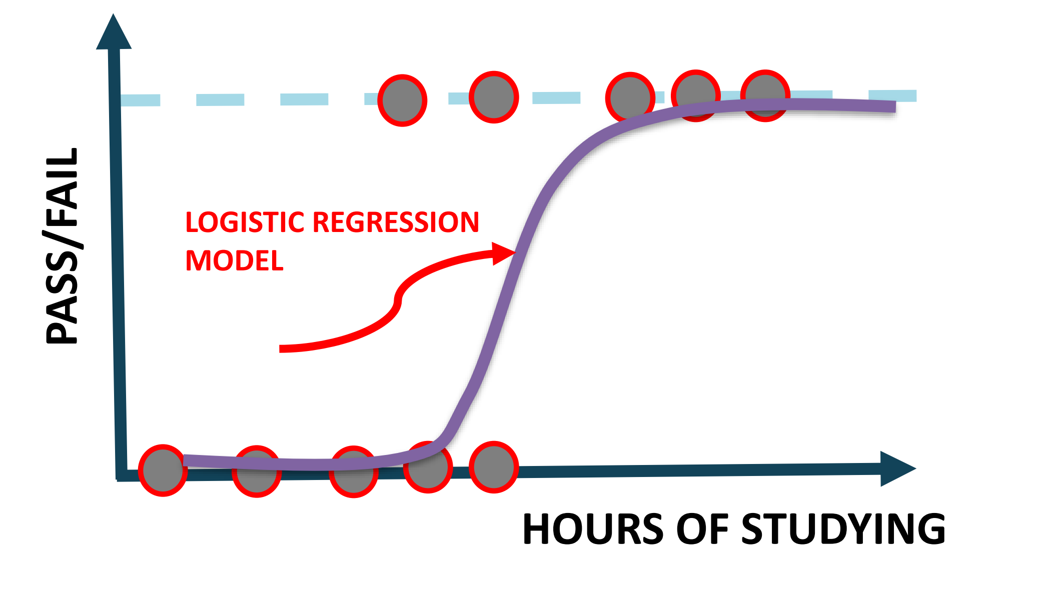
Logistic regression

- Linear regression is not suitable for classification problem.
- Linear regression is unbounded, so logistic regression will be better candidate in which the output value ranges from 0 to 1.



Logistic regression

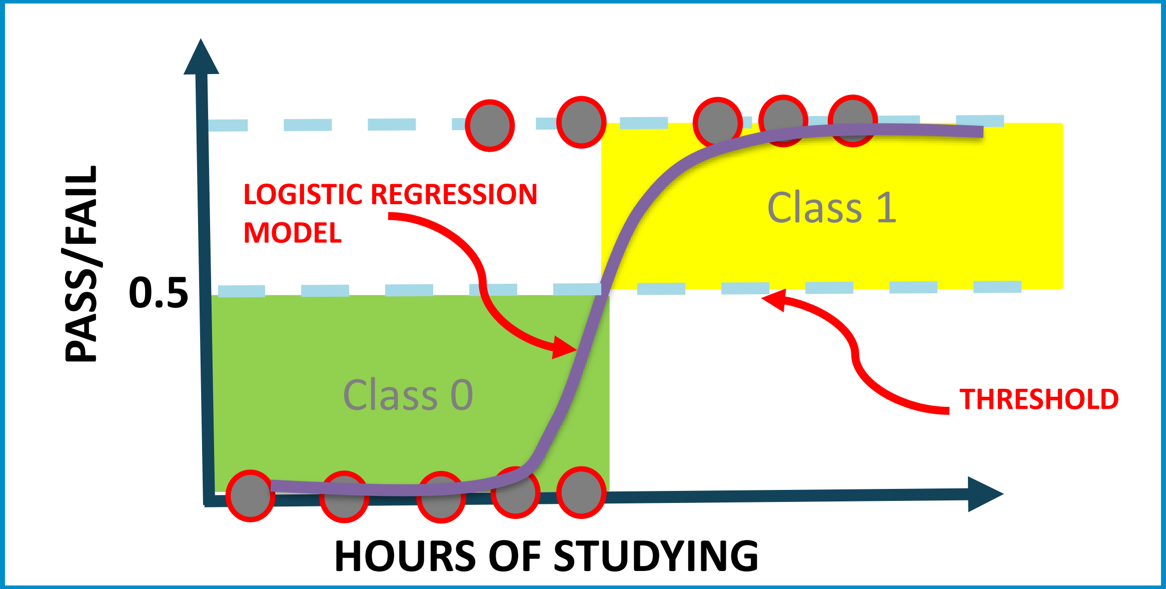
- Logistic regression algorithm works by implementing a linear equation first with independent predictors to predict a value.
- We then need to convert this value into a probability that could range from 0 to 1.



- Linear equation:
  - $y = b_0 + b_1 * x$
- Apply Sigmoid function:
  - $P(x) = \ sigmoid\ (y)$
  - $P(x) = \frac{1}{1+e^{-y}}$
  - $P(x) = \frac{1}{1+e^{-(b_0+b_1*x)}}$

Logistic regression **FROM PROBABILITY TO CLASS**

- Now we need to convert from a probability to a class value which is “0” or “1”.



- Linear equation:
  - $y = b_0 + b_1 * x$
- Apply Sigmoid function:
  - $P(x) = sigmoid\,(y)$
  - $P(x) = \frac{1}{1+e^{-y}}$

Evaluation Metrics in Machine Learning

HUMBER

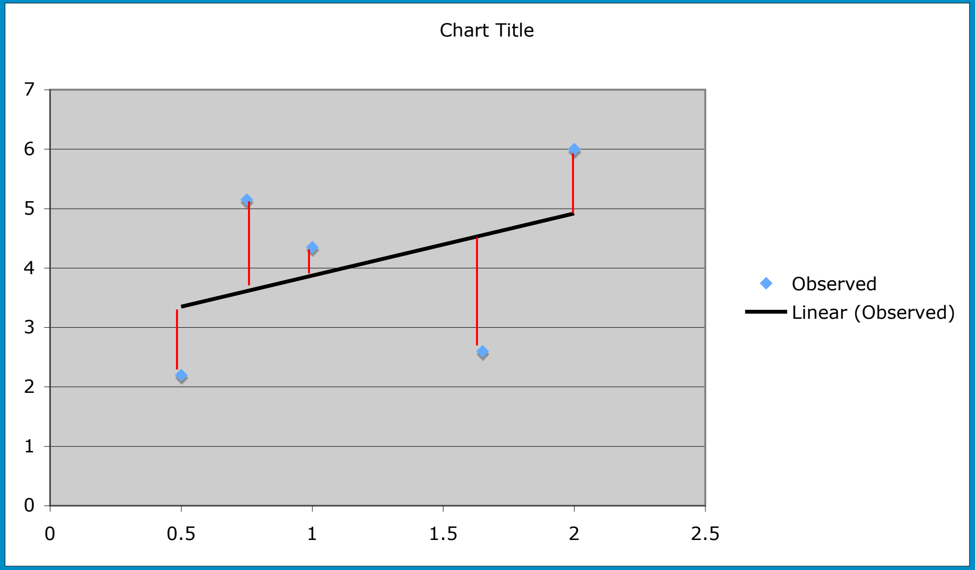
WE ARE HUMBER

Evaluation Metrics in Machine Learning

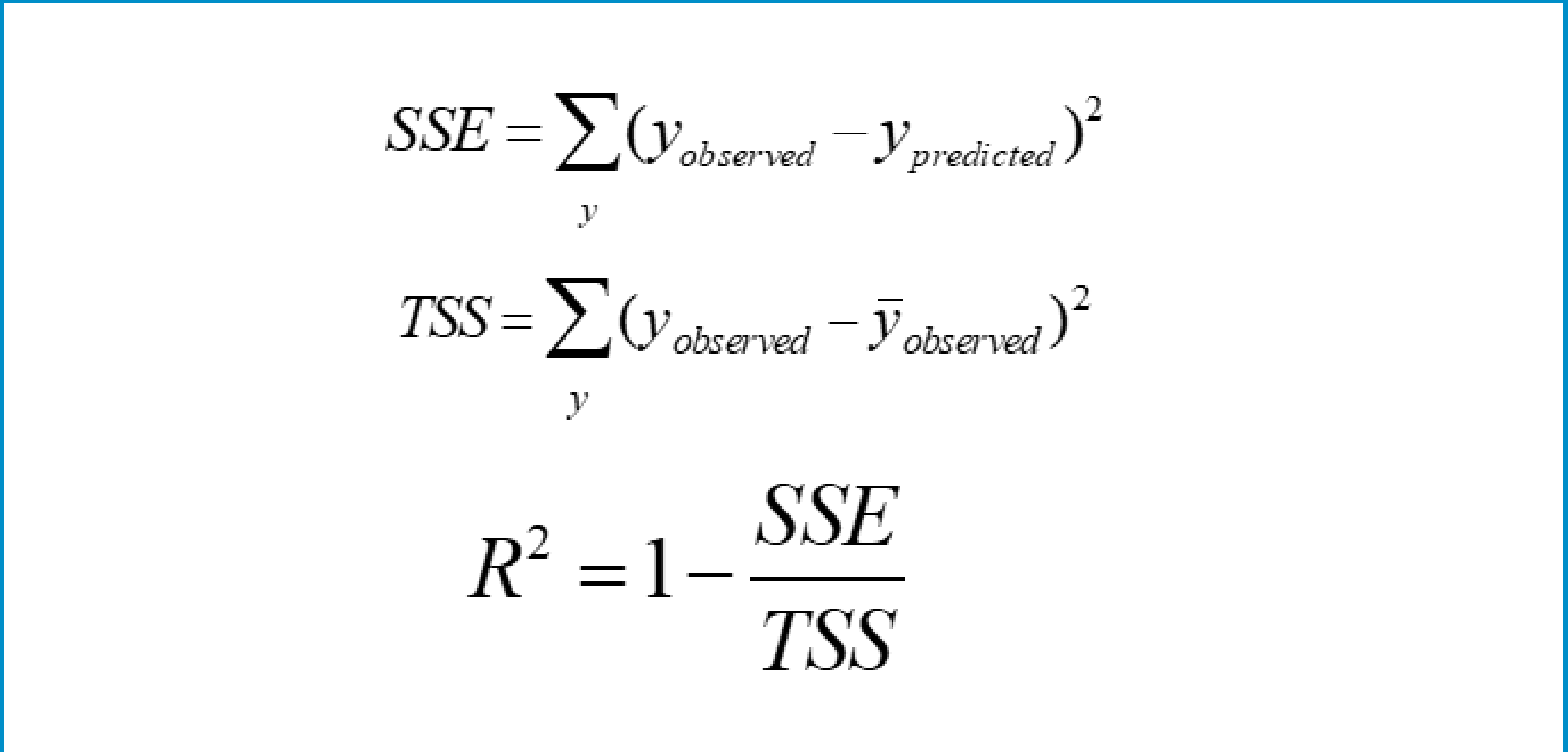
- It is extremely important to use quantitative metrics for evaluating a machine learning model
- For classification
  - Accuracy/Precision/Recall/F1-score, ROC curves,…
- For regression
  - Normalized RMSE, Normalized Mean Absolute Error (NMAE)

Residual Error

- – Observed value - Predicted value



Sum-squared Error (SSE)



$$SSE = \sum_{y} (y_{observed} - y_{predicted})^2$$

$$TSS = \sum_{y} (y_{observed} - \bar{y}_{observed})^2$$

$$R^2 = 1 - \frac{SSE}{TSS}$$

What is R-Squared

- It’s a statistical measure between 0 and 1 which calculates how similar a regression line is to the data it’s fitted to.
  - If it’s a 1, the model 100% predicts the data variance
  - if it’s a 0, the model predicts none of the variance.

$$R^2 = 1 - \frac{SSE}{TSS}$$

Evaluation Metrics in Machine Learning

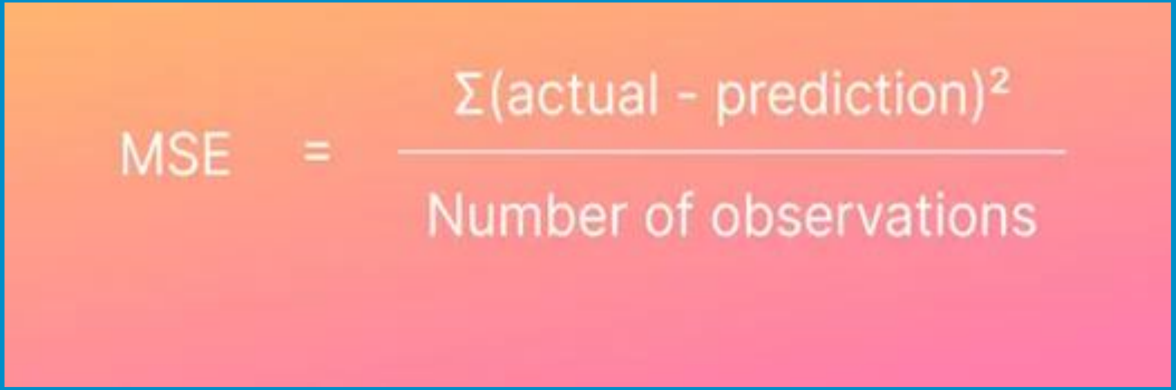
- The essential step in any machine learning model is to evaluate the accuracy of the model.
- The Mean Squared Error, Mean absolute error, Root Mean Squared Error, and R-Squared

MSE

- MSE (Mean Squared Error) is the average squared error between actual and predicted values.

- Squared error, is a row-level error calculation where the difference between the prediction and the actual is squared.

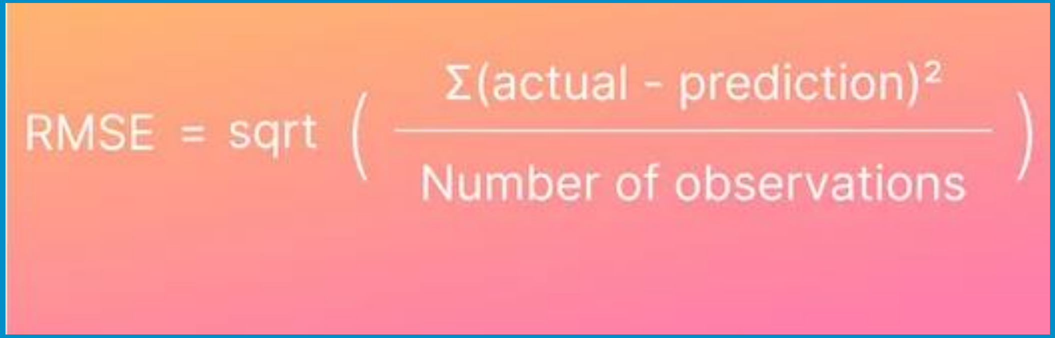
- The main draw for using MSE is that it squares the error, which results in large errors being punished or clearly highlighted.



$$MSE = \frac{\Sigma(\text{actual - prediction})^2}{\text{Number of observations}}$$

RMSE

- Root Mean Squared Error (RMSE) is the square root of the mean squared error (MSE) between the predicted and actual values.
- A benefit of using RMSE is that the metric it produces is in terms of the unit being predicted. For example, using RMSE in a house price prediction model would give the error in terms of house price, which can help end users easily understand model performance.



$$\text{RMSE} = \text{sqrt}\left(\frac{\Sigma(\text{actual} - \text{prediction})^2}{\text{Number of observations}}\right)$$

R squared compared to RMSE

- RMSE (or MSE) is the measure of goodness of predicting the validation/test values, while R^2 is a measure of goodness of fit in capturing the variance in the training set.
- R Square is not only a measure of Goodness-of-fit, it is also a measure of how much the model (the set of independent variables you selected) explain the behavior of your dependent variable.

R squared compared to RMSE - Important Note

- While R^2 measures goodness-of-fit, it also gives insight into how much the model (and the selected independent variables) explains the behavior of the dependent variable.

- A high R^2 suggests that the independent variables used in the model explain much of the variation in the target variable, meaning the model captures the underlying relationships in the data well.

- In Practice: A model might have a high R^2(good fit on training data) but a poor RMSE/MSE on the validation set, indicating overfitting (the model fits the training data well but does not generalize to new data). On the other hand, a low R^2 with a low RMSE could indicate that while the model doesn't capture all the variance in the data, it is still able to make good predictions.

R squared compared to RMSE (Cont’d)

- So, both R Square (and Adjusted R Square) and the Standard Error are extremely useful in assessing the statistical robustness of a model. And, as indicated they have completely different practical application. One measures the explanatory power of the model. The other one allows you to build Confidence Intervals. Both, very useful but different stuff.

Precision and recall

Suppose that $y = 1$ in presence of a **rare class** that we want to detect

Precision *(How much we are precise in the detection)*

Of all patients where we predicted $y = 1$,
what fraction actually has the disease?

$$\frac{\text{True Positive}}{\#\,\text{Predicted Positive}} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

Recall *(How much we are good at detecting)*

Of all patients that actually have the disease, what fraction did we correctly detect as having the disease?

$$\frac{\text{True Positive}}{\#\,\text{Actual Positive}} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

Confusion matrix

		Actual class	
		1 (p)	**0 (n)**
Predicted class	**1 (Y)**	**True positive (TP)**	**False positive (FP)**
	0 (N)	**False negative (FN)**	**True negative (TN)**

Trading off precision and recall

Logistic regression: $0 \leq h(\boldsymbol{x}) \leq 1$

- Predict 1 if $h(\boldsymbol{x}) \geq 0.5$
- Predict 0 if $h(\boldsymbol{x}) < 0.5$

These thresholds can be different from 0.5!

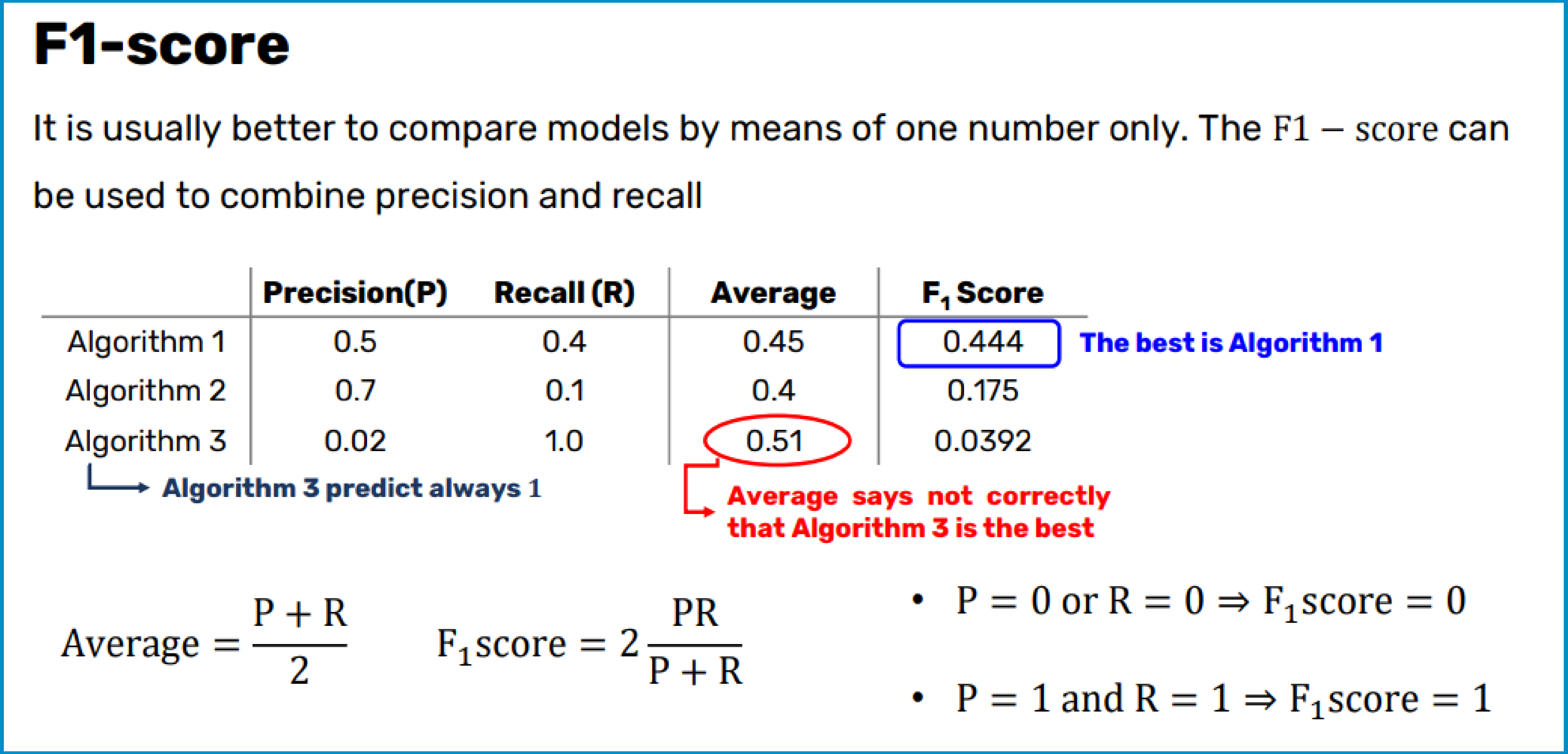
→ *At different thresholds, correspond different confusion matrices!*

Suppose we want to predict $y = 1$ (disease) only if very confident

- Increase threshold → Higher precision, lower recall

Suppose we want to avoid missing too many cases of disease (avoid false negatives).

- Decrease threshold → Higher recall, lower precision



F1-score

It is usually better to compare models by means of one number only. The $\mathrm{F1-score}$ can be used to combine precision and recall

	Precision(P)	Recall (R)	Average	F_1 Score
Algorithm 1	0.5	0.4	0.45	$\boxed{0.444}$
Algorithm 2	0.7	0.1	0.4	0.175
Algorithm 3	0.02	1.0	0.51	0.0392

The best is Algorithm 1

Algorithm 3 predict always 1

Average says not correctly that Algorithm 3 is the best

$$\mathrm{Average} = \frac{\mathrm{P} + \mathrm{R}}{2} \qquad \mathrm{F_1score} = 2\frac{\mathrm{PR}}{\mathrm{P} + \mathrm{R}}$$

- $\mathrm{P} = 0 \text{ or } \mathrm{R} = 0 \Rightarrow \mathrm{F_1score} = 0$
- $\mathrm{P} = 1 \text{ and } \mathrm{R} = 1 \Rightarrow \mathrm{F_1score} = 1$

Frameworks

- Programming languages
  - Python
  - R
  - C++
  - …
- Many libraries
  - scikit-learn
  - PyTorch
  - TensorFlow
  - Keras
  - …

classic machine learning

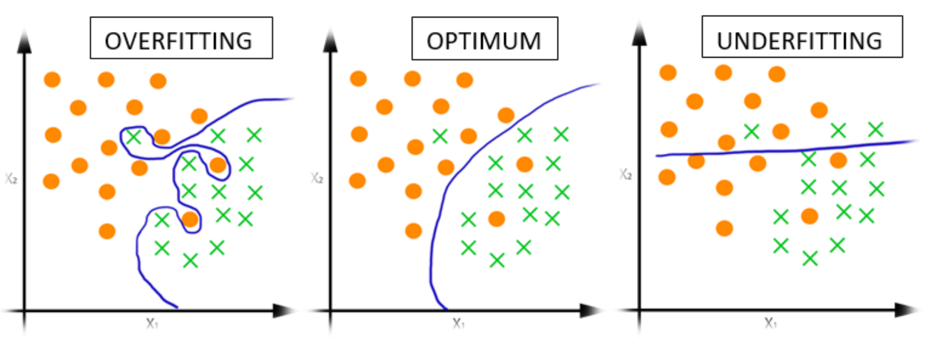
deep learning frameworks

scikit-learn

- Nice end-to-end framework
  - data exploration
  - data preprocessing (+ pandas)
    - cleaning/missing values
    - normalization
  - training
  - testing
  - application
- "Classic" machine learning only

Supervised learning: workflow

- Select model, e.g., random forest, (deep) neural network, ...
- Train model, i.e., determine parameters
  - – Data: input + output
    - training data → determine model parameters
    - validation data → to avoid overfitting
- Test model
  - – Data: input + output
    - testing data → final scoring of the model
- Production
  - – Data: input → predict output



Developing Machine Learning Systems

- Problem formulation
  - Determine problem/solution, specify a loss function
  - metrics that should be tracked
- Data collection, assessment, and management
  - When data are limited, data augmentation can help
  - For unbalanced class, undersample the majority, over-sample the minority
  - Consider outliers
  - Feature engineering, Exploratory data analysis (EDA)

Developing Machine Learning Systems (Cont’d)

- Model selection and training
  - Receiver operating characteristic (ROC) curve
  - AUC provides a single-number summary of the ROC curve
  - Confusion matrix
- Trust, interpretability, and explainability
  - Source control Testing Review, Monitoring, Accountability,
  - Inspect the actual model and understand why it got a particular answer for input
- Operation, monitoring, and maintenance
  - Monitor your performance on live data
  - Nonstationarity—the world changes over time

Summary

- If the available feedback provides the correct answer for example inputs, then the learning problem is called supervised learning.
- Learning a function whose output is a continuous or ordered value (like weight) is called regression.
- Learning a function with a small number of possible output categories is called classification.
- Decision trees can represent all Boolean functions.
- The information-gain heuristic provides an efficient method for finding a simple, consistent decision tree.
- A linear classifier with a hard threshold—also known as a perceptron—can be trained by a simple weight update rule to fit data that are linearly separable.
- Logistic regression replaces the perceptron's hard threshold with a soft threshold defined by a logistic function