Machine Learning Course

Week 3: Logistic Regression

LSI, 2022

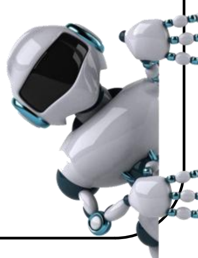
Pr. Khadija SLIMANI

Previously on ML..

- What are the 3 main Machine Learning categories ?
- What is the formula for a typical simple linear regression ?
- Which expression of the Cost function do we prefer and why ?

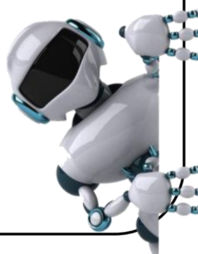
$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \qquad J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

- How do we evaluate the performance of a simple linear regression ?
- What is the difference between simple and multiple regression ?
- How can you improve a linear regression model ?



Course overview …..

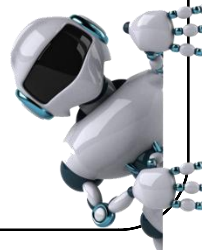
1. Week 1 : Introduction to Data Science and Machine Learning
2. Week 2: Univariate & Multivariate Linear Regression
3. **Week 3: Logistic Regression (Classification)**
4. Week 4: Decision Trees (Regression & Classification)
5. Week 5: Model evaluation (overfitting, bias-variance, crossfolding, ...)
6. Week 6: ….



Course overview

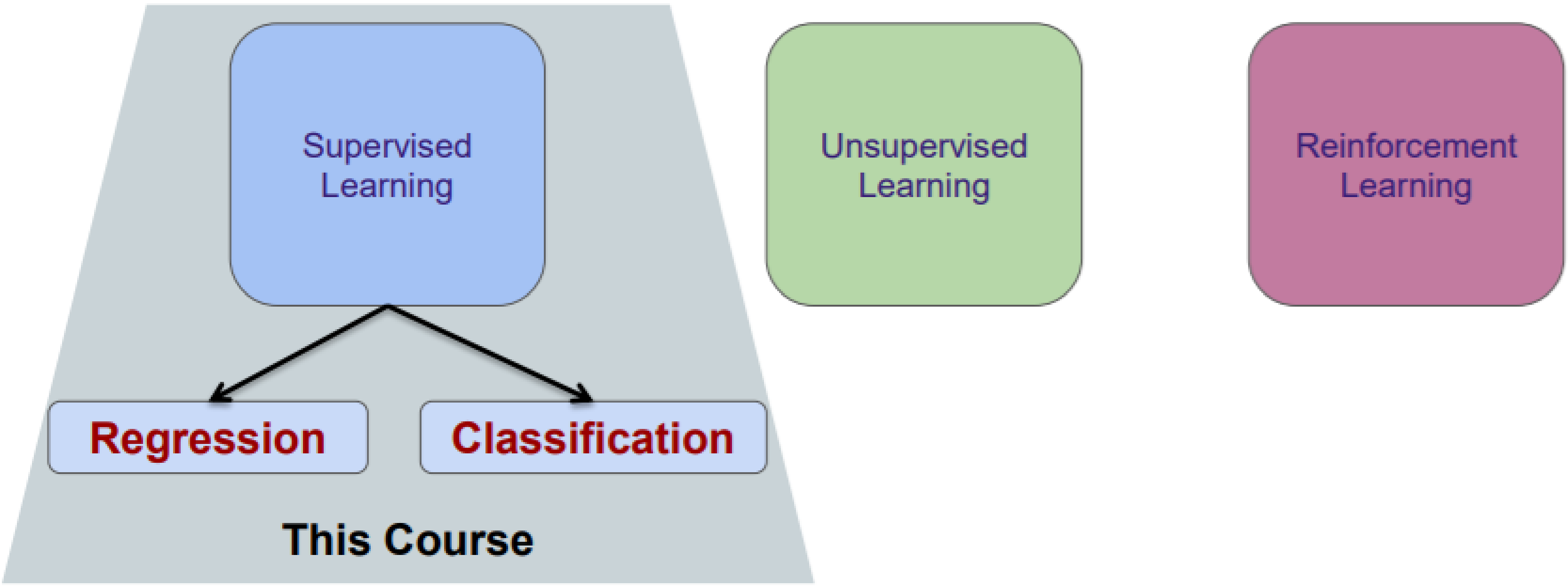
- Week 3 : Logistic Regression (Classification)

1. Introduction to Classification
2. Linear Regression for Classification ?
3. Logistic Regression Model (Binary Classification)
4. Multi-Class Logistic Regression
5. Evaluating Classifiers
6. Cross Validation

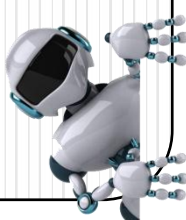


Reminder of Machine Learning Types

- Machine learning tasks are typically classified into **three broad categories.**



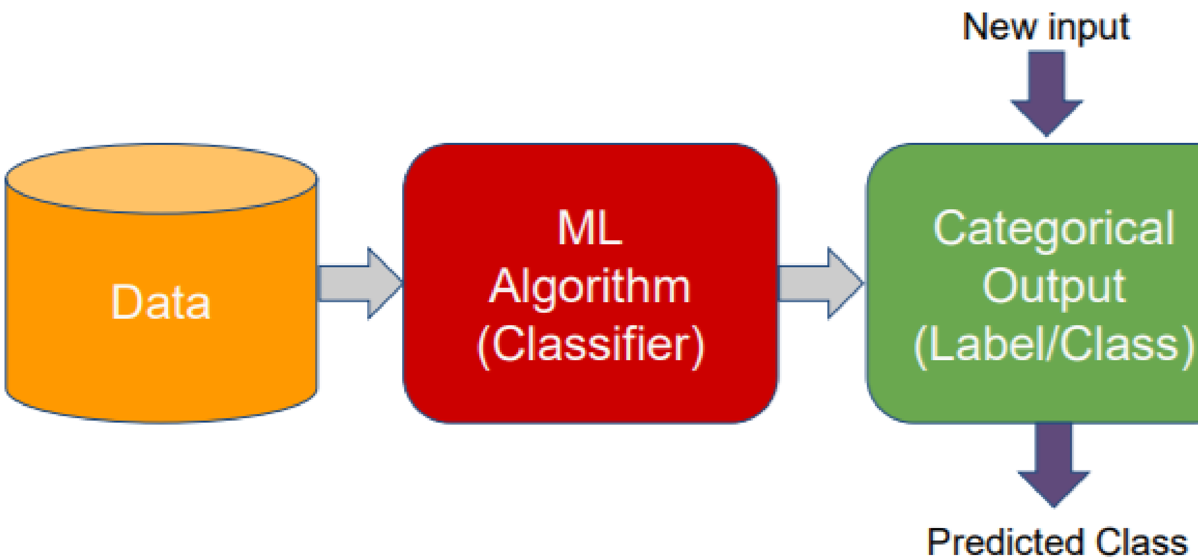
3.1
Introduction to Classification

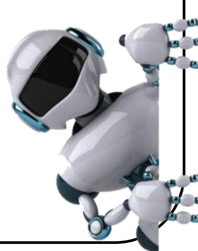


Classification

- **Goal**: Inputs are divided into two or more classes, and the ML algorithm must produce a model that assigns unseen inputs to one or more of these classes .

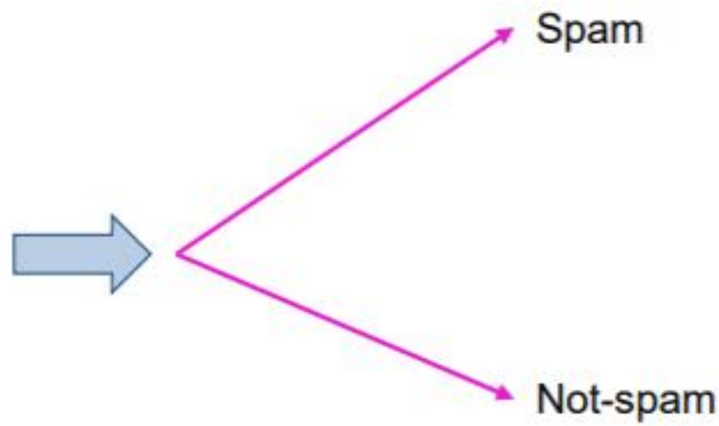
- An algorithm that implements classification is known as a **classifier**





Two-class(Binary) Classification

- **Emails type:** Output y has 2 categories

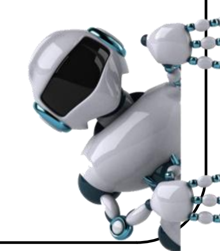


Input: x

Mail sender, subjet, keywords, etc…

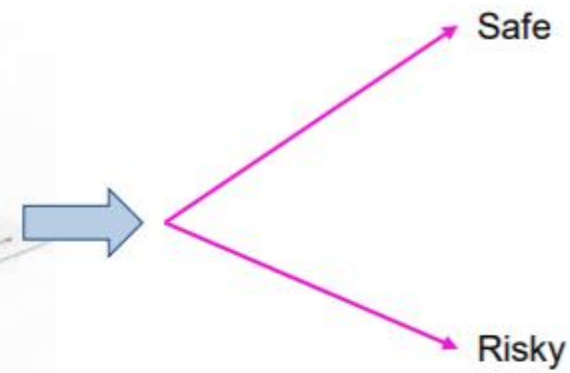
Output: y

Email type



Two-class(Binary) Classification

- **Loan demand:** Output y has 2 categories

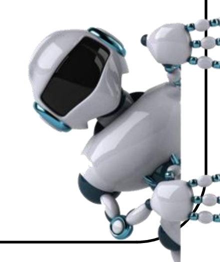


Input: x

Client's characteristics
(age, Revenue, credit, etc..)

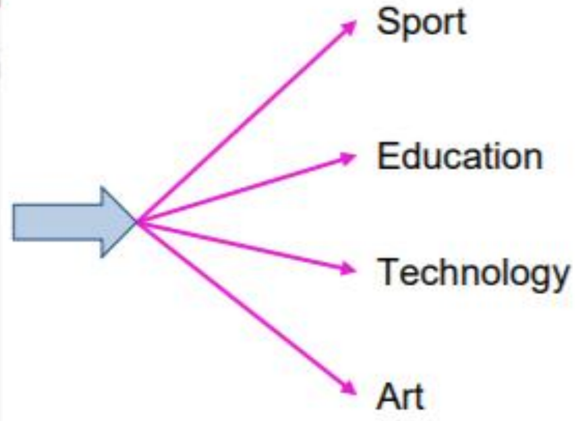
Output: y

Loan safety evaluation



Multi-class Classifier

- **News:** Output y has more than 2 categories

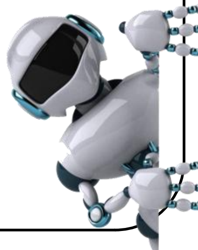


Input: x

Webpage: title, keywords, etc..

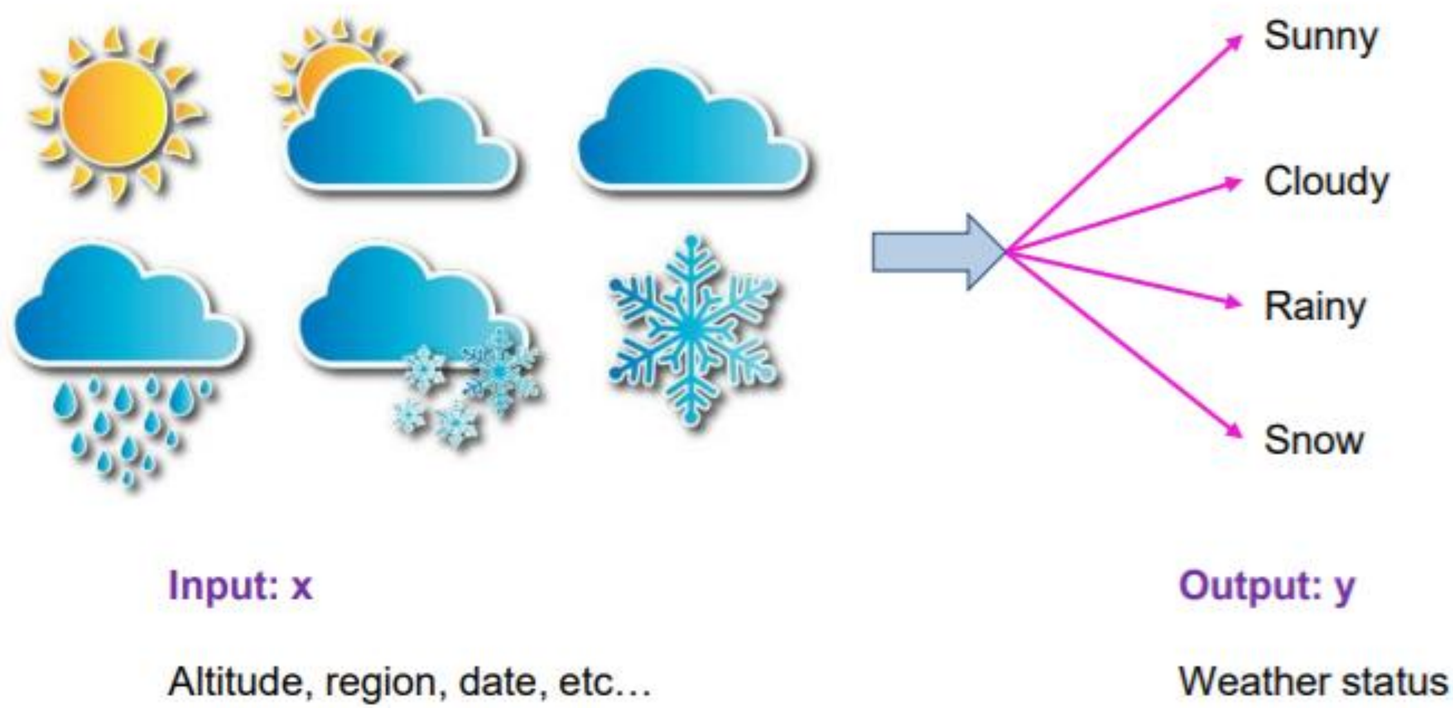
Output: y

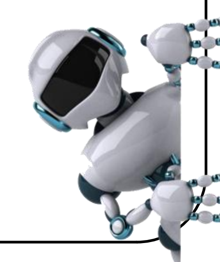
News category



Multi-class Classifier

- **Weather:** Output y has more than 2 categories





Classification Algorithms

Linear Classifiers

- Logistic Regression
- Naive Bayes classifier
- Linear discriminant

Support vector machines

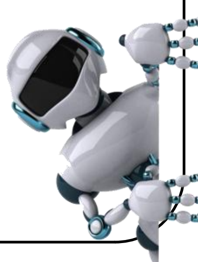
Decision Trees

- Random Forests
- Boosting

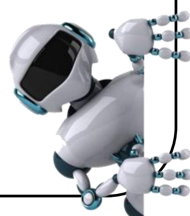
Quadratic Classifiers

Neural Netwoks

K-nearest neighbor

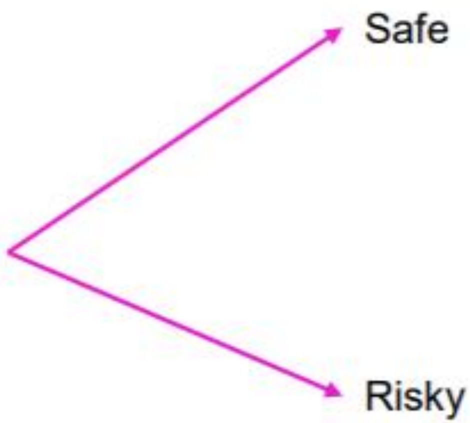


Classification problem example



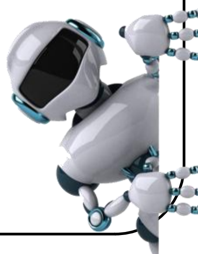
Can Linear Regression help ?

- Classification problem: Loan demand safety ?



Input: x
Client's characteristics
(age, Revenue, charges, etc..)

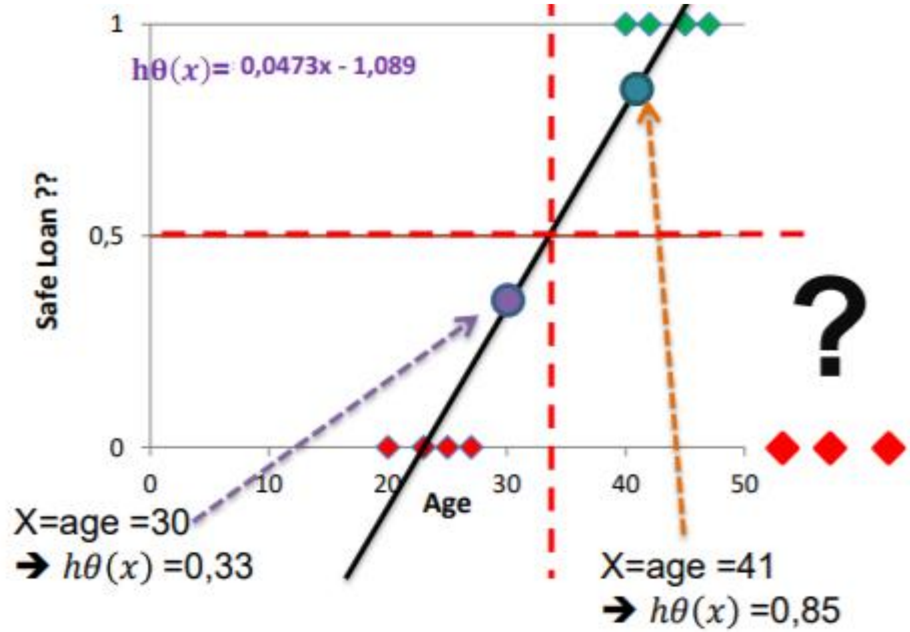
Output: y
Loan safety evaluation



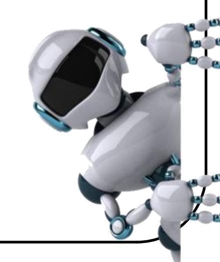
Can Linear Regression help ?

- One method is to use linear regression.

x Age	y Loan safety
20	0
23	0
25	0
27	0
40	1
42	1
45	1
47	1

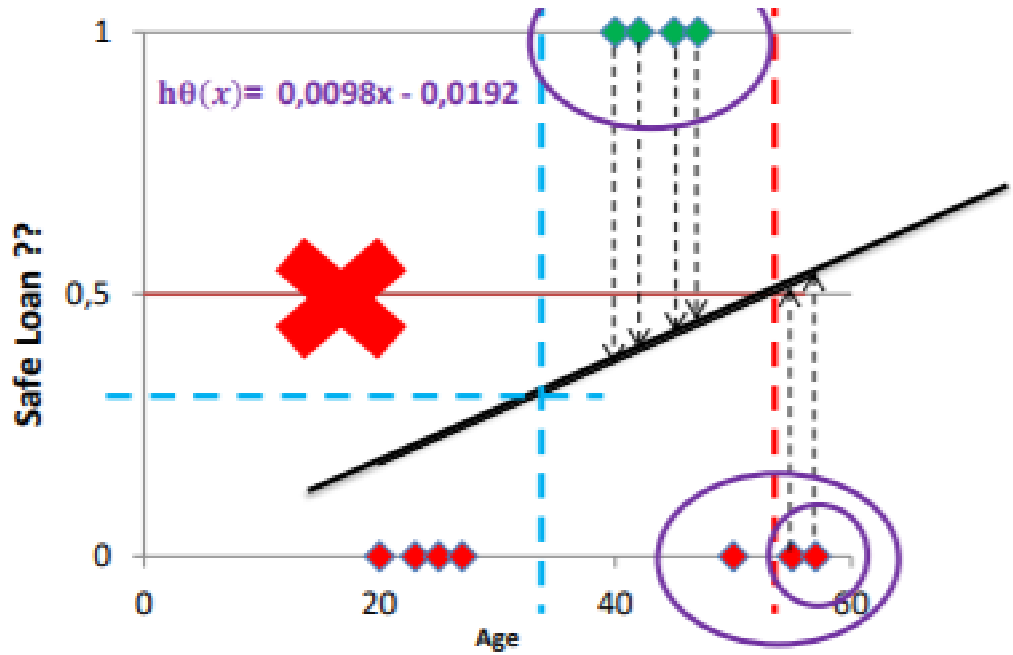


- Map all predictions greater than 0.5 as a 1 and all less than 0.5 as a 0.
- Threshold classifier output $h\theta(x)$ at 0.5 :
  - if $h\theta(x) \geq 0{,}5$ ➔ predict: y="1"
  - if $h\theta(x) < 0{,}5$ ➔ predict: y="0"



Can Linear Regression help ?

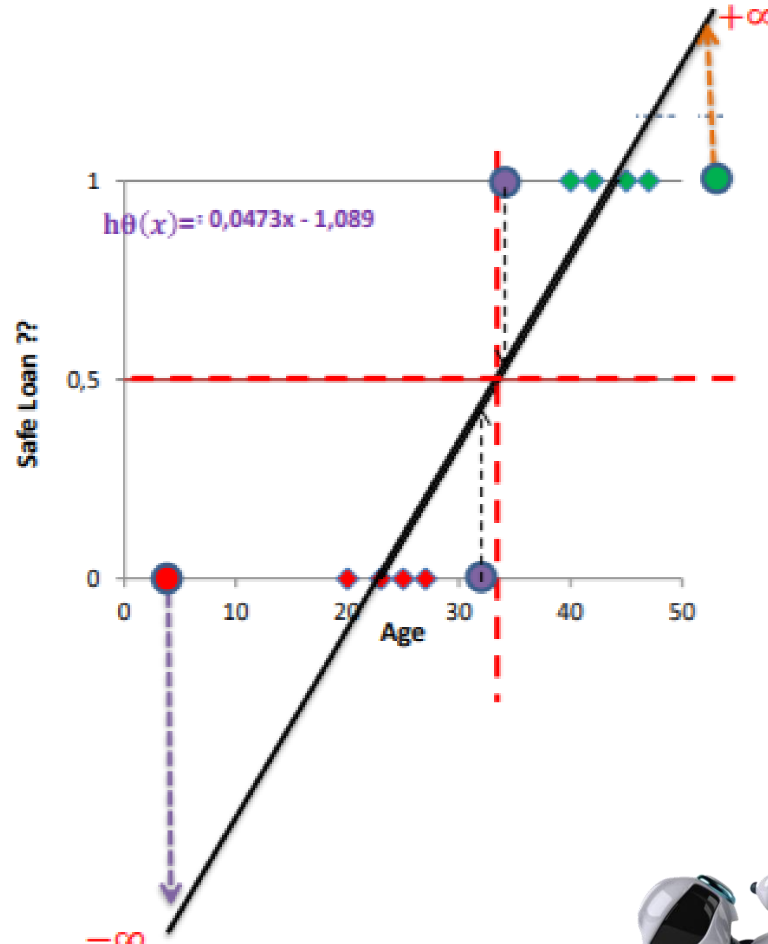
- Linear Regression can sometimes be lucky…but it is often not useful for classification problems. 1st



- Another problem: is that classification need categorical values: 2nd
  - y = 0 or 1
  - But with LR: $h\theta(x)$ can be > 1 or < 0

Can Linear Regression help ?

- How the model behaves with extreme points?
  - Certain predictions.
- And with middle points ?
  - Not very certain
- **We need to know how confident our prediction is.** 3rd

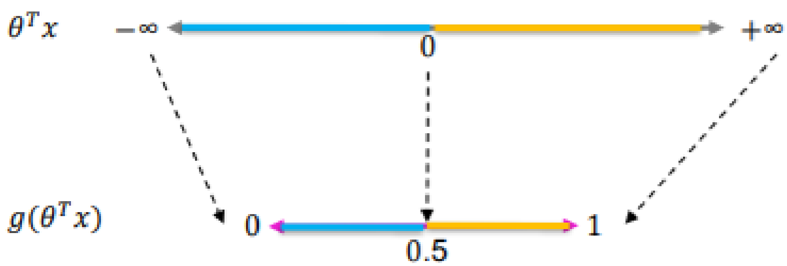


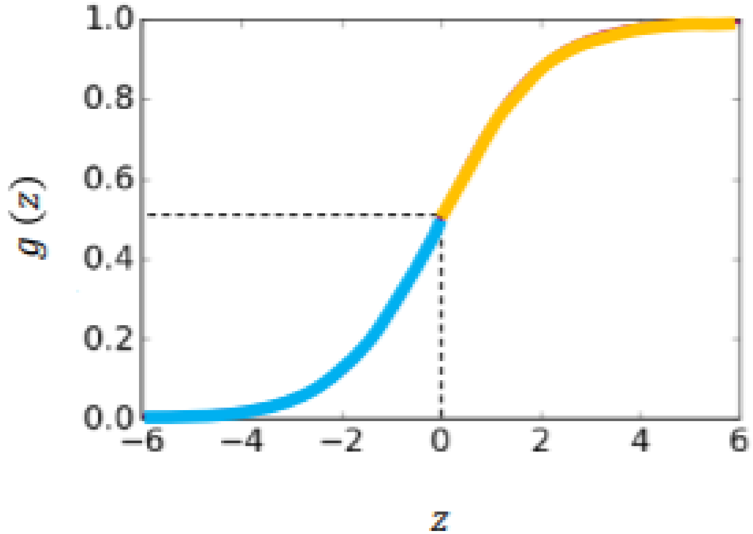
- Need for another Linear Classifier !!
- **➔ Logistic Regression**

3.3
Logistic Regression

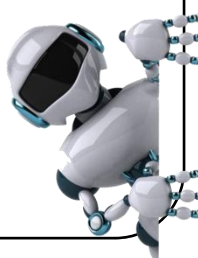
Logistic Regression Model

- Change the form for our hypotheses $h_\theta(x) = \theta^T x$ to satisfy $0 \leq h_\theta(x) \leq 1$



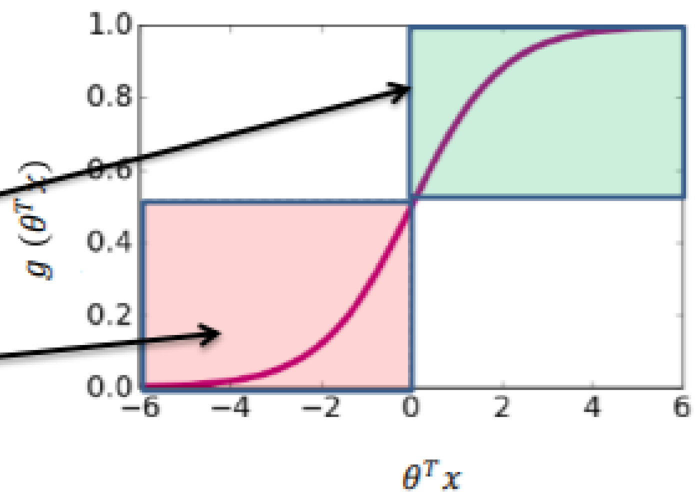


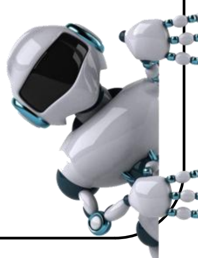
- Use the Sigmoid / Logistic Function : $g(z) = \frac{1}{1+e^{-z}}$
- $h_\theta(x) = g\left(\theta^T x\right) = \frac{1}{1+e^{-\theta^T x}}$



Interpretation of $h\theta(\mathrm{x}) = g(\theta T\mathrm{x})$

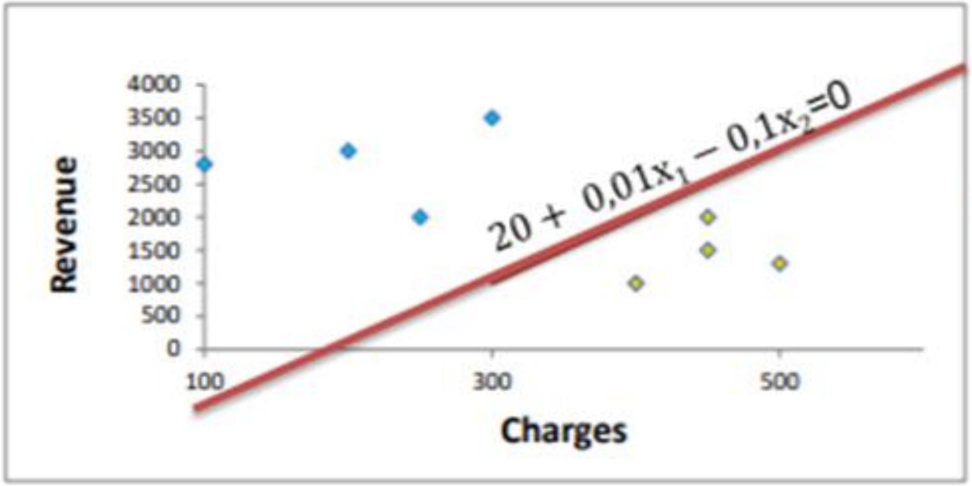
- $h_\theta(x) = g\,(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$
- $h_\theta(x)$ = **estimated probability that y = 1** given the input x parameterized by θ
  - Example: $h_\theta(x)$= 0,8 ➔ The probability that the loan is safe (y=1) is equal to 80%
- $h_\theta(x) = \mathrm{P}(\mathrm{y} = 1|\,\mathrm{x};\theta) = 1 - P(\mathrm{y} = 0|\,\mathrm{x};\theta)$
- $\mathrm{P}(\mathrm{y} = 1|\,\mathrm{x};\theta) + P(\mathrm{y} = 0|\,\mathrm{x};\theta) = 1$
- Hypothesis:
  - ❑ $y = 1\ when\ g(\theta^T x) \geq 0{,}5$ ➔ $\theta^T x \geq 0$
  - ❑ $y = 0\ when\ g(\theta^T x) < 0{,}5$ ➔ $\theta^T x < 0$

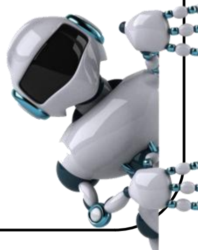




Decision Boundary

- The decision regions are separated by surfaces called the **decision boundaries.**
- These separating surfaces represent points where there are links between two or more categories.

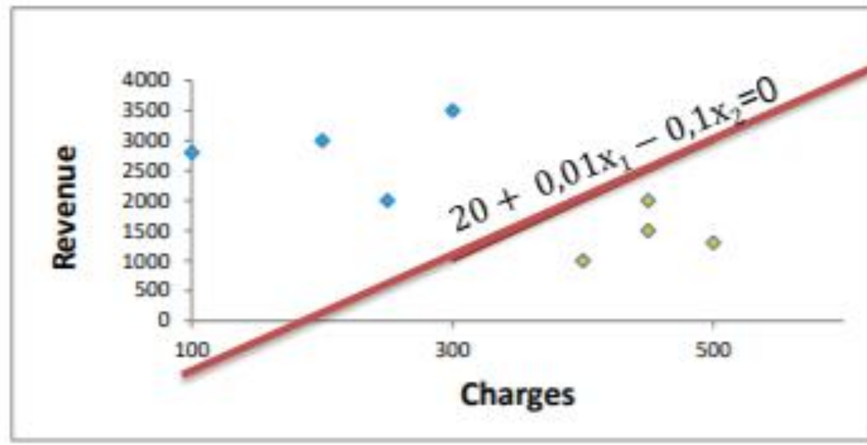




Decision Boundary

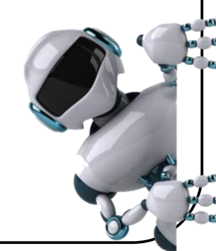
- Example: Loan demand evaluation model
  - Predict the loan safety class given the revenue and the charges values
  - $h_\theta(x) = g\,(\theta_0 + \theta_1 \mathrm{x}_1 + \theta_2 \mathrm{x}_2) = \mathrm{g}\,(20 + 0{,}01\ \#\mathrm{revenue} - 0{,}1\#\mathrm{charges})$

- Predict **1** : if $g(\theta^T x) \geq 0{,}5 \;\rightarrow\; \theta^T x \geq 0$
  - if $g\,(20 + 0{,}01\mathrm{x}_1 - 0{,}1\mathrm{x}_2) \geq 0.5$
  - $\Rightarrow 20 +\ 0{,}01\mathrm{x}_1 - 0{,}1\mathrm{x}_2 \geq 0$



- Predict **0** : if $g(\theta^T x) < 0{,}5 \;\rightarrow\; \theta^T x < 0$
  - if $g\,(20 + 0{,}01\mathrm{x}_1 - 0{,}1\mathrm{x}_2) < 0.5$
  - $\Rightarrow 20 +\ 0{,}01\mathrm{x}_1 - 0{,}1\mathrm{x}_2 < 0$

- $20 +\ 0{,}01\mathrm{x}_1 - 0{,}1\mathrm{x}_2$=0 is our decision boundary

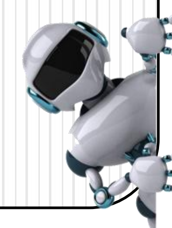


Decision Boundary

- Example: Loan demand evaluation model.
- Predict the loan safety class given the revenue and the charges values
  - $h_\theta(x) = g\ (\theta_0 + \theta_1 x_1 + \theta_2 x_2) = \mathrm{g}\ (20 + 0{,}01\ \#\mathrm{revenue} - 0{,}1\#\mathrm{charges})$

Charges	Revenue	$\theta^T x$	$g(\theta^T x)$	Safe loan ? (prediction)
500	1300	-17	4,14E-08	0
450	1500	-10	4,54E-05	0
400	1000	-10	4,54E-05	0
450	2000	-5	6,69E-03	0
250	2000	15	1,00E+00	1
200	3000	30	1,00E+00	1
300	3500	25	1,00E+00	1
100	2800	38	1,00E+00	1
450	2700	2	0,880797	1

3.4
Multiclass classification



Multi-class Classification: Example

- Image labelling

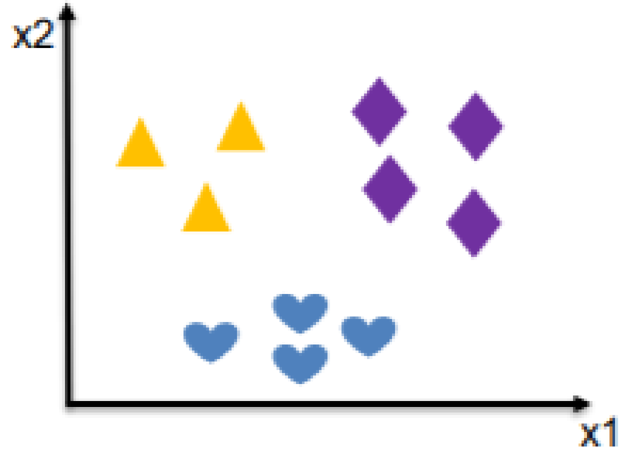
dog 0.997… | pet 0.992… | canine 0.991…
mammal 0.990… | cute 0.990…
puppy 0.989… | animal 0.989…
grass 0.988… | breed 0.965…
purebred 0.959… | adorable 0.956…
pedigree 0.956… | little 0.955…

Input: x
Image pixels

Output: y
Object in image

Multi-class Classification: Formulation

- **C** possible classes: **y** can be 1, 2,…, C
- **m** data points



Data point	x_1	x_2	y
$x^{(1)}, y^{(1)}$	1	4	▲
$x^{(2)}, y^{(2)}$	3	1	♥
$x^{(3)}, y^{(3)}$	3	3	◆
$x^{(4)}, y^{(4)}$	4	4	◆
…………			

Learn:

$P(y = ▲ \mid x; \theta)$

$P(y = ♥ \mid x; \theta)$

$P(y = ◆ \mid x; \theta)$

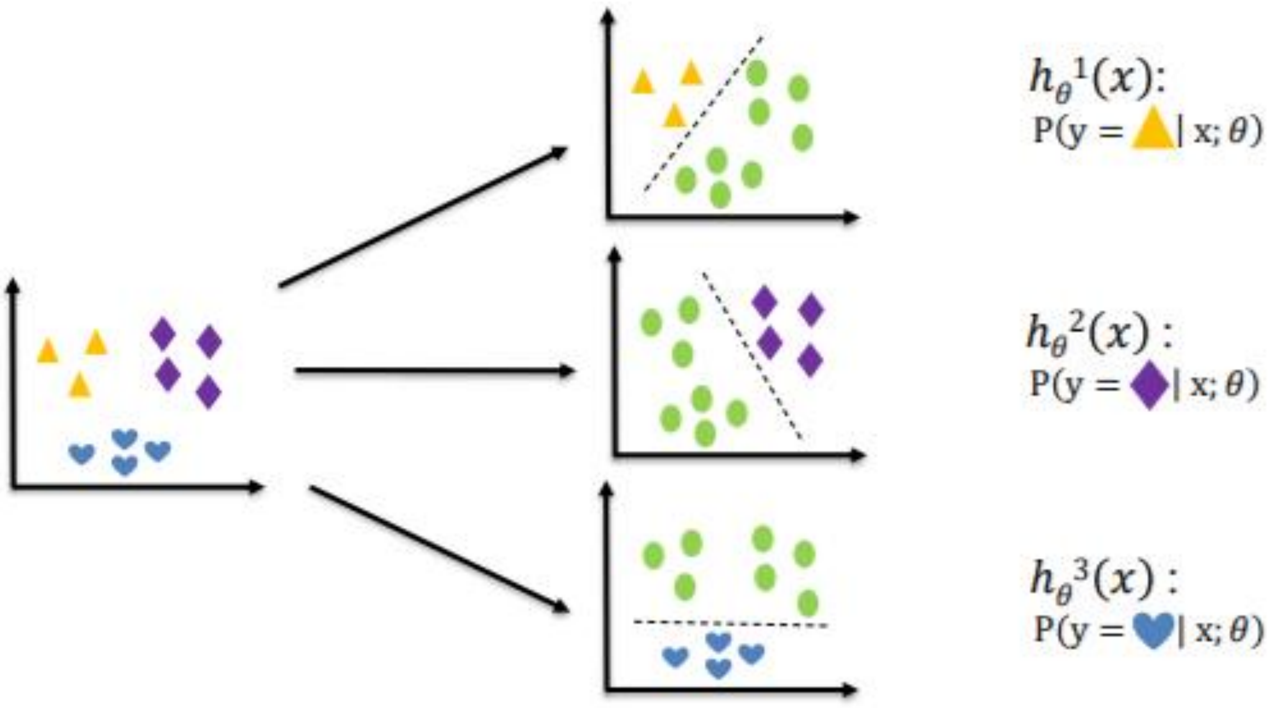
Multi-class Classification: One-vs-all (one-vs-rest)

- In one-vs-All classification, for the N-class instances dataset, we have to generate the N-binary classifier models.

- The number of class labels present in the dataset and the number of generated binary classifiers must be the same.

Multi-class Classification: One-vs-all (one-vs-rest)

- Transform the original classification model to C 2-class models



- On a new input, to make a prediction, pick the class that maximizes: $max_i \; h_\theta^i(x)$

3.5
Evaluating classifiers

Evaluating classifiers: Confusion matrix

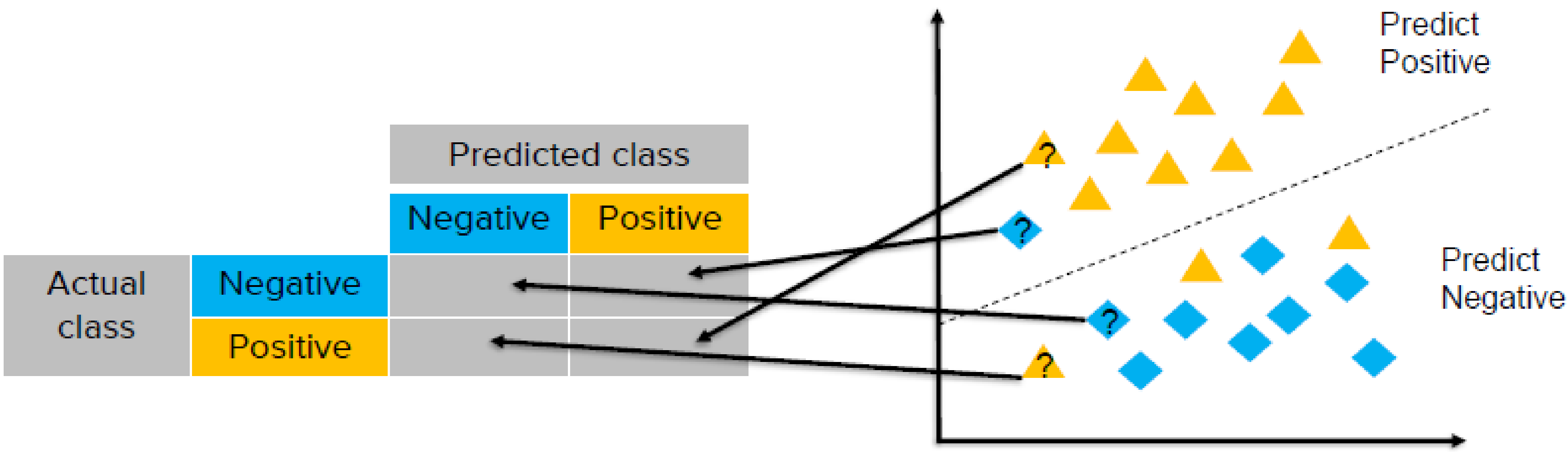
- In many cases in real life problems, you care more about well predicting one class than the others:
  - Cancer detection: care more about cancer gets detected. You can tolerate occasionally false detections but not overlooking real cancers.

Y = 0 (no cancer)	You can tolerate having errors when predicting this class: predict patient has cancer when it's not the case
Y = 1 (cancer)	You can not tolerate having errors when predicting this class: predict patient has no cancer when it's the case

- There is a need for a performance metric that can favor one type of an error than an other.

Evaluating classifiers: Confusion matrix

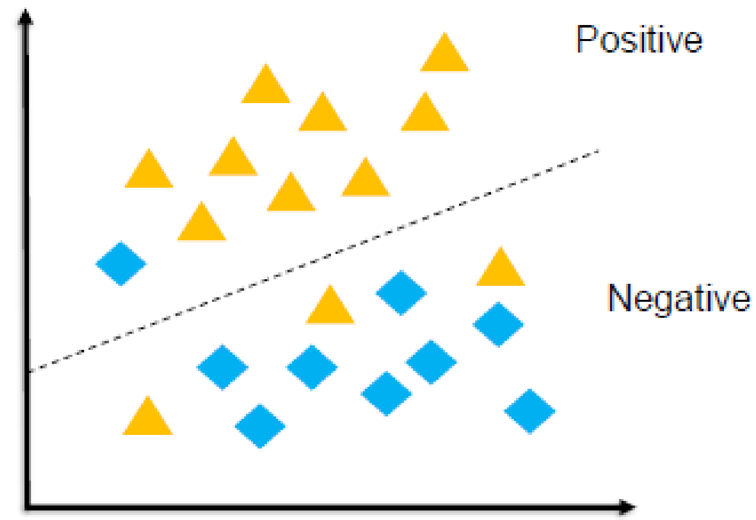
- We have two classes:
  - positive class (y = 1)
  - Negative class (y=0)



- Match each data point to the appropriate cell

Evaluating classifiers: Confusion matrix

		Predicted class	
		Negative	Positive
Actual class	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

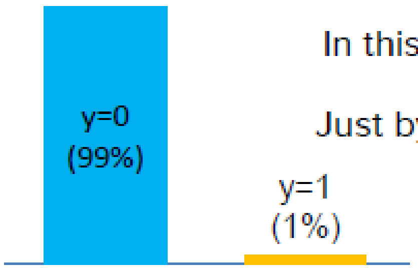


- Sklearn: http://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html

Evaluating classifiers: Accuracy

$$\text{Accuracy} = \frac{\textit{number of data points classified correctly}}{\textit{all data points}}$$

- is 99% accuracy good?
  - Can be excellent, good, mediocre, poor, terrible
  - It depends on the proportion of the classes in your dataset.



In this example, 99% od the data is labeled as the negative class.

Just by predicting everything to y = 0, we can have an accuracy of 99%

- Accuracy is not ideal for skewed (imbalanced) classes !!

Evaluating classifiers: Accuracy

- Accuracy = (TP+TN)/(FP+FN+TP+TN)
- Error of classification = 1-Accuracy = (FP+FN)/(FP+FN+TP+TN)

Evaluating classifiers: Precision - Recall

		Predicted class	
		Negative	Positive
Actual class	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

- Precision for the positive class answers the following question:
- Out of all the examples the classifier labeled as positive, what fraction were correct?

$$\text{Precision} = \frac{\textit{True positive}}{\textit{True positive} + \textit{False positive}} = \frac{9}{9+1} = 90\%$$

Evaluating classifiers: Precision - Recall

		Predicted class	
		Negative	Positive
Actual class	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

- Recall for the positive class answers the following question:
- Out of all the positive examples there were, what fraction did the classifier pick up ?

$$\text{Recall} = \frac{\textit{True positive}}{\textit{True positive} + \textit{False negative}} = \frac{9}{9+3} = 75\%$$

Evaluating classifiers: F1 score

- Given the nature of the problem, you can optimize the model to get a better precision or a better recall.
- It is possible to optimize both by combining precision and recall into a single value, called F1 score.

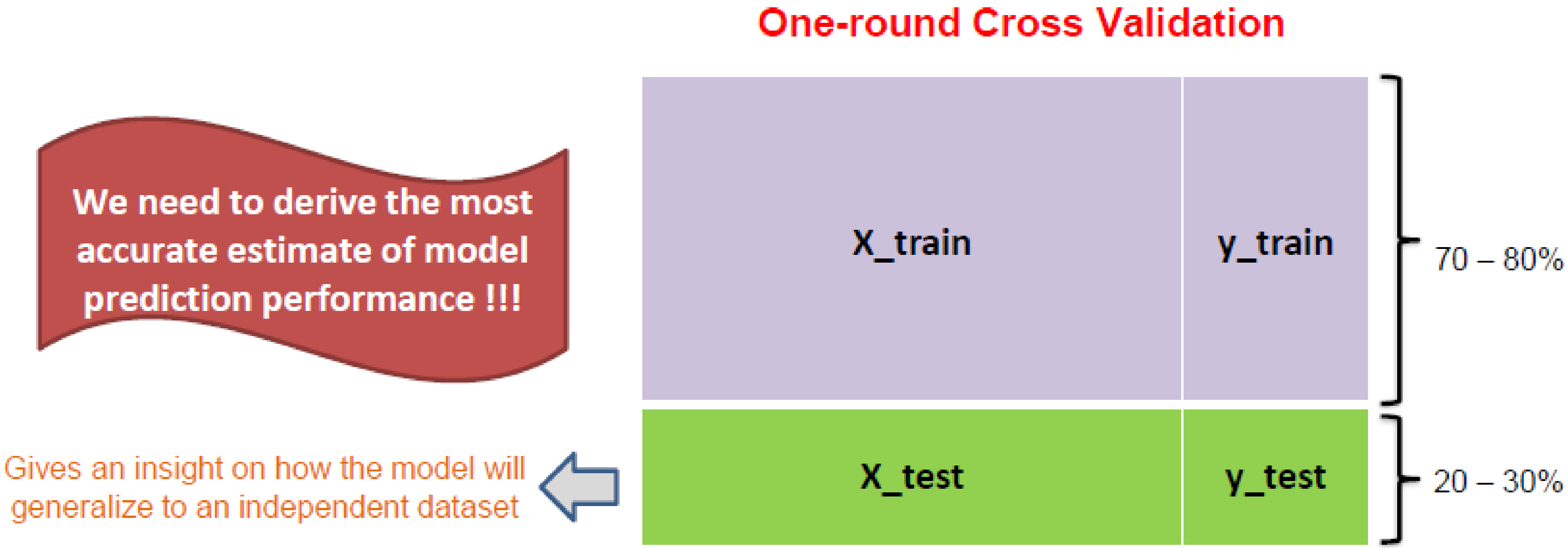
$$F1\ score = 2 * \frac{precsion * recall}{precision + recall} = 81.81\%$$

- Best value at 1, worse at 0.

3.6
K-fold cross-validation

Cross-Validation: Why ?

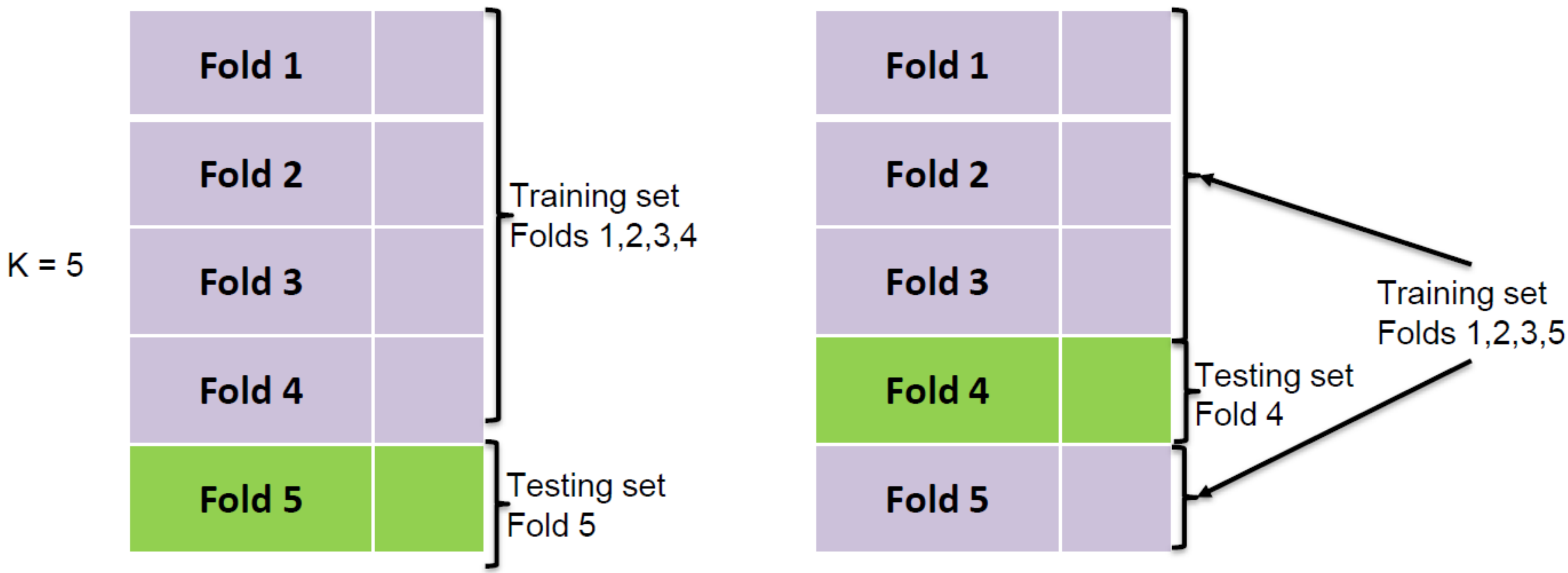
- Data = Training set + Testing set



- Each data point is used only once for training or for testing
- Variability may arise !!
- Moreover: Sometimes we do not have enough data available to make reliable partitions

Cross-Validation: Why ?

- Partition the dataset into **K** folds (bins) of equal size.



- for each k = 1, 2, . . . K, fit the model to the other K −1 parts and compute its error in predicting the k_{th} part.

Cross-Validation example: K-fold

- Run **K** separate learning experiments.
  - Pick testing set
  - Train
  - Test on testing test and compute performance
    - Example: Linear Regression : R2, Logistic Regression: Accuracy

- Average the performance from those **K** experiments
- Typically, K=5 or 10.
- K-Fold is more robust for parameter tuning (choose the regularization parameter, the learning rate, ..)

More about Cross-Validation

- Multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds.

- Common Types of Cross-Validation:
  - Non-exhaustive cross-validation
    - k-fold cross-validation
    - 2-fold cross-validation
  - Exhaustive cross-validation
    - Leave-p-out cross-validation
    - Leave-one-out cross-validation

Practical work

LAB3: Back in 10min!