# Machine Learning

Week 3

## Logistic Regression

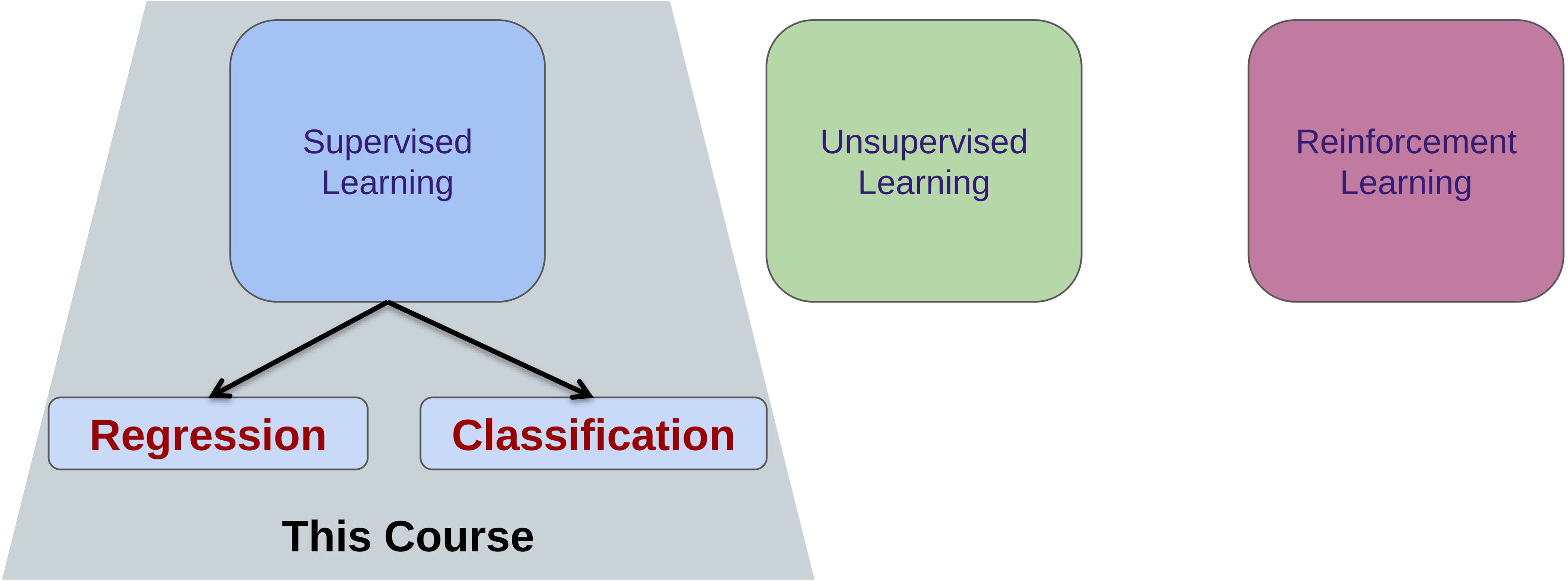
ECE
22/01/2019

# Course overview

1. Introduction to Classification
2. Linear Regression for Classification ?
3. Logistic Regression Model (Binary Classification)
    1. Logistic Regression Model
    2. Logistic Regression Cost function
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**

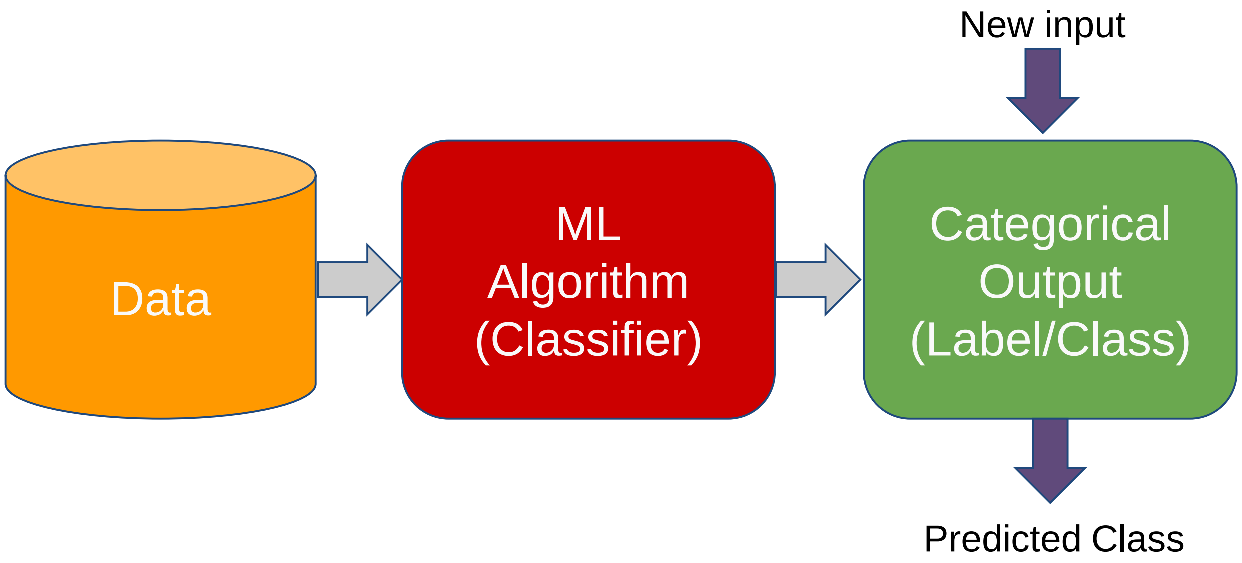


# 3.1
# Introduction to Classification

# Classification

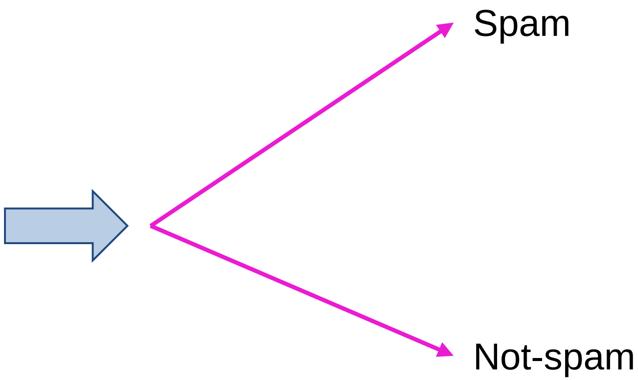
- **Goal:** Inputs are divided into two or more classes, and the ML algorithm must produce a model that assigns unseen inputs to one or more of these classes

- 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

Safe

Risky

**Input: x**

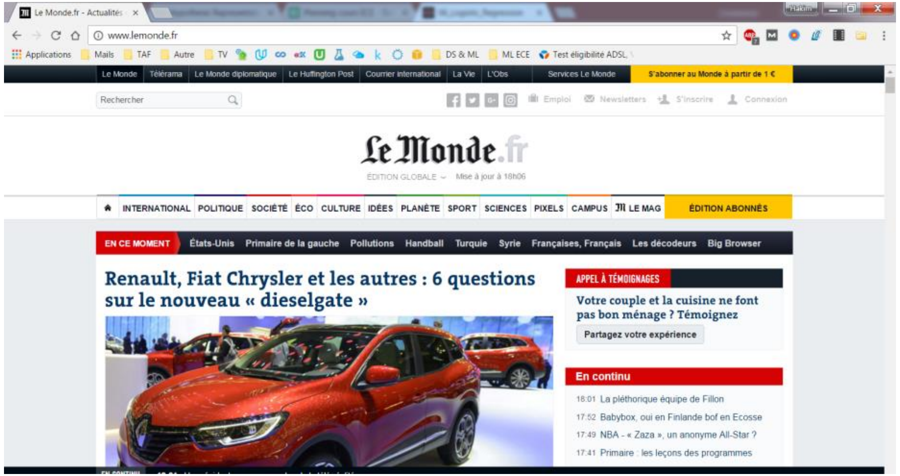
Client’s characteristics
(age, Revenue, credit, etc..)

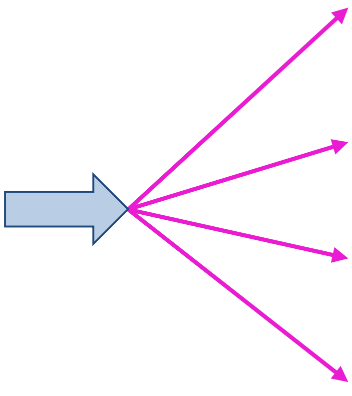
**Output: y**

Loan safety evaluation

# Multi-class Classifier

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



Sport

Education

Technology

Art

**Input: x**

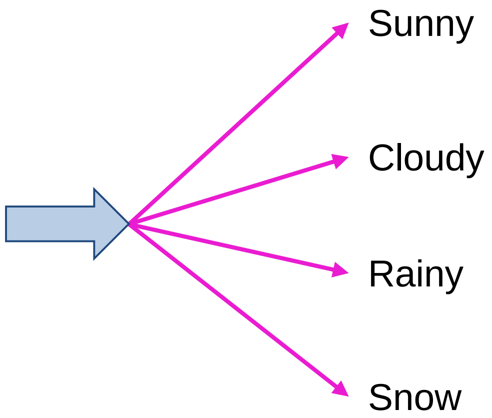
Webpage: title, keywords, etc..

**Output: y**

News category

# Multi-class Classifier

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




**Input: x**

Altitude, region, date, etc…

**Output: y**

Weather status

# 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 ?

Safe

Risky

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

Output: y
Loan safety evaluation

# Can Linear Regression help ?

- The classification problem is just like the regression problem
- Except that the **y values** to be predicted are discrete values.
  - Example: Loan demand evaluation: Safe or risky ?

| Age | Revenue | Charges | … | Loan safety |
|---|---|---|---|---|
| 20 | 1200 | 500 | … | **0** |
| 23 | 1700 | 450 | … | **0** |
| 25 | 1900 | 400 | … | **0** |
| 27 | 2000 | 450 | … | **0** |
| 40 | 2500 | 250 | … | 1 |
| 42 | 2750 | 200 | … | 1 |
| 45 | 2750 | 300 | … | 1 |
| 47 | 3000 | 100 | … | 1 |
| $x_1$ | $x_2$ | $x_3$ | | $y$ |

**?**

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

$y \in \{0, 1\}$

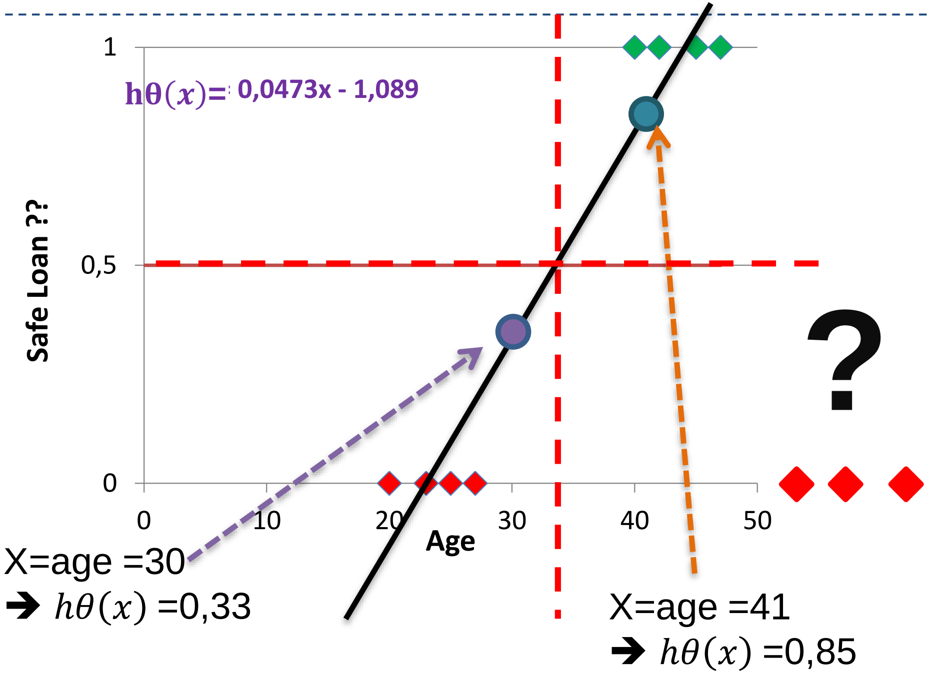
➔ y = 1: Positive Class **Safe**
➔ y = 0: Negative Class **Risky**

# 3.2
# Linear Regression for classification ?

# Can Linear Regression help ?

- One method is to use linear regression

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

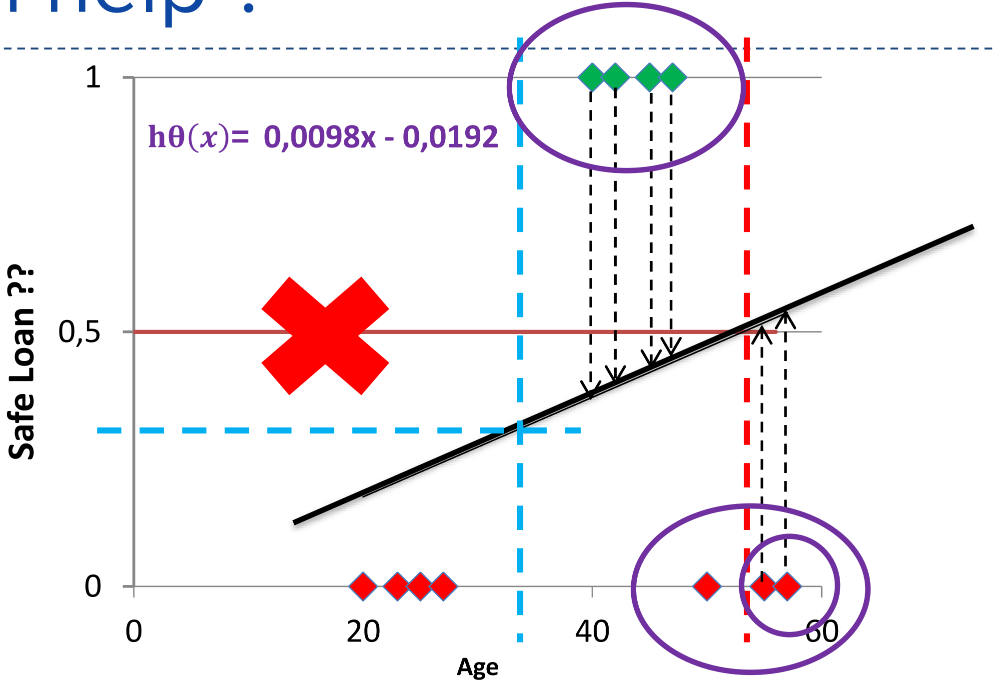



- Map all predictions greater than 0.5 as a 1 and all less than 0.5 as a 0
- Threshold classifier output $h\theta(x)$ at 0.5 :
  - if $h\theta(x) \geq 0{,}5$ ➔ predict: y="1"
  - if $h\theta(x) < 0{,}5$ ➔ predict: y="0"

# 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:
  - $y = 0 \text{ or } 1$
  - But with LR: $h_\theta(x)$ can be > 1 or < 0

2nd

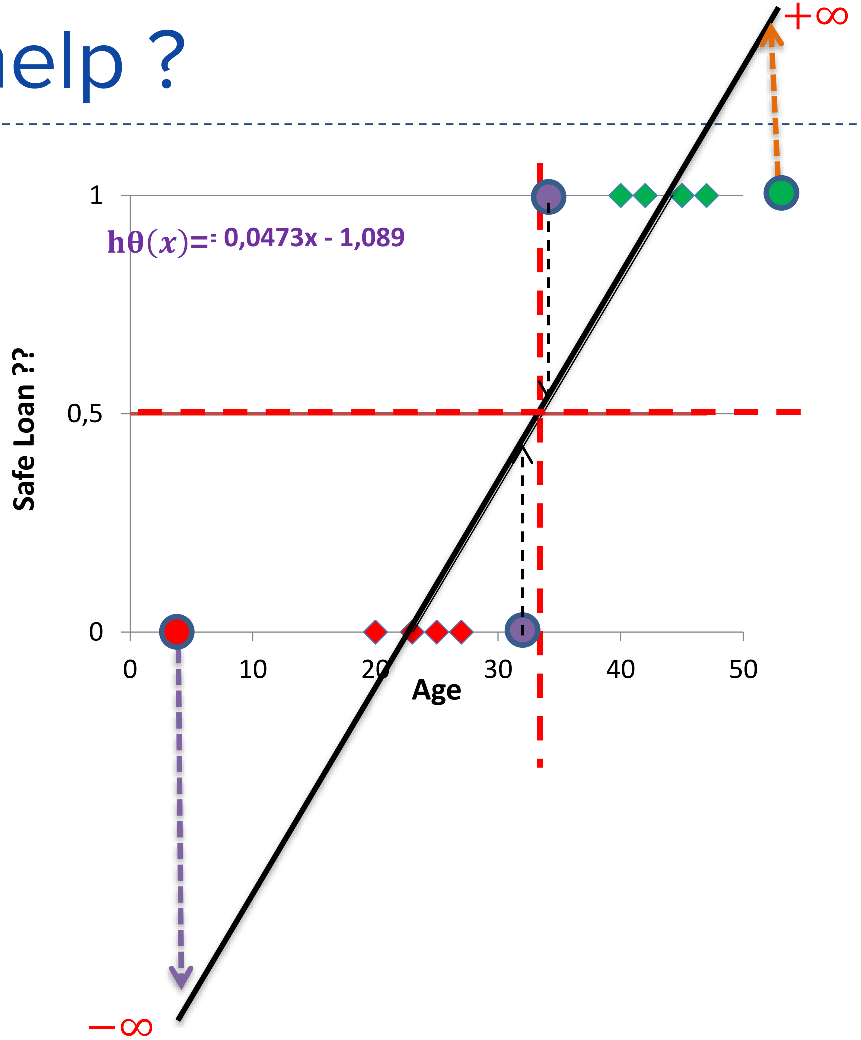
# Can Linear Regression help ?

- 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** $0 \leq h_\theta(x) \leq 1$

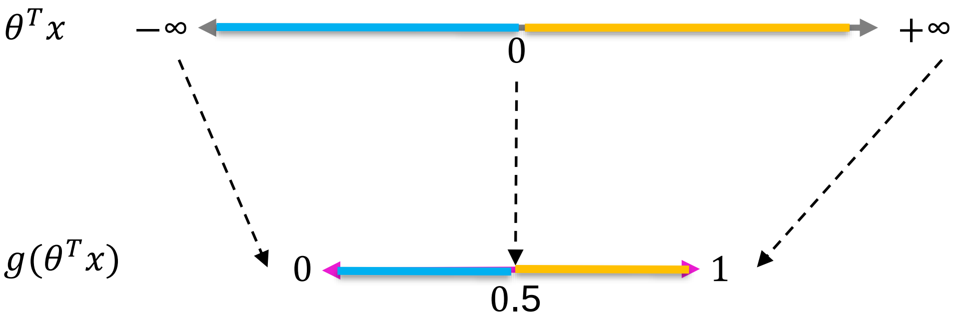


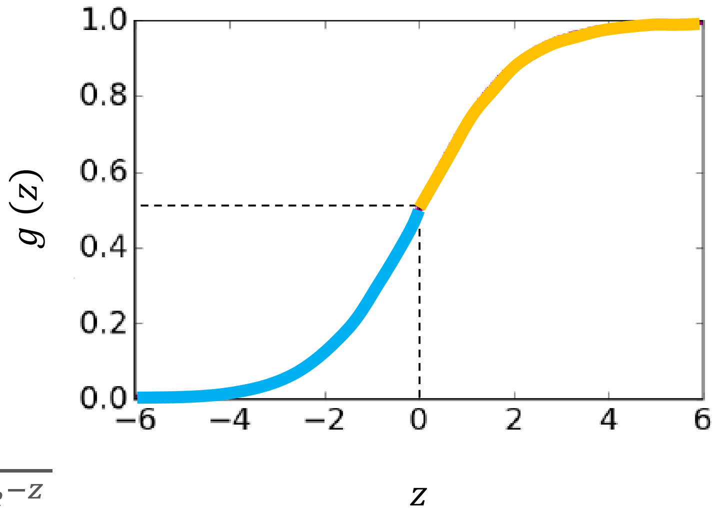
# 3.3
# Logistic Regression

# 3.3.1
# Logistic Regression intuition

# Logistic Regression Model

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




- Use the Sigmoid / Logistic Function : $g(z) = \dfrac{1}{1+e^{-z}}$

- $h_\theta(x) = g\,(\theta^T x) = \dfrac{1}{1+e^{-\theta^T x}}$

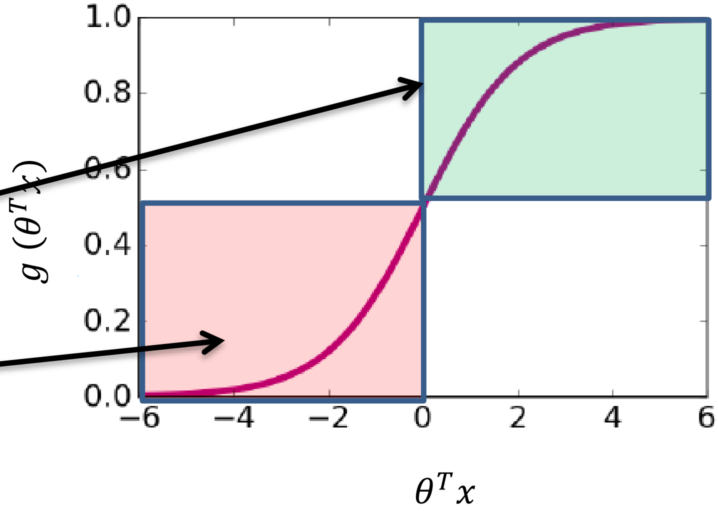
# Interpretation of $h_\theta(x) = g\ (\theta^T x)$

- $h_\theta(x) = g\ (\theta^T x) = \dfrac{1}{1 + e^{-\theta^T x}}$

- $h_\theta(x)$ = **estimated probability that y = 1** given the input x parameterized by $\theta$
  - 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:
  - ❑ $y = 1\ when\ g(\theta^T x) \geq 0{,}5$ ➔ $\theta^T x \geq 0$
  - ❑ $y = 0\ when\ g(\theta^T x) < 0{,}5$ ➔ $\theta^T x < 0$

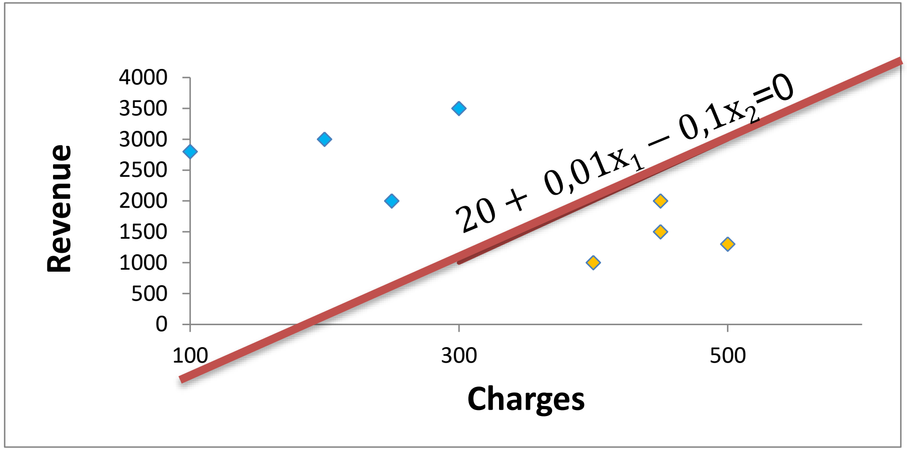


# Decision Boundary

- 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 $g(\theta^T x) \geq 0{,}5$ ➔ $\theta^T x \geq 0$
  - 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 $g(\theta^T x) < 0{,}5$ ➔ $\theta^T x < 0$
  - 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
  - $h_\theta(x) = g\ (\theta_0 + \theta_1 \mathrm{x}_1 + \theta_2 \mathrm{x}_2) = \mathrm{g}\ (20 + 0{,}01\ \#\mathrm{revenue} - 0{,}1\#\mathrm{charges})$

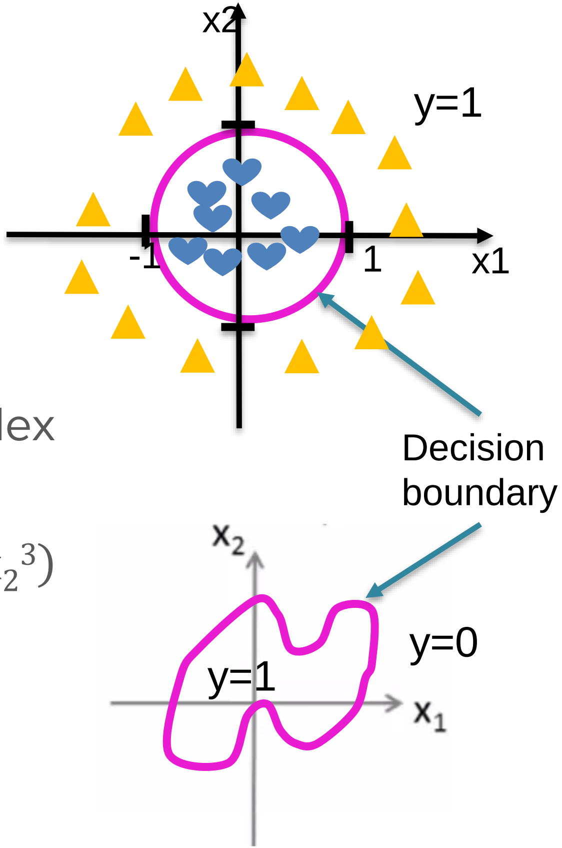
| Charges | Revenue | $\theta^T x$ | $g(\theta^T x)$ | Safe loan ? (prediction) |
|---|---|---|---|---|
| 500 | 1300 | -17 | 4,14E-08 | 0 |
| 450 | 1500 | -10 | 4,54E-05 | 0 |
| 400 | 1000 | -10 | 4,54E-05 | 0 |
| 450 | 2000 | -5 | 6,69E-03 | 0 |
| 250 | 2000 | 15 | 1,00E+00 | 1 |
| 200 | 3000 | 30 |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

| | | | | |
|---|---|---|---|---|
| 450 | 2700 | 2 | 0,880797 | 1 |

# Non-linear decision boundaries

- Example: $h_\theta(x) = g\,(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1{}^2 + \theta_4 x_2{}^2)$
  - $\theta = [-1, 0, 0, 1, 1]$
  - Predict 1 if $-1 + x_1{}^2 + x_2{}^2 \geq 0 \Rightarrow x_1{}^2 + x_2{}^2 \geq 1$
  - Predict 0 if $-1 + x_1{}^2 + x_2{}^2 < 0 \Rightarrow x_1{}^2 + x_2{}^2 < 1$

- As with polynomial regression, we can have more complex decision boundaries by adding higher polynomial terms
  - $h_\theta(x) = g\,(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1{}^2 + \theta_4 x_2{}^2 + \theta_5 x_1{}^3 + \theta_6 x_2{}^3)$



# 3.3.2
# Cost Function (1)

# Cost Fuction 1: Intuition

- Quality Metric

**Real Data**

| Charges | Revenue | $\theta^T x$ | Safe loan ? |
|---|---|---|---|
| 100 | 2800 | 38 | **1** |

| Charges | Revenue | $\theta^T x$ | Safe loan ? |
|---|---|---|---|
| 450 | 1500 | -10 | **0** |

**Predictions ?**

A good model must predict: 1

A good model must predict: 0

Pick θ to maximize:
$\mathrm{P}(\mathrm{y} = 1 |\ \mathrm{x}; \theta)$

That is

$\mathrm{P}(\mathrm{y} = 1 \,|\, \mathrm{x1} = 100, x2 = 2800;\ \theta)$

Pick θ to maximize:
$\mathrm{P}(\mathrm{y} = 0 |\ \mathrm{x}; \theta)$

That is

$\mathrm{P}(\mathrm{y} = 0 \,|\, \mathrm{x1} = 450, x2 = 1500;\ \theta)$

# Cost Fuction 1: Intuition

- Cost function for Linear regression

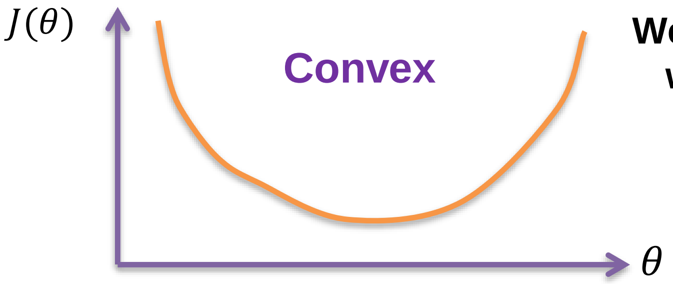
$$J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

| Charges | Revenue | $\theta^T x$ | y =Safe loan (real) | predicion |
|---|---|---|---|---|
| 500 | 1300 | -17 | 0 | 0 |
| 450 | 1500 | -10 | 0 | 0 |
| 400 | 1000 | -10 | 0 | 0 |
| 450 | 2000 | -5 | 0 | 1 |
| 250 | 2000 | 15 | 1 | 1 |
| 200 | 3000 | 30 | 1 | 1 |
| 300 | 3500 | 25 | 1 | 0 |
| 100 | 2800 | 38 | 1 | 1 |

| Cost (error) |
|---|
| $h_\theta(x^{(1)}) - y^{(1)}$ |
| |
| |
| |
| $h_\theta(x^{(i)}) - y^{(i)}$ |
| |
| … |
| … |

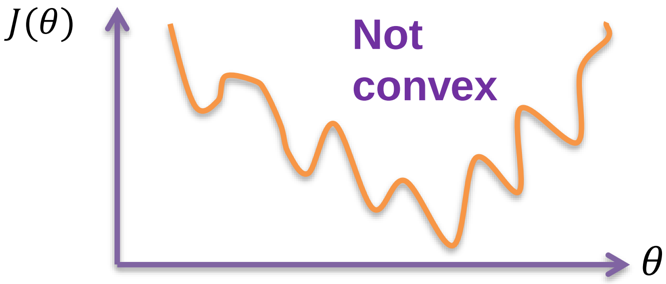
$$\frac{1}{1 + e^{-\theta^T x}}$$

Not linear




**We need a cost fucntion with a convex shape**

?



# Cost Fuction 1: Intuition

- Cost function for Logistic regression model

| Charges | Revenue | $\theta^T x$ | y =Safe loan (real) | predicion | Cost (error) | |
|---|---|---|---|---|---|---|
| 500 | 1300 | -17 | 0 | 0 | Cost$(h_\theta(x^{(1)}), y^{(1)})$ | Cost= 0 |
| 450 | 1500 | -10 | 0 | 0 | … | Cost= 0 |
| 400 | 1000 | -10 | 0 | 0 | … | |
| 450 | 2000 | -5 | 0 | 1 | … | |
| 250 | 2000 | 15 | 1 | 1 | Cost$(h_\theta(x^{(i)}), y^{(i)})$ | |
| 200 | 3000 | 30 | 1 | 1 | … | |
| 300 | 3500 | 25 | 1 | 0 | … | |
| 100 | 2800 | 38 | 1 | 1 | … | Cost= 0 |

**Impose high penalties (costs)**

# Cost Fuction 1: Intuition (for one data point)

- $\mathrm{Cost}(h_\theta(x), y) = \boxed{-(y) * \log(h_\theta(x))} \boxed{-(1-y) * \log(1-h_\theta(x))}$

Predicted value        Real value

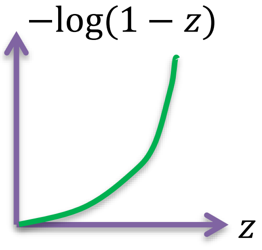
- $\boxed{\textbf{If } \boldsymbol{y=0} : \mathrm{Cost}(h_\theta(x), y) = -(1-y) * \log(1-h_\theta(x))}$
- $\boxed{\textbf{If } \boldsymbol{y=1} : \mathrm{Cost}(h_\theta(x), y) = -(y) * \log(h_\theta(x))}$

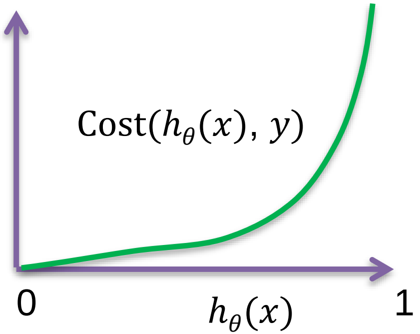
$h_\theta(x) = 0 \Rightarrow \mathrm{Cost}(h_\theta(x), y) = 0$

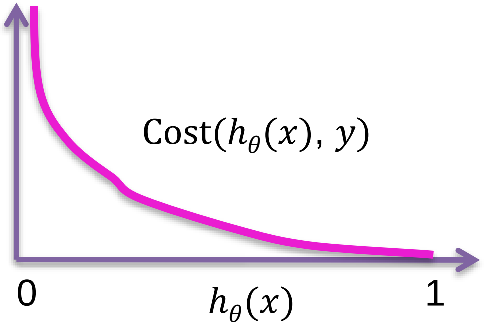
$h_\theta(x) = 1 \Rightarrow \mathrm{Cost}(h_\theta(x), y) = \infty$ High penalty

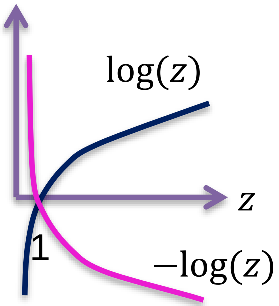
$h_\theta(x) = 1 \Rightarrow \mathrm{Cost}(h_\theta(x), y) = 0$

$h_\theta(x) = 0 \Rightarrow \mathrm{Cost}(h_\theta(x), y) = \infty$ High penalty



# Cost Fuction 1: Intuition (for all data points)

- For one data point: $\text{Cost}(h_\theta(x), y) = -(y) * \log(h_\theta(x)) - (1-y) * \log(1-h_\theta(x))$

- For all data points: $J(\boldsymbol{\theta}) = \sum_{i=1}^{m} \text{Cost}(h_\theta(x), y)$

$$= \frac{1}{m} \sum_{i=1}^{m} (-(y^{(i)}) * \log(h_\theta(x^{(i)})) - (1-y^{(i)}) * \log(1-h_\theta(x^{(i)})))$$

- To find the optimal values of $\boldsymbol{\theta}$: $\min_\theta J(\boldsymbol{\theta})$

- Gradien descent:
  - Initiate with a random set of values for $\theta$
  - At each iteration: update the values of $\boldsymbol{\theta}$
  - For each feature $x_j$ ,$\theta_j = \theta_j - \alpha \sum_{i=1}^{m} (h_\theta\left(x^{(i)}\right) - y^{(i)}).x_j^{(i)}$

# 3.3.2
# Cost Function (2)

# Cost Fuction 2: Maximum-Likelihood

- Quality Metric

**Real Data**

| Charges | Revenue | $\theta^T x$ | Safe loan ? |
| --- | --- | --- | --- |
| 100 | 2800 | 38 | **1** |

| Charges | Revenue | $\theta^T x$ | Safe loan ? |
| --- | --- | --- | --- |
| 450 | 1500 | -10 | **0** |

**Predictions ?**

A good model must predict: 1

A good model must predict: 0

Pick θ to maximize:
$P(y = 1 | x; \theta)$

That is

$P\left(y = 1 \middle| x1 = 100, x2 = 2800;\ \theta\right)$

Pick θ to maximize:
$P(y = 0 | x; \theta)$

That is

$P\left(y = 0 \middle| x1 = 450, x2 = 1500;\ \theta\right)$

# Cost Fuction 2: Maximum-Likelihood

- Quality Metric: Maximizing likelihood, i.e the probability of good predictions

| Charges | Revenue | $\theta^T x$ | y =Safe loan | Choose $\theta$ to maximize |
|---|---|---|---|---|
| 500 | 1300 | -17 | 0 | $\mathrm{P}(\mathrm{y} = 0 \mid \mathrm{x}; \theta)$ |
| 450 | 1500 | -10 | 0 | $\mathrm{P}(\mathrm{y} = 0 \mid \mathrm{x}; \theta)$ |
| 400 | 1000 | -10 | 0 | $\mathrm{P}(\mathrm{y} = 0 \mid \mathrm{x}; \theta)$ |
| 450 | 2000 | -5 | 0 | $\mathrm{P}(\mathrm{y} = 0 \mid \mathrm{x}; \theta)$ |
| 250 | 2000 | 15 | 1 | $\mathrm{P}(\mathrm{y} = 1 \mid \mathrm{x}; \theta)$ |
| 200 | 3000 | 30 | 1 | $\mathrm{P}(\mathrm{y} = 1 \mid \mathrm{x}; \theta)$ |
| 300 | 3500 | 25 | 1 | $\mathrm{P}(\mathrm{y} = 1 \mid \mathrm{x}; \theta)$ |
| 100 | 2800 | 38 | 1 | $\mathrm{P}(\mathrm{y} = 1 \mid \mathrm{x}; \theta)$ |

- Maximize function over all possible $\theta_0, \theta_1, \theta_2$:

$$\max_{\theta_0,\theta_1,\theta_2} \prod_{i=0}^{m} \mathrm{P}(\mathrm{y}_i \mid \mathrm{x}; \theta)$$

# 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…

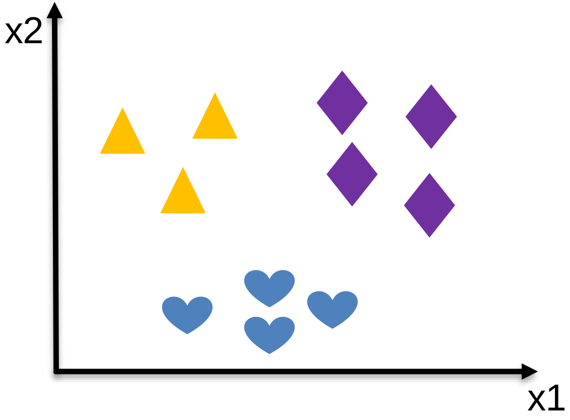
Input: x
Image pixels

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:

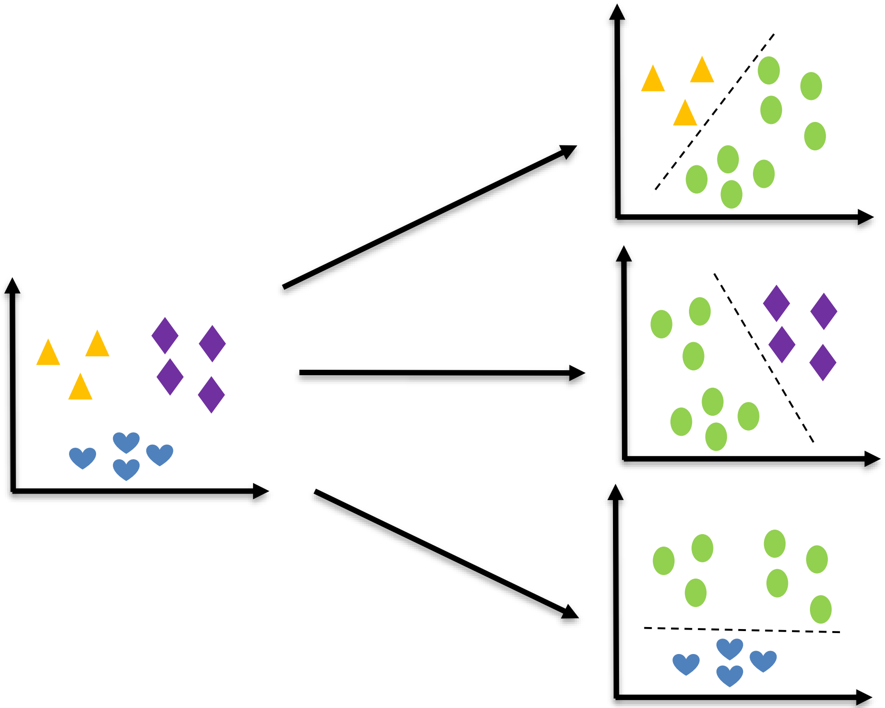
$P(y =$  $\mid x; \theta)$

$P(y =$  $\mid x; \theta)$

$P(y =$  $\mid x; \theta)$

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

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



$h_\theta^{1}(x)$:
$P(y = ▲ | x; \theta)$

$h_\theta^{2}(x)$ :
$P(y = ◆ | x; \theta)$

$h_\theta^{3}(x)$ :
$P(y = ♥ | x; \theta)$

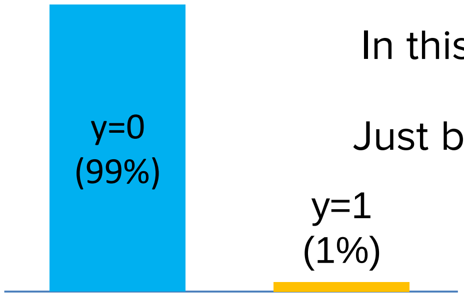
- On a new input, to make a prediction, pick the class that maximizes: $max_i \; h_\theta^{i}(x)$

# 3.5
# Evaluating classifiers

# Evaluating classifiers: Accuracy

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

- 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: 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.

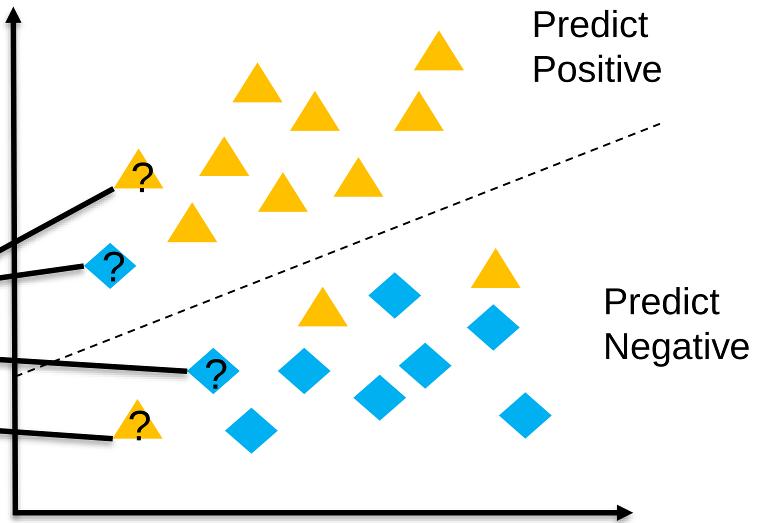
| | |
|---|---|
| Y = 0 (no cancer) | You can tolerate having errors when predicting this class: predict patient has cancer when it’s not the case |
| Y = 1 (cancer) | You can not tolerate having errors when predicting this class: predict patient has no cancer when it’s the case |

- 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)

| | | Predicted class | |
|---|---|---|---|
| | | Negative | Positive |
| Actual class | Negative | | |
| | Positive | | |

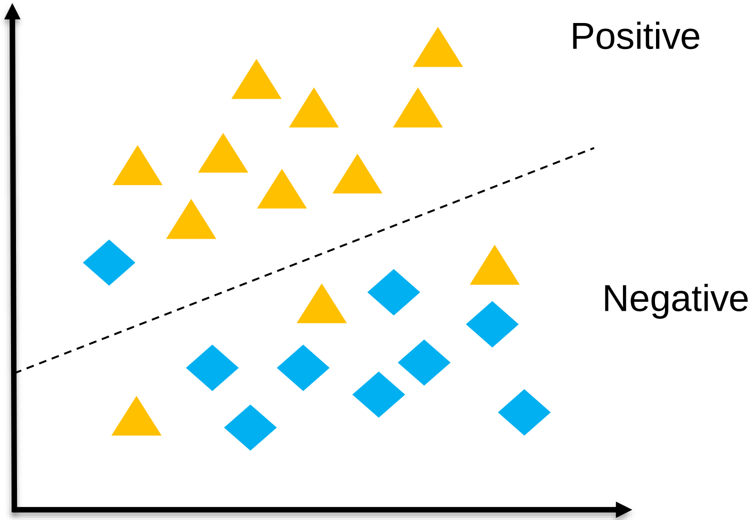



- Match each data point to the appropriate cell

# Evaluating classifiers: Confusion matrix

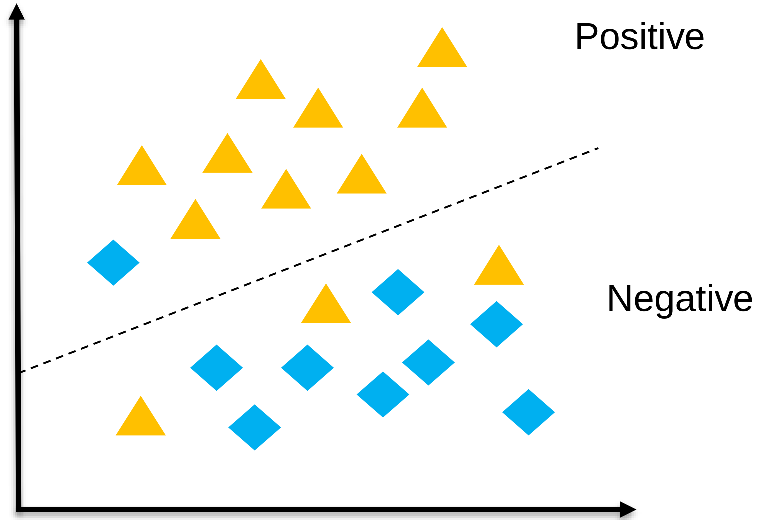
- Fill the cells with the corresponding numbers
  - ▲ : positive class (y = 1)
  - ◆ : Negative class (y=0)

| | | Predicted class | |
|---|---|---|---|
| | | Negative | Positive |
| Actual class | Negative | ? | ? |
| | Positive | ? | ? |



# 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: 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?

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

# 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?

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

# 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.

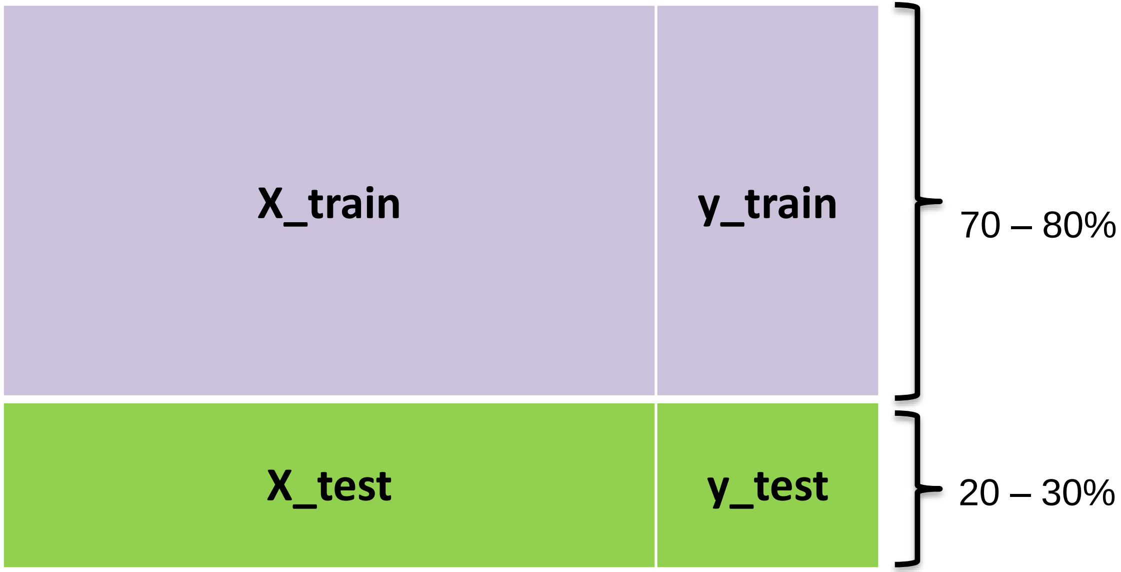
# 3.6
# K-fold cross-validation

# Cross-Validation: Why ?

- **Data = Training set + Testing set**

**One-round Cross Validation**

We need to derive the most accurate estimate of model prediction performance !!!

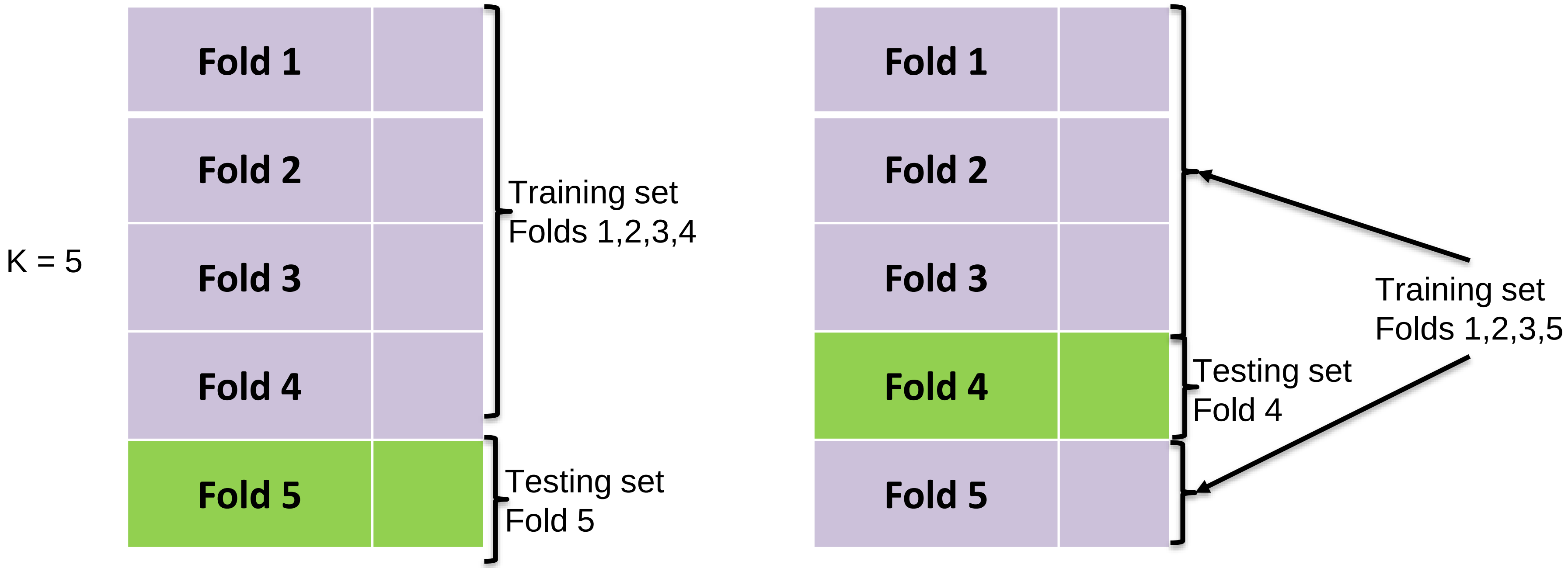



Gives an insight on how the model will generalize to an independent dataset

- 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 example: K-fold

- 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 : $R^2$ , 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

# Thank you for your attention