Politecnico di Bari

How to build a ML System

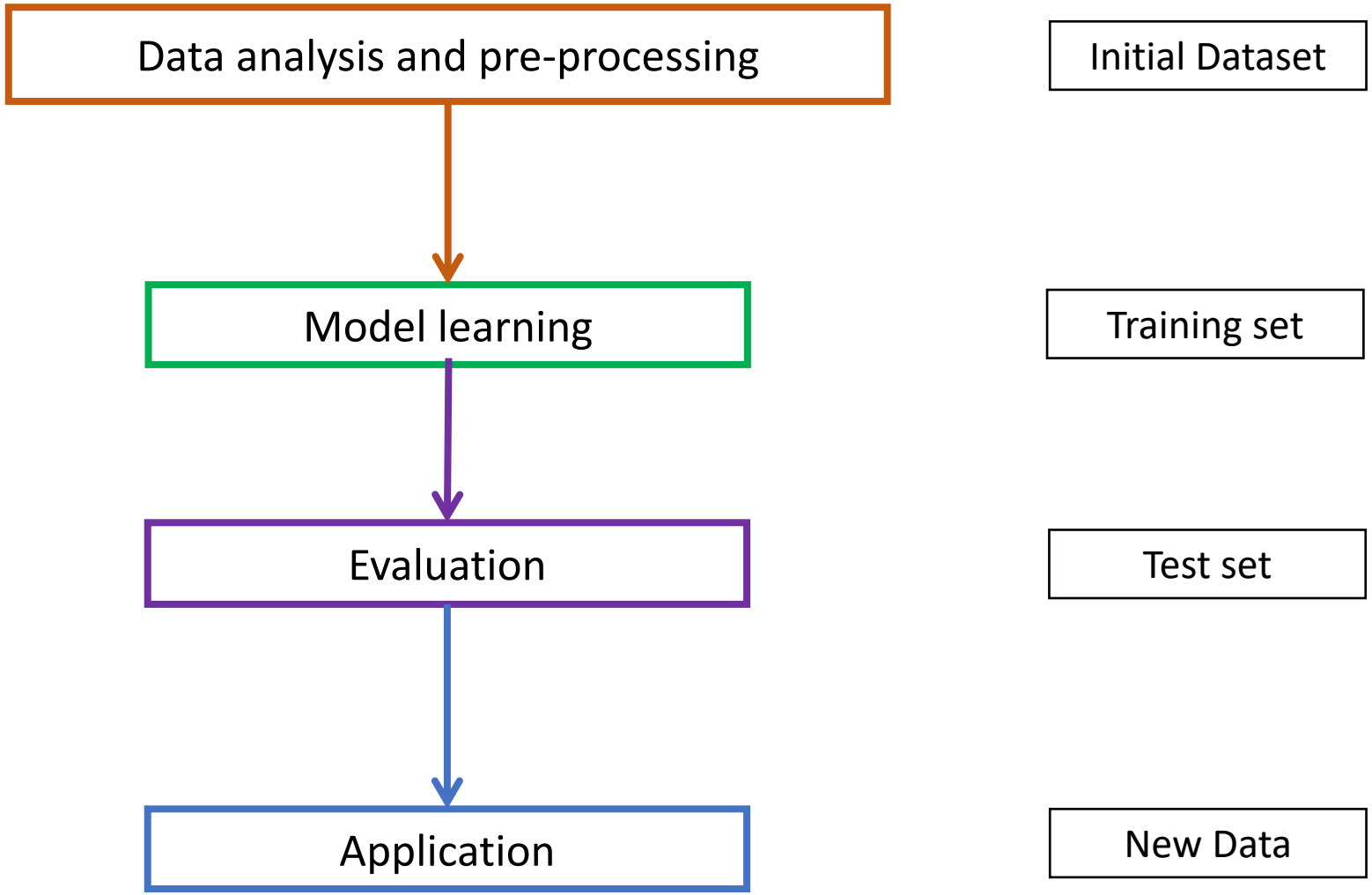
Overview

Machine Learning System

Learning grounds on three parts: representation + optimization + evaluation

- Representation: identify the hypotesis space of the learner and decide which features to use to represent data
- Optimization: choose the method to train the learner (e.g. Gradient descent, greedy search);
- Evaluation: choose an evaluation function (scoring/objective function) to distinguish a good from a bad learner.
- Goal: generalize well from seen examples (training set) on unseen examples.

ML system life cycle (for starters)



Politecnico di Bari

How to build a ML System

What we do first

Data analysis and pre-processing

Real word data are dirty:

- Incomplete: the value of some attributes is missing or interesting attributes are completely missing.
- Inaccurate: data contain wrong values deriving from innacurate or partial observations

GIGO: garbage in – garbage out

An accurate data analysis and cleaning is required. This pre-processing phase takes generally a lot of time.

Data analysis and pre-processing

Pre-processing:

- Cleaning of data: outliers removing, noise removing, duplicates removing
- Changing data:
  - Discretize
  - Aggregate
  - Normalization and re-scaling
- Creating new attributes

Outlier removal example (boxplot)

Quartiles, deciles, and percentiles divide the data set into equal parts

Fractiles	Summary	Symbols
Quartiles	Divide a data set into four equal parts.	Q1 , Q 2, Q3
Deciles	Divide a data set into ten equal parts.	D1 , D2 , D3 ...D9
Percentiles	Divide a data set into one hundred equal parts.	P1, P2, P3 … P99

QUARTILES:

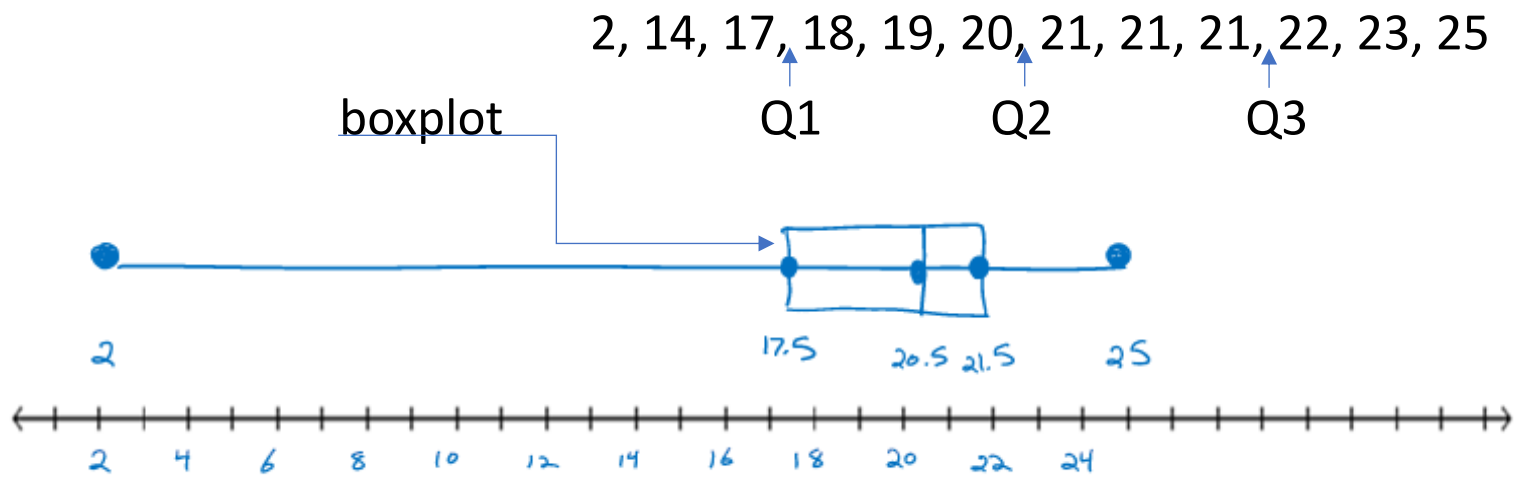
Q2 is equal to the median of the data set.

Q1 is the median of the values that are below Q2 .

Q3 is the median of the values that are above Q2 .

Outlier removal example (boxplot)

Let us see an example:



$$Q_2 = \frac{20+21}{2} = 20.5$$

$$Q_1 = \frac{17+18}{2} = 17.5$$

$$Q_3 = \frac{21+22}{2} = 21.5$$

The interquartile range is defined as IQR= Q3-Q1 , it tells us the spread of the middle 50% of the data. (in this case IQR= 21.5 – 17.5 = 4)

The interquartile range can be used to identify outliers in a data set. If a data value is less than Q1-1.5*IQR or greater than Q3+1.5*IQR , it is considered an outlier.

In this case, the lower fence is 17.5-1.5*4=11.5 hence the value “2” < 11.5 (It is an outlier)

Normalization (recap)

min-max normalization:

$$x = ((x-min)/(max-min)) * (b-a)+a$$

z-score normalization:

$$x = (x-mean(x))/(dev\_std(x))$$

Feature Selection

Selection of relevant features for the learning task.

Sometimes there are many features, some of them could be redudant or not really important

- Domain experts
- Filter: measure the importance of each feature to discriminate across the classes. Es. Information Gain, Entropy, Mutual Information
- Wrappers: iterative method to find a good subset of features using a subset
- Dimensionality reduction (PCA, SVD)

Politecnico di Bari

How to build a ML System

How to evaluate the hypothesis

Evaluating our hypothesis

Area	Price
31770	215000
11622	105000
14267	172000
11160	244000
13830	189900
9978	195500
4920	213500
5005	191500
5389	236500
7500	189000
10000	175900
7980	185000
8402	180400
10176	171500
6820	212000
53504	538000
12134	164000
11394	394432
19138	141000

80%

Training set

$$(x^{(1)}, y^{(1)})$$
$$(x^{(2)}, y^{(2)})$$
$$\vdots$$
$$\vdots$$
$$(x^{(m)}, y^{(m)})$$

20%

Test set

$$(x_{test}^{(1)}, y_{test}^{(1)})$$
$$\vdots$$
$$(x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$$

Training/testing procedure for linear regression

- Learn parameters θ from training data (minimizing training error $J(\theta)$)

- Compute the test set error:

$$J_{test}(\theta) = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} (h_\theta(\mathbf{x^{(i)}}) - y^{(i)})^2$$

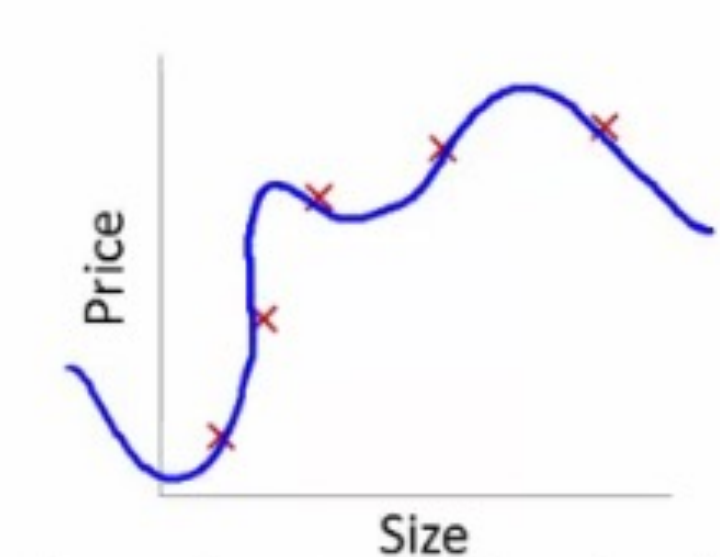
Training/testing procedure for logistic regression

- Learn parameters θ from training data (minimizing training error $J(\theta)$)

- Compute the test set error:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m_{test}}\left(y^{(i)}\log h_\theta(\mathbf{x^{(i)}}) + (1-y^{(i)})\log\left(1-h_\theta(\mathbf{x^{(i)}})\right)\right).$$

The generalization error in practice



- The overfitting example

Once we trained our model to overfit, we know that the error on the training set will be very low but our algorithm does not generalize well.

We can suppose that the actual generalization error is higher than the one we measured on the training data.

This is the reason why we need to evaluate the hypothesis on a portion of data NEVER USED BEFORE.

Model selection

Imagine now that we want to compare different hypothesis in which we change the degree of the polynomial.

$$
\begin{array}{llll}
d1 & 1. & h_\theta(x) = \theta_0 + \theta_1 x & \rightarrow \theta^{(1)} \rightarrow J_{test}(\theta^{(1)}) \\
d2 & 2. & h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 & \rightarrow \theta^{(2)} \rightarrow J_{test}(\theta^{(2)}) \\
d3 & 3. & h_\theta(x) = \theta_0 + \ldots + \theta_3 x^3 & \rightarrow \theta^{(3)} \rightarrow J_{test}(\theta^{(3)}) \\
 & \vdots & \vdots & \\
d10 & 10. & h_\theta(x) = \theta_0 + \ldots + \theta_{10} x^{10} & \rightarrow \theta^{(10)} \rightarrow J_{test}(\theta^{(10)})
\end{array}
$$

Imagine the 5 degree polynomial show the lowest cost function $J_{test}(\theta^{(5)})$.

Does our model generalize well?

The hypothesis we chose is the one that guarantees the best performance on that specific test set (and for this reason is likely to be an optimistic estimation of the generalizazion error).

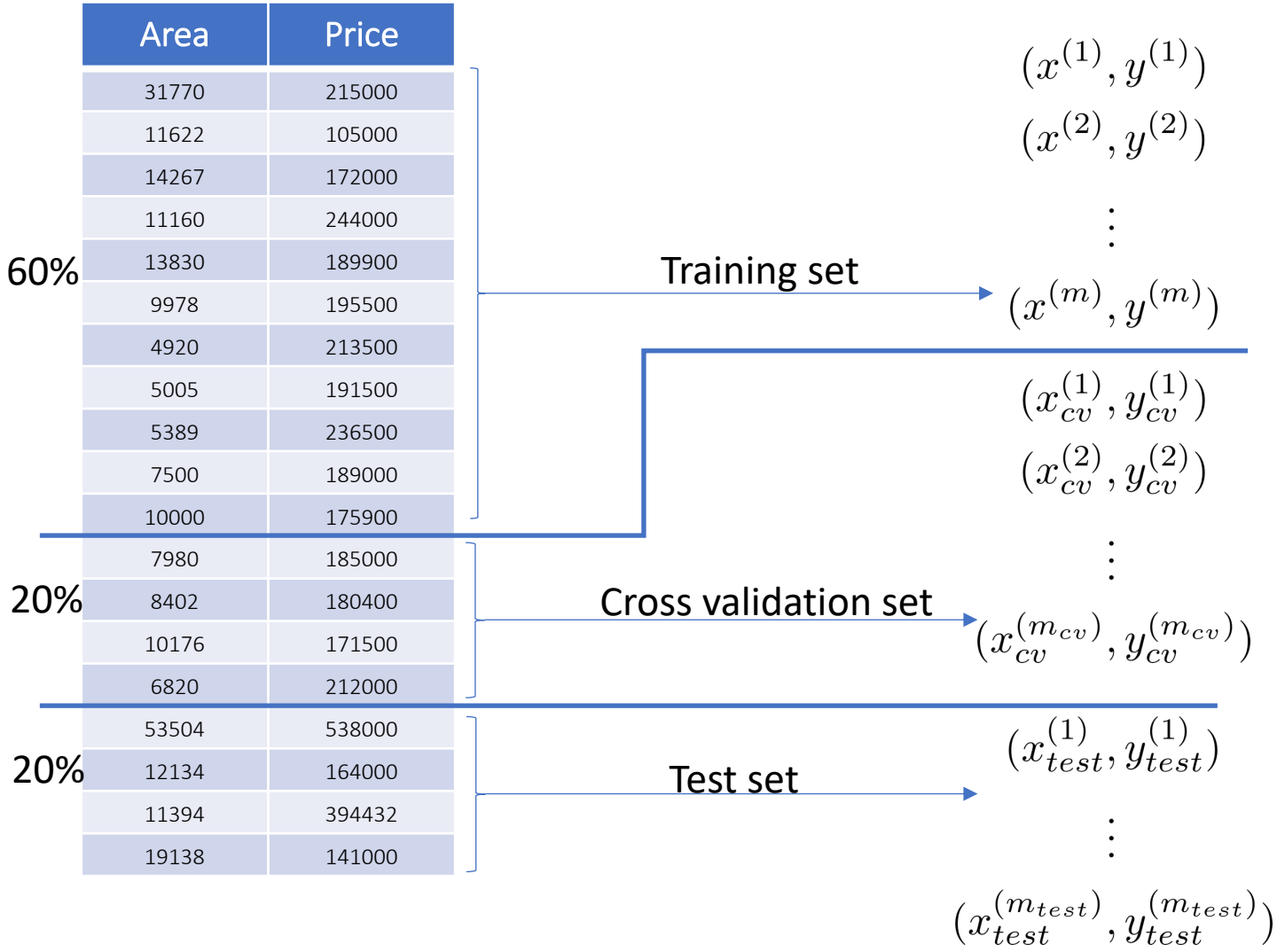
NOW WE NEED ANOTHER TEST SET…

Model Selection: What is the best model?

How can we find it?

Cross Validation

Using a part of the Training Set as Validation Set



Data splitting

Given a dataset, we can split it in three sets:

- Training set, to train the models.
- Validation set, to choose the best model. It is also useful for the feature selection and set the hyper parameters.
- Test set, to evaluate the best model.

Model Selection (recap)

Can we choose the model that gives the best performance on the Training Set? NO!

- We risk overfitting: the model will not be able to generalize.
- The error on the training data is an underestimation of the generalization error.

Model Selection (recap)

Can we choose the model that gives the best performance on the Validation Set? YES!

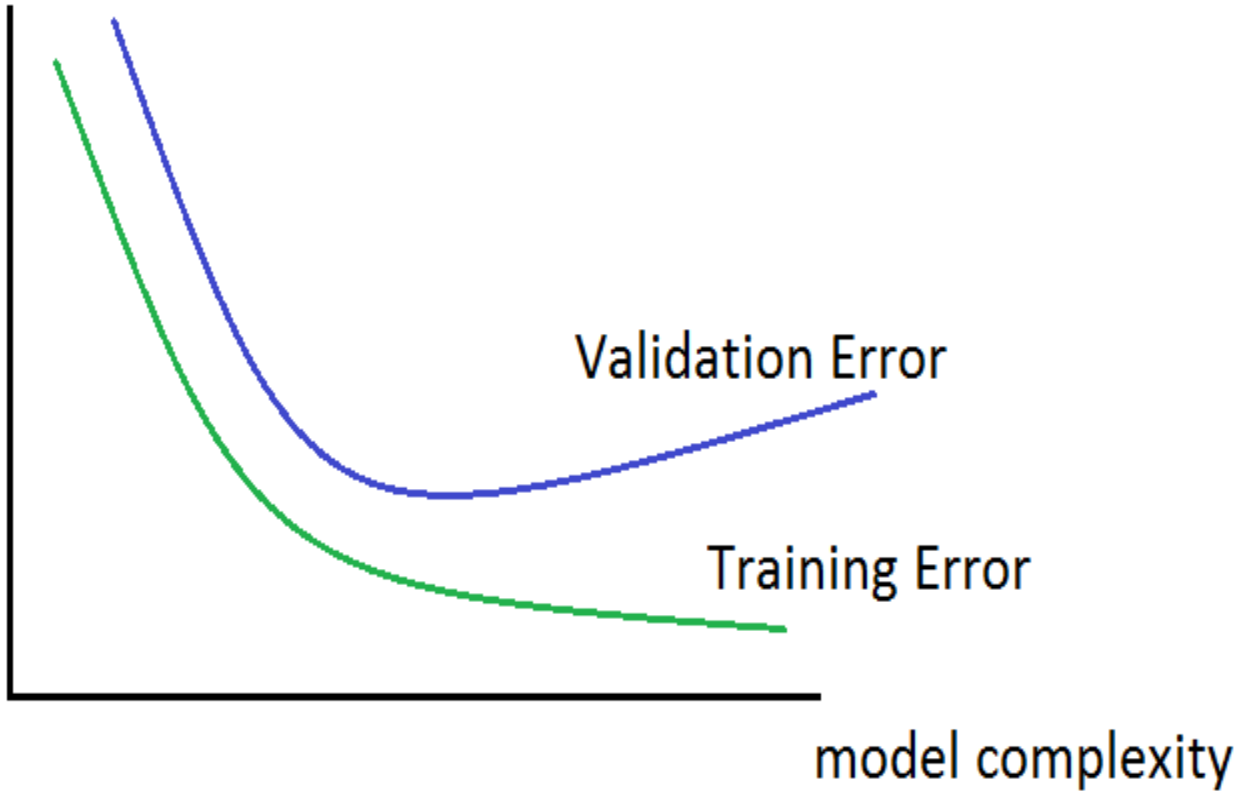
- We must train the model on the training set and the validate it on the validation set.

- In this way we evaluate the generalization error on unseen examples.

Cross Validation

- Usually, the validation set is built by a 1/3 of training set (**hold-out cross validation**) or we can use a **k-folds cross validation.**
- Once we choose a model, it can be trained again on the whole training set
- Cross validation can also be used to find the hyper-parameters. For example, the regularization factor.
- Or it can be used for the selection of a subset of relevant features.

Cross Validation



Model selection

Imagine now that we want to compare different hypothesis in which we change the degree of the polynomial.

$$
\begin{array}{llll}
d1 & 1. & h_\theta(x) = \theta_0 + \theta_1 x & \rightarrow \theta^{(1)} \rightarrow J_{cv}(\theta^{(1)}) \\
d2 & 2. & h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 & \rightarrow \theta^{(2)} \rightarrow J_{cv}(\theta^{(2)}) \\
d3 & 3. & h_\theta(x) = \theta_0 + \ldots + \theta_3 x^3 & \rightarrow \theta^{(3)} \rightarrow J_{cv}(\theta^{(3)}) \\
 & \vdots & \vdots & \\
d10 & 10. & h_\theta(x) = \theta_0 + \ldots + \theta_{10} x^{10} & \rightarrow \theta^{(10)} \rightarrow J_{cv}(\theta^{(10)})
\end{array}
$$

Imagine the 5 degree polynomial show the lowest cost function for the cross validation set.

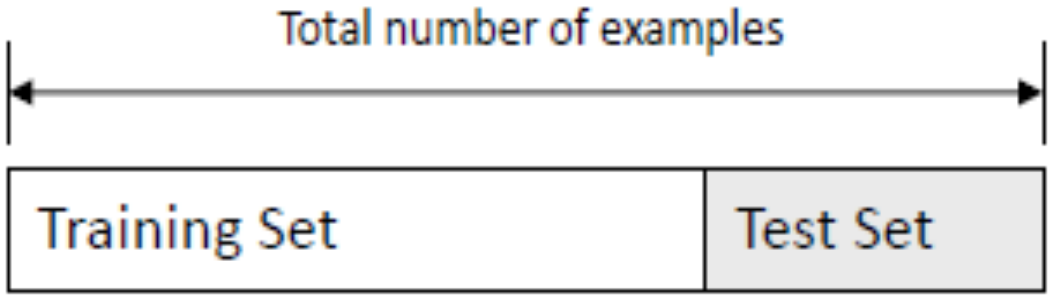
Now we can evaluate the generalization error of the 5 degree polynomial

on the test set data ($J_{test}(\theta^{(5)})$)

Model evaluation

- To finally evaluate the model, we need a Test Set: data not in Training and Validation sets.
- Evaluating the model on a test set is useful to see if the model really generalize well enough
- It is not correct to use the test set for choosing a model for an algorithm. You must use the test set only when the model is chosen and you want to evaluate it
- As we have seen before, there are two ways to split a dataset in Training and Test sets:
  - Holdout
  - K-fold cross validation

Holdout



Drawback : The choice of training and test sets can influence training and test. For instance, a class is not represented equally in the two sets

Solution: We can use the ***stratification in order to better distribuite*** the classes on the two sets

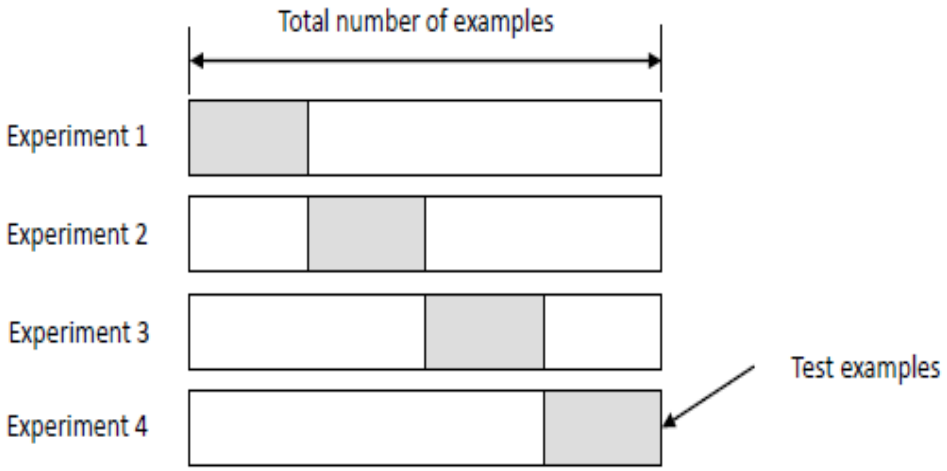
Limitations of using a single training/test partition

- We may not have enough data to make sufficiently large training and test sets
  - a larger test set gives us more reliable estimate of accuracy (i.e., a lower variance estimate)
  - but… a larger training set will be more representative of how much data we actually have for learning process
- A single training set doesn't tell us how sensitive accuracy is to a particular training sample

K-folds Cross validation

The original dataset is randomly partitioned into k equal sized subsets (folds). Of the k folds, a single fold is retained as the test set, and the other folds are used together as training set. The cross-validation process is then repeated k times, with each of the k folds used exactly once as test set. The k results from the folds can then be averaged to produce a single estimation.

Ten-folds cross validations is usually used



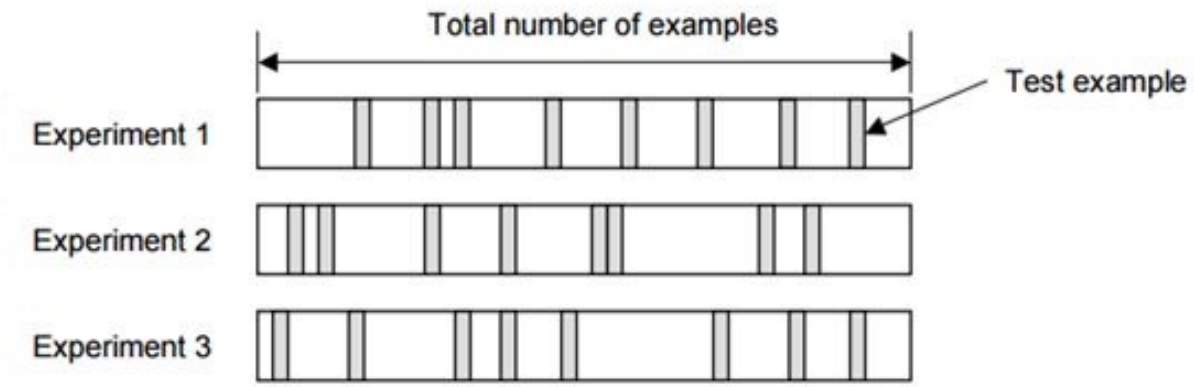
Advantage: all observations are used for both training and test, and each observation is used for test exactly once.

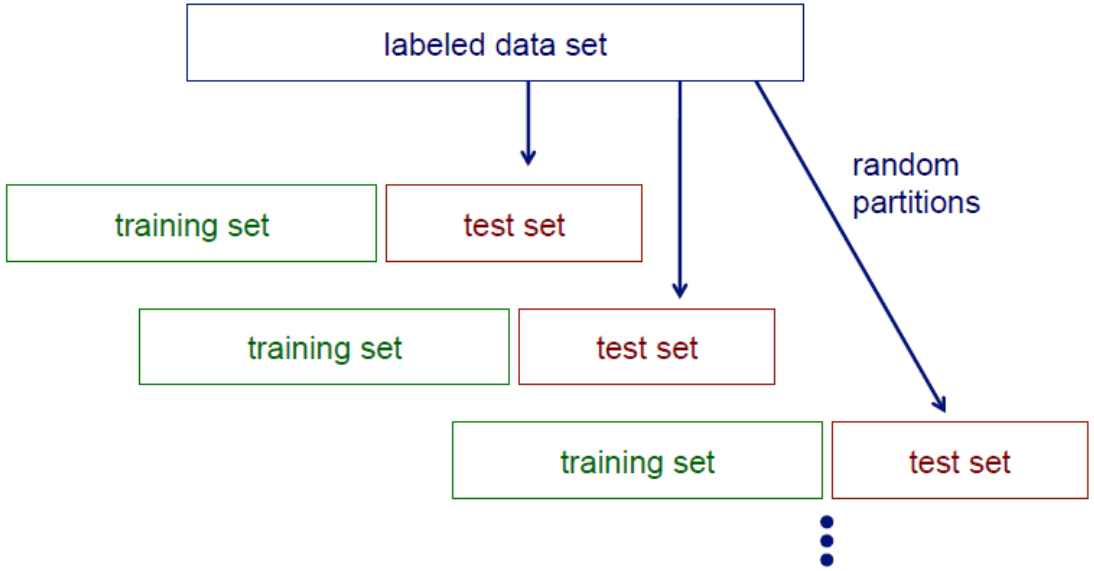
Random subsampling

We can address the second issue by repeatedly randomly partitioning the available data into training and set sets.

In details random subsampling performs K data splits of the dataset

- Each split randomly selects fixed number of examples without replacement
- For each data split we retrain the classifer from scratch with the training examples and estimate the Errors with the test samples





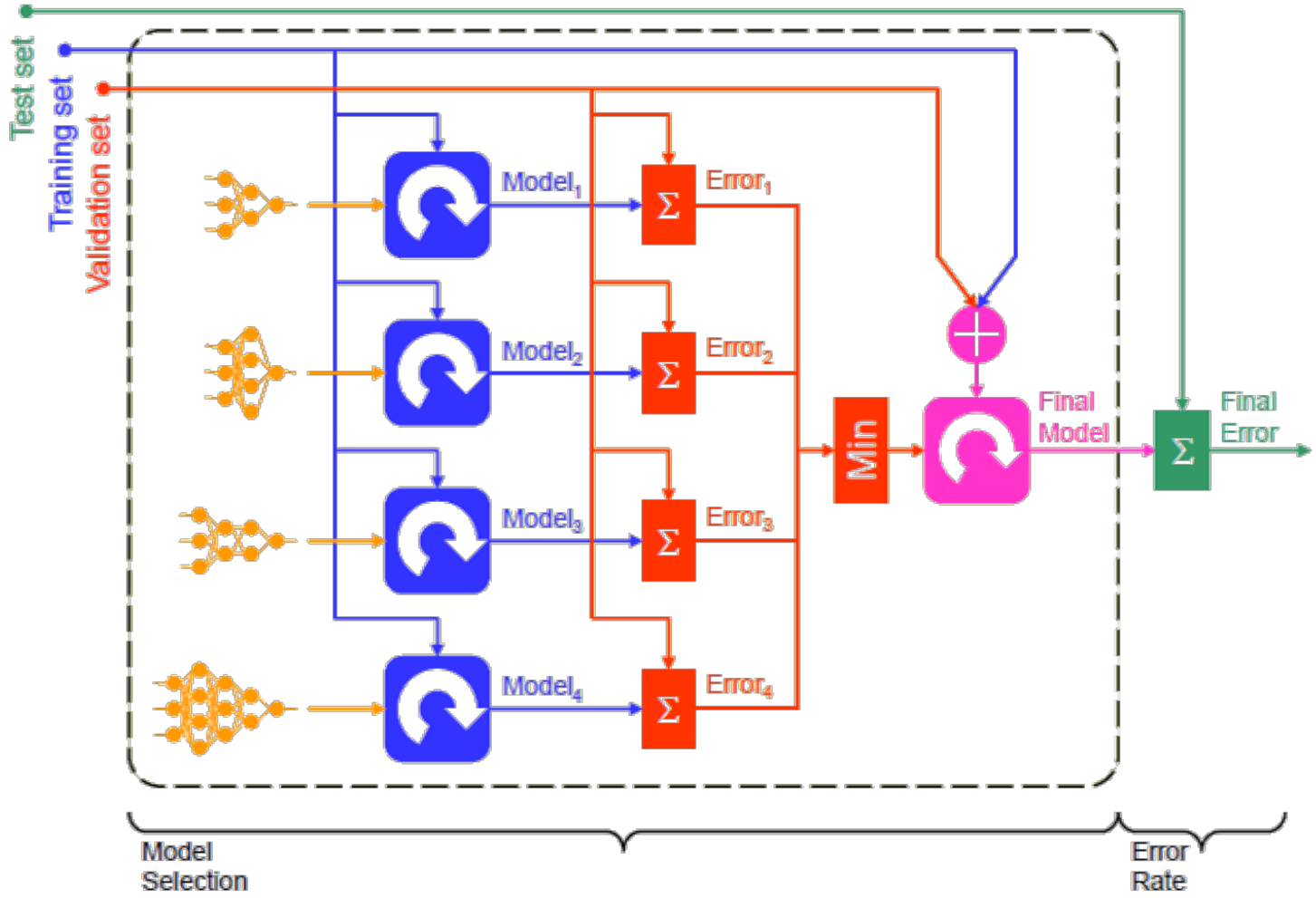
Error estimate on K-folds

After we partitioned the dataset, we commonly use the «leave-one-out» method and we perform evaluation for each fold case.

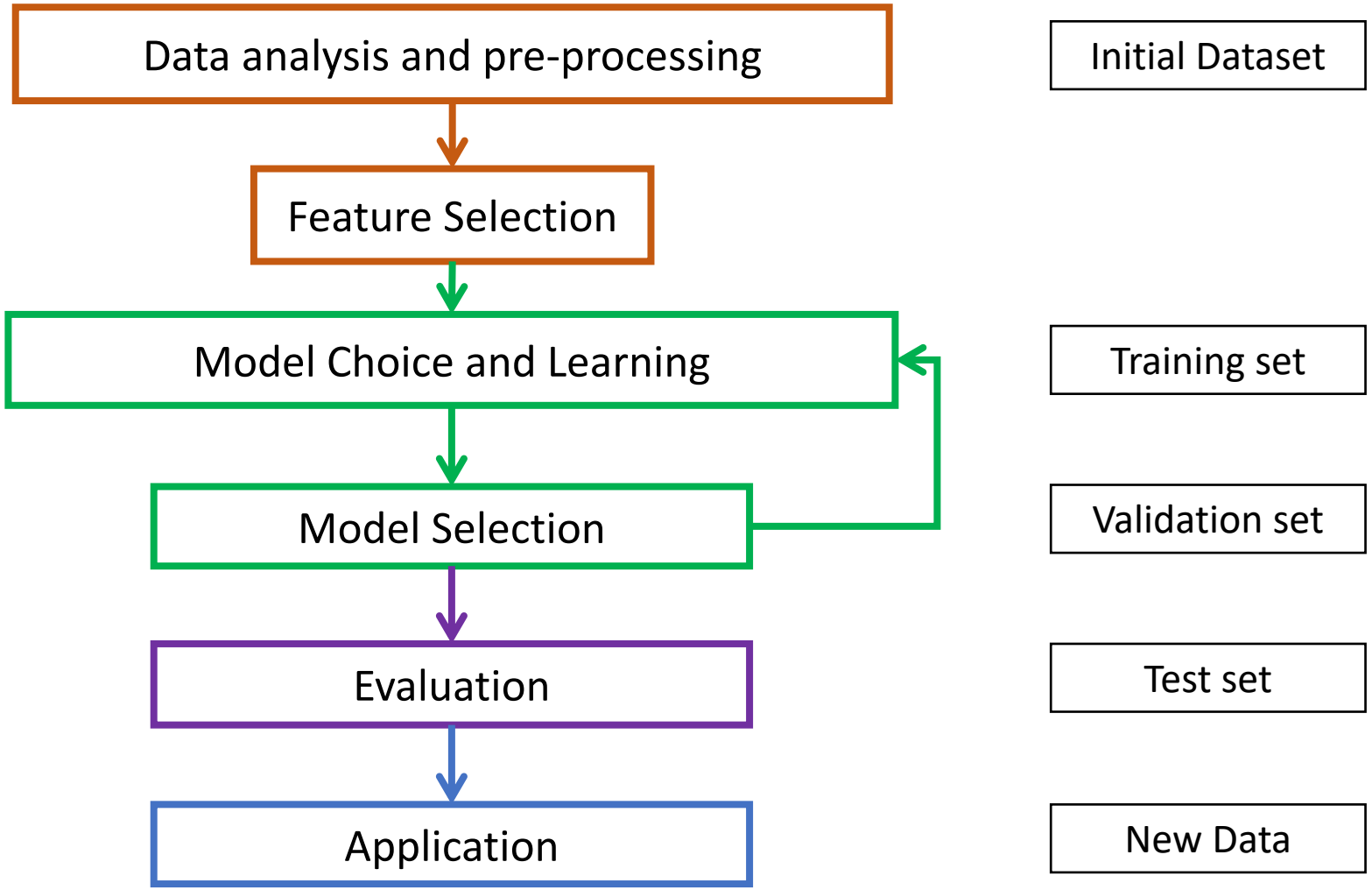
Case i	Train on					Test on	Error
Case1		F2	F3	F4	F5	F1	1.5
Case2	F1		F3	F4	F5	F2	0.5
Case3	F1	F2		F4	F5	F3	0.3
Case4	F1	F2	F3		F5	F4	0.9
Case5	F1	F2	F3	F4		F5	1.1

$$Error = \frac{1}{k}\sum_{i=1}^{k} Error_i$$

Cross-Validation overview



ML system life cycle (revisited)



Politecnico di Bari

How to build a ML System

How to choose the next hypothesis

Debugging a learning algorithm:

Suppose you have implemented a reguarized linear regression to predict housing prices.

$$J(\theta) = \frac{1}{2m}\left(\sum_{i=1}^{m}(h(\mathbf{x}^{(\mathbf{i})}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right)$$

When you test your hypothesis on a new set of houses, you find that your regressor performs very bad with very large errors. What should you try next?

- Get more training examples
- Try smaller sets of features
- Try getting additional features
- Try adding polynomial features
- Try decreasing λ
- Try increasing λ

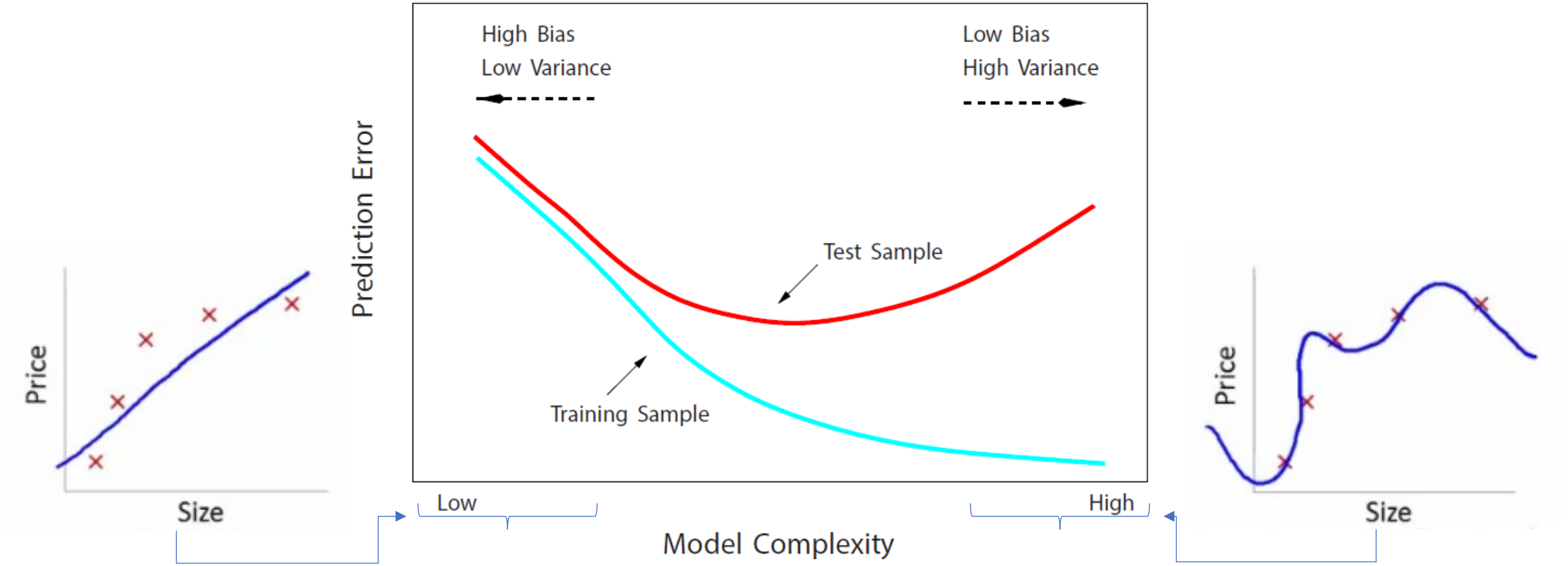
Machine learning diagnostic

Diagnostic: A test that you can run to gain insight what is/isn't working with a learning algorithm, and gain guidance as to how best to improve its performance.

Diagnostics can take time to implement, but doing so can be a very good use of your time.

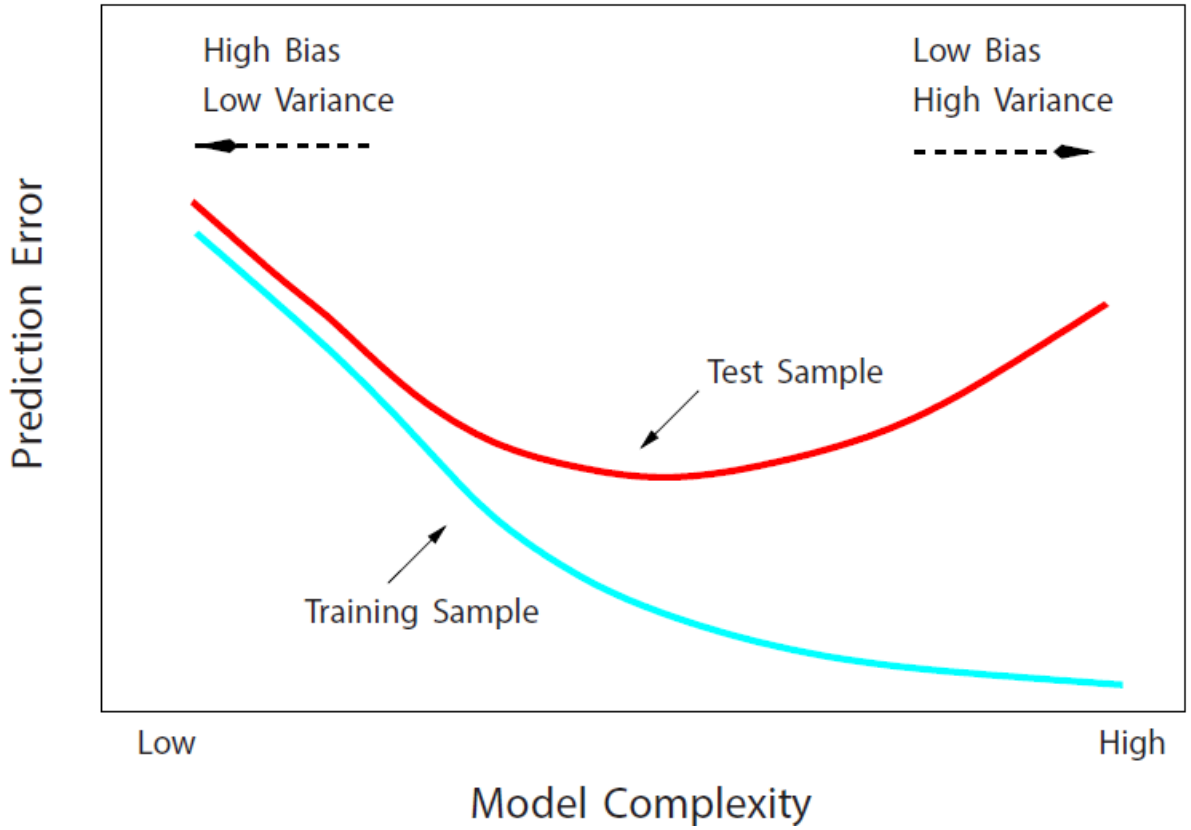
Diagnosing bias/variance

Let us recap the Bias variance behavior w.r.t. the complexity of the model



Diagnosing bias/variance

Suppose your learning algorithm is performing less well than you were hoping. ($J_{cv}(\theta)$ or $J_{test}(\theta)$ is high) Is it a bias problem or a variance problem?



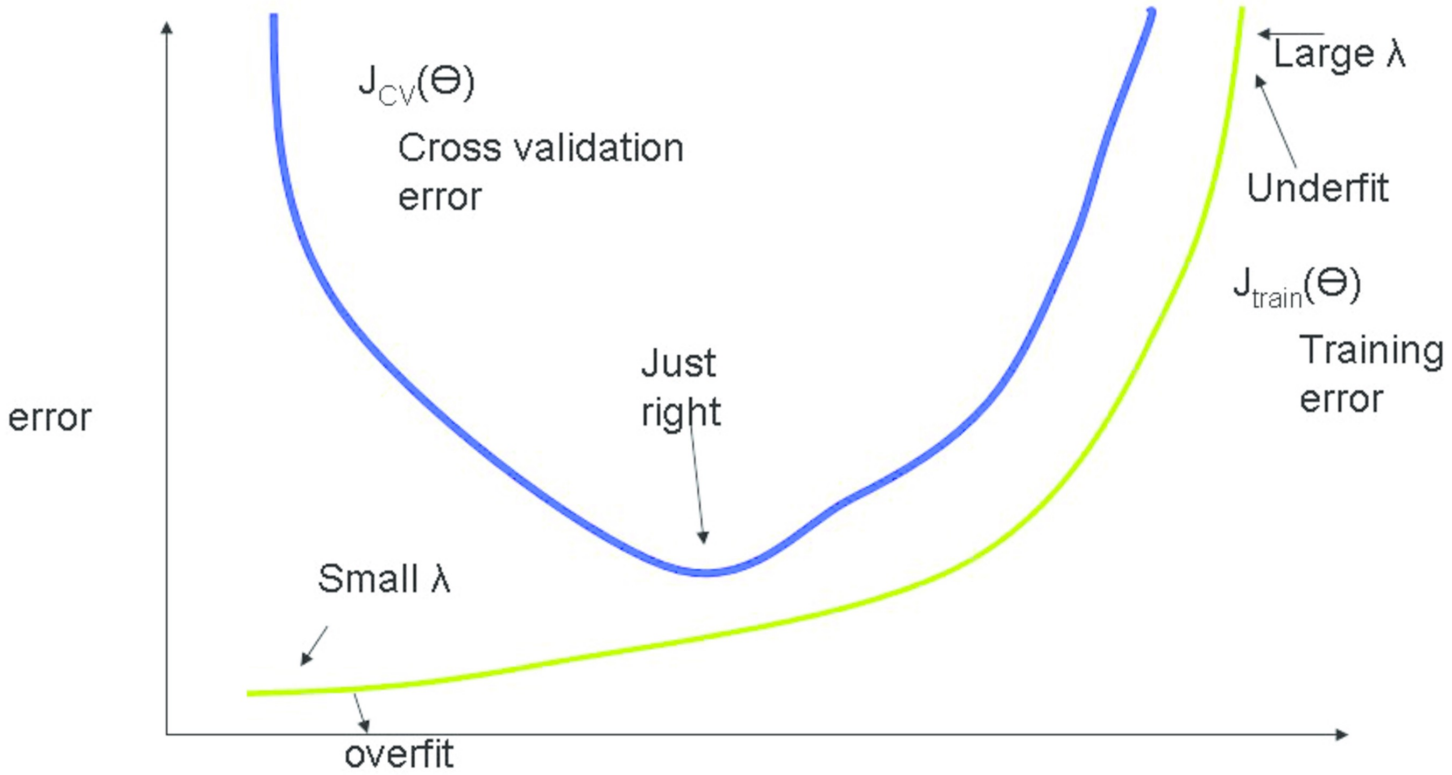
Bias (underfit):

$$J_{train}(\theta) \quad \text{will be high}$$
$$J_{cv}(\theta) \approx J_{test}(\theta)$$

Variance (overfit):

$$J_{train}(\theta) \quad \text{will be low}$$
$$J_{cv}(\theta) \gg J_{train}(\theta)$$

Diagnosing Lambda (regularization)



Learning curves

Learning curve is a good technique

- To sanity-check a model
- To improve performance

A learning curve is a plot where we have two functions of m (m is a set size):

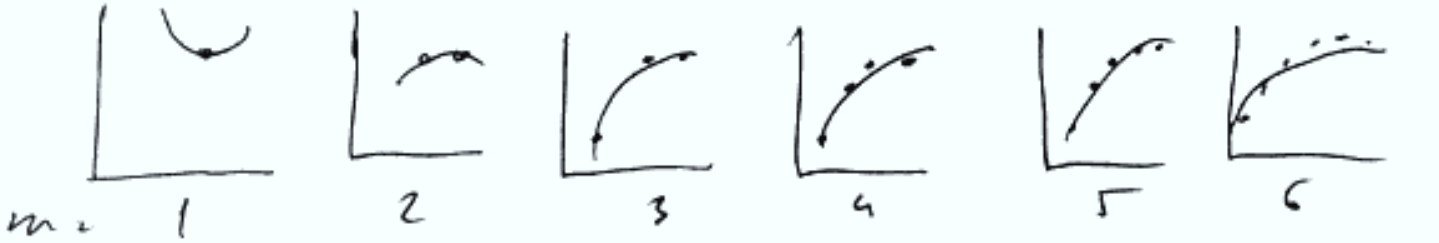
- training set error $J_{train}(\theta)$
- the cross-validation error $J_{cv}(\theta)$

We can artificially reduce our training set size.

We start from m=1, then m=2 and so on

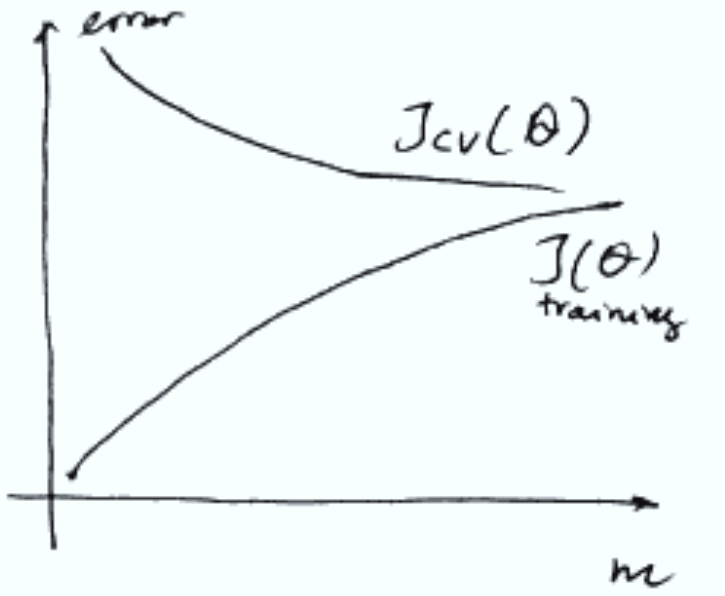
Learning curves (cont.d)

Suppose we have the following model: $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$



For each m we compute $J_{train}(\theta)$ and $J_{cv}(\theta)$ and plot the values.

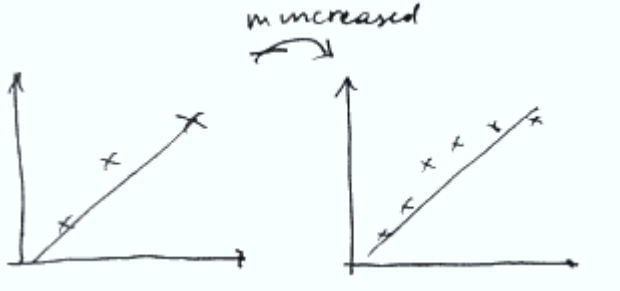
This is the learning curve of the model:



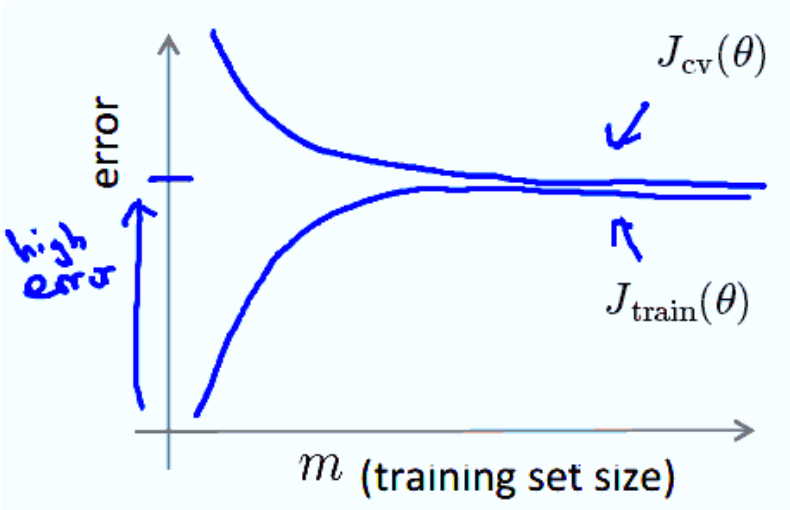
Diagnose High Bias (Underfitting)

Suppose we want to fit a straight line to our data: $h_\theta(x) = \theta_0 + \theta_1 x$

As m increases, we have almost the same line:



If we draw the learning curves, we'll have



We see that as m grows

$J_{cv}(\theta) \rightarrow J_{train}(\theta)$

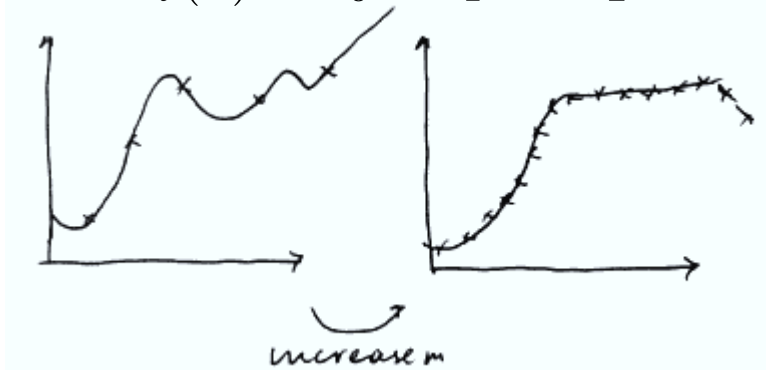
and both errors are high

If learning algorithm is suffering from high bias, getting more examples will not help

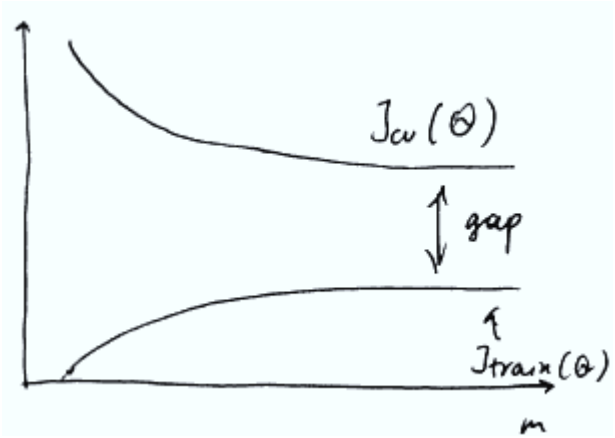
Diagnose High Variance (Overfitting)

Now suppose we have a model

with polynomial of very high order: $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \ldots + \theta_{100} x^{100}$



at the beginning we very much overfit, as we increase m, we are still able to fit the data well.



So, we can see that as m increases, $J_{train}(\theta)$ increases.
(we have more and more data - so it's harder and harder to fit $h_\theta(x)$), but it increases very slowly.
On the other hand, $J_{cv}(\theta)$ decreases, but also very very slowly and there's a huge gap between these 2 to fill that gap we need many many more training examples.
If a learning algorithm is suffering from high variance (i.e. it overfits), getting more data is likely to help

Debugging a learning algorithm (revisited):

Suppose you have implemented a reguarized linear regression to predict housing prices.

$$J(\theta) = \frac{1}{2m}\left(\sum_{i=1}^{m}(h(\mathbf{x}^{(\mathbf{i})}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right)$$

When you test your hypothesis on a new set of houses, you find that your regressor performs very bad with very large errors. What should you try next?

- Get more training examples → fixes high variance
- Try smaller sets of features → fixes high variance
- Try getting additional features → fixes high bias
- Try adding polynomial features → fixes high bias
- Try decreasing λ → fixes high bias
- Try increasing λ → fixes high variance

Politecnico di Bari

How to build a ML System

How to measure if a ML System works well

Evaluation metrics

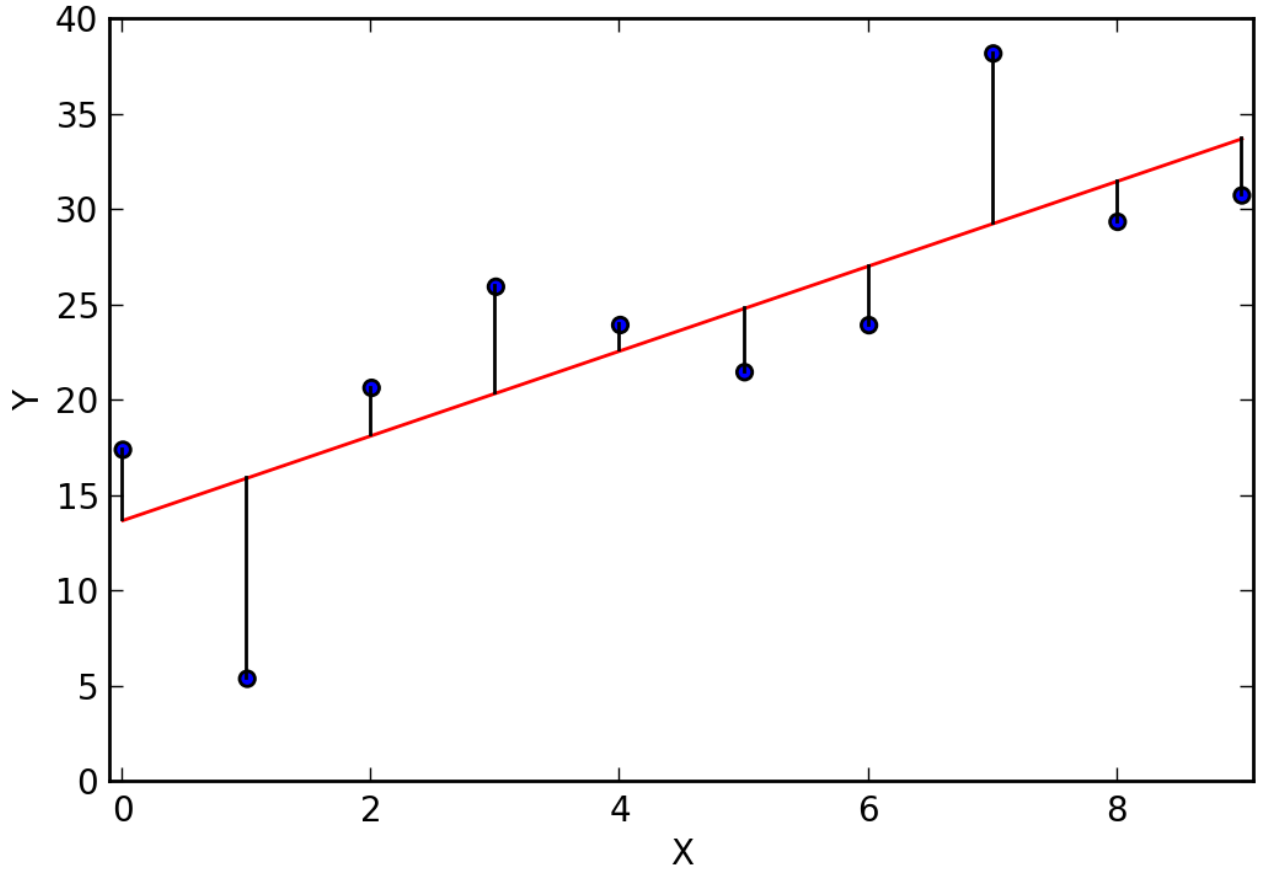
Regression

- MAE, MSE, RMSE

Classification

- Accuracy, Precision/Recall, ROC

Regression error



Evaluation metrics for Regression

We should consider the residuals: difference between values predicted and actual value:

Mean Absolute Error (MAE)

$$MAE = \frac{\sum_{i=1}^{m}(y_i^* - y_i)}{m}$$

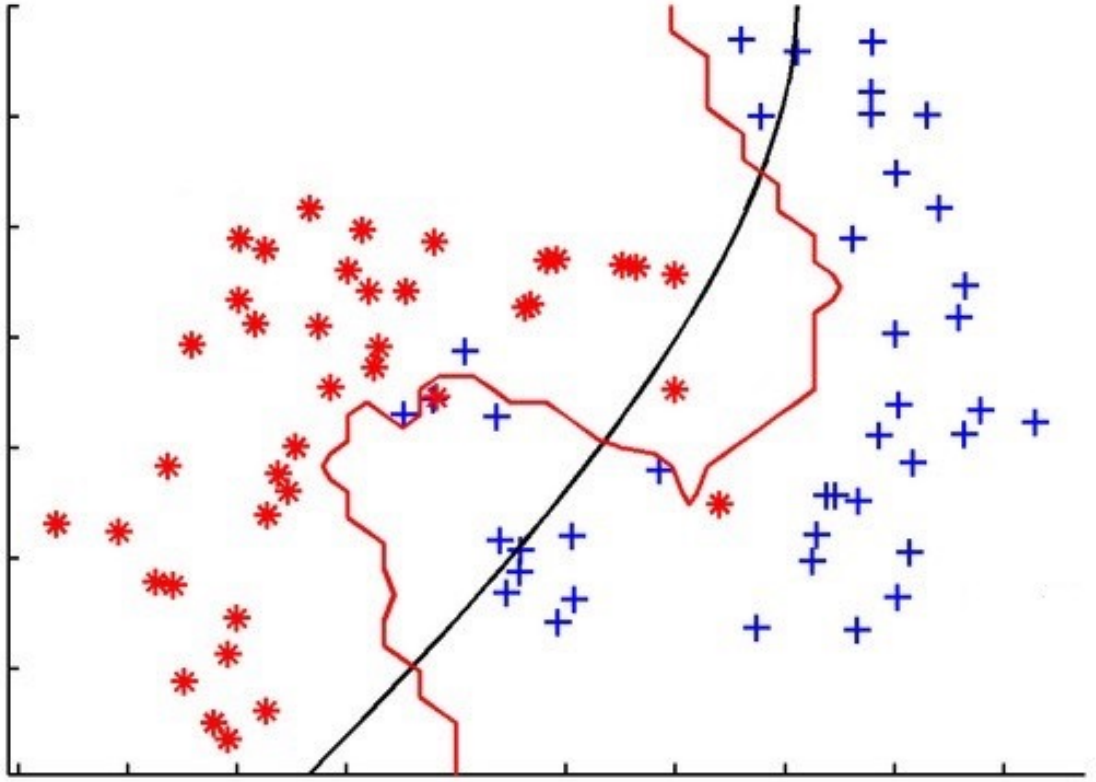
Mean Squared Error (MSE)

$$MSE = \frac{\sum_{i=1}^{m}(y_i^* - y_i)^2}{m}$$

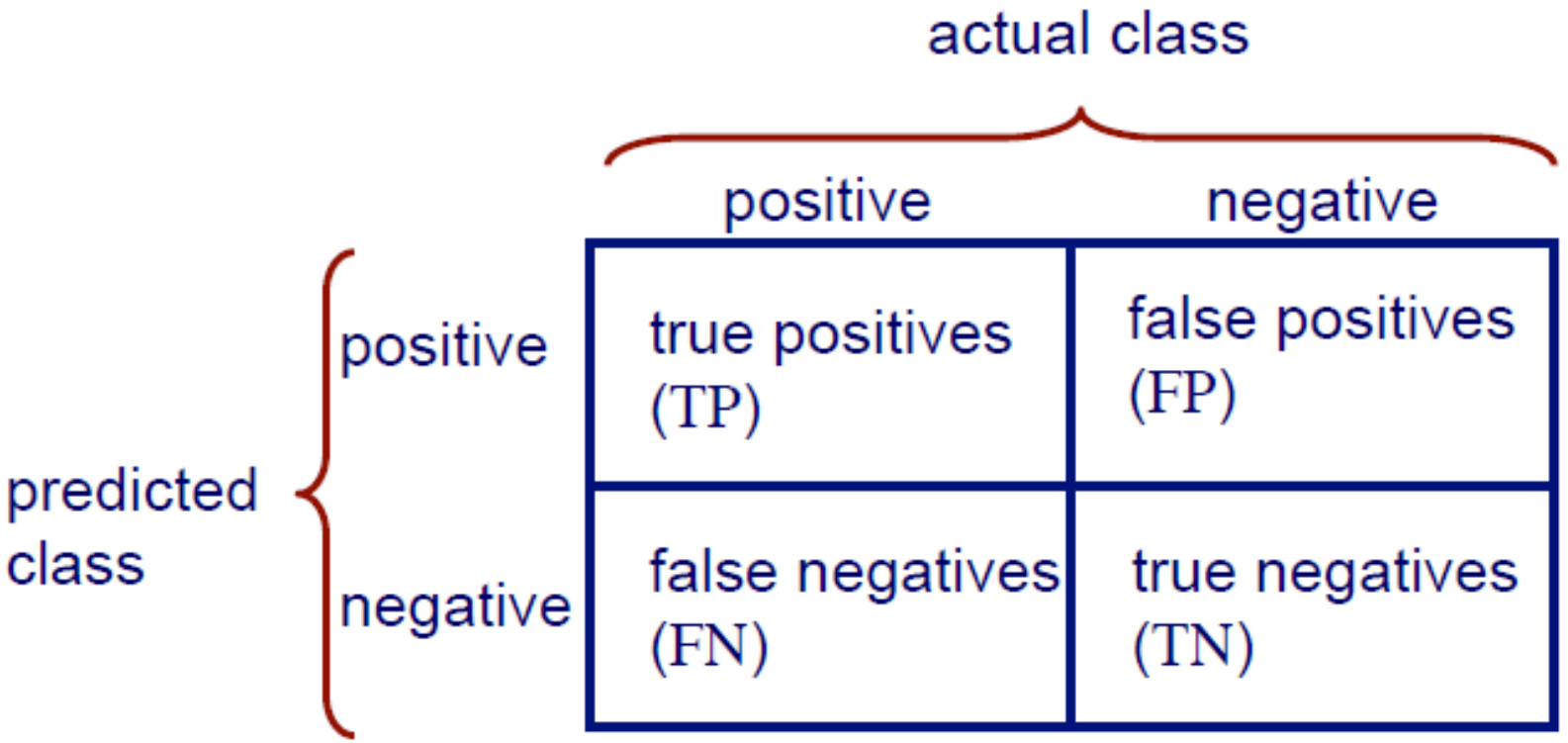
Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{m}(y_i^* - y_i)^2}{m}}$$

Classification error



Confusion matrix



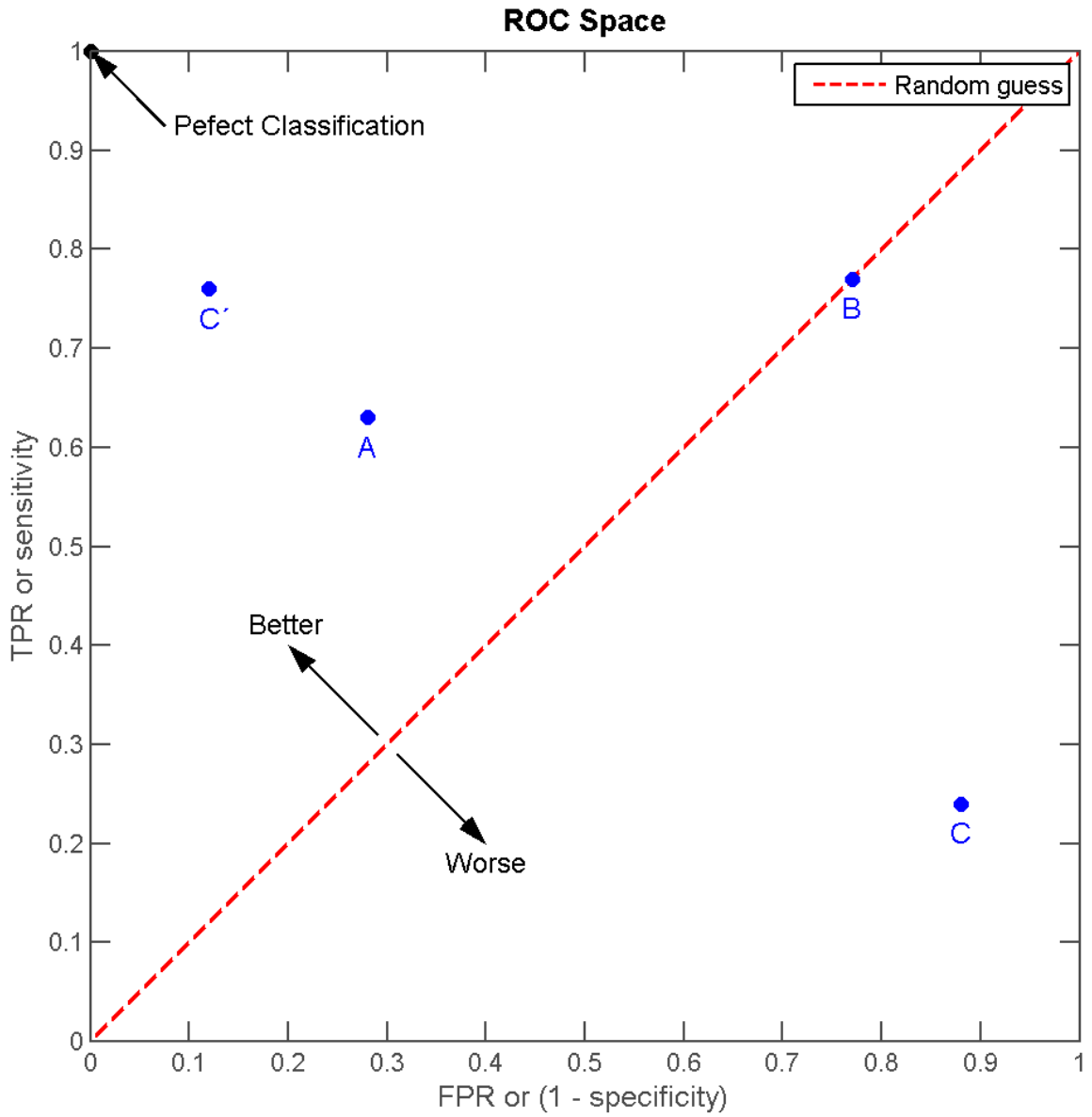
Confusion matrix

Accuracy	(TP+TN)/(TP+TN+FP+FN)
Precision	TP/(TP+FP)
Recall or Sensitivity or True Positive Rate	TP/(TP+FN)
Specificity or True Negative Rate	TN/(FP+TN)
Error rate = 1- accuracy	(FP+FN)/(TP+TN+FP+FN)
F-measure	2*precision*recall/(precision+recall)
False Positive Rate=1-specificity	FP/(TN+FP)

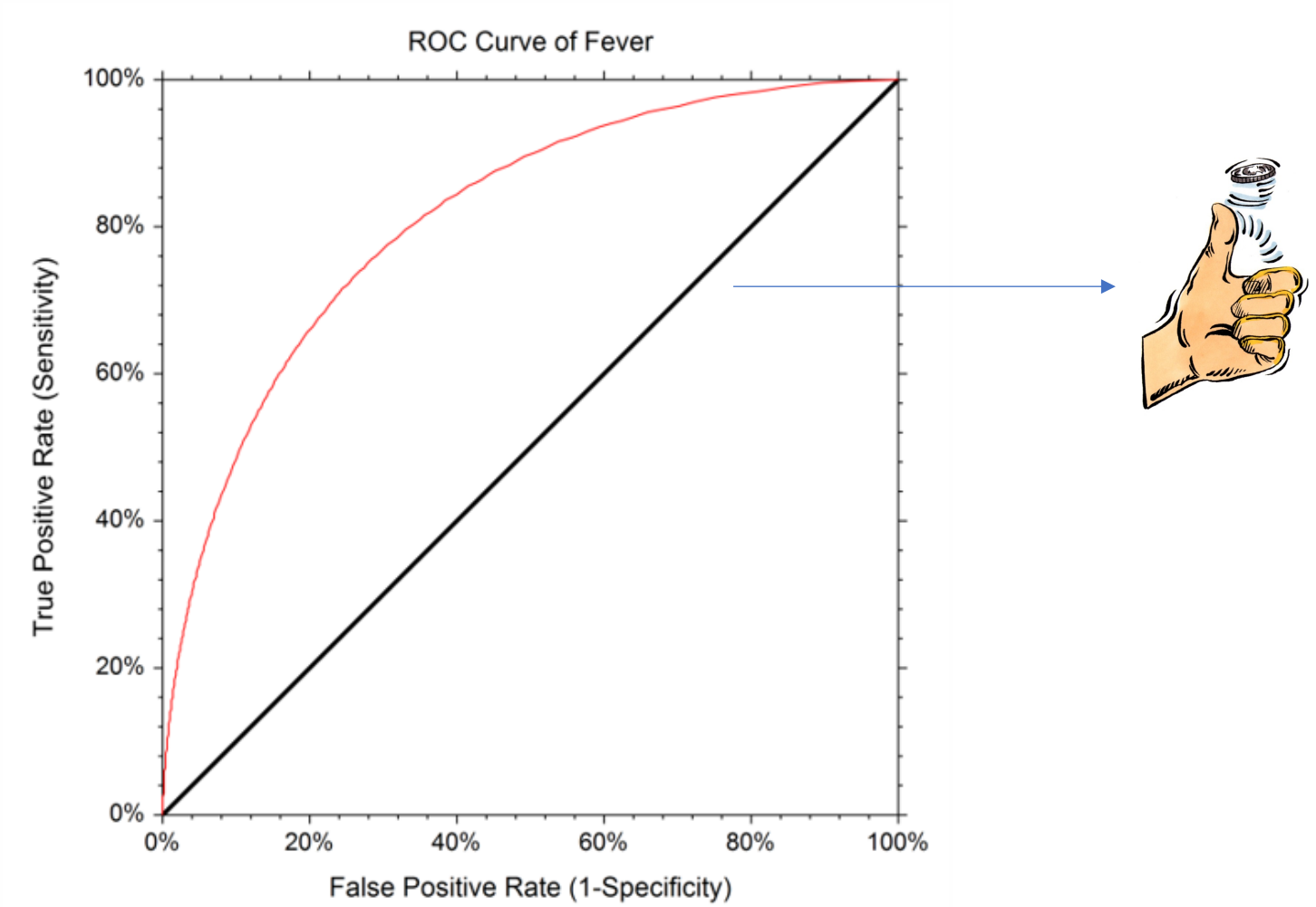
Receiver Operating Characteristic (ROC) Space

A			B		
TP=63	FP=28	91	TP=77	FP=77	154
FN=37	TN=72	109	FN=23	TN=23	46
100	100	200	100	100	200
TPR = 0.63			TPR = 0.77		
FPR = 0.28			FPR = 0.77		
PPV = 0.69			PPV = 0.50		
F1 = 0.66			F1 = 0.61		
ACC = 0.68			ACC = 0.50		

C			C′		
TP=24	FP=88	112	TP=76	FP=12	88
FN=76	TN=12	88	FN=24	TN=88	112
100	100	200	100	100	200
TPR = 0.24			TPR = 0.76		
FPR = 0.88			FPR = 0.12		
PPV = 0.21			PPV = 0.86		
F1 = 0.23			F1 = 0.81		
ACC = 0.18			ACC = 0.82		



Receiver Operating Characteristic (ROC) curve

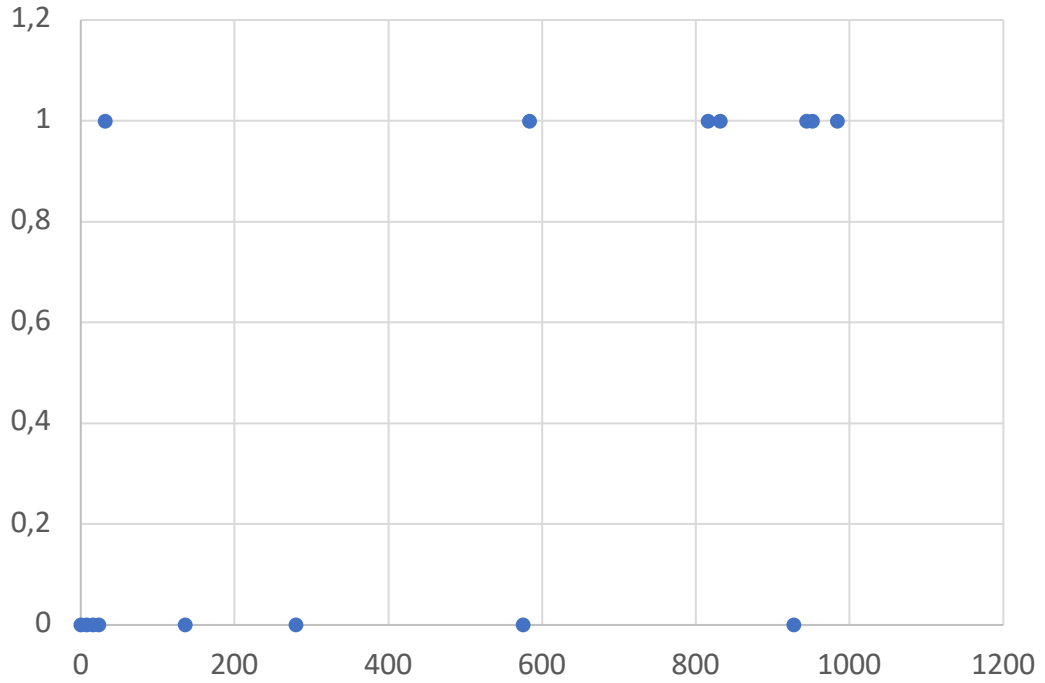


ROC curve example

Year	Observed event (1) or non-event(0)	Forecast Probability (FP)
1994	1	0.984
1995	1	0.952
1984	1	0.944
1981	0	0.928
1985	1	0.832
1986	1	0.816
1988	1	0.584
1982	0	0.576
1991	0	0.28
1987	0	0.136
1989	1	0.032
1992	0	0.024
1990	0	0.016
1983	0	0.008
1993	0	0

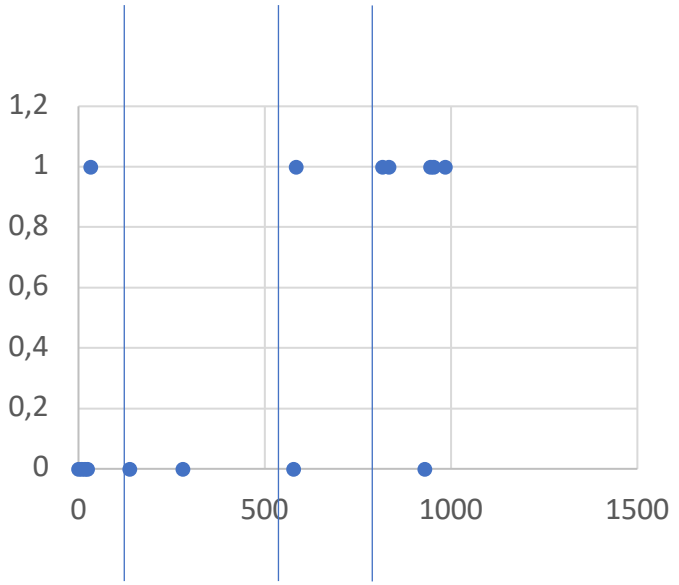
Data describes March-May precipitation over North-East Brazil for 1981-1995

Arranged in decreasing probability



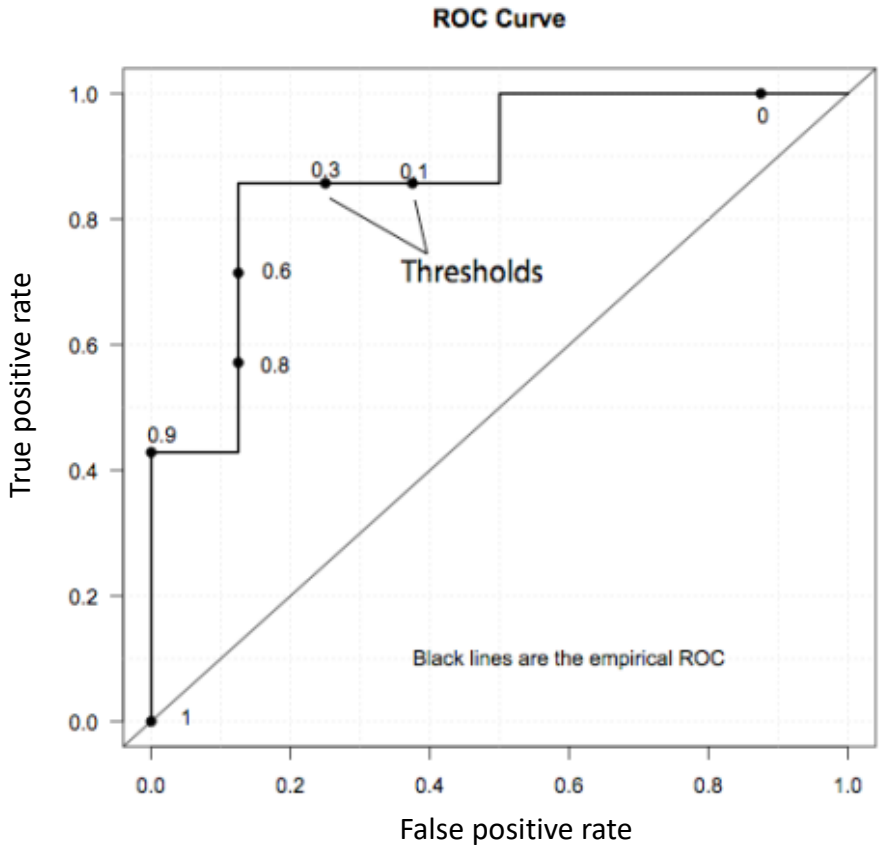
ROC curve example (cont.d)

	Year	Observed event (1) or non-event(0)	Forecast Probability	T=0.1	T=0.5	T=0.8
Correct positive	1994	1	0.984	1	1	1
	1995	1	0.952	1	1	1
	1984	1	0.944	1	1	1
	1981	0	0.928	1	1	1
	1985	1	0.832	1	1	1
	1986	1	0.816	1	1	1
	1988	1	0.584	1	1	0
	1982	0	0.576	1	1	0
	1991	0	0.28	1	0	0
	1987	0	0.136	1	0	0
	1989	1	0.032	0	0	0
Correct negative	1992	0	0.024	0	0	0
	1990	0	0.016	0	0	0
	1983	0	0.008	0	0	0
	1993	0	0	0	0	0



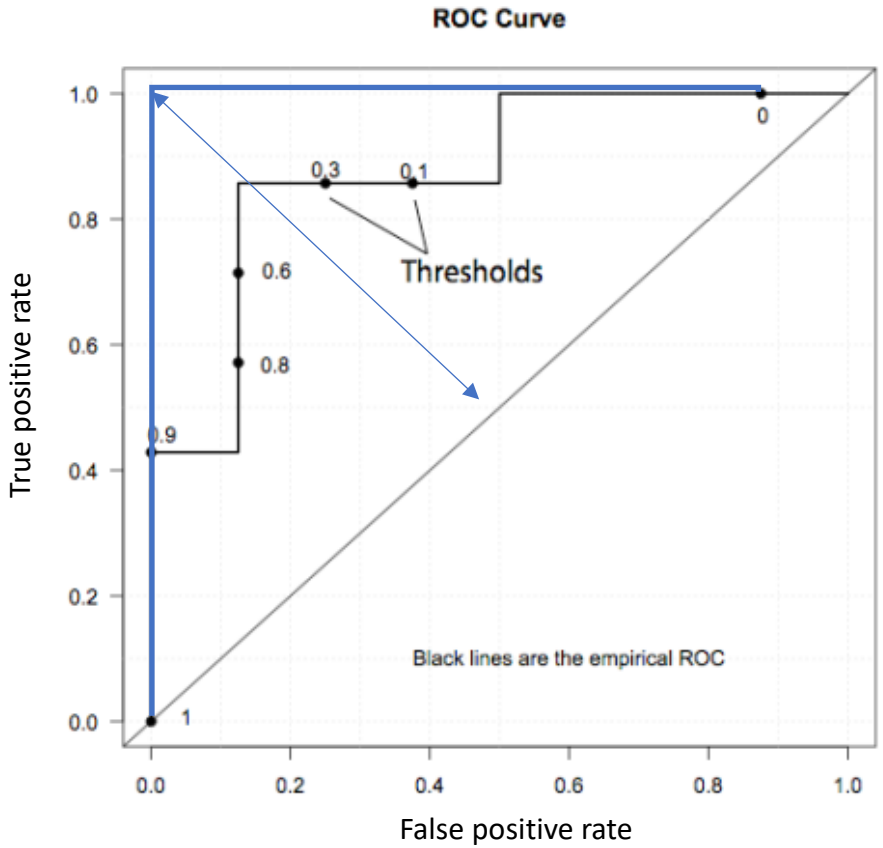
ROC curve example (cont.d)

We can plot the points of the curve for each sample:



ROC curve example (cont.d)

Optimal ROC curve



Politecnico di Bari

How to build a ML System

How to measure if the results are significant

Comparing systems using a paired t test

- The paired sample *t*-test is a statistical procedure used to determine whether the mean difference between two sets of observations is zero.
- In a paired sample *t*-test, each subject or entity is measured twice, resulting in *pairs* of observations.

- In our case, pairs of observations are, e.g. accuracy values computed by two systems varying the threshold

$$\boldsymbol{y}^a = \{y_1^a, y_2^a, \dots, y_n^a\} \qquad \boldsymbol{y}^b = \{y_1^b, y_2^b, \dots, y_n^b\}$$

Comparing systems using a paired t test (cont'd)

Null Hypothesis: the 2 learning systems have the same accuracy

Alternative Hypothesis: one of the systems is more accurate than the other

Under the null hypothesis, all observable differences are explained by random variation.

Hypothesis Test:

- use paired t-test to determine the probability p that the mean difference supports the null hypothesis
- if p is sufficiently small (typically < 0.05) then reject the null hypothesis

Comparing systems using a paired t test (cont.d)

$\vec{\boldsymbol{y}}^a = \{y_1^a, y_2^a, \ldots, y_n^a\}$ $\quad$ $\vec{\boldsymbol{y}}^b = \{y_1^b, y_2^b, \ldots, y_n^b\}$ $\quad$ $\vec{\boldsymbol{\delta}} = \{y_1^a - y_1^b, y_2^a - y_2^b, \ldots, y_n^a - y_n^b\}$

1. calculate the sample mean

$$\overline{\delta} = \frac{1}{n}\sum_{i=1}^{n}\delta_i$$

2. calculate the t statistic

$$t = \frac{\overline{\delta}}{\sqrt{\frac{1}{n(n-1)}\sum_{i=1}^{n}(\delta_i - \overline{\delta})^2}}$$

3. determine the corresponding p-value, by looking up t in a table of values for the Student's t-distribution with n-1 degrees of freedom

Comparing systems using a paired t test (cont.d)

$p = 2 \cdot \Pr(T > |t|)$ Two-tailed

$p = \Pr(T > t)$ upper-tailed

$p = \Pr(T < t)$ lower-tailed

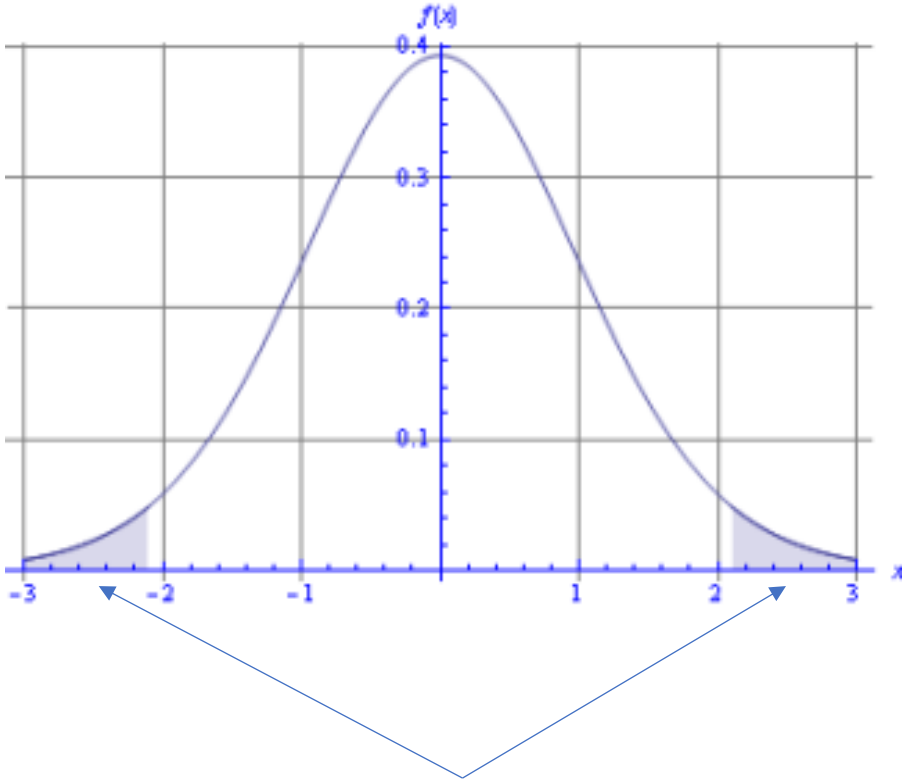
Comparing systems using a paired t test (cont.d)

- A two-tailed test asks the question: is the accuracy of the two systems different?

- A one-tailed test asks the question: is system A better than system B?

You have to choose the right one for your case

Comparing systems using a paired t test (cont.d)



for a two-tailed test, the *p*-value represents the probability mass in these two regions

The null distribution of our *t* statistic looks like this

The *p*-value indicates how far out in a tail our *t* statistic is

If the *p*-value is sufficiently small, we reject the null hypothesis, since it is unlikely we'd get such a *t* by chance

Coefficient of determination - R^2

R^2 is a measure of the proportion of the variance in the dependent variable that is **predictable** from the independent variable.

In other words, it says how much of the total deviation can be **explained** with the regression.

$$
\begin{aligned}
Dev(T) &= \sum_{i=1}^{m}(y^{(i)} - \bar{y})^2 && \text{Total Deviation} \\
Dev(R) &= \sum_{i=1}^{m}(y^{(i)*} - \bar{y})^2 && \text{Regression Deviation} \\
Dev(E) &= \sum_{i=1}^{m}(y^{(i)} - y^{(i)*})^2 && \text{Residual Deviation}
\end{aligned}
$$

$$
R^2 = \frac{Dev(R)}{Dev(T)} = 1 - \frac{Dev(E)}{Dev(T)} = \frac{cov(X,Y)^2}{Dev(X)Dev(Y)}
$$

Coefficient of determination - R^2 (cont'd)

R^2 values are in the range [0…1]

The closer the value to 1 the better the data fit the model

It provides a measure of how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model.