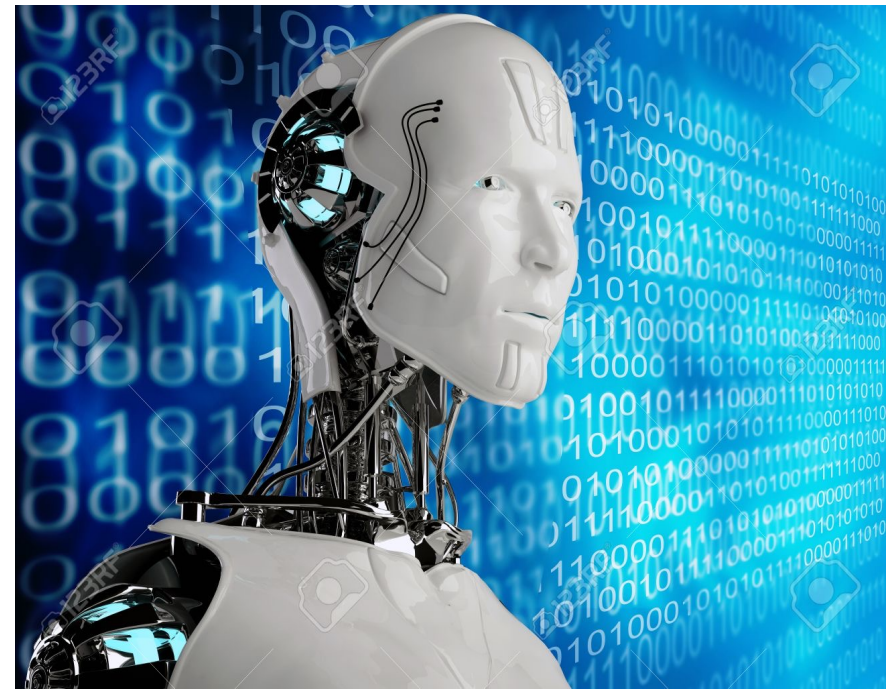


# Accuracy Metrics and Model Evaluation

Mariette Awad

Slide sources for this set of slides: Stanford Intro to ML course



# Lecture Outcomes

## Evaluation metrics

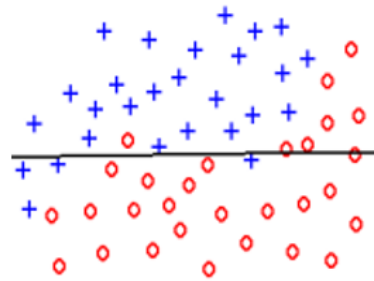
- Accuracy
- Sensitivity
- Specificity
- Precision
- Recall

## Comparing Models

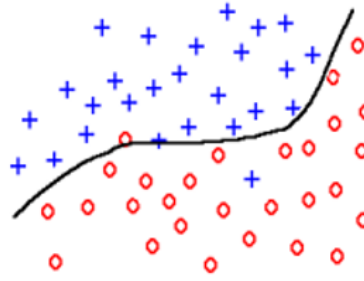
# Danger of overfitting

- Learning the training data too precisely usually leads to poor classification results on new data.

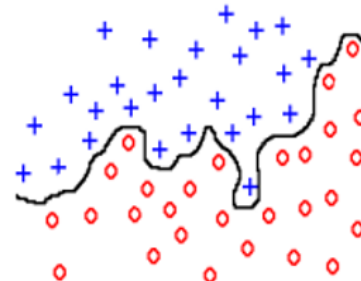
- Classifier has



underfit



fit



overfit

# Evaluation

- Also in Classification!!

Rule #1

Never evaluate on training data!

Rule #2

Never train on test data!

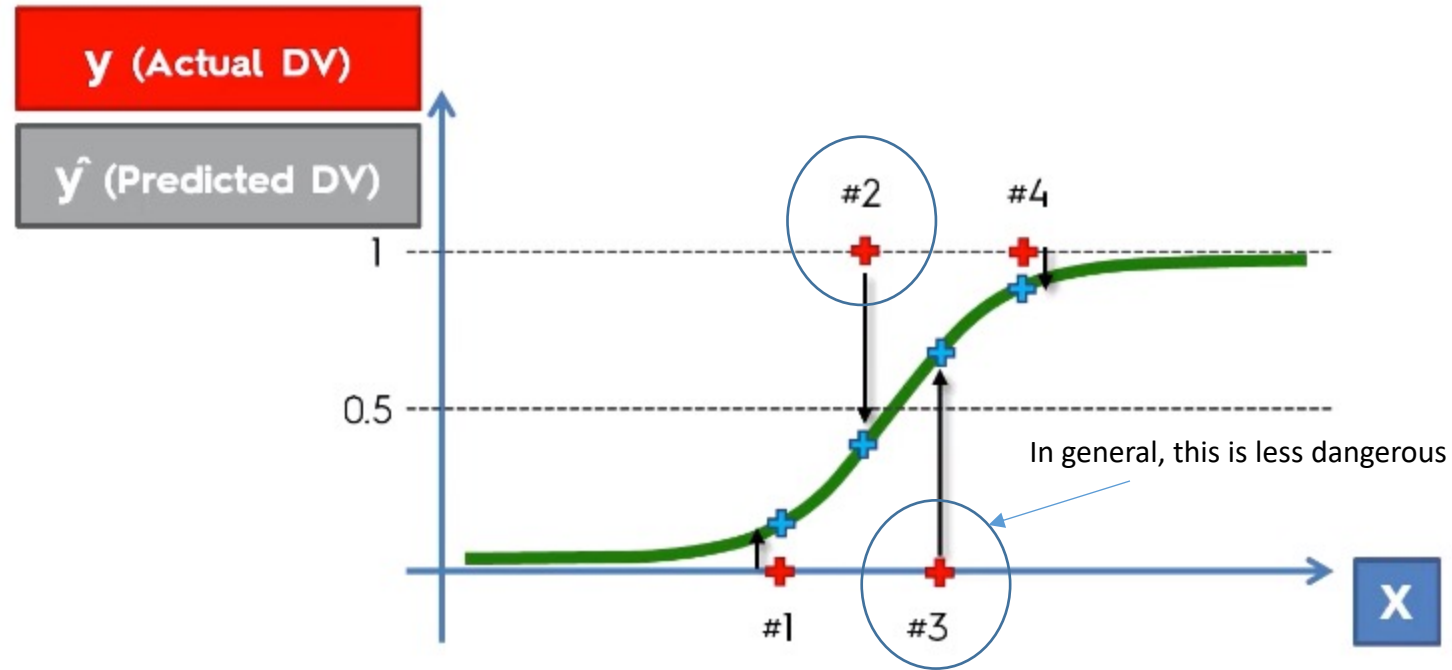
Evaluation metrics  
(Accuracy, Sensitivity, Specificity,  
Precision, Recall)



# Metrics

- Confusion or Error Matrix
- Accuracy
- Recall or Sensitivity or TPR (True Positive Rate)
- Precision
- Specificity or TNR (True Negative Rate)
- F1-Score
- Area under the receiver operating (ROC) curve (AUC)
- Logarithmic loss
- Cohen's Kappa

# Logistic Regression



© Kirill Eremenko

- #3 predicted as 1 (Positive) however actually it is 0 (negative) => False Positive (Type I error)
- #2 predicted as 0 (Negative) however actually it is 1 (positive) => False Negative (Type II error)

# Example

- A telecom company has developed a classifier for detecting whether a cell-phone network base station is faulty or not. We want to evaluate the classifier on a test set so we obtained 1000 network base stations and tried the classifier. The results obtained showed that out of the 985 NOT FAULTY stations, 965 were classified as NOT FAULTY. However, 20 stations which are actually NOT FAULTY were classified as FAULTY. The test set has 15 FAULTY stations out of which 10 were actually classified as FAULTY.

- Compute the accuracy of the classifier.

$$\begin{aligned}\text{accuracy} &= \\ &= \frac{10+965}{10+5+20+965} \\ &= 0.975\end{aligned}$$



# Evaluation

## Test Dataset

	Expected, Predicted
1	man, woman
2	man, man
3	woman, woman
4	man, man
5	woman, man
6	woman, woman
7	woman, woman
8	man, man
9	man, woman
10	woman, woman

		Actual/Expected	
		Woman	Man
Predicted	Woman	4	2
	Man	1	3

Classification accuracy is the ratio of correct predictions to total predictions made.

men classified as men: 3

women classified as women: 4

men classified as women: 2

woman classified as men: 1

accuracy = total correct predictions / total predictions made \* 100

$$\text{accuracy} = 7 / 10 * 100$$

Classification accuracy can also easily be turned into a misclassification rate or error rate by inverting the value, such as:

$$\text{error rate} = (1 - (\text{correct predictions} / \text{total predictions})) * 100$$

# Confusion Matrix

- Confusion or error Matrix is a table that describes the performance of a supervised machine learning model on the testing data, where the true values are unknown. It is called “confusion matrix” because it makes it easy to spot where your system is confusing two classes.

	Actual = Yes	Actual = No
Predicted = Yes	TP	FP
Predicted = No	FN	TN

**1.True Positives (TP):** when the actual class of the data point was 1(True) and the predicted is also 1(True)

**2.True Negatives (TN):** when the actual class of the data point was 0(False) and the predicted is also 0(False)

**3.False Positives (FP):**when the actual class of the data point was 0(False) and the predicted is 1(True).

**4.False Negatives (FN):** When the actual class of the data point was 1(True) and the predicted is 0(False).

# Confusion Matrix

- How to build a confusion matrix?
- For example let us take the case of predicting a disease.
- You have done some medical testing and with the help of the results of those tests, you are going to predict whether the person is having a disease (Y/N).
- Say, among **100** people you are predicting **20** people to have the disease.
- In actual only **15** people to have the disease and among those **15** people you have diagnosed **12** people correctly.

		Actual	
		Having Disease	Not Having Disease
Predicted	Having Disease	12	8
	Not Having Disease	3	77

# Confusion Matrix

- The matrix can come in an inverse manner so make sure you read it well!!

		$\hat{y}$ (Predicted DV)	
		0	1
$y$ (Actual DV)	0	35	5
	1	10	50

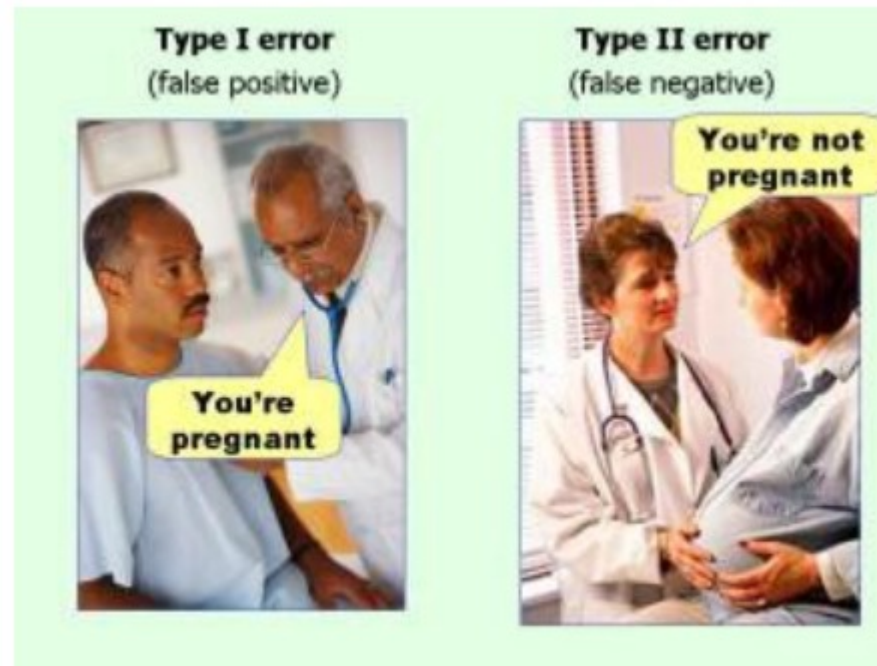
False Positive (Type I Error) points to the cell (Actual 0, Predicted 1) containing 5.

False Negative (Type II Error) points to the cell (Actual 1, Predicted 0) containing 10.

© Kirill Ereminko

# Confusion Matrix

- The ideal scenario that we all want is that the model should give 0 False Positives and 0 False Negatives.



# Example

- **1**: When a person is having cancer **0**: When a person is NOT having cancer.

	Actual - Cancer	Actual - NOT Cancer	Total
Predicted - Cancer	TP = 20	FP = 70	90
Predicted - NOT Cancer	FN = 10	TN = 200	210
Total	30	270	300

The Confusion matrix in itself is not a performance measure as such, but almost all of the performance metrics are based on Confusion Matrix and the numbers inside it.

# Example



- In the cancer detection problem example let's say that out of 100 people, only 5 people have cancer.
- We definitely want to capture all cancer cases.
- We might end up making a classification when the person actually NOT having cancer is classified as Cancerous. This might be **okay!!** Do further examination to make sure whether he has cancer or not.
- Missing a cancer patient will be a huge mistake as no further examination will be done on them. Therefore, it is better to **minimize the false negatives** in this case.

# Example





- Let's consider now an email spam detection problem and that you are expecting an important email like hearing back from a recruiter or awaiting an admit letter from a university.
- Let's assign a label to the target variable and say, **1**: "Email is a spam" and **0**: "Email is not a spam"
- Suppose the Model classifies that important email that you are desperately waiting for, as Spam (case of False positive).
- So in case of Spam email classification, **minimizing false positives** is more important than False Negatives.



# Accuracy Paradox

		$\hat{y}$ (Predicted DV)	
		0	1
$y$ (Actual DV)	0	9,700	150 
	1	50 	100



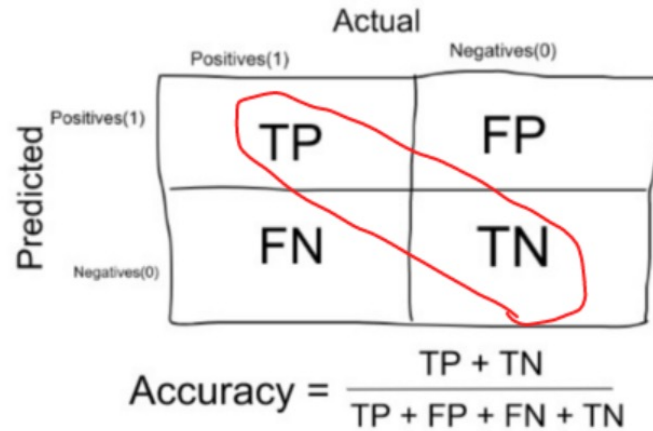
		$\hat{y}$ (Predicted DV)	
		0	1
$y$ (Actual DV)	0	9,850 	0 
	1	150 	0 

Accuracy Rate = Correct / Total  
AR = 9,800/10,000 = 98%

Accuracy Rate = Correct / Total  
AR = 9,850/10,000 = 98.5%

- But the second one is not even a model!! You are just predicting everything as 0!

# Accuracy

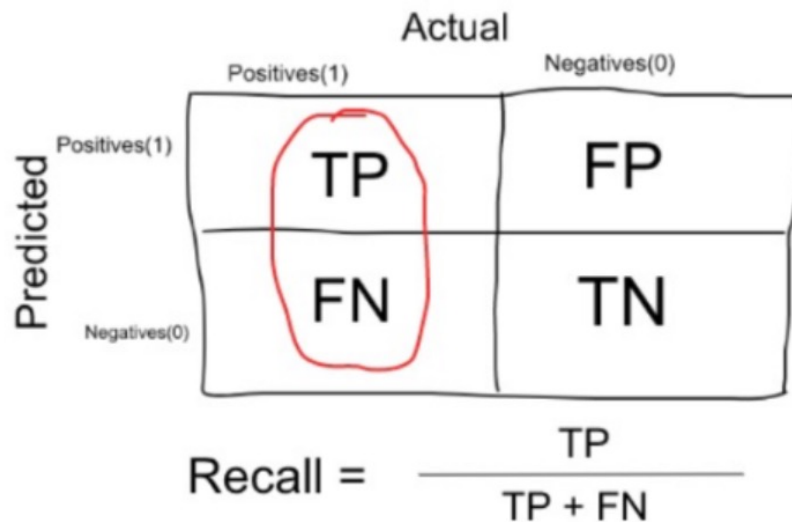


	Actual - Cancer	Actual - NOT Cancer	Total
Predicted - Cancer	TP = 20	FP = 70	90
Predicted - NOT Cancer	FN = 10	TN = 200	210
Total	30	270	300

- So, for our example:  
Accuracy =  $(20 + 200) / (20 + 10 + 70 + 200) = 220 / 300$ .
- It is the most straightforward measure of classifiers quality. It's a value between 0 and 1. **The higher, the better.**

# Recall or Sensitivity or TPR (True Positive Rate)

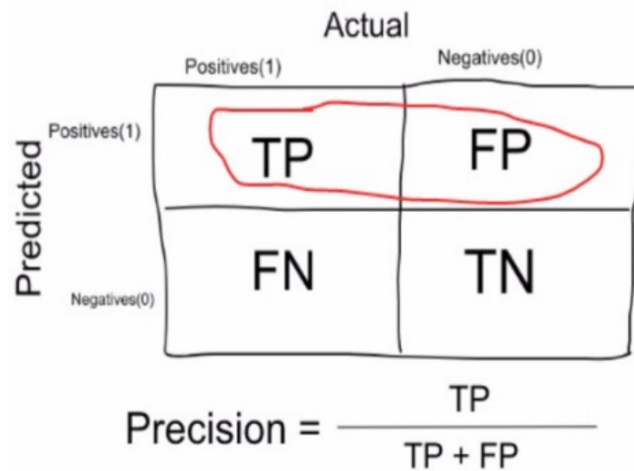
- It is the number of items correctly identified as positive out of total true positives.
- So, for our example:  $\text{Recall} = 20 / (20 + 10) = 20 / 30$



	Actual - Cancer	Actual - NOT Cancer	Total
Predicted - Cancer	TP = 20	FP = 70	90
Predicted - NOT Cancer	FN = 10	TN = 200	210
Total	30	270	300

# Precision

- It is the number of items correctly identified as positive out of total items identified as positive.



	Actual - Cancer	Actual - NOT Cancer	Total
Predicted - Cancer	TP = 20	FP = 70	90
Predicted - NOT Cancer	FN = 10	TN = 200	210
Total	30	270	300

- So, for our example:  $\text{Precision} = 20 / (20 + 70) = 20 / 90$

# Precision VS Recall

- It is clear that recall gives us information about a classifier's performance with respect to false negatives (how many did we miss), while precision gives us information about its performance with respect to false positives (how many did we caught).
- **Precision** is about being precise. So even if we managed to capture only one cancer case, and we captured it correctly, then we are 100% precise. (minimizing False positives)
- **Recall** is not so much about capturing cases correctly but more about capturing all cases that have "cancer" with the answer as "cancer". So if we simply always say every case as "cancer", we have 100% recall. (minimizing False Negatives)

# Specificity or TNR (True Negative Rate)

- It is the number of items correctly identified as negative out of total negatives.

		Actual	
		Positives(1)	Negatives(0)
Predicted	Positives(1)	TP	FP
	Negatives(0)	FN	TN

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

	Actual - Cancer	Actual - NOT Cancer	Total
Predicted - Cancer	TP = 20	FP = 70	90
Predicted - NOT Cancer	FN = 10	TN = 200	210
Total	30	270	300

- So, for our example:  $\text{Specificity} = 70 / (200 + 70) = 70 / 270$ .
- Specificity is the exact opposite of Recall.**

# Example

- **Example:** In the cancer detection example let's consider that contains 100 people, only 5 people have cancer. We predict all patient to have cancer:
- So,  $TP=5$ ,  $FP=95$  and  $TN=FN=0$
- Precision =  $5/(5+95)=5\%$   
Recall =  $5/(5+0)=100\%$   
Specificity =  $0/(0+5) = 0\%$
-

# F1 score

- It is always better to establish as a data scientist **a single-number evaluation metric for your team to optimize.**

- Suppose your algorithms perform as follows:

Classifier	Precision	Recall
A	%95	%90
B	%98	%85

- neither classifier is obviously superior, so it doesn't immediately guide you toward picking one.
- $F1 \text{ Score} = 2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall})$



# F1 score

- Suppose we have 100 credit card transactions, of which 97 are legit and 3 are fraud and let's say we came up a model that predicts everything as fraud.

		Actual	
		Fraud	Not Fraud
Predicted	Fraud	3	97
	Not Fraud	0	0

$$\text{Precision} = \frac{3}{100} = 3\%$$

$$\text{Recall} = \frac{3}{3} = 100\%$$

Note: Why not take the arithmetic mean of P and R??

if we simply take arithmetic mean of both, then it comes out to be nearly 51%.

We shouldn't be giving such a moderate score to a terrible model since it's just predicting every transaction as fraud.

- $\text{F1 Score} = 2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall}) = 2 * 3 * 100 / 103 = 5\%$

# Metrics

- Precision — Out of all the examples that predicted as positive, how many are really positive?
- Recall — Out of all the positive examples, how many are predicted as positive?
- Specificity — Out of all the people that do not have the disease, how many got negative results?
- Sensitivity — Out of all the people that have the disease, how many got positive test results?

# Classifier Accuracy Measures

	$C_1$	$C_2$
$C_1$	True positive	False negative
$C_2$	False positive	True negative

True Classes\Predicted	buy_computer = yes	buy_computer = no	total	recognition(%)
buy_computer = yes	6954	46	7000	99.34 (sensitivity/recall)
buy_computer = no	412	2588	3000	86.27 (specificity)
total	7366	2634	10000	95.42 (accuracy)

precision

- Given  $m$  classes,  $CM_{i,j}$ , an entry in a **confusion matrix**, indicates # of tuples in class  $i$  that are labeled by the classifier as class  $j$
  - Baseline Accuracy** = Majority Class / Total
  - Accuracy of a classifier M,  $\text{acc}(M)$** : percentage of test set tuples that are correctly classified by the model M
  - Error rate (misclassification rate)** of M =  $1 - \text{acc}(M)$
  - Alternative accuracy measures (e.g., for cancer diagnosis, or text, or yield)
- sensitivity** = t-pos/actual pos /\* same as **recall** - true positive recognition rate \*/
- specificity** = t-neg/actual neg /\* true negative recognition rate \*/
- accuracy** = sensitivity \* pos/(pos + neg) + specificity \* neg/(pos + neg)

# More Classifier Metrics...

## Precision and Recall, and F-measures

- **Precision: exactness** – what % of tuples that the classifier labeled as positive are actually positive

$$precision = \frac{TP}{TP + FP}$$

- **Recall: completeness** – what % of positive tuples did the classifier label as positive?

- Perfect score is 1.0

- Inverse relationship between precision & recall

$$recall = \frac{TP}{TP + FN}$$

- **F measure ( $F_1$  or F-score)**: harmonic mean of precision and recall,
- **$F_\beta$** : weighted measure of precision and recall
  - assigns  $\beta$  times as much weight to recall as to precision
  - The beta parameter determines the weight of recall in the combined score. beta < 1 lends more weight to precision, while beta > 1 favors recall (beta - > 0 considers only precision, beta -> +inf only recall).

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

$$F_\beta = \frac{(1 + \beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$$

# Classifier Evaluation Metrics: Example

Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	<b>90</b>	<b>210</b>	300	30.00 <i>(sensitivity)</i>
cancer = no	<b>140</b>	<b>9560</b>	9700	98.56 <i>(specificity)</i>
Total	230	9770	10000	96.40 <i>(accuracy)</i>

$$\text{Precision} = 90/230 = 39.13\%$$

$$\text{Recall} = 90/300 = 30.00\%$$

# Performance Metrics for Numerical Prediction

- ***MSE or RMSE***

- Calculation: *Root of [Mean of Square of the Error]*
- Interpretation: How far from zero.

- **$R^2$**  
$$R^2 = 1 - \frac{(\text{Sum of residual squared})}{(\text{Variance of observed response})} = 1 - \frac{\sum_{i=1}^{|D|} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{|D|} (y_i - y_{mean})^2}$$

- Interpretation: How well the model fits the data. Always between 0 and 1: 1 is very good fit; 0 is poor fit.
- For polynomials, adjusted  $R^2$ : 
$$\text{Adjusted } R^2 = 1 - \frac{(\text{Sum of residual squared})}{(\text{Variance of observed response})} * \frac{(n-1)}{(n-d-1)}$$
- Where n: Number of data points, and d = degree of polynomial fit.

- **Variance-Bias Trade-off**

- Variance: Compare predictions to expected (mean) predictions
- Bias: Compare predictions to actual values

# Example: Salary data

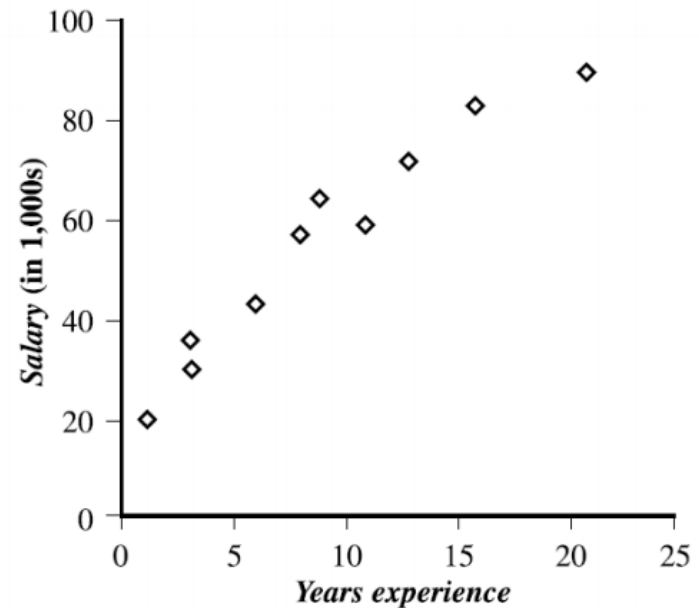
- The table shows a set of paired data where  $x$  is the number of years of work experience of a college graduate and  $y$  is the corresponding salary of the graduate.

$x$ years experience	$y$ salary (in \$1000s)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83



# Example: Salary data

- The 2-D data can be graphed on a scatter plot.
- The plot suggests a linear relationship between the two variables,  $x$  and  $y$ .



# Example: Salary data

- Given the above data, we compute

$$\bar{x} = 9.1 \text{ and } \bar{y} = 55.4$$

- we get

$$\beta_1 = \frac{(3-9.1)(30-55.4) + (8-9.1)(57-55.4) + \dots + (16-9.1)(83-55.4)}{(3-9.1)^2 + (8-9.1)^2 + \dots + (16-9.1)^2} = 3.5$$

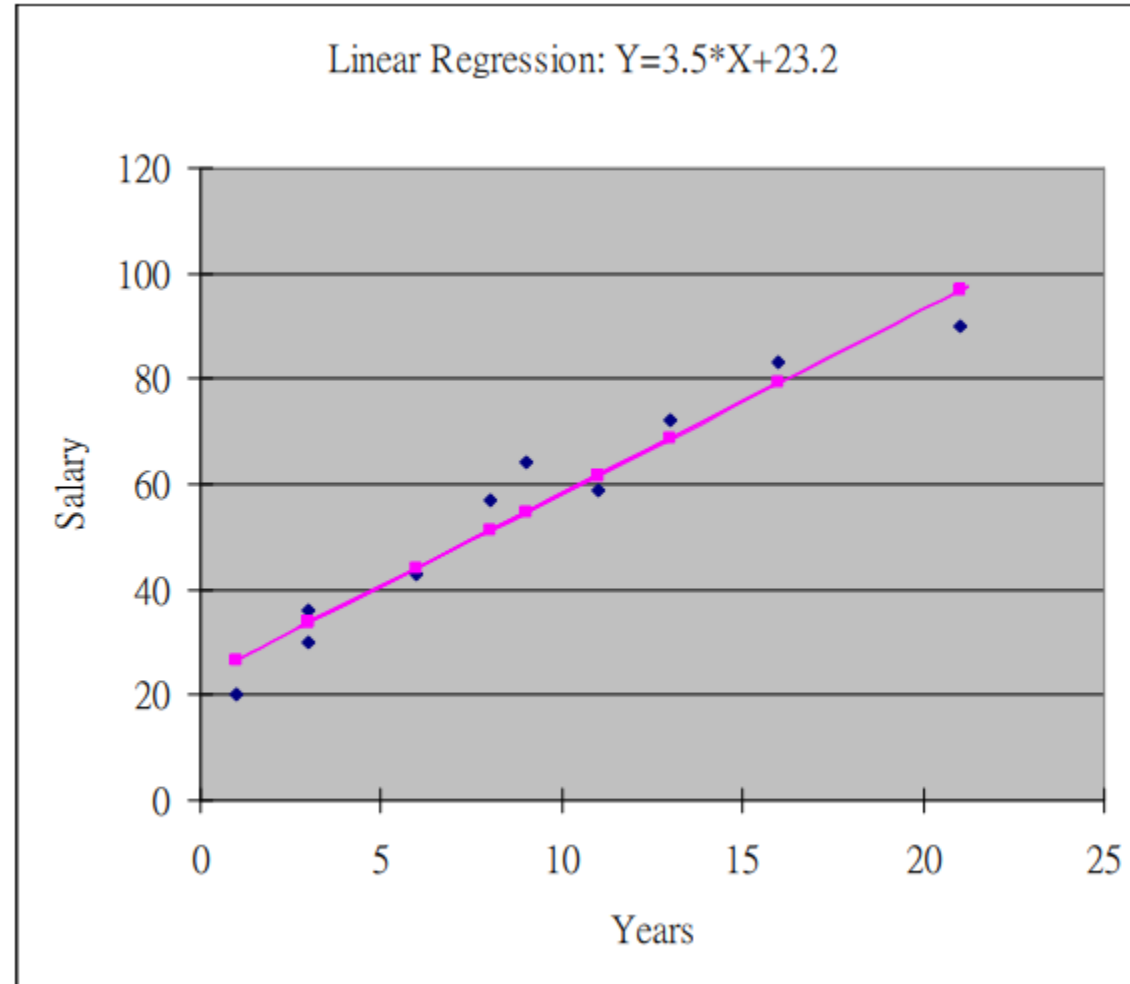
$$\beta_0 = 55.4 - (3.5)(9.1) = 23.6$$

- The equation of the least squares line is estimated by

$$y = 23.6 + 3.5x$$

<i>x</i> years experience	<i>y</i> salary (in \$1000s)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83

# Example: Salary data



# How Good is the Fit?

$$y = 23.6 + 3.5x$$

$$\text{RSS} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = 261.75$$

<i>x</i> years experience	<i>y</i> salary (in \$1000s)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83

$$\begin{aligned} &= (34.1 - 30)^2 + (51.6 - 57)^2 + (55.1 - 64)^2 + (69.1 - 72)^2 + (34.1 - 36)^2 + (44.6 - 43)^2 \\ &\quad + (62.1 - 59)^2 + (97.1 - 90)^2 + (27.1 - 20)^2 + (79.6 - 83)^2 \end{aligned}$$

# How Good is the Fit?

Back to Advertising Data  
Performance Metrics

$$\text{RSS} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = 2102.5$$

If this number is very small, then the model fits the data very well.

But how small is small?

This number, RSS, is difficult to interpret by itself.

# Error & Loss Functions

- In order to quantify how well a model performs, we define a ***loss*** or ***error function***.
- A common loss function for quantitative outcomes is the ***Mean Squared Error (MSE)***:

$$MSE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- The quantity  $|y_i - \hat{y}_i|$  is called a ***residual*** and measures the error at the  $i$ -th prediction.

# Error & Loss Functions

- MAE = Mean Absolute Error

The diagram illustrates the Mean Absolute Error (MAE) formula with the following components and annotations:

- $MAE$ : The overall loss function.
- $=$ : The equals sign.
- $\frac{1}{n}$ : A blue box containing the fraction. An annotation "Divide by the total number of data points" points to it.
- $\Sigma$ : The summation symbol.
- $|$ : The absolute value bars.
- $y$ : The actual output value, enclosed in a green box. An annotation "Actual output value" points to it.
- $-$ : The minus sign.
- $\hat{y}$ : The predicted output value, enclosed in an orange box. An annotation "Predicted output value" points to it.
- $|$ : The closing absolute value bar.
- $\underbrace{\hspace{10em}}$ : A bracket underneath the  $y - \hat{y}$  term. An annotation "The absolute value of the residual" points to it.
- $\text{Sum of}$ : An annotation pointing to the summation symbol  $\Sigma$ .

$$MAE = \frac{1}{n} \sum |y - \hat{y}|$$

- Mean Absolute Error (MAE) is obtained by calculating the absolute difference between the model predictions and the true (actual) values MAE is a measure of the average magnitude of error generated by the regression model.

# Model Evaluation

- Residual Standard Error:

$$RSE(y, \hat{y}) = \sqrt{\frac{1}{n-2} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

- For Advertising data:
- $RSE = 3.26$
- ➔ actual sales in each market deviate from true regression by approximately 3260 units on average.
- It measures the lack of fit...
- Is it a good fit?
- A more easily interpretable number is the  **$R^2$  value**.



# Model Evaluation

- Performance Metrics
- The  $R^2$  value is an alternative way to measure how good of a fit the model is to the data.
- The benefit of the  $R^2$  value is that it is a proportion (takes values between 0 and 1) so it is easier to interpret what a good value is.
- It is independent of the scale of Y.

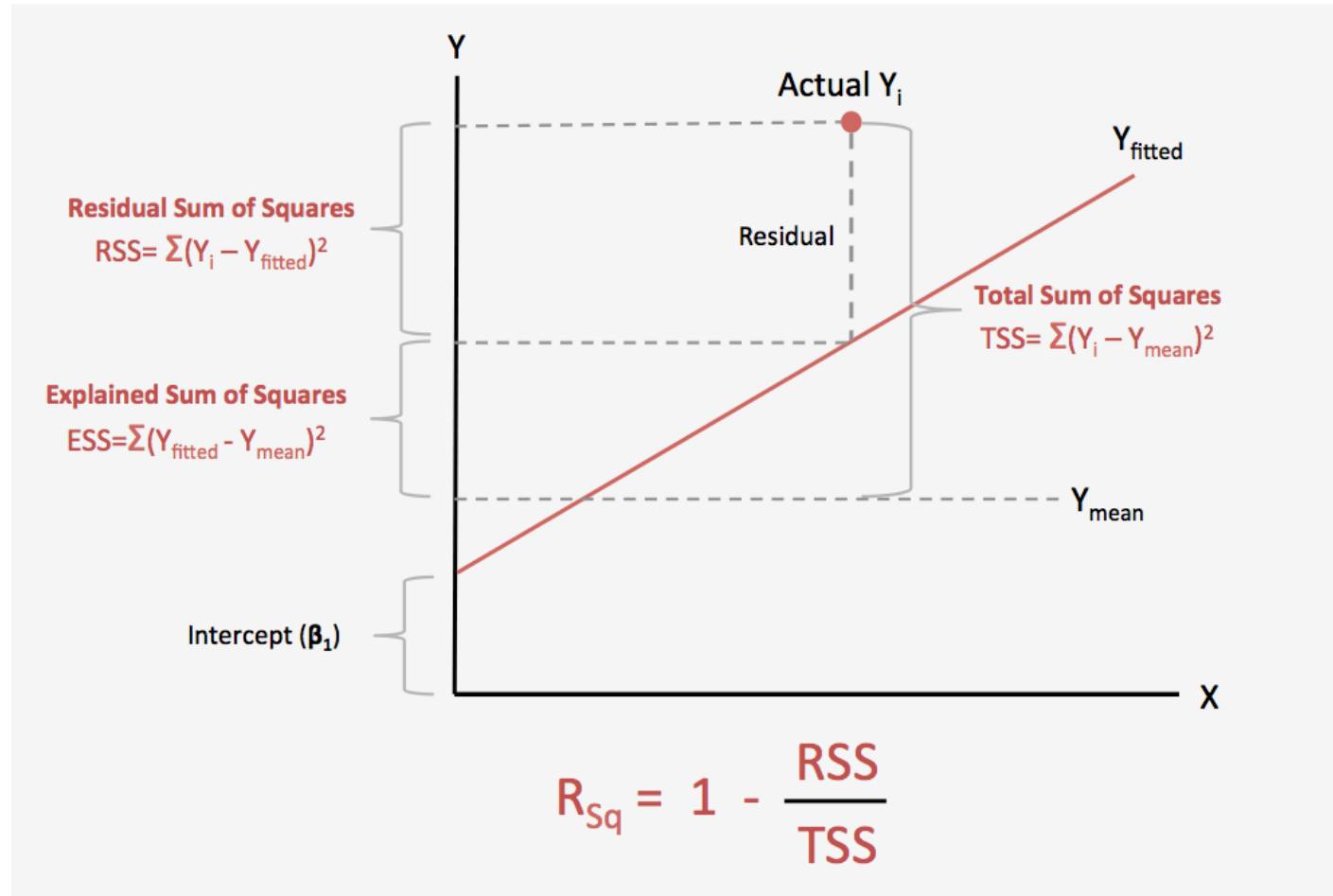
# *R-squared*

- This can be interpreted as the proportion of variance explained by our model. Note that mean squared error is in there getting divided by total error, which is the proportion of variance *unexplained* by our model and we calculate 1 minus that.

# Model Evaluation

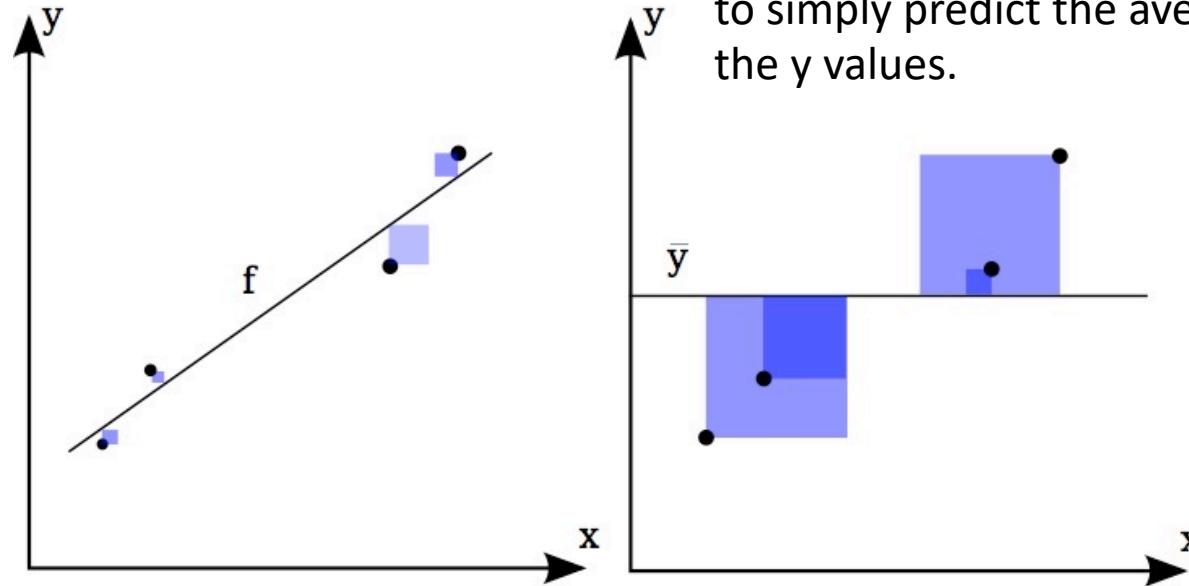
- How much (what %) of the total variation in **y** is described by the variation in **x** or the regression line?
- Total variation in y?
  - Variance  $(y_1 - \bar{y})^2 + (y_2 - \bar{y})^2 + \dots + (y_n - \bar{y})^2$
  - Denominator
  - Total Sum of Squares (TSS) = squared error from the mean  $\bar{y}$
- How much of the total variation is **not** described by the regression line?
  - Squared distances from each point to the regression line
  - Squared error of the line
  - Residue Sum of Squares (RSS)
  - $\sum_{i=1}^n (y_i - \hat{y}_i)^2 = (y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2 + \dots + (y_n - \hat{y}_n)^2$
- Percentage **not** described by the regression line is:
  - RSS/TSS
- Percentage described:
  - 1-RSS/TSS

# Mode



# Model Evaluation

Without using regression on the x variable, our most reasonable estimate would be to simply predict the average of the y values.



$R^2$  near 0?  
 $R^2$  near 1?

Average RSS →  $MSE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$

$$1 - (\text{distance\_value\_for\_best\_fit\_line} / \text{distance\_value\_for\_avg\_fit\_line})$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

Amount of variability left unexplained after performing the regression

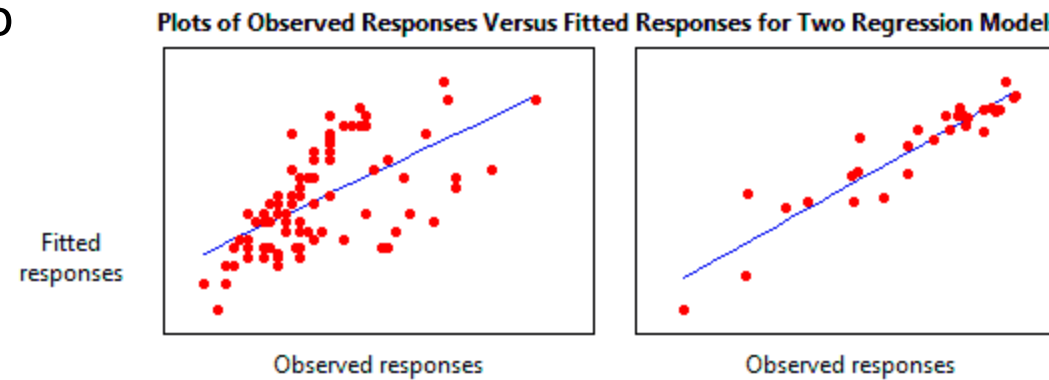
Proportion of variability in Y that can be explained using X

$$R^2 = 1 - \frac{SSE_{res}}{SSE_{tot}} = 1 - \frac{RSS}{TSS}$$

Amount of variability inherent in the response before the regression is performed

# Model Evaluation

- **R-squared or coefficient of determination**
- $R^2$ : This is the proportion of the variance explained by the model. A model is good if the value is nearly 1 (close to the data).



- Which has higher  $R^2$
- The regression model on the left accounts for 38.0% of the variance while the one on the right accounts for 87.4%. The more variance that is accounted for by the regression model the closer the data points will fall to the fitted regression line.

# $R^2$ Python

- `model = LinearRegression().fit(x, y)`
- `r_sq = model.score(x, y)`
- `print('coefficient of determination:', r_sq)`

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}|$$

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2}$$

$$R^2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2}$$

Where,

$\hat{y}$  – predicted value of  $y$

$\bar{y}$  – mean value of  $y$

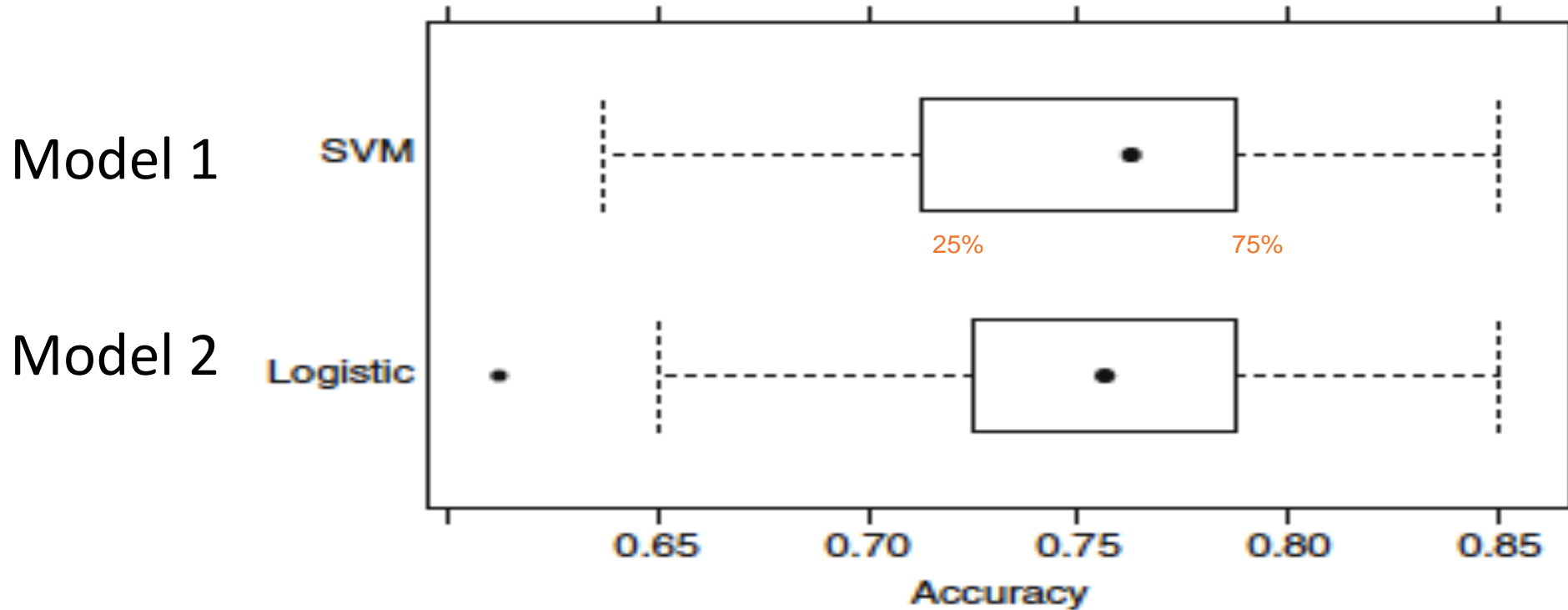


# Other considerations for Evaluation

- Speed
  - Time to construct the model (training time)
  - Time to score (classification/prediction time)
- Robustness: handling noise and missing values
- **Generalizability**
- **Interpretability**
  - **understanding and insight provided by the model**
- Scalability:
  - efficiency with disk-resident databases
  - Energy
  - Other measures, e.g., goodness of rules, such as decision tree size or compactness of classification rules

# Comparing models

# Method 1: Boxplot comparison



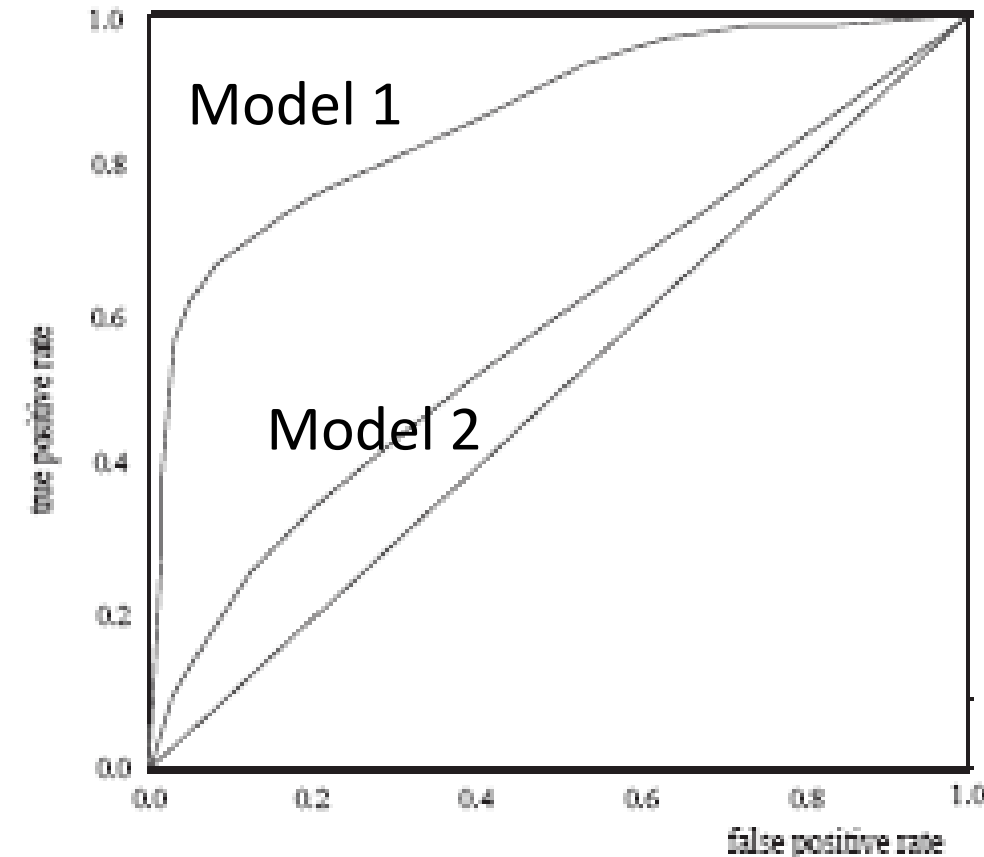
Capture the profile of tests conducted through resampling for each method, compare the summary statistics.

# Method 2: ROC Curves

- ROC (Receiver Operating Characteristics) curves: for visual comparison of classification models.
- Useful when interest on a particular class: Positive.
- Compare using TP rate (Sensitivity/recall) versus false positive rate (false alarm = 1-specificity) (1<sup>st</sup> row cell vs 2<sup>nd</sup> row cell).
- The diagonal reflects an “inaccurate” dummy classifier that is doing as good as bad: TP = FP.
- A model with perfect accuracy will have an area of 1.0

$$\text{TPR} = \text{TP}/(\text{TP}+\text{FN})$$
$$\text{FPR} = \text{FP}/(\text{FP}+\text{TN})$$

	Predicted		
	Positive	Negative	
Actual	Positive	TP = 50 FN = 10	TPR
	Negative	FP = 5 TN = 20	FPR



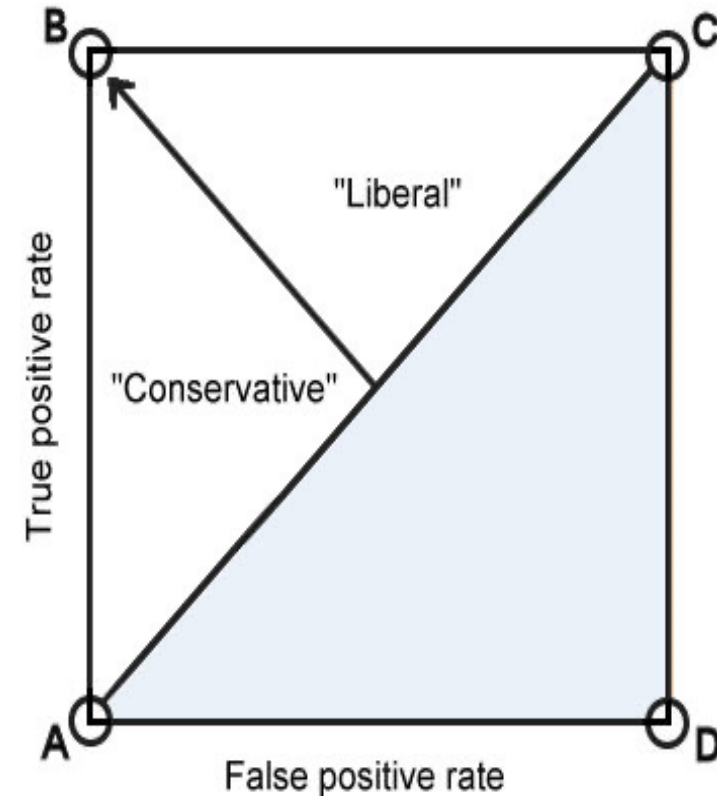
- Vertical axis represents the true positive rate
- Horizontal axis rep. the false positive rate
- The plot also shows a diagonal line

# More comments in ROC

- Originated from signal detection theory
- Shows the trade-off between the true positive rate and the false positive rate
- The area under the ROC curve is a measure of the accuracy of the model
- For different scenarios (e.g. thresholds) of the classifier, compute the True Positive Rate (TPR), and the False Positive Rate (FPR).
- Plot these points on the graph.
- Example: (choose different sets of training data) Rank the test tuples in decreasing order: the one that is most likely to belong to the positive class appears at the top of the list.
- The closer to the diagonal line (i.e., the closer the area is to 0.5), the less accurate is the model

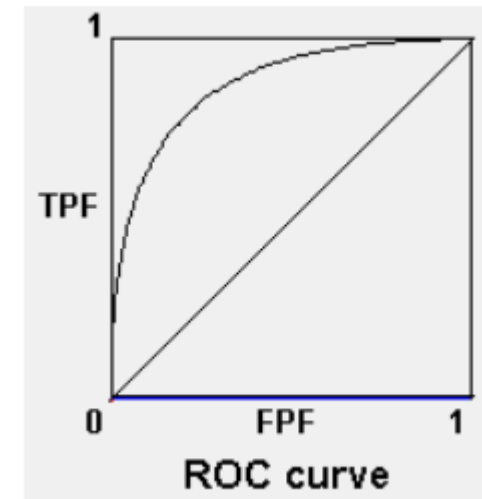
# Remarks on ROC

- A (0, 0) never identifies any case as positive, but all positive are FN.
- C (1, 1) is the converse. All cases are labelled as positives and no FN.
- B (0, 1) is the ideal classifier with perfect classification.
- D (1, 0) is a strange classifier: every case incorrectly. Inverting predictions creates a perfect classifier.
- Diagonal line from the A to C = chance performance. Any classifier in grey area below AC line is performing worst than chance.
- Any classifier whose performance is upper left triangle is doing better than chance. Better classifiers are towards the point of the arrow (top left).
- Classifiers to the left of the arrow are "conservative", i.e. they make positive classifications only with strong evidence. This produces few FP, but tends to misclassify many positive cases as negative.
- Classifiers to the right of the arrow are "liberal", i.e. they make positive classifications with weak evidence. Positive cases correct but at expense of a high FP.



# ROC

- ROC: Receiver Operating Characteristic
  - • It is a performance graphing method.
  - • A plot of True positive (TP) and false positive (FP) rates (fractions).
  - • Used for evaluating data mining schemes, and comparing the relative performance among different classifiers.
- 
- ROC space:
    - – TPR is plotted on the Y axis
    - – FPR is plotted on the X axis.
    - – depicts relative trade-offs between
      - • benefits (true positives)
      - • costs (false positives).



# ROC

- Figure shows a ROC graph with five discrete classifiers labeled A through E.
- Each discrete classifier has one (fpr, tpr) pair corresponding to a single point in ROC space.

Get everything perfect!

this perfect classifier commits  
and gets

no false positive errors  
all true positives

Never issue a positive classification!

such a classifier commits  
but also gains

no false positive errors  
no true positives.

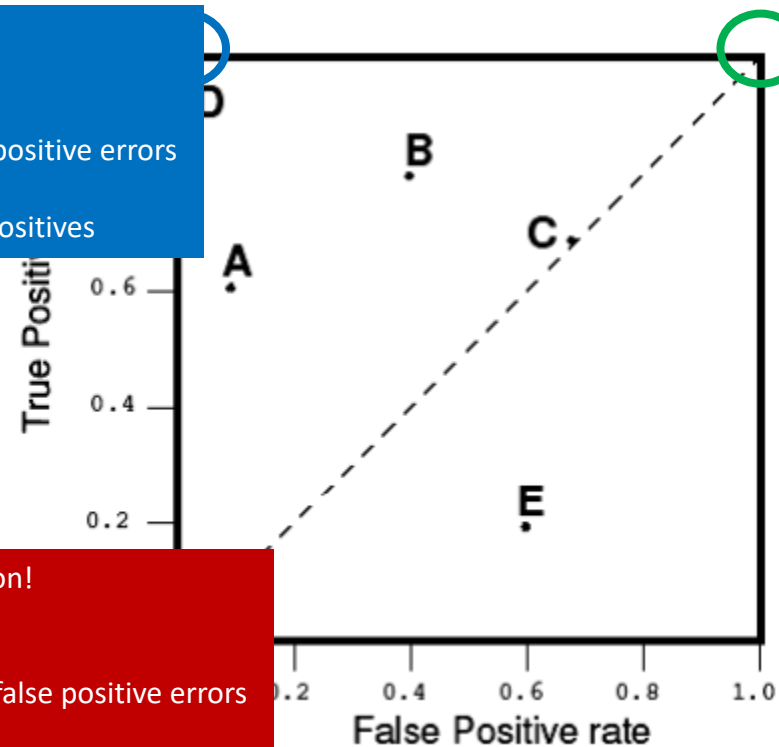
Unconditionally issue positive classification!

such a classifier predicts

all positive instances correctly

but at the cost of predicting

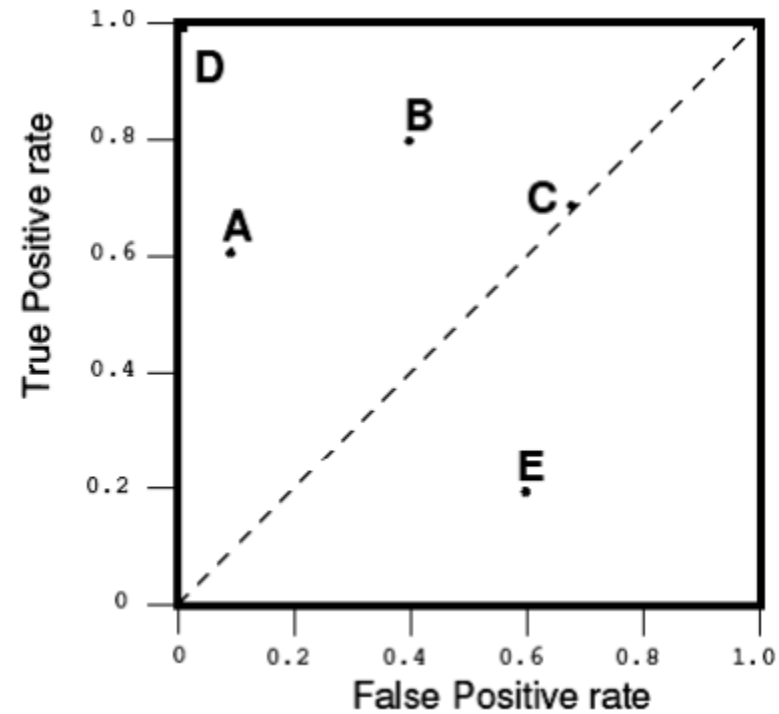
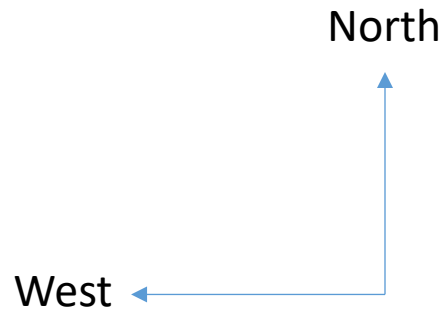
all negative instances wrongly





# ROC

- A point in ROC space is better than another if it is to the **northwest** of the other, i.e.
  - – TP rate is higher,
  - – FP rate is lower,
  - or both.



- To visualize ROC:

<http://www.navan.name/roc/>

# ROC

- Many classifiers, such as decision trees or KNN, are designed to produce only a class decision, i.e., a Y or N on each instance.
  - – When such a discrete classifier is applied to a test set, it yields a single confusion matrix, which in turn corresponds to one ROC point.
- • Some classifiers, such as a Naive Bayes classifier, yield an instance probability or score.
  - – Such a ranking or scoring classifier can be used with a threshold to produce a discrete (binary) classifier:
    - • if the classifier output is above the threshold, the classifier produces a Y,
    - • else it produces an N.
  - Each different threshold value produces a different point in ROC space (corresponding to a different confusion matrix).

# ROC Analysis

	True	
	pos	neg
Predicted		
pos	40	30
neg	60	70

Classifier 1

TPR = 0.4

FPR = 0.3

	True	
	pos	neg
Predicted		
pos	70	50
neg	30	50

Classifier 2

TPR = 0.7

FPR = 0.5

	True	
	pos	neg
Predicted		
pos	60	20
neg	40	80

Classifier 3

TPR = 0.6

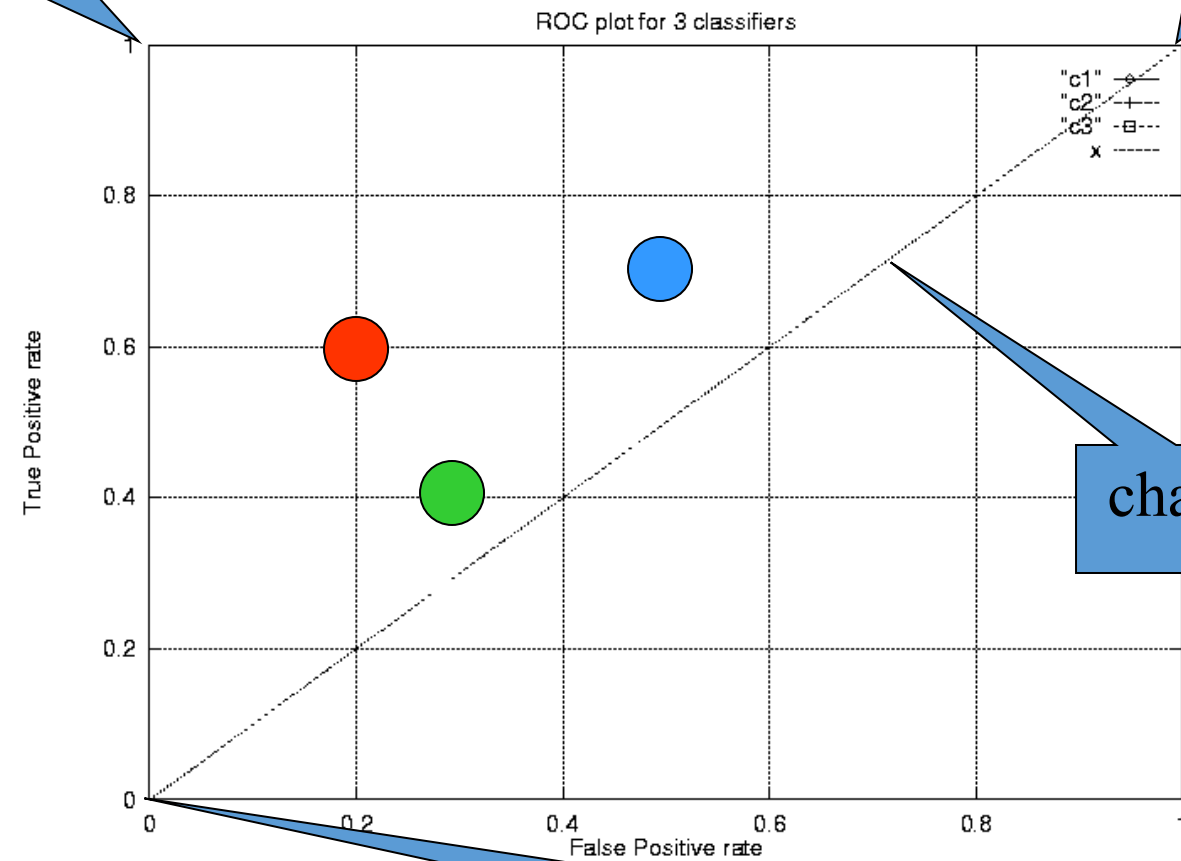
FPR = 0.2

The **false positive rate (FPR)** is the number of people who do not have the disease but are identified as having the disease (all FPs), divided by the total number of people who do not have the disease (includes all FPs and TNs).

# ROC Analysis

Ideal classifier

always positive



chance

always negative

# AUC

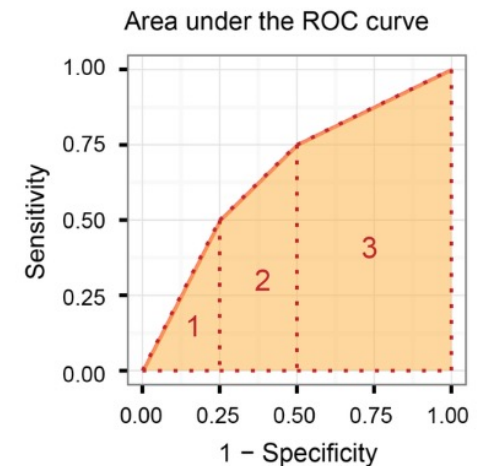
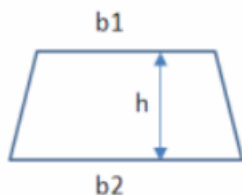
- Another advantage of using the ROC plot is a single measure called the AUC (area under the ROC curve) score.
- The AUC score can be calculated by the trapezoidal rule, which is adding up all trapezoids under the curve.
- The areas of the three trapezoids 1, 2, 3 are **0.0625**, **0.15625**, and **0.4375**. The AUC score is then **0.65625**.

The area of a trapezoid is given by the formula

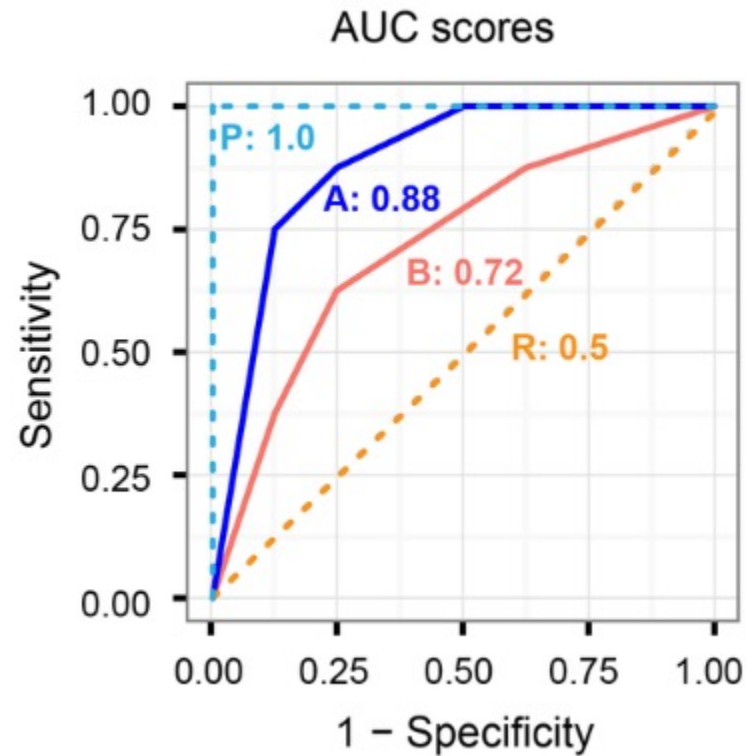
$$\text{Area} = h \cdot \left( \frac{b1 + b2}{2} \right)$$

where

$b1$ ,  $b2$  are the lengths of each base  
 $h$  is the altitude (height)



# ROC Curves for Multiple Models



## Method 3 – Statistically significant lower error

- Suppose we have 2 classifiers,  $M_1$  and  $M_2$ . Check if the difference is **statistically significant**.
- Use 10-fold cross-validation to obtain  $\overline{err}(M_1)$  and  $\overline{err}(M_2)$
- Consider the errors of each model as random variables with each having its own (mean, stdev)
- **Null Hypothesis:**  $M_1$  &  $M_2$  are the same
- Use **t-test** (or **Student's t-test**)
- If we can **reject** null hypothesis, then
  - we conclude that the difference between  $M_1$  &  $M_2$  is **statistically significant**
  - Choose model with lower error rate

# Estimating Confidence Intervals: **t-test**

If only 1 test set available: **pairwise comparison**

- For  $i^{\text{th}}$  round of 10-fold cross-validation, the same cross partitioning is used to obtain  $err(M_1)_i$  and  $err(M_2)_i$
- Average over 10 rounds to get  $\overline{err}(M_1)$  and  $\overline{err}(M_2)$
- **t-test** computes **t-statistic** with  $k-1$  degrees of freedom:

$$t = \frac{\overline{err}(M_1) - \overline{err}(M_2)}{\sqrt{var(M_1 - M_2)/k}}$$



# Paired **t-test**

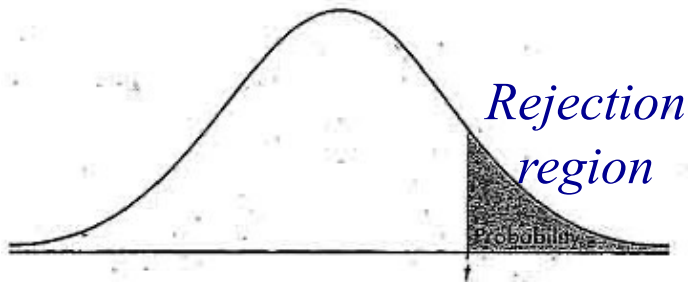
- In practice we have limited data and a limited number of estimates for computing the mean
- Student's **t-test** tells whether the means of two samples are significantly different
- In our case, the samples are cross-validation estimates for datasets from the domain
- Use a paired **t-test** because the individual samples are paired
  - Same Cross Validation is applied twice

William Gosset

Born: 1876 in Canterbury; Died: 1937 in Beaconsfield, England  
Obtained a post as a chemist in the Guinness brewery in Dublin in 1899. Invented the **t-test** to handle small samples for quality control in brewing. Wrote under the name "Student."



# Table for t-distribution



- **Significance level**, e.g.,  $\text{sig} = 0.05$  or 5% means we have **Confidence limit**,  $z = \text{value}(\text{sig}/2)$
- Symmetric, so  $-z = -\text{value}(\text{sig}/2)$

TABLE B: t-DISTRIBUTION CRITICAL VALUES

df	Tail probability $p$											
	.25	.20	.15	.10	.05	.025	.02	.01	.005	.0025	.001	.0005
1	1.000	1.376	1.963	3.078	6.314	12.71	15.89	31.82	63.66	127.3	318.3	636.6
2	.816	1.061	1.386	1.886	2.920	4.303	4.849	6.965	9.925	14.09	22.33	31.60
3	.765	.978	1.250	1.638	2.353	3.182	3.482	4.541	5.841	7.453	10.21	12.92
4	.741	.941	1.190	1.533	2.132	2.776	2.999	3.747	4.604	5.598	7.173	8.610
5	.727	.920	1.156	1.476	2.015	2.571	2.757	3.365	4.032	4.773	5.893	6.869
6	.718	.906	1.134	1.440	1.943	2.447	2.612	3.143	3.707	4.317	5.208	5.959
7	.711	.896	1.119	1.415	1.895	2.365	2.517	2.998	3.499	4.029	4.785	5.408
8	.706	.889	1.108	1.397	1.860	2.306	2.449	2.896	3.355	3.833	4.501	5.041
9	.703	.883	1.100	1.383	1.833	2.262	2.398	2.821	3.250	3.690	4.297	4.781
10	.700	.879	1.093	1.372	1.812	2.228	2.359	2.764	3.169	3.581	4.144	4.587
11	.697	.876	1.088	1.363	1.796	2.201	2.328	2.718	3.106	3.497	4.025	4.437
12	.695	.873	1.083	1.356	1.782	2.179	2.303	2.681	3.055	3.428	3.930	4.318
13	.694	.870	1.079	1.350	1.771	2.160	2.282	2.650	3.012	3.372	3.852	4.221
14	.692	.868	1.076	1.345	1.761	2.145	2.264	2.624	2.977	3.326	3.787	4.140
15	.691	.866	1.074	1.341	1.753	2.131	2.249	2.602	2.947	3.286	3.733	4.073
16	.690	.865	1.071	1.337	1.746	2.120	2.235	2.583	2.921	3.252	3.686	4.015
17	.689	.863	1.069	1.333	1.740	2.110	2.224	2.567	2.898	3.222	3.646	3.965
18	.688	.862	1.067	1.330	1.734	2.101	2.214	2.552	2.878	3.197	3.611	3.922
19	.688	.861	1.066	1.328	1.729	2.093	2.205	2.539	2.861	3.174	3.579	3.883
20	.687	.860	1.064	1.325	1.725	2.086	2.197	2.528	2.845	3.153	3.552	3.850
21	.686	.859	1.063	1.323	1.721	2.080	2.189	2.518	2.831	3.135	3.527	3.819
22	.686	.858	1.061	1.321	1.717	2.074	2.183	2.508	2.819	3.119	3.505	3.792
23	.685	.858	1.060	1.319	1.714	2.069	2.177	2.500	2.807	3.104	3.485	3.768
24	.685	.857	1.059	1.318	1.711	2.064	2.172	2.492	2.797	3.091	3.467	3.745
25	.684	.856	1.058	1.316	1.708	2.060	2.167	2.485	2.787	3.078	3.450	3.725
26	.684	.856	1.058	1.315	1.706	2.056	2.162	2.479	2.779	3.067	3.435	3.707
27	.684	.855	1.057	1.314	1.703	2.052	2.158	2.473	2.771	3.057	3.421	3.690
28	.683	.855	1.056	1.313	1.701	2.048	2.154	2.467	2.763	3.047	3.408	3.674
29	.683	.854	1.055	1.311	1.699	2.045	2.150	2.462	2.756	3.038	3.396	3.659
30	.683	.854	1.055	1.310	1.697	2.042	2.147	2.457	2.750	3.030	3.385	3.646
40	.681	.851	1.050	1.303	1.684	2.021	2.123	2.423	2.704	2.971	3.307	3.551
50	.679	.849	1.047	1.299	1.676	2.009	2.109	2.403	2.678	2.937	3.261	3.496
60	.679	.848	1.045	1.296	1.671	2.000	2.099	2.390	2.660	2.915	3.232	3.460
80	.678	.846	1.043	1.292	1.664	1.990	2.088	2.374	2.639	2.887	3.195	3.416
100	.677	.845	1.042	1.290	1.660	1.984	2.081	2.364	2.626	2.871	3.174	3.390
1000	.675	.842	1.037	1.282	1.646	1.962	2.056	2.330	2.581	2.813	3.098	3.300
$\infty$	.674	.841	1.036	1.282	1.645	1.960	2.054	2.326	2.576	2.807	3.091	3.291
	50%	60%	70%	80%	90%	95%	96%	98%	99%	99.5%	99.8%	99.9%
	Confidence level $C$											

# Statistical Significance

- Are  $M_1$  &  $M_2$  **significantly different**?
  - Compute  $t$ . Select *significance level* (e.g.  $sig = 5\%$ )
  - Consult table for t-distribution: Find  $t$  value corresponding to  $k-1$  *degrees of freedom* (here, 9 for 10-fold cross-validation)
  - t-distribution is symmetric: typically upper % points of distribution shown → look up value for **confidence limit**  $z=sig/2$  (here, 0.025)
  - **If  $t > z$  or  $t < -z$** , then  $t$  value lies in rejection region:
    - **Reject null hypothesis** that mean error rates of  $M_1$  &  $M_2$  are same
    - Conclude: statistically significant difference between  $M_1$  &  $M_2$
  - **Otherwise**, conclude that any difference is **chance**

# Recap: Performing the Test

- Fix a significance level
  - If a difference is significant at the  $\alpha\%$  level, there is a  $(100-\alpha)\%$  chance that the true means differ
- Divide the significance level by two because the test is two-tailed
- Look up the value for  $z$  that corresponds to  $\alpha/2$
- If  $t \leq -z$  or  $t \geq z$  then the difference is significant
  - i.e. the *null hypothesis* (that the difference is zero) can be rejected

# EXAMPLE

Have two prediction models,  $M_1$  and  $M_2$ . We have performed 10 rounds of 10-fold cross validation on each model, where the same data partitioning in *round*  $i$  is used for both  $M_1$  and  $M_2$ .

The error rates obtained for  $M_1$  are 30.5, 32.2, 20.7, 20.6, 31.0, 41.0, 27.7, 26.0, 21.5, 26.0.

The error rates for  $M_2$  are 22.4, 14.5, 22.4, 19.6, 20.7, 20.4, 22.1, 19.4, 16.2, 35.0.

Is one model is significantly better than the other considering a significance level of 1%?

## EXAMPLE *continued*

We hypothesis test to determine if there is a significant difference in average error. We used the same test data for each observation so we use the “paired observation” hypothesis test to compare two means:

$H_0: \bar{x}_1 - \bar{x}_2 = 0$  (Null hypothesis, difference is chance)

$H_1: \bar{x}_1 - \bar{x}_2 \neq 0$  (Statistical difference in the model errors)

Where  $\bar{x}_1$  is the mean error of model  $M_1$ , and  $\bar{x}_2$  is the mean error of model  $M_2$ .

Compute the test statistic  $t$  using the formula:

$$t = \frac{\text{(mean of the differences in error)}}{\text{(std dev of the differences in error) / sqrt (number of observations)}}$$

# EXAMPLE (the Calculations)

$$t = \frac{\text{(mean of the differences in error)}}{\text{(std dev of the differences in error) / sqrt (number of observations)}}$$

M <sub>1</sub>	M <sub>2</sub>		
30.5	22.4	8.1	(8.1 - 6.45) <sup>2</sup>
32.2	14.5	17.7	(17.7 - 6.45) <sup>2</sup>
20.7	22.4	-1.7	(-1.7 - 6.45) <sup>2</sup>
20.6	19.6	1.0	(1.0 - 6.45) <sup>2</sup>
31.0	20.7	10.3	(10.3 - 6.45) <sup>2</sup>
41.0	20.4	20.6	(20.6 - 6.45) <sup>2</sup>
27.7	22.1	5.6	(5.6 - 6.45) <sup>2</sup>
26.0	19.4	6.6	(6.6 - 6.45) <sup>2</sup>
21.5	16.2	5.3	(5.3 - 6.45) <sup>2</sup>
26.0	35.0	-9.0	(-9.0 - 6.45) <sup>2</sup>

**Average= 6.45**

*Average and take square root to get* rad of summation of all std

**Std Dev= 8.25**



# Example: Table Lookup

Significance level 1% (0.01), so look up  $t_{\text{sig}/2}$  value for probability 0.005  
9 degrees of freedom

if  $-z \leq t \leq z$ , i.e.  $-3.25 \leq 2.47 \leq 3.25$

then ~~accept~~ fail to reject null hypothesis, i.e., the two models are not different at a significance level of 0.01

significance/2

TABLE B: t-DISTRIBUTION CRITICAL VALUES

df	Tail probability p											
	.25	.20	.15	.10	.05	.025	.02	.01	.005	.0025	.001	.0005
1	1.000	1.376	1.963	3.078	6.314	12.71	15.89	31.82	63.66	127.3	318.3	636.6
2	.816	1.061	1.386	1.886	2.920	4.303	4.849	6.965	9.925	14.09	22.33	31.60
3	.765	.978	1.250	1.638	2.353	3.182	3.482	4.541	5.841	7.453	10.21	12.92
4	.741	.941	1.190	1.533	2.132	2.776	2.999	3.747	4.604	5.598	7.173	8.610
5	.727	.920	1.156	1.476	2.015	2.571	2.757	3.365	4.032	4.773	5.893	6.869
6	.718	.906	1.134	1.440	1.943	2.447	2.612	3.143	3.707	4.317	5.208	5.959
7	.711	.896	1.119	1.415	1.895	2.365	2.517	2.998	3.499	4.029	4.785	5.408
8	.706	.889	1.108	1.397	1.860	2.306	2.449	2.896	3.355	3.833	4.501	5.041
9	.703	.883	1.100	1.383	1.833	2.262	2.398	2.821	3.250	3.690	4.297	4.781
10	.700	.879	1.093	1.372	1.812	2.228	2.359	2.764	3.169	3.581	4.144	4.587
11	.697	.876	1.088	1.363	1.796	2.201	2.328	2.718	3.106	3.497	4.025	4.437
12	.695	.873	1.083	1.356	1.782	2.179	2.303	2.681	3.055	3.428	3.930	4.318
13	.694	.870	1.079	1.350	1.771	2.160	2.282	2.650	3.012	3.372	3.852	4.221
14	.692	.868	1.076	1.345	1.761	2.145	2.264	2.624	2.977	3.326	3.787	4.140
15	.691	.866	1.074	1.341	1.753	2.131	2.249	2.602	2.947	3.286	3.733	4.073
16	.690	.865	1.071	1.337	1.746	2.120	2.235	2.583	2.921	3.252	3.686	4.015
17	.689	.863	1.069	1.333	1.740	2.110	2.224	2.567	2.898	3.222	3.646	3.965
18	.688	.862	1.067	1.330	1.734	2.101	2.214	2.552	2.878	3.197	3.611	3.922
19	.688	.861	1.066	1.328	1.729	2.093	2.205	2.539	2.861	3.174	3.579	3.883
20	.687	.860	1.064	1.325	1.725	2.086	2.197	2.528	2.845	3.153	3.552	3.850

degrees of freedom



# Statistical Significance - Intervals: t-test

- If only 1 test set available: **pairwise comparison**
  - For  $i^{\text{th}}$  round of 10-fold cross-validation, the same cross partitioning is used to obtain  $err(M_1)_i$  and  $err(M_2)_i$
  - Average over 10 rounds to get  $\overline{err}(M_1)$  and  $\overline{err}(M_2)$
  - **t-test** computes **t-statistic** with  $k-1$  degrees of freedom:

$$t = \frac{\overline{err}(M_1) - \overline{err}(M_2)}{\sqrt{var(M_1 - M_2)/k}} \quad \text{where} \quad var(M_1 - M_2) = \frac{1}{k} \sum_{i=1}^k [err(M_1)_i - err(M_2)_i - (\overline{err}(M_1) - \overline{err}(M_2))]^2$$

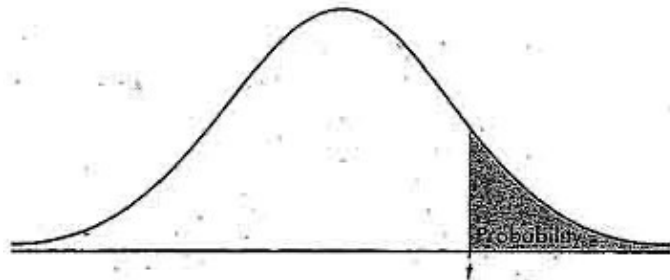
- If two test sets available: use **non-paired t-test**

where

$$var(M_1 - M_2) = \sqrt{\frac{var(M_1)}{k_1} + \frac{var(M_2)}{k_2}},$$

where  $k_1$  &  $k_2$  are # of cross-validation samples used for  $M_1$  &  $M_2$ , resp.

# Estimating Confidence Intervals: Table for t-distribution



- Symmetric
- **Significance level**, e.g.,  $sig = 0.05$  or 5% means  $M_1$  &  $M_2$  are *significantly different* for 95% of population
- **Confidence limit**,  $z = sig/2$

TABLE B: t-DISTRIBUTION CRITICAL VALUES

df	Tail probability $p$											
	.25	.20	.15	.10	.05	.025	.02	.01	.005	.0025	.001	.0005
1	1.000	1.376	1.963	3.078	6.314	12.71	15.89	31.82	63.66	127.3	318.3	636.6
2	.816	1.061	1.386	1.886	2.920	4.303	4.849	6.965	9.925	14.09	22.33	31.60
3	.765	.978	1.250	1.638	2.353	3.182	3.482	4.541	5.841	7.453	10.21	12.92
4	.741	.941	1.190	1.533	2.132	2.776	2.999	3.747	4.604	5.598	7.173	8.610
5	.727	.920	1.156	1.476	2.015	2.571	2.757	3.365	4.032	4.773	5.893	6.869
6	.718	.906	1.134	1.440	1.943	2.447	2.612	3.143	3.707	4.317	5.208	5.959
7	.711	.896	1.119	1.415	1.895	2.365	2.517	2.998	3.499	4.029	4.785	5.408
8	.706	.889	1.108	1.397	1.860	2.306	2.449	2.896	3.355	3.833	4.501	5.041
9	.703	.883	1.100	1.383	1.833	2.262	2.398	2.821	3.250	3.690	4.297	4.781
10	.700	.879	1.093	1.372	1.812	2.228	2.359	2.764	3.169	3.581	4.144	4.587
11	.697	.876	1.088	1.363	1.796	2.201	2.328	2.718	3.106	3.497	4.025	4.437
12	.695	.873	1.083	1.356	1.782	2.179	2.303	2.681	3.055	3.428	3.930	4.318
13	.694	.870	1.079	1.350	1.771	2.160	2.282	2.650	3.012	3.372	3.852	4.221
14	.692	.868	1.076	1.345	1.761	2.145	2.264	2.624	2.977	3.326	3.787	4.140
15	.691	.866	1.074	1.341	1.753	2.131	2.249	2.602	2.947	3.286	3.733	4.073
16	.690	.865	1.071	1.337	1.746	2.120	2.235	2.583	2.921	3.252	3.686	4.015
17	.689	.863	1.069	1.333	1.740	2.110	2.224	2.567	2.898	3.222	3.646	3.965
18	.688	.862	1.067	1.330	1.734	2.101	2.214	2.552	2.878	3.197	3.611	3.922
19	.688	.861	1.066	1.328	1.729	2.093	2.205	2.539	2.861	3.174	3.579	3.883
20	.687	.860	1.064	1.325	1.725	2.086	2.197	2.528	2.845	3.153	3.552	3.850
21	.686	.859	1.063	1.323	1.721	2.080	2.189	2.518	2.831	3.135	3.527	3.819
22	.686	.858	1.061	1.321	1.717	2.074	2.183	2.508	2.819	3.119	3.505	3.792
23	.685	.858	1.060	1.319	1.714	2.069	2.177	2.500	2.807	3.104	3.485	3.768
24	.685	.857	1.059	1.318	1.711	2.064	2.172	2.492	2.797	3.091	3.467	3.745
25	.684	.856	1.058	1.316	1.708	2.060	2.167	2.485	2.787	3.078	3.450	3.725
26	.684	.856	1.058	1.315	1.706	2.056	2.162	2.479	2.779	3.067	3.435	3.707
27	.684	.855	1.057	1.314	1.703	2.052	2.158	2.473	2.771	3.057	3.421	3.690
28	.683	.855	1.056	1.313	1.701	2.048	2.154	2.467	2.763	3.047	3.408	3.674
29	.683	.854	1.055	1.311	1.699	2.045	2.150	2.462	2.756	3.038	3.396	3.659
30	.683	.854	1.055	1.310	1.697	2.042	2.147	2.457	2.750	3.030	3.385	3.646
40	.681	.851	1.050	1.303	1.684	2.021	2.123	2.423	2.704	2.971	3.307	3.551
50	.679	.849	1.047	1.299	1.676	2.009	2.109	2.403	2.678	2.937	3.261	3.496
60	.679	.848	1.045	1.296	1.671	2.000	2.099	2.390	2.660	2.915	3.232	3.460
80	.678	.846	1.043	1.292	1.664	1.990	2.088	2.374	2.639	2.887	3.195	3.416
100	.677	.845	1.042	1.290	1.660	1.984	2.081	2.364	2.626	2.871	3.174	3.390
1000	.675	.842	1.037	1.282	1.646	1.962	2.056	2.330	2.581	2.813	3.098	3.300
∞	.674	.841	1.036	1.282	1.645	1.960	2.054	2.326	2.576	2.807	3.091	3.291
	50%	60%	70%	80%	90%	95%	96%	98%	99%	99.5%	99.8%	99.9%
	Confidence level $C$											