Model Evaluation

Chenghua Lin
Department of Computing Science
University of Aberdeen

Outline

- Confusion matrix
- Accuracy
- Recall, precision and F-measure
- ROC curve
- Cross validation

Confusion Matrix

- The four possible outcomes of a binary classifier are usually shown in a confusion matrix
- A number of performance metrics defined using these counts

		Predicted Class: Positive'	Predicted Class: negative'	
Actual Class	positive	TP	FN	
	negative	FP	TN	

Confusion Matrix

Actual Class	Positive'	negative'	Predicted Class
positive	TP	FN	
negative	FP	TN	

- True Positives (TP)
  - # of correct predictions that an instance is positive
- True Negatives (TN)
  - # of correct predictions that an instance is negative
- False Positives (FP)
  - # of incorrect predictions that an instance is positive
- False Negatives (FN)
  - # of incorrect of predictions that an instance negative

Accuracy

Actual Class	Positive'	negative'	Predicted Class
positive	TP	FN	
negative	FP	TN	

- **Accuracy of the positive class**: the proportions of positive class instances have been correctly predicted
  - Acc_pos = TP / (TP + FN)
- **Accuracy of the negative class**: the proportions of negative class instances have been correctly predicted
  - Acc_pos = TN / (FP + TN)
- **Overall accuracy**: the proportion of the total number of predictions that were correct
  - Acc = (TP + TN) / (TP + FP + FN + TN)

Confusion Matrix: Example1

	Spam (Predicted)	Non-Spam (Predicted)	Accuracy
Spam (Actual)	27	6	81.81
Non-Spam (Actual)	10	57	85.07
Overall Accuracy			84

The spam dataset:

- Contains 100 instances
- 33 instances are spam
- 67 instances are non-spam

Accuracy:

- Acc(spam) = 27/(27 + 6) = 81.81%
- Acc(non-spam) =57/(10 + 57) = 85.07%
- Overall_acc = (27 + 57) / (27+6+10+57) = 84%

Confusion Matrix: Example2

	Spam (Predicted)	Non-Spam (Predicted)	Accuracy
Spam (Actual)	0	10	??
Non-Spam (Actual)	0	990	??
Overall Accuracy			??

The spam dataset:

- 10 patterns are spam
- 990 pattern are non-spam

Accuracy:

- Acc(spam) = 0/10 = 0%
- Acc(non-spam) =990/990 = 100%
- Overall_acc = (0+990)/(0+10+0+990) = 99%

Issues with accuracy

- The confusion matrix tells us how the classifier is behaving for individual classes.
- Accuracy
  - Work well for (more or less) balanced dataset (e.g., 100 positive and 100 negative data instances)
  - Cannot capture true classifier performance when dataset is highly unbalanced.

Beyond accuracy…

- Recall
  - Aka True Positive rate (TP), or sensitivity
  - Recall = TP/(TP+FN)
- Precision
  - The proportion of the predicted positive instance that were correct (positive predictive value)
  - Precision = TP/(TP + FP)
- F-measure
  - Aka F1- score, is the harmonic mean of precision and recall
  - Suitable for cases where one of the classes is rare
  - F1=2x(recall x precision) / (recall + precision)

Confusion Matrix: example3

	Positive (Predicted)	Negative (Predicted)
Positive (Actual)	100	50
Negative (Actual)	150	9700

The dataset:

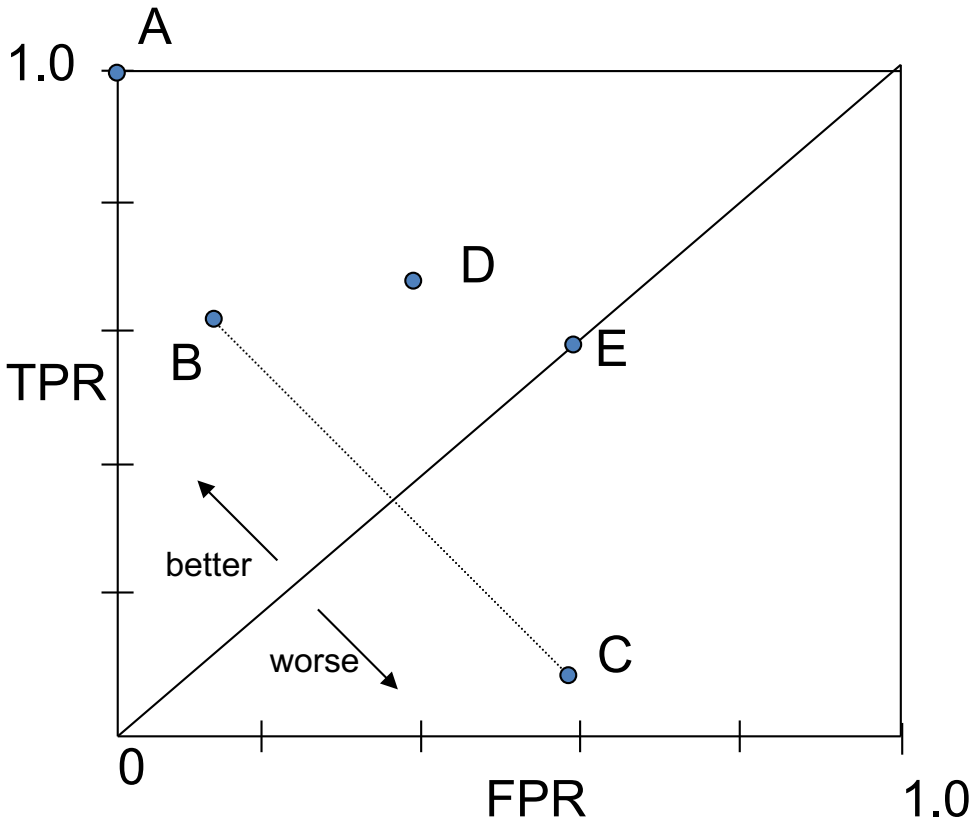
- 150 positive class instances
- 9850 negative class instances

Accuracy:

- Overall_acc = (100+9700)/(100+50+150+9700) = 0.98
- Recall = 100/(100+50) = 0.667
- Precision = 100/(100+150) = 0.4
- F1 = 2 x (0.667x0.4)/(0.667+0.4) = 0.5

ROC - Receiver Operating Characteristic

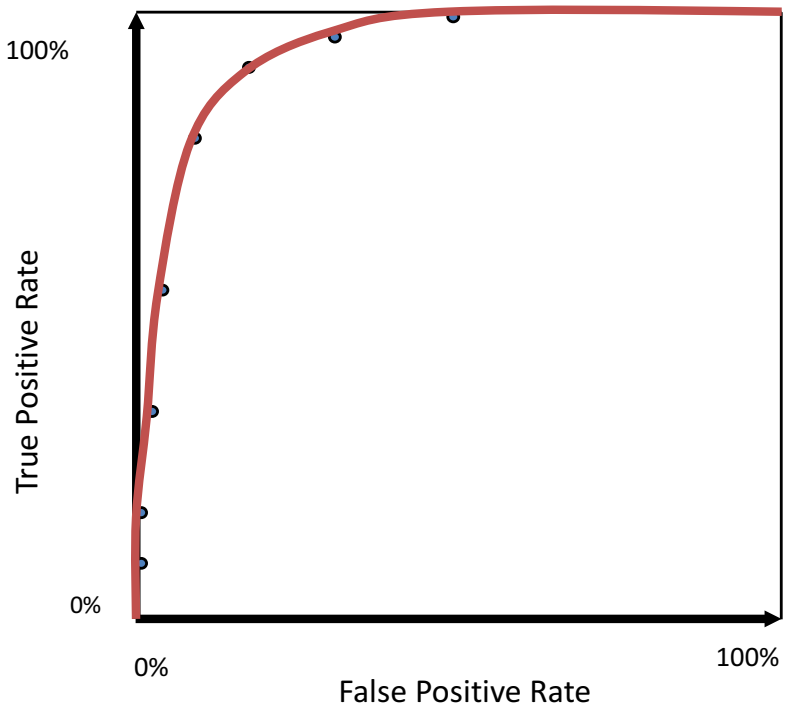
- Particularly a plot of TPR on y-axis against FPR on x axis is known as ROC
- A, B, C, D and E are five classifiers with different TPR and FPR values
- A is the ideal classifier because it has TPR = 1.0 and FPR = 0
- E is on the diagonal which stands for random guess
- C performs worse than random guess
  - But inverse of C which is B is better than D
- Classifiers should aim to be in the northwest



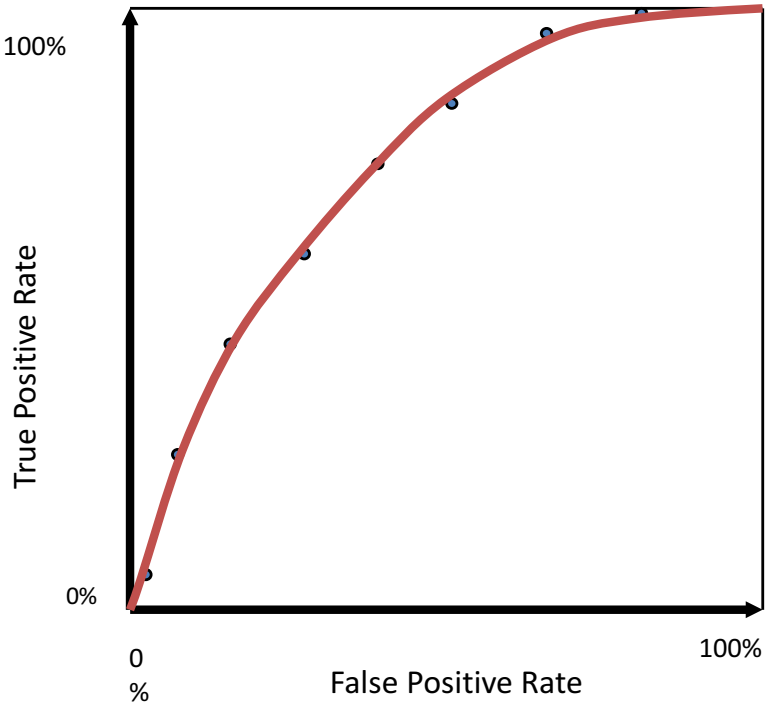
TPR = TP/(TP+FN)
FPR= FP/(FP+TN)

ROC curve comparison

A good test:



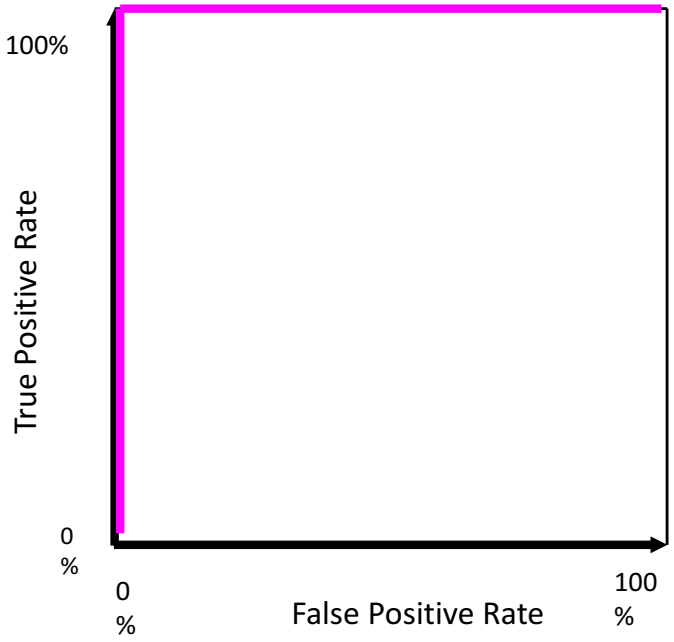
A poor test:



AUC: area under the curve

Summary

Best Test:



The distributions don't overlap at all

Testing Classifier

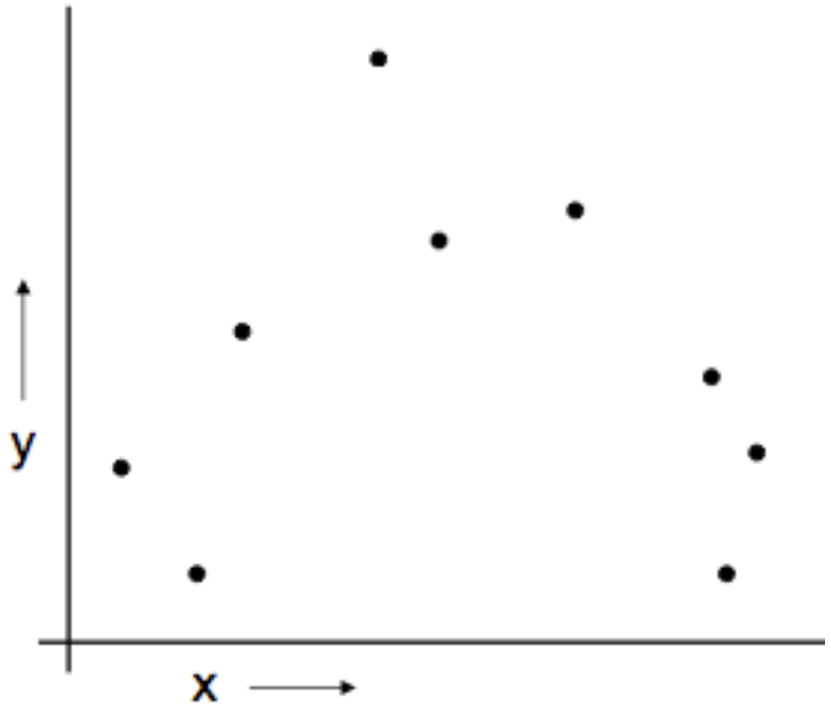
- Testing the classifier on training data is not useful
  - Performance figures from such testing will be optimistic
  - Because the classifier is trained from the very same data
- Ideally, a new data set called ‘test set’ needs to be used for testing
  - If test set is large performance figures will be more realistic
  - Creating test set needs experts’ time and therefore creating large test sets is expensive
  - After testing, test set is combined with training data to produce a new classifier
  - Sometimes, a third data set called ‘validation data’ used for fine tuning a classifier or to select a classifier among many
- In practice several strategies used to make up for lack of test data
  - Holdout procedure – a certain proportion of training data is held as test data and remaining used for training
  - Cross-validation
  - Leave-one-out cross-validation

Cross-validation

Outline

- Test-set cross-validation
- Leave-one-out cross-validation
- k-fold cross-validation

A Regression Problem



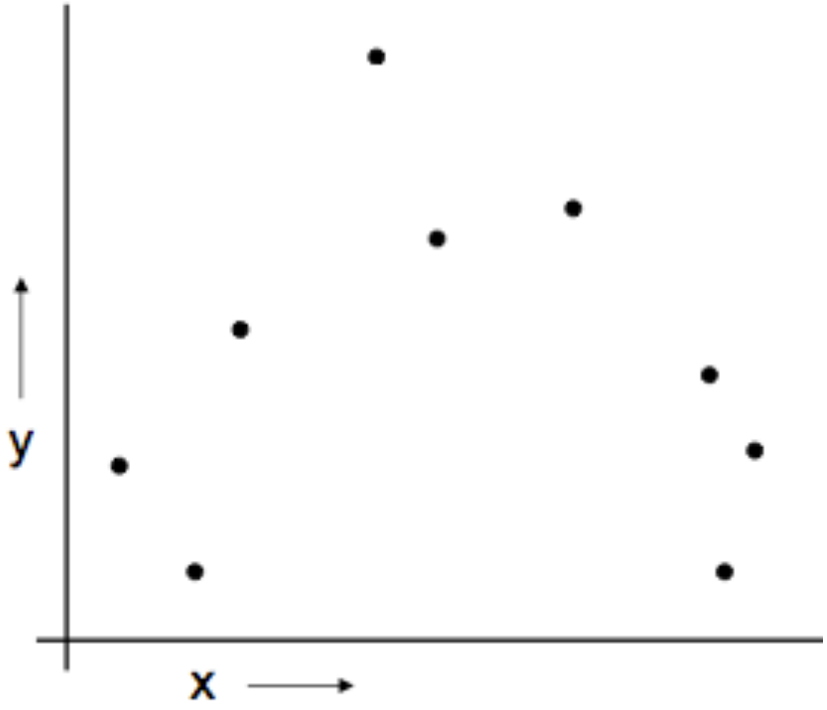
Regression

- a statistical process for estimating the relationships among variables.

Regression vs. classification

- Regression: the output variable takes continuous values.
- Classification: the output variable takes class labels.

A Regression Problem



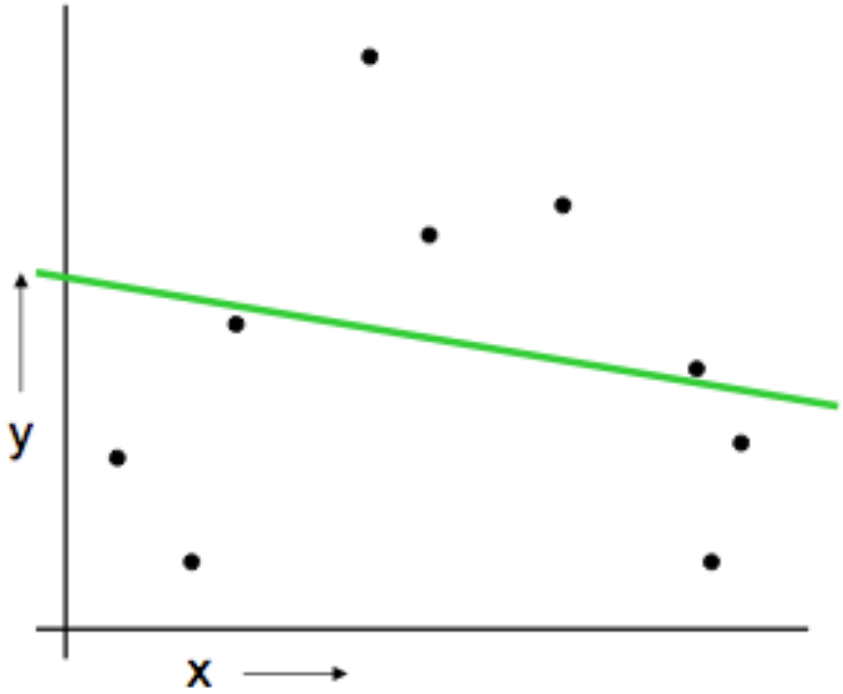
y = f(x) + noise

Can we learn **f** from this data?

Let's consider three methods

- Linear regression
- Quadratic regression
- Linear non-parametric regression

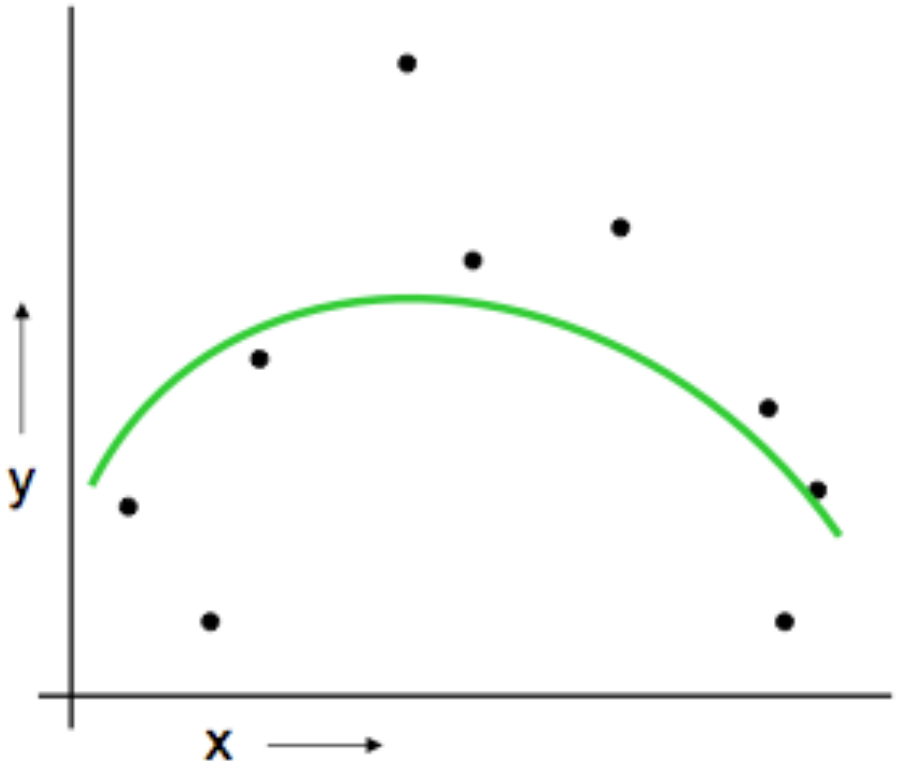
Linear Regression



Linear regression:

- an approach to model the relationship between a scalar dependent variable **y** and one or more explanatory variables denoted **X**

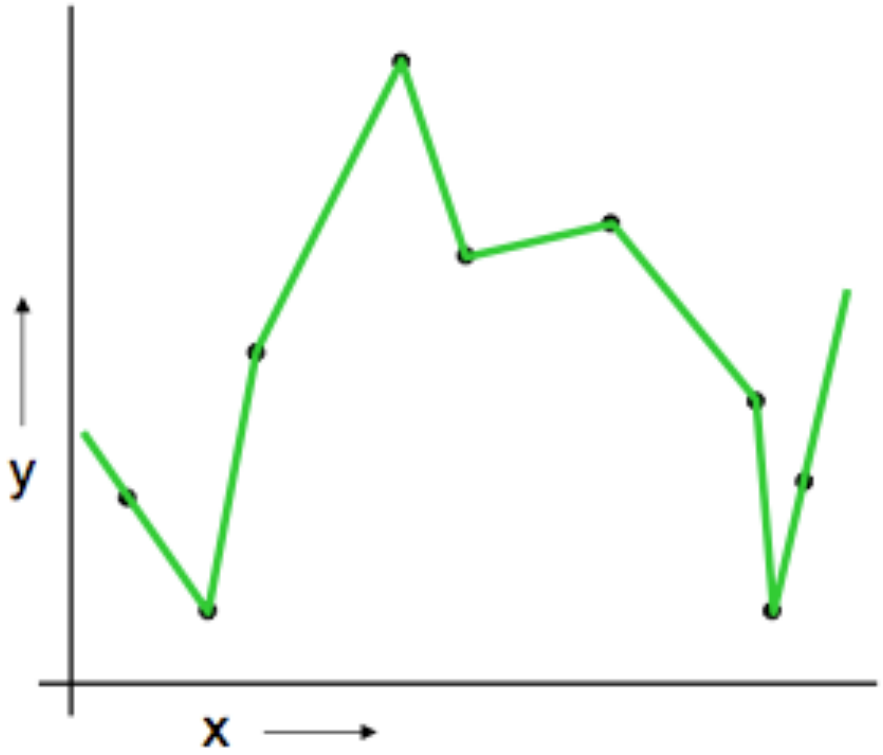
Quadratic Regression



Quadratic regression:

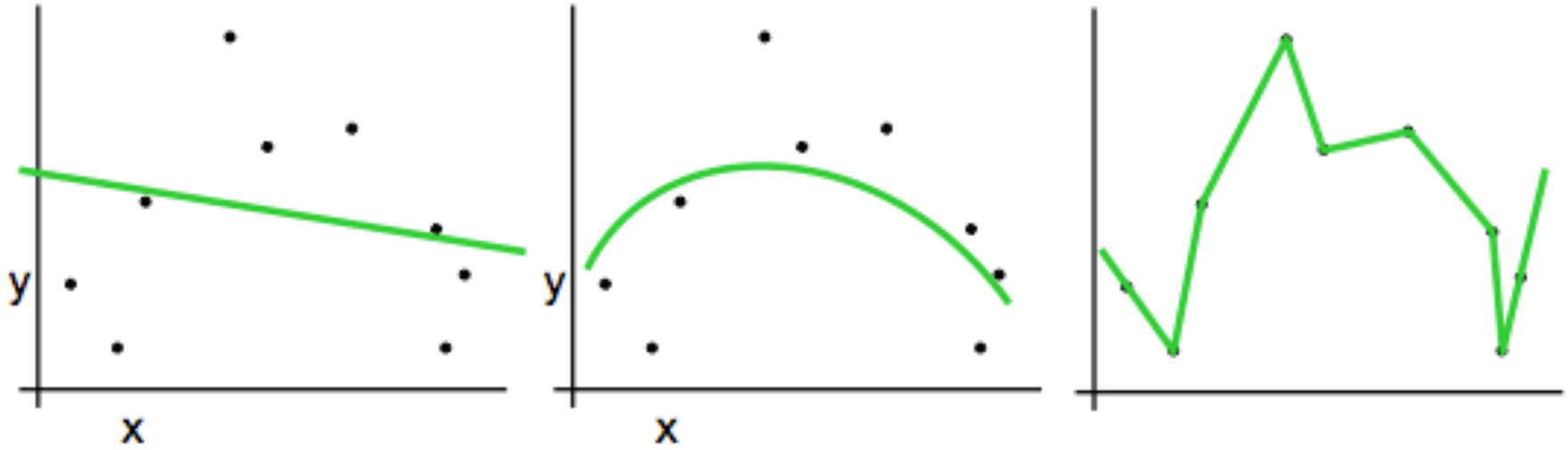
- the process of finding the equation of the parabola that fits best for a set of data

Join-the-dots



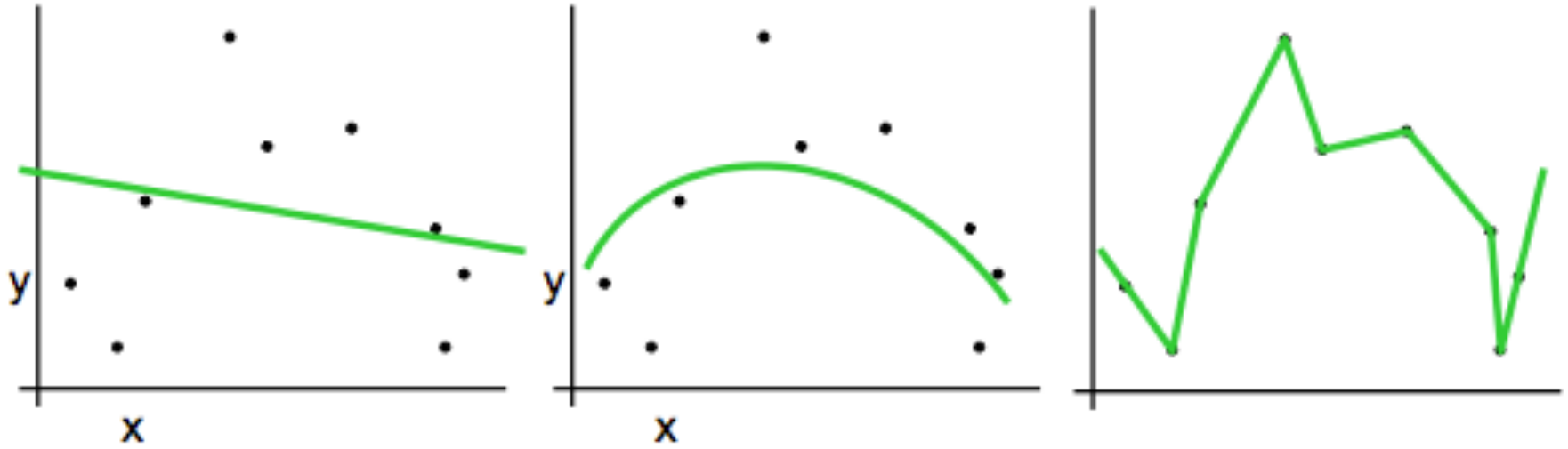
Also known as piecewise linear nonparametric regression

Which is best?



How to choose the method with the best fit to the data?

What do we really want?



How to choose the method with the best fit to the data?

How well a model is going to predict future data drawn from the same distribution?

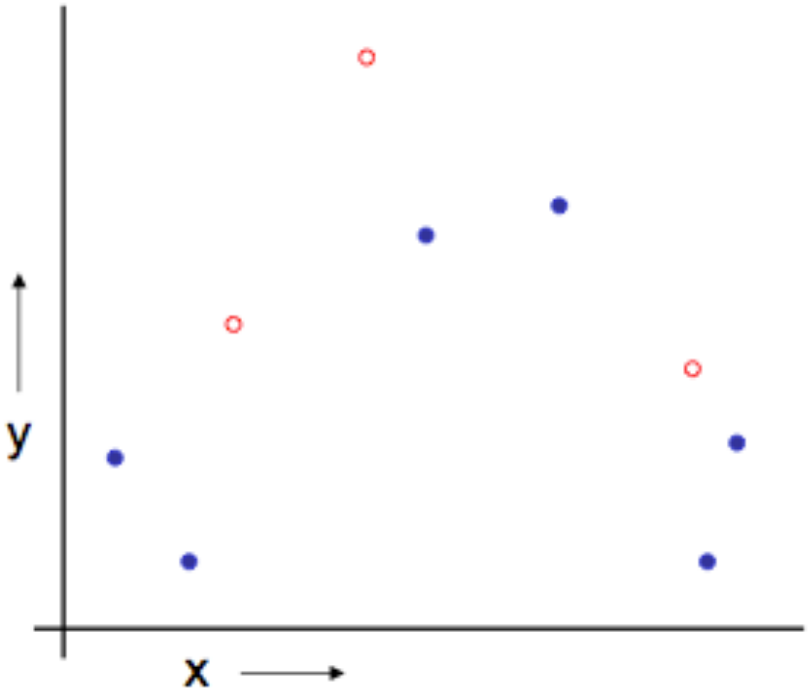
Mean Squared Error

Mean Squared Error (MSE)

- one of many ways to quantify the difference between values implied by a model (aka estimator) and the true values of the quantity being estimated
- Commonly used in regression analysis

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2.$$

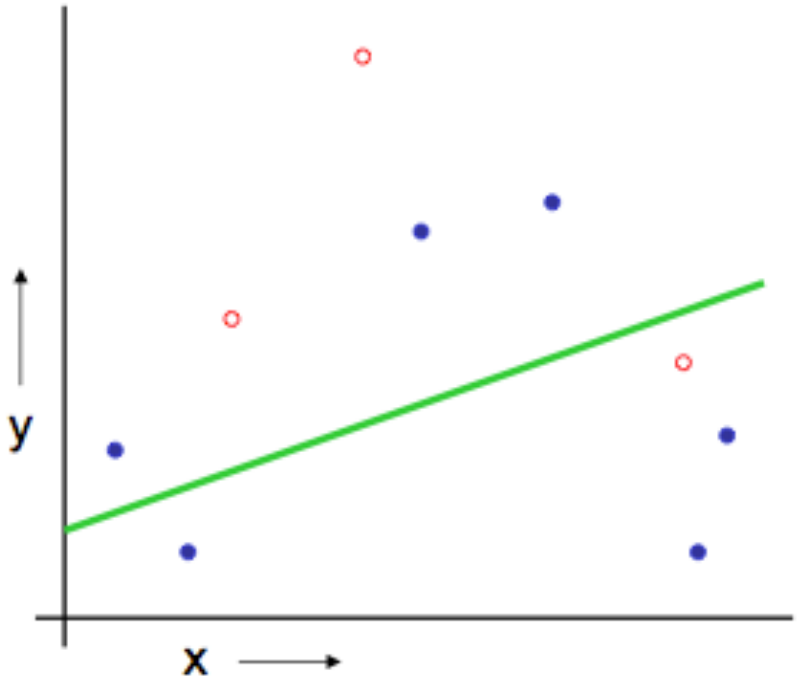
The test set method



1. **Randomly** choose 30% of the data to be in a test set

2. The remainder is a training set

The test set method



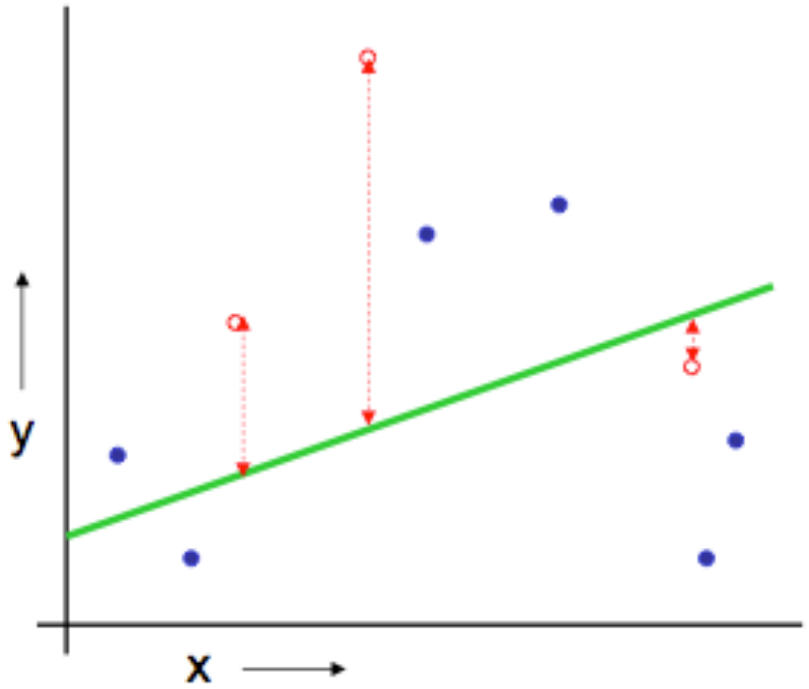
(Linear regression example)

1. Randomly choose 30% of the data to be in a test set

2. The remainder is a training set

3. Perform your regression on the training set

The test set method



(Linear regression example)

Mean Squared Error = 2.4

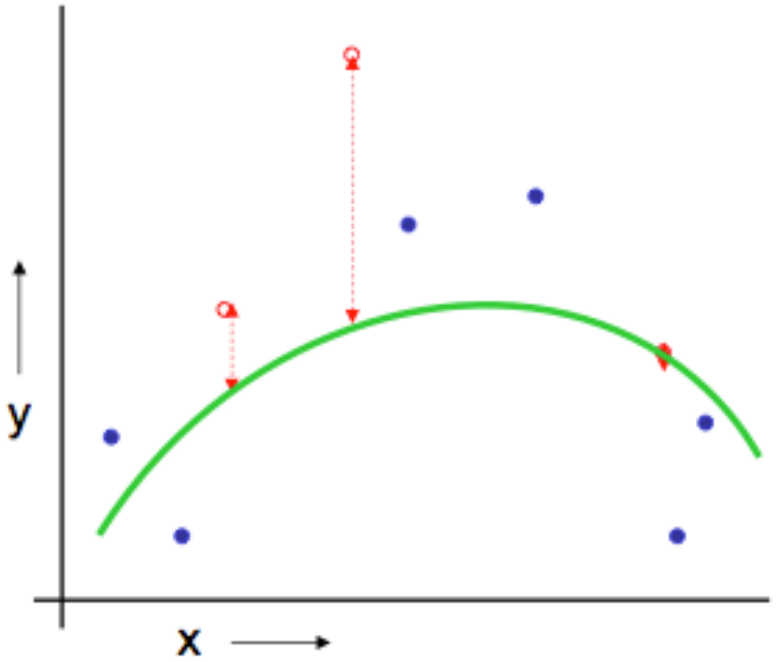
1. Randomly choose 30% of the data to be in a test set

2. The remainder is a training set

3. Perform your regression on the training set

4. Estimate your future performance with the test set

The test set method



(Quadratic regression example)

Mean Squared Error = 0.9

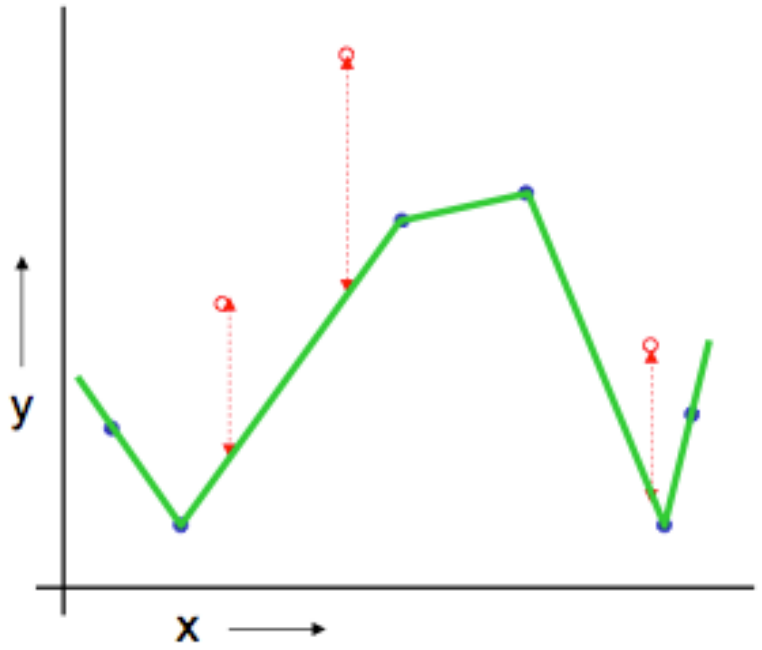
1. Randomly choose 30% of the data to be in a test set

2. The remainder is a training set

3. Perform your regression on the training set

4. Estimate your future performance with the test set

The test set method



(Join the dots example)

Mean Squared Error = 2.2

1. Randomly choose 30% of the data to be in a test set

2. The remainder is a training set

3. Perform your regression on the training set

4. Estimate your future performance with the test set

The test set method

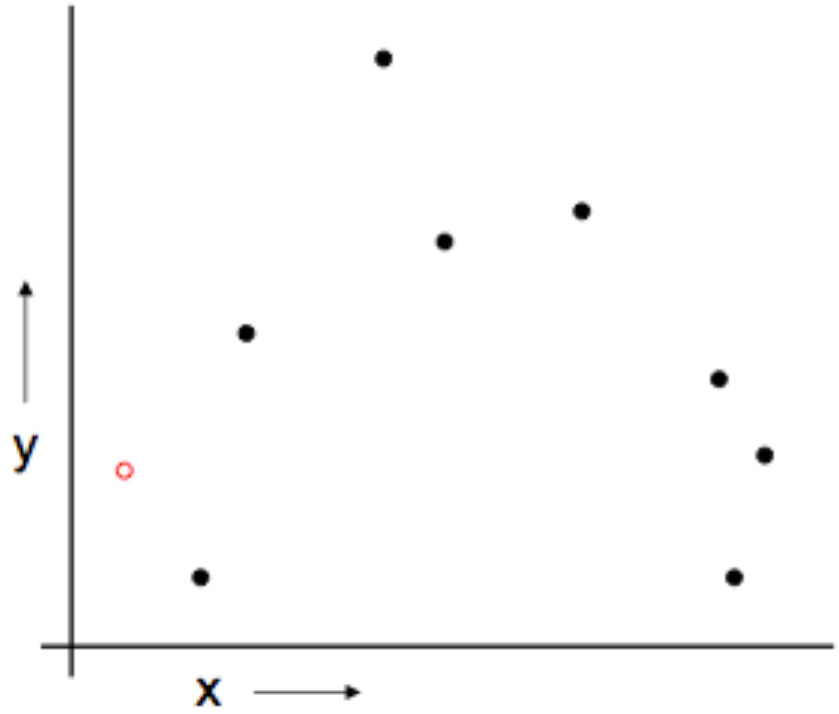
Good news

- Very very simple
- Can then simply choose the method with the best test-set score

Bad news

- Wastes data: we get an estimate of the best method to apply to 30% less data
- If we don’t have much data, our test-set might just be lucky or unlucky

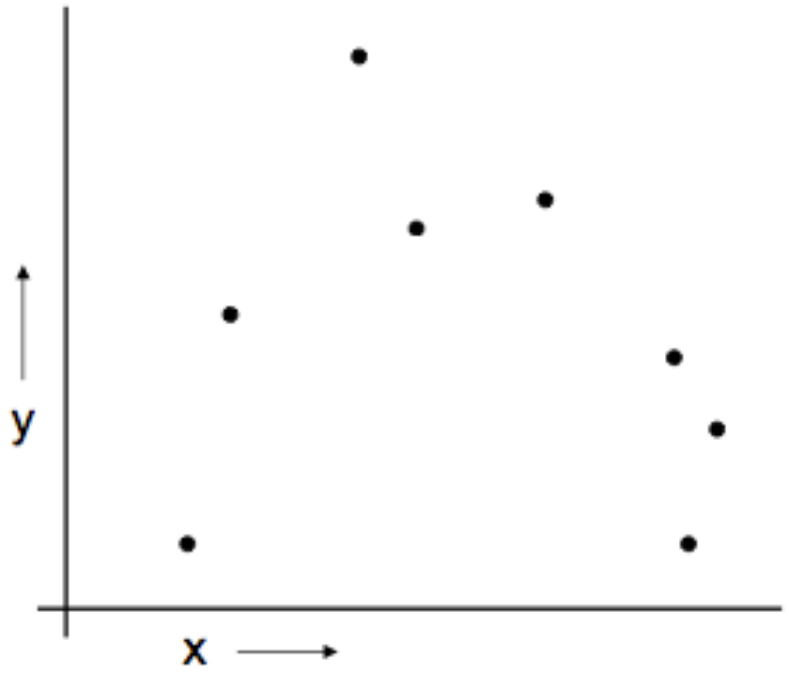
Leave-one-out Cross Validation(LOOCV)



For k=1 to R

1. Let (x_k, y_k) be the k^{th} record

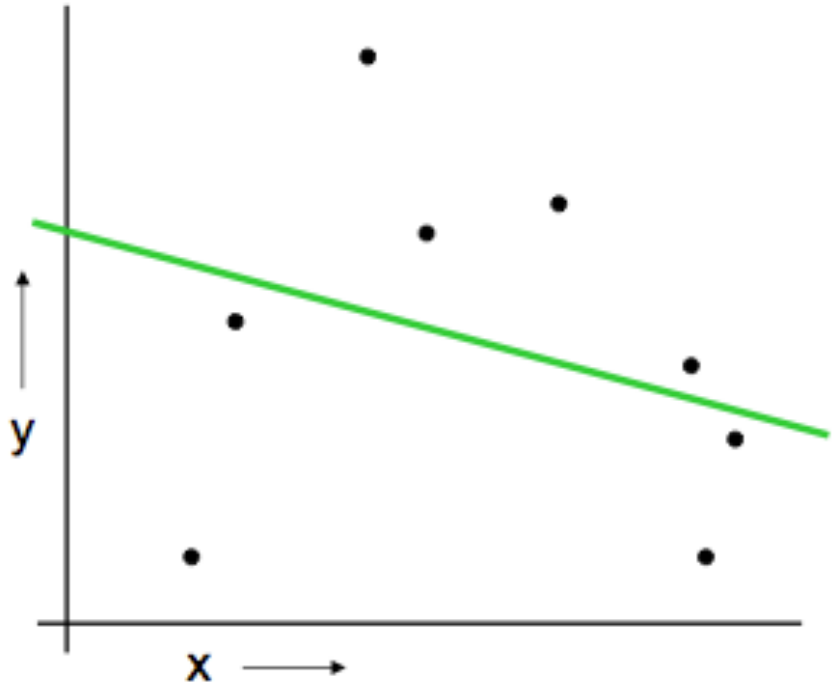
Leave-one-out Cross Validation(LOOCV)



For k=1 to R

1. Let (x_k, y_k) be the k^{th} record
2. Temporarily remove (x_k, y_k) from the dataset

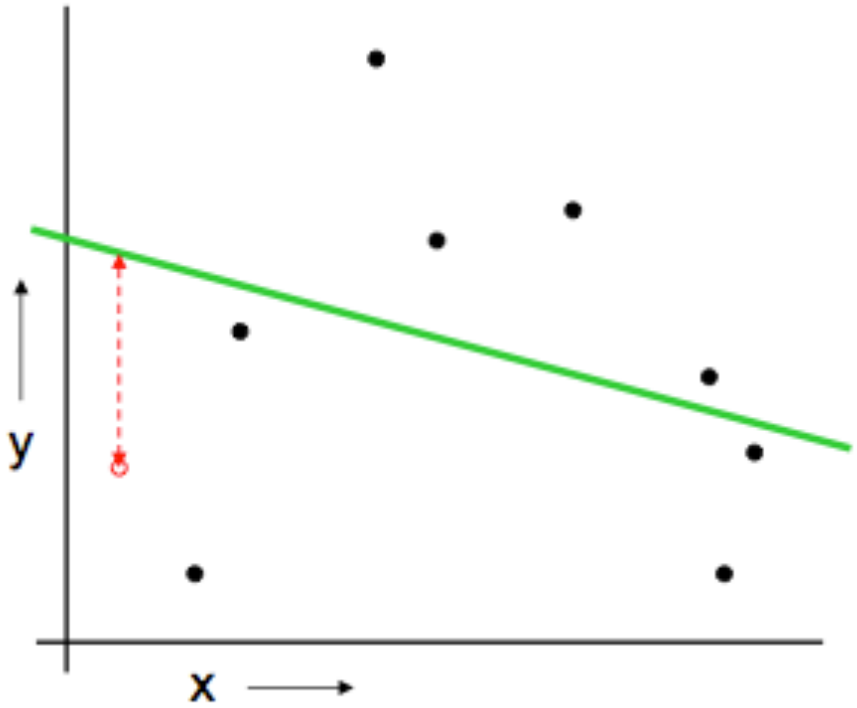
Leave-one-out Cross Validation(LOOCV)



For k=1 to R

1. Let (x_k, y_k) be the k^{th} record
2. Temporarily remove (x_k, y_k) from the dataset
3. Train on the remaining R-1 datapoints

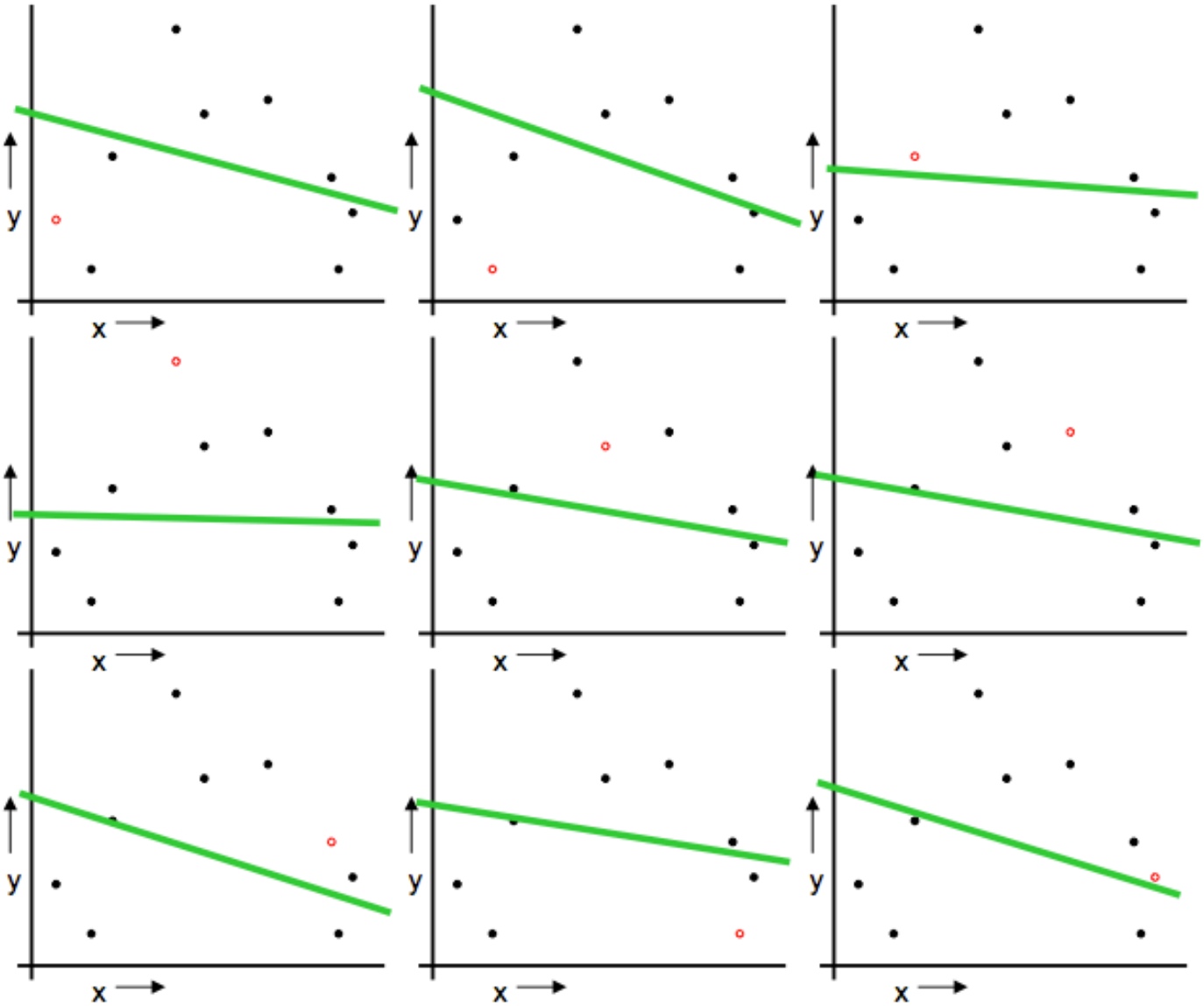
Leave-one-out Cross Validation(LOOCV)



For k=1 to R

1. Let (x_k, y_k) be the k^{th} record
2. Temporarily remove (x_k, y_k) from the dataset
3. Train on the remaining R-1 datapoints
4. Note your error (x_k, y_k)

Leave-one-out Cross Validation(LOOCV)



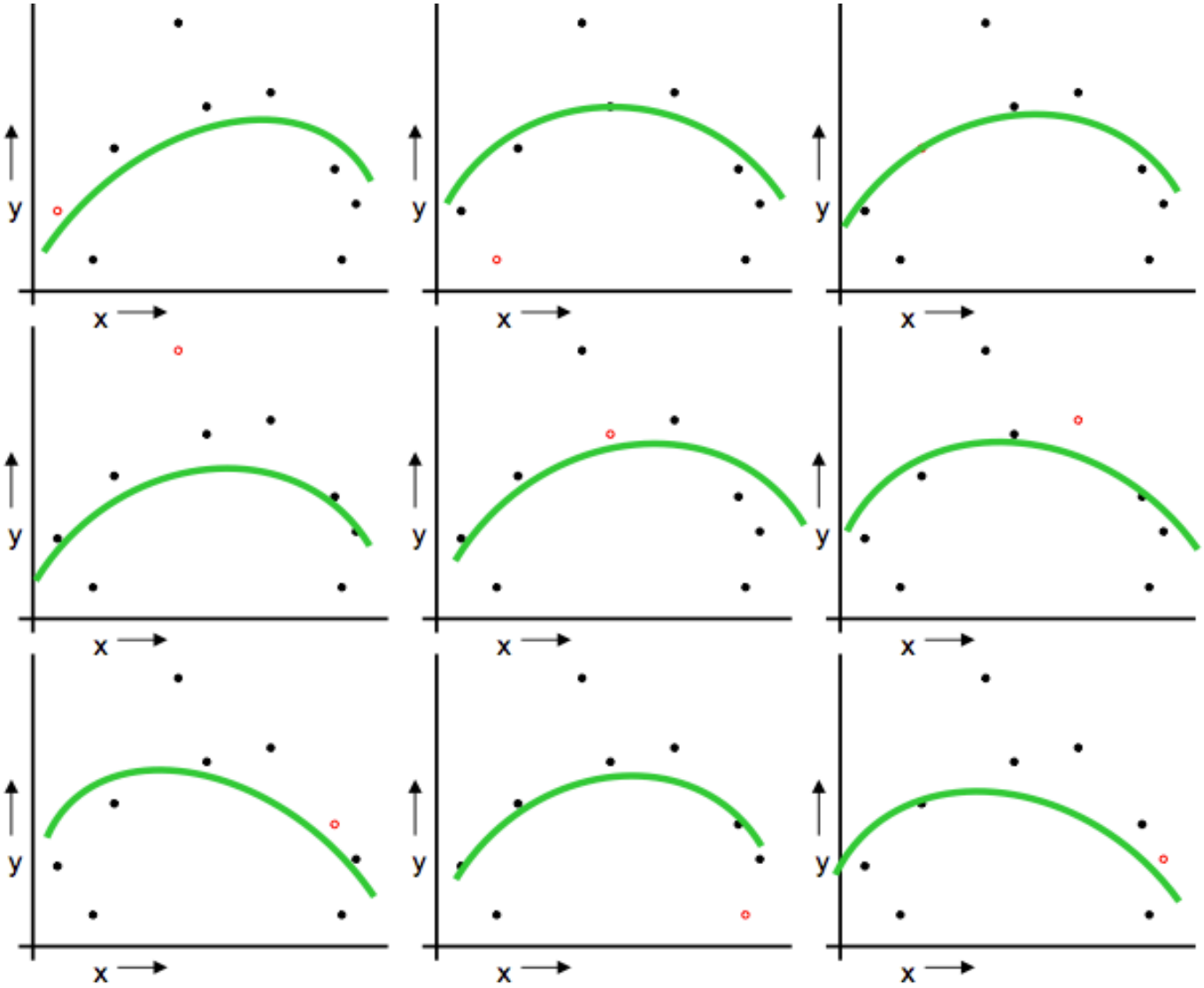
For k=1 to R

1. Let (x_k, y_k) be the k^{th} record
2. Temporarily remove (x_k, y_k) from the dataset
3. Train on the remaining R-1 datapoints
4. Note your error (x_k, y_k)

When you've done all points, report the mean error.

$$MSE_{LOOCV} = 2.12$$

LOOCV for Quadratic Regression



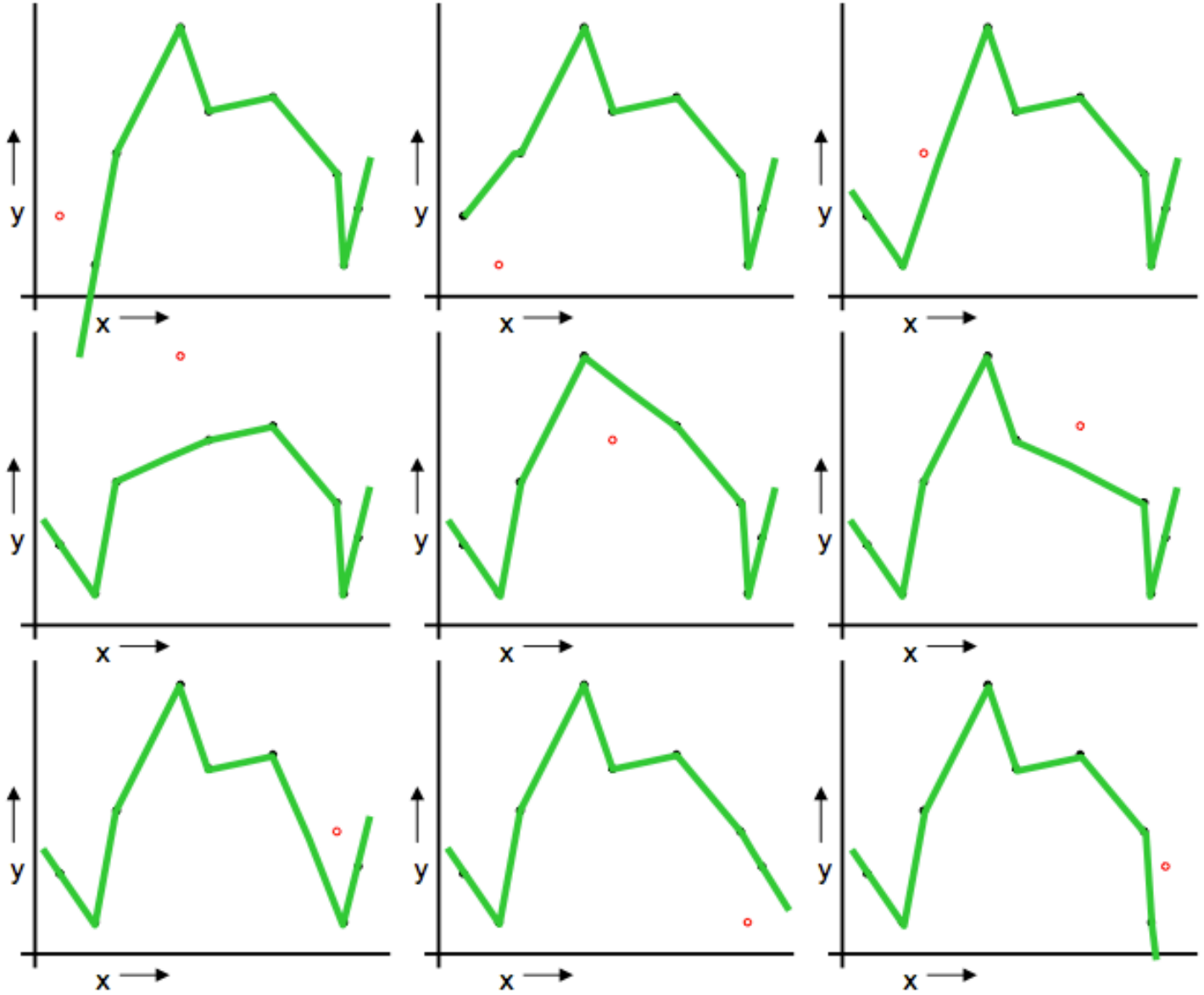
For k=1 to R

1. Let (x_k, y_k) be the k^{th} record
2. Temporarily remove (x_k, y_k) from the dataset
3. Train on the remaining R-1 datapoints
4. Note your error (x_k, y_k)

When you've done all points, report the mean error.

$MSE_{LOOCV} = 0.962$

LOOCV for Non-Parametric Regression



For k=1 to R

1. Let (x_k, y_k) be the k^{th} record
2. Temporarily remove (x_k, y_k) from the dataset
3. Train on the remaining R-1 datapoints
4. Note your error (x_k, y_k)

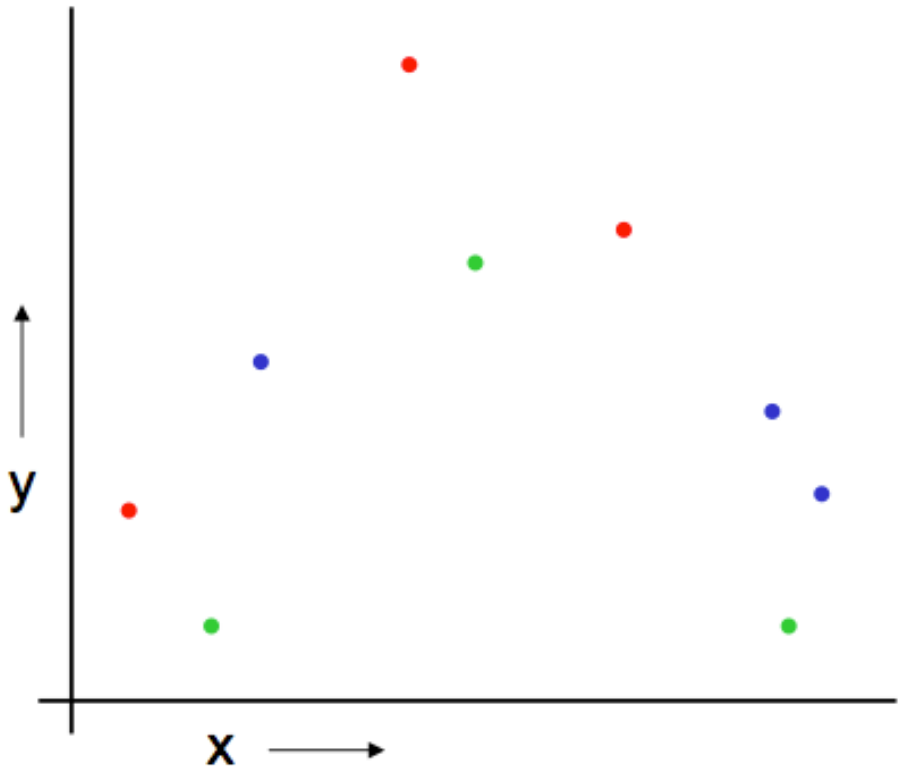
When you've done all points, report the mean error.

$MSE_{LOOCV} = 3.33$

Which kind of validation?

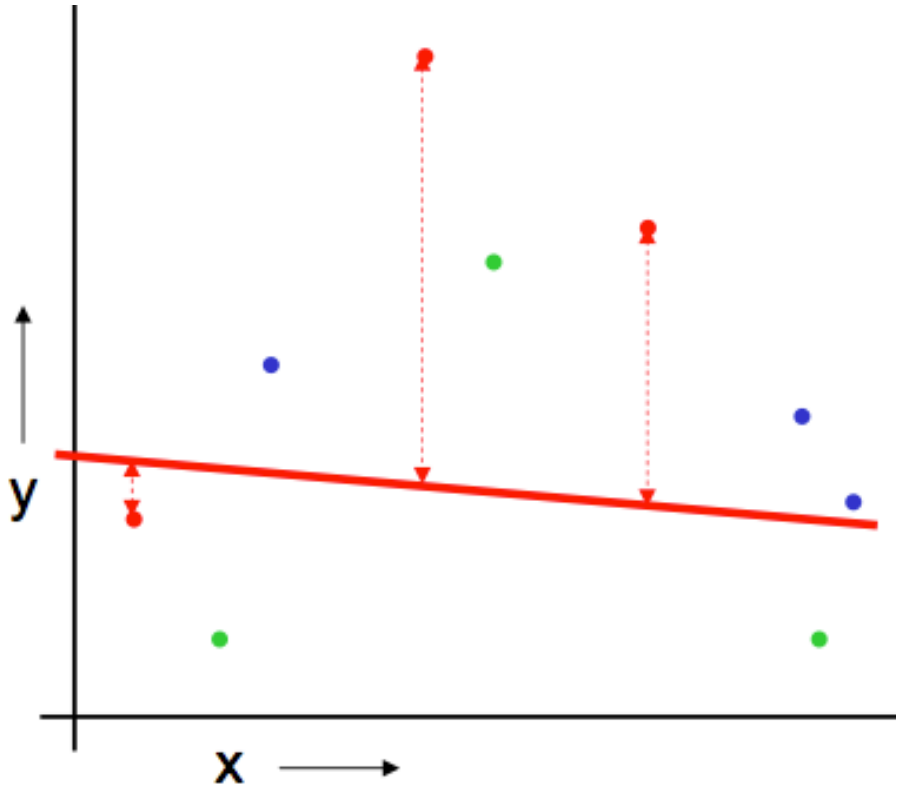
	Downside	Upside
Test-set	Variance: unreliable estimate of future performance	Cheap
Leave-one-out	Expensive.	Doesn't waste data

K-fold Cross validation



Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored Red Green and Blue)

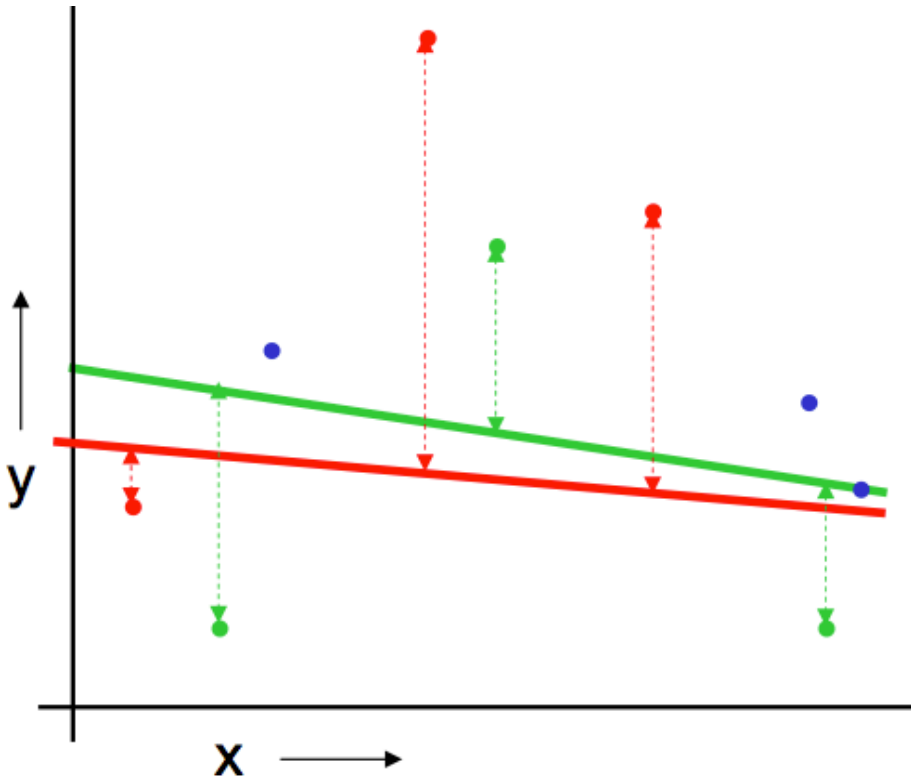
K-fold Cross validation



Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored Red Green and Blue)

For the red partition: Train on all the points not in the red partition. Find the test-set sum of errors on the red points.

K-fold Cross validation

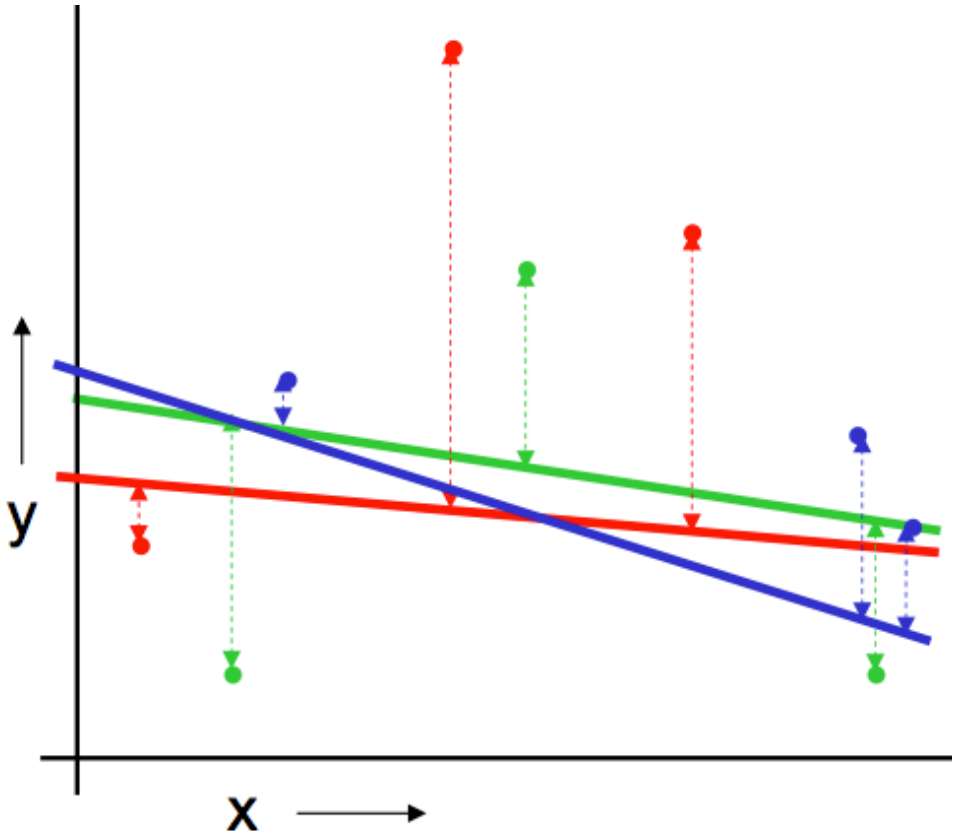


Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored Red Green and Blue)

For the red partition: Train on all the points not in the red partition. Find the test-set sum of errors on the red points.

For the green partition: Train on all the points not in the green partition. Find the test-set sum of errors on the green points.

K-fold Cross validation



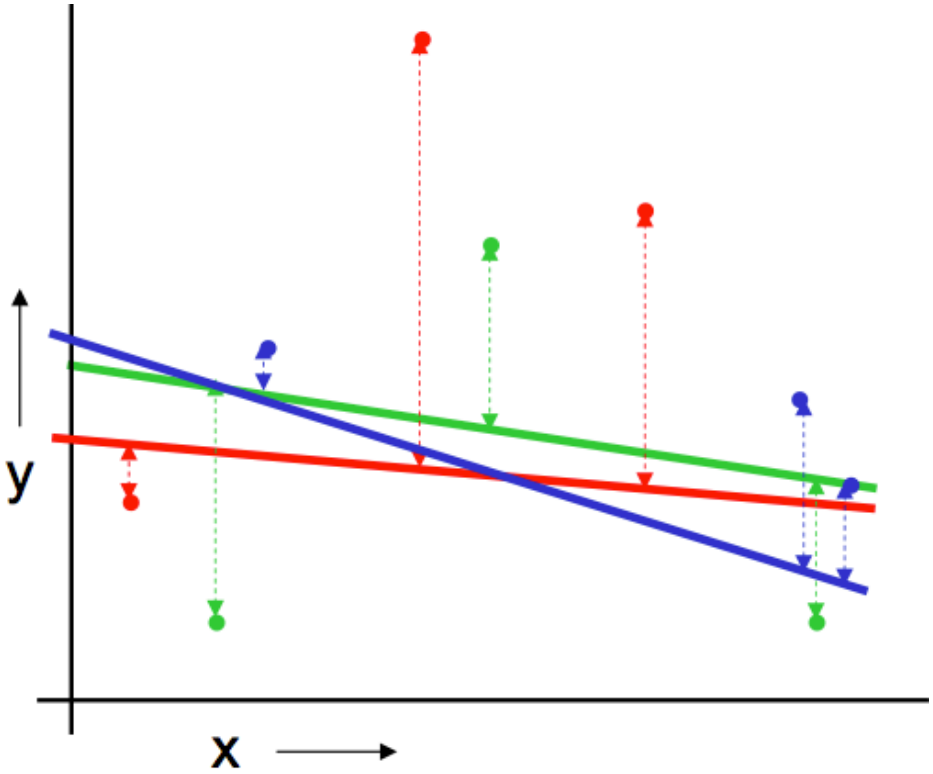
Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored Red Green and Blue)

For the red partition: Train on all the points not in the red partition. Find the test-set sum of errors on the red points.

For the green partition: Train on all the points not in the green partition. Find the test-set sum of errors on the green points.

For the blue partition: Train on all the points not in the blue partition. Find the test-set sum of errors on the blue points.

K-fold Cross validation



Linear Regression

$MSE_{3FOLD}=2.05$

Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored Red Green and Blue)

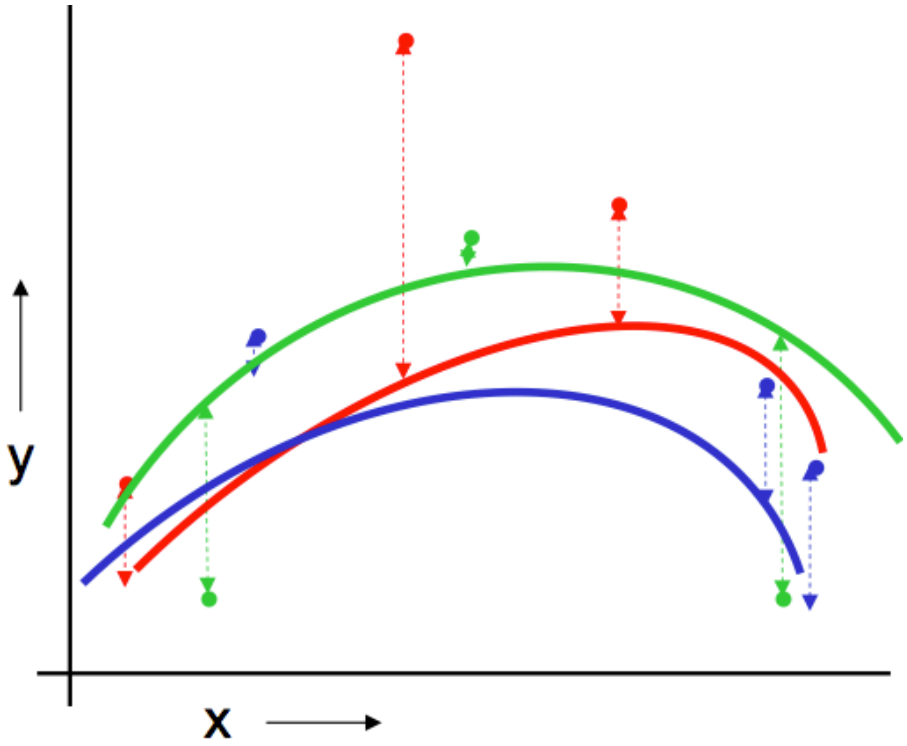
For the red partition: Train on all the points not in the red partition. Find the test-set sum of errors on the red points.

For the green partition: Train on all the points not in the green partition. Find the test-set sum of errors on the green points.

For the blue partition: Train on all the points not in the blue partition. Find the test-set sum of errors on the blue points.

Then report the mean error

K-fold Cross validation



Quadratic Regression

$MSE_{3FOLD}=1.11$

Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored Red Green and Blue)

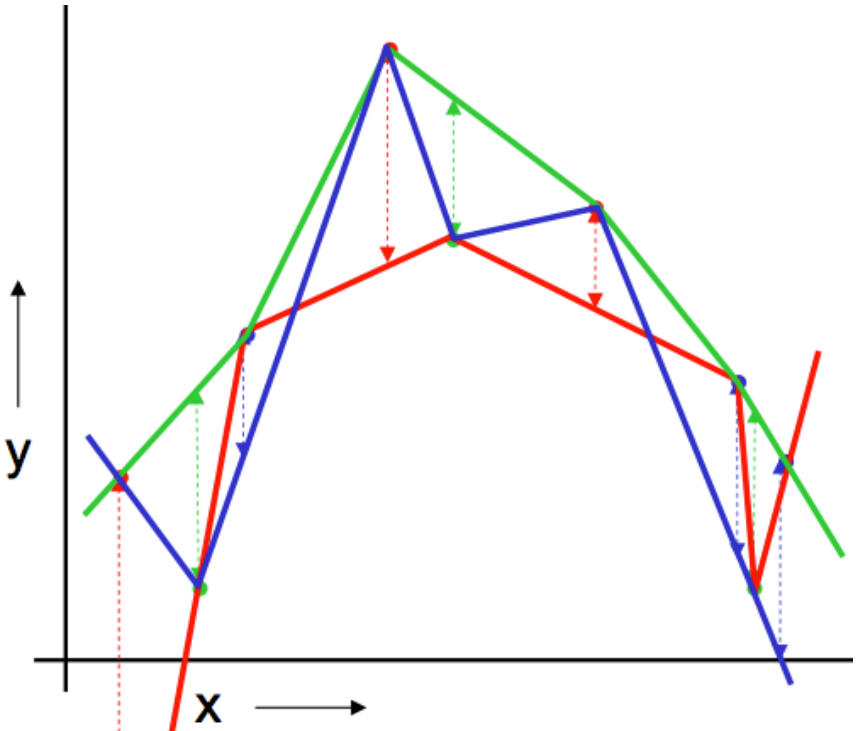
For the red partition: Train on all the points not in the red partition. Find the test-set sum of errors on the red points.

For the green partition: Train on all the points not in the green partition. Find the test-set sum of errors on the green points.

For the blue partition: Train on all the points not in the blue partition. Find the test-set sum of errors on the blue points.

Then report the mean error

K-fold Cross validation



Joint-the-dots

$MSE_{3FOLD}=2.93$

Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored Red Green and Blue)

For the red partition: Train on all the points not in the red partition. Find the test-set sum of errors on the red points.

For the green partition: Train on all the points not in the green partition. Find the test-set sum of errors on the green points.

For the blue partition: Train on all the points not in the blue partition. Find the test-set sum of errors on the blue points.

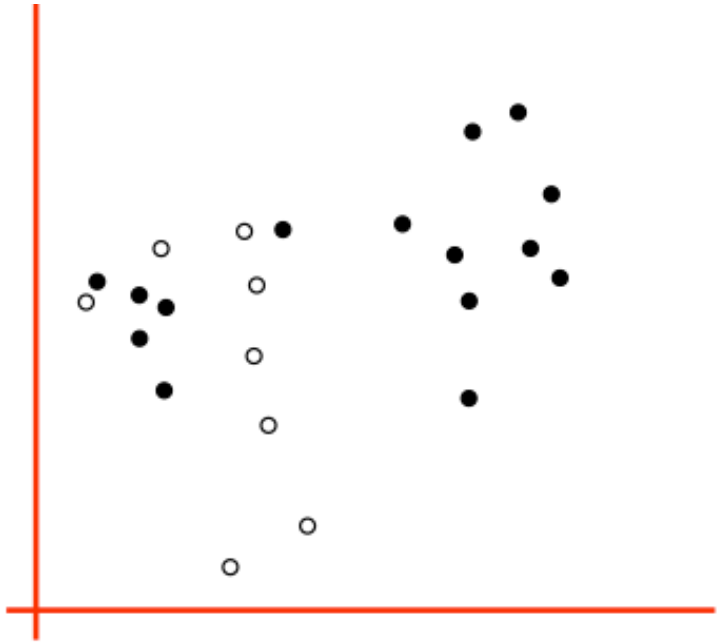
Then report the mean error

Which kind of validation

	Downside	Upside
Test-set	Variance: unreliable estimate of future performance	Cheap
Leave-one-out	Expensive.	Doesn't waste data
10-fold	Wastes 10% of the data. 10 times more expensive than test set	Only wastes 10%. Only 10 times more expensive instead of R times.
3-fold	Wastier than 10-fold. Expensivier than test set	Slightly better than test-set
R-fold	Identical to Leave-one-out	

Cross-validation for classification

- Instead of computing the sum squared errors on a test set, you should compute
  - The total number of misclassifications on a testset.
  - E.g., the test set has 10 data points (4 data point -> positive, 6 data point -> negative)
  - Your classifier somehow predicted them all as positive …



Summary

What you should know

- Accuracy
- Precision, recall, F-measure
- ROC curve
- when to use accuracy or F-measure
- Why you can't use "training-set-error" to estimate the quality of your learning algorithm on your data, or to choose the learning algorithm
- Test-set cross-validation
- Leave-one-out cross-validation
- k-fold cross-validation