Data Science for Business
Logistic Regression and Gradient Descent

Asst. Prof. Teerapong Leelanupab (Ph.D.)
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang (KMITL)

Week 9.1

Regression with categorical features

Given a dataset $\{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$, where the y are categorical (or qualitative), we would like to be able to predict which category y takes, given an example instance x.

Linear regression does not work well for this problem.

Consider the following dataset:

Features are computed from a digitised image of a fine needle aspirate (FNA) of a breast mass.

x: radius (mean of distances from centre to points on the perimeter

y: Diagnosis (M=malignant, B=benign)

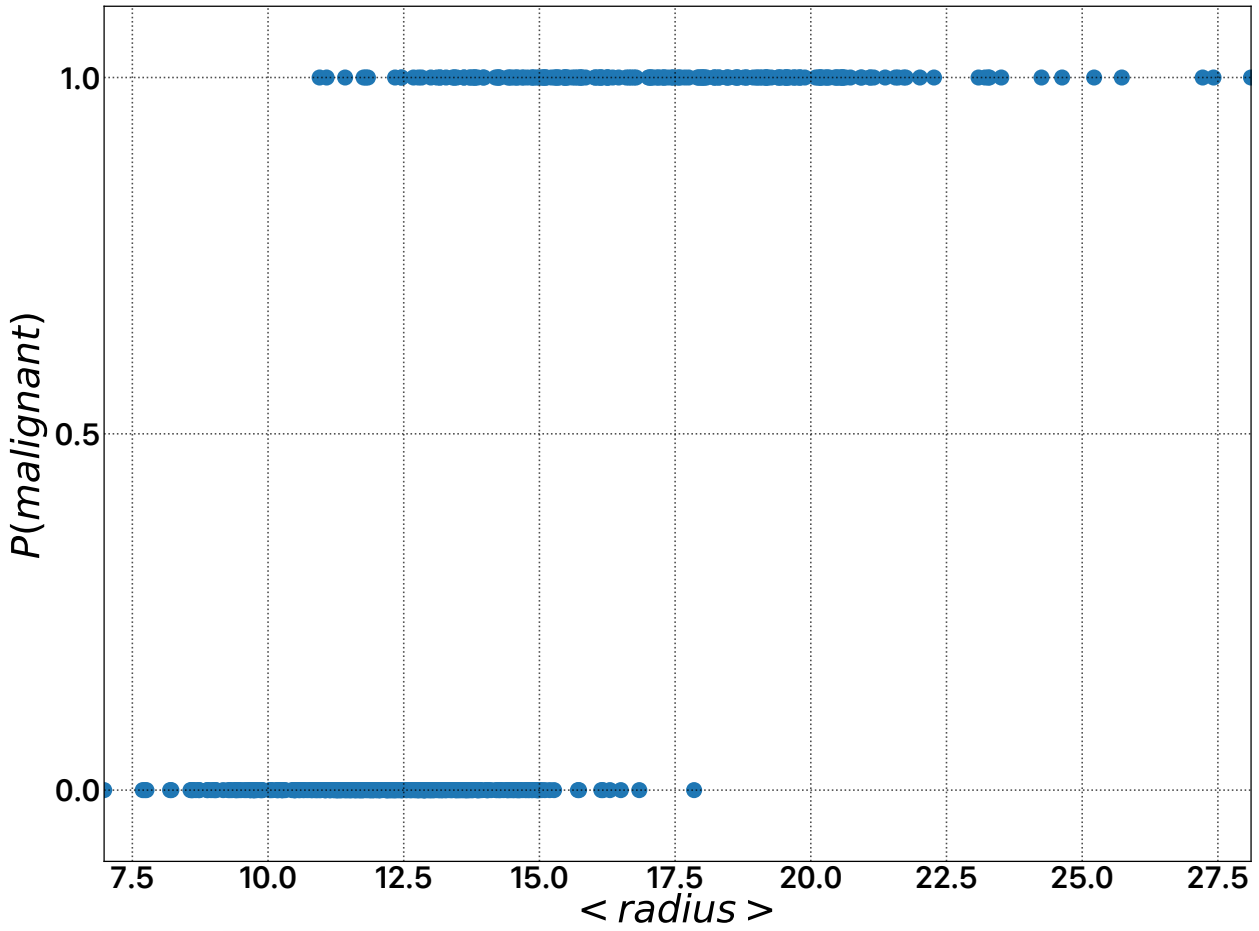
Regression with categorical features

y = 0: benign

1: malignant

x: mean radius size of tutor

What happens if we try to fit a linear regression model to this data?

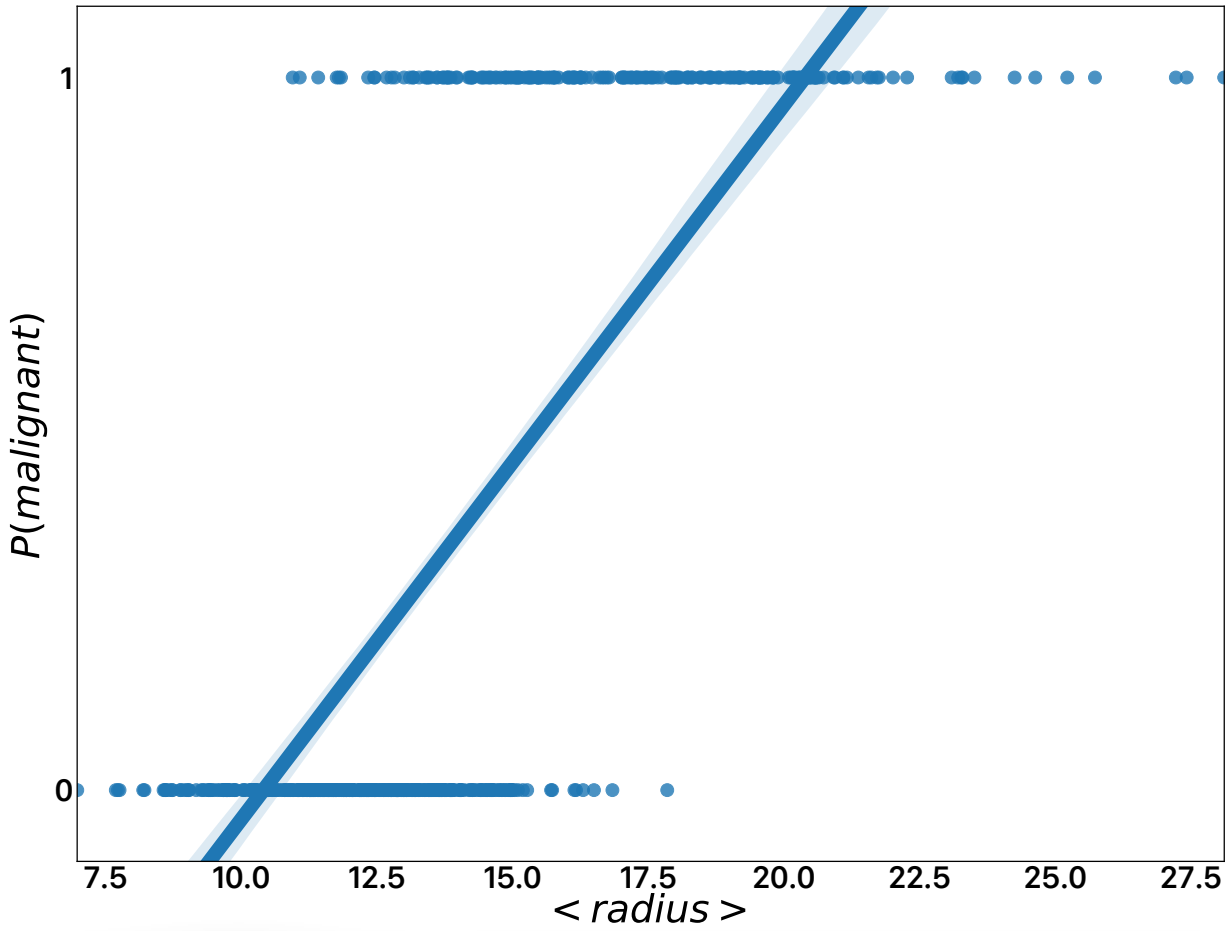


Regression with categorical features

We can certainly fit a linear regression model to the data, but there are clearly a number of issues:

What does the regression model predict for a given tumour size (eg. If the mean radius is 25.0 then we predict a value of 1.46).

We would prefer a mapping to the {malignant, benign} classes



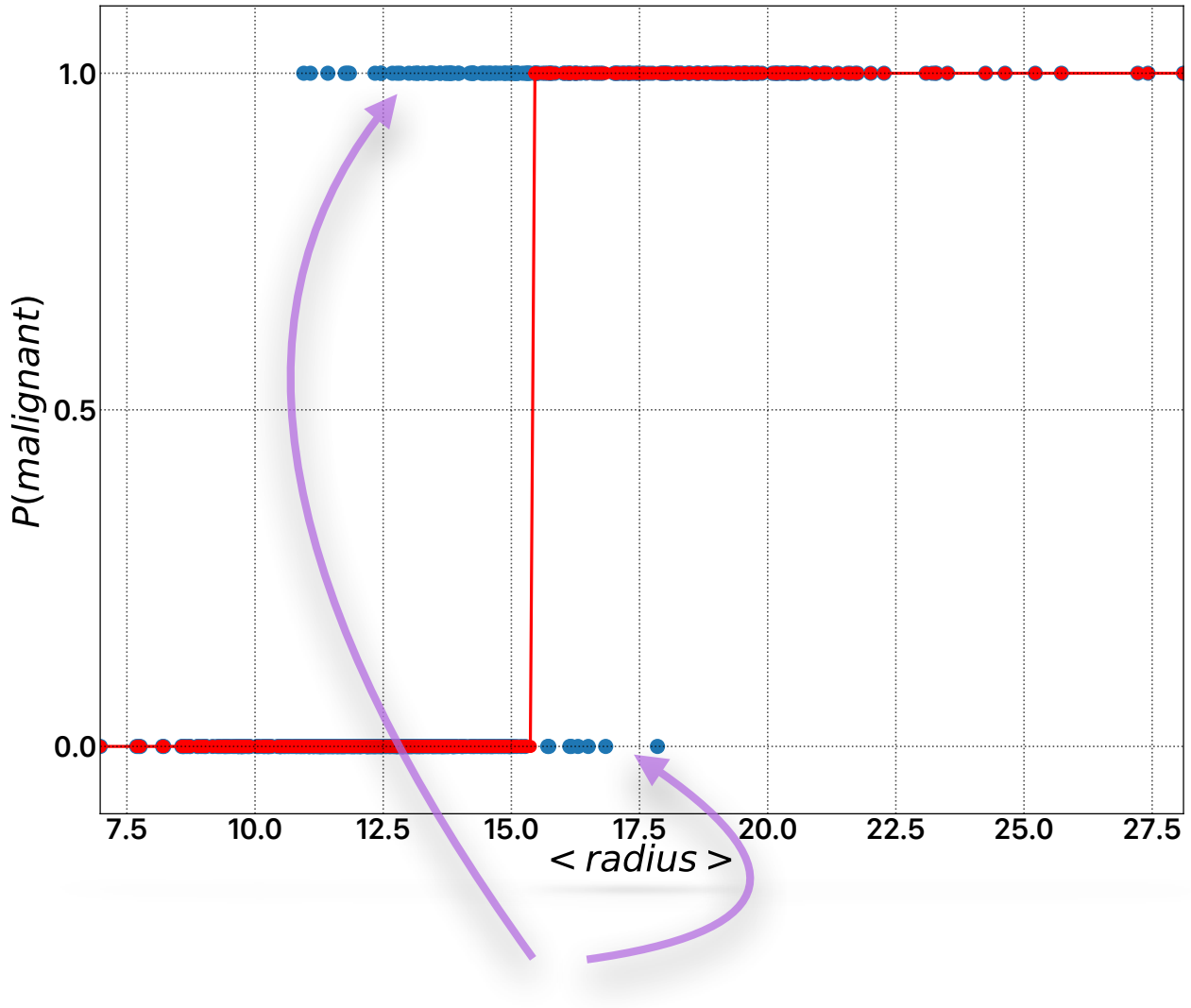
$$P(\text{malignant}) = -1.0436 + 0.1002 \times \langle \text{radius} \rangle$$

Regression with categorical features

We can apply the linear regression model and then use the following:

$$\hat{y} = \begin{cases} 1 & \text{when } x > 0.5 \\ 0 & \text{when } x \leq 0 \end{cases}$$

to predict whether the cancer is malignant based on the tumour size.



Mis-classified examples

Logistic Regression

Logistic Regression addresses the problem of estimating a probability, P(Y = 1), to be outside the range of [0, 1].

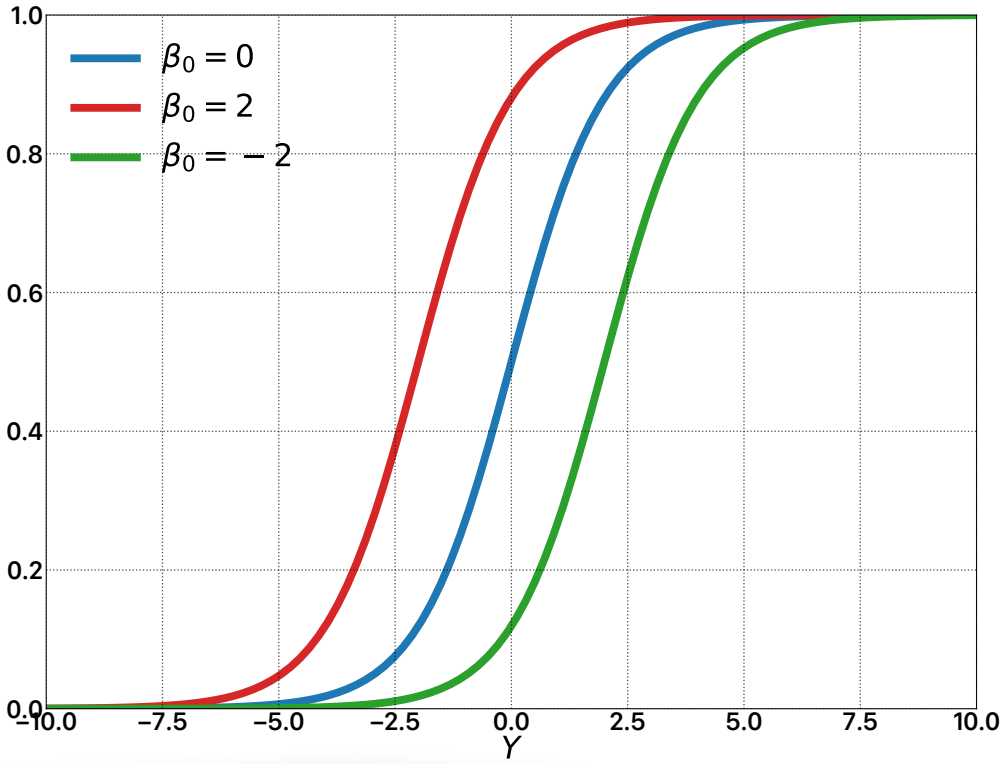
The logistic regression model uses a function, called the logistic function, to model P(Y = 1):

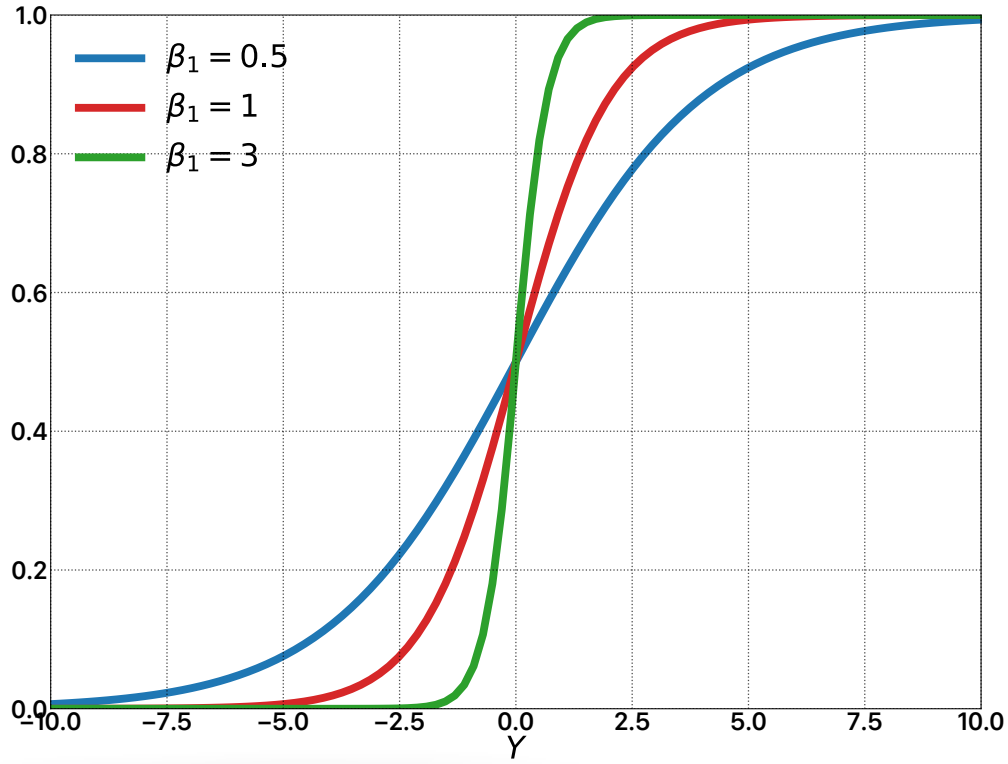
$$P(Y = 1) = \frac{\exp^{\beta_0 + \beta_1 X}}{1 + \exp^{\beta_0 + \beta_1 X}}$$

As a result the model will predict P(Y = 1) with an S-shaped curve which is the general shape of the logistic function.

- β_0 shifts the curve right or left
- β_1 controls how steep the S-shaped curve is.

Logistic Regression





β_0 shifts the curve right or left

β_1 controls how steep the S-shaped curve is.

Logistic Regression

We can re-arrange the logistic formula:

$$\log\left(\frac{P(Y=1)}{1-P(Y=1)}\right) = \beta_0 + \beta_1 X$$

where: $\text{odds ratio} = \left(\frac{P(Y=1)}{1-P(Y=1)}\right)$

when the odds ratio is greater than 1, it describes a ***positive*** relationship (eg. *as tumour size "increases,"* the odds of malignancy *increases*.

when the odds ratio is less than 1, it describes a ***negative*** relationship (eg. *as tumour size "decreases,"* the odds of malignancy de*creases*.

$$\text{odds ratio} = \frac{\text{odds(malignant)}}{\text{odds(benign)}}$$

logistic regression is said to model the log-odds with a linear function of the predictors or features, *X*.

Interpretation:

- β_0 is the log-odds of a benign tumour, P(*Y*=0)
- $\beta_0 + \beta_1$ is the log-odds of a malignant tumour, P(*Y*=1)
- a one unit change in *X* is associated with a β_1 change in the log-odds of $Y = 1$;
- a one unit change in *X* is associated with an e^{β_1} change in the odds that $Y = 1$.

Logistic Regression

- Estimating Parameters:
- In simple linear regression there are closed form solutions for the estimated parameters β_i's.
- In Logistic regression there is no such closed-form solution
- We must use other techniques to find the best fit parameters

- Given independent observations of y, what is the likelihood function for p:

$$L(p|Y) = \prod P(Y_i = y_i) = \prod p_i^{y_i}(1 - p_i)^{(1-y_i)}$$

- we could try to take the log and differentiate to find the maximum, but this is messy. Iterative approaches are easier.

Logistic Regression

Interpreting the logistic regression model:

Logistic Regression

Coefficients...

Variable M

==========

radius_mean 1.0336

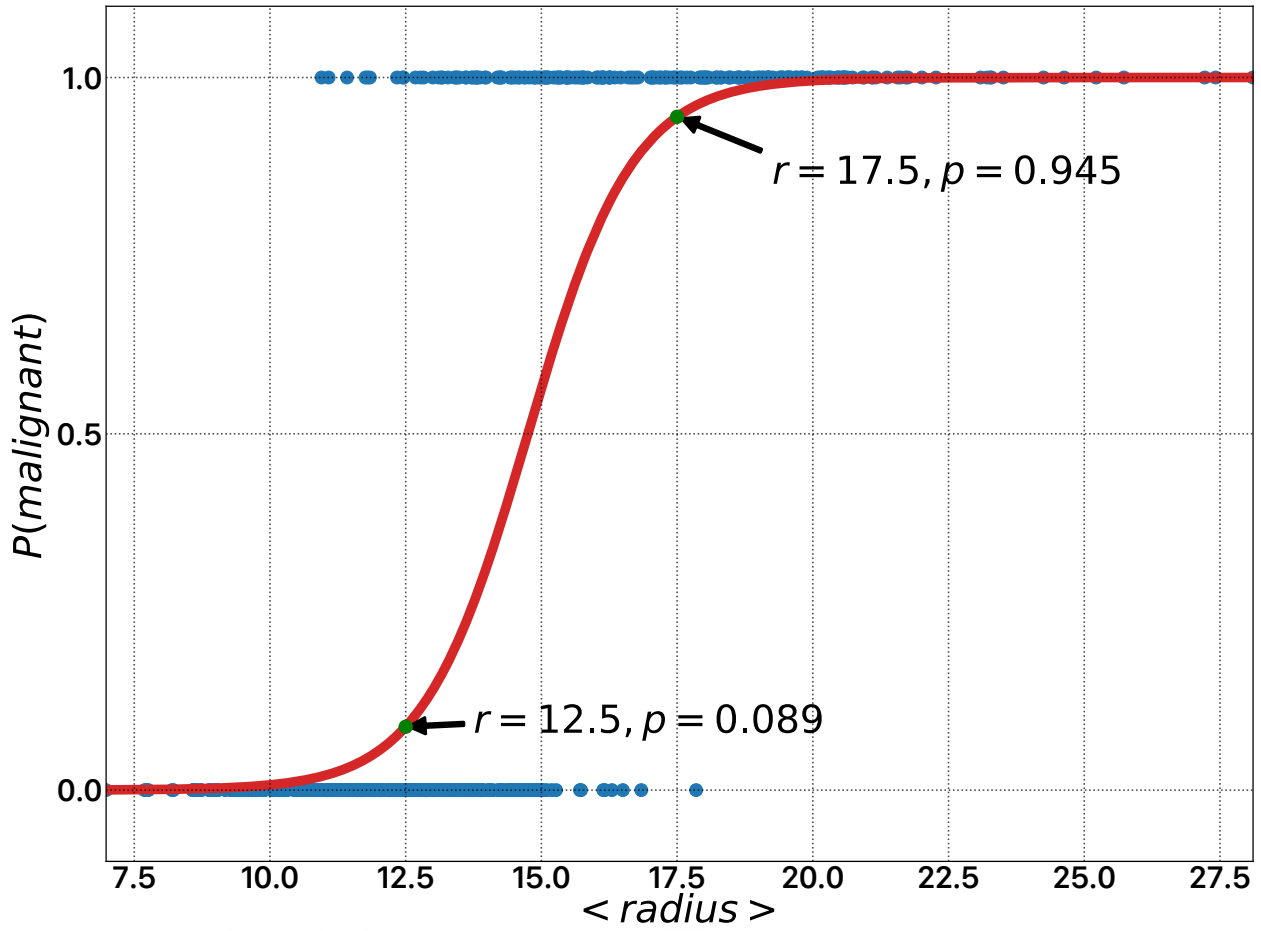
Intercept -15.2459

Odds Ratios...

Variable M

==========

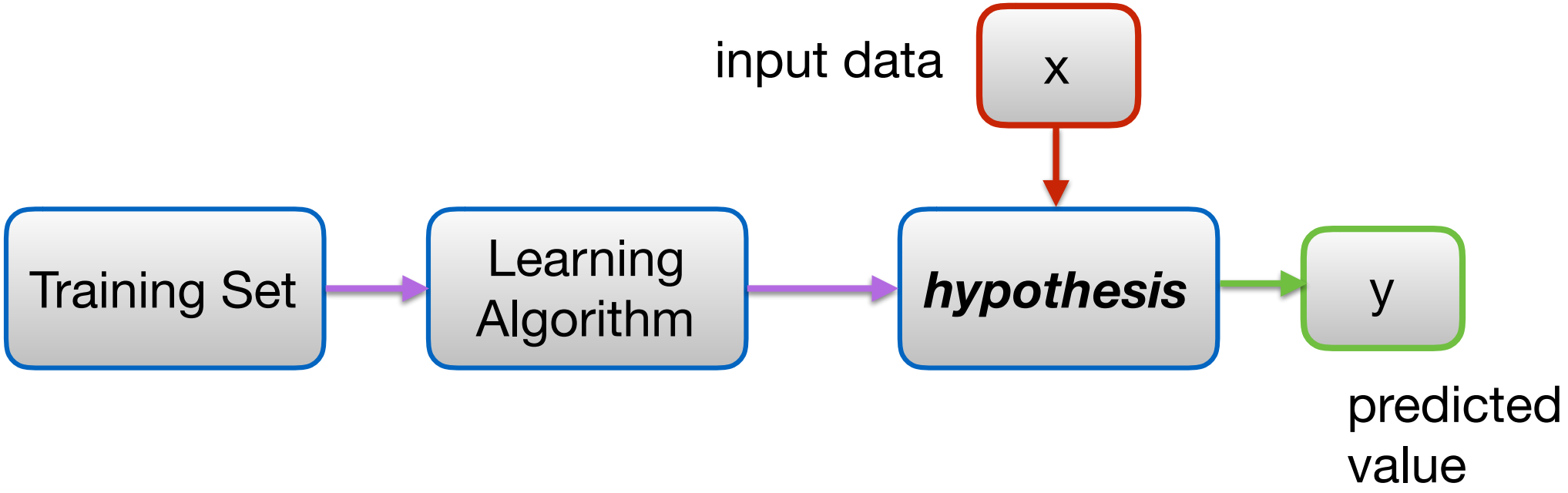
radius_mean 2.8111



$$P(\text{malignant}) = \text{logistic}(\langle \text{radius} \rangle, \beta_0 = -15.25, \beta_1 = 1.03)$$

Supervised Learning Problem

- Given a training set, we would like to learn a function h : $X \rightarrow Y$ so that h(x) is a “good” predictor for the corresponding value of y.
- the function h is called a hypothesis
- example:
- input data: tumour radius size
- output data: malignant or benign



Supervised Learning Problem

- we have already seen examples of the hypothesis function:

- linear regression:

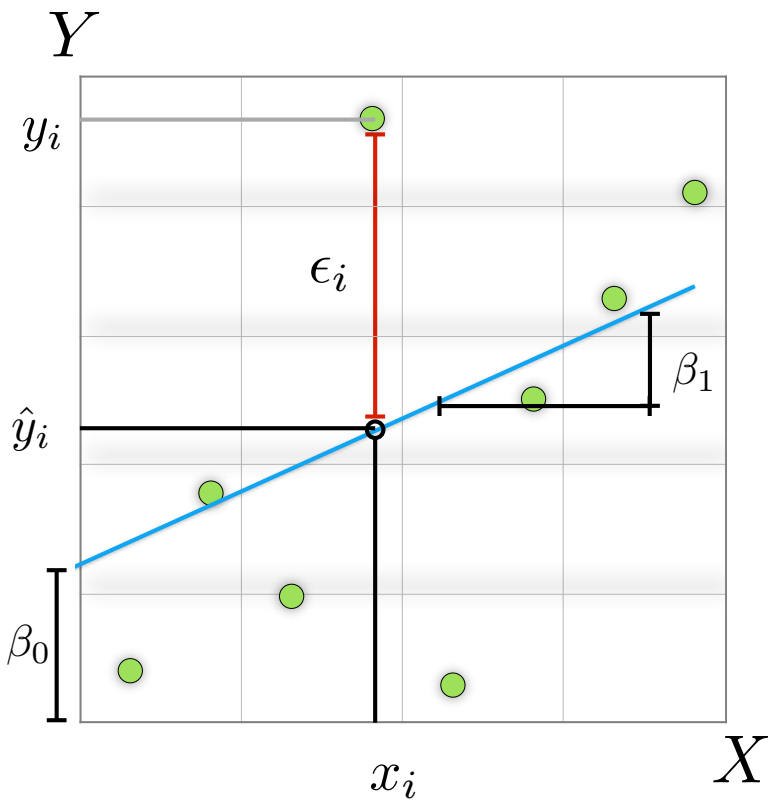
$$h(x) = \beta_0 + \beta_1 x$$

- logistic regression:

$$h(x) = \frac{\exp^{\hat{\beta}_0 + \hat{\beta}_1 x}}{1 + \exp^{\hat{\beta}_0 + \hat{\beta}_1 x}}$$

Error Based Learning

- The residual, or error is the difference between our predicted value and the actual value for some input *x*
- Given a training set, how do we pick, or learn, the parameters β?
- One method is to make h(*x*) close to *y*, for the training examples we have.
- For linear regression, we can find the parameters β which minimise the sum of the squared residuals: SSE:



$$\text{SSE} = \sum_{i=1}^{n} \epsilon_i^2$$

Error Based Learning

- This is a specific example of a more general approach to finding the best parameters for a model.
- We start by defining a loss function, or error function

$$J(\beta) = \frac{1}{2n}\sum_{i=1}^{n}(h_\beta(x^{(i)}) - y^{(i)})^2$$

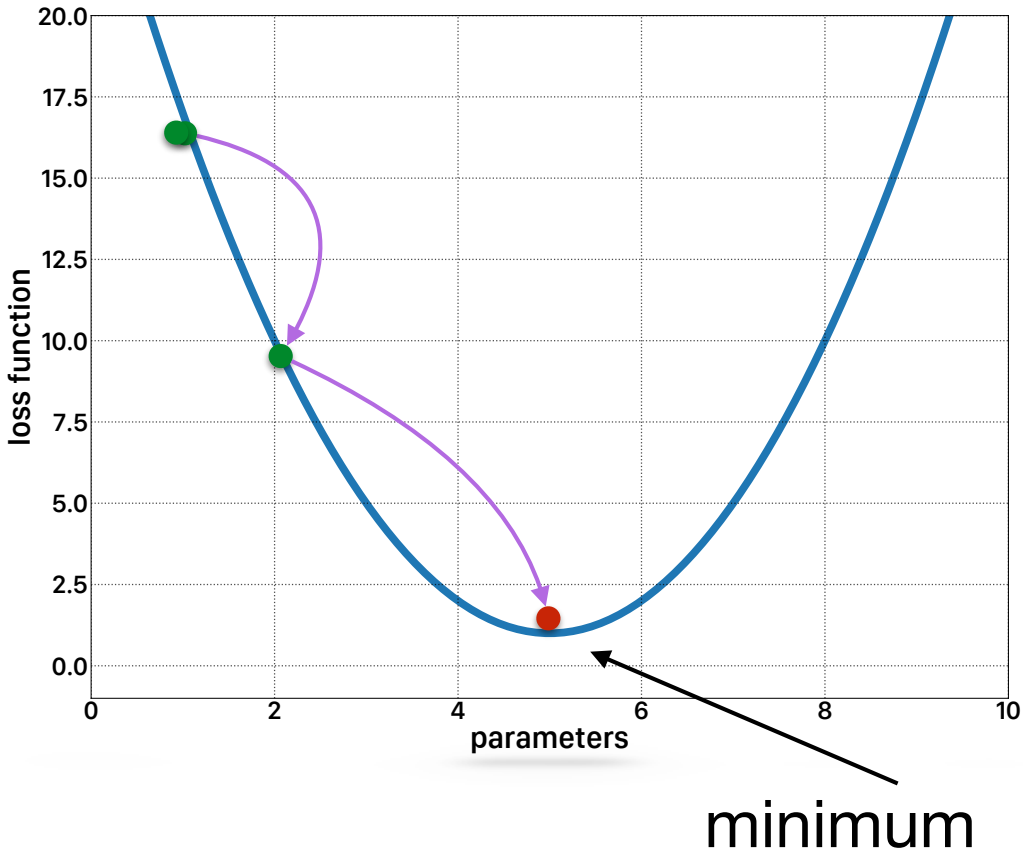
$$h(x) = \beta_0 + \beta_1 x$$

$$h(x) = \frac{\exp^{\hat{\beta}_0+\hat{\beta}_1 x}}{1 + \exp^{\hat{\beta}_0+\hat{\beta}_1 x}}$$

- The task is now to find the set of parameters β which minimise this loss function.
- This is a process of minimising the distance between our hypothesis h(*x*) and the data *y*.

Gradient Descent

- Gradient descent is an algorithm that makes small steps along a function to find a local minimum.
- We start at some point and find the gradient (slope)
- We take a step in the opposite direction to the gradient (ie. downhill)
- The size of the step is controlled by an adjustable parameter
- This algorithm gets us closer and closer to the local minimum.



In a 3D space, it would be like rolling a ball down a hill to find the lowest point

Gradient Descent

- How to do the update:

$$\beta_j^{(i+1)} := \beta_j^{(i)} - \alpha \frac{\partial}{\partial \beta_j} J(\beta^{(i)})$$

- we start with some initial value of the parameters (could be randomly chosen or a reasonable first guess)
- the we compute the gradient of the loss function with respect to that parameter
- then adjust the parameter by a small amount (controlled by α), the opposite direction to the gradient (minus sign).
- By adjusting α, we can change how quickly we converge to the minimum. Large α -> risk of overshooting the minimum, small α -> might not converge on the local minimum.

Gradient Descent - Linear Regression

- for each parameter, we can compute the gradient and we find the update step is given by:

$$\begin{aligned}\beta_0 :=& \beta_0 - \alpha \frac{\partial}{\partial \beta_0} J(\beta) \\ =& \beta_0 - \alpha \frac{\partial}{\partial \beta_0} \frac{1}{2n} \sum_{i=1}^{n} (h_\beta(x^{(i)}) - y^{(i)})^2 \\ =& \beta_0 - \frac{\alpha}{n} \sum_{i=1}^{n} (\beta_0 + \beta_1 x^{(i)} - y^{(i)})\end{aligned}$$

$$\begin{aligned}\beta_1 :=& \beta_1 - \alpha \frac{\partial}{\partial \beta_1} J(\beta) \\ =& \beta_1 - \alpha \frac{\partial}{\partial \beta_1} \frac{1}{2n} \sum_{i=1}^{n} (h_\beta(x^{(i)}) - y^{(i)})^2 \\ =& \beta_1 - \frac{\alpha}{n} \sum_{i=1}^{n} (\beta_0 + \beta_1 x^{(i)} - y^{(i)}) x^{(i)}\end{aligned}$$

factor of x here!

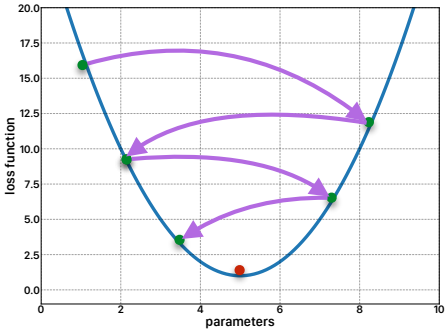
The update step for logistic regression is found by similar steps- using the derivative of the logistic function:

$$\frac{\partial}{\partial P} Logistic(P) = Logistic(P)(1 - Logistic(P))$$

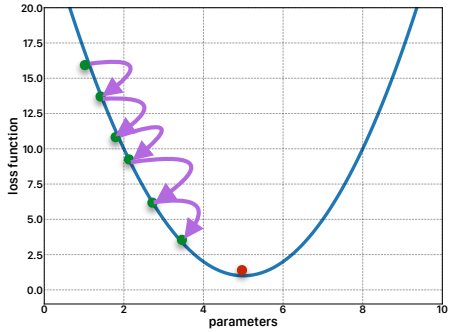
Gradient Descent

- The learning rate, α, determines the size of the adjustment made to each weight at each step in the process.
- What is the best value of α to choose?
  - There are many strategies to choosing the best value of α.
  - Most of the time we use rules of thumb, and also trial and error.
  - A typical range for α is [0.00001, 10]
- Similarly for the weights, we might start with initial values $[\beta_0, \beta_1] = [-0.2, 0.2]$

steps too large



steps too small

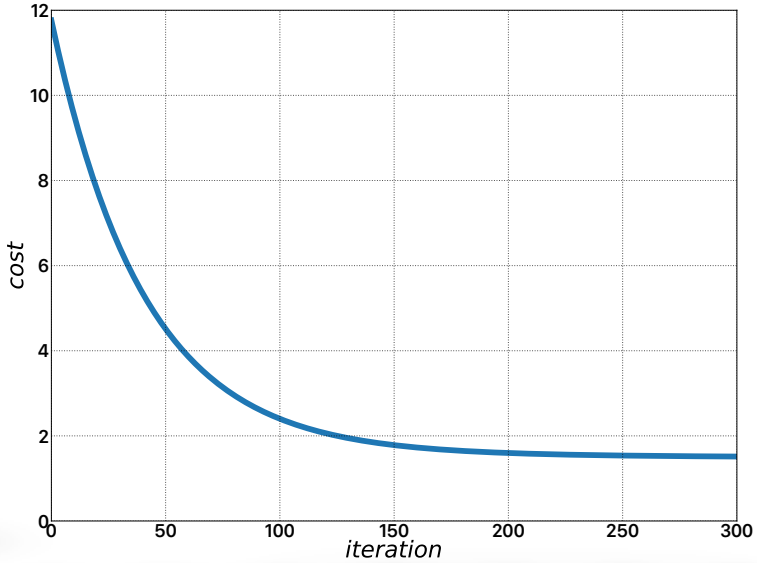


Gradient Descent

A basic algorithm for gradient descent:

1: **function** GRADIENTDESCENT(x, α, k_{max})
2: Require: Set of training instances D
3: Require: learning rate α
4: Require: Maximum number of iterations k_{max} or
5: other convergence criterion
6: $\beta_0 \leftarrow$ random point parameter space
7: $\beta_1 \leftarrow$ random point parameter space
8: **for** $k = 0$ to k_{max} **do**
9: $\beta_0^* = \beta_0 - \frac{\alpha}{n} \sum_{i=1}^{n} (h(x^{(i)}) - y^{(i)})$
10: $\beta_1^* = \beta_1 - \frac{\alpha}{n} \sum_{i=1}^{n} (h(x^{(i)}) - y^{(i)}) x^{(i)}$
11: $\beta_0 \leftarrow \beta_0^*$
12: $\beta_1 \leftarrow \beta_1^*$
13: **end for**
14: **end function**

updates are done after the sum!



We can compute the loss function, or cost at each iteration, and we see that it decreases monotonically.

Once the loss stops changing, we have done enough iterations.

A sequence of steps of the algorithm, β_0=0, β_1=0, and α=0.01

i	β_0	β_1	$\frac{\alpha}{n} \sum_{i=1}^{n} (h(x^{(i)}) - y^{(i)})$	$\frac{\alpha}{n} \sum_{i=1}^{n} (h(x^{(i)}) - y^{(i)}) x^{(i)}$	cost
0	0.045043	0.023356	-0.045043	-0.023356	11.762546
1	0.089526	0.046440	-0.044482	-0.023084	11.512937
2	0.133454	0.069254	-0.043928	-0.022815	11.269426
3	0.176834	0.091803	-0.043380	-0.022549	11.031864
4	0.219674	0.114089	-0.042840	-0.022286	10.800105
5	0.261979	0.136116	-0.042306	-0.022027	10.574008

Bike rental dataset: moodle link

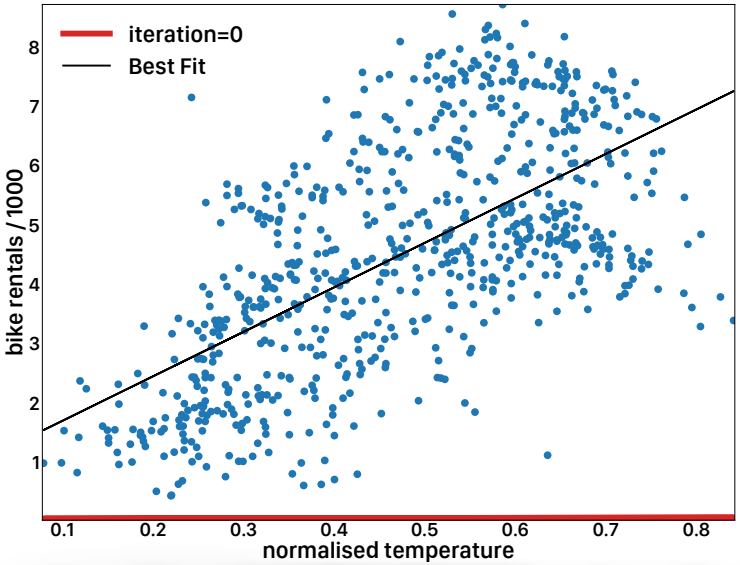
Gradient Descent

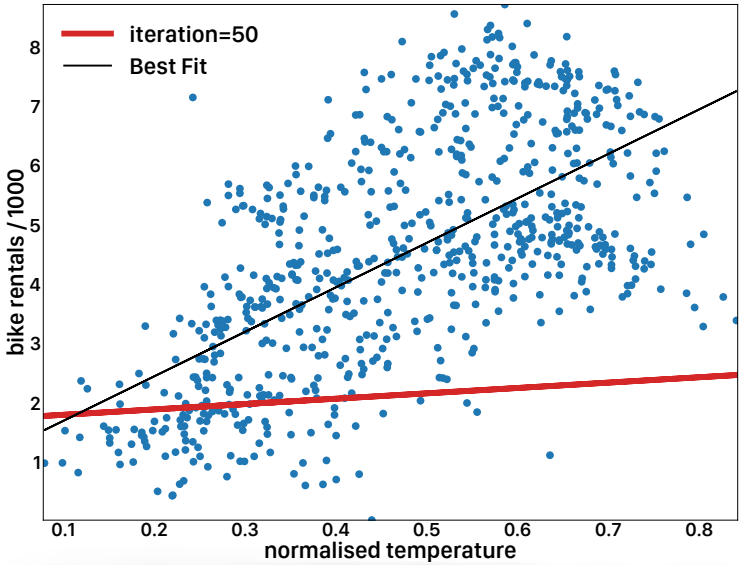
At the first step we have a poor guess for the parameters and our fit is bad.

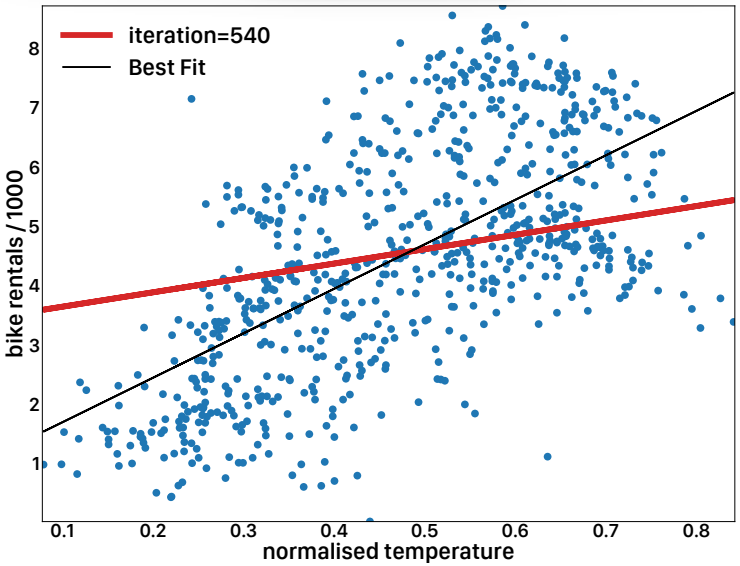
As we perform more iterations of gradient descent, the fit improves (quite quickly)

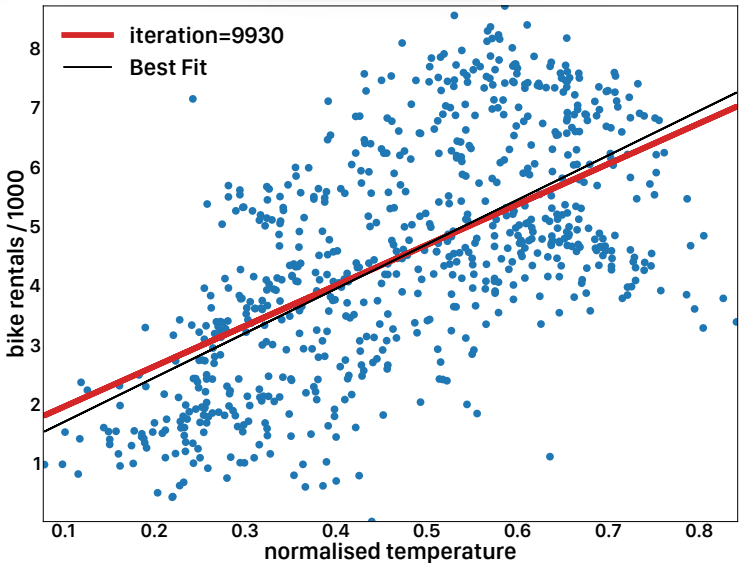
When to stop the iterations:

- a suitable condition might be when the change in the loss function is below some (small) threshold value









Simple Linear Regression

Classifier output

```
=== Run information ===

Scheme:       weka.classifiers.functions.LinearRegression -S 1 -C -R 1.0E-8 -additional-s
Relation:     bike_sharing-weka.filters.unsupervised.attribute.Remove-R1-10,12-15
Instances:    731
Attributes:   2
              atemp
              cnt
Test mode:    evaluate on training data

=== Classifier model (full training set) ===
```

bike rentals $= 7501.8339 \times temperature + 945.824$

```
cnt =

   7501.8339 * atemp +
    945.824
```

Gradient Descent

500 iterations

Classifier output

```
=== Run information ===

Scheme:       weka.classifiers.functions.SGD -F 2 -L 0.01 -R 1.0E-4 -E 500 -C 0.001 -N -S 1
Relation:     bike_sharing-weka.filters.unsupervised.attribute.Remove-R1-10,12-15
Instances:    731
Attributes:   2
              atemp
              cnt
```

bike rentals $= 7495.5297 \times temperature + 1020.4324$

```
Loss function: Squared loss (linear regression)

cnt =

      7495.5297 atemp
 +    1020.4324
```

Gradient Descent

- This method is called **“batch” gradient descent** because we use the entire **batch** of points x to calculate each gradient.
- Stochastic gradient descent uses a sample of points at each step and is faster for large dataset (we do not consider it here), since we don’t need to iterate over all examples
- GD is a general learning algorithm:
  - define a loss function
  - start with some initial set of parameters
  - repeat until convergence:
    - update the parameters in the opposite direction to the gradient

Summary

- We looked at logistic regression models for categorical features
- We looked at Gradient Descent to learn the parameters of a simple linear regression model.
- Gradient Descent is an optimisation algorithm used to find the values of parameters of a function that minimises a cost function.
- Gradient descent is best used when the parameters cannot be calculated analytically, and must be searched for by an optimisation algorithm (eg. logistic regression)
- Gradient Descent is used in many areas of Machine Learning.

Data Science for Business
Model Performance Analytics

Asst. Prof. Teerapong Leelanupab (Ph.D.)
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang (KMITL)

Week 9.2

Overfitting the data

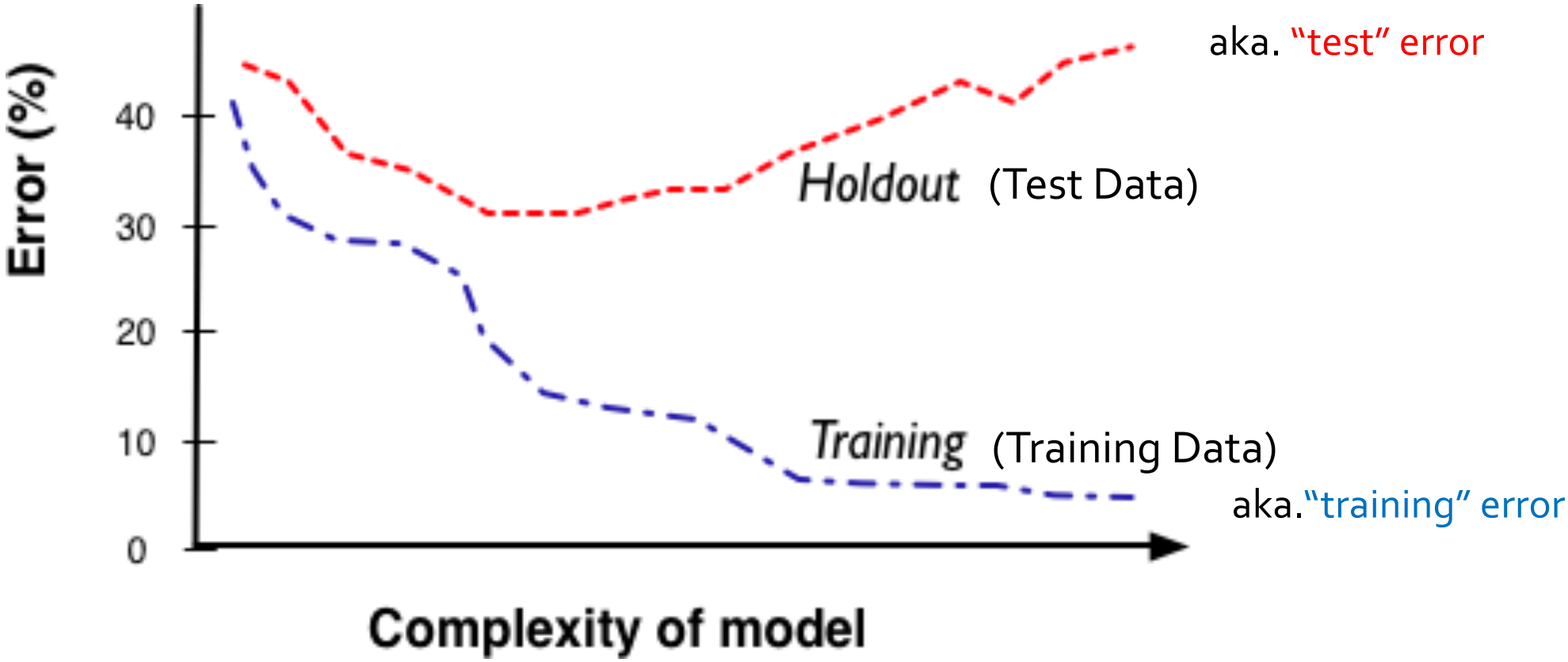
- Finding chance occurrences in data that look like interesting patterns, but which do not generalize, is called overfitting the data
- We want models to apply not just to the exact training set but to the general population from which the training data came
    - Generalization

Overfitting

- The tendency of DM procedures to tailor models to the training data, *at the expense of generalization* to previously unseen data points.
- All data mining procedures have the tendency to over-fit to some extent
    - Some more than others.
- "If you torture the data long enough, it will confess"
- There is no single choice or procedure that will eliminate over-fitting
    - recognize overfitting and manage complexity in a principled way.

Fitting Graph to estimate *generalization performance*

Error is simply an inversion of accuracy.

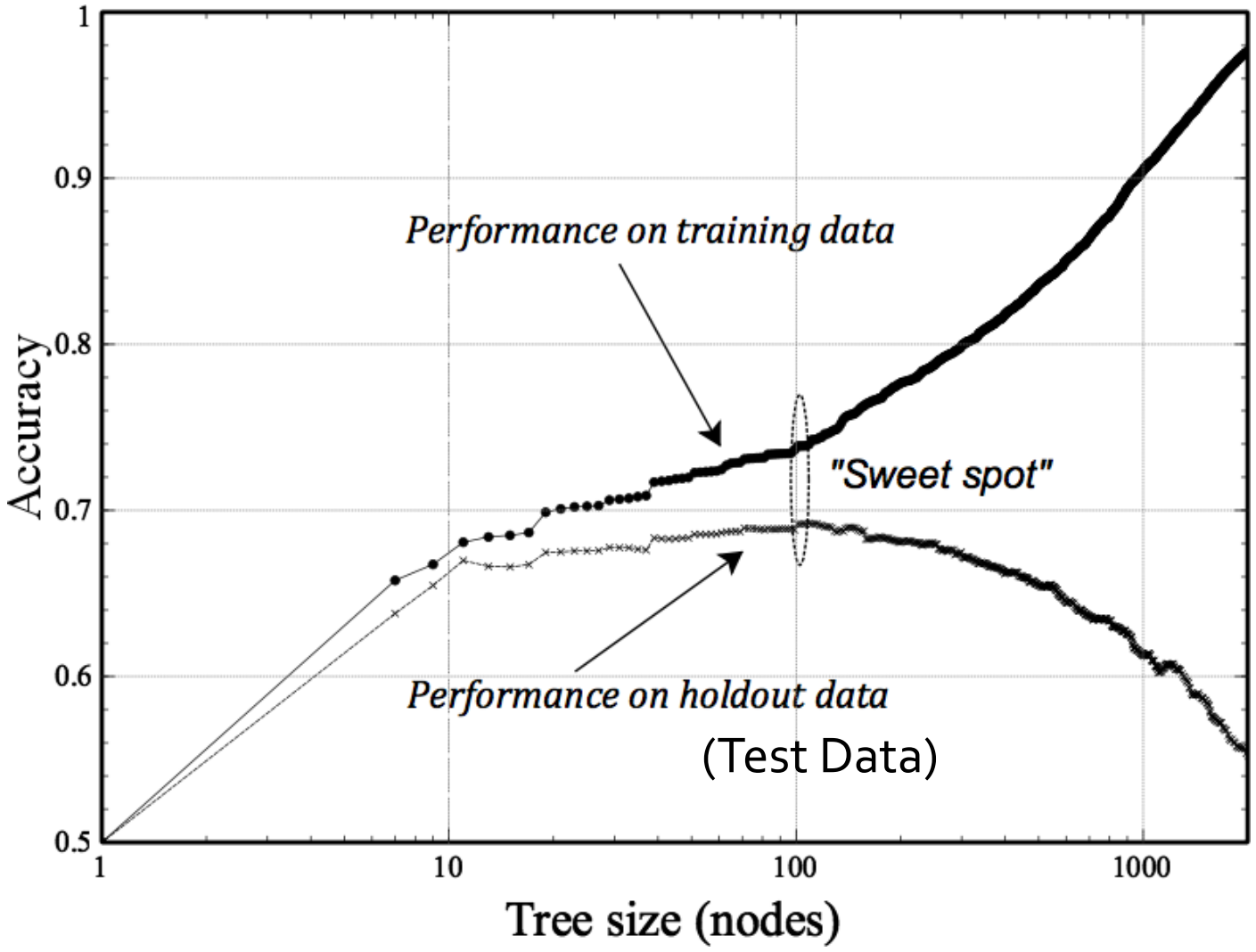


aka. "test" error

aka."training" error

We will later use these two error metrics to calculate **Bias** and **Variance**

Overfitting in tree induction



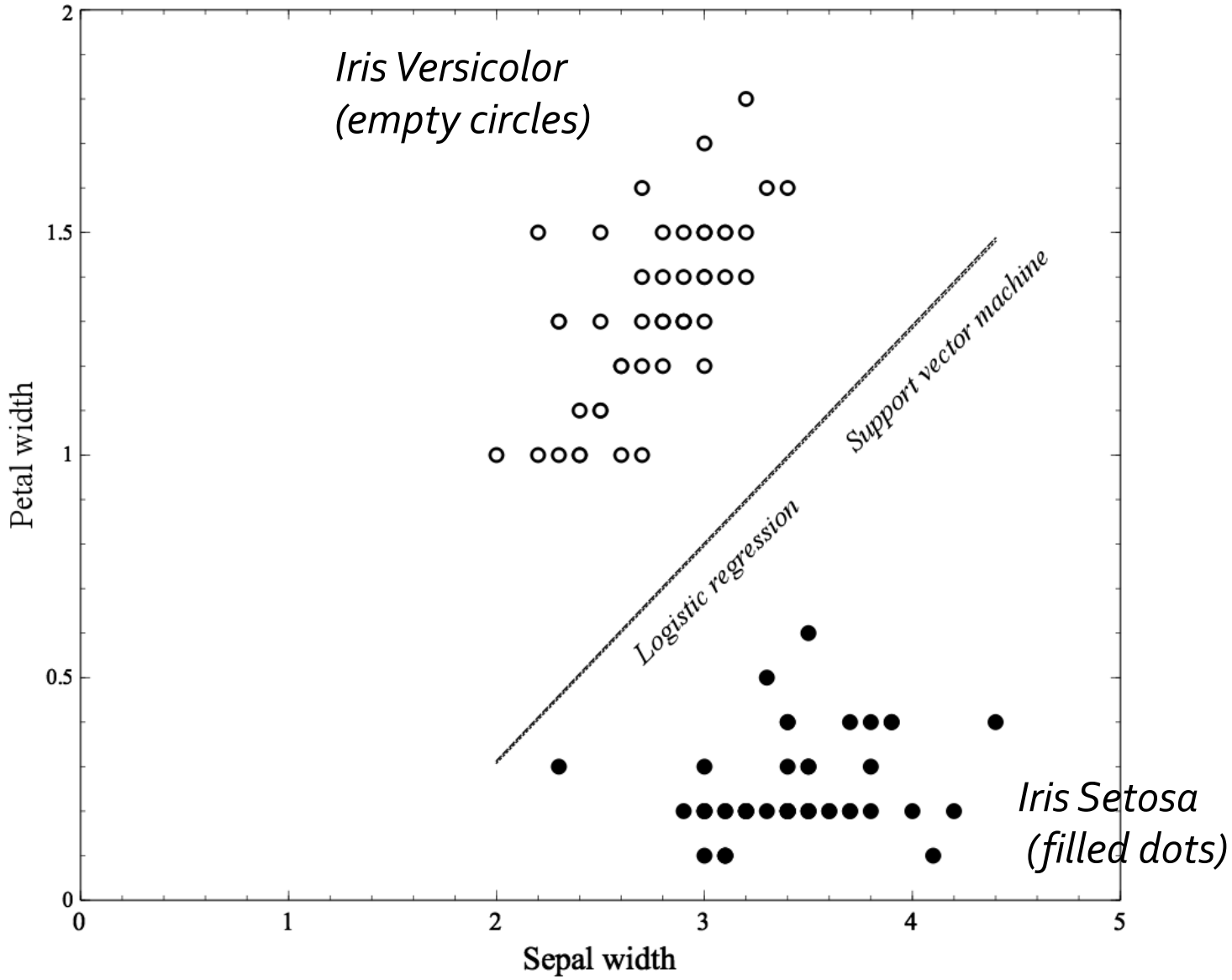
straightforward called "training" accuracy or also called "in-sample" accuracy

straightforward called "test" accuracy, or also called "holdout", "out-of-sample" or "generalization" accuracy

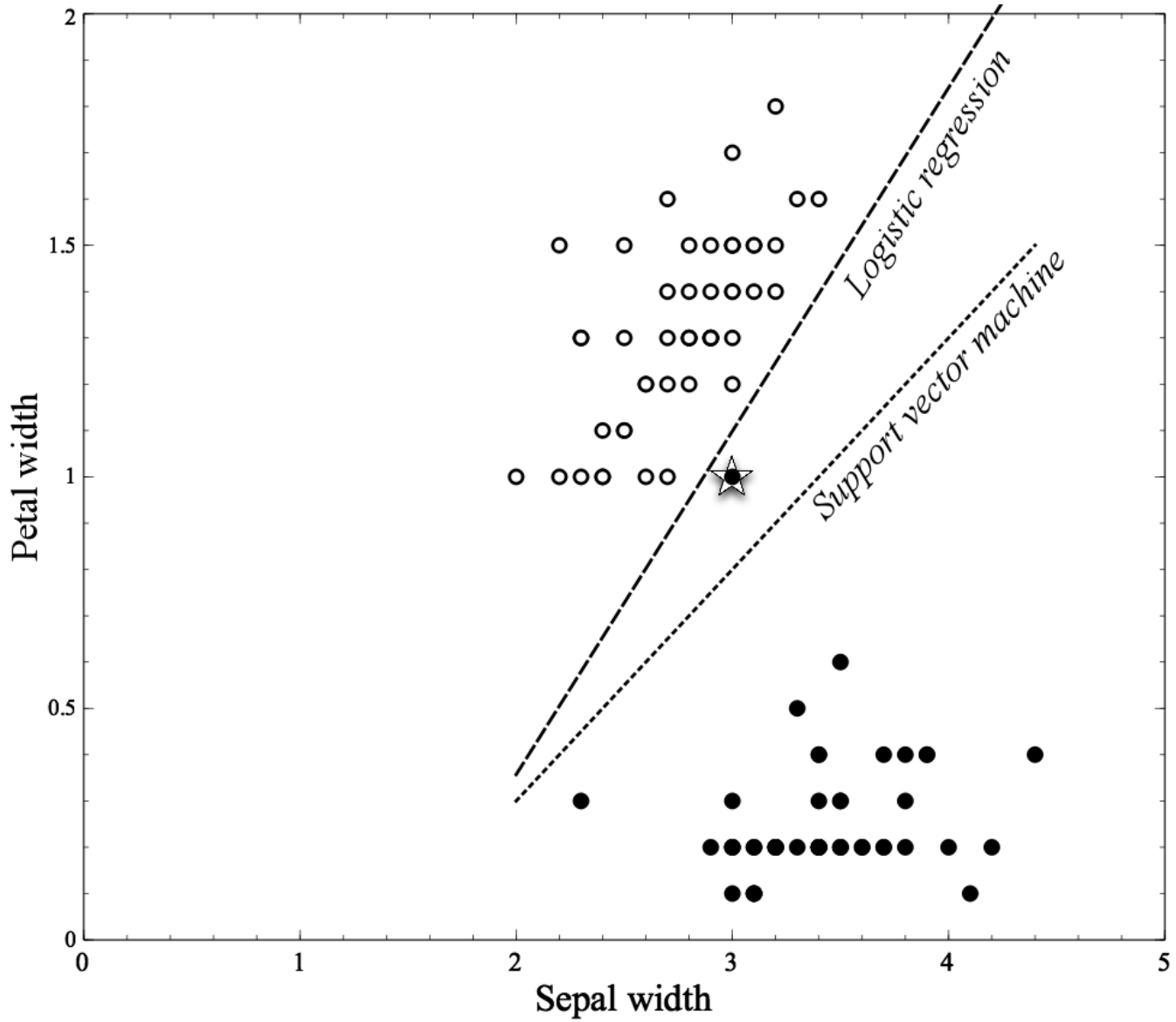
Overfitting in linear discriminants

- $f(x) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$
- $f(x) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5$
- $f(x) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 + w_6 x_1^2$
- $f(x) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 + w_6 x_1^2 + w_7 * x_2 / x_3$

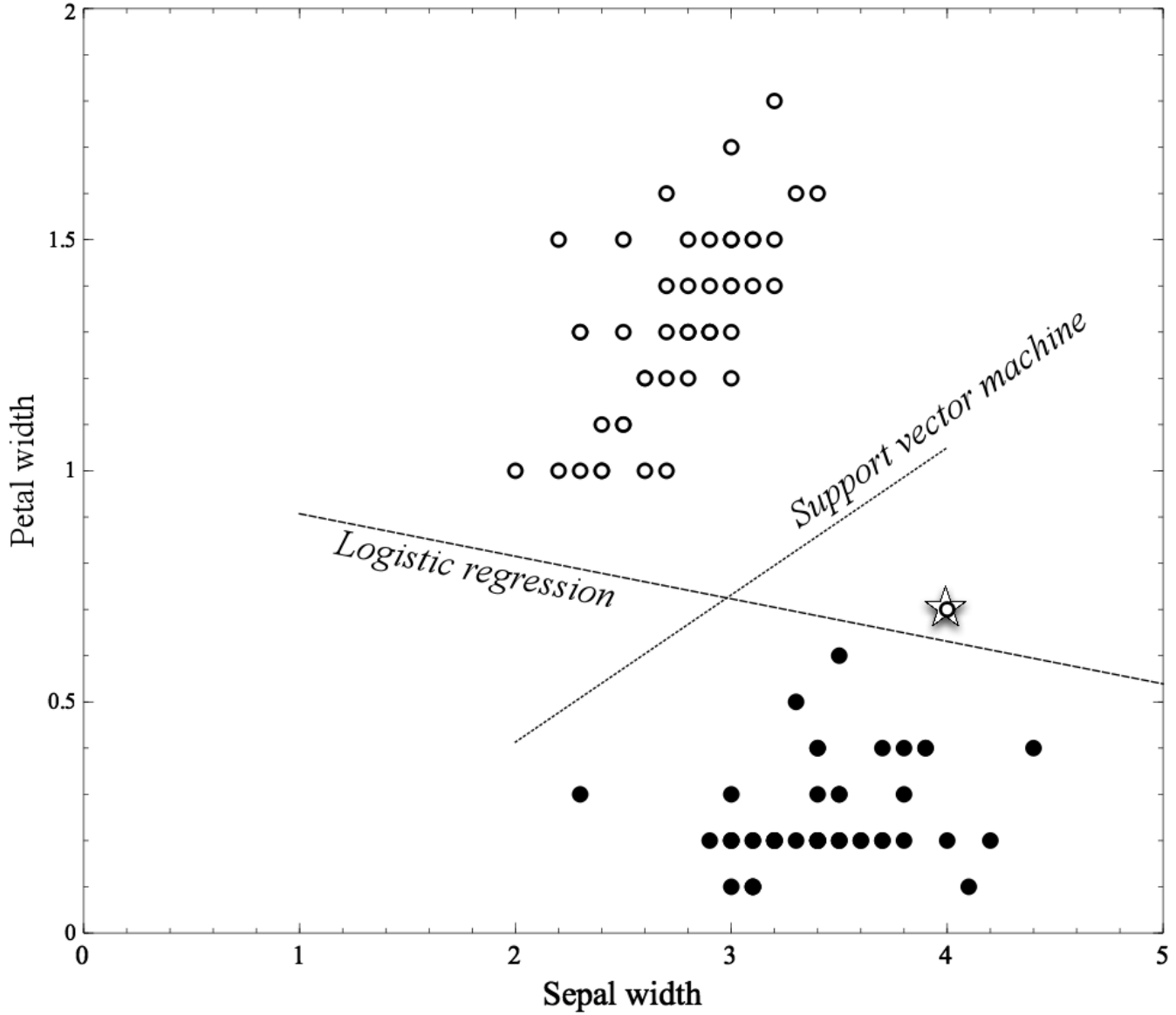
Example: Classifying Flowers



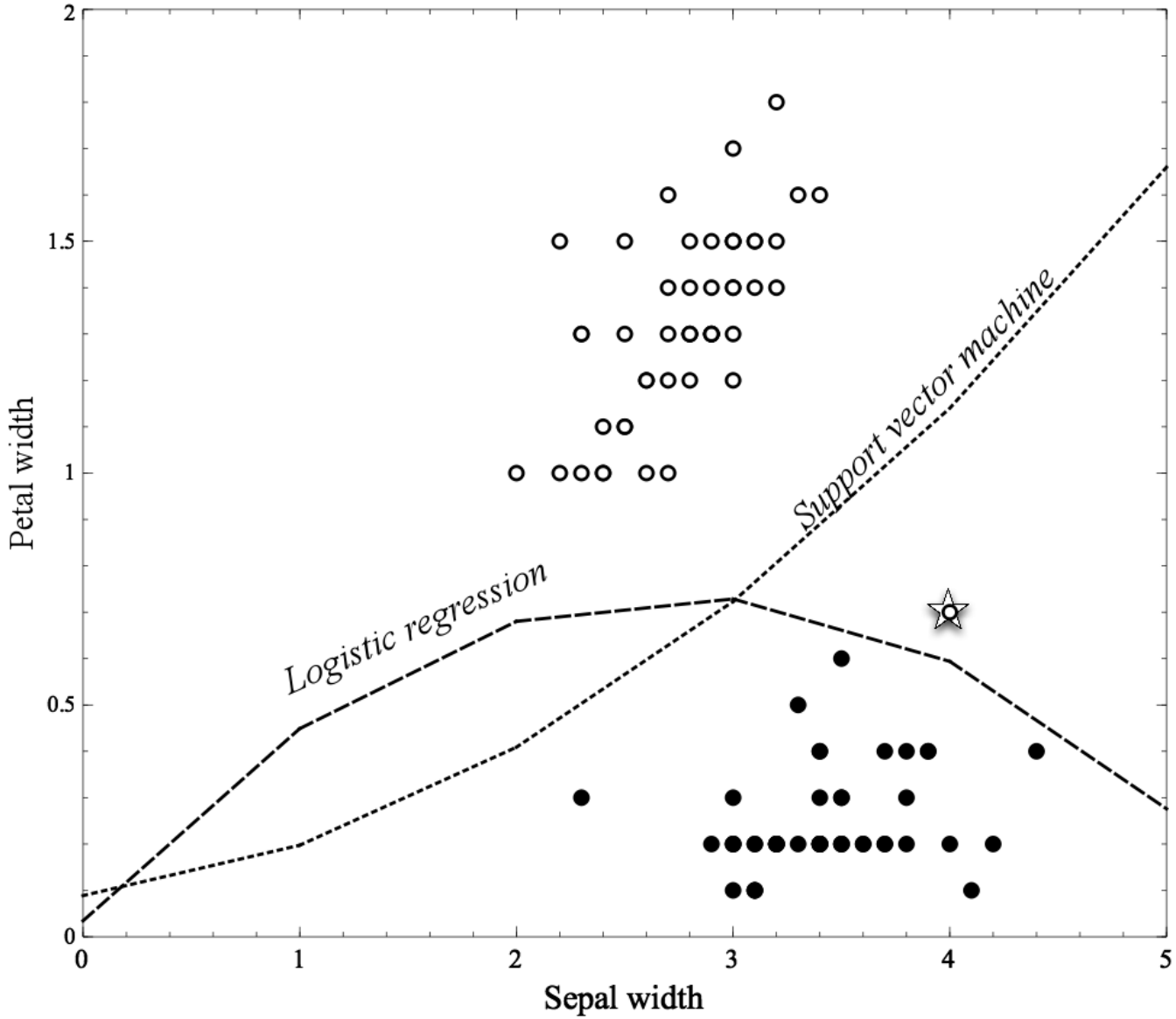
Example: Classifying Flowers



Example: Classifying Flowers

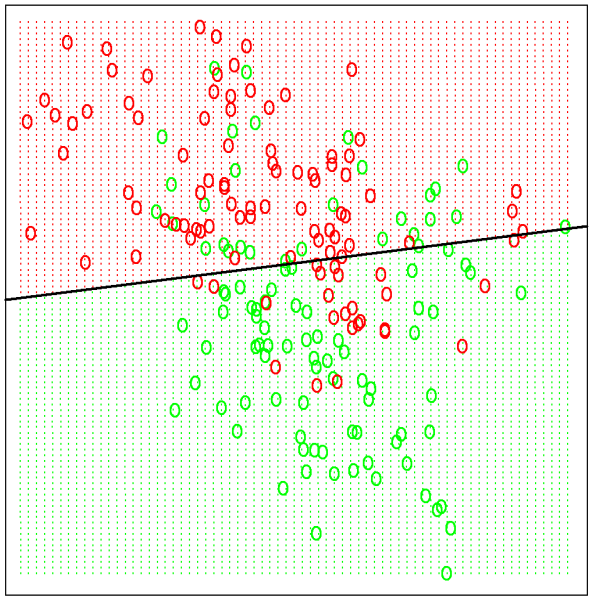


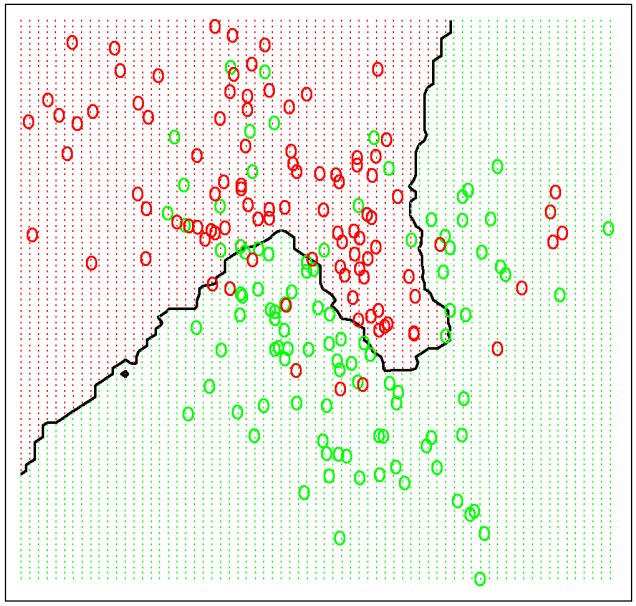
Example: Classifying Flowers

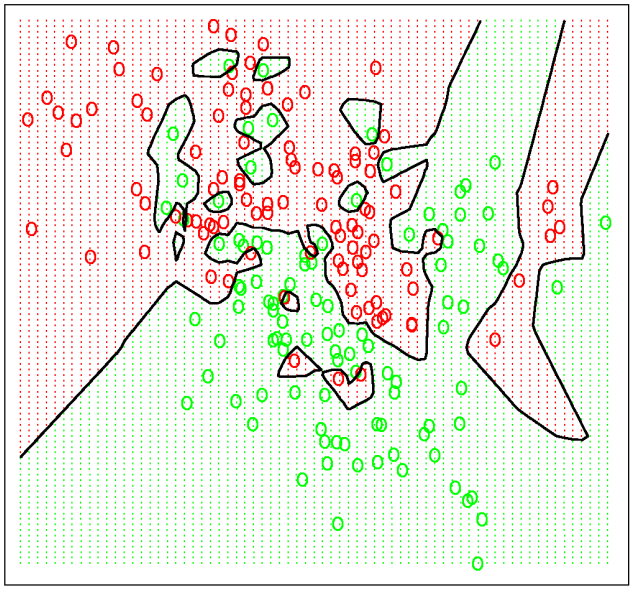


By adding more variables, our mathematical functions can become more complex.

Need for holdout evaluation



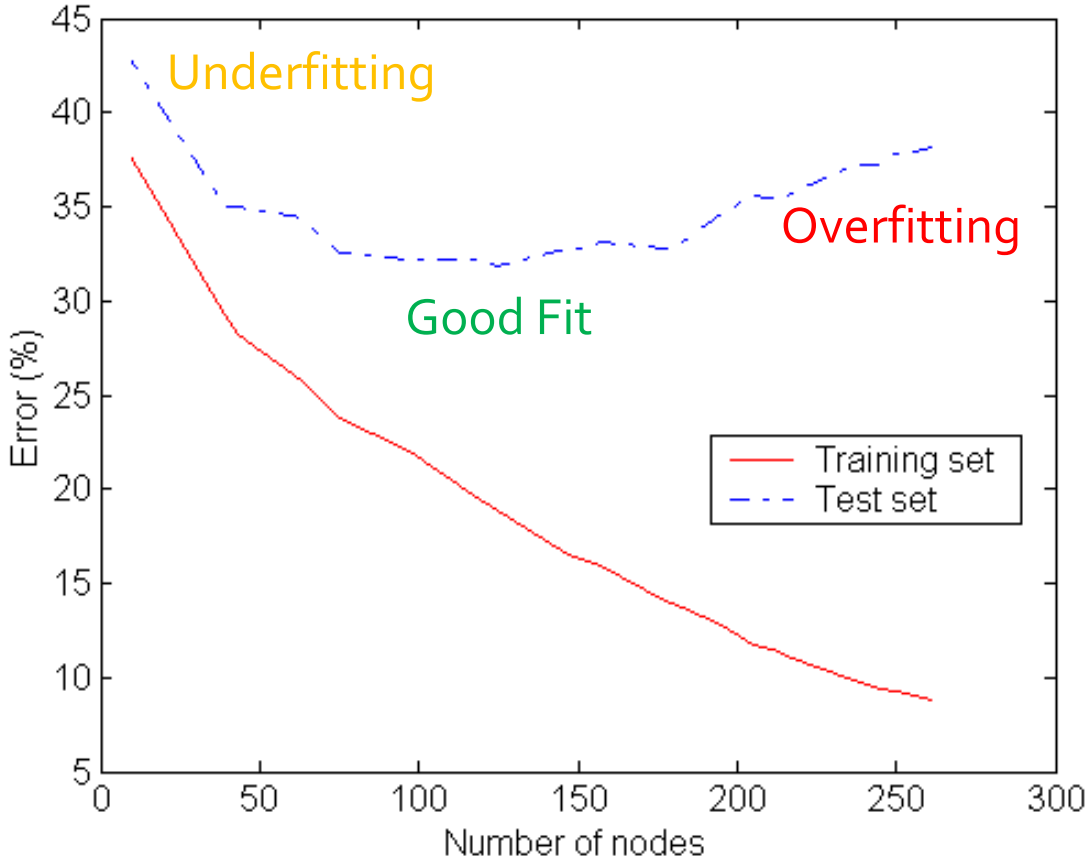




Underfitting Good Overfitting

- In sample evaluation is in favor or "memorizing"
- On the *training data* the right model would be best
- But on *new data* it would be bad

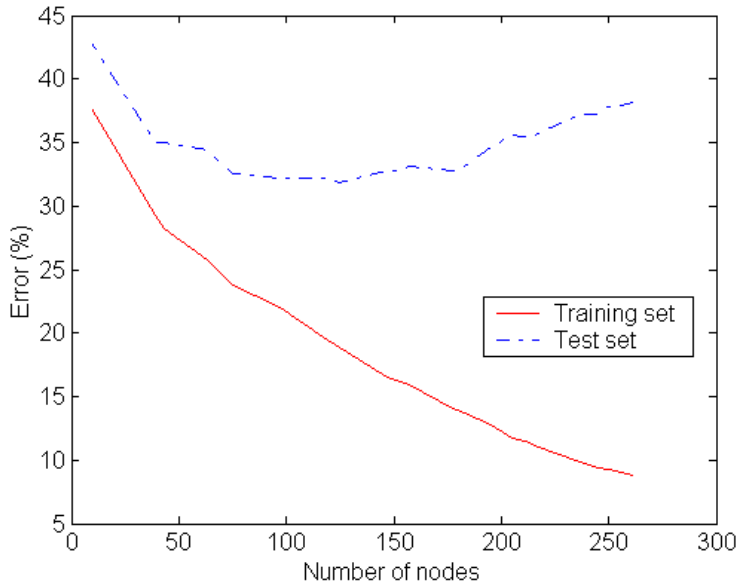
Overfitting



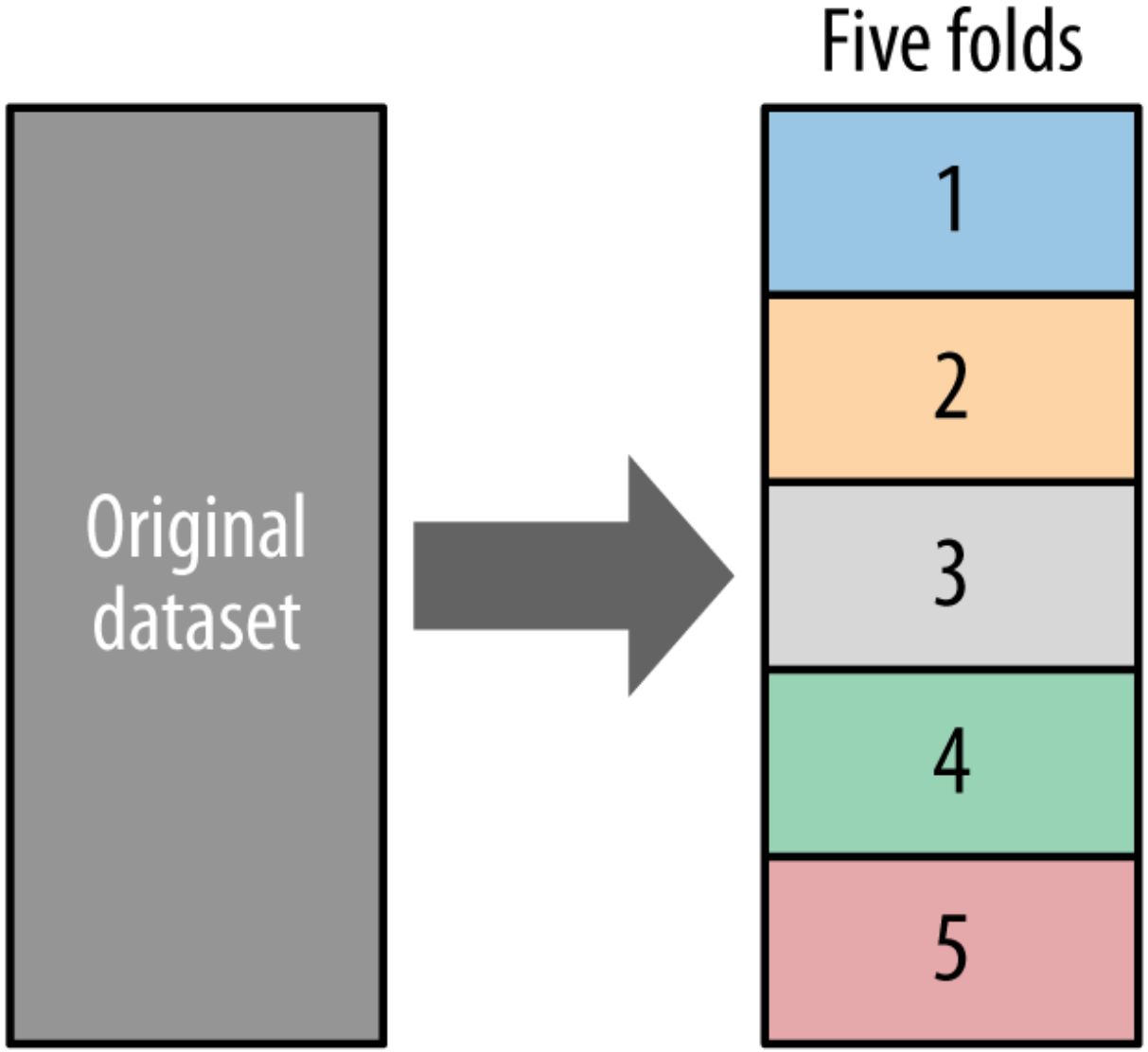
- Overfitting: Model "memorizes" the properties of the particular training set rather than learning the underlying concept or phenomenon

Holdout validation

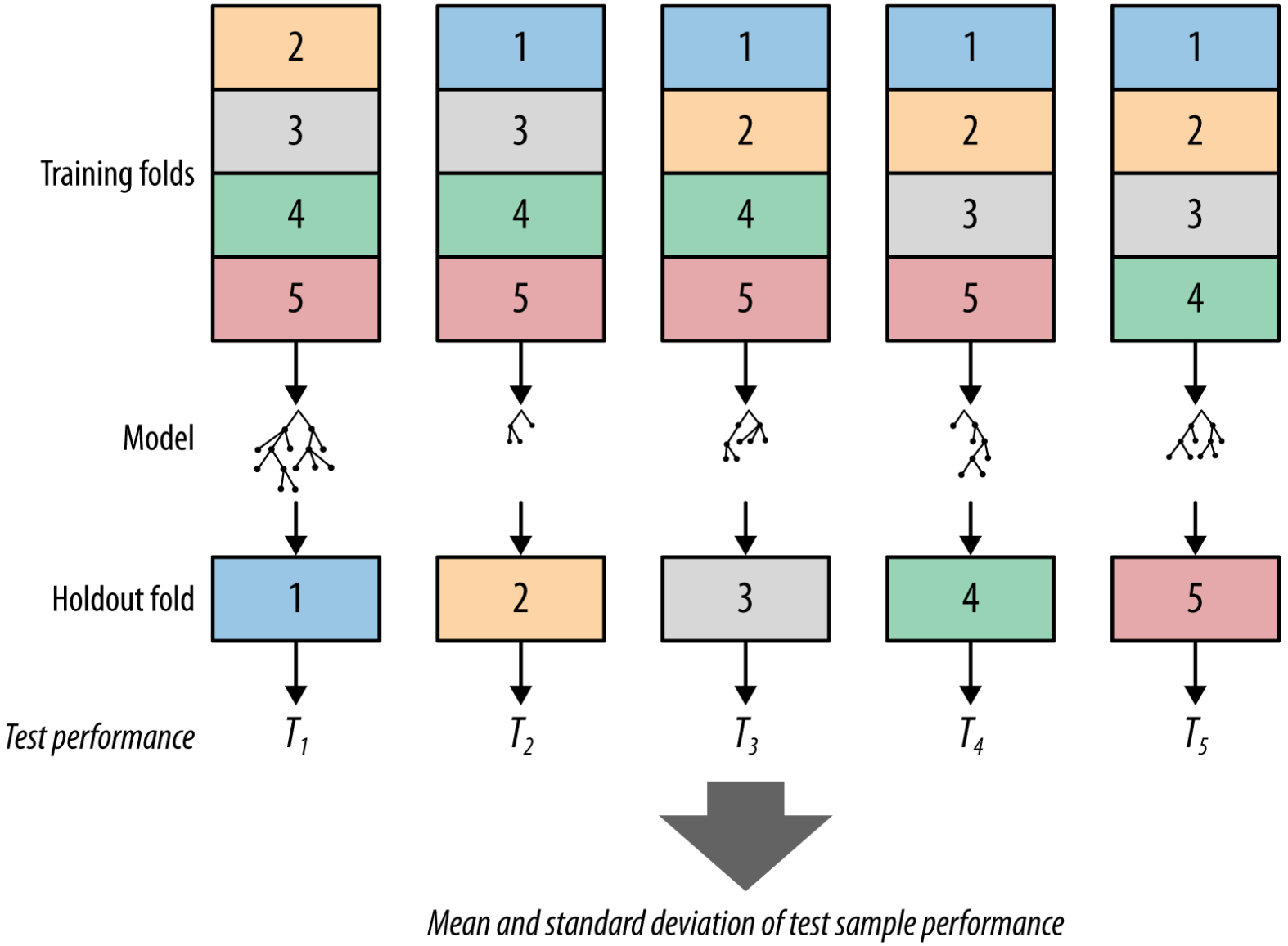
- We are interested in generalization
  - The performance on data not used for training
- Given only one data set, we hold out some data for evaluation
  - Holdout set for final evaluation is called the test set
- Accuracy on training data is sometimes called **"in-sample" accuracy**, vs. **"out-of-sample" accuracy** on test data



Cross-Validation



Cross-Validation

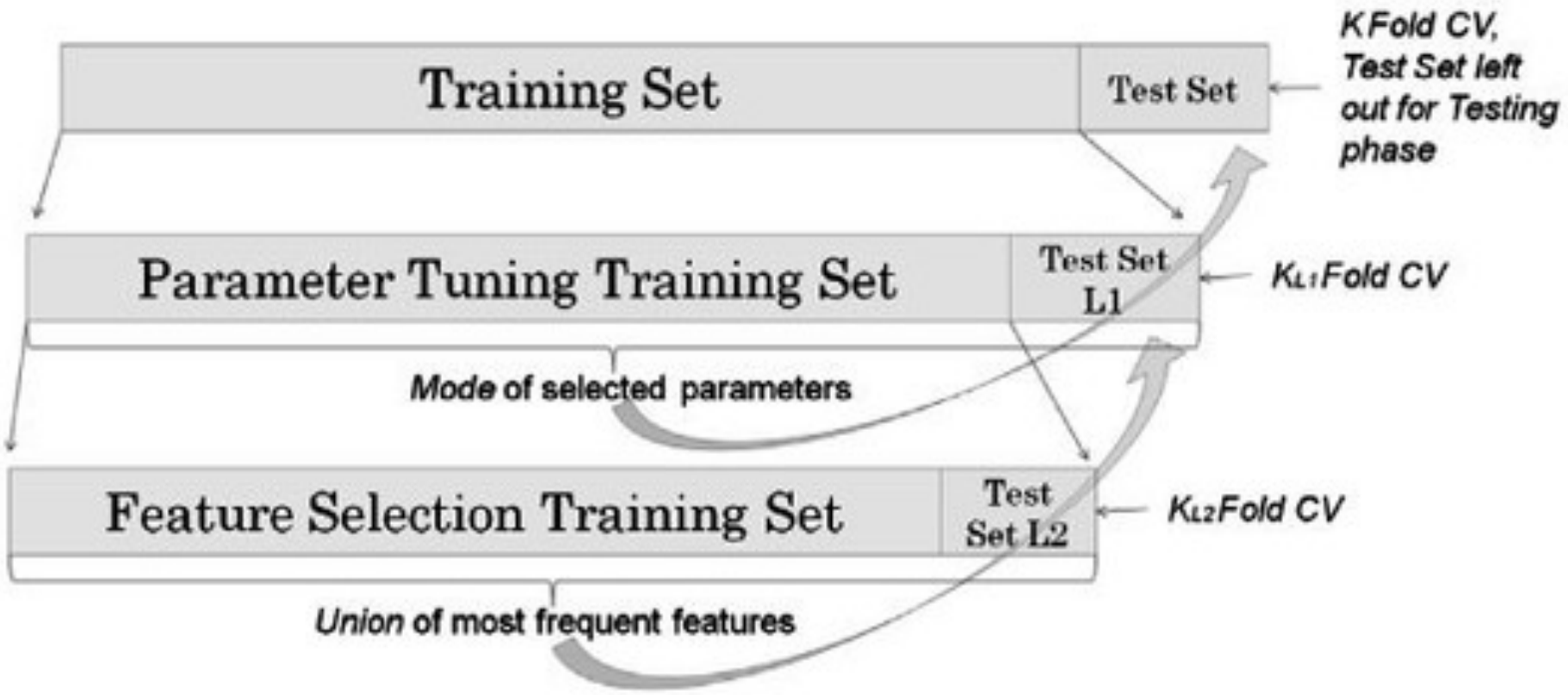


We will come back on more details about a variation of dataset splitting in the next lecture.

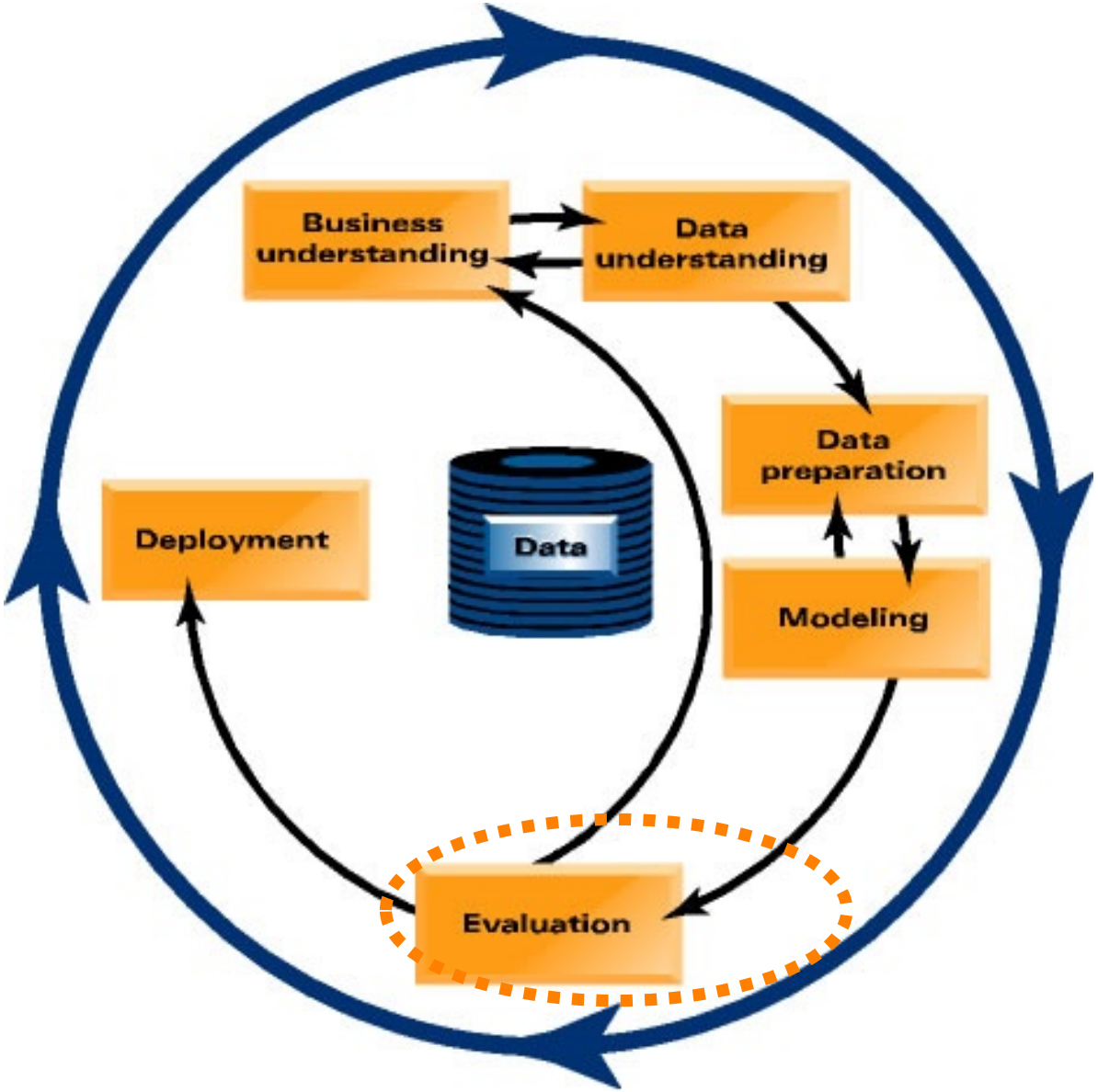
From Holdout Evaluation to Cross-Validation

- Not only a simple estimate of the generalization performance, but also some statistics on the estimated performance,
    - such as the mean and variance
- Better use of a limited dataset
    - Cross-validation computes its estimates over *all* the data

Nested Cross-Validation

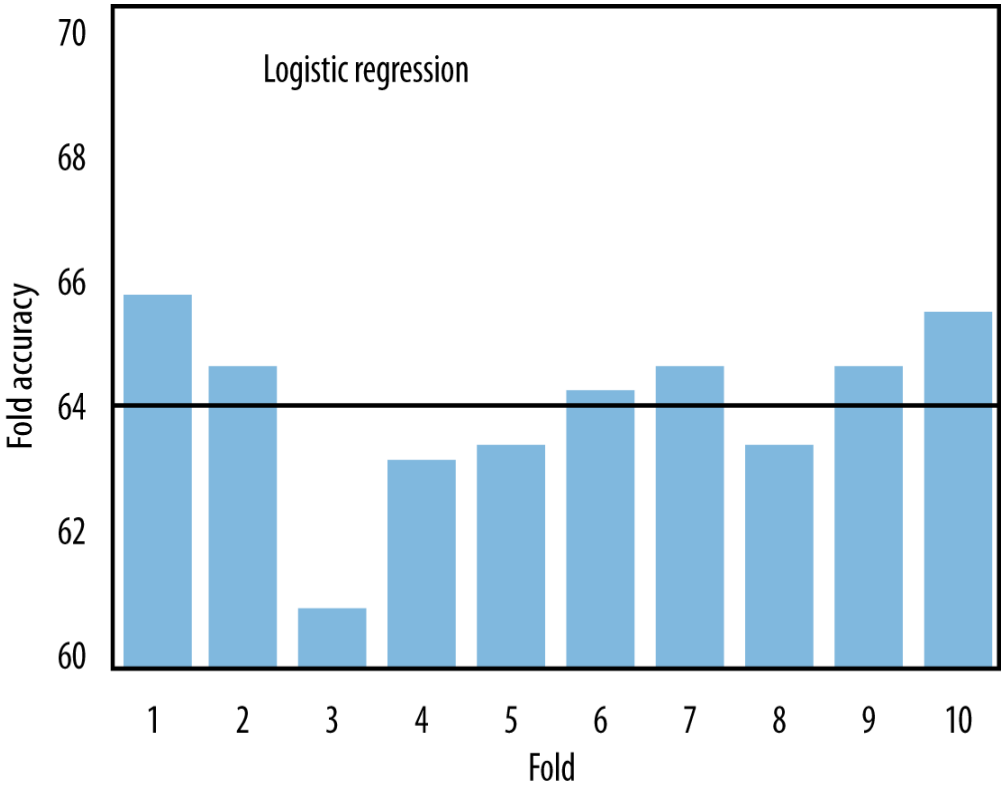


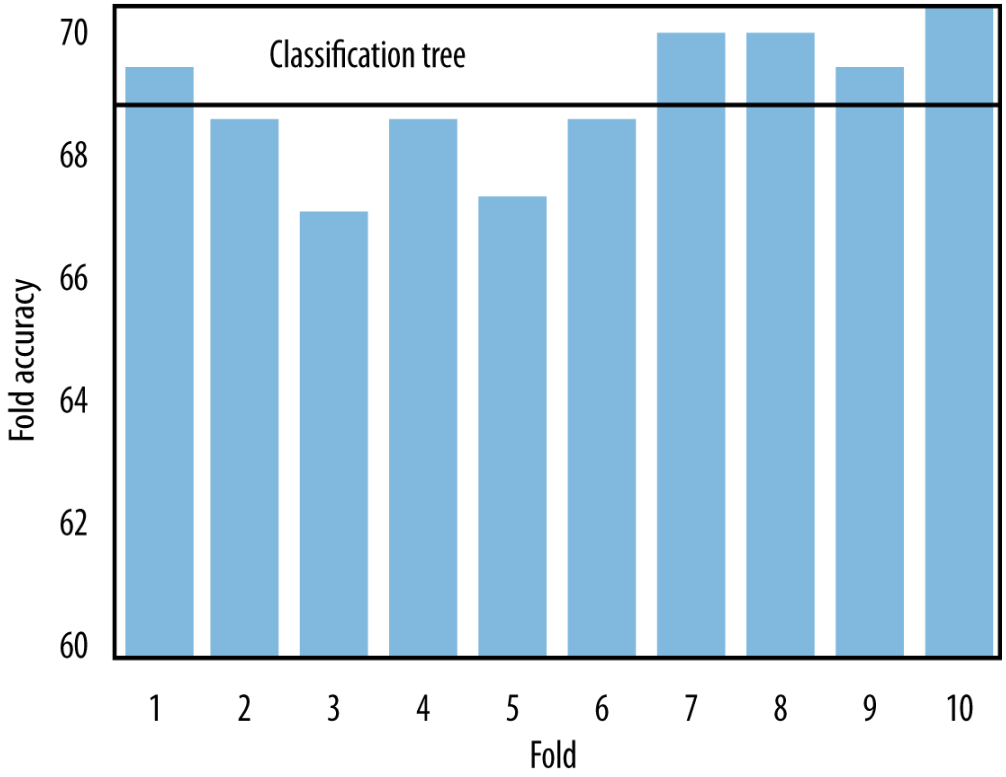
Let's focus back in on actually mining the data..



Which customers should TelCo target with a special offer, prior to contract expiration?

MegaTelCo


Logistic regression
Fold accuracy
70
68
66
64
62
60
1
2
3
4
5
6
7
8
9
10
Fold

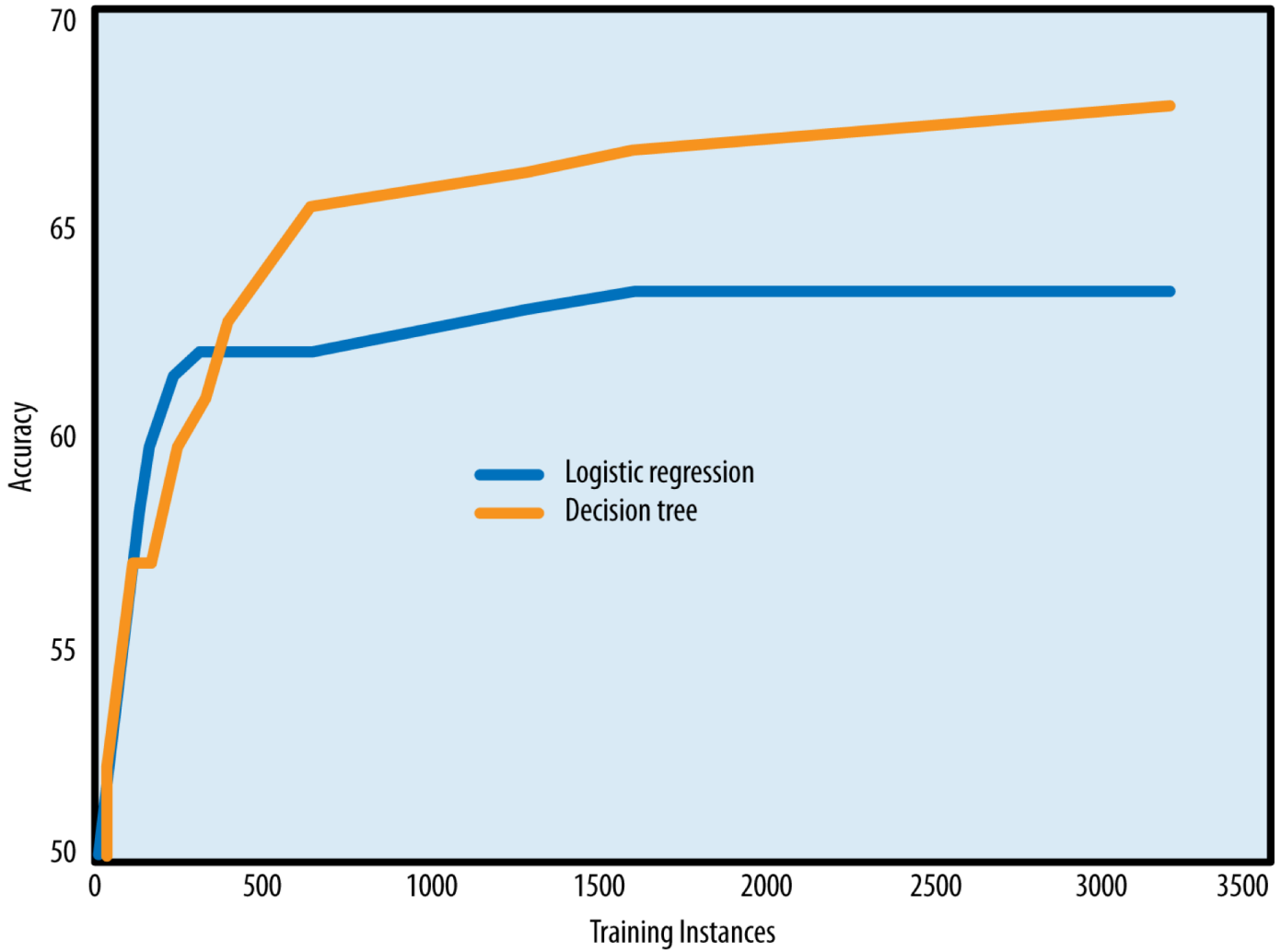

Classification tree
Fold accuracy
70
68
66
64
62
60
1
2
3
4
5
6
7
8
9
10
Fold

Generalization Performance

- Different modeling procedures may have different performance on the same data
- Different training sets may result in different generalization performance
- Different test sets may result in different estimates of the generation performance
- If the training set size changes, you may also expect different generalization performance from the resultant model

Learning Curves

Test Acc.

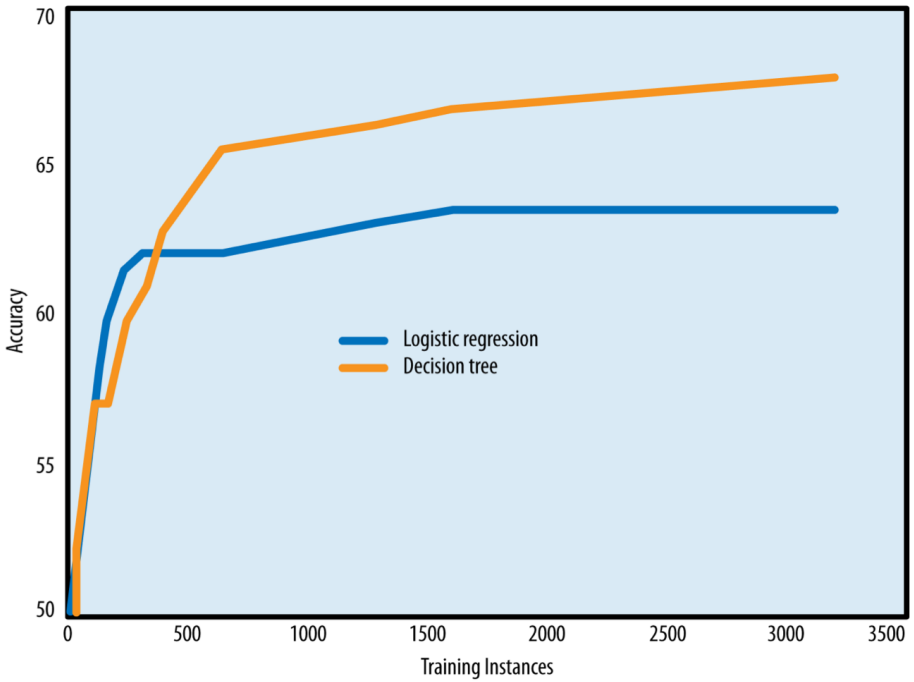


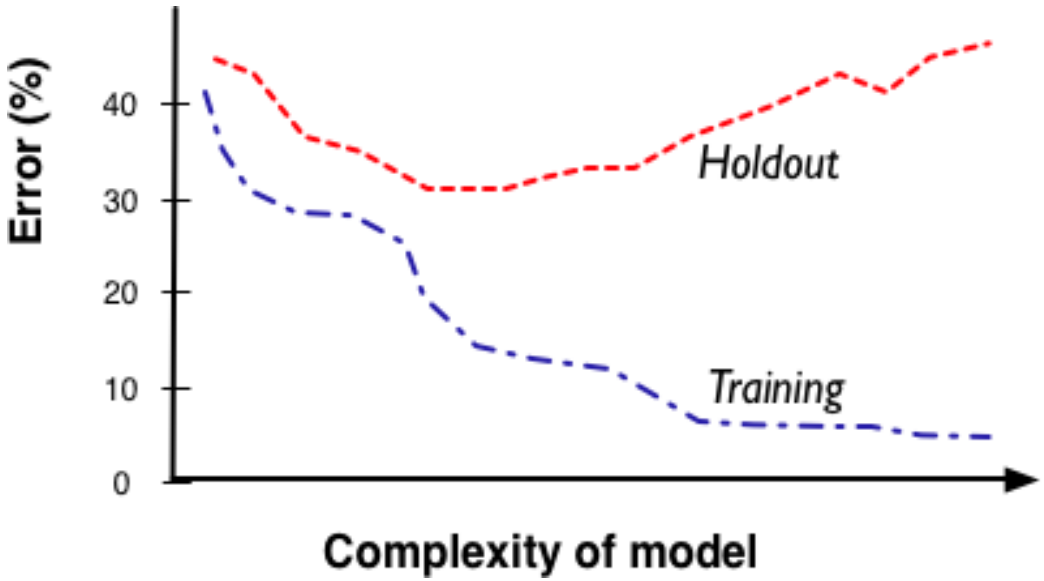
The change in the size of training set affects the generalization performance of the resultant model

Logistic Regression vs Tree Induction

- For smaller training-set sizes, logistic regression yields better generalization accuracy than tree induction
    - For smaller data, tree induction will tend to over-fit more
- Classification trees are a more flexible model representation than linear logistic regression
- Flexibility of tree induction can be an advantage with larger training sets:
    - Trees can represent substantially nonlinear relationships between the features and the target

Learning curves vs Fitting graphs





- A learning curve shows the generalization performance plotted against the amount of training data used.
- A fitting graph (or fitting curve) shows the generalization performance as well as the performance on the training data, but plotted against model complexity.
- Fitting graphs generally are shown or a fixed amount of training data.

Avoiding Overfitting

- Tree Induction:
- Post-pruning
    - takes a fully-grown decision tree and discards unreliable parts
- Pre-pruning
    - stops growing a branch when information becomes unreliable

Linear Models:

- Feature Selection
- Regularization
    - Optimize some combination of fit and simplicity

Regularization

- Regularized linear model:

- $\underset{\boldsymbol{w}}{\text{argmax}}[\text{fit}(\boldsymbol{x}, \boldsymbol{w}) - \lambda * \text{penalty}(\boldsymbol{w})]$
- Different Penalties can applied, such as
  - "L2-norm" (Euclidean distance from origin)
    - The square root of the sum of the *squares* of the weights
    - L2-norm + standard least-squares linear regression = ridge regression
  - "L1-norm" (Manhattan/taxicab distance from origin)
    - The sum of the *absolute values* of the weights
    - L1-norm + standard least-squares linear regression = lasso
    - Automatic feature selection

Regularization is trying to optimize not just the fit to the data, but a combination of fit to the data and simplicity of the model (The latter is conveyed by penalty.)

Data Science for Business
Evaluation in Machine Learning – Part 1

Asst. Prof. Teerapong Leelanupab (Ph.D.)
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang (KMITL)

Week 9.3

Overview

- Part 1
  - Objectives of Evaluation
  - A/B Testing
  - Basic Statistics Reminder
  - Statistical Significance
    - Student’s t-test
    - Tests for proportions

Objectives of Evaluation

Q. Is machine learning algorithm *A* better than algorithm *B*?

- **Supervised Learning**
  - Does classifier *A* have better accuracy than *B* on a given dataset?
  - Does classifier *A* have better accuracy across many different datasets?
  - What is the difference in *generalisation performance* on new data not seen in training?
- **Unsupervised Learning**
  - Does clustering algorithm X provide more useful or interpretable results than algorithm Y?

Q. Is the difference between the results statistically significant?

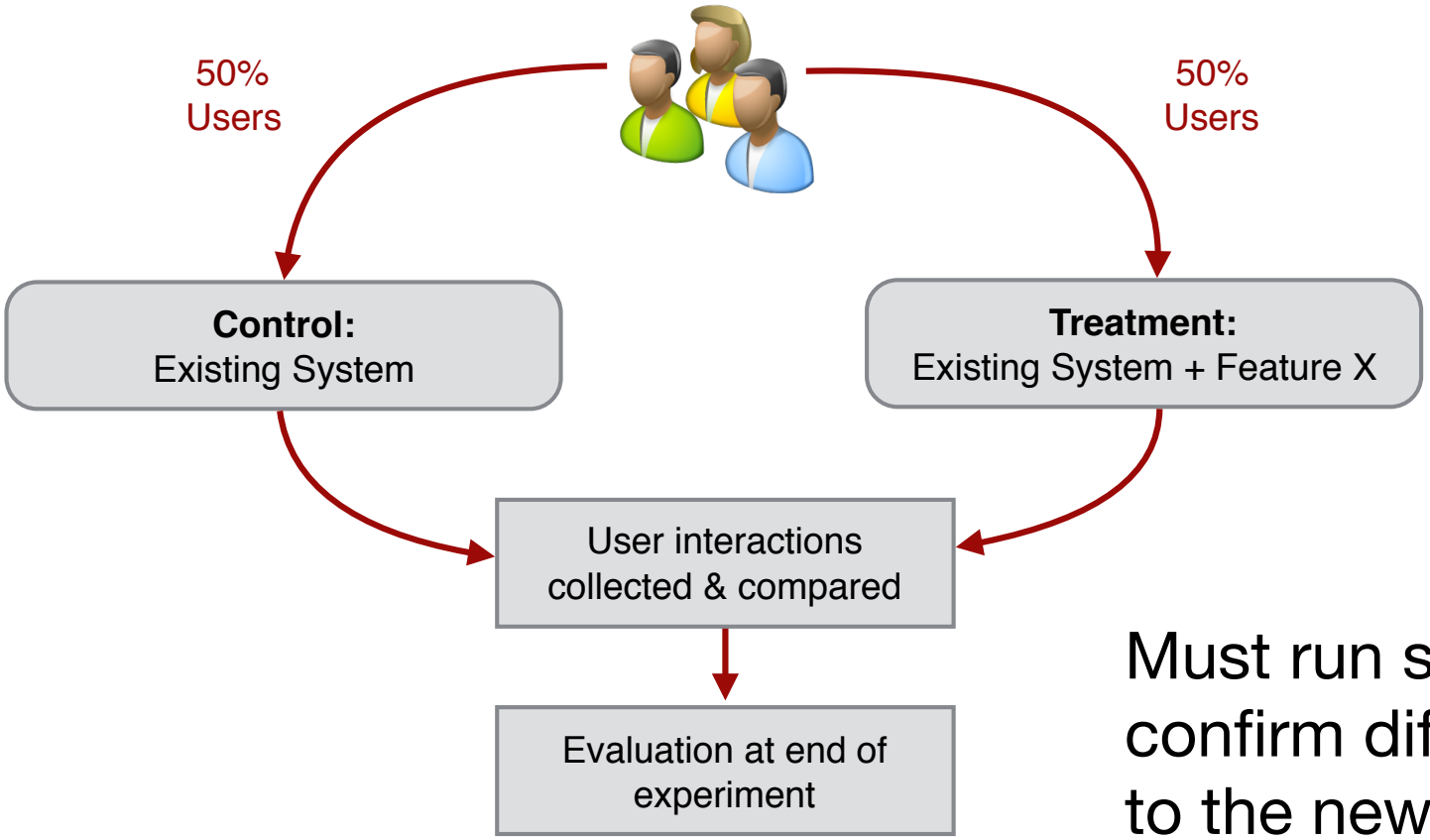
A / B Testing

- **Example: Amazon Shopping Cart Recommendations**
  - Add an item to your shopping cart at a website, most sites then show cart to the user.
  - At Amazon, Greg Linden suggested showing the user recommendations based on cart items instead.
  - What are the possible effects of this website change?
    - ✓ Pro: cross-sell more items (increase average basket size)
    - X Con: distract people from checking out (reduce conversion)
  - Evaluation: Simple user experiment was run, change was wildly successful.

http://glinden.blogspot.com/2006/04/early-amazon-shopping-cart.html

Simple Controlled Experiments

1. Randomly split traffic between two or more versions
   e.g. (A) Control, (B) Treatment
2. Collect and analyse metrics of interest



Must run statistical tests to confirm differences are due to the new feature, not due to chance!

Basic Statistics Reminder

- Let $(x_1, x_2, \ldots, x_n)$ be the values of some variable (data) *X*, for a sample of size *n*.
- The arithmetic mean of the data *X* is calculated as:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- Measures of dispersion characterise how spread out the distribution of the sample is - i.e. how variable the data are.
- The variance is the arithmetic mean of the squared deviations from the sample mean.

$$var(X) = \frac{\sum_{i=1}^{n}(x_i - \bar{X})^2}{n-1}$$

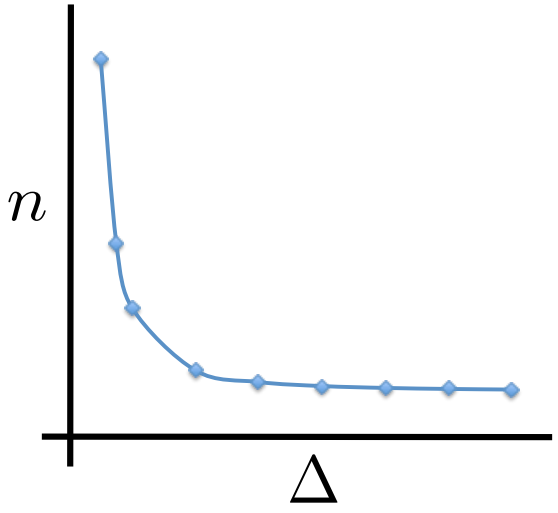
- The standard deviation is the square-root of the variance.

$$\sigma(X) = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{X})^2}{n-1}}$$

Hypothesis Testing

- For statistical significance, the most important relationship is between the difference (delta) and sample size *n*.

$$n \propto \frac{1}{\Delta^2}$$



Inverse square relationship

- The smaller the difference, the more data required to test the hypothesis.
- In the past, getting enough data to test a hypothesis was the problem.
- Now we often (but not always) have to deal with an overabundance of data.

Example: Statistical Significance

- Have cases of two different treatment for broken wrists. Two groups:
  1. *Control:* Plaster Cast
  2. *Treatment:* Surgery (Pins) + Cast
- Want to test for difference in proportions. Is there a significant difference between the control and the treatment?

	Control	Treatment
Size	50	50
Cured	10	20

Contingency Table

- Difference between two groups is statistically significant ($p \cong 0.04$).
- If only 18/50 patients in treatment group had been cured (instead of 20/50), this difference would not be significant.
- Small sample size has a substantial impact on significance here.

➡ Good News: Significant effects do not always require big data

Example: Statistical Significance

- Report on deaths after surgery surveyed over one week in 2011.
- Is there a significant difference between death rates in UK and Ireland?

	UK	Ireland
Cohort	10630	856
Died	378	55
	3.56%	6.43%

- Difference between two groups is statistically significant.
- If only 41/856 patients in Ireland had died (instead of 55/856), difference would not be significant. Difference could be due to chance.
- Small sample size has a substantial impact on significance here.

Mortality after surgery in Europe

http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3493988/

Hypothesis Testing

- The goal of hypothesis testing is to formally examine two opposing hypotheses H_0 and H_A. These two hypotheses are mutually exclusive, so one is true to the exclusion of the other.

Definitions

- Null Hypothesis H_0: States the assumption to be tested.
  e.g. There is no difference between the performance of two machine learning algorithms.
- *p*-Value: If H_0 is true, the probability of observing the test statistic.
- *Type I error*: Rejecting H_0 when it is in fact true.
  i.e. “false alarm” - detecting a difference, when none exists.
- *Type II error*: Failing to reject H_0 when it is in fact false.
  i.e. concluding there is no difference, when there is.
- *Power of a test*: The potential of a statistical test to correctly reject a false null hypothesis H_0 (i.e. not commit a Type II error).

Type I and Type II Errors

- *Type I error*: Rejecting H_0 when it is in fact true.
  i.e. “false alarm” - detecting a difference, when none exists.
- *Type II error*: Failing to reject H_0 when it is in fact false.
  i.e. concluding there is no difference, when there is.

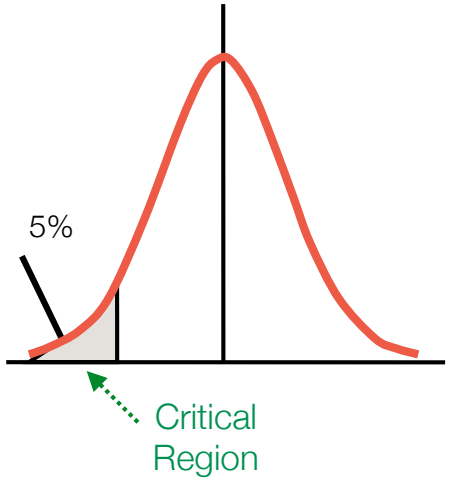
Statistical Test Result

Real World		H_0 Rejected	H_0 Not Rejected
	There is a real difference	**Correct** A Hit	**Type II Error** Missed a real difference
	There is in fact no difference	**Type I Error** False alarm	**Correct** Right to be sceptical of H_A

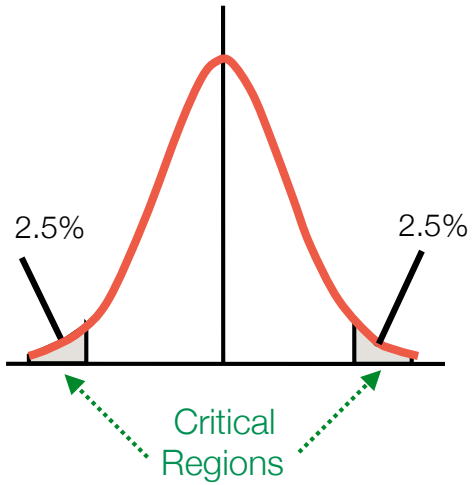
Two Tail vs One Tail

Before testing, we need to decide if we are interested in a *one-tailed* or a *two-tailed* statistical test.

- One-tailed: We decide in advance of looking at the data that one *mean value* will be larger than the other.
  e.g. “Did a generic drug work <u>better than</u> a brand name drug?”



- Two-tailed: We have no strong belief on whether the sample mean is likely to be higher or lower than the mean in the null hypothesis.
  e.g. “Did a generic drug work <u>better than or worse than</u> a brand name drug?”



P-Value Testing

General Approach for Testing

1. Calculate a test statistic on the sample data that is relevant to the hypothesis being examined.
2. Convert the result to a *p*-value by comparing its value to the distribution of test statistics under the null hypothesis.
3. Decide, for a specific level of significance, if we should reject or not reject the null hypothesis, based on the *p*-value:

$$p \leq \alpha \implies \text{reject } H_0 \text{ at level } \alpha$$
$$p > \alpha \implies \text{do not reject } H_0 \text{ at level } \alpha$$

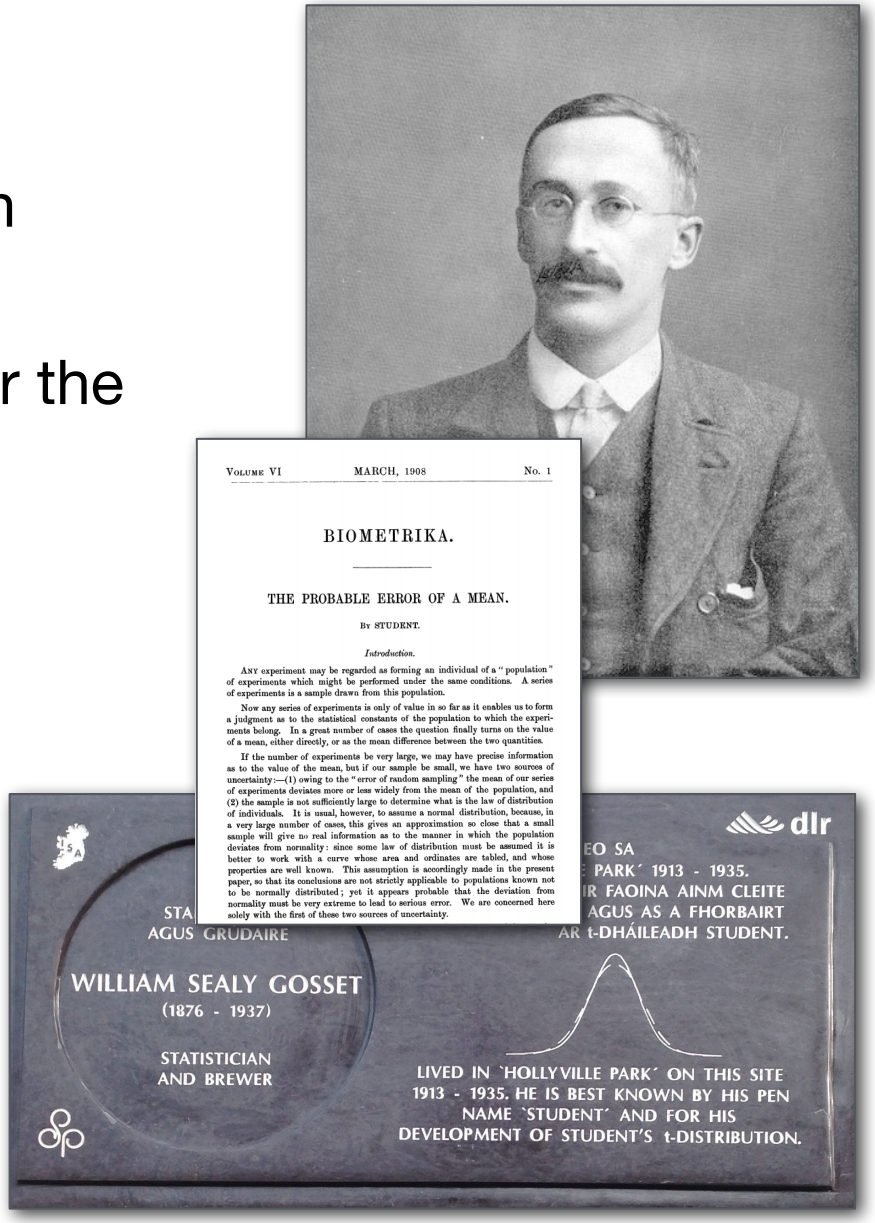
"Is it low enough to be significant?"

- The actual *p*-value threshold (α) depends on the problem, but 0.05 or 0.01 are often chosen "by default".
- The choice controls the Type I Error rate: "How serious is it to believe that something is true when it is in fact false?"

Student's t-Test

- William Sealy Gosset - an English statistician who was employed as a chemist by Arthur Guinness & Son in Dublin.
- Wrote papers in his spare time under the pen name "Student".
- Most noteworthy achievement is called Student's t-test (1908), designed to compare small samples from quality control experiments in brewing.

➡ Are the means of two groups statistically different from each other?



Student's t-Test

- Comparing scores for 2 teams. Is Team A <u>better than</u> Team B?

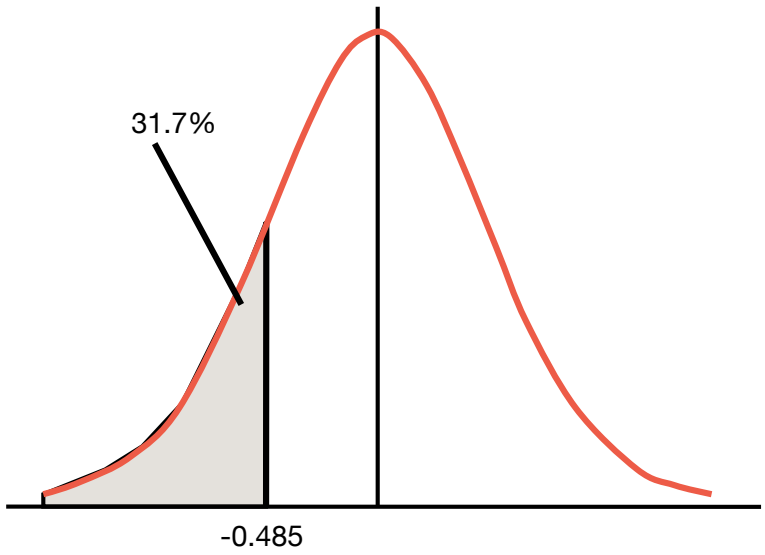
Team A	Team B
23	26
12	15
14	17
54	57
34	45
12	15
9	12
9	18
18	9
21	24

	Team A	Team B
N	10	10
Mean	20.600	23.800
Std Dev	14.017	15.455
Variance	196.489	238.844

Test statistic	t = -0.4850
P-Value	P(T≤t) one tail = 0.317

one tail

มากกว่า alpha ยอมรับ h0



- For a given *t*-statistic value you can look up the confidence.
- There is a 31.7% chance that this difference is due to chance (according to this test).

➡ Difference between Team A and Team B is unlikely to be statistically significant.

Student's t-Test

- More observations and/or greater difference more likely to give statistical significance.

Team A	Team B
23	29
12	20
14	17
23	26
34	45
12	15
9	12
9	18
18	9
21	24
12	15
12	15
14	17
33	36
34	45
12	15
9	12
9	18
18	21
12	15

	Team A	Team B
N	20	20
Mean	17.000	21.200
Std Dev	8.423	10.288
Variance	70.947	105.853

Test statistic	t = -1.413
P-Value	P(T≤t) one tail = 0.083

one tail

➡ There is now a 8.3% chance that this difference is due to chance (according to this test).

Paired t-Tests

- Scores can be paired. e.g. Compare results achieved against the same teams: *Team A v Team C* & *Team B v Team C*
- Interested in the differences between each pair of scores.
- With paired data statistical significance can be determined using fewer observations.

Team A	Team B	Delta
23	26	-3
12	15	-3
14	17	-3
54	57	-3
34	45	-11
12	15	-3
9	12	-3
9	18	-9
18	9	9
21	24	-3

	Team A	Team B
N	10	10
Mean	20.600	23.800
Std Dev	14.017	15.455
Variance	196.489	238.844

Test statistic	t = -1.945
P-Value	P(T≤t) one tail = 0.042

one tail

➡ Lower P-value. We can now say with 95% confidence that Team B are better than Team A.

Student's t-Test: Formulae

- How are t-statistics calculated?
- Two unpaired samples, *A* and *B*:

$$t = \frac{\overline{X}_A - \overline{X}_B}{\sqrt{\frac{var(A)}{n_A} + \frac{var(B)}{n_B}}}$$

Notation:

$\overline{X}_A$	Mean of sample A
$\overline{X}_B$	Mean of sample B
$var(A)$	Variance of sample A
$var(B)$	Variance of sample B
n_A	Number of observations in A
n_B	Number of observations in B

- What about paired data?
- Two paired samples, *A* and *B*:

$$t = \frac{\overline{X}_D \times \sqrt{n}}{\sigma_D}$$

Notation:

D	Difference in pairs in A and B
$\overline{X}_D$	Mean of differences D
σ_D	Standard Dev of differences B
n	Number of observations

Example: Unpaired t-Test

- Is Team A better than Team B, based on unpaired results?

Team A	Team B
23	26
12	15
14	17
54	57
34	45
12	15
9	12
9	18
18	9
21	24

	Team A	Team B
N	10	10
Mean	20.600	23.800
Std Dev	14.017	15.455
Variance	196.489	238.844

$$t = \frac{\overline{X}_A - \overline{X}_B}{\sqrt{\frac{var(A)}{n_A} + \frac{var(B)}{n_B}}}$$

$$t = \frac{20.6 - 23.8}{\sqrt{\frac{196.489}{10} + \frac{238.844}{10}}}$$

$$t = -0.4850$$

Apply a one-tailed-test

Test statistic	t = -0.4850
P-Value	P(T≤t) one tail = 0.317

one tail

Example: Paired t-Test

- Is Team A better than Team B, based on paired results?

Team A	Team B	Delta
23	26	-3
12	15	-3
14	17	-3
54	57	-3
34	45	-11
12	15	-3
9	12	-3
9	18	-9
18	9	9
21	24	-3

$$t = \frac{\overline{X}_D \times \sqrt{n}}{\sigma_D}$$

Look at the mean and standard deviation of the differences (deltas)

$$t = \frac{-3.2 \times \sqrt{10}}{5.2} = -1.946$$

Observations (n)	10
Mean of differences (deltas)	-3.2
Std Dev of differences (deltas)	5.20

Test statistic	t = -1.946
P-Value	P(T≤t) = 0.084

two tails

- Paired t-tests are often used for comparing classifiers if multiple test sets are available, and also in cross validation experiments.

Testing - Implementations

- Many libraries and packages are available for hypothesis testing e.g. SciPy for Python, Apache Commons Math for Java

Standard t-tests (**two tails**):

```
>>> from scipy import stats
>>> a = [23,12,14,23,34,12,9,9,18,21,12,12,14,33,34,12,9,9,18,12]
>>> b = [29,20,17,26,45,15,12,18,9,24,15,15,17,36,45,15,12,18,21,15]
>>> t, pvalue = stats.ttest_ind(a,b)
>>> print "The t-statistic is %.3f and the p-value is %.3f." % (t,pvalue)
The t-statistic is -1.413 and the p-value is 0.166.
```

Paired t-tests (**two tails**):

```
>>> from scipy import stats
>>> a = [23, 12, 14, 54, 34, 12, 9, 9, 18, 21]
>>> b = [26, 15, 17, 57, 45, 15, 12, 18, 9, 24]
>>> t, pvalue = stats.ttest_rel(a,b)
>>> print "The paired t-statistic is %.3f and the p-value is %.3f." % (t,pvalue)
The paired t-statistic is -1.945 and the p-value is 0.084.
```

Difference in Proportions

- A *t*-test is sometimes used to analyse differences in proportions e.g. comparison of conversion rates in A/B testing.
- Requires a number of assumptions about the population which are usually not true.

	Control	Treatment
Samples	n_1	n_2
Conversions	c_1	c_2

$$p = \frac{c1 + c2}{n1 + n2} \qquad p_1 = \frac{c_1}{n_1} \qquad p_2 = \frac{c_2}{n_2}$$

$$t \text{ statistic} = \frac{\text{Difference in proportions}}{\text{Standard error}}$$

$$t = \frac{p1 - p2}{\sqrt{p(1-p) \times (\frac{1}{n_1} + \frac{1}{n_2})}}$$

http://stattrek.com/hypothesis-test/difference-in-proportions.aspx

McNemar's Test

- Measure for comparing paired proportions.
  e.g. Which is better, classifier C2 or C3?
- Applied to 2x2 contingency tables.
- Test captures two key differences:

  n_{01}: number misclassified by 1st but not 2nd classifier.

  n_{10}: number misclassified by 2nd but not 1st classifier.

C1	C2	C3
✓	✓	✗
✓	✓	✓
✓	✓	✗
✓	✓	✓
✓	✓	✗
✗	✓	✗
✗	✓	✓
✗	✗	✗
✗	✗	✓
✗	✗	✓

Contingency for C2 v C1

3 n_{00}	2 n_{01}
0 n_{10}	5 n_{11}

McNemar C2 v C1

$$\chi^2 = 1/2 = 0.5$$

$$\chi^2 = \frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}}$$

Note: For test to be applicable require (n_{01}+n_{10}) > 10

Contingency for C3 v C1

1 n_{00}	2 n_{01}
4 n_{10}	3 n_{11}

McNemar C3 v C1

$$\chi^2 = 1/6 = 0.1666$$

➡ $\chi^2 > 3.84$ required for statistical significance at 95%. So neither classifier significantly better!

Summary

- Objectives of Evaluation
- A/B Testing
- Hypothesis Testing
  - Student's t-test
  - t-Test for paired data
  - Differences in proportions
  - McNemar's test for proportions
- Next: Evaluation measures and setup for classification