

---

# Real World Applications of Data Science

**In partnership with:**  
**Proscia Inc, Betamore, Spark B-more**

Lecture 2 notes: Machine learning 101 + Model evaluation

---

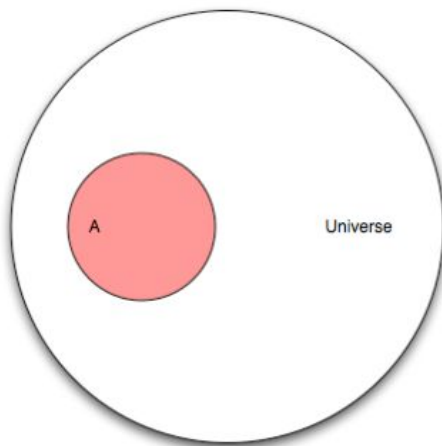
# Supervised Learning

- 1) Intro Prob + Bayes Theorem
- 2) Naive Bayes Classification
- 3) Decision Trees
- 4) Support Vector Machines

---

---

# Probability



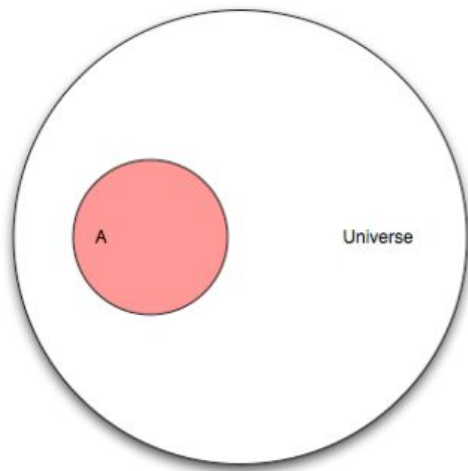
Let's pretend you are flipping a coin. This diagram represents the “universe” of all possible outcomes, also known as events. This universe is known as the sample space.

Q: What are the mutually exclusive events that make up the sample space for a coin flip?

---

---

# Probability



Let's pretend you are flipping a coin. This diagram represents the "universe" of all possible outcomes, also known as events. This universe is known as the sample space.

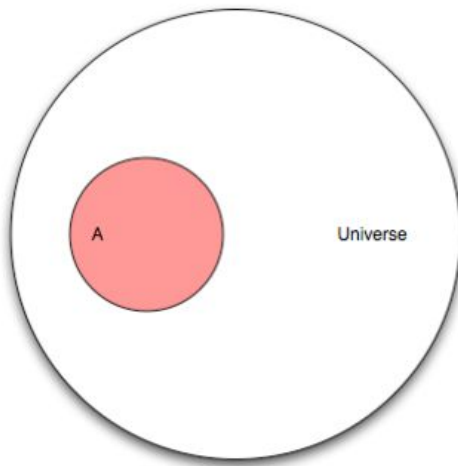
Q: What are the mutually exclusive events that make up the sample space for a coin flip?

A: Heads and tails

---

---

# Probability



Let's now pretend that our universe involves a research study on humans. Event "A" is people in that study who have cancer.

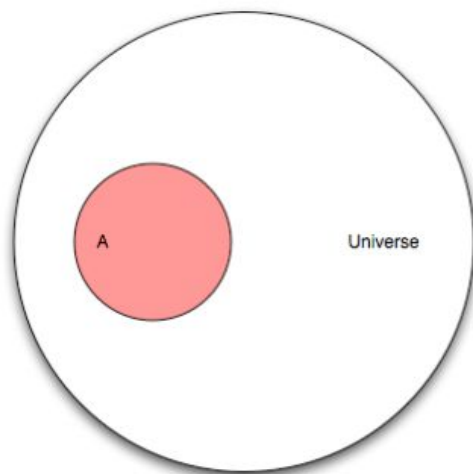
Q: If our study has 100 people and "A" has 25 people, what is the probability of A?

Q: What is the max probability of any event?

---

---

# Probability



Let's now pretend that our universe involves a research study on humans. Event "A" is people in that study who have cancer.

Q: If our study has 100 people and "A" has 25 people, what is the probability of A?

A:  $P(A) = 25/100$

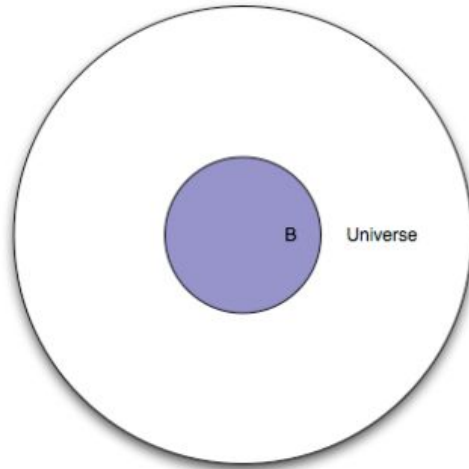
Q: What is the max probability of any event?

A: 1

---

---

# Probability



This represents the same set of people, except everyone in the study is given a test. Event “B” is everyone in the study for whom the test is positive.

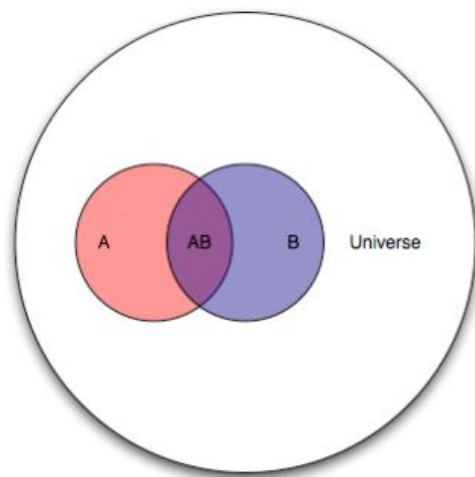
Q: What portion of the diagram represents the subset of people with a negative test?

A: The white area between the smaller circle and the larger circle.

---

---

# Probability



Because “A” and “B” are events from the same study, we can show them together.

Q: How would you describe the “cancer status” and “test status” of people in each area of the diagram?

A: Pink: cancer, negative test

Purple: cancer, positive test

Blue: no cancer, positive test

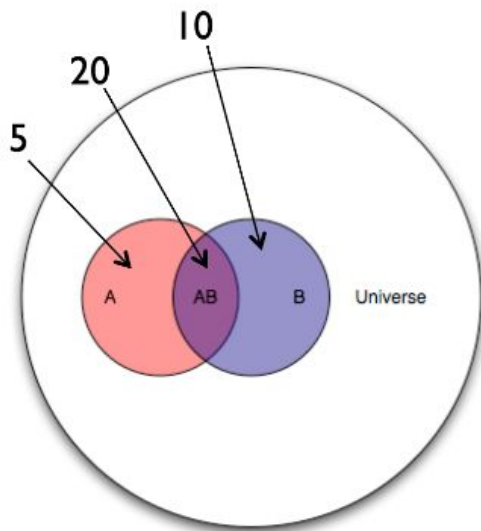
White: no cancer, negative test

---



---

# Probability

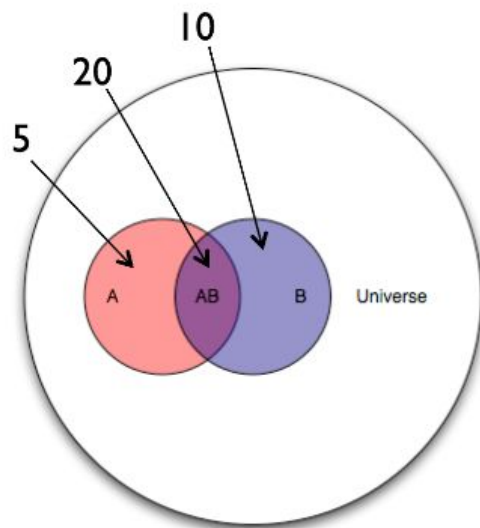


The purple section is known as the intersection of A and B, denoted as  $P(AB)$ .

Thinking of this test as a classifier for predicting cancer, draw the confusion matrix.

n=100	Predicted: NO	Predicted: YES
	Actual: NO	Actual: YES
	65	10
	5	20

# Probability

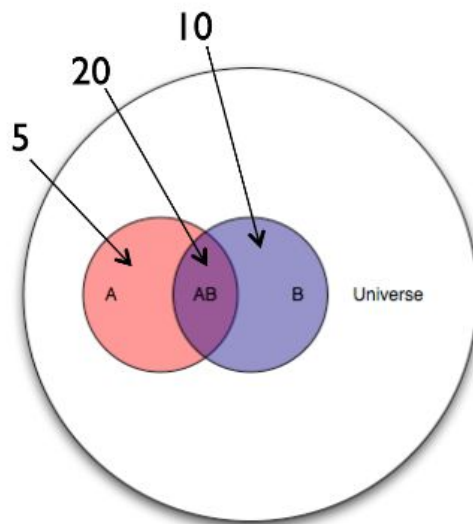


Q: Let's pick an arbitrary person from this study. If you were told their test result was positive, what is the probability they actually have cancer?

n=100	Predicted: NO	Predicted: YES
Actual: NO	65	10
Actual: YES	5	20

---

# Probability



Q: Let's pick an arbitrary person from this study. If you were told their test result was positive, what is the probability they actually have cancer?

A: 20/30

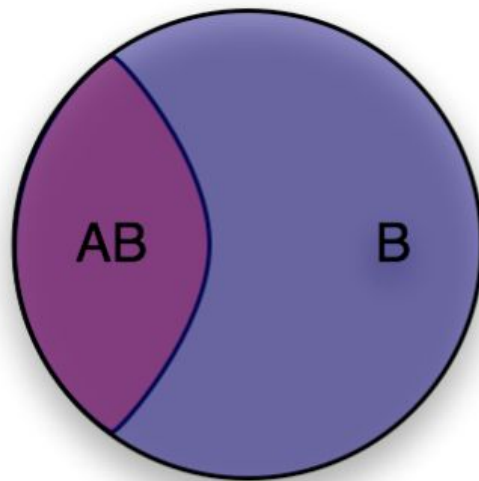
This is the conditional probability of A given B, denoted as  $P(A|B)$ .

$$P(A|B) = P(AB) / P(B) = (20/100) / (30/100)$$

---

---

# Probability



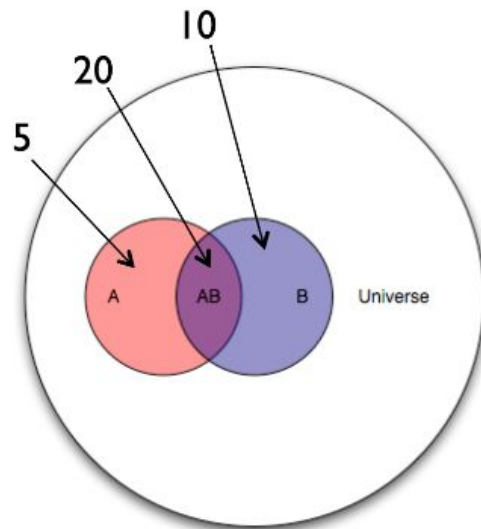
You can think of conditional probability as “changing the relevant universe.”  $P(A|B)$  is a way of saying “Given that my entire universe is now B, what is the probability of A?”

This is also known as transforming the sample space.

---

---

# Probability

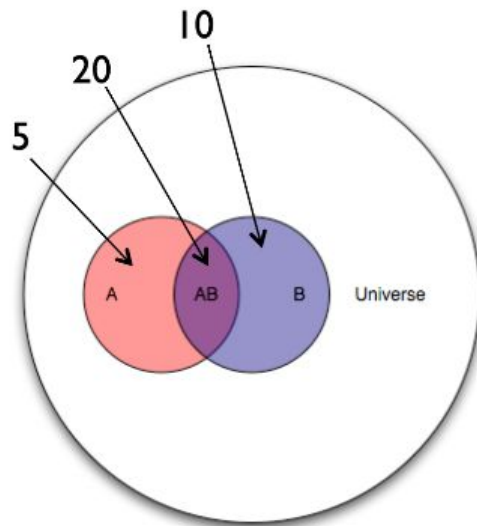


Q: Let's pick another arbitrary person from this study. If you were told they have cancer, what is the probability they had a positive test result?

---

---

# Probability



Q: Let's pick another arbitrary person from this study. If you were told they have cancer, what is the probability they had a positive test result?

A:  $P(B|A) = P(AB) / P(A) = 20/25$

---

---

# Bayes Theorem

---

---

# Bayes Theorem derived

Deriving Bayes' theorem:

We know:

$$P(A|B) = P(AB) / P(B) \text{ and } P(B|A) = P(AB) / P(A)$$

Thus:

$$P(AB) = P(A|B) * P(B) = P(B|A) * P(A)$$

Rearrange to get Bayes' theorem:

$$P(A|B) = P(B|A) * P(A) / P(B)$$

---



---

# Bayes Theorem Example

Suppose you might have a rare life-threatening disease and so you get tested. The disease's test is 99% sensitive and 99% specific (if you have it, the test is correct 99% of the time and same if you don't have it). This disease occurs in 1 in every 10,000 people.

Q. Your test is positive. What is the probability that you have the disease?

---

---

# Bayes Theorem Example

Q. Your test is positive. What is the probability that you have the disease?

A. 1%

Let  $A$  be the event that you have the disease

$B$  be the event that your test was positive

$P(B|A) = .99$  (sensitivity)

$P(B|\text{not } A) = .01$  ( $1 - \text{sensitivity}$ ) this is our false positive

---

---

# Bayes Theorem Example

Q. Your test is positive. What is the probability that you have the disease?

A. 1%

Let  $A$  be the event that you have the disease  
Let  $B$  be the event that your test was positive

$P(B|A) = .99$  (sensitivity)       $P(B|\text{not } A) = .01$  (1 - sensitivity) this is our false positive

$$\begin{aligned} P(B) &= P(\text{the test was positive}) = P(B|A) * P(A) \quad \text{OR} \quad P(B|\text{not } A) * P(\text{not } A) \\ P(B) &= .99 * .0001 + .01 * .9999 \\ &= .010098 \end{aligned}$$

$$\begin{aligned} \text{Bayes Theorem:} \quad P(A|B) &= P(B|A)P(A) / P(B) \\ &= .99 * .0001 / .010098 = 0.00980 \end{aligned}$$

---

---

# Naive Bayes Classification

---

# Bayesian Inference

Suppose we have a dataset with features  $x_1, \dots, x_n$  and a class label  $c$ . What can we say about classification using Bayes' theorem?

$$P(\text{class } C | \{x_i\}) = \frac{P(\{x_i\} | \text{class } C) \cdot P(\text{class } C)}{P(\{x_i\})}$$


Bayes' theorem can help us to determine the probability of a record belonging to a class, *given* the data we observe.

---

---

# Bayesian Inference


This term is the prior probability of  $c$ . It represents the probability of a record belonging to class  $c$  before the data is taken into account.

$$P(\text{class } C | \{x_i\}) = \frac{P(\{x_i\} | \text{class } C) \cdot P(\text{class } C)}{P(\{x_i\})}$$


---

# Bayesian Inference


This term is the likelihood function. It represents the joint probability of observing features  $\{x_i\}$  given that that record belongs to class  $c$ .

$$P(\text{class } C | \{x_i\}) = \frac{P(\{x_i\} | \text{class } C) \cdot P(\text{class } C)}{P(\{x_i\})}$$


---

# Bayesian Inference

This term is the normalization constant. It doesn't depend on  $c$ , and is generally ignored.

$$P(\text{class } C \mid \{x_i\}) = \frac{P(\{x_i\} \mid \text{class } C) \cdot P(\text{class } C)}{P(\{x_i\})}$$




---

# Bayesian Inference

This term is the posterior probability of  $c$ . It represents the probability of a record belonging to class  $c$  after the data is taken into account.

$$P(\text{class } C | \{x_i\}) = \frac{P(\{x_i\} | \text{class } C) \cdot P(\text{class } C)}{P(\{x_i\})}$$

The idea of Bayesian inference, then, is to update our beliefs about the distribution of  $c$  using the data (“evidence”) at our disposal.

---

---

# Naive Bayes Classification

Q: What piece of the puzzle we've seen so far looks like it could intractably difficult in practice?

A: Estimating the full likelihood function.

$$P(\{x_i\}|C) = P(\{x_1, x_2, \dots, x_n\}|C)$$

Observing this exactly would require us to have enough data for every possible combination of features to make a reasonable estimate.

---

---

# Naive Bayes Classification

Q: So what can we do about it?

A: Make a simplifying assumption. In particular, we assume that the features  $x_i$  are conditionally independent from each other:

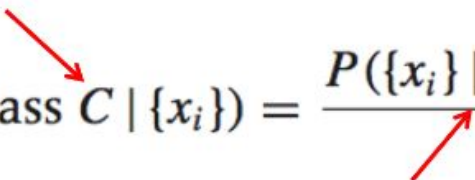
$$P(\{x_i\}|C) = P(\{x_1, x_2, \dots, x_n\}|C) \approx P(x_1|C) * P(x_2|C) * \dots * P(x_n|C)$$

This “naïve” assumption simplifies the likelihood function to make it tractable.

---

---

# Naive Bayes Classification

$$P(\text{class } C | \{x_i\}) = \frac{P(\{x_i\} | \text{class } C) \cdot P(\text{class } C)}{P(\{x_i\})}$$


In summary, the training phase of the model involves computing the likelihood function, which is the conditional probability of each feature given each class.

The prediction phase of the model involves computing the posterior probability of each class given the observed features, and choosing the class with the highest probability.

---

# Decision Trees

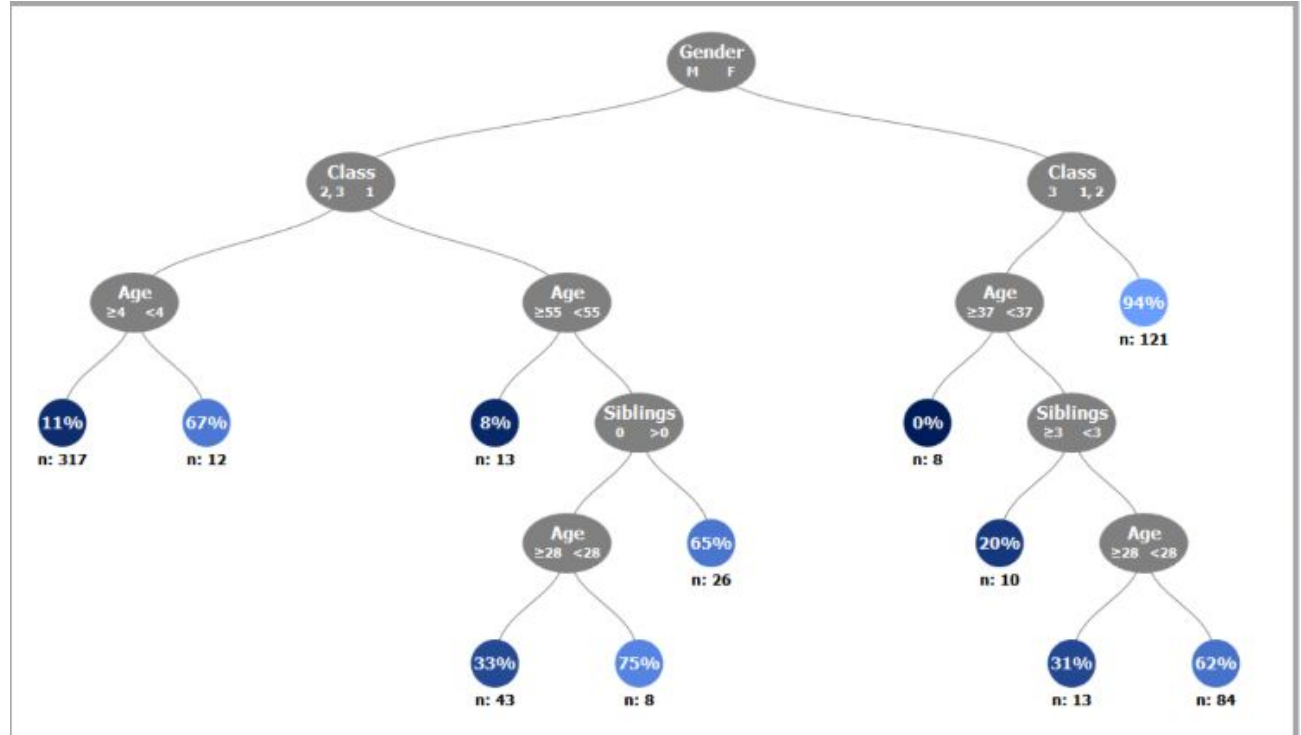
---

# Decision Trees

- A supervised learning technique that can be used for discrete and continuous output
  - Visually engaging and easy to interpret
  - Excellent model for transition into data science
  - Foundation to learning powerful, advanced techniques like Random Forest and boosted trees
  - Prone towards high-variance
-

# Trees

- Trained on Titanic
- Interconnected nodes which act as a series of questions/test conditions
- Terminal nodes show output metric, here it shows percentage of titanic survivors with combo of variables



---

---

# Trees

So, how does the algorithm choose which variables to include on the tree?

How does the algorithm choose where variables should be located on the tree?

How does the algorithm choose when to stop the tree?

---



---

# Decision Trees: the algorithm

- 1) Calculate purity of the data
  - 2) Select a candidate split
  - 3) Calculate the purity of the data after the split
  - 4) Repeat for all variables
  - 5) Choose the variables with greatest increase in purity
  - 6) Repeat for each split until some stop criteria is met
-

---

# Support Vector Machines

---

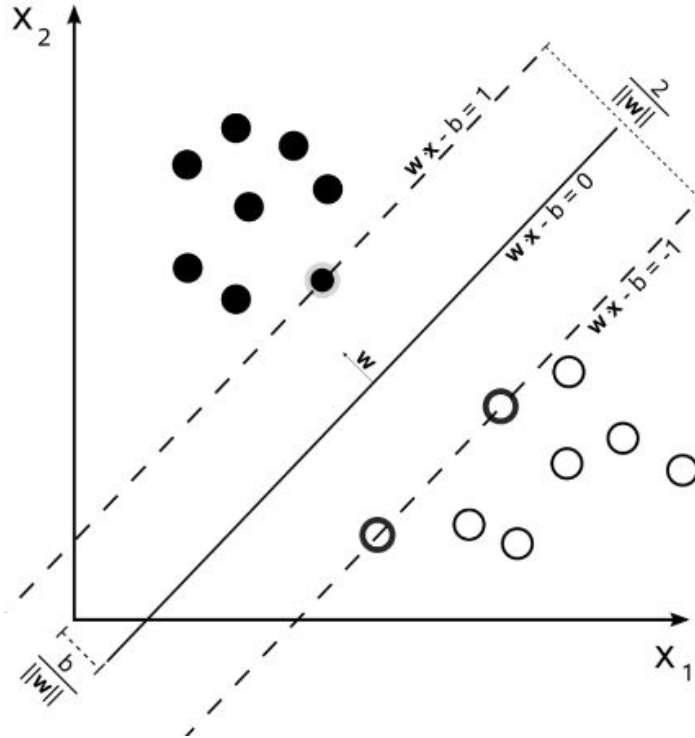
---

# **Support Vector Machines**

Constructs a hyperplane to  
separate classes in space

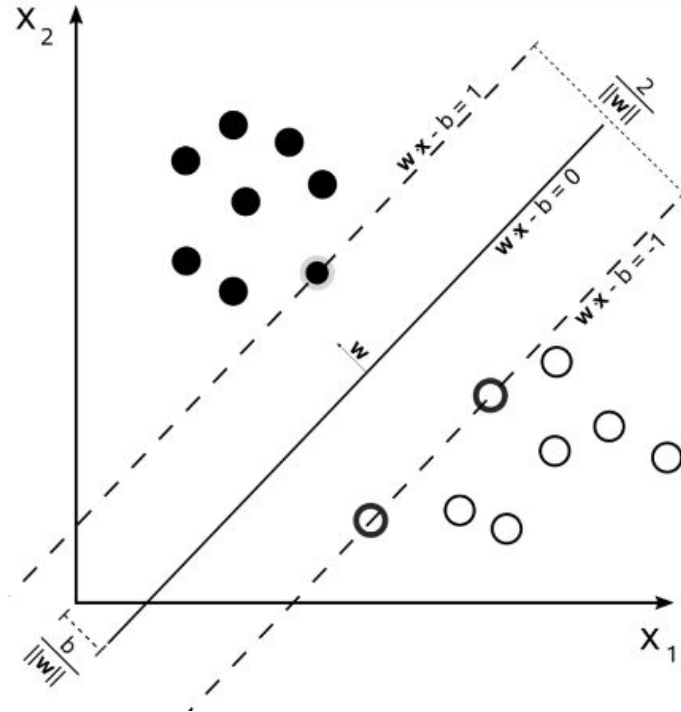
---

# Support Vector Machines



# Support Vector Machines

We want to maximize the width of the margin



---

---

# SVM

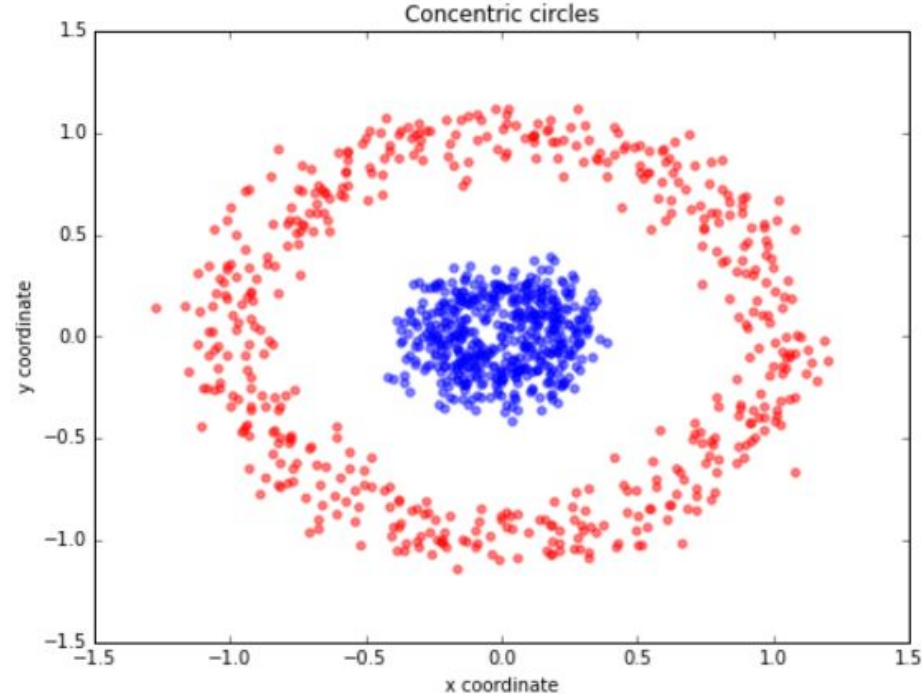
What if there is no easy  
hyperplane?

---

---

# Concentric Circles

- Pretty much no hyperplane will separate this out, but what if we could add a THIRD dimension?



---

## SVM

Q. OK fine, but what if I have 100 predictors? How many dimensions should I project into?

A. An arbitrary amount, possible infinite..

---



---

## Kernel Trick

Assume a certain shape of data  
and kernel trick saves huge  
computation time

---

---

# Kernel Trick

## Example:

Linear      ( assumes a linear boundary)  
Poly        ( assumes a curved boundary)  
Gaussian    ( assumes a spherical boundary)

---

---

# Support Vector Machines

## Pros

- Very fast training and predicting with kernel trick
- Built on solid mathematical foundation (unlike ANN)
- Very common and in sklearn

## Cons

- A lot of “guess work” with kernels
  - Hard to grasp math behind it (ok if you accept the black box)
-

# Model Evaluation

- 1) Confusion Matrix
- 2) ROC/AUC Curves

---

---

# Confusion Matrix

Confusion Matrix: table to describe the performance

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

Example: Test for presence of disease

NO = negative test = False = 0

YES = positive test = True = 1

- How many classes are there?
  - How many patients?
  - How many times is disease predicted?
  - How many patients actually have the disease?
-

---

# Confusion Matrix

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Accuracy:

- Overall, how often is it **correct**?
- $(TP + TN) / \text{total} = 150/165 = 0.91$

Basic Terminology:

- True Positives (TP)
- True Negatives (TN)
- False Positives (FP)
- False Negatives (FN)

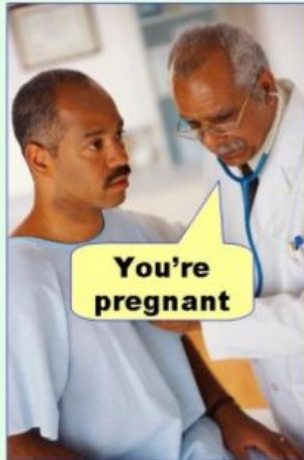
Misclassification Rate (Error Rate):

- Overall, how often is it **wrong**?
  - $(FP + FN) / \text{total} = 15/165 = 0.09$
-

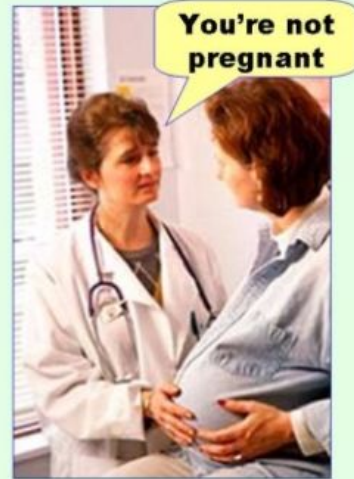
---

# Errors

**Type I error**  
(false positive)



**Type II error**  
(false negative)



---

# Confusion Matrix

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

False Positive Rate:

- When actual value is **negative**, how often is prediction **wrong**?
- $FP / \text{actual no} = 10/60 = 0.17$

Sensitivity:

- When actual value is **positive**, how often is prediction **correct**?
- $TP / \text{actual yes} = 100/105 = 0.95$
- “True Positive Rate” or “Recall”

Specificity:

- When actual value is **negative**, how often is prediction **correct**?
  - $TN / \text{actual no} = 50/60 = 0.83$
-



# ROC/AUC Curve

---

# ROC Curve / AUC

Email Number	Score	True Label
5	0.99	Spam
8	0.82	Spam
2	0.60	Spam
1	0.60	Ham
7	0.48	Spam
3	0.22	Ham
4	0.10	Ham
6	0.02	Ham

Every email is assigned a “spamminess” score by our classification algorithm. To actually make our predictions, we choose a numeric cutoff for classifying as spam.

An ROC Curve will help us to visualize how well our classifier is doing without having to choose a cutoff!

---

---

# ROC Curve / AUC

Email Number	Score	True Label
5	0.99	Spam
8	0.82	Spam
2	0.60	Spam
1	0.60	Ham
7	0.48	Spam
3	0.22	Ham
4	0.10	Ham
6	0.02	Ham

The ROC plots the True Positive Rate (TRP) on the y-axis against the False Positive Rate (FPR) on the x-axis.

TPR: When actual value is **spam**, how often is prediction **correct**?

FPR: When actual value is **ham**, how often is prediction **wrong**?

---

---

# ROC Curve / AUC

Email Number	Score	True Label
5	0.99	Spam
8	0.82	Spam
2	0.60	Spam
1	0.60	Ham
7	0.48	Spam
3	0.22	Ham
4	0.10	Ham
6	0.02	Ham

TPR: When actual value is **spam**, how often is prediction **correct**?

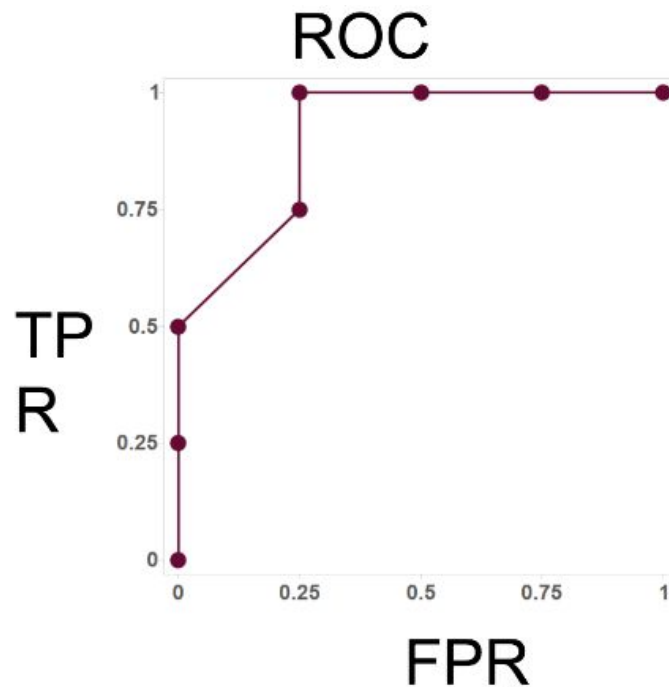
FPR: When actual value is **ham**, how often is prediction **wrong**?

Cutoff	TPR (y)	FPR (x)	Cutoff	TPR (y)	FPR (x)
0	1	1	0.50	0.75	0.25
0.05	1	0.75	0.65	0.5	0
0.15	1	0.5	0.85	0.25	0
0.25	1	0.25	1	0	0

---

# ROC Curve / AUC

Email Number	Score	True Label
5	0.99	Spam
8	0.82	Spam
2	0.60	Spam
1	0.60	Ham
7	0.48	Spam
3	0.22	Ham
4	0.10	Ham
6	0.02	Ham



---

# ROC Curve / AUC

Email Number	Score	True Label
5	0.99	Spam
8	0.98	Spam
2	0.97	Spam
1	0.97	Ham
7	0.96	Spam
3	0.95	Ham
4	0.94	Ham
6	0.93	Ham

Q: Would the ROC Curve (and AUC) change if the **scores** changed but the **ordering** remained the same?

A: Not at all! The ROC Curve is only sensitive to **rank ordering** and does not require **calibrated scores**.

---

---

**Any Questions?**

---