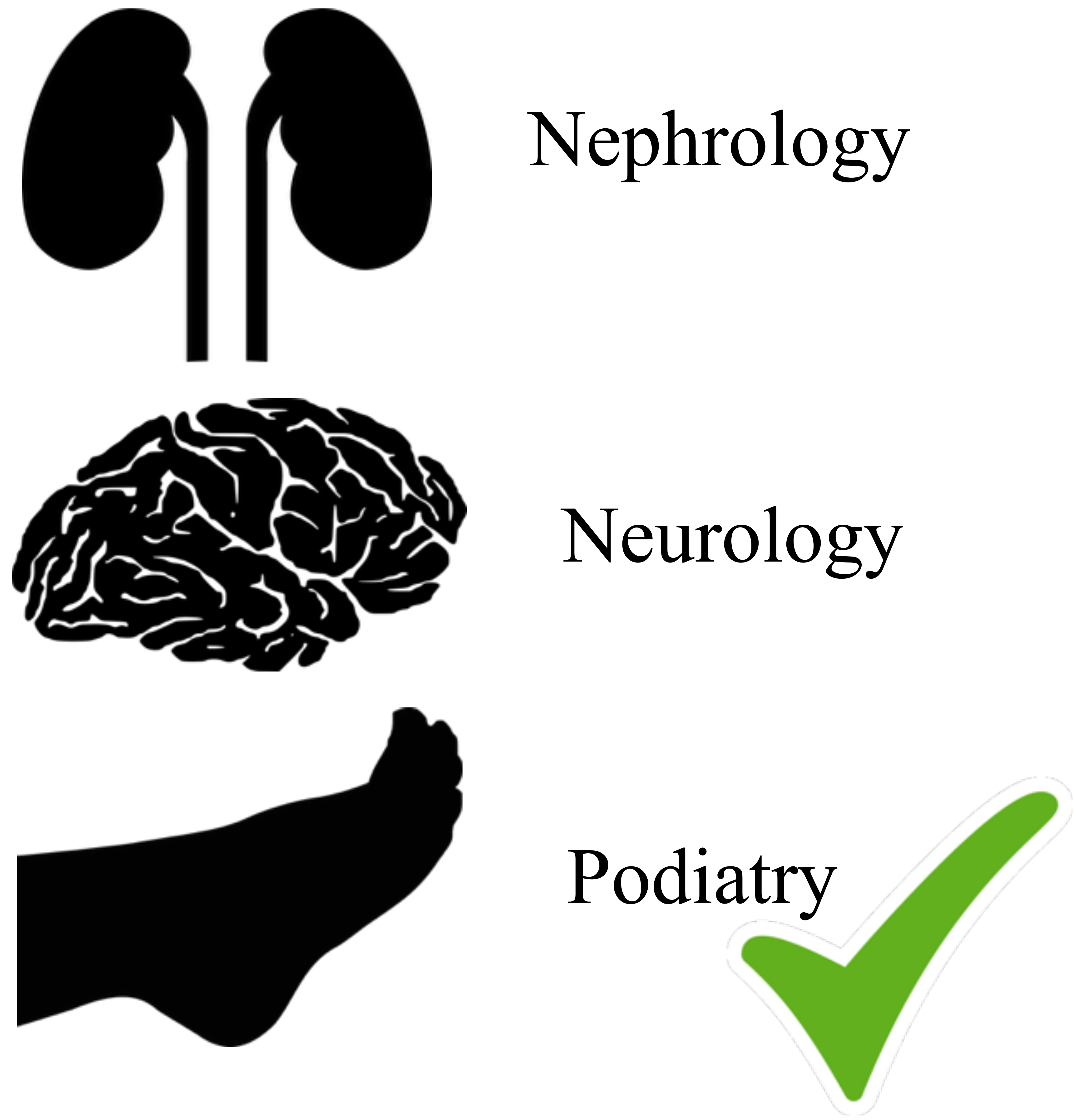
Applied Text Mining in Python

Text Classification

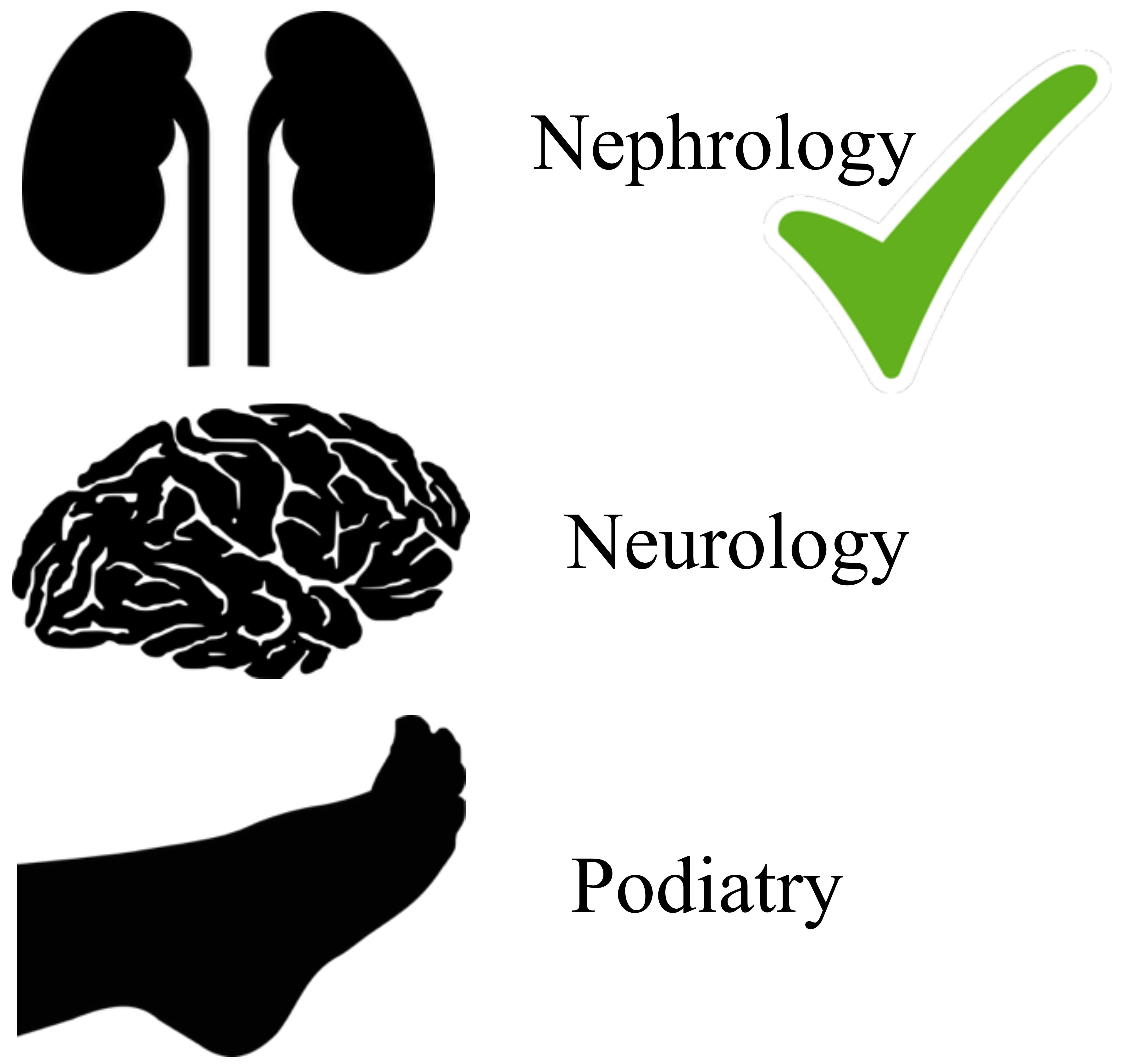
Which medical speciality does this relate to?

TINEA PEDIS, or ATHLETE'S FOOT, is a very common fungal skin infection of the foot. It often first appears between the toes. It can be a one-time occurrence or it can be chronic. The fungus, known as Trichophyton, thrives under warm, damp conditions so people whose feet sweat a great deal are more susceptible. It is easily transmitted in showers and pool walkways. Those people with immunosuppressive conditions, such as diabetes mellitus, are also more susceptible to athlete's foot.



Which medical speciality does this relate to?

KIDNEY FAILURE, also known as RENAL FAILURE or RENAL INSUFFICIENCY, is a medical condition of impaired kidney function in which the kidneys fail to adequately filter metabolic wastes from the blood.The two main forms are acute kidney injury, which is often reversible with adequate treatment, and chronic kidney disease, which is often not reversible. In both cases, there is usually an underlying cause.



What is Classification?

Given a set of classes:

Classification: Assign the correct class label to the given input

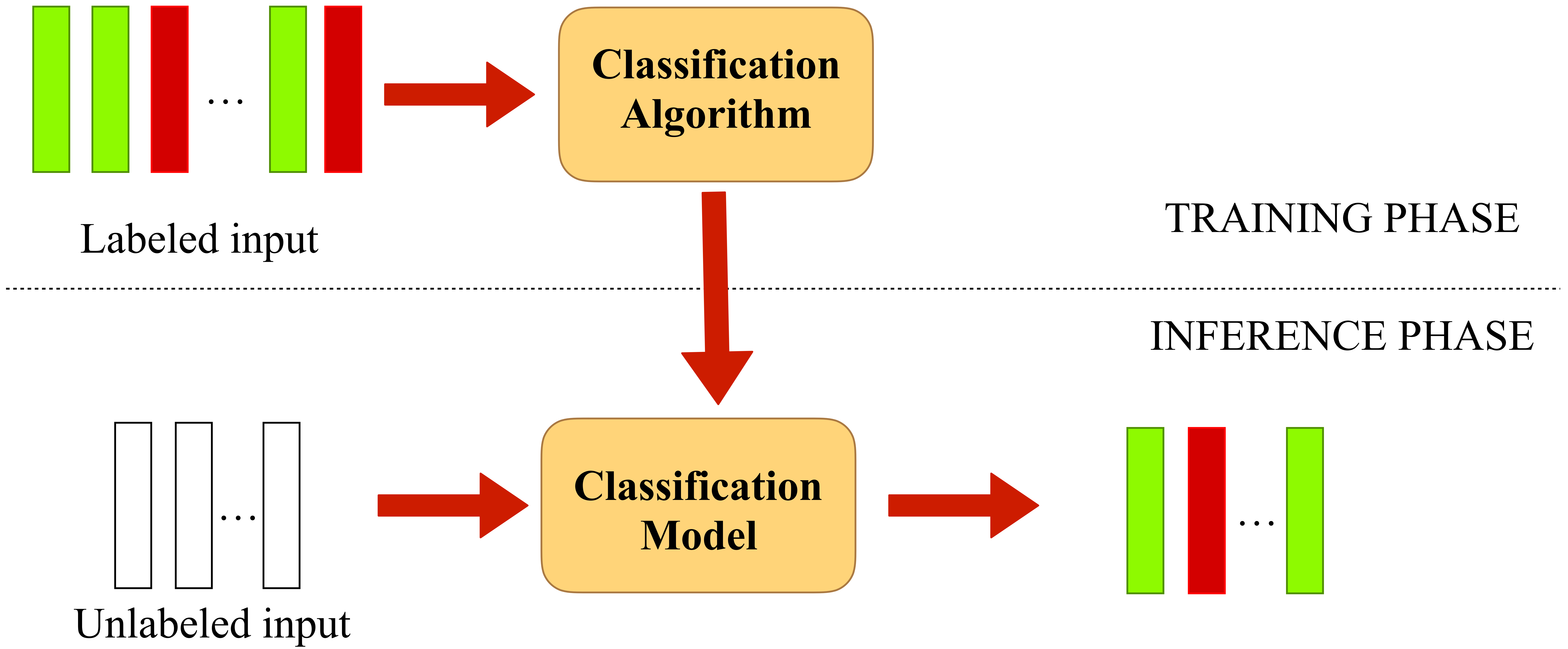
Examples of Text Classification

- Topic identification: Is this news article about Politics, Sports, or Technology?
- Spam Detection: Is this email a spam or not?
- Sentiment analysis: Is this movie review positive or negative?
- Spelling correction: weather or whether?
  color or colour?

Supervised Learning

- Humans learn from past experiences, machines learn from past instances!

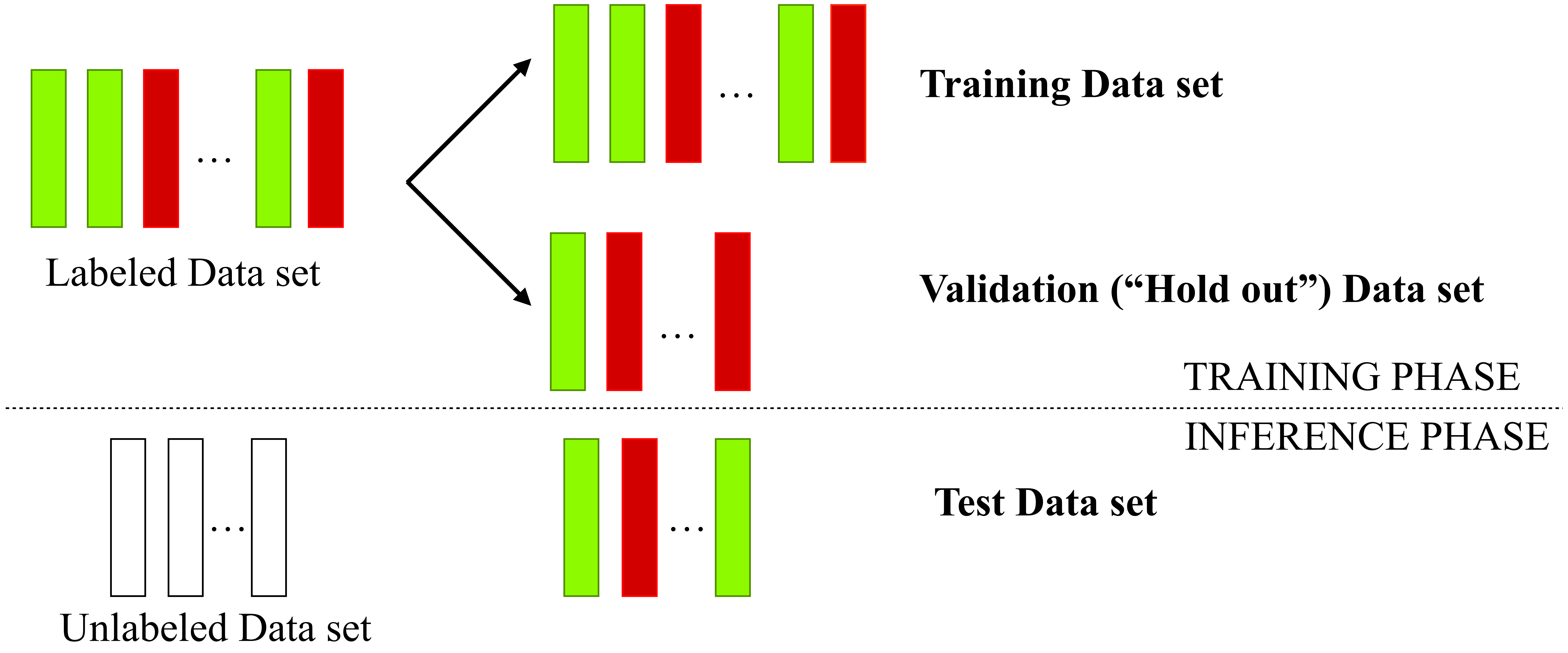
Supervised Classification



Supervised Classification

- Learn a classification model on properties (“features”) and their importance (“weights”) from labeled instances
  - X: Set of attributes or features $\{x_1, x_2, \ldots, x_n\}$
  - y: A “class” label from the label set $Y = \{y_1, y_2, \ldots, y_k\}$
- Apply the model on new instances to predict the label

Supervised Classification: Phases and Datasets



Classification Paradigms

- When there are only two possible classes; $|Y| = 2$:

  Binary Classification
- When there are more than two possible classes; $|Y| > 2$:

  Multi-class Classification
- When data instances can have two or more labels :

  Multi-label Classification

Questions to ask in Supervised Learning

- Training phase:
  - What are the features? How do you represent them?
  - What is the classification model / algorithm?
  - What are the model parameters?
- Inference phase:
  - What is the expected performance? What is a good measure?

Applied Text Mining in Python

Identifying Features from Text

Why is Textual Data Unique?

- Textual data presents a unique set of challenges
- All the information you need is in the text
- But features can be pulled out from text at different granularities!

Types of Textual Features (1)

- Words
  - By far the most common class of features
  - Handling commonly-occurring words: Stop words
  - Normalization: Make lower case vs. leave as-is
  - Stemming / Lemmatization

Types of Textual Features (2)

- Characteristics of words : Capitalization
- Parts of speech of words in a sentence
- Grammatical structure, sentence parsing
- Grouping words of similar meaning, semantics
  - {buy, purchase}
  - {Mr., Ms., Dr., Prof.}; Numbers / Digits; Dates

Types of Textual Features (3)

- Depending on classification tasks, features may come from inside words and word sequences
  - bigrams, trigrams, n-grams: “White House”
  - character sub-sequences in words: “ing”, “ion”, …

How would you do it?

- Recall lectures from previous week

Applied Text Mining in Python

Naïve Bayes Classifier

Case study: Classifying text search queries

- Suppose you are interested in classifying search queries in three classes: Entertainment, Computer Science, Zoology
- Most common class of the three is Entertainment.

Case study: Classifying text search queries

- Suppose the query is "Python"
  - Python, the snake (Zoology)
  - Python, the programming language (Computer Science)
  - Python, as in Monty Python (Entertainment)

- Most common class, given "Python", is Zoology.

Case study: Classifying text search queries

- Suppose the query is "Python download"
- Most probable class, given "Python download", is Computer Science.

Probabilistic Model

- Update the likelihood of the class given new information
- Prior Probability: Pr(y = Entertainment), Pr(y = CS), Pr(y=Zoology)
- Posterior probability: Pr(y = Entertainment|x = “Python”)

Bayes' Rule

- $$\text{Posterior probability} = \frac{\text{Prior probability} \times \text{Likelihood}}{\text{Evidence}}$$

- $$\Pr(y \mid X) = \frac{\Pr(y) \times \Pr(X \mid y)}{\Pr(X)}$$

Naïve Bayes Classification

- $\Pr(y=\text{CS}|\text{“Python”}) = \dfrac{\Pr(y=\text{CS}) \times \Pr(\text{“Python”} \mid y=\text{CS})}{\Pr(\text{“Python”})}$

- $\Pr(y=\text{Zoology}|\text{“Python”})$
  $= \dfrac{\Pr(y=\text{Zoology}) \times \Pr(\text{“Python”} \mid y=\text{Zoology})}{\Pr(\text{“Python”})}$

- $\Pr(y=\text{CS} \mid \text{“Python”}) > \Pr(y=\text{Zoology} \mid \text{“Python”}) \Rightarrow$

Naïve Bayes Classification

- $\Pr(y \mid X) = \dfrac{\Pr(y) \times \Pr(X \mid y)}{\Pr(X)}$

$$y^* = \underset{y}{\operatorname{argmax}} \Pr(y \mid X) = \underset{y}{\operatorname{argmax}} \Pr(y) \times \Pr(X \mid y)$$

- Naïve assumption: Given the class label, features are assumed to be independent of each other

$$y^* = \underset{y}{\operatorname{argmax}} \Pr(y \mid X) = \underset{y}{\operatorname{argmax}} \Pr(y) \times \prod_{i=1}^{n} \Pr(x_i \mid y)$$

Naïve Bayes Classifier

$$y^* = \underset{y}{\operatorname{argmax}} \Pr(y \mid X) = \underset{y}{\operatorname{argmax}} \Pr(y) \times \prod_{i=1}^{n} \Pr(x_i \mid y)$$

Query: “Python download”

$$y^* = \underset{y}{\operatorname{argmax}} \Pr(y) \times \Pr(\text{“Python”} \mid y) \times \Pr(\text{“download”} \mid y)$$

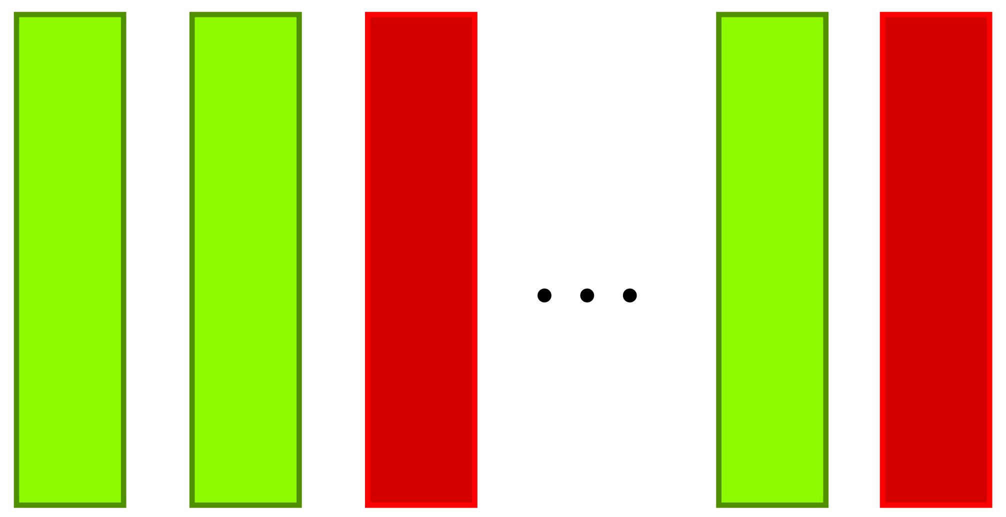
Naïve Bayes: What are the parameters?

- Prior probabilities: Pr(y) for all y in Y
- Likelihood: Pr(x_i | y) for all features x_i and labels y in Y

- If there are 3 classes ($|Y| = 3$) and 100 features in X, how many parameters does naïve Bayes models have?

Naïve Bayes: Learning parameters

- Prior probabilities: Pr(y) for all y in Y
  - Remember training data?



  - Count the number of instances in each class
  - If there are N instances in all, and n out of those are labeled as class y $\Rightarrow$ Pr

Naïve Bayes: Learning parameters

- Likelihood: Pr(x_i | y) for all features x_i and labels y in Y
  - Count how many times feature x_i appears in instances labeled as class y
  - If there are p instances of class y, and x_i appears in k of those, Pr(x_i | y) = k / p

Naïve Bayes: Smoothing

- What happens if $\Pr(x_i \mid y) = 0$?
  - Feature x_i never occurs in documents labeled y
  - But then, the posterior probability $\Pr(y \mid x_i)$ will be 0!!
- Instead, smooth the parameters
- Laplace smoothing or Additive smoothing: Add a dummy count
  - $\Pr(x_i \mid y) = (k+1) / (p+n)$; where n is number of features

Take Home Concepts

- Naïve Bayes is a probabilistic model
- Naïve, because it assumes features are independent of each other, given the class label
- For text classification problems, naïve Bayes models typically provide very strong baselines
- Simple model, easy to learn parameters

Applied Text Mining in Python

Naïve Bayes Variations

Two Classic Naïve Bayes Variants for Text

- Multinomial Naïve Bayes
  - Data follows a multinomial distribution
  - Each feature value is a count (word occurrence counts, TF-IDF weighting, …)

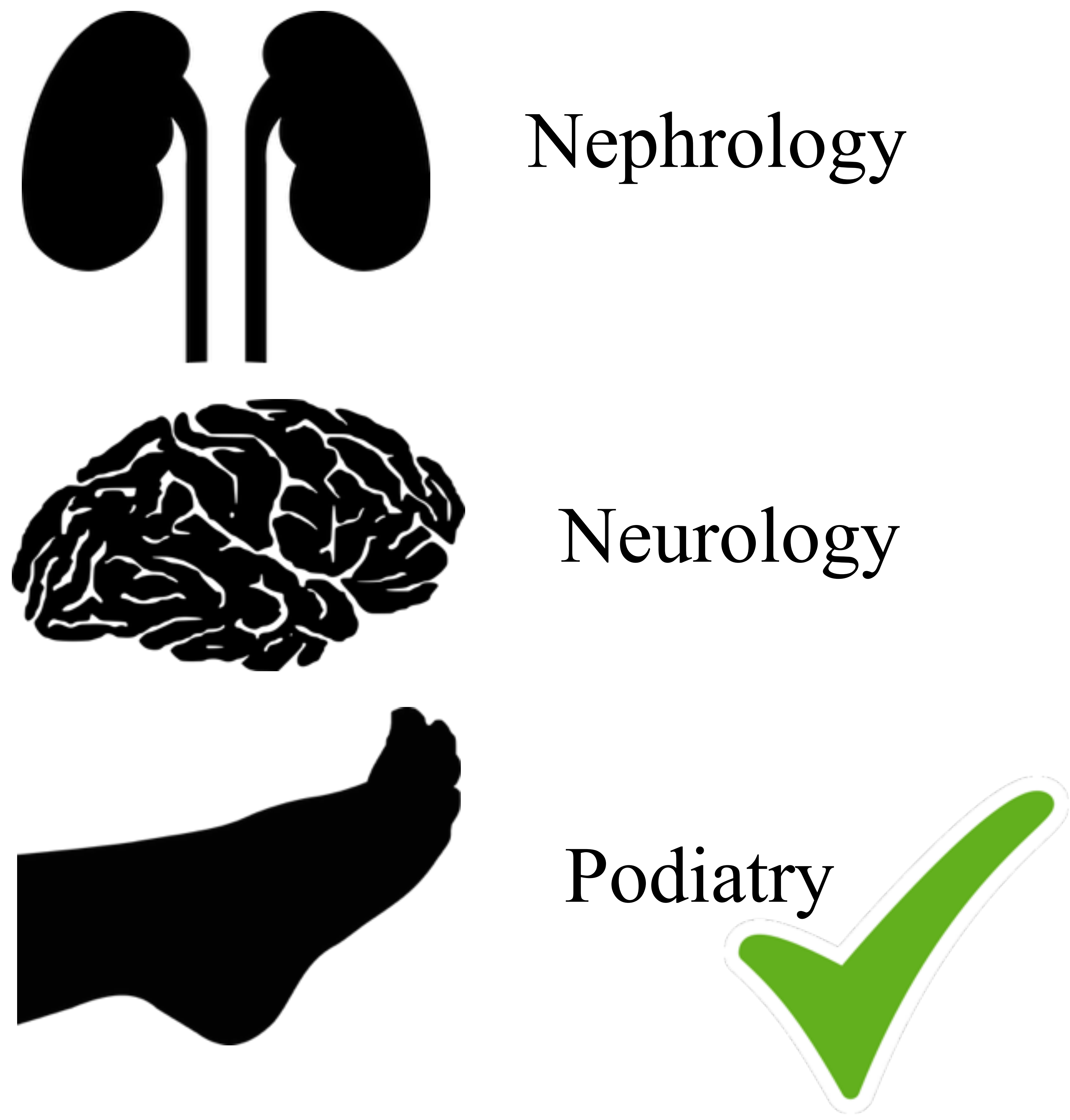
- Bernoulli Naïve Bayes
  - Data follows a multivariate Bernoulli distribution
  - Each feature is binary (word is present / absent)

Applied Text Mining in Python

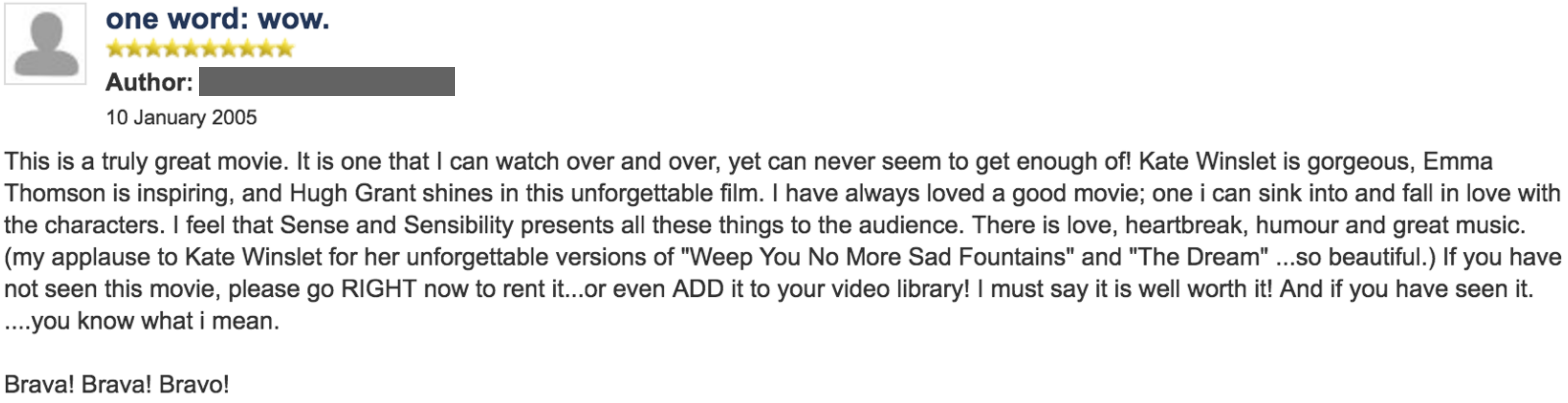
Support Vector Machines

Which medical speciality does this relate to?

TINEA PEDIS, or ATHLETE'S FOOT, is a very common fungal skin infection of the foot. It often first appears between the toes. It can be a one-time occurrence or it can be chronic. The fungus, known as Trichophyton, thrives under warm, damp conditions so people whose feet sweat a great deal are more susceptible. It is easily transmitted in showers and pool walkways. Those people with immunosuppressive conditions, such as diabetes mellitus, are also more susceptible to athlete's foot.



Case study: Sentiment analysis



one word: wow.

★★★★★★★★★★

Author: [redacted]

10 January 2005

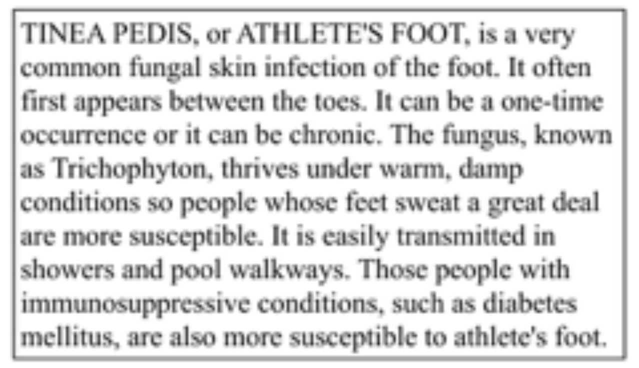
This is a truly great movie. It is one that I can watch over and over, yet can never seem to get enough of! Kate Winslet is gorgeous, Emma Thomson is inspiring, and Hugh Grant shines in this unforgettable film. I have always loved a good movie; one i can sink into and fall in love with the characters. I feel that Sense and Sensibility presents all these things to the audience. There is love, heartbreak, humour and great music. (my applause to Kate Winslet for her unforgettable versions of "Weep You No More Sad Fountains" and "The Dream" ...so beautiful.) If you have not seen this movie, please go RIGHT now to rent it...or even ADD it to your video library! I must say it is well worth it! And if you have seen it. ....you know what i mean.

Brava! Brava! Bravo!

- Words that you might find in typical reviews
  - wow, great, Bravo!
  - boring, lame, worst

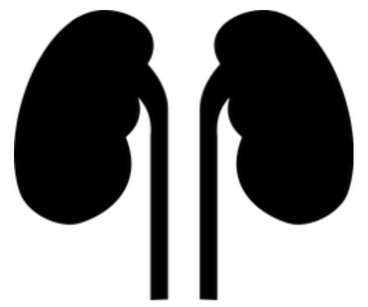
Classifier = Function on input data

- f (



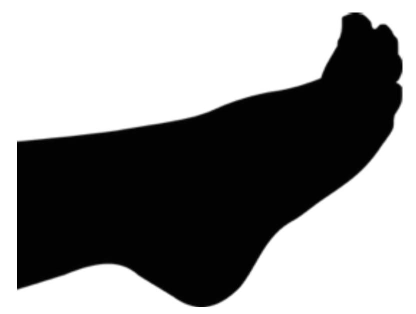
TINEA PEDIS, or ATHLETE'S FOOT, is a very common fungal skin infection of the foot. It often first appears between the toes. It can be a one-time occurrence or it can be chronic. The fungus, known as Trichophyton, thrives under warm, damp conditions so people whose feet sweat a great deal are more susceptible. It is easily transmitted in showers and pool walkways. Those people with immunosuppressive conditions, such as diabetes mellitus, are also more susceptible to athlete's foot.

) $\longrightarrow$ {



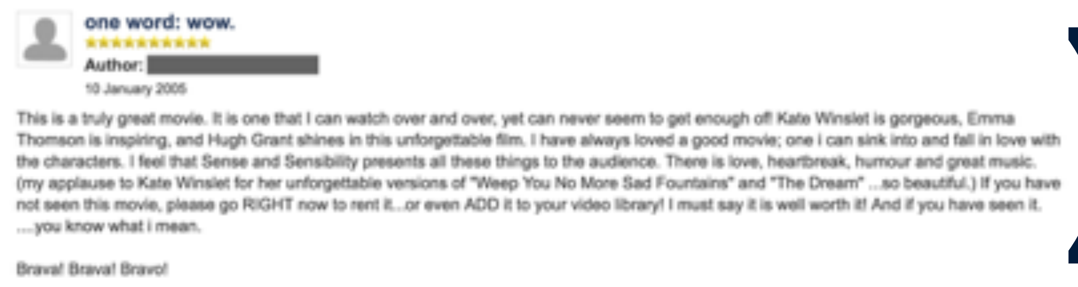
,

,



}

- g(



one word: wow.

★★★★★★★★★★

Author: [redacted]

10 January 2005

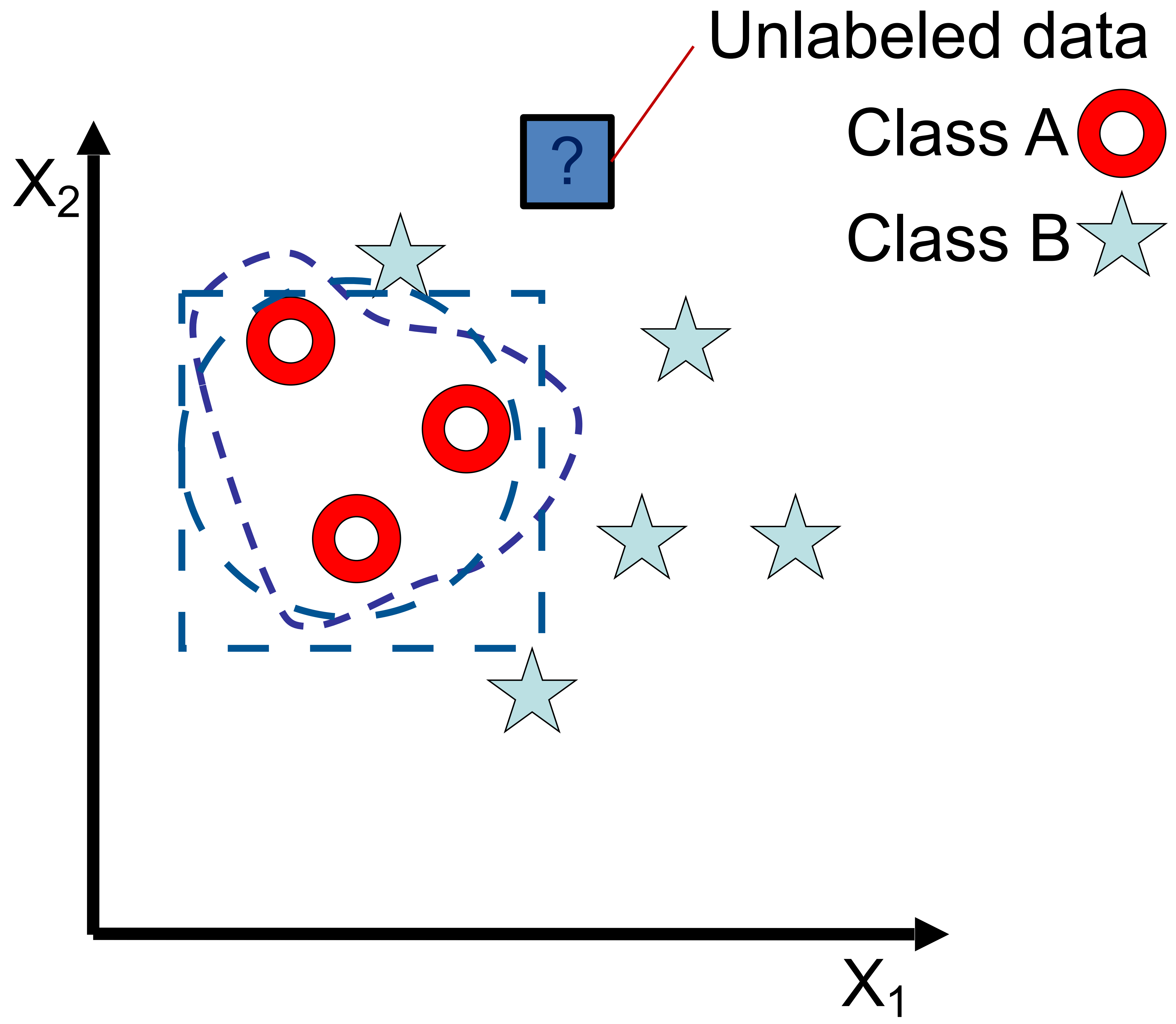
This is a truly great movie. It is one that I can watch over and over, yet can never seem to get enough of! Kate Winslet is gorgeous, Emma Thomson is inspiring, and Hugh Grant shines in this unforgettable film. I have always loved a good movie; one i can sink into and fall in love with the characters. I feel that Sense and Sensibility presents all these things to the audience. There is love, heartbreak, humour and great music. (my applause to Kate Winslet for her unforgettable versions of "Weep You No More Sad Fountains" and "The Dream" ...so beautiful.) If you have not seen this movie, please go RIGHT now to rent it...or even ADD it to your video library! I must say it is well worth it! And if you have seen it. ....you know what i mean.

Brava! Brava! Bravo!

) $\longrightarrow$ { +1 , -1 }

Decision Boundaries

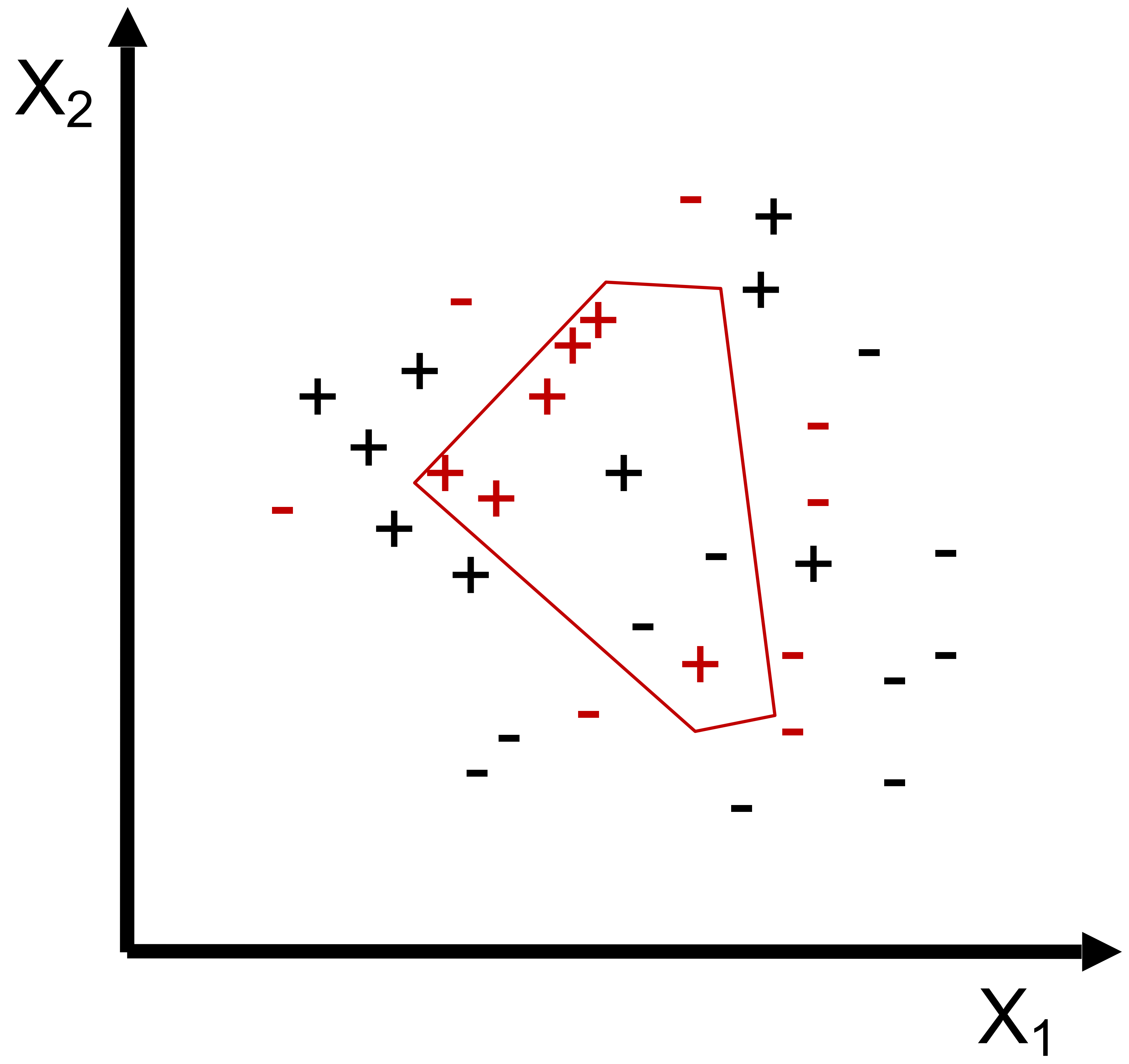
- Classification function is represented by decision surfaces
  - How do you find them?



Choosing a Decision Boundary

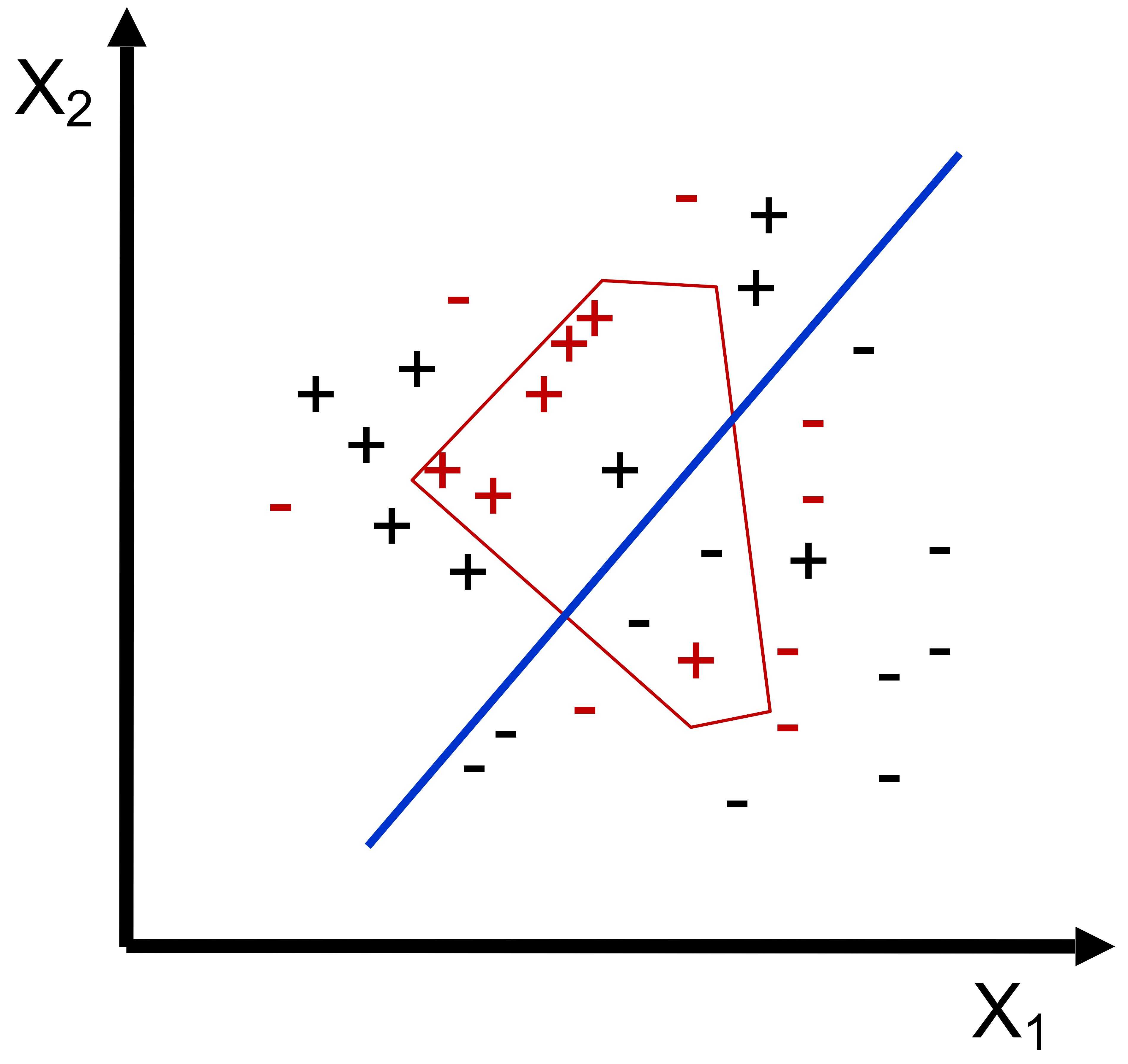
- Red +/- : Training data
- Black +/- : Test data

- Data overfitting: Decision boundary learned over training data doesn't generalize to test data



Linear Boundaries

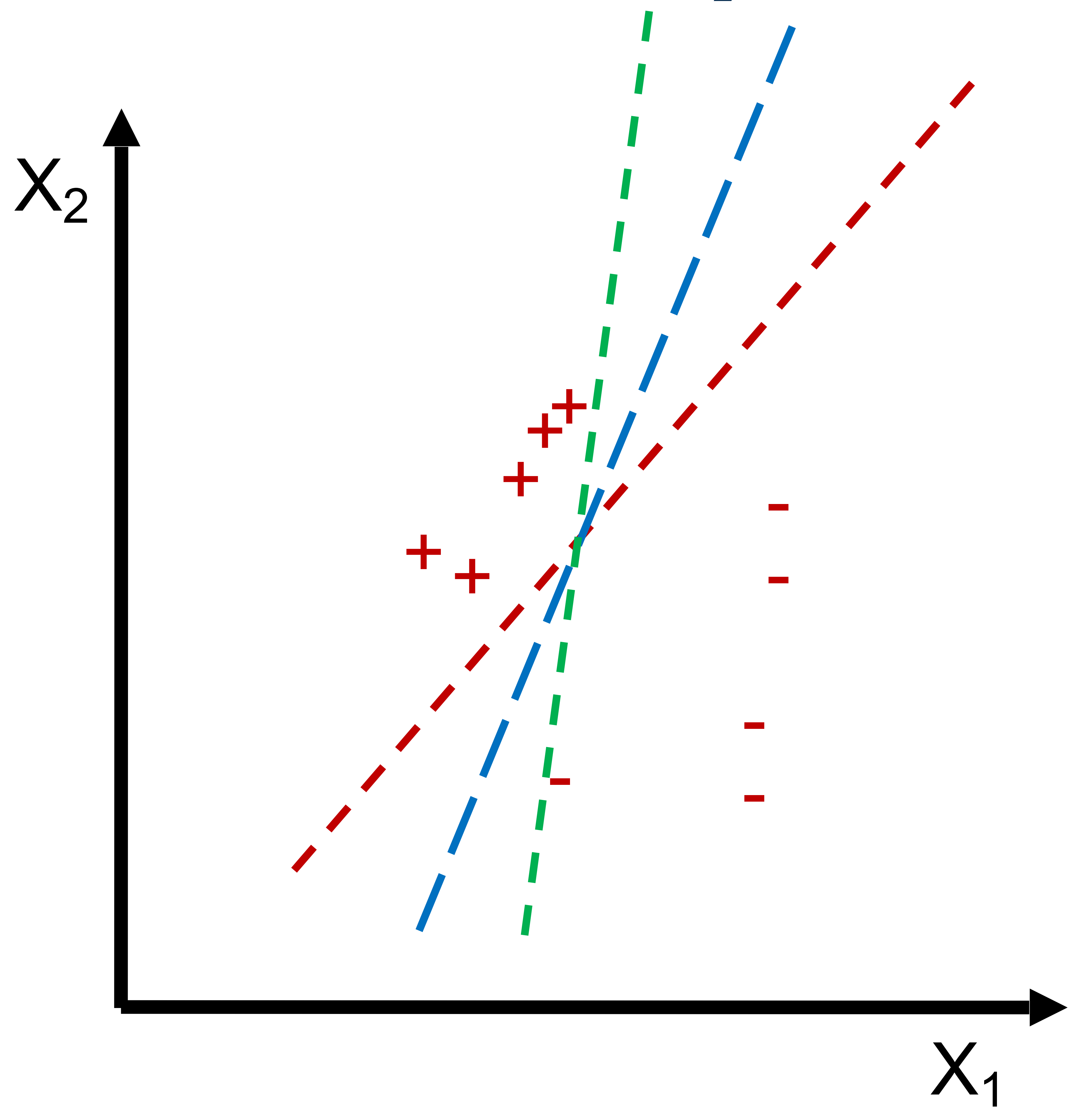
- Easy to find
- Easy to evaluate
- More generalizable: "Occam's razor"



Finding a Linear Boundary

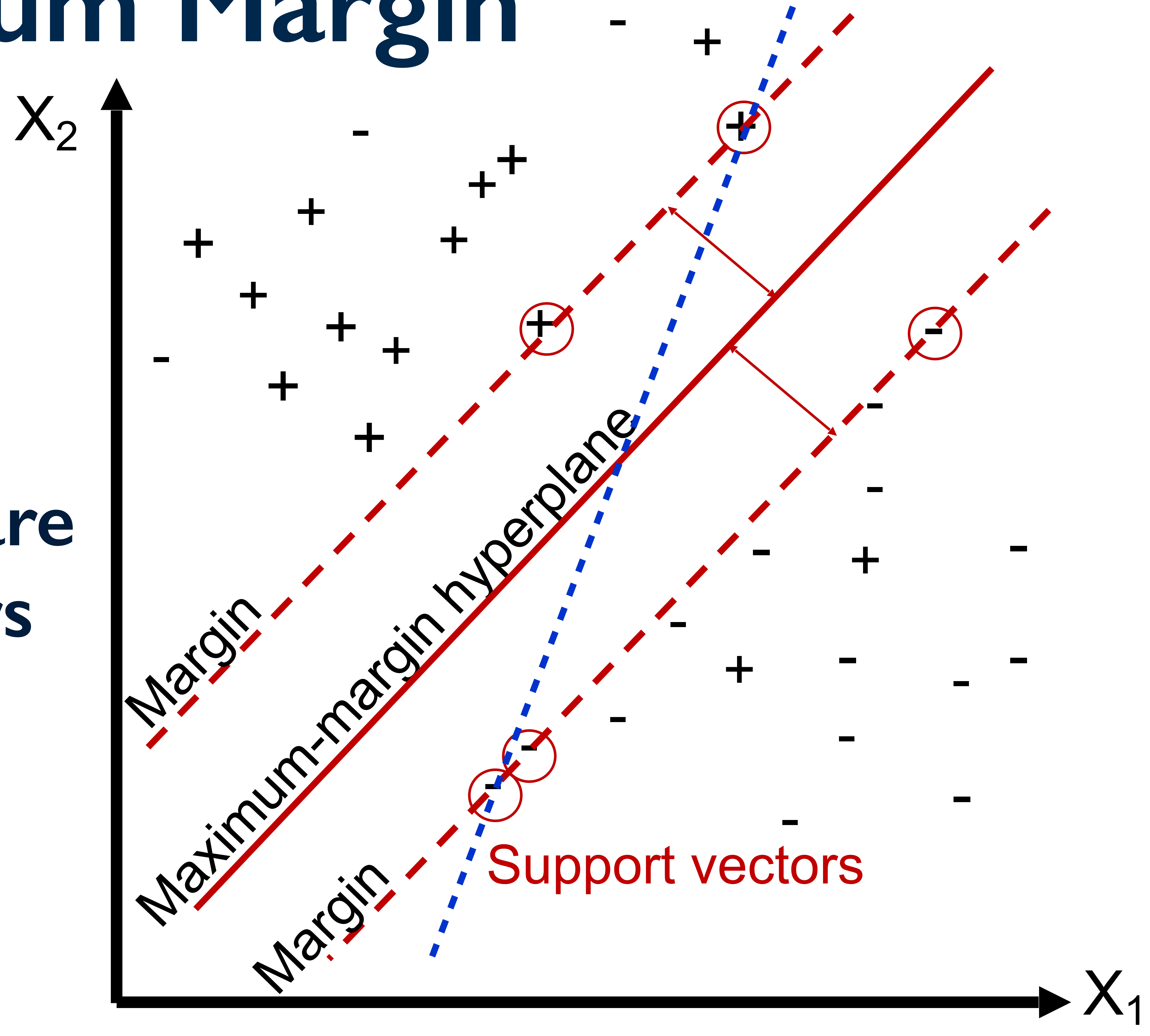
- Find the linear boundary = Find w
- Many methods
  - Perceptron
  - Linear Discriminative Analysis
  - Linear least squares
  - …

- Problem: If linearly separable, then infinite number of linear boundaries!



Maximum Margin

- What is a reasonable boundary?
- Support Vector Machines are maximum-margin classifiers
- How do you find it?



Support Vector Machines (SVM)

- SVMs are linear classifiers that find a hyperplane to separate two classes of data: positive and negative
- Given training data $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots$; where $\mathbf{x}_i = (x_1, x_2, \ldots, x_n)$ is instance vector and y_i is one of $\{-1, +1\}$
  - SVM finds a linear function w (weight vector)

$$f(\mathbf{x}_i) = < \mathbf{w} \cdot \mathbf{x}_i > + b$$

$$\text{if } f(\mathbf{x}_i) \geq 0, y_i = +1; \text{else } y_i = -1$$

SVM: Multi-class classification

- SVMs work only for binary classification problems

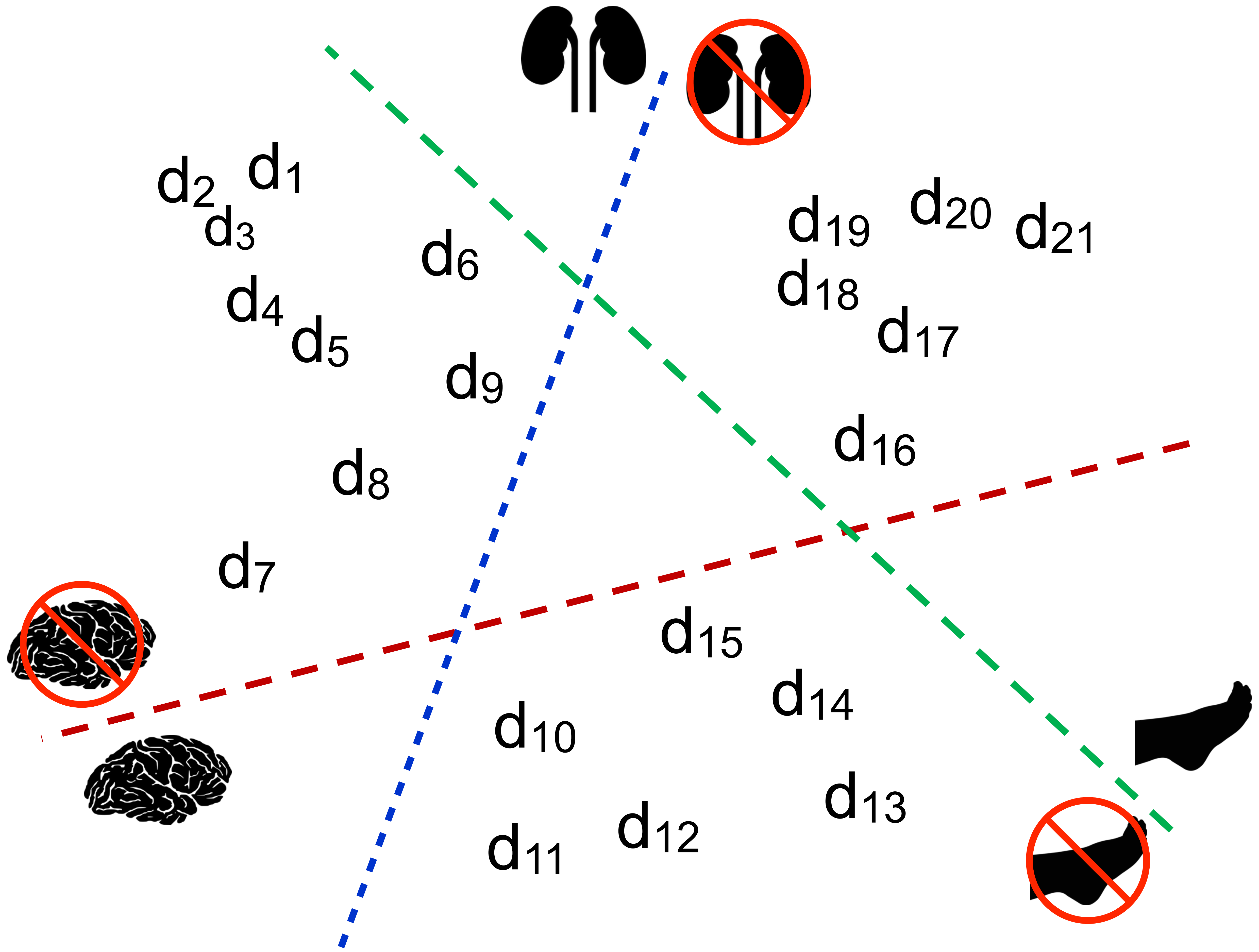
$$f(\mathbf{x}_i) = < \mathbf{w} \,.\, \mathbf{x}_i > + b$$

$$\text{if } f(\mathbf{x}_i) \geq 0, y_i = +1; \text{else } y_i = -1$$

- What about three classes?

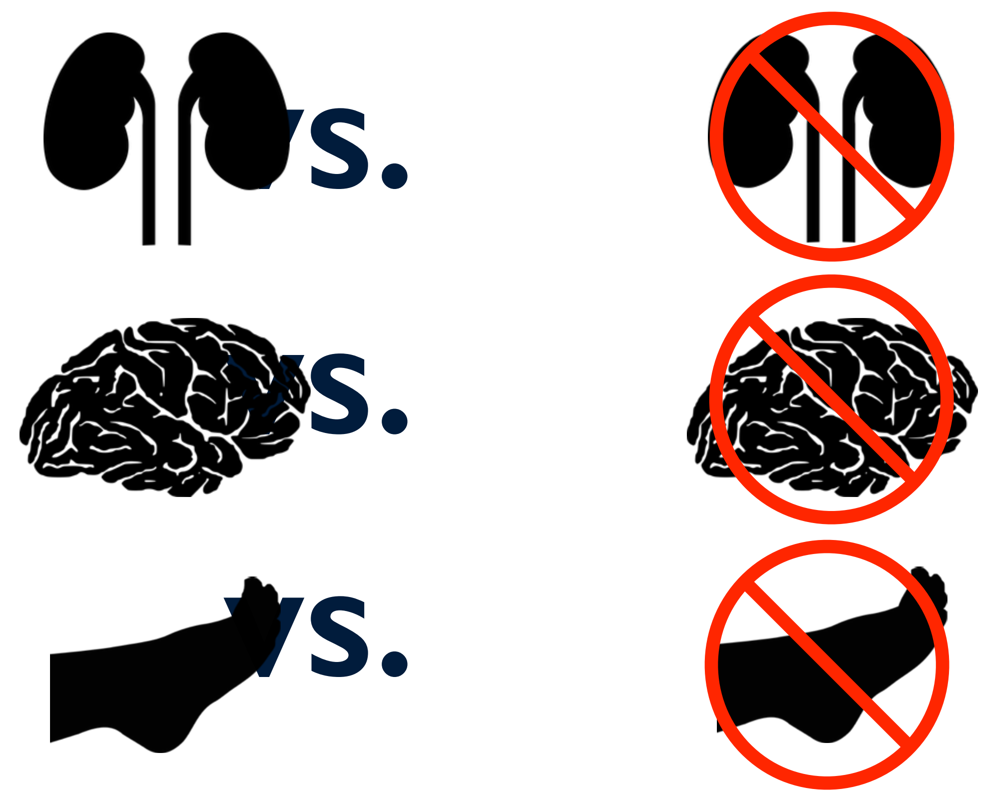
SVM: Multi-class classification

- One vs Rest
  - vs.
  - vs.
  - vs.

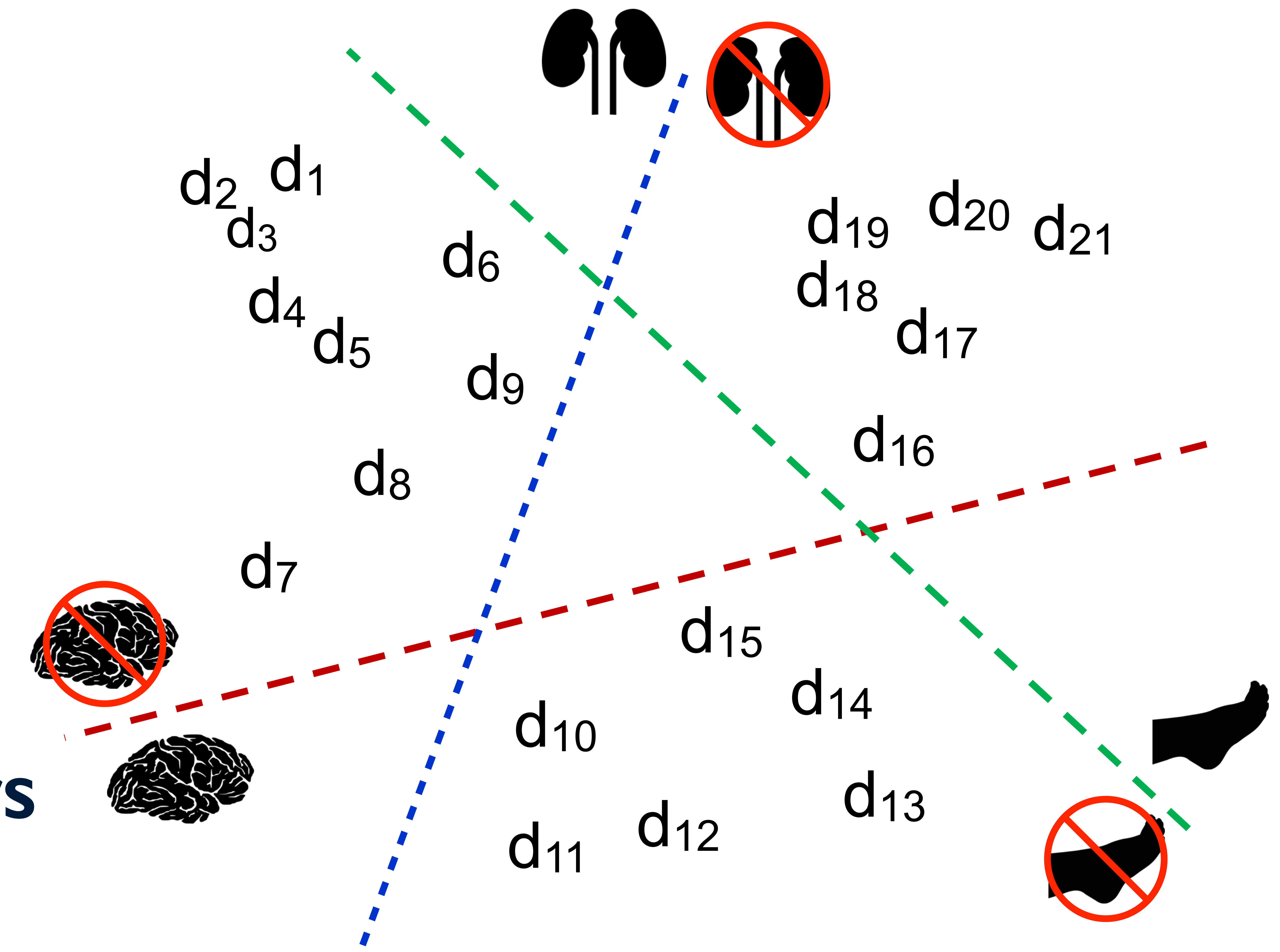


SVM: Multi-class classification

- One vs Rest
  - vs.
  - vs.
  - vs.

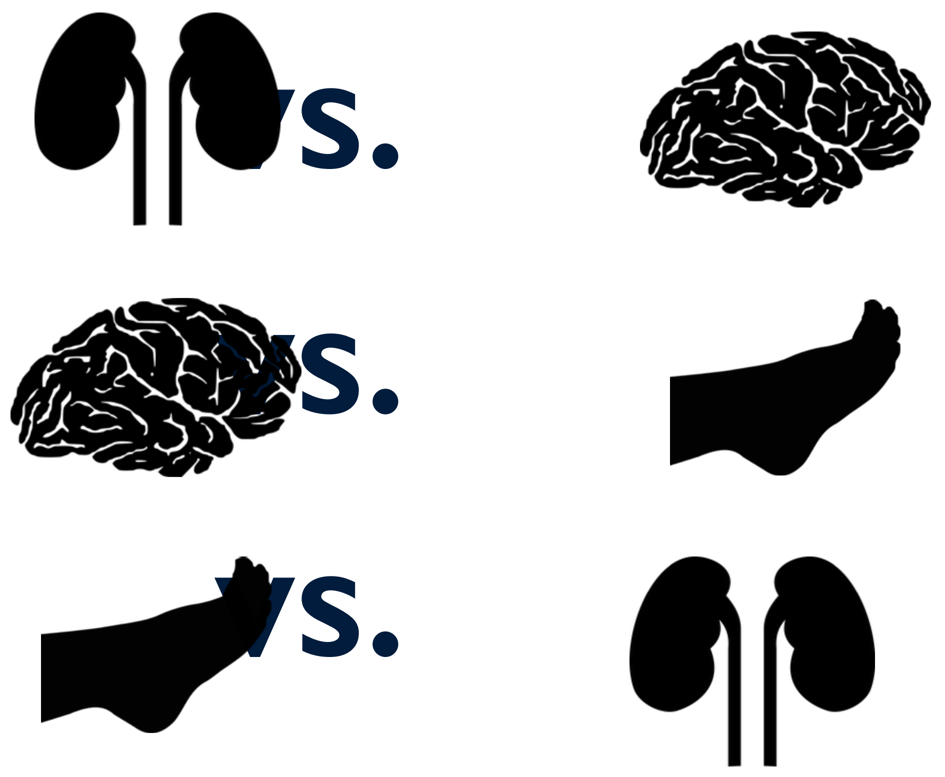


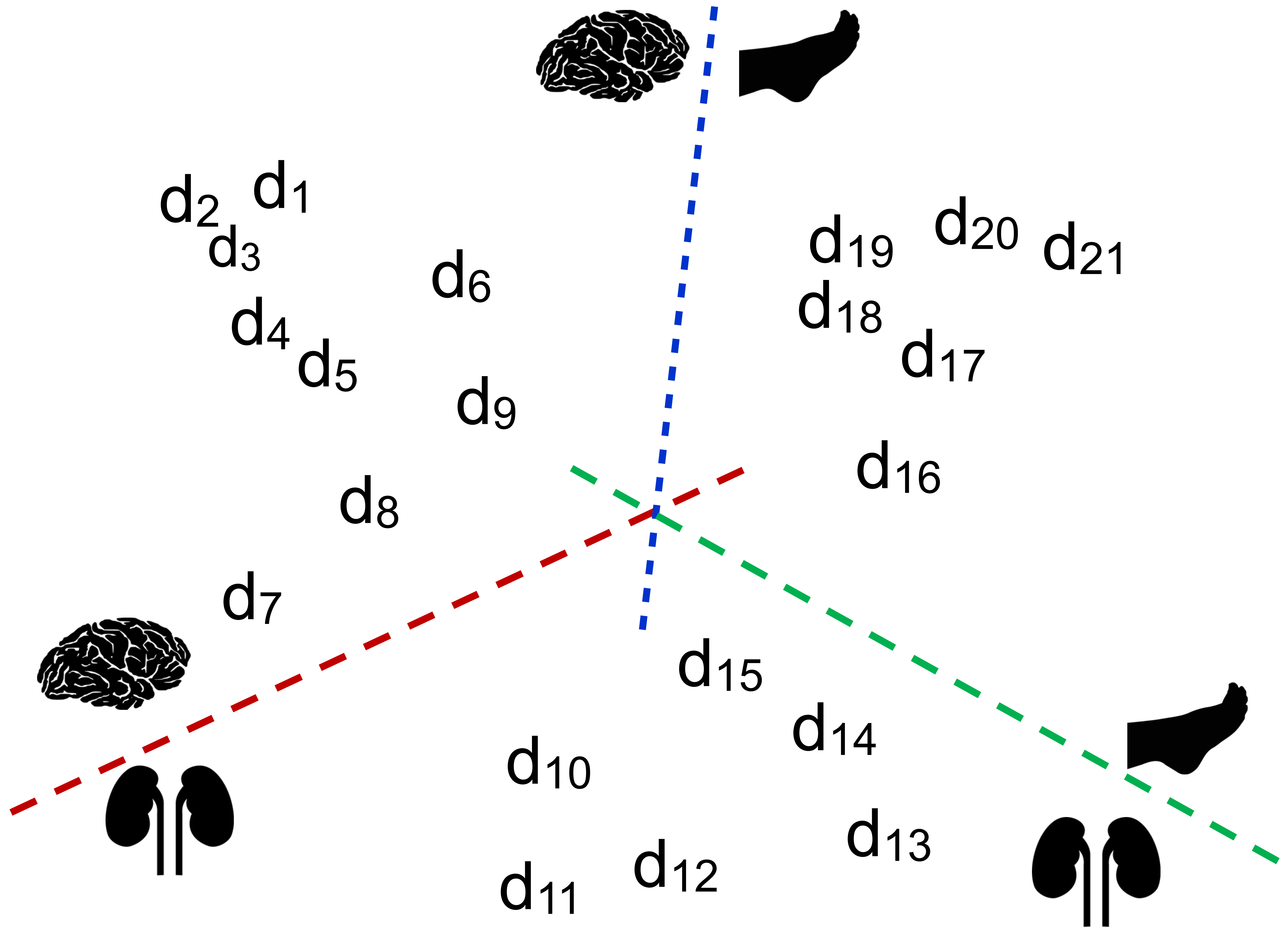
- n-class SVM has n classifiers



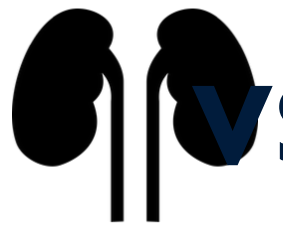
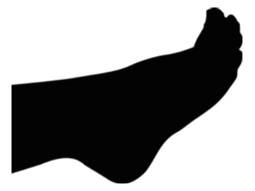
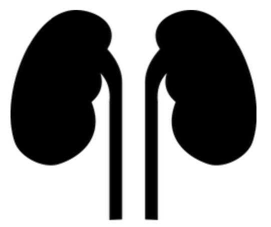
SVM: Multi-class classification

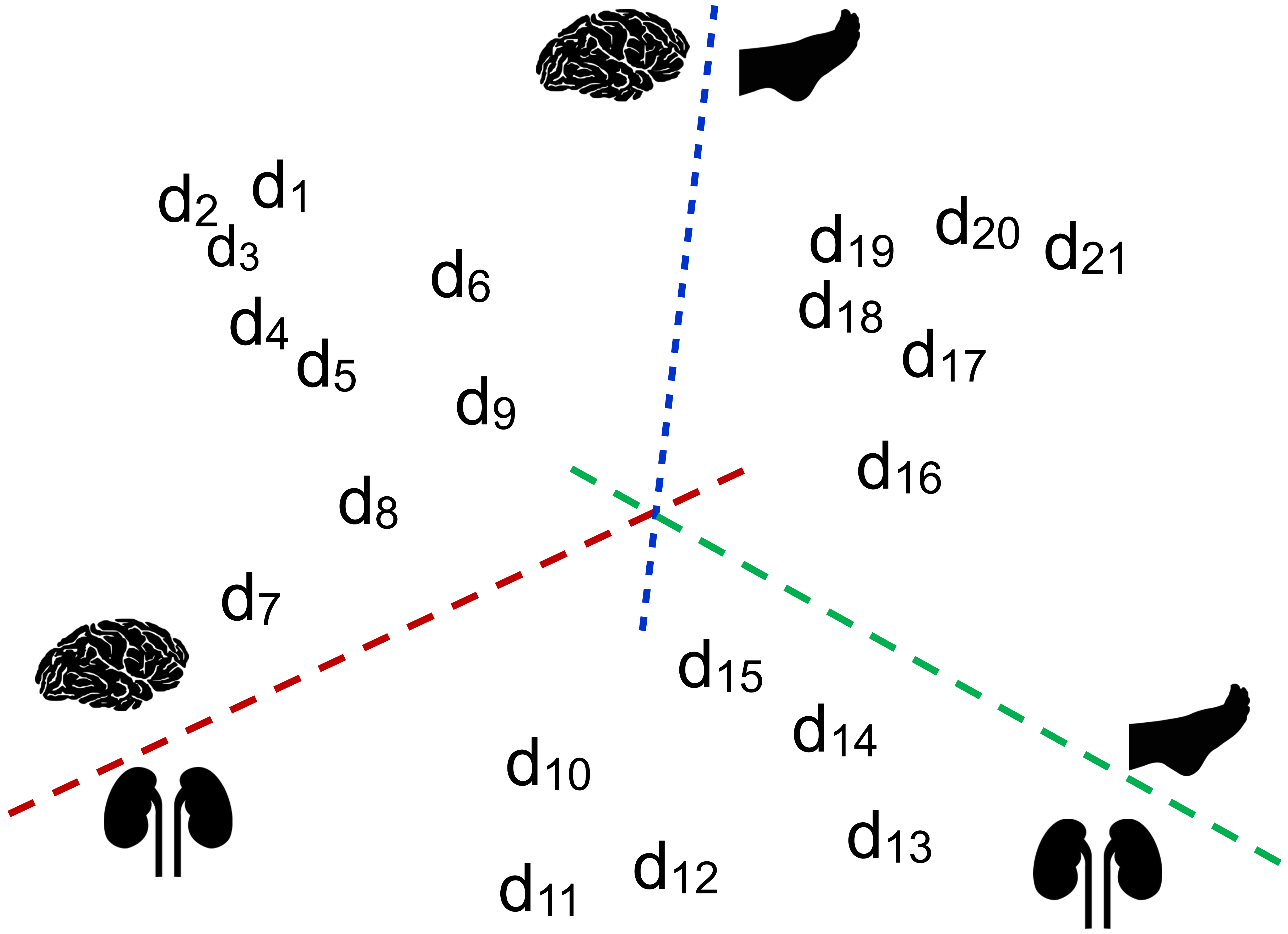
- One vs One
  - vs.
  - vs.
  - vs.





SVM: Multi-class classification

- One vs One
  -  vs.
  - vs. 
  - vs. 
- n-class SVM has C(n,2) classifiers



SVM Parameters (I): Parameter C

- Regularization: How much importance should you give individual data points as compared to better generalized model
- Regularization parameter c
  - Larger values of c = less regularization
    - Fit training data as well as possible, every data point important
  - Smaller values of c = more regularization
    - More tolerant to errors on individual data points

SVM Parameters (2): Other params

- Linear kernels usually work best for text data
  - Other kernels include rbf, polynomial
- multi_class: ovr (one-vs-rest)
- class_weight: Different classes can get different weights

Take Home Messages

- Support Vector Machines tend to be the most accurate classifiers, especially in high-dimensional data
- Strong theoretical foundation
- Handles only numeric features
  - Convert categorical features to numeric features
  - Normalization
- Hyperplane hard to interpret

Applied Text Mining in Python

Learning Text Classifiers in Python

Toolkits for Supervised Text Classification

- Scikit-learn
- NLTK
  - Interfaces with sklearn and other ML toolkits (like Weka)!

Scikit-learn

- Open-source Machine Learning library
- Started as Google Summer of Code by Dave Cournapeau, 2007
- Has a more programmatic interface

Using Sklearn's NaiveBayesClassifier

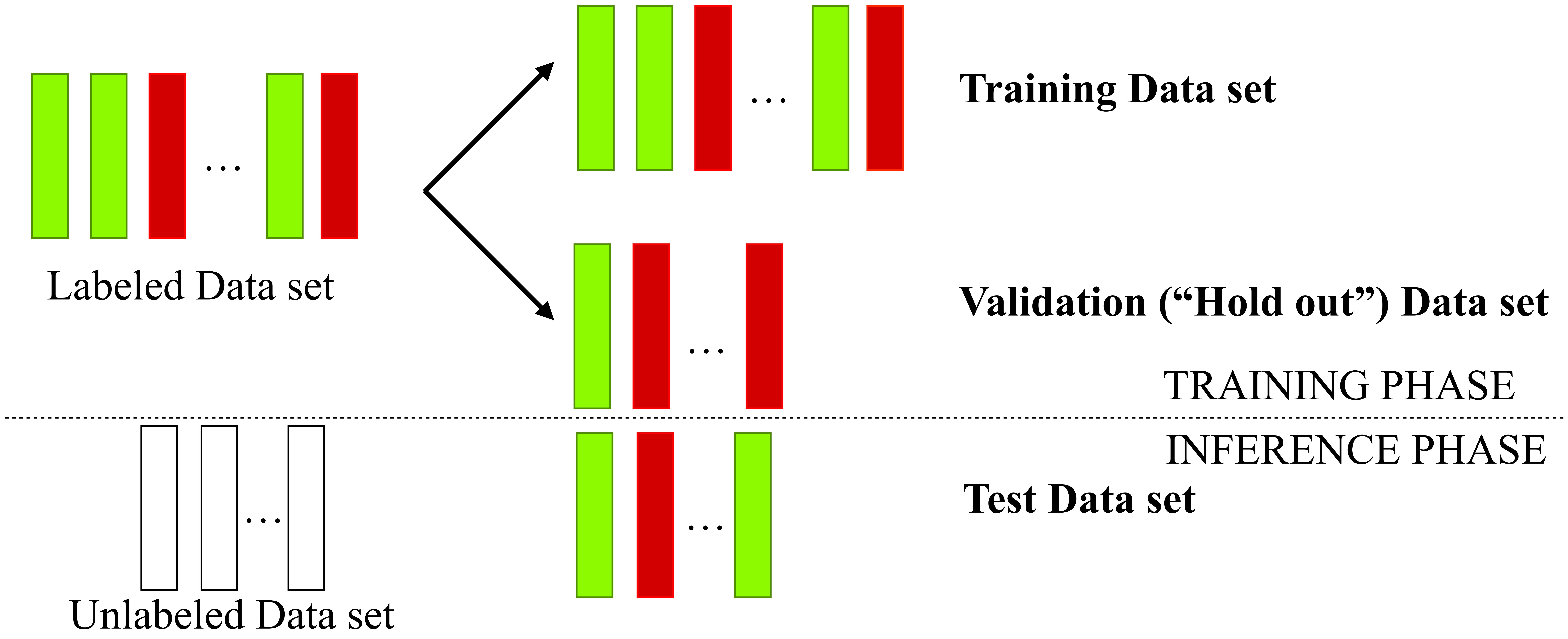
```
from sklearn import naive_bayes

clfrNB = naive_bayes.MultinomialNB()

clfrNB.fit(train_data, train_labels)

predicted_labels = clfrNB.predict(test_data)

metrics.f1_score(test_labels, predicted_labels, average='micro')
```

Using Sklearn’s SVM classifier

```
from sklearn import svm

clfrSVM = svm.SVC(kernel=‘linear’, C=0.1)

clfrSVM.fit(train_data, train_labels)

predicted_labels = clfrSVM.predict(test_data)
```

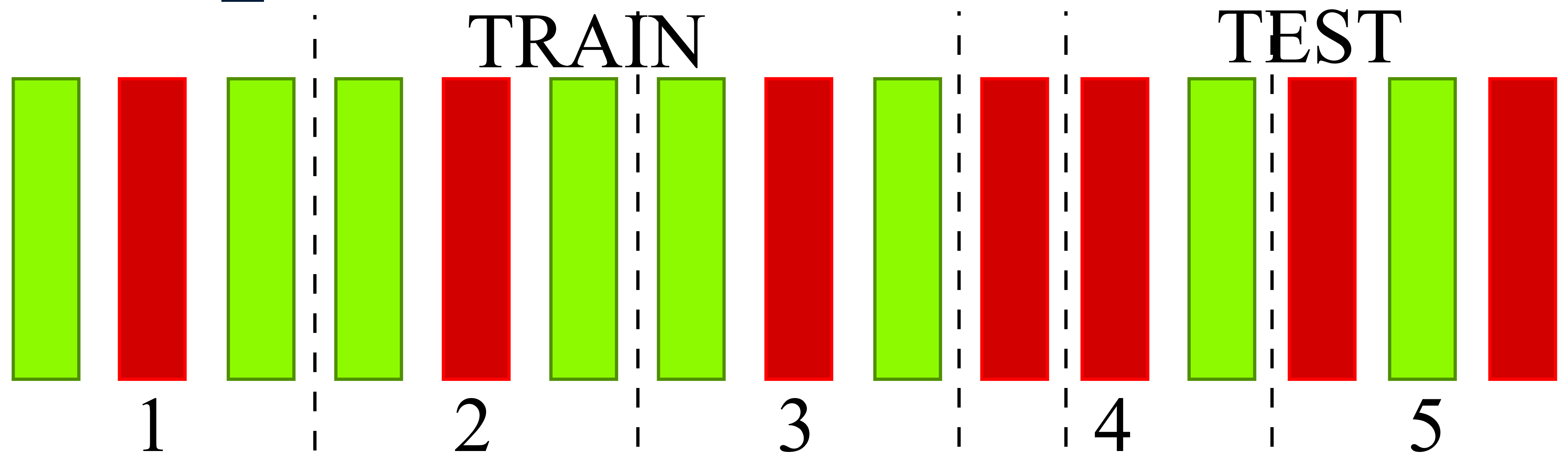
Model Selection

- Recall the discussion on multiple phases in a supervised learning task



Model selection in Scikit-learn

```
from sklearn import model_selection

X_train, X_test, y_train, y_test =
model_selection.train_test_split(train_data, train_labels,
test_size = 0.333, random_state = 0)

predicted_labels = model_selection.cross_val_predict(clfrSVM,
train_data, train_labels, cv=5)
```



Supervised Text Classification in NLTK

- NLTK has some classification algorithms
  - NaiveBayesClassifier
  - DecisionTreeClassifier
  - ConditionalExponentialClassifier
  - MaxentClassifier
  - WekaClassifier
  - SklearnClassifier

Using NLTK's NaiveBayesClassifier

```
from nltk.classify import NaiveBayesClassifier

classifier = NaiveBayesClassifier.train(train_set)

classifier.classify(unlabaled_instance)
classifier.classify_many(unlabeled_instances)

nltk.classify.util.accuracy(classifier, test_set)

classifier.labels()

classifier.show_most_informative_features()
```

Using NLTK’s SklearnClassifier

```
from nltk.classify import SklearnClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import SVC

clfrNB = SklearnClassifier(MultinomialNB()).train(train_set)

clfrSVM =
SklearnClassifier(SVC(),kernel=‘linear’).train(train_set)
```

Take Home Concepts

- Scikit-learn most commonly used ML toolkit in Python
- NLTK has its own naïve Bayes implementation
- NLTK can also interface with Scikit-learn (and other ML toolkits like Weka)