Machine Learning Crash Course: Part I

Ariel Kleiner

August 21, 2012

ampcamp
Big Data Bootcamp

Machine learning
exists at the intersection of
computer science and statistics.

Examples

- Spam filters
- Search ranking
- Click (and clickthrough rate) prediction
- Recommendations (e.g., Netflix, Facebook friends)
- Speech recognition
- Machine translation
- Fraud detection
- Sentiment analysis
- Face detection, image classification
- Many more

A Variety of Capabilities

- Classification
- Regression
- Ranking
- Clustering
- Dimensionality reduction
- Feature selection
- Structured probabilistic modeling
- Collaborative filtering
- Active learning and experimental design
- Reinforcement learning
- Time series analysis
- Hypothesis testing
- Structured prediction

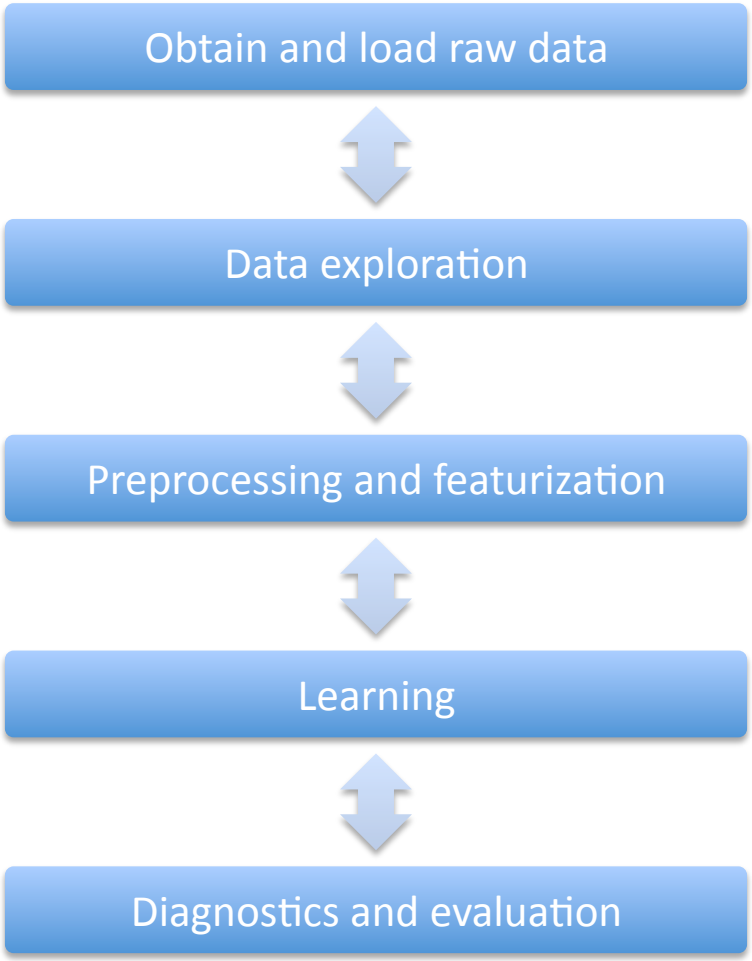
For Today

Classification

Clustering

(with emphasis on
implementability and scalability)

Typical Data Analysis Workflow



Classification

- **Goal:** Learn a mapping from entities to discrete labels.
  - Refer to entities as x and labels as y.

- **Example**: spam classification
  - Entities are emails.
  - Labels are {spam, not-spam}.
  - Given past labeled emails, want to predict whether a new email is spam or not-spam.

Classification

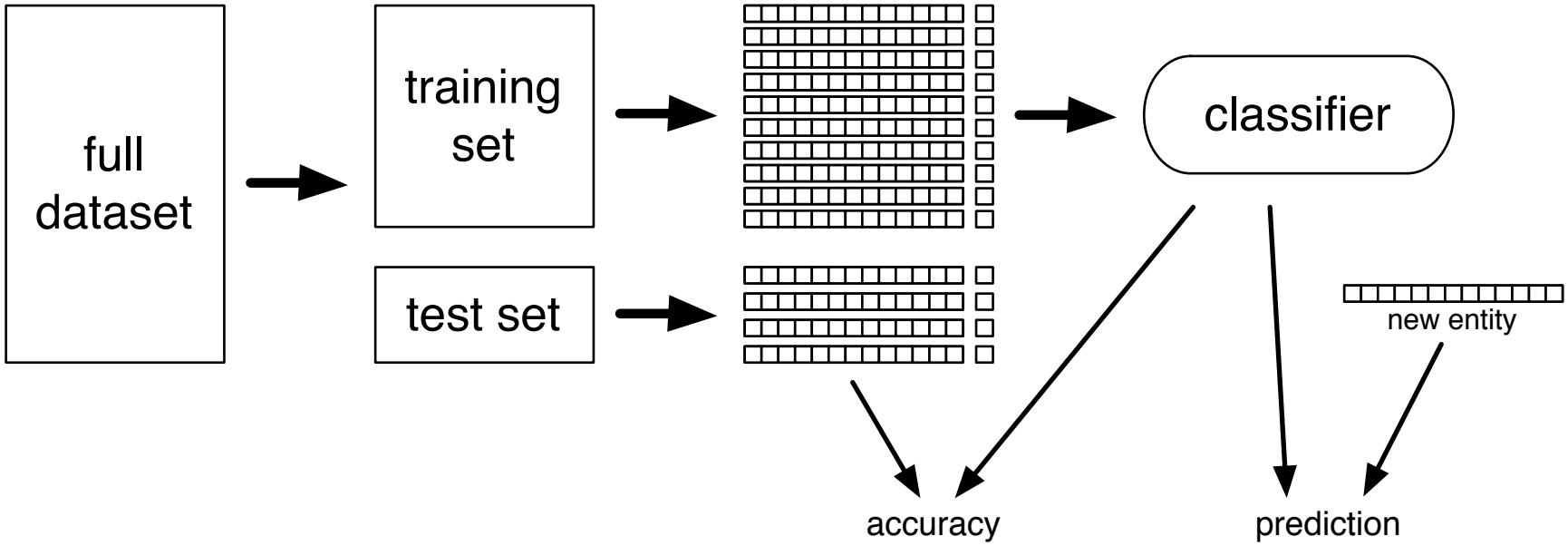
- Examples
  - Spam filters
  - Click (and clickthrough rate) prediction
  - Sentiment analysis
  - Fraud detection
  - Face detection, image classification

Classification

Given a labeled dataset (x_1, y_1), ..., (x_N, y_N):

1. Randomly split the full dataset into two disjoint parts:
   - A larger training set (e.g., 75%)
   - A smaller test set (e.g., 25%)
2. Preprocess and featurize the data.
3. Use the training set to learn a classifier.
4. Evaluate the classifier on the test set.
5. Use classifier to predict in the wild.

Classification



Example: Spam Classification

```
From: illegitimate@bad.com

"Eliminate your debt by
giving us your money..."
```

spam

```
From: bob@good.com

"Hi, it's been a while!
How are you? ..."
```

not-spam

Featurization

- Most classifiers require numeric descriptions of entities as input.
- **Featurization:** Transform each entity into a vector of real numbers.
  - Straightforward if data already numeric (e.g., patient height, blood pressure, etc.)
  - Otherwise, some effort required. But, provides an opportunity to incorporate domain knowledge.

Featurization: Text

- Often use "bag of words" features for text.
  - Entities are documents (i.e., strings).
  - **Build vocabulary**: determine set of unique words in training set. Let V be vocabulary size.
  - **Featurization of a document:**
    - Generate V-dimensional feature vector.
    - Cell i in feature vector has value 1 if document contains word i, and 0 otherwise.

Example: Spam Classification

```
From: illegitimate@bad.com

"Eliminate your debt by
giving us your money..."
```

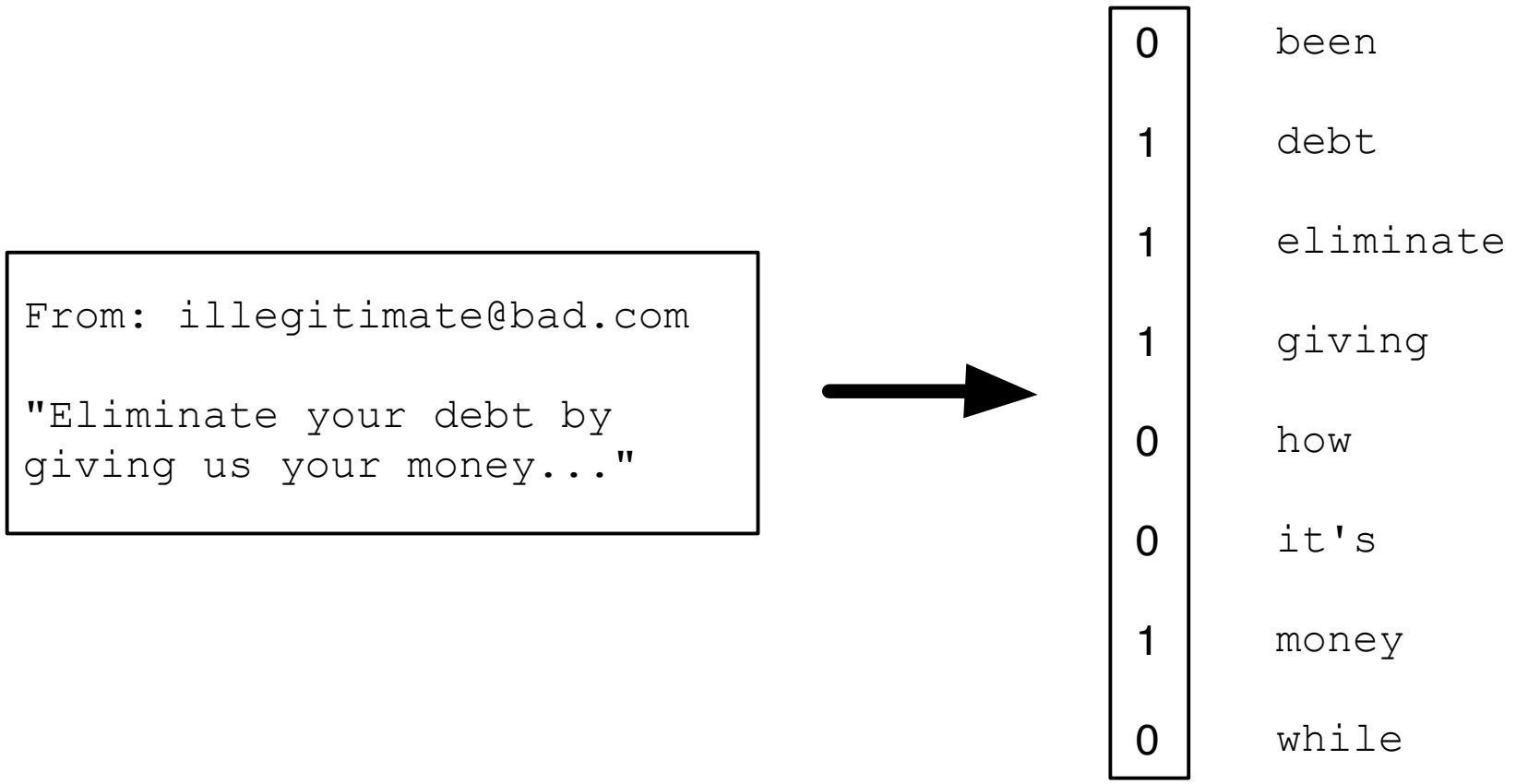
```
From: bob@good.com

"Hi, it's been a while!
How are you? ..."
```

Vocabulary

```
been
debt
eliminate
giving
how
it's
money
while
```

Example: Spam Classification



Example: Spam Classification

- How might we construct a classifier?
- *Using the training data*, build a model that will tell us the likelihood of observing any given (x, y) pair.
  - x is an email's feature vector
  - y is a label, one of {spam, not-spam}
- Given such a model, to predict label for an email:
  - Compute likelihoods of (x, spam) and (x, not-spam).
  - Predict label which gives highest likelihood.

Example: Spam Classification

- What is a reasonable probabilistic model for (x, y) pairs?
- A baseline:
  - Before we observe an email's content, can we say anything about its likelihood of being spam?
  - Yes: p(spam) can be estimated as the fraction of training emails which are spam.
  - p(not-spam) = 1 – p(spam)
  - Call this the "class prior." Written as p(y).

Example: Spam Classification

- How do we incorporate an email's content?
- Suppose that the email were spam. Then, what would be the probability of observing its content?

Example: Spam Classification

- **Example:** "Eliminate your debt by giving us your money" with feature vector (0, 1, 1, 1, 0, 0, 1, 0)
- Ignoring word sequence, probability of email is

  p(seeing "debt" AND seeing "eliminate" AND seeing "giving" AND seeing "money" AND not seeing any other vocabulary words | given that email is spam)
- In feature vector notation:

  $p(x_1=0, x_2=1, x_3=1, x_4=1, x_5=0, x_6=0, x_7=1, x_8=0$ | given that email is spam)

Example: Spam Classification

- Now, to simplify, model each word in the vocabulary independently:
  - Assume that (given knowledge of the class label) probability of seeing word i (e.g., eliminate) is independent of probability of seeing word j (e.g., money).
  - As a result, probability of email content becomes

    $p(x_1=0 \mid \text{spam})\ p(x_2=1 \mid \text{spam})\ \ldots\ p(x_8=0 \mid \text{spam})$

    rather than

    $p(x_1=0, x_2=1, x_3=1, x_4=1, x_5=0, x_6=0, x_7=1, x_8=0 \mid \text{spam})$

Example: Spam Classification

- Now, we only need to model the probability of seeing (or not seeing) a particular word i, assuming that we knew the email's class y (spam or not-spam).
  - But, this is easy!
  - To estimate $p(x_i = 1 \mid y)$, simply compute the fraction of emails in the set {emails in training set with label y} which contain the word i.

Example: Spam Classification

- Putting it all together:
  - Based on the training data, estimate the class prior p(y).
    - i.e., estimate p(spam) and p(not-spam).
  - Also estimate the (conditional) probability of seeing any individual word i, given knowledge of the class label y.
    - i.e., estimate $p(x_i = 1 \mid y)$ for each i and y
  - The (conditional) probability p(x | y) of seeing an entire email, given knowledge of the class label y, is then simply the product of the conditional word probabilities.
    - e.g., p(x=(0, 1, 1, 1, 0, 0, 1, 0) | y) =

      $p(x_1=0 \mid y)\ p(x_2=1 \mid y) \ldots p(x_8=0 \mid y)$

Example: Spam Classification

- **Recall:** we want a model that will tell us the likelihood p(x, y) of observing any given (x, y) pair.
- The probability of observing (x, y) is the probability of observing y, and then observing x given that value of y:

$$p(x, y) = p(y)\, p(x \mid y)$$

- Example:

  p("Eliminate your debt...", spam) =

  p(spam) p("Eliminate your debt..." | spam)

Example: Spam Classification

- To predict label for a new email:
  - Compute log[p(x, spam)] and log[p(x, not-spam)].
  - Choose the label which gives higher value.
  - We use logs above to avoid underflow which otherwise arises in computing the p(x | y), which are products of individual $p(x_i \mid y) < 1$:

$$\log[p(x, y)] = \log[p(y)\, p(x \mid y)]$$
$$= \log[\, p(y)\, p(x_1 \mid y)\, p(x_2 \mid y) \ldots]$$
$$= \log[p(y)] + \log[p(x_1 \mid y)] + \log[p(x_2 \mid y)] + \ldots$$

Classification: Beyond Text

- You have just seen an instance of the **Naive Bayes** classifier.
- Applies as shown to any classification problem with binary feature vectors.
- What if the features are real-valued?
  - Still model each element of the feature vector independently.
  - But, change the form of the model for $p(x_i \mid y)$.

Classification: Beyond Text

- If x_i is a real number, often model $p(x_i \mid y)$ as Gaussian with mean μ_{iy} and variance σ_{iy}^2:

$$p(x_i|y) = \frac{1}{\sigma_{iy}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i - \mu_{iy}}{\sigma_{iy}}\right)^2}$$

- Estimate the mean and variance for a given i,y as the mean and variance of the x_i in the training set which have corresponding class label y.
- Other, non-Gaussian distributions can be used if know more about the x_i.

Naive Bayes: Benefits

- Can easily handle more than two classes and different data types
- Simple and easy to implement
- Scalable

Naive Bayes: Shortcomings

- Generally not as accurate as more sophisticated methods (but still generally reasonable).
- Independence assumption on the feature vector elements
  - Can instead directly model p(x | y) without this independence assumption.
- Requires us to specify a full model for p(x, y)
  - In fact, this is not necessary!
  - To do classification, we actually only require p(y | x), the probability that the label is y, given that we have observed entity features x.

Logistic Regression

- **Recall:** Naive Bayes models the full (joint) probability p(x, y).
- But, Naive Bayes actually only uses the conditional probability p(y | x) to predict.
- Instead, why not just directly model p(y | x)?
  - **Logistic regression** does exactly that.
  - No need to first model p(y) and then separately p(x | y).

Logistic Regression

- Assume that class labels are {0, 1}.
- Given an entity's feature vector x, probability that label is 1 is taken to be

$$p(y = 1|x) = \frac{1}{1 + e^{-b^T x}}$$

  where b is a parameter vector and $b^T x$ denotes a dot product.
- The probability that the label is 1, given features x, is determined by a weighted sum of the features.

Logistic Regression

- This is liberating:
  - Simply featurize the data and go.
  - No need to find a distribution for $p(x_i \mid y)$ which is particularly well suited to your setting.
  - Can just as easily use binary-valued (e.g., bag of words) or real-valued features without any changes to the classification method.
  - Can often improve performance simply by adding new features (which might be derived from old features).

Logistic Regression

- Can be trained efficiently at large scale, but not as easy to implement as Naive Bayes.
  - Trained via maximum likelihood.
  - Requires use of iterative numerical optimization (e.g., gradient descent, most basically).
  - However, implementing this effectively, robustly, and at large scale is non-trivial and would require more time than we have today.
- Can be generalized to multiclass setting.

Other Classification Techniques

- Support Vector Machines (SVMs)
- Kernelized logistic regression and SVMs
- Boosted decision trees
- Random Forests
- Nearest neighbors
- Neural networks
- Ensembles

See *The Elements of Statistical Learning* by Hastie, Tibshirani, and Friedman for more information.

Featurization: Final Comments

- Featurization affords the opportunity to
  - Incorporate domain knowledge
  - Overcome some classifier limitations
  - Improve performance
- Incorporating domain knowledge:
  - Example: in spam classification, we might suspect that sender is important, in addition to email body.
  - So, try adding features based on sender's email address.

Featurization: Final Comments

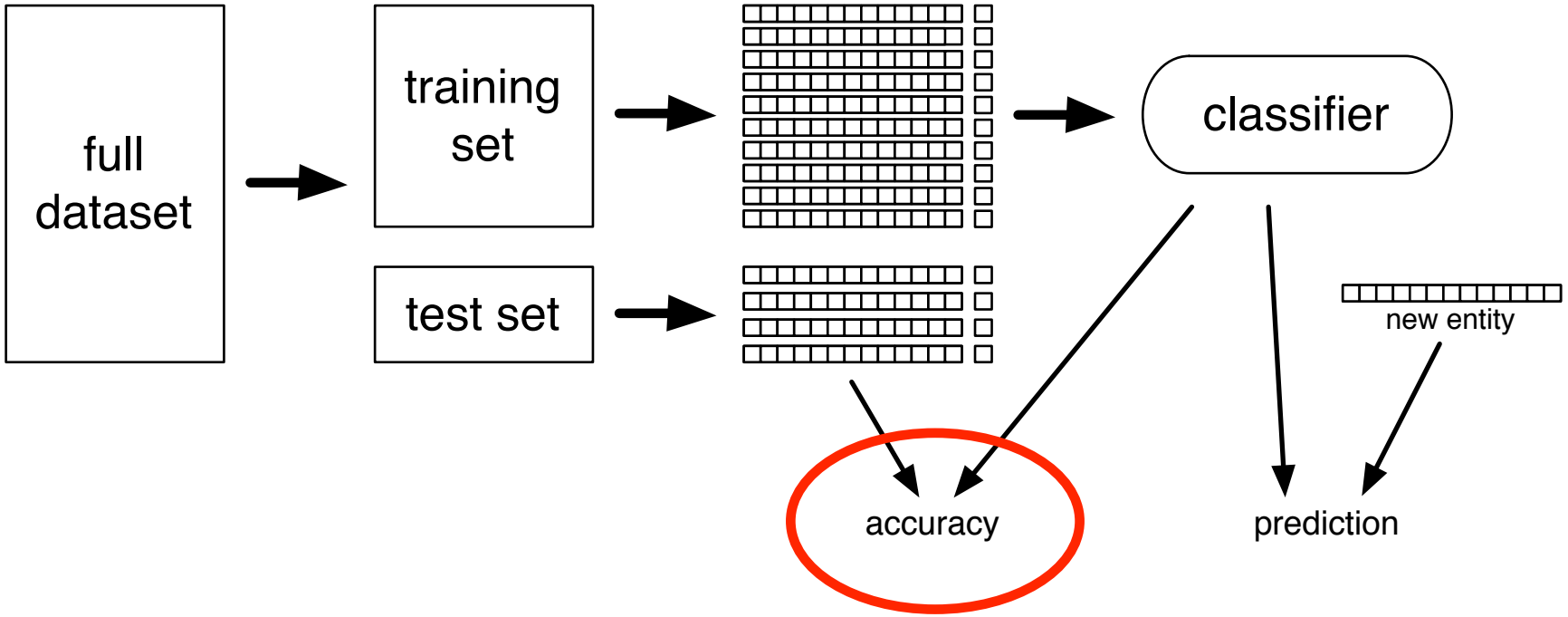
- Overcoming classifier limitations:
  - Naive Bayes and logistic regression do not model multiplicative interactions between features.
  - For example, the presence of the pair of words [eliminate, debt] might indicate spam, while the presence of either one individually might not.
  - Can overcome this by adding features which explicitly encode such interactions.
  - For example, can add features which are products of all pairs of bag of words features.
  - Can also include nonlinear effects in this manner.
  - This is actually what kernel methods do.

Classification

Given a labeled dataset (x_1, y_1), ..., (x_N, y_N):

1. Randomly split the full dataset into two disjoint parts:
   - A larger training set (e.g., 75%)
   - **A smaller test set (e.g., 25%)**
2. Preprocess and featurize the data.
3. Use the training set to learn a classifier.
4. **Evaluate the classifier on the test set.**
5. Use classifier to predict in the wild.

Classification



Classifier Evaluation

- How do we determine the quality of a trained classifier?
- Various metrics for quality, most common is accuracy.
- How do we determine the probability that a trained classifier will correctly classify a new entity?

Classifier Evaluation

- Cannot simply evaluate a classifier on the same dataset used to train it.
  - This will be overly optimistic!
- This is why we set aside a disjoint test set before training.

Classifier Evaluation

- To evaluate accuracy:
  - Train on the training set without exposing the test set to the classifier.
  - Ignoring the (known) labels of the data points in the test set, use the trained classifier to generate label predictions for the test points.
  - Compute the fraction of predicted labels which are identical to the test set's known labels.
- Other, more sophisticated evaluation methods are available which make more efficient use of data (e.g., cross-validation).