supervised vs. unsupervised	labeled training data vs. no labels provided
classification vs. regression	ouputting a class vs. outputting a continuous number
training data vs. testing data	what you use to train the model vs. the data you input to the model once it's trained
“clumpy” vs. “manifoldy”	data that is clustered in feature space vs. spread out smoothly
binomial vs. Bernoulli distribution	probability of k heads in n coin flips vs. special case of binomial distribution with n=1
generative vs. discriminative	model captures p(x,y) vs. model only provides p(y|x)
agglomerative vs. divisive	merging vs. splitting in hierarchical clustering
Bayes rate vs. Base rate	statistical lower bound on achievable error for a classifier & features vs. just using the prior to guess
central vs. pairwise	clustering based on distances of k prototypes to n data points vs. all possible n x n distances, e.g., k-means vs. spectral clustering
lazy learning vs. eager learning	just keep all the training data vs. be smart about what you keep, e.g., K-NN vs. SVM
overfitting and generalization	you can force a model to get zero error on training data but it won't do well on unseen data
sequestered data	because people still 'forget' to keep training and testing data separate, it's good to keep some extra testing data on hand
cross validation	break training data into folds, simulating training/testing splits to prevent overfitting
Bayes’ Rule: prior, likelihood, etc.	how to turn a class conditional density into a posterior probability
covariance matrix	vector counterpart to variance for a scalar random variable, captures spread of data around the mean
mixture of Gaussians	when a single mode won't do
curse of dimensionality	the intuition of distances on 2D examples on the whiteboard evaporates as we go to high numbers of dimensions
loss function	a.k.a. a cost function, the price we pay for a misclassification
extensions to multiclass	methods that leverage a binary classifier to perform k-way classification, e.g., error correcting output codes
the kernel trick	using a Mercer kernel to 'lift' features from the input space into a high (possibly infinite) dimensional space where linear separability is more plausible
cost-complexity tradeoff	you can get a great fit to your data but if the model is very complex you're probably overfitting
regularization	tricks to prevent overfitting
confusion matrix	2D array collecting instances of class i getting classified into class j
type I and type II errors	false positive vs. false negative
ROC curve	plot of type I vs. type II error as a function of threshold on a similarity/dissimilarity score
Precision-Recall curve	similar to ROC curve, more commonly used in document retrieval
softmax function	converts a vector of real numbers into vector of the same length with values between 0 and 1 and that adds up to 1
sigmoid function	a compressive nonlinearity that looks like a smoothed step function commonly used in neural networks
log odds	a.k.a. logit transformation, for a 2-class classifier, log of the ratio of the probabilities of the two outcomes
histogram and bag-of-words	discrete approximation of pdf
χ2 (chi squared) distance	popular method of comparing histograms
dimensionality reduction	expoiting redundancies in high dimensional data to find a smaller number of 'important' dimensions
frequent itemsets	commonly occurring transactions containing a particular set of items
hierarchical clustering	tree-like family of dataset partitions formed by sweeping a similarity threshold
boosting weak learners	creating a strong classifier via a weighted combination of classifiers that perform just better than chance