The second midterm (April 24 in lab) covers in-class material days 11-31 (including Handouts 4-16), labs 4-7, and reading weeks 4-11. You may bring a 1 page (front and back) "cheat-sheet" that was *created by you*, but no other notes or resources. You will not need a calculator. I have put vocab in blue. Although this exam is not explicitly cumulative, look over the "Introduction to Machine Learning" section on the previous study guide for concepts that carry over to the second half.

1. Naive Bayes
   - Bayes rule and how to apply it; idea of conditional probability
   - Bayes rule in ML: identify and explain the evidence, prior, posterior, likelihood
   - Derivation of the Naive Bayes model for $p(y = k|\vec{x})$ (via the Naive Bayes assumption)
   - How can we predict the label of a new point after fitting a Naive Bayes model?
   - How do we estimate the probabilities of a Naive Bayes model?
   - Laplace counts (motivation, application details)
   - Generative vs. discriminative models
   - What types of features/output did we initially require for Naive Bayes?
   - Basic idea of fitting a Gaussian to continuous features (not math)
   - Idea of using normalization to compute the evidence (see Handout 5, clinical trials example)
2. Evaluation Metrics
   - Confusion matrices as evaluation metrics for classification problems
   - Binary classification problems with "positive" (atypical) and "negative" (typical) results
   - Confusion matrices for the above specific case: FP, TP, FN, TN
   - Precision: fraction of flagged examples that are truly positive (intuition: "purity")
   - Recall/TPR: fraction of true positives that we found (intuition: "completeness")
   - Using the FPR and TPR at different thresholds to create a ROC curve
   - What is a "random guessing" ROC curve and what is the ideal ROC curve?
   - Where you want to be on the ROC curve depends on the application
   - Cross-validation: procedure and goals, how to choose the fold number k (LOOCV: $k = n$)
3. Ensemble Methods
   - Idea of using an ensemble of classifiers (ideally all with low bias) to reduce variance
   - To test, let each classifier in the ensemble "vote" (could be weighted or unweighted)
   - Bootstrap: sampling from our data with replacement (usually keeping n the same)
   - Bagging (Bootstrap Aggregation): create a classifier for each bootstrapped training dataset
   - How does averaging the results of many "weak" classifiers reduce the overall error?
   - See Day 17 for ensemble notation and an example of reducing the error via bagging!
   - Random Forest classifiers as ensembles of decision stumps
   - What was the idea behind Random Forests? Why might they be better than regular Bagging?

- AdaBoost: upweight training examples that were classified incorrectly in the previous iter
- AdaBoost details: weighted error, score, update example weights, testing with weighted vote
- Decision Trees with weighted examples (how do we modify the probability calculations?)

4. Support Vector Machines
    - Idea and equation of a separating hyperplane (weight vector points toward the + side)
    - Perceptron algorithm and derivation of the weight updates; perceptron cost function
    - Perceptron weight updates: geometric interpretation and gradient descent interpretation
    - Guarantees and limitations of the perceptron algorithm
    - Support Vector Machines (SVMs) can find the maximum margin hyperplane
    - What are support vectors? What is the geometric (γ) vs. the functional ($\hat{\gamma}$) margin?
    - How we used the functional and geometric margins to cast SVMs as an optimization problem
    - Motivation and method of Lagrange multipliers, application to SVMs
    - High-level steps of transforming the SVM Lagrangian into a problem involving only α values
    - What are these α values and how can we use them to find $\vec{w}^*$ and then b^*?
    - Reformulation of SVMs as maximizing $W(\vec{\alpha})$ uses only inner products between examples
    - Idea of a kernel function and how it can replace the dot product (not Gaussian kernel details)
    - Incremental SVM optimization algorithm for training (with coordinate ascent subroutine)
    - Idea (not details) of soft-margin SVMs and coordinate ascent

5. Neural Networks
    - What is a Neural Network (NN)? Motivation and goals when using them
    - Training NNs using backpropagation (high-level idea is gradient descent on the loss function)
    - Fully Connected architectures, dimensionality analysis, parameters vs. hyperparameters
    - Choice of activation function, pros and cons of sigmoid, tanh, and ReLU
    - Softmax function as the activation function for the last layer, cross-entropy loss after that
    - Training: how to initialize the weights/biases, what is the point of mini-batches?
    - Motivation behind Convolutional Neural Networks (CNNs); application to images
    - CNN architectures: idea of 3D volumes, typical steps CONV, RELU, POOL, FC
    - CONV layer details: filters computing cross-correlations, slide filter over width and height
    - Dimensionality analysis (shapes of filter weights/biases, shapes of input/output via formula)
    - Skip: any pooling besides 2×2 with stride 2, dropout, regularization

6. Unsupervised Learning
    - What is unsupervised learning? When might we use it, what are the goals?
    - Idea of autoencoders as a way to perform unsupervised pre-training of fully connected NNs
    - Very high-level idea of the K-means clustering algorithm (iteratively assign points to clusters based on min distance, then update cluster means)