

The second midterm covers in-class material Days 9–19, Labs 5–8, and reading weeks 5–10. It is not explicitly cumulative, though some topics carry over from the first half of the semester (i.e. **probability, confusion matrices, SGD, runtime, Python implementation skills like OOP, dictionaries, file reading, training/testing, feature importance, etc**). You may use a 1 page (front and back), hand-written resource sheet (created by *you*), but no other notes or resources. I have put vocab in [blue](#).

1. Probability

- Basics of probability and [conditional probability](#)
- [Independence](#) and [conditional independence](#)
- [Bayes rule](#) and how to apply it
- Idea of using normalization to compute the evidence (see Handout 8, page 2)

2. Naive Bayes

- Bayes rule in data science: 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 do we estimate the probabilities of a Naive Bayes model?
- [Laplace counts](#) (motivation, application details)
- How can we predict the label of a new example after fitting a Naive Bayes model?
- What types of features/label do we currently require for Naive Bayes?
- How Naive Bayes can be implemented using [dictionaries](#) in Python

3. Algorithmic Bias and Disparate Impact

- Sample size disparity and how it can impact results
- May need different models for different groups, so a single model is not possible
- General idea that training on past data will recapitulate historical biases
- Problem setup/notation for [redundant encoding](#) of features (X, Y, C)
- Definitions of: [direct vs. indirect discrimination](#), [disparate impact](#)
- Idea of training a classifier to predict X (protected) from Y to detect disparate impact

4. Information Theory

- Conceptual idea of [entropy](#) as well as formal definition
- [Shannon encoding](#) (and decoding), plus how to use entropy to compute average number of bits needed to send one piece of information
- Use of [conditional entropy](#) and [information gain](#) to choose best features
- Comparison with classification accuracy as a way to choose best features
- How to transform continuous features into binary features? (see Handout 14)

5. Logistic Regression

- Motivation for [logistic regression](#); our model is a [logistic function](#) that takes in $\vec{w} \cdot \vec{x}$
- Logistic regression creates a *linear* decision boundary (compute/visualize for $p = 1$)
- In logistic regression our cost is the [negative log likelihood](#) (don't need to derive)
- Intuition/visualization of the cost function (and relationship to [cross entropy](#))
- [Stochastic gradient descent](#) (SGD) for logistic regression, relationship to linear regression
- Interpretation of the weights as feature importance

6. Data Visualization, Dimensionality Reduction, and Unsupervised Learning

- Best ways of visualizing [discrete](#) vs. [continuous](#) data
- How to choose colors; idea of [sequential](#), [diverging](#), or [qualitative](#) color schemes
- How to make color schemes color-blind and black/white printing friendly
- Idea of [principal component analysis \(PCA\)](#) as a way to accomplish [dimensionality reduction](#)
- Using dimensionality reduction to visualize high-dimensional data
- Details of the PCA algorithm (except computing eigenvalues and eigenvectors)
- Runtime of PCA
- Genealogical interpretation of PCA plots for genetic data
- Basic idea of [tSNE](#) as an alternative to PCA
- Idea of [clustering](#) as a form of [unsupervised learning](#), for example [K-means](#)

7. Statistics

- Motivation for studying statistics and [hypothesis testing](#)
- [Probability distributions](#) (discrete vs. continuous)
- Computing (theoretical) [expected value](#) and [variance](#) for discrete distributions
- [Sample mean](#) and [sample variance](#)
- [Central limit theorem \(CLT\)](#) and application in cases where the mean/variance are known
- Computation and interpretation of [Z-scores](#) and [p-values](#)
- [Null vs. alternative hypotheses](#); when to reject the null hypothesis; [significance level \$\alpha\$](#)
- Using [randomized trials](#) and [permutation testing](#) to obtain more precise p-values
- Idea of a [t-test](#) as a way to test differences in means (not details)
- [Bootstrap](#): sampling from our data with replacement (usually keeping n the same)
- How to use bootstrapping to obtain confidence intervals
- [Bagging](#) (Bootstrap Aggregation): create a classifier for each bootstrapped training dataset
- Idea of using an [ensemble](#) of classifiers (ideally with low [bias](#)) to reduce [variance](#)
- To test, let each classifier in the ensemble “vote”