

The first midterm covers in-class material days 1-8, labs 1-4, and reading weeks 1-4. You will have **3 hours** to complete the exam. You may use a 1 page (front and back), hand-written “study sheet” (created by *you*), and a calculator, but no other notes or resources. (You shouldn’t need a calculator, but it may make things quicker – this should be a regular calculator, not a phone or the internet.)

1. Python Topics (*exact syntax will not be required; be able to read code and write pseudocode*)

- Basics of Python style and [top down design](#) (TDD)
- [Object-oriented programming](#) (OOP) in Python
- File reading in Python
- Plotting in Python
- Dictionaries in Python

2. Data Representation and Modeling

- Informal definitions of [Data Science](#)
- Relationship between explanatory variables ([features](#)) and response variable ([label/output](#))
- Common data science notation ( $\mathbf{X}$ ,  $\mathbf{y}$ ,  $n$ ,  $p$ , etc) and matrix/vector representation
- Feature names vs. feature values vs. feature vector
- What is [classification](#)? Understand the [discrete](#) setting of predicting [classes](#) or [categories](#)
- What is [regression](#)? Understand the setting where we predict a [continuous](#) response variable
- [Featurization](#) (e.g. converting categorical features to numerical)
- What is a [model](#)? Why are models useful?
- Understand how a [decision tree model](#) works and can be used for prediction
- What are the [internal nodes](#) of a decision tree? The [edges](#)? The [leaves](#)?

3. Linear Models

- What is a [linear model](#)? What are the *goals* of fitting a linear model to a dataset?
- Using a linear model for prediction
- Notation of linear models (both with and without using matrices/vectors)  $\mathbf{X}$ ,  $\mathbf{y}$ , weights  $\mathbf{w}$
- Goal of minimizing the [RSS](#) (residual sum of squared errors) or [SSE](#) (sum of squared errors)
- [Simple](#) vs. [multiple linear regression](#) (also: why do we add a column of 1’s?)
- [Cost function](#)  $J(\mathbf{w})$  (add  $\frac{1}{2}$  to make derivative work out) and geometric interpretation
- Analytic solution (definition and interpretation) for simple and multiple linear regression
- Idea of [model complexity](#) and that more complex is not necessarily better
- [Polynomial models](#) extend the idea of linear models
- Ways of evaluating polynomial models ([residuals](#), predictions on new data, [elbow plot](#))
- Vector magnitude, [dot product](#); matrix dimensions, multiplication, transpose, inverse, etc

#### 4. Gradient Descent

- General idea of using [gradient descent](#) to minimize any differentiable function
- Mathematical details of moving the weight vector in the opposite direction of the derivative
- [Stochastic gradient descent](#) algorithm for linear regression
- SGD algorithm derivation (gradient computation) and implementation (stopping criteria)
- [Step size](#) (also called learning rate)  $\alpha$  for SGD and pros/cons of high/low  $\alpha$
- Geometric interpretation of gradient descent
- Pros and cons of the analytic solution vs. the SGD solution for linear regression
- Runtime of the analytic solution vs. the SGD solution for linear regression
- Interpreting the final model (most important features, etc)
- [numpy](#) matrix/vector operations (no need to memorize but be able to understand code)

#### 5. Evaluation Metrics

- [Binary classification](#) setting for discussing evaluation metrics
- Single feature decision trees ([decision stumps](#)) as a system of binary classification models
- Creating decision stumps and using them for probabilistic prediction (i.e. with a [threshold](#))
- Understand problems where [positives](#) are rare and [negatives](#) are common
- Goals and process of model evaluation (fit model with [training data](#), test with [testing data](#))
- [Confusion matrices](#) in the binary classification setting
- Classification [accuracy](#) and relationship to classification [error](#)
- [False positives](#), [true positives](#), [false negatives](#), [true negatives](#) (as well as notation)
- [False positive rate](#) (FPR) and [true positive rate](#) (TPR)
- How to vary the classification threshold for a model in order to create a [ROC curve](#)
- ROC curve interpretation and comparison of ROC curves for different models
- [Precision](#) and [recall](#)
- Ethical and practical considerations when selecting the best threshold for an application