## Classical Machine Learning Units

1. Basic Skills
   1. python
   2. matplotlib
   3. pandas, data processing
   4. numpy
   5. Matrix operations
   6. Cover above with various projects in class
2. Measures for model evaluation
   1. Categorical model evaluation - accuracy, precision, recall
   2. Quantitative model evaluation - , sum of squared errors
3. Linear Regression
   1. Mathematical background
   2. Introduce scikit-learn 1D data and multi-dimensional data
   3. Non-linear regression (via data preprocessing)
   4. Subset selection (Forward and Backward)
   5. Lasso and Ridge Regression (regularization techniques – reducing variance and overfitting)
   6. Basic Concepts covered:
      1. Test and training data.
      2. Under and overfitting.
      3. Model evaluation
      4. Possibly ways of dealing with missing data
4. Logistic Regression
   1. First classification problem - contrast with regression
   2. Cover the mathematical model
   3. Introduce AROC curve and “cutoff” values for classification
   4. Introduce the math of the logistic curve and its derivative (which will be used with NN)
5. Data Preprocessing
   1. One hot encoding
   2. Text? TF-IDF? Text embeddings? (would introduce vector cosine distance)
   3. Image formats? (RGB, BGR, conversion) Processing? (sharpen, blur, resample)
   4. Principal Component Analysis
      1. Uses some linear algebra concepts
      2. reduces dimensionality of data
      3. important for large datasets
      4. Project idea: compare PCA with forward/backward subset selection for regression
6. Decision Trees
   1. This is another classification problem
   2. Work with preprocessing categorical data, missing data
   3. Pruning methods (reducing variance and overfitting)
   4. Connect to subset selection and PCA (reducing input dimension)
   5. Random Forests? Other variants?(here or later)
7. Support Vector Machines
   1. Classification by embedding in high dimensional space
   2. Quick overview of math interpretation (splitting data with a plane that maximizes distance between points)
   3. Talk about different kernel functions (provide math for kids who care)
8. Naive Bayes Classifier and k-nearest neighbors (? optinal ?)
   1. Easy to understand and even code from scratch
   2. Gives good baseline on learnability of concepts
   3. Extension: Gaussian Mixture models, etc.
9. Meta-techniques
   1. Ensemble methods (combining models)
   2. Bootstrapping and Boosting
   3. Adaboost, xgboost, others?
10. Things to fit in when applicable above
    1. Cross-validation
    2. Hyperparameter grid search techniques

This is a good 2 quarters maybe more (3-5 lessons per unit, give or take?). Then we do neural net stuff in the spring. This follows (mostly) the order of https://github.com/AET-CS/opencv-machine-learning and lessons are available here for each topic.

Notice once we get going with unit 3, there’s not much real math. Mainly knowing how to interpret results. Some techniques will have optional math explanations. Some (like lasso and ridge regression) are very easy to explain if they understand normal regression, for example.