Chapter 3: Classification

- Intro notes
 - MNIST data set: "Hello world!" of ml classification
 - Some ML algos sensitive to order and perform poorly if many similar instances in a row
 - Shuffling solves that. But shuffling is not good for time series data!
- Training a binary classifier
 - Stochastic Gradient Descent
- Performance Measures
 - o Training a classifier trickier than a regressor
 - o Sometimes need more than sklearn cross-validation → implement own validation
 - Don't use Accuracy for skewed datasets
 - Measuring accuracy using cross-validation
 - Confusion matrix
 - Accuracy: (TP+TN)/(TP + FP + TN + FN)
 - Precision: TP/(TP + FP)
 - Recall: TP/(TP + FN)
 - AKA: Sensitivity, True Positive Rate
 - o Precision and recall
 - F-score: harmonic mean of precision & recall
 - Low scores get more weight
 - Only high F if both precision & recall high
 - $2 * \frac{precision*recall}{precision+recall}$
 - $\bullet \qquad \frac{TP}{TP + \left(\frac{FN + FP}{2}\right)}$
 - Predict bad videos to protect kids → favor high precision
 - Predict shoplifters → favor high recall
 - Precision/Recall Trade-off
 - If out all your positive picks, most were correct: high precision
 - If out of all positive instances, you picked most of them: high recall
 - If most of your positive picks were correct, but your positive picks did not capture many positive instances: high precision/low recall
 - Decision threshold: achieve high precision or recall
 - Sometimes threshold ↑, precision ↓ due to size of data set
 - Recall can only
 ↓ when threshold increases
 → higher requirement to classify as positive
 - Some asks for high precision, respond: "At what recall?"
 - o The ROC Curve
 - True Positive vs. false positive rate
 - FPR = 1 TNR; TNR AKA specificity
 - Sensitivity (TPR) vs. 1 Specificity (TNR)
 - Compare classifiers: Area Under the Curve
 - Perfect = 1, Random = 0.5

- PR or ROC?
 - PR if +ve class is rare or care more about false positives than false negatives
- Multiclass Classification
 - Algos: SGC, Random Forest, Naïve Bayes; Logistic and SVM are binary
 - Ways to train multiclass using binary:
 - One-versus-rest (or all): train as many classifiers as labels, pick the one with highest decision score
 - Preferred for most binary algos
 - One-versus-one: train one classifier for every pair; N classes: N*(N-1)/2 classifiers
 - OvO preferred for large datasets with algos that don't scale well
- Error Analysis
 - o SGD classifier is linear model: assigns weight per class to each data point
 - Seeing a new piece of info, sums up weighted intensities to get a score
- Multilabel classification
 - o Distinguishing multiple objects within one space
 - Evaluate using f1 score (or other binary classifier metric) for each label, then average
 - Assumes all labels equally important
- Multioutput classification
 - o Generalization of multilabel where each label can be multiclass