HM6

* Hand engineering the algorithm is possible (page 11)
* When Training distribution is much different from test distribution, random splits are not helpful (page10)
* Dev set is used to tune parameters (page 11)
* Choose dev and test sets to reflect data expected in the future
* This means not doing a random split, but rather selecting test cases, or defining how data will appear in the future, and building test data from assumptions
* Dev set needs to specifically include things you want better performance on, because the dev team will work on increasing performance on this set
* Dev and test sets must be designed together
* Mainly because after tuning, if the algorithm performs well on the dev set then the team goes back to the test set and the algorithm does bad, this gives a basis for analysis
* 1) The team overfit the dev set because the dev and test set are so similar
* 2) the test set is harder than the dev set
* 3) the test set might be as similar to the dev set as possible, but it’s still too different to make an algorithm that compromises both sets.
* Sizing (page 19)
* Dev large enough to detect difference between algorithms, 1,000 to 10,000
* It’s good to use single number evaluation metrics
* The metrics are used to evaluate different approaches for tuning on the dev set.
* The evaluation metric tells how well a certain approach performs relative to other approaches
* Signs to change the dev set
* The actual distribution you need to do well is different between dev and test
  + Keep a clear idea of what you expect from the future
* You have overfit the dev set
  + Solution, evaluate against test set regularly, roll back changes on the current week if test set does worse
* The metric is measuring something other than what the project needs to optimize
  + Choose a new metric to achieve a certain goal, do not choose a metric by looking at classifiers
* Error analysis stems from building a system and iterating around clues given by the system
* Estimate accuracy savings before undertaking error analysis project
* Some errors overlap, so you can eliminate in parallel
* Partitioning the dev set helps with manual review of misclassification
* Eyeball set is for viewing results and tuning
* Blackbox set is for identifying when you overfit the eyeball set
* Adding more data to the training set can reduce the variance
* Scaling variables to a consistent size also helps
* Variance stems from too many features
* Features can be trimmed during error analysis from observations noticed
* Human level performance is a good metric because data is easy obtains from human labellers
* Humans can not overfit a model with data, because our intuition is not good enough