Classification Model Pros and Cons (Generalized)

* Logistic Regression
  + Pros
    - low variance
    - provides probabilities for outcomes
    - works well with diagonal (feature) decision boundaries
    - NOTE: logistic regression can also be used with kernel methods
  + Cons
    - high bias
* Decision Trees
  + Regular (not bagged or boosted)
    - Pros
      * easy to interpret visually when the trees only contain several levels
      * Can easily handle qualitative (categorical) features
      * Works well with decision boundaries parellel to the feature axis
    - Cons
      * prone to overfitting
      * possible issues with diagonal decision boundaries
  + Bagged Trees : train multiple trees using bootstrapped data to reduce variance and prevent overfitting
    - Pros
      * reduces variance in comparison to regular decision trees
      * Can provide variable importance measures
        + classification: Gini index
        + regression: RSS
      * Can easily handle qualitative (categorical) features
      * Out of bag (OOB) estimates can be used for model validation
    - Cons
      * Not as easy to visually interpret
      * Does not reduce variance if the features are correlated
  + Boosted Trees : Similar to bagging, but learns sequentially and builds off previous trees
    - Pros
      * Somewhat more interpretable than bagged trees/random forest as the user can define the size of each tree resulting in a collection of stumps (1 level) which can be viewed as an additive model
      * Can easily handle qualitative (categorical) features
    - Cons
      * Unlike bagging and random forests, can overfit if number of trees is too large
* Random Forest
  + Pros
    - Decorrelates trees (relative to bagged trees)
      * important when dealing with mulitple features which may be correlated
    - reduced variance (relative to regular trees)
  + Cons
    - Not as easy to visually interpret
* SVM
  + Pros
    - Performs similarly to logistic regression when linear separation
    - Performs well with non-linear boundary depending on the kernel used
    - Handle high dimensional data well
  + Cons
    - Susceptible to overfitting/training issues depending on kernel
* Neural Network (This section needs further information based on different types of NN's)
* Naive Bayes
  + Pros
    - Computationally fast
    - Simple to implement
    - Works well with high dimensions
  + Cons
    - Relies on independence assumption and will perform badly if this assumption is not met