Jacqueline Liu

**Models**

The first model I used was a 'GradientBoostingRegressor', an ensemble of regression trees. Advantages of this model include allowing for different types of features (columns don't need to be normalized or scaled) and detecting nonlinear relationships. One of the major disadvantages is speed; because the trees are built sequentially, the process can't be parallelized easily (vs random forests). GBTs are also prone to overfitting and require careful tuning. Controlling the depth of the tree and learning rate can help generalize the model, as does the use of attribute and data bootstrap aggregating (bagging).

**Python**

Most of changes I made were syntactic and are documented in the comments.

* In printing the names in the 'friends' array, %s should have been used as the placeholder for a string, not %i
* For counting the names with the letter 'a'
  + 'a' is a string when there are quotes around it and is a variable when there aren't
  + Python doesn't support unit increment operators (++) so we make do with += 1
  + Storing the count\_a variable as a float prevents rounding when doing integer division
  + Multiplying the decimal by 100 makes it a percentage
  + Adding ‘.1’ to the %f argument formats the percentage to show one decimal place
* In printing hi to all friends, default arguments (greetings) go after non-default ones (name)
* In printing sorted names, the call friends.sort() sorts the array in place and returns none. Either call the sort function first and print friends second, or use the sorted(friends) function that returns the sorted array while leaving the original untouched
* The factorial function needed to multiply the recursive call by the current number
  + Also added a check for numbers less than 1 to prevent infinite loops during bad calls