Noah Lutz
Perm: 8026163
noahlutz@ucsb.edu
CS 165A
MP1 Report

Architecture:

My main architecture is my naiveBayes class. This class contains all the functionality of the project. The class is defined in a header file and a cpp file with the definitions of all the functions. It has the main functions of fit and predict, which takes in files. Fit function fits the data updating the needed private variables and vectors, and predicts outputs to the console based on the given inputted file. The constructor sets most of the private variables; which are mostly vectors, counters, and probability scores; to empty or zero. There are also 4 helper functions: find_means, find_variances, conditional_prob, and split_into_classes. Once fit(file) spareses the file and stores the data into a vector of vectors, each being a feature, split_into_classes() is called which takes all that data and splits it into new vectors based on whether it was a 1 or a 0 target. Then find_means and find_variances are called and stored to get the mean and variance of each feature, split up between 1 and 0 targets. Finally in fit the probability of getting a target 1 or 0 is calculated. All these data points like means, variance, and probability are stored in private vectors or double variables in the class. In the predict function, the file is sparased like before and then probability of the if it is a 0 and 1 is calculated using the conditional_prob helper to find the conditional probability for a given feature. ‘

Preprocessing:

The data comes in and each line is broken up. It is converted into a double value from a string, including a check for gender in which M = 1.0 and F = 0.0. Each value is then stored in a vector with all other features brought in. The index indicates its position so for the gender vector the item at the 5th index correlates to the same person for the weight vector's 5th index. These vectors themselves are stored in a vector, of which that index indicates what feature it is. For example gender is the second feature so it is at index 1 of the big vector.

Model Building:

To fit the data, after collecting all the given training data points and split them into 2 groups: 0 target and 1 target. So all the values of target 1’s are grouped and same for target 0’s. Then of each group I found the mean and variance of each feature. So there was a Target 1 age mean and variance and Target 0 ages mean and variance and so on for each feature. Using the means and variance can calculate the conditional probability for a given feature using the normal distribution function.

Normal Probability Density Function

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2 / 2\sigma^2}$$

Results:

I got an accuracy of 75.616% in 0.14 time.

Challenges:

Biggest challenge was keeping track of all the data and properly accessing the correct vectors. There were multiple vectors in vectors so properly indexing them I messed up a few times. To ease this I did replace one of the vectors of vectors with a list of vectors since I knew it would be a set size for the list. Also ran into a point where I forgot to store a feature and ran into the issue of trying to find the bug and eventually realized I had a 10 instead of 11.

Weaknesses:

There is probably a better way to store the data than all the vectors I was using. Furthermore my preprocessing was minimal to none, there probably is a better way to handle the binary feature like gender. For storing the data, a Map may have been useful, making it easier to track and access it. For the preprocessing probably accountant and calculate the effect of the gender differently than just finding the mean and variance and plugging it into the equation like everything else. Maybe count how much male/female is 1 and 0 and use that instead of normal distribution.