Justin Lai/ justinlai@ucsb.edu/9848664

Professor Xifeng

CS 165A Artificial Intelligence

Machine Problem 1 Report

Architecture

For my machine problem one, I decided to use the write my code using Python. In my program, I wrote two functions, the first one called training, and another called testing. This training function takes in the first argument that is passed in through the command line arguments, which is the training.txt. It then reads in each line from the training file, splits each line of the training file by commas to make a list, then stores every list into a two-dimensional list called trainingData. From this, I then created a dictionary to hold the counts of every athlete who passed and who failed. Next, I created a two-dimensional array in which I stored the values of every athlete by category and then derived percentile data from the values of each category by using Numpy's percentile function, storing the value of the 25th, 50th, 75th, and 100th percentile. After that, I created a three-dimensional list, the first dimension splitting the training data between whether the athlete had a high grade or not, the second dimension splitting the training data between the 11 other data points of the athlete, and the third dimension split the 11 data points into the four percentiles that I derived earlier, except for gender, in which the counts were stored by male or female. Next, I would then divide all of the counts stored in the three-dimensional list by the counts of high grades and low grades. Like the training function of

my program, the testing function also reads in all of the lines of code, splits the lines by commas to make lists, then stores it into a two-dimensional list called testingResults. From there, I created two variables for high or low grades which start out at 0.5 to represent the equal probability of high or low grade in the training data, then multiplied the two variables separately by the conditional probabilities. I then compared these two variables at the end, whichever probability was highest would be stored in an array and printed out.

Preprocessing

As mentioned above, in order to store an instance of an athlete, I first used a two-dimensional list called trainingData to store all of the values of an athlete, which I got from reading in training.txt and splitting on the commas. I then used this list to populate the three-dimensional array with the counts of each athlete that matched a certain category, based on whether that athlete received a high grade or not as the first dimension, then which of the four percentiles the athlete's values for each category fell within, with the exception of gender, where it was only male and female. This gives us the counts of each category, split into a high grade or not, and the percentiles.

Model Building

In order to build the model, I created a three-dimensional array that stored the counts of every category, split into high grade or low grade, the different categories of the statistics of every athlete, and by the percentile the value of the category falls into. From this, we can then divide each count by the probability of high grade or low grade, depending on which side of the three-dimensional array the count falls into. This will give us the conditional probabilities of every percentile in each category, conditioned on both high grade and low grade, which is what we need in order to do a Naïve Bayes classification.

Results

The result of running this program with the testing data and comparing my prediction of a high or low grade using my Naïve Bayes classification with the true value for the grade of each athlete gives us a prediction accuracy of 77.45% on the private testing set, and a score of 78.07% on the public testing set. Also, from my local machine, I received a running time of 0.249 seconds for the training function, and 0.0527 for the testing function, while on Gradescope, my program received a total running time of 0.5 seconds.

Challenges

One challenge that I faced was trying to code my program in Python. Originally, I wanted to use C++ as I am most familiar with it but I encountered some issues when trying to represent the counts of every athlete in the three-dimensional array as well as just calculating the percentiles. To solve this problem, I opted to code in Python instead, a language that I have never formally learned and have only used sparingly when doing LeetCode. This made the assignment much easier when building the data structures to represent the data and was a very good learning experience for me.

Weaknesses

One weakness with my code is just the general sloppiness of my code, like variable names, the overabundance of if and elif statements, and probably for loops. As this was my first time coding a project in Python, I was constantly lost when writing my code and I feel that my code is very inefficient. I am hoping that as I code more in Python, I will be able to write more legible and efficient code.