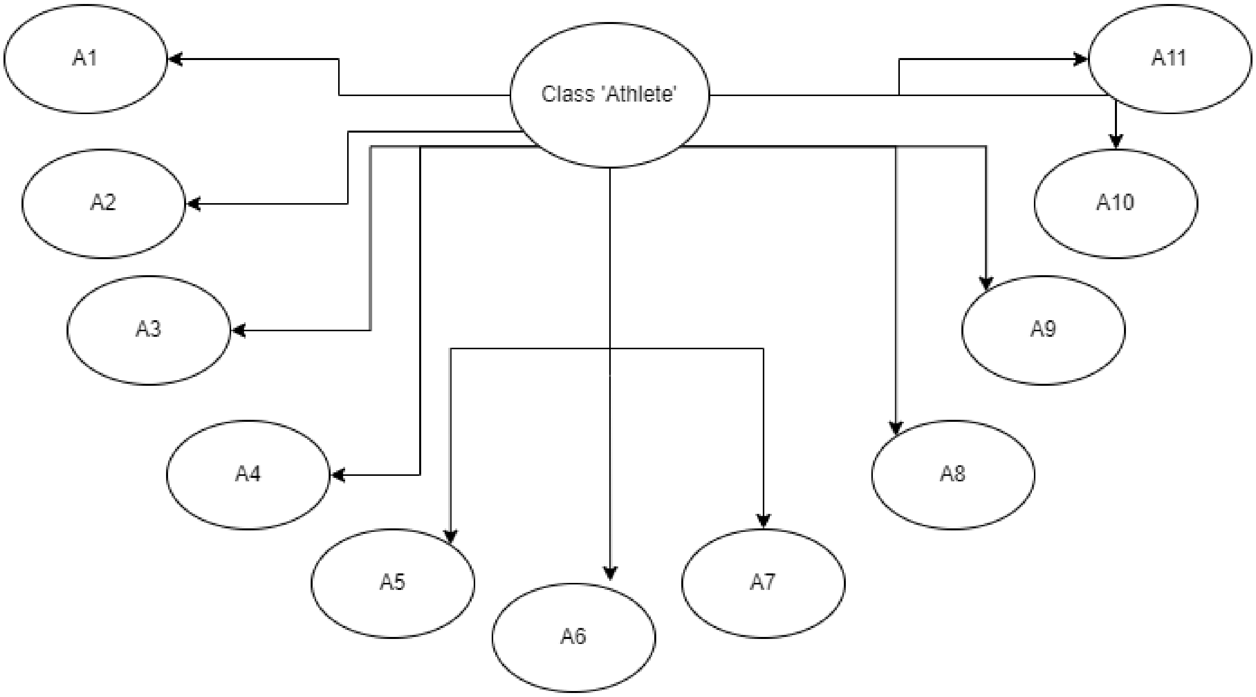
Report for Machine-problem 1

Name: Umang Garg, Perm# 6787683

Architecture

The implemented architecture follows the lines of the Naive Bayes classifier model where there are 2 classes ‘Good Athlete’ and ‘Not so good athlete’. Each classification is built upon 10 different athlete measurement which serves as classification criteria for unseen samples (test and validation dataset). The corresponding structure is given below.



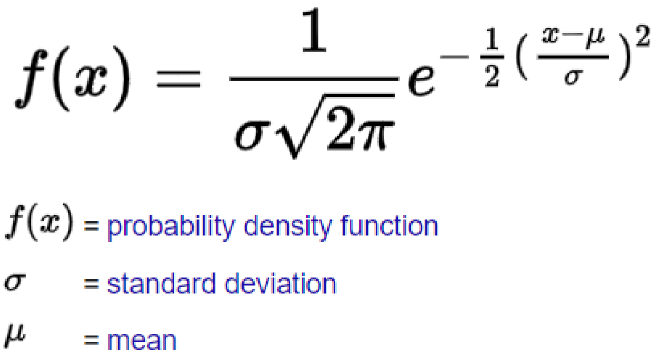
Preprocessing

For different attributes, we observe different types of data like characters, integer or float types representations. In order to uniformly apply the Naive Bayesian classification, the designed model converts each character representation into an alternate numerically mapped representation. For example, attribute 2 which is gender: the model maps “M” to 0, and “F” to 1. Further, the code filters different training samples based on their class labels to learn specific conditional probabilities for the specific class.

Model Building:

The model is trained first by separating different classes on the basis of provided training class labels. After separation, each class sample is assumed to be taken from a gaussian distribution, and hence each attribute’s mean and variance are calculated for the given class. Assuming each individual attribute is contributing independently, this forms a justified method for statistical analysis. Hence, we get the weights for any input vector by comparing it with its deviation from the attribute's mean, which in turn determines the correlation between the input and the class's attributes.

The gaussian formula used is given below:



$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

$f(x)$ = probability density function

σ = standard deviation

μ = mean

Results

The results from on training dataset:
Training time: 1.2157344818115234 seconds

Testing on Training dataset:
Accuracy: 76.71185661764706 %
Running time: 3.6253464221954346 seconds

Testing on test dataset:
Accuracy: 77.82197911486823 %
Running time: 1.0382602214813232 seconds

Challenges

The different challenges were:

1. Realizing different attributes have different data formats. It was solved by mapping different character attributes to a numerical value used for classification.
2. Realizing that in order to build the Bayes model, we need conditional probabilities. In order to get these probabilities from the provided training data, a statistical quantifying mechanism was needed. It was solved by referring to the online articles and understanding that in case the training samples set was sampled from a uniform distribution, the characteristics of the entire distribution

could be accurately represented by just 2 statistical measures, i.e. mean and variance of the corresponding attribute for a particular class. Giving it a shot, a statistical measure was extracted from the list of values for each attribute for corresponding to each class, and it was assumed that the normal distribution controls the correlation mapping to a particular class for a certain attribute value. Implementation of this yielded very good results and hence this was kept as the standard model technique.

3. Machine precision issues: As the naive Bayes classifier involves multiplying many small values to yield the final correlation probability for a particular class, a log scale was introduced to solve the extremely small probabilistic multiplication issue. The comparison of final probabilities was done based on these computed log values and still yield the correct comparative analysis results.

Weaknesses

The model possesses many weaknesses still and the potential solutions are provided herewith:

1. The model also uses Gaussian distribution for highly discrete and succinct attributes like gender whose only values could be ‘0’ and ‘1’. Corresponding to this attribute, a particular correlation probability can easily be calculated corresponding to each class type. This is sure to make the inference more sound and should increase the accuracy of model.
2. Each attribute is given equal weightage in the model. In real life, many attributes hold higher correlation as compared to others and hence this should be accounted for in the model where the probability for each attribute should be multiplied by a weight for final consideration of belonging to a class. These weights can be equal for all classes corresponding to an attribute.