**Question 5.2: Implementation of Gaussian Naïve Bayes and Logistic Regression**

1. **Pseudocode**

Load X(Samples), y(class) from the dataset

For fraction in fractions (0.01, 0.02, 0.05, 0.1, 0.625, 1)

Divide data into 3 folds making one of the folds test and other 2 folds training set

We will end up with 3 different combinations of train, test data

In each such combination, randomly pick a fraction of train data

Repeat the above step 5 times so that we will have 5 sets of data for each combination

Train the model for each of the set and learn the parameters

Predict the class for the test data from the learnt parameters and calculate the accuracy

Calculate the mean of accuracies which will be the accuracy for the fraction

Plot accuracy vs the fraction size curve

**Gaussian Naïve Bayes**

Function **Train**(X\_train, y\_train):

P(y=1) = number of positive samples / total number of samples

Find Meani,1 for training data of class y=1

Find Meani,0 for training data of class y=0

Find Variancei,1 for training data of class y=1

Find Variancei,0 for training data of class y=0

Learnt\_parameters = P(y=1), Meani,1, Meani,0, Variancei,1, Variancei,0

Return Learnt\_parameters

Function **Predict**(X\_test, y\_test, Learnt\_parameters):

P(X/Y=1) = (1/sqrt(2\*pi\* Variancei,1)) exp(-((X\_test - Meani,1)^2)/(2\* Variancei,1))

H\_pos = P(Y=1) \* ∏I P(Xi/Y=1)

P(X/Y=0) = (1/sqrt(2\*pi\* Variancei,0)) exp(-((X\_test - Meani,0)^2)/(2\* Variancei,0))

H\_zero = P(Y=0) \* ∏I P(Xi/Y=0)

If H\_pos > H-zero then

y\_pred = 1.0

else

y\_pred = 0.0

Accuracy = (# y\_pred == y\_test) / (# y\_pred)

Return Accuracy

**Logistic Regression**

Function **train**(X\_train, y\_train, learning\_rate)

W0=0, W=(0, 0, 0, 0)

For 500 iterations

Z = w0 + ∑iwiX\_traini

P(Y=1/X\_train,w) = exp(z)/(1+exp(z)) = 1/(1+exp(-z)) = sigmoid(-z) = a

w\_0 = w\_0 + learning\_rate \* (∑j (y\_trainj – P(Y=1/X\_trainj,w)))

w = w + learning\_rate \* (∑j Xj(y\_trainj – P(Y=1/X\_trainj,w)))

return w, w\_0

Function **predict**(X\_test, y\_test, w, w\_0)

Z = w0 + ∑iwiX\_testi

P(Y=1/X\_test,w) = exp(z)/(1+exp(z)) = 1/(1+exp(-z)) = sigmoid(-z) = a

If P(Y=1/X\_test,w) > 0.5 then

y\_pred = 1.0

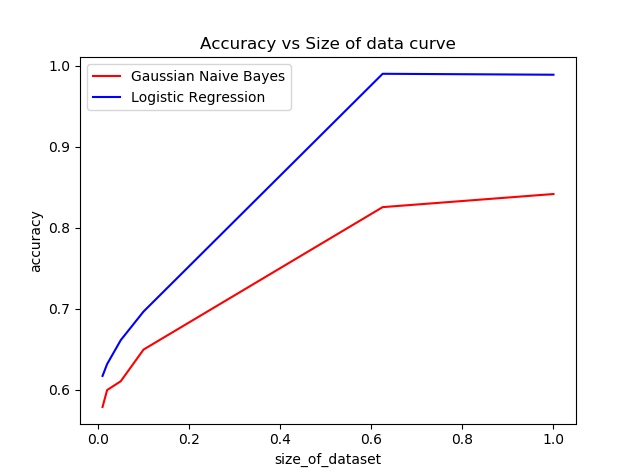
else

y\_pred = 0.0

Accuracy = (# y\_pred == y\_test) / (# y\_pred)

Return Accuracy

1. **Learning curve**



1. **Generate examples and compare their mean and variance with the training data**

**Fold 1**

Accuracy = 0.849015317287

Mean of training set: [-1.84753379 -0.89508644 2.03368838 -1.26910129]

Mean of generated set: [-1.76017452 -0.99775458 2.19185291 -1.19954756]

**Percentage of Mean Deviated**: [ 4.72842612 10.28991948 7.21601956 5.48054968]

Variance of training set: [ 3.52720065 29.00592331 26.85858549 4.40942191]

Variance of generated set: [ 3.92013342 28.76514662 26.13434515 4.76906046]

**Percentage of Variance Deviated**: [ 10.02345405 0.83009489 2.6964947 7.54107763]

**Fold 2**

Accuracy = 0.875273522976

Mean of training set: [-1.80954275 -0.92629223 2.0706694 -1.2504281 ]

Mean of generated set: [-1.811013 -0.9795895 2.1839032 -1.20256079]

**Percentage of Mean Deviated**: [ 0.08118422 5.44077653 5.18492754 3.82807361]

Variance of training set: [ 3.54048344 28.86707855 27.99184211 4.24441466]

Variance of generated set: [ 3.61466085 30.28615443 27.9742008 4.2960772 ]

**Percentage of Variance Deviated**: [ 2.05212644 4.68555982 0.06302304 1.20255159]

**Fold 3**

Accuracy = 0.80306345733

Mean of training set: [-1.94775075 -1.16378884 2.34564159 -1.21900678]

Mean of generated set: [-1.99693031 -0.95619555 2.63533007 -1.11657451]

**Percentage of Mean Deviated**: [ 2.46275768 17.83771089 10.99249318 8.40292865]

Variance of training set: [ 3.52197547 29.57897479 28.0784832 4.18090146]

Variance of generated set: [ 3.44216542 26.90525506 29.04871296 4.52075785]

**Percentage of Variance Deviated**: [ 2.26605923 9.0392576 3.34000947 7.5176862 ]