Q5] Pseudo Code –

```
Initialize all variables
Load the Data in variable "fullData"
Loop for each training Data Fraction in range [0.01, 0.02, 0.05, 0.1, 0.625, 1]{
   Loop each training data fraction 5 times for average accuracy{
      split "fullData" by the splitRatio - get "trainData"
      Here I assumed the "fullData" as the "testData"   (We can use the remaining fraction as testing data too)
      ####################################################
      #######       GAUSSIAN NB     ###################
      ####################################################
      seperate "trainData" by classs label into -> negative class="separated[0]" and positive class="separated[1]"
      calcualte P(Y=1) & P(Y=0)

      calculate mean at Y=1 and Y=0 for each attribute
         mean_i_k = sum(x_i)/N   -> for all X attributes=i and all classes=k
      calculate variance at Y=1 and Y=0 for each attribute
         mean_i_k = sum[(x_i-mean_i)^2]/N   -> for all X attributes=i and all classes=k

      #Testing
      For each row in testData:
         calculate P(X_i/Y=0) and P(X_i/Y=1) for all X attributes=i
            P(X_i/Y=k) = exp([(x_i-mean_i_k)^2/(-2*variance_i_k)])/sqrt(2*pi*variance)
         We calculate P(X1,X2,X3,X4/Y=0) and P(X1,X2,X3,X4/Y=1) by assuming conditional independence between all attributes X1,X2,X3,X4
            P(X1,X2,X3,X4/Y) = P(X1/Y)*P(X2/Y)*P(X3/Y)*P(X4/Y)
         We calculate P(Y=0/X1,X2,X3,X4) and P(Y=1/X1,X2,X3,X4) by using the Bayes Rule
            P(Y/X1,X2,X3,X4) = P(X1,X2,X3,X4/Y) / [P(X1,X2,X3,X4/Y=0)*P(Y=0)+P(X1,X2,X3,X4/Y=1)*P(Y=1)]
         if P(Y=0/X1,X2,X3,X4) > P(Y=1/X1,X2,X3,X4) -> then Prdicted class id Y=0 for this sample row
         else   ->  Y=1 for this sample row
         if (Predicted class == Actual class)   -> then CorrectPrediction_GNB incremented by 1

      accuracy_GNB = CorrectPrediction_GNB * 100 / Total # rows in testData
          avg_Accuracy_GNB will have summation of each accuracy_GNB calculated in each loop

      ##############################################################
      ##########       LOGISTIC REGRESSION     ##################
      ##############################################################

      Initialize k+1 Weights ( where k-> # of attributes)
      Set learning_Rate = 0.9
      Loop for 300 iterations or till new_Weight & old_weight differ in value
```

```
        {
          Calculate P(Y/X) for each row in trainData:
            sum = w0 + summation_for_every_attribute_i(wi*xi)   -> (where X1,x2,x3,x4 are attributes in a
sample row)
            P(Y/X) = sigmoid_Fuction(sum)   -> for this row

          ### Update w0
          errorDiff  = summation_for_each_row_in_trainData[Actual_Output - P(Y/X)]
          new_Weight = old_weight + learning_Rate * errorDiff

          ### Update w1,w2,....
          Loop for each weight w_i{
            errorDiff  = summation_for_each_row_in_trainData[(Actual_Output - P(Y/X_i)) * X_i]
            new_Weight = old_weight + learning_Rate * errorDiff
          }
        }

        # Testing
        Calculate P(Y/X) for each row in testData:
          sum = w0 + summation_for_every_attribute_i(wi*xi)   -> (where X1,x2,x3,x4 are attributes in a
sample row)
          P(Y/X) = sigmoid_Fuction(sum)   -> for this row
          If P(Y=0/X) > P(Y=1/X)   -> Actual_Output = 1
          else  ->   Actual_Output = 0

          if (Predicted class == Actual class)   ->   CorrectPrediction_LR++

        accuracy_LR = + [CorrectPrediction_LR * 100 / Total # rows in testData]
        avg_Accuracy_LR will have summation of each accuracy_LR calculated in each loop

    }
    Getting average accuracy from the 5 iterations for each training set:
        FINAL_Accuracy_GNB list will hold each (avg_Accuracy_GNB/5)
            FINAL_Accuracy_LR list will hold each (avg_Accuracy_LR/5)
}
Plot FINAL_Accuracy_GNB vs TestDataSize for GNB output
Plot FINAL_Accuracy_LR vs TestDataSize for LR output
```