```
clear all, close all, clc;

% opts = detectImportOptions('biopsy_data_missing_values.csv',
 'NumHeaderLines', 1);
% preview('biopsy_data_missing_values.csv', opts)

A = readtable('biopsy_data_missing_values.csv', 'NumHeaderLines', 1);
```

Part A - Data Formatting and Cleaning

```
var2_mt = find(strcmp(A.Var2, ''));
for i = 1:length(var2_mt)
    ii = var2_mt(i);
    A.Var2{ii} = 'Irregular';
end

var1_nan = find(~isfinite(A.Var1));
for i = var1_nan
    A.Var1(i) = i;
end

disp(A(:,1:2))
```

Part B - Naive Bayes

```
new_data = table();
new_data.Var1 = [1; 2; 3; 4; 5];
new_data.Var2 =
 {'Irregular'; 'Irregular'; 'Circle'; 'Circle'; 'Triangle'};
new_data.Var3 = {'Large'; 'Small'; 'Large'; 'Large'; 'Large'};
new_data.Var4 = {'Convex'; 'Flat'; 'Concave'; 'Convex'; 'Concave'};
new_data.Var5 = {'Rough'; 'Rough'; 'Smooth'; 'Smooth'; 'Smooth'};
new_data.Var6 = {'Neutral'; 'Red'; 'Neutral'; 'Dark'; 'Neutral'};

mal_inds = find(strcmp(A.Var7, 'Malignant'));
ben_inds = find(strcmp(A.Var7, 'Benign'));

NewBiopProbData = struct();
for biop = 1:height(new_data)

    sample = new_data(biop, :);

    BiopProbs = struct();
    temp_mal_probs = [];
    temp_ben_probs = [];

    for var_num = 2:length(sample.Properties.VariableNames)

        var = string(sample.Properties.VariableNames(var_num));
        biop_var_val = string(table2array(sample(:,var_num)));
        mal_cond_inds = find(strcmp(A.Var7, 'Malignant') & strcmp(A.
(var), biop_var_val));
        p_var_giv_mal = length(mal_cond_inds)/length(mal_inds);
        temp_mal_probs = [temp_mal_probs, p_var_giv_mal];
        BiopProbs.(strcat('P_of_', biop_var_val, '_given_Mal')) =
 p_var_giv_mal;

        ben_cond_inds = find(strcmp(A.Var7, 'Benign') & strcmp(A.
(var), biop_var_val));
        p_var_giv_ben = length(ben_cond_inds)/length(ben_inds);
        temp_ben_probs = [temp_ben_probs, p_var_giv_ben];
        BiopProbs.(strcat('P_of_', biop_var_val, '_given_Ben')) =
 p_var_giv_ben;

    end
    p_mal = length(mal_inds)/height(A);
    temp_mal_probs = [temp_mal_probs, p_mal];

    p_ben = length(ben_inds)/height(A);
    temp_ben_probs = [temp_ben_probs, p_ben];

    BiopProbs.('P_big_pos_Mal') = prod(temp_mal_probs);
    BiopProbs.('P_big_neg_Ben') = prod(temp_ben_probs);
    BiopProbs.('log_P_big_pos_Mal') = log(prod(temp_mal_probs));
    BiopProbs.('log_P_big_neg_Ben') = log(prod(temp_ben_probs));

    if BiopProbs.('P_big_pos_Mal') > BiopProbs.('P_big_neg_Ben')
        BiopProbs.('Predicted_Class') = {'Malignant'};
    elseif BiopProbs.('P_big_pos_Mal') < BiopProbs.('P_big_neg_Ben')
        BiopProbs.('Predicted_Class') = {'Benign'};
    else
        BiopProbs.('Predicted_Class') = {'Inconclusive'};
    end

    NewBiopProbData.(['biop' num2str(biop)]) = BiopProbs;
    clear BiopProbs temp_mal_probs temp_ben_probs;

end

fields = fieldnames(NewBiopProbData);
for biop_num = 1:length(fields)
    label = fields(biop_num);
    biop_prob_set = NewBiopProbData.(['biop' num2str(biop_num)]);

    disp(label)
    disp(biop_prob_set)
end
```

```
    Var1        Var2
    ____    _____________

      1     {'Circle'   }
      2     {'Circle'   }
      3     {'Circle'   }
```

```
  4     {'Irregular'}
  5     {'Circle'   }
  6     {'Circle'   }
  7     {'Circle'   }
  8     {'Irregular'}
  9     {'Triangle' }
 10     {'Circle'   }
 11     {'Irregular'}
 12     {'Irregular'}

{'biop1'}

P_of_Irregular_given_Mal: 0.1667
P_of_Irregular_given_Ben: 0.5000
    P_of_Large_given_Mal: 0.8333
    P_of_Large_given_Ben: 0.8333
   P_of_Convex_given_Mal: 0.1667
   P_of_Convex_given_Ben: 0.3333
    P_of_Rough_given_Mal: 0.5000
    P_of_Rough_given_Ben: 0.1667
  P_of_Neutral_given_Mal: 0.1667
  P_of_Neutral_given_Ben: 0.3333
           P_big_pos_Mal: 9.6451e-04
           P_big_neg_Ben: 0.0039
       log_P_big_pos_Mal: -6.9439
       log_P_big_neg_Ben: -5.5576
         Predicted_Class: {'Benign'}

{'biop2'}

P_of_Irregular_given_Mal: 0.1667
P_of_Irregular_given_Ben: 0.5000
    P_of_Small_given_Mal: 0.1667
    P_of_Small_given_Ben: 0.1667
     P_of_Flat_given_Mal: 0.3333
     P_of_Flat_given_Ben: 0.1667
    P_of_Rough_given_Mal: 0.5000
    P_of_Rough_given_Ben: 0.1667
      P_of_Red_given_Mal: 0.1667
      P_of_Red_given_Ben: 0.3333
           P_big_pos_Mal: 3.8580e-04
           P_big_neg_Ben: 3.8580e-04
       log_P_big_pos_Mal: -7.8602
       log_P_big_neg_Ben: -7.8602
         Predicted_Class: {'Inconclusive'}

{'biop3'}

 P_of_Circle_given_Mal: 0.8333
 P_of_Circle_given_Ben: 0.3333
  P_of_Large_given_Mal: 0.8333
  P_of_Large_given_Ben: 0.8333
P_of_Concave_given_Mal: 0.5000
P_of_Concave_given_Ben: 0.5000
```

```
 P_of_Smooth_given_Mal: 0.5000
 P_of_Smooth_given_Ben: 0.8333
P_of_Neutral_given_Mal: 0.1667
P_of_Neutral_given_Ben: 0.3333
         P_big_pos_Mal: 0.0145
         P_big_neg_Ben: 0.0193
     log_P_big_pos_Mal: -4.2358
     log_P_big_neg_Ben: -3.9482
       Predicted_Class: {'Benign'}

{'biop4'}

P_of_Circle_given_Mal: 0.8333
P_of_Circle_given_Ben: 0.3333
 P_of_Large_given_Mal: 0.8333
 P_of_Large_given_Ben: 0.8333
P_of_Convex_given_Mal: 0.1667
P_of_Convex_given_Ben: 0.3333
P_of_Smooth_given_Mal: 0.5000
P_of_Smooth_given_Ben: 0.8333
  P_of_Dark_given_Mal: 0.6667
  P_of_Dark_given_Ben: 0.3333
        P_big_pos_Mal: 0.0193
        P_big_neg_Ben: 0.0129
    log_P_big_pos_Mal: -3.9482
    log_P_big_neg_Ben: -4.3536
      Predicted_Class: {'Malignant'}

{'biop5'}

P_of_Triangle_given_Mal: 0
P_of_Triangle_given_Ben: 0.1667
   P_of_Large_given_Mal: 0.8333
   P_of_Large_given_Ben: 0.8333
 P_of_Concave_given_Mal: 0.5000
 P_of_Concave_given_Ben: 0.5000
  P_of_Smooth_given_Mal: 0.5000
  P_of_Smooth_given_Ben: 0.8333
 P_of_Neutral_given_Mal: 0.1667
 P_of_Neutral_given_Ben: 0.3333
          P_big_pos_Mal: 0
          P_big_neg_Ben: 0.0096
      log_P_big_pos_Mal: -Inf
      log_P_big_neg_Ben: -4.6413
        Predicted_Class: {'Benign'}
```

Published with MATLAB® R2021a