Program to implement Naive Bayes classifier.

Usage: [accuracy, misclass] = naive_bayes; The function gives the accuracy in percentage and the number of times each class is misclassified.

```
function [accuracy, misclass] = naive_bayes
clear; % Clear all.
clc; % Clear command window.

% Read data.
data = load('train.data');
label = load('train.label');
% [topic, topic_id] = textread('train.map', '%s %d');
delta = 0.1;
[m,~] = size(data);
% Count number of word_types, words, classes, and documents.
a = max(data);
% docs = a(1); % number of docs
word_types = a(2); % number of word types
b = word_types;
% words = sum(data(:,3)); % number of words
classes = max(label); % number of classes

% Count number of docs in each class.
h = zeros(classes,1);
for i = 1:classes
    h(i) = sum(label == i);
end

% Cumulative number of docs in 20 classes.
g = zeros(classes,1);
g(1) = h(1);
for p = 2:classes
    g(p) = h(p) + g(p-1);
end

% Assign class to each doc in data matrix.
s = 1;
for q = 1:classes
    [pos,~] = find(data(:,1)==g(q));
    last_pos = max(pos);
    for r = s:last_pos
        data(r,4) = q;
        s = s + 1;
    end
end

% Word count matrix.
final_mat = zeros(classes,word_types);
for v = 1:m
    final_mat(data(v,4),data(v,2)) = final_mat(data(v,4),data(v,2)) +
 data(v,3);
```

```
end

% Probability of word counts in word count matrix.
prob_matrix = zeros(classes,word_types);
for w = 1:classes
    prob_matrix(w,:) = (1 - delta)*((final_mat(w,:))/
sum(final_mat(w,:))) + (delta/word_types);
end

% Test data
data = load('test.data'); % Read data
label = load('test.label');
[topic, topic_id] = textread('test.map', '%s %d');
[c,~] = size(data);

% Count the number of documents and word types.
a = max(data);
docs_test = a(1); % number of docs
word_types_test = a(2); % number of word types

% Word count in docs matrix.
mat_test = zeros(docs_test,word_types_test);
for v = 1:c
    mat_test(data(v,1),data(v,2)) = mat_test(data(v,1),data(v,2)) +
 data(v,3);
end

% BAYES

Ck = histcounts(label);
Prob_Ck = Ck ./ (sum(Ck,2));
log_Prob_Ck = log(Prob_Ck);


% prob_matrix = [prob_matrix zeros(classes,word_types - 53975)];
log_prob_matrix = log(prob_matrix);
Pred = zeros(7505,1);
for x = 1:docs_test
    [~,Pred(x,1)] = max(log_Prob_Ck + (mat_test(x,1:b) *
 log_prob_matrix'));
end

% Accuracy in percentage
accuracy = ((sum((label-Pred) == 0))/docs_test)*100;

% Error count for each class.
e = zeros(1,7505);
for i = 1:docs_test
    if (label(i)~=Pred(i))
        e(i) = label(i);
    end
end

% Number of times each class is misclassified.
```

```
hist_count = histcounts(e);
f = (hist_count(2:21))';
topic_id = num2cell(topic_id);
f = num2cell(f);
misclass = [topic topic_id f];
end
```

```
accuracy =

   80.5730
```

```
misclass =

    'alt.atheism'             [ 1]    [ 65]
    'comp.graphics'           [ 2]    [ 88]
    'comp.os.ms-window...'    [ 3]    [150]
    'comp.sys.ibm.pc.h...'    [ 4]    [ 97]
    'comp.sys.mac.hard...'    [ 5]    [ 81]
    'comp.windows.x'          [ 6]    [115]
    'misc.forsale'            [ 7]    [ 83]
    'rec.autos'               [ 8]    [ 39]
    'rec.motorcycles'         [ 9]    [ 26]
    'rec.sport.baseball'      [10]    [ 32]
    'rec.sport.hockey'        [11]    [ 21]
    'sci.crypt'               [12]    [ 42]
    'sci.electronics'         [13]    [117]
    'sci.med'                 [14]    [ 68]
    'sci.space'               [15]    [ 53]
    'soc.religion.chri...'    [16]    [ 33]
    'talk.politics.guns'      [17]    [ 44]
    'talk.politics.mid...'    [18]    [ 67]
    'talk.politics.misc'      [19]    [128]
    'talk.religion.misc'      [20]    [109]
```

Published with MATLAB® R2016a