```
import csv
import math
import matplotlib.pyplot as plt
import numpy as np

def read_csv_file(file_name):
    """
    Reads a csv file after second line and returns three list of lists based on the
second column
    """
    numbers = []
    with open(file_name, 'r') as csv_file:
        csv_reader = csv.reader(csv_file)
        next(csv_reader)
        for item in list(csv_reader):
            my_dict = { 'value': int(item[0]), 'class': int(item[1]) , 'prediction_z
ero_one' : -1, 'prediction_reject' : -1}
            numbers.append(my_dict)
    return numbers


def calculate_likelihood(number, mean, std_dev):
    """
    Calculates the likelihood
    """
    likelihood = 1.0 / (std_dev * (2.0 * math.pi) ** 0.5) * (math.exp(-((number - me
an) ** 2) / (2.0 * std_dev ** 2)))
    return likelihood

'''
Below values are gathered from the training data.
'''
prior_class_1 = 1/3
prior_class_2 = 1/3
prior_class_3 = 1/3

mean_class_1 = 24.48
mean_class_2 = 34.12
mean_class_3 = 49.44

std_class_1 = 1.992385504865963
std_class_2 = 4.1358916813669095
std_class_3 = 5.091797325110258

numbers = read_csv_file("training.csv")
numbers_one   = [x['value'] for x in numbers if x['class'] == 1]
numbers_two   = [x['value'] for x in numbers if x['class'] == 2]
numbers_three = [x['value'] for x in numbers if x['class'] == 3]
minimum = min([x['value'] for x in numbers])
maximum = max([x['value'] for x in numbers])

plot_nums =  np.linspace(minimum, maximum, (maximum-minimum)*3+1)

likelihoods_class_1 = [calculate_likelihood(plot_nums[i], mean_class_1, std_class_1)
 for i in range(len(plot_nums))]
likelihoods_class_2 = [calculate_likelihood(plot_nums[i], mean_class_2, std_class_2)
 for i in range(len(plot_nums))]
likelihoods_class_3 = [calculate_likelihood(plot_nums[i], mean_class_3, std_class_3)
 for i in range(len(plot_nums))]

posterior_ones      = [prior_class_1 * likelihoods_class_1[i] / (likelihoods_class_1
[i] * prior_class_1 + likelihoods_class_2[i] * prior_class_2 + likelihoods_class_3[i
] * prior_class_3) for i in range(len(plot_nums))]
posterior_twos      = [prior_class_2 * likelihoods_class_2[i] / (likelihoods_class_1
[i] * prior_class_1 + likelihoods_class_2[i] * prior_class_2 + likelihoods_class_3[i
] * prior_class_3) for i in range(len(plot_nums))]
posterior_threes    = [prior_class_3 * likelihoods_class_3[i] / (likelihoods_class_1
[i] * prior_class_1 + likelihoods_class_2[i] * prior_class_2 + likelihoods_class_3[i
] * prior_class_3) for i in range(len(plot_nums))]

plt.title("Likelihoods and Posteriors for Training Dataset")
plt.plot(numbers_one  , [-0.05 for i in numbers_one], 'rx')
plt.plot(numbers_two  , [-0.1 for i in numbers_two] , 'go')
```

```
plt.plot(numbers_three, [-0.15 for i in numbers_three], 'b+')

plt.plot(plot_nums, posterior_ones, 'r--', label='P(C=1|X)')
plt.plot(plot_nums, posterior_twos, 'g--', label='P(C=2|X)')
plt.plot(plot_nums, posterior_threes, 'b--', label='P(C=3|X)')

plt.plot(plot_nums, likelihoods_class_1, 'r-', label='P(X|C=1)')
plt.plot(plot_nums, likelihoods_class_2, 'g-', label='P(X|C=2)')
plt.plot(plot_nums, likelihoods_class_3, 'b-', label='P(X|C=3)')

plt.legend(loc='center right')

plt.axis([minimum-3, maximum+3, -0.3, 1.1])
plt.xlabel('Age')
plt.show()



test_list = read_csv_file("testing.csv")
test_list_one   = [x['value'] for x in test_list if x['class'] == 1]
test_list_two   = [x['value'] for x in test_list if x['class'] == 2]
test_list_three = [x['value'] for x in test_list if x['class'] == 3]

minimum = min([x['value'] for x in test_list])
maximum = max([x['value'] for x in test_list])

plot_nums_test =  np.linspace(minimum, maximum, (maximum-minimum)*3+1)

likelihoods_class_1 = [calculate_likelihood(plot_nums_test[i], mean_class_1, std_cla
ss_1) for i in range(len(plot_nums_test))]
likelihoods_class_2 = [calculate_likelihood(plot_nums_test[i], mean_class_2, std_cla
ss_2) for i in range(len(plot_nums_test))]
likelihoods_class_3 = [calculate_likelihood(plot_nums_test[i], mean_class_3, std_cla
ss_3) for i in range(len(plot_nums_test))]

posterior_ones      = [prior_class_1 * likelihoods_class_1[i] / (likelihoods_class_1
[i] * prior_class_1 + likelihoods_class_2[i] * prior_class_2 + likelihoods_class_3[i
] * prior_class_3) for i in range(len(plot_nums_test))]
posterior_twos      = [prior_class_2 * likelihoods_class_2[i] / (likelihoods_class_1
[i] * prior_class_1 + likelihoods_class_2[i] * prior_class_2 + likelihoods_class_3[i
] * prior_class_3) for i in range(len(plot_nums_test))]
posterior_threes    = [prior_class_3 * likelihoods_class_3[i] / (likelihoods_class_1
[i] * prior_class_1 + likelihoods_class_2[i] * prior_class_2 + likelihoods_class_3[i
] * prior_class_3) for i in range(len(plot_nums_test))]


plt.title("Likelihoods and Posteriors for Test Dataset")
plt.plot(test_list_one  , [-0.05 for i in test_list_one], 'rx')
plt.plot(test_list_two  , [-0.1 for i in test_list_two] , 'go')
plt.plot(test_list_three, [-0.15 for i in test_list_three], 'b+')

plt.plot(plot_nums_test,posterior_ones, 'r--', label='P(C=1|X)')
plt.plot(plot_nums_test,posterior_twos, 'g--', label='P(C=2|X)')
plt.plot(plot_nums_test,posterior_threes, 'b--', label='P(C=3|X)')

plt.plot(plot_nums_test,likelihoods_class_1, 'r-', label='P(X|C=1)')
plt.plot(plot_nums_test,likelihoods_class_2, 'g-', label='P(X|C=2)')
plt.plot(plot_nums_test,likelihoods_class_3, 'b-', label='P(X|C=3)')

plt.legend(loc='center right')

plt.axis([min([x['value'] for x in test_list])-3, max([x['value'] for x in test_list
])+3, -0.3, 1.1])
plt.xlabel('Age')
plt.show()
```