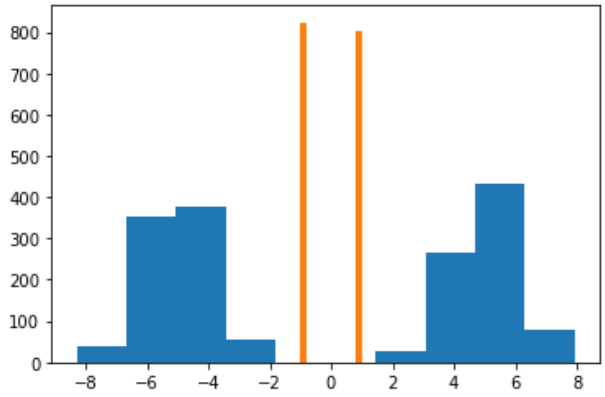
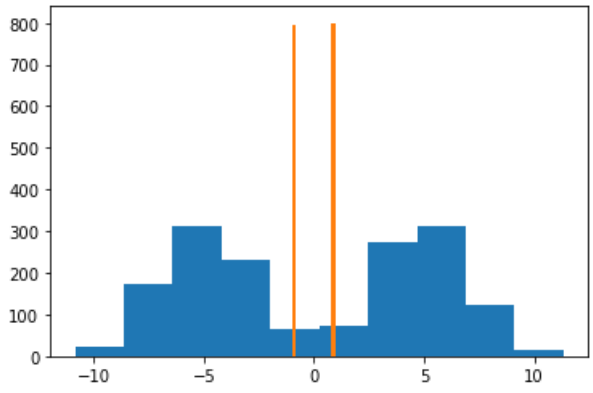
Andrew Nalundasan
OMSBA 5067, Seattle University
Lab 02
April 11, 2021

Step 6:
I observe that when we consider more standard deviations, the distribution gets wider and wider. From 4 standard deviations and below, one can clearly see that there are 2 distributions. With 1 standard deviation, we see that misclassification is at 0, which means that the classification rate is 100. As we continue to increase the number of standard deviations, we see the misclassification rate increase as well, the two distributions slowly overlapping each other. By the time we get to 5 standard deviations, we see so see a lot of overlap between the two distributions, where it can easily be mistaken as a single distribution.

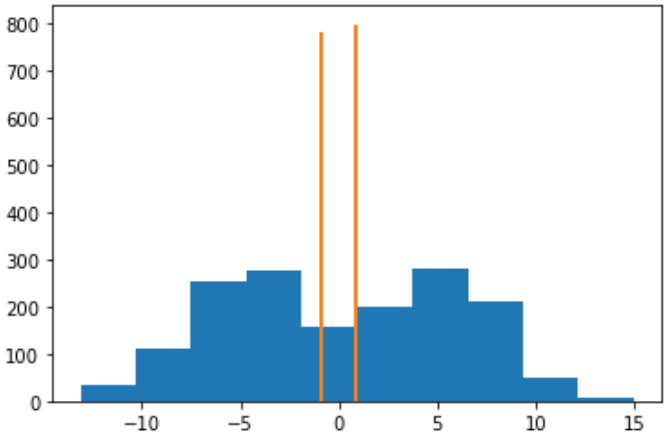
Standard Deviations: 1
Misclassification Rate: 0.0



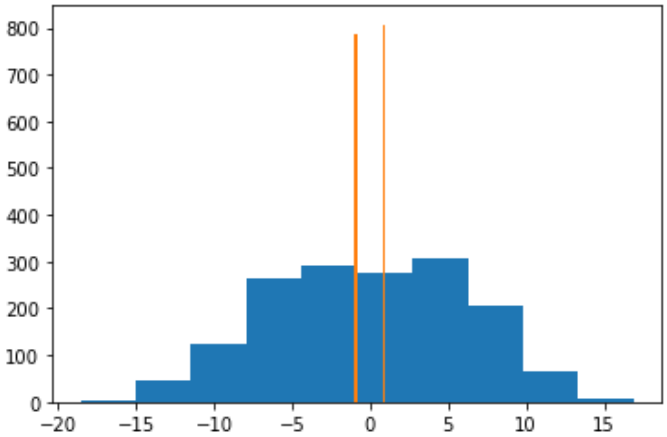
Standard Deviations: 2
Misclassification Rate: 0.00719



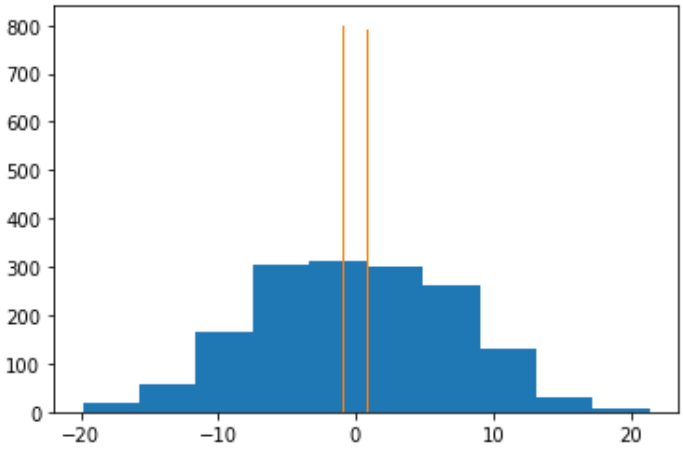
Standard Deviations: 3
Misclassification Rate: 0.0228



Standard Deviations: 4
Misclassification Rate: 0.089



Standard Deviations: 5
Misclassification Rate: 0.1698



Step 7:
I observe that the misclassification increases as train portion decreases. With less training data, the model has higher chances of making misclassifications. We want to minimize misclassifications so that our model performs more accurately.

train portion	standard deviations	misclassification rate
0.8	misrate_2	0.00775
0.2	misrate_2	0.00821
0.01	misrate_2	0.00856
0.005	misrate_2	0.07923

Step 9:
Code:

```
import numpy as np
from numpy import genfromtxt
from sklearn.naive_bayes import GaussianNB

data = genfromtxt('transfusion.csv', delimiter=',', skip_header=1)

## select data from transfusion file
X = data[:, [0, 1, 2, 3]]
Y = data[:, -1]
```

```
## train the data at 80% training, 20% testing
TrainPortion = 0.8
msk = np.random.rand(len(X)) < TrainPortion

# train data
trainX = X[msk]
trainY = Y[msk]

# test data
testX = X[~msk]
testY = Y[~msk]

# train the model
gnb = GaussianNB()
gnb.fit(trainX, trainY)

# solve for misclassification
estimatedY = gnb.predict(testX)
misrate = np.sum(np.abs(testY-estimatedY))/len(testY)
print(misrate)
```

```
In [256]: runcell(0, 'C:/Users/analundasan/OneDrive - ARCADIS/SU/OMSBA 5067/week-02/lab_02/lab_02_code.py')
0.27631578947368424
```

misrate = 0.276