Question 7: By: Roman Formicola

a)

In [155]:
```
from sklearn.model_selection import StratifiedKFold
from sklearn.preprocessing import OrdinalEncoder
from sklearn.preprocessing import MinMaxScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn import metrics
import numpy as np
```

In [146]:
```
def spec_score(y, y_pred):
    if(len(y) != len(y_pred)):
        print( "Error in spec_score")
        exit(-1)
    fp = 0.
    tn = 0.
    for i in range(len(y_pred)):
        if y_pred[i] == 0 and y[i] == 0:
            tn += 1
        elif y_pred[i] == 1 and y[i] == 0:
            fp += 1

    return tn / (fp + tn)
```

In [187]:
```
import csv

spam = []
with open('spam.csv') as csv_file:
    spam_file = csv.reader(csv_file, delimiter=',')
    linenum = 0;
    for row in spam_file:
        if(linenum != 0):
            spam.append(row)
        linenum += 1


#Transform Data to numeric type
enc = OrdinalEncoder()
enc.fit(spam)
t_spam = enc.transform(spam)

#Scale Data [0, 1]
scaler = MinMaxScaler()
scaler.fit(t_spam)
s_spam = scaler.transform(t_spam)

#Split data and target vector
y = np.array(s_spam.T[len(s_spam[0]) - 1]) #target vector
x = np.delete(s_spam, len(s_spam[0]) - 1, 1) #data


skf = StratifiedKFold(n_splits=10)
skf.get_n_splits(x, y)

train_data = []
test_data = []
for train_i, test_i in skf.split(x, y):
    x_train, x_test = x[train_i], x[test_i]
    y_train, y_test = y[train_i], y[test_i]
    train_data.append((x_train, y_train))
    test_data.append((x_test, y_test))
```

b)

In [188]:
```
model_array = [ KNeighborsClassifier(n_neighbors = 3),
                KNeighborsClassifier(n_neighbors = 7),
                KNeighborsClassifier(n_neighbors = 11),
                KNeighborsClassifier(n_neighbors = 15),
                DecisionTreeClassifier(max_depth = 5),
                DecisionTreeClassifier(criterion = "entropy", max_depth = 100, min_samples_split = 2
5),
                GaussianNB()]

for model in model_array:
    print(model)
    split_accuracy = []
    split_auc = []
    split_recall = []
    split_spec = []
    for i in range(len(train_data)):
        model.fit(train_data[i][0], train_data[i][1])
        y_pred = model.predict(test_data[i][0])

        split_accuracy.append(metrics.accuracy_score(test_data[i][1], y_pred))
        split_auc.append(metrics.roc_auc_score(test_data[i][1], y_pred))
        split_recall.append(metrics.recall_score(test_data[i][1], y_pred))
        split_spec.append(spec_score(test_data[i][1], y_pred))


    for i in range(10):
        print("Split: ", i, " Accuracy=", round(split_accuracy[i], 5),
              "AUC=", round(split_auc[i], 5), " Recall=", round(split_recall[i], 5), " Specificity="
, round(split_spec[i], 5))
    print("Average Accuracy over 10 splits:", np.mean(split_accuracy))
    print("Average AUC over 10 splits:", np.mean(split_auc))
    print("Average Recall over 10 splits:", np.mean(split_recall))
    print("Average Specificity over 10 splits:", np.mean(split_spec))
```

```
KNeighborsClassifier(n_neighbors=3)
Split:  0  Accuracy= 0.76606 AUC= 0.80833  Recall= 0.92958  Specificity= 0.68707
Split:  1  Accuracy= 0.97235 AUC= 0.97222  Recall= 0.97183  Specificity= 0.9726
Split:  2  Accuracy= 0.97696 AUC= 0.97926  Recall= 0.98592  Specificity= 0.9726
Split:  3  Accuracy= 0.96313 AUC= 0.95813  Recall= 0.94366  Specificity= 0.9726
Split:  4  Accuracy= 0.94009 AUC= 0.93377  Recall= 0.91549  Specificity= 0.95205
Split:  5  Accuracy= 0.92627 AUC= 0.91626  Recall= 0.88732  Specificity= 0.94521
Split:  6  Accuracy= 0.92627 AUC= 0.91626  Recall= 0.88732  Specificity= 0.94521
Split:  7  Accuracy= 0.95853 AUC= 0.95833  Recall= 0.95775  Specificity= 0.9589
Split:  8  Accuracy= 0.97235 AUC= 0.96498  Recall= 0.94366  Specificity= 0.9863
Split:  9  Accuracy= 0.85714 AUC= 0.79254  Recall= 0.60563  Specificity= 0.97945
Average Accuracy over 10 splits: 0.9259142603475248
Average AUC over 10 splits: 0.9200087675432898
Average Recall over 10 splits: 0.9028169014084508
Average Specificity over 10 splits: 0.9372006336781288
KNeighborsClassifier(n_neighbors=7)
Split:  0  Accuracy= 0.74771 AUC= 0.79472  Recall= 0.92958  Specificity= 0.65986
Split:  1  Accuracy= 0.97696 AUC= 0.98288  Recall= 1.0  Specificity= 0.96575
Split:  2  Accuracy= 0.97235 AUC= 0.97583  Recall= 0.98592  Specificity= 0.96575
Split:  3  Accuracy= 0.95853 AUC= 0.95471  Recall= 0.94366  Specificity= 0.96575
Split:  4  Accuracy= 0.96774 AUC= 0.96517  Recall= 0.95775  Specificity= 0.9726
Split:  5  Accuracy= 0.92166 AUC= 0.9056  Recall= 0.85915  Specificity= 0.95205
Split:  6  Accuracy= 0.92166 AUC= 0.91284  Recall= 0.88732  Specificity= 0.93836
Split:  7  Accuracy= 0.97235 AUC= 0.97222  Recall= 0.97183  Specificity= 0.9726
Split:  8  Accuracy= 0.96774 AUC= 0.95794  Recall= 0.92958  Specificity= 0.9863
Split:  9  Accuracy= 0.87097 AUC= 0.81729  Recall= 0.66197  Specificity= 0.9726
Average Accuracy over 10 splits: 0.927766033906904
Average AUC over 10 splits: 0.9239202665438162
Average Recall over 10 splits: 0.9126760563380282
Average Specificity over 10 splits: 0.9351644767496604
KNeighborsClassifier(n_neighbors=11)
Split:  0  Accuracy= 0.73394 AUC= 0.78452  Recall= 0.92958  Specificity= 0.63946
Split:  1  Accuracy= 0.97235 AUC= 0.97945  Recall= 1.0  Specificity= 0.9589
Split:  2  Accuracy= 0.97235 AUC= 0.97583  Recall= 0.98592  Specificity= 0.96575
Split:  3  Accuracy= 0.95853 AUC= 0.95471  Recall= 0.94366  Specificity= 0.96575
Split:  4  Accuracy= 0.97235 AUC= 0.97222  Recall= 0.97183  Specificity= 0.9726
Split:  5  Accuracy= 0.91705 AUC= 0.9058  Recall= 0.87324  Specificity= 0.93836
Split:  6  Accuracy= 0.91705 AUC= 0.90942  Recall= 0.88732  Specificity= 0.93151
Split:  7  Accuracy= 0.97235 AUC= 0.97222  Recall= 0.97183  Specificity= 0.9726
Split:  8  Accuracy= 0.95392 AUC= 0.94043  Recall= 0.90141  Specificity= 0.97945
Split:  9  Accuracy= 0.87097 AUC= 0.81729  Recall= 0.66197  Specificity= 0.9726
Average Accuracy over 10 splits: 0.9240857396524754
Average AUC over 10 splits: 0.9211875296134275
Average Recall over 10 splits: 0.9126760563380282
Average Specificity over 10 splits: 0.9296990028888269
KNeighborsClassifier(n_neighbors=15)
Split:  0  Accuracy= 0.7156 AUC= 0.77091  Recall= 0.92958  Specificity= 0.61224
Split:  1  Accuracy= 0.96313 AUC= 0.9726  Recall= 1.0  Specificity= 0.94521
Split:  2  Accuracy= 0.97235 AUC= 0.97583  Recall= 0.98592  Specificity= 0.96575
Split:  3  Accuracy= 0.96774 AUC= 0.96156  Recall= 0.94366  Specificity= 0.97945
Split:  4  Accuracy= 0.97235 AUC= 0.97583  Recall= 0.98592  Specificity= 0.96575
Split:  5  Accuracy= 0.91705 AUC= 0.90942  Recall= 0.88732  Specificity= 0.93151
Split:  6  Accuracy= 0.91244 AUC= 0.90237  Recall= 0.87324  Specificity= 0.93151
Split:  7  Accuracy= 0.96774 AUC= 0.96517  Recall= 0.95775  Specificity= 0.9726
Split:  8  Accuracy= 0.94931 AUC= 0.93339  Recall= 0.88732  Specificity= 0.97945
Split:  9  Accuracy= 0.86175 AUC= 0.8032  Recall= 0.6338  Specificity= 0.9726
Average Accuracy over 10 splits: 0.9199467298017165
Average AUC over 10 splits: 0.9170293778325531
Average Recall over 10 splits: 0.908450704225352
Average Specificity over 10 splits: 0.9256080514397540
DecisionTreeClassifier(max_depth=5)
Split:  0  Accuracy= 0.98624 AUC= 0.98251  Recall= 0.97183  Specificity= 0.9932
Split:  1  Accuracy= 0.94931 AUC= 0.92254  Recall= 0.84507  Specificity= 1.0
Split:  2  Accuracy= 0.94931 AUC= 0.93701  Recall= 0.90141  Specificity= 0.9726
Split:  3  Accuracy= 0.97235 AUC= 0.96136  Recall= 0.92958  Specificity= 0.99315
Split:  4  Accuracy= 0.91244 AUC= 0.87705  Recall= 0.77465  Specificity= 0.97945
Split:  5  Accuracy= 0.91705 AUC= 0.88047  Recall= 0.77465  Specificity= 0.9863
Split:  6  Accuracy= 0.97235 AUC= 0.97222  Recall= 0.97183  Specificity= 0.9726
Split:  7  Accuracy= 0.98157 AUC= 0.98268  Recall= 0.98592  Specificity= 0.97945
Split:  8  Accuracy= 0.98157 AUC= 0.98268  Recall= 0.98592  Specificity= 0.97945
Split:  9  Accuracy= 0.8894 AUC= 0.8346  Recall= 0.67606  Specificity= 0.99315
Average Accuracy over 10 splits: 0.9511584154229908
Average AUC over 10 splits: 0.9333131535461956
Average Recall over 10 splits: 0.8816901408450704
Average Specificity over 10 splits: 0.9849361662473208
DecisionTreeClassifier(criterion='entropy', max_depth=100, min_samples_split=25)
Split:  0  Accuracy= 0.92661 AUC= 0.92373  Recall= 0.91549  Specificity= 0.93197
Split:  1  Accuracy= 0.94009 AUC= 0.93016  Recall= 0.90141  Specificity= 0.9589
Split:  2  Accuracy= 0.94931 AUC= 0.93701  Recall= 0.90141  Specificity= 0.9726
Split:  3  Accuracy= 0.91705 AUC= 0.92027  Recall= 0.92958  Specificity= 0.91096
Split:  4  Accuracy= 0.92166 AUC= 0.89837  Recall= 0.83099  Specificity= 0.96575
Split:  5  Accuracy= 0.8894 AUC= 0.85993  Recall= 0.77465  Specificity= 0.94521
Split:  6  Accuracy= 0.88018 AUC= 0.87478  Recall= 0.85915  Specificity= 0.89041
Split:  7  Accuracy= 0.96313 AUC= 0.96175  Recall= 0.95775  Specificity= 0.96575
Split:  8  Accuracy= 0.97235 AUC= 0.97222  Recall= 0.97183  Specificity= 0.9726
Split:  9  Accuracy= 0.88018 AUC= 0.83137  Recall= 0.69014  Specificity= 0.9726
Average Accuracy over 10 splits: 0.9239969559886696
Average AUC over 10 splits: 0.9109580837930388
Average Recall over 10 splits: 0.8732394366197183
Average Specificity over 10 splits: 0.9486767309663593
GaussianNB()
Split:  0  Accuracy= 0.72477 AUC= 0.69397  Recall= 0.60563  Specificity= 0.78231
Split:  1  Accuracy= 0.91244 AUC= 0.87343  Recall= 0.76056  Specificity= 0.9863
Split:  2  Accuracy= 0.99078 AUC= 0.99315  Recall= 1.0  Specificity= 0.9863
Split:  3  Accuracy= 0.90323 AUC= 0.89552  Recall= 0.87324  Specificity= 0.91781
Split:  4  Accuracy= 0.95853 AUC= 0.96194  Recall= 0.97183  Specificity= 0.95205
Split:  5  Accuracy= 0.85714 AUC= 0.82148  Recall= 0.71831  Specificity= 0.92466
Split:  6  Accuracy= 0.88018 AUC= 0.84946  Recall= 0.76056  Specificity= 0.93836
Split:  7  Accuracy= 0.91705 AUC= 0.89495  Recall= 0.83099  Specificity= 0.9589
Split:  8  Accuracy= 0.89401 AUC= 0.86335  Recall= 0.77465  Specificity= 0.95205
Split:  9  Accuracy= 0.80645 AUC= 0.72231  Recall= 0.47887  Specificity= 0.96575
Average Accuracy over 10 splits: 0.8844586310404046
Average AUC over 10 splits: 0.8569576296657964
Average Recall over 10 splits: 0.7774647887323944
Average Specificity over 10 splits: 0.9364504705991985
```

In [ ]:
```
```