In [7]:
```
import csv
import time
import random
from math import *
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sklearn # make sure this is installed in your environment.
from sklearn import tree
from sklearn.datasets import *
from sklearn.linear_model import SGDClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import make_classification
from sklearn.naive_bayes import BernoulliNB
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.ensemble import GradientBoostingClassifier
```

Data Import/Cleaning Helper Function

In [8]:
```
#test if a string is number
def is_number(n):
    try:
        float(n)
    except ValueError:
        return False
    return True
```

In [9]:
```
def csv_to_dataframe(csv_file):
    bank_list0 = [] #read original csv in to list of list(with splited c
ells)
    bank_title = []
    bank_data = []
    with open(csv_file) as csvfile:
        readCSV = csv.reader(csvfile, delimiter=',')
        for i in readCSV:
            bank_list0.append(i[0].split(';'))
    bank_title = list(ele.strip('"') for ele in bank_list0[0]) #store al
l col names
    #read all data into list, each list inside bank_data is a row(recor
d) from raw data
    for it in bank_list0[1:]:
        temp_list = []
        for item in it:
            temp_key = item.strip('"') # Delete all extra ' " '
            if is_number(temp_key): # Convert all string of number into
 float
                temp_key = float(temp_key)
            temp_list.append(temp_key)
        bank_data.append(temp_list)
    df = pd.DataFrame (bank_data,columns=bank_title)
    return df
```

In [10]:
```
def one_hot_encode(dataframe):
    col_name = list(dataframe.columns)
    cat_columns = []
    record = dataframe[:1]
    for i in col_name:
        if record[i].dtypes=="object":
            cat_columns.append(i)
    df_onehot = pd.get_dummies(dataframe,columns = cat_columns)
    return df_onehot
```

encode_record_into_vector Function

In [11]:
```
#record: one row from bank_df
def encode_record_into_vector(record):
    col_name = list(record.columns)
    x = []
    for i in col_name:
        x.append(float(record[i]))
    end = time.time()
    return x
```

Matrix of Encoded Record

In [12]:
```
def parse_file_into_matrix(file_name):
    bank_df = csv_to_dataframe('bank-additional-full.csv')
    bank_df_y = csv_to_dataframe('bank-additional-full.csv')
    bank_df.drop('y', inplace=True, axis=1)
    clean_df = one_hot_encode(bank_df)
    X = []
    y = []
    for i in range(clean_df.shape[0]):
        X.append(encode_record_into_vector(clean_df[i:i+1]))
    y = LabelBinarizer().fit_transform(bank_df_y["y"]).tolist()
    return (X, y)
```

In [13]:
```
matrix = parse_file_into_matrix("bank-additional-full.csv")
X = matrix[0]
y = matrix[1]
```

Q1 [5 points]. Write some code to count the number of 1s and 0s in y. How many positive and negative instances each are in your dataset?

In [14]:
```
#function to count number of 1s and 0s in y
def count_ins_y(y):
    pos = 0
    neg = 0
    for i in y:
        if i==0 or i==[0]:
            neg+=1
        if i==1 or i == [1]:
            pos+=1
    print("Number of 1: ",pos)
    print("Number of 0: ",neg)
    return (pos,neg)
```

In [15]:
```
count_ins_y(y)
```

```
Number of 1:  4640
Number of 0:  36548
```

Out[15]:
```
(4640, 36548)
```

In [16]:
```
def training_testing_split(X, y, training_ratio=0.8):
    if type(X)==np.ndarray:
        X = X.tolist()
    if type(y)==np.ndarray:
        y = y.tolist()
    X_pos = []
    X_neg = []
    #split 1 and 0 instances
    for i in range(len(y)):
        if y[i] == 1 or y[i]==[1]:
            X_pos.append(X[i])
        elif y[i] == 0 or y[i]==[0]:
            X_neg.append(X[i])
    X_train_pos_sample_index = random.sample(range(len(X_pos)),ceil(len(
X_pos)*training_ratio))#index
    X_train_neg_sample_index = random.sample(range(len(X_neg)),ceil(len(
X_neg)*training_ratio))#index
    X_test_pos_sample_index=[]
    X_test_neg_sample_index=[]
    #fill all X_test indexes
    for j in range(len(X_pos)):
        if j not in X_train_pos_sample_index:
                X_test_pos_sample_index.append(j)
    for k in range(len(X_neg)):
         if k not in X_train_neg_sample_index:
                X_test_neg_sample_index.append(k)
    X_train=[]
    X_test=[]
    for m in X_train_pos_sample_index:
        X_train.append(X[m])
    for n in X_train_neg_sample_index:
        X_train.append(X[n])
    for o in X_test_pos_sample_index:
        X_test.append(X[o])
    for p in X_test_neg_sample_index:
        X_test.append(X[p])

    y_train = [0]*len(X_train_neg_sample_index)+[1]*len(X_train_pos_samp
le_index)
    y_test = [0]*len(X_test_neg_sample_index)+[1]*len(X_test_pos_sample_
index)

    return (X_train, y_train, X_test, y_test)
```

Q2 (20 points). Run the code above with the X and y that you got from parse_file_into_matrix, with training ratios of 0.8, 0.5, 0.3 and 0.1. For each of these four cases, what is the 'ratio' of positive instances in the training dataset to the total number of instances in the training dataset? Verify that this same ratio is achieved in the test dataset. Write additional code to run these verifications if necessary (5 points per case).

In [17]:
```
def instance_ratio(X,y,training_ratio):
    split_data = training_testing_split(X, y, training_ratio)
    y_train = split_data[1]
    y_test = split_data[3]
    train_pos_ratio = count_ins_y(y_train)[0]/len(y_train)
    test_pos_ratio = count_ins_y(y_test)[0]/len(y_test)

    print("Positive ratio of training dataset ="
          ,train_pos_ratio)
    print("Positive ratio of test dataset ="
          ,test_pos_ratio)
    return (train_pos_ratio,test_pos_ratio)
```

Case 1: Training ratio = 0.8

In [18]:
```
instance_ratio(X,y,0.8)
```

```
Number of 1:  3712
Number of 0:  29239
Number of 1:  928
Number of 0:  7309
Positive ratio of training dataset = 0.11265211981426967
Positive ratio of test dataset = 0.11266237707903363
```

Out[18]:
```
(0.11265211981426967, 0.11266237707903363)
```

Case 1: Training ratio = 0.5

In [19]:
```
instance_ratio(X,y,0.5)
```

```
Number of 1:  2320
Number of 0:  18274
Number of 1:  2320
Number of 0:  18274
Positive ratio of training dataset = 0.11265417111780131
Positive ratio of test dataset = 0.11265417111780131
```

Out[19]:
```
(0.11265417111780131, 0.11265417111780131)
```

Case 1: Training ratio = 0.3

In [20]:
```
instance_ratio(X,y,0.3)
```

```
Number of 1:  1392
Number of 0:  10965
Number of 1:  3248
Number of 0:  25583
Positive ratio of training dataset = 0.11264870114105366
Positive ratio of test dataset = 0.11265651555617218
```

Out[20]:
```
(0.11264870114105366, 0.11265651555617218)
```

Case 1: Training ratio = 0.1

In [21]:
```
instance_ratio(X,y,0.1)
```

```
Number of 1:  464
Number of 0:  3655
Number of 1:  4176
Number of 0:  32893
Positive ratio of training dataset = 0.11264870114105366
Positive ratio of test dataset = 0.11265477892578704
```

Out[21]:
```
(0.11264870114105366, 0.11265477892578704)
```

In [85]:
```
def train_models(X_train,y_train,model):
    if model == 'decision_tree':
        clf = tree.DecisionTreeClassifier()
        clf = clf.fit(X_train, y_train)

    if model == 'naive_bayes':
        clf = BernoulliNB(alpha=1,binarize=0,fit_prior=False, class_prio
r=None)
        clf.fit(X_train, y_train)

    if model == 'linear_SGD_classifier':
        clf = make_pipeline(StandardScaler(),SGDClassifier(loss='squared
_loss'))
        clf.fit(X_train, y_train)

    if model == 'gradient_tree_boosting':
        clf=GradientBoostingClassifier()
        clf.fit(X_train, y_train)

    return clf
```

[10 points] Expand/replace 'pass' above to return the other two models based on the value of the model parameter 'model' [10 points]

[5 points] Expand to return one other model that I have not taught in class.

In [23]:
```
def evaluate_model(X_test, y_test, model):
    y_predict = model.predict(X_test)
    return sklearn.metrics.f1_score(y_test, y_predict)
```

In [79]:
```
a = time.time()
M=training_testing_split(X, y, 0.5)
X_train = M[0]
y_train = M[1]
X_test  = M[2]
y_test  = M[3]
b = time.time()
print(b-a)
```

```
6.5735180377960205
```

In [84]:
```
clf = BernoulliNB(alpha=1,binarize=0,fit_prior=False, class_prior=None)
model=clf.fit(X_train, y_train)
y_predict = model.predict(X_test)
sklearn.metrics.f1_score(y_test, y_predict)
```

Out[84]:
```
0.29096671949286845
```

==========

Export 5-Table Report

In [86]:
```
def trials(train_percent,num_trails):
    #store the f-measures from all 10 trails
    decision_tree = []
    naive_bayes = []
    linear_SGD_classifier = []

    for i in range(num_trails):
        M=training_testing_split(X, y, train_percent)
        X_train = M[0]
        y_train = M[1]
        X_test  = M[2]
        y_test  = M[3]
        decision_tree.append(evaluate_model(X_test, y_test, train_models
(X_train,y_train,model='decision_tree')))
        naive_bayes.append(evaluate_model(X_test, y_test, train_models(X
_train,y_train,model='naive_bayes')))
        linear_SGD_classifier.append(evaluate_model(X_test, y_test, trai
n_models(X_train,y_train,model='linear_SGD_classifier')))
        print(i)
    df = pd.DataFrame()
    df['decision_tree']=decision_tree
    df['naive_bayes']=naive_bayes
    df['linear_SGD_classifier']=linear_SGD_classifier
    return df
```

In [87]:
```
df1 = trials(0.1,10)
print(1)
df3 = trials(0.3,10)
print(2)
df5 = trials(0.5,10)
print(3)
df7 = trials(0.7,10)
print(4)
df9 = trials(0.9,10)
```

```
0
1
2
3
4
5
6
7
8
9
1

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

0

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

1

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

2

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

3
4

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

5

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "
```

```
6

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

7
8

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

9
2

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

0

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

1

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

2

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

3

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

4
5
```

```
/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

6

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

7

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

8

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

9
3

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

0

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

1

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

2
```

```
/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

3

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

4

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

5

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

6

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

7

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

8

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

9
4
```

```
/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

0

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

1

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

2

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

3

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

4

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

5

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

6
```

```
/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

7
8
9

/Users/shaoqianchen/opt/anaconda3/lib/python3.8/site-packages/sklearn/l
inear_model/_stochastic_gradient.py:570: ConvergenceWarning: Maximum nu
mber of iteration reached before convergence. Consider increasing max_i
ter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "
```

In [88]:
```
df1.to_csv(r'10% Table.csv', index = True, header=True)
df3.to_csv(r'30% Table.csv', index = True, header=True)
df5.to_csv(r'50% Table.csv', index = True, header=True)
df7.to_csv(r'70% Table.csv', index = True, header=True)
df9.to_csv(r'90% Table.csv', index = True, header=True)
```

In [ ]:
```

```