Importation

We shall import seed, randrange, reader and sqrt

In [1]:

```
# Linear Regression With Stochastic Gradient Descent for Wine Quality
from random import seed
from random import randrange
from csv import reader
from math import sqrt
```

Load Csv File

We will use the csv module that is part of the standard library to load the file We will use the load_csv() function that will wrap this behaviour and return the dataset. Then we will take the loaded dataset as list of lists. We shall open the file loaded in read only mode. The reader() takes the file as an argument. The first list is the list of observation of rows and the second is a list of column valus for a given row.

In [2]:

```
# Load a CSV file
def load_csv(filename):
 dataset = list()
 with open(filename, 'r') as file:
  csv_reader = reader(file)
  for row in csv_reader:
   if not row:
    continue
   dataset.append(row)
 return dataset
```

Datatype Conversion

Through the str_column_to_float() We shall convert the given column values to floating point value. We then use strip() to remove any whitespace before the conversion.

In [3]:

```
# Convert string column to float
def str_column_to_float(dataset, column):
 for row in dataset:
  row[column] = float(row[column].strip())
```

Finding Minimum And Maximum Values

We use dataset_minmax() with dataset as parameters that will lastly return the dataset. We take min ans max values as list of lists. Use for to wrap the first dataset with the column values in each row. Assign value_min to min(col_values) and value_max to max(col_values) We shall then join the minimum col_values and maximum col_values.

In [4]:

```
# Find the min and max values for each column
def dataset_minmax(dataset):
 minmax = list()
 for i in range(len(dataset[0])):
  col_values = [row[i] for row in dataset]
  value_min = min(col_values)
```

```
  value_max = max(col_values)
  minmax.append([value_min, value_max])
 return minmax
```

Normalisation Of Data

We shall use normalize_dataset() function with parameters dataset and minmax We use loop each row by taking values in a row and subtracting the min value then dividing by Max value minus minimum value. We apply the normalization formula (X - x(min)) / (x(max) - x(min))

In [5]:

```
# Rescale dataset columns to the range 0-1
def normalize_dataset(dataset, minmax):
 for row in dataset:
  for i in range(len(row)):
   row[i] = (row[i] - minmax[i][0]) / (minmax[i][1] - minmax[i][0])
```

Spliting Data Into K-folds

Each group of a data is called fold. We use cross_validation_split() function to split a dataset into train and test split. We assign to it parameters: dataset and n_folds. We take dataset_split as list of lists. We create a copy of the dataset from which to draw randomly chosen rows. We take the fold_size as total number of rows / total number of folds. Random rows are removed from the copied dataset to the fold_size until the total number of rows target is achieved. The randrange() function gives a random integer between 0 and the size of the list. The list of rows that we have created is added to the folds that is returned at the end.

In [6]:

```
# Split a dataset into k folds
def cross_validation_split(dataset, n_folds):
 dataset_split = list()
 dataset_copy = list(dataset)
 fold_size = int(len(dataset) / n_folds)
 for i in range(n_folds):
  fold = list()
  while len(fold) < fold_size:
   index = randrange(len(dataset_copy))
   fold.append(dataset_copy.pop(index))
  dataset_split.append(fold)
 return dataset_split
```

Calculating Rmse

We use rmse_metric() function with actual and predicted parameters. We initialise sum error to 0.0 To get the predicted error we subtract the actual value from the predicted value. We square the predicted error to avoid having negative values and assign it to sum_error. We assign mean_error to sum_error / actual value. Then we return sqrt with parameter mean_error at the end.

In [7]:

```
# Calculate root mean squared error
def rmse_metric(actual, predicted):
 sum_error = 0.0
 for i in range(len(actual)):
  prediction_error = predicted[i] - actual[i]
  sum_error += (prediction_error ** 2)
 mean_error = sum_error / float(len(actual))
 return sqrt(mean_error)
```

Evaluation Of An Algorithm Using A Cross Validation Split

We use evaluate_algorithm() function with 3 fixed parameters dataset, algorithm and n_folds. We split the data into n_folds called folds. We take scores as list of lists. We loop each fold, giving it an opportunity to be held out of the traing and be used in evaluating the algorithm. We split the data into training and test elements. We loop each row then create a copy of the fold and held out fold is removed from the list. The sum() function flattens the folds into a long list of rows to match the algorithms expectations of the training dataset. A copy of the test_dataset is made and output cleared, to avoid accidental cheating algorithms. Each output is cleared by setting value to None. The algorithm is a function that expects the trained and tesy datasets on which to make predictions We loop the predicted values from each row using the algorithm() function. The rmse_matric() function has parameters actual and predicted. It compares the predicted values to actual values of unmodified test dataset. The predicted values are stored in a list.

In [8]:

```
# Evaluate an algorithm using a cross validation split
def evaluate_algorithm(dataset, algorithm, n_folds, *args):
 folds = cross_validation_split(dataset, n_folds)
 scores = list()
 for fold in folds:
  train_set = list(folds)
  train_set.remove(fold)
  train_set = sum(train_set, [])
  test_set = list()
  for row in fold:
   row_copy = list(row)
   test_set.append(row_copy)
   row_copy[-1] = None
  predicted = algorithm(train_set, test_set, *args)
  actual = [row[-1] for row in fold]
  rmse = rmse_metric(actual, predicted)
  scores.append(rmse)
 return scores
```

Making Prediction

Let us start by defining predict function using row and coefficients parameters. Then we initialise inputs as 0 to get predicted values. We shall proceed by giving range of rows in array form Later, we iterate getting coefficients as per epoch for each instance. Lastly, we shall return the predicted value

In [9]:

```
# Make a prediction with coefficients
def predict(row, coefficients):
 yhat = coefficients[0]
 for i in range(len(row)-1):
  yhat += coefficients[i + 1] * row[i]
 return yhat
```

Estimation Of Linear Regression Coefficients Using SGD

First define coefficients_sgd function with train, l_rate and epoch parameters. We will use for loop 4 times. --> use for to set coeffient row(1) to 0. --> use for loop to give epoch range set sum error to 0 --> use for loop to train each row set yhat predict function with row, coef parameters. square the error gotten to get a positive value define the equation to be used. --> use for loop to give range of rows. print epoch as an int, l_rate as float and error as float. return the coefficients.

In [10]:

```
# Estimate linear regression coefficients using stochastic gradient descent
def coefficients_sgd(train, l_rate, n_epoch):
 coef = [0.0 for i in range(len(train[0]))]
 for epoch in range(n_epoch):
  for row in train:
   yhat = predict(row, coef)
   error = yhat - row[-1]
   coef[0] = coef[0] - l_rate * error
```

```
    for i in range(len(row)-1):
     coef[i + 1] = coef[i + 1] - l_rate * error * row[i]
    # print(l_rate, n_epoch, error)
  return coef
```

Linear Regression Algorithm Using SGD

We use linear_regression_sgd() function with train, test, l_rate and n_epoch parameters. We take predictions as list of lists. We take the coeffients_sgd() function and assign it coef value. We loop each row to take the inputs of the prediction in rows and coef. We join the inputs gotten and pass them through predictions which is later returned.

In [11]:

```
# Linear Regression Algorithm With Stochastic Gradient Descent
def linear_regression_sgd(train, test, l_rate, n_epoch):
 predictions = list()
 coef = coefficients_sgd(train, l_rate, n_epoch)
 for row in test:
  yhat = predict(row, coef)
  predictions.append(yhat)
 return(predictions)
```

Loading And Preparing Data

We assign a file from our directory a value. We loop each dataset to be read and convert the values from string to float We normalize the dataset by assigning it minmax value.

In [12]:

```
# Linear Regression on African Crises dataset
seed(1)
# load and prepare data
filename = 'African_Crises_Project1.csv'
dataset = load_csv(filename)
for i in range(len(dataset[0])):
 str_column_to_float(dataset, i)
# normalize
minmax = dataset_minmax(dataset)
normalize_dataset(dataset, minmax)
```

Evaluating Algorithm

We assign n_folds, l_rate and n_epoch values. We call coefficients_sgd() function and output the coefficients. We call evaluate_algorithm() function and output the scores. We loop each row to strip the whitspaces. We output the RMSE as a float.

In [13]:

```
# evaluate algorithm
n_folds = 5
l_rate = 0.01
n_epoch = 50

coef = coefficients_sgd(dataset, l_rate, n_epoch)
print(coef)


scores = evaluate_algorithm(dataset, linear_regression_sgd, n_folds, l_rate, n_epoch)
print('Scores: %s' % scores)
def str_column_to_int(dataset, column):
 for row in dataset:
  row[column] = int(row[column].strip())
print('Mean RMSE: %.3f' % (sum(scores)/float(len(scores))))
```

```
[0.23926865274441833, 0.29856866823060524]
Scores: [0.3392682096863841, 0.3122106838922133, 0.3340352614987483, 0.34963793531969556, 0.3400613084016554]
Mean RMSE: 0.335
```

Making A Prediction Using Coefficients

We have our coefficients as [0.2393, 0.2986] We can plug these coeffients manually in the equation y = B0 + B1*x. Where y is our inflation_crises and x our inflation_annual_cpi.Let us predict the values using the coefficients calculated. First we start by defining predict function having row and coefficents parameters. Initialise input as 0 to get predicted value. Give the range of rows in array form iterate getting co-efficients as per epoch for each instance return the predicted value. We convert string to float in order to multiply. We then insert the coefficients gotten above to predict the model. Use for loop to get the predicted output for each row Lastly print the expected output together with the predicted output.

In [16]:

```python
# Make a prediction with coefficients
def predict(row, coefficients):
 yhat = coefficients[0]
 for i in range(len(row)-1):
  yhat += coefficients[i + 1] * row[i]
 return yhat

filename = 'African_Crises_Project1.csv'
dataset = load_csv(filename)
for i in range(len(dataset[0])):
 str_column_to_float(dataset, i)
coef = [0.23926865274441833, 0.29856866823060524]
for row in dataset:
 yhat = predict(row, coef)
 print("Expected=%.3f, Predicted=%.3f" % (row[-1], yhat))
```

```
Expected=0.000, Predicted=1.267
Expected=0.000, Predicted=4.464
Expected=0.000, Predicted=-0.871
Expected=0.000, Predicted=3.584
Expected=0.000, Predicted=-0.910
Expected=0.000, Predicted=-6.008
Expected=0.000, Predicted=-0.289
Expected=1.000, Predicted=8.932
Expected=0.000, Predicted=-0.206
Expected=0.000, Predicted=-4.786
Expected=0.000, Predicted=1.398
Expected=0.000, Predicted=4.007
Expected=0.000, Predicted=-3.450
Expected=0.000, Predicted=-0.176
Expected=0.000, Predicted=-4.521
Expected=0.000, Predicted=3.152
Expected=1.000, Predicted=6.874
Expected=0.000, Predicted=5.125
Expected=1.000, Predicted=8.637
Expected=1.000, Predicted=14.075
Expected=1.000, Predicted=12.680
Expected=1.000, Predicted=9.021
Expected=0.000, Predicted=5.668
Expected=1.000, Predicted=20.909
Expected=1.000, Predicted=19.239
Expected=1.000, Predicted=7.289
Expected=0.000, Predicted=0.239
Expected=0.000, Predicted=2.252
Expected=0.000, Predicted=2.125
Expected=0.000, Predicted=-0.056
Expected=0.000, Predicted=0.538
Expected=0.000, Predicted=0.239
Expected=0.000, Predicted=0.830
Expected=0.000, Predicted=1.109
Expected=0.000, Predicted=3.901
Expected=0.000, Predicted=2.999
```

```
Expected=0.000, Predicted=1.847
Expected=0.000, Predicted=1.111
Expected=0.000, Predicted=2.210
Expected=0.000, Predicted=1.023
Expected=0.000, Predicted=1.331
Expected=0.000, Predicted=2.082
Expected=0.000, Predicted=1.653
Expected=0.000, Predicted=2.915
Expected=0.000, Predicted=2.890
Expected=0.000, Predicted=3.852
Expected=0.000, Predicted=5.367
Expected=0.000, Predicted=3.658
Expected=0.000, Predicted=3.126
Expected=0.000, Predicted=4.601
Expected=0.000, Predicted=3.126
Expected=0.000, Predicted=4.601
Expected=0.000, Predicted=2.208
Expected=0.000, Predicted=2.579
Expected=0.000, Predicted=2.123
Expected=0.000, Predicted=3.354
Expected=0.000, Predicted=4.421
Expected=0.000, Predicted=1.988
Expected=0.000, Predicted=2.012
Expected=0.000, Predicted=2.978
Expected=0.000, Predicted=3.008
Expected=1.000, Predicted=7.972
Expected=1.000, Predicted=9.704
Expected=1.000, Predicted=6.360
Expected=1.000, Predicted=8.898
Expected=1.000, Predicted=9.137
Expected=0.000, Predicted=5.823
Expected=0.000, Predicted=1.941
Expected=0.000, Predicted=1.717
Expected=0.000, Predicted=1.016
Expected=0.000, Predicted=0.329
Expected=0.000, Predicted=1.493
Expected=0.000, Predicted=0.666
Expected=0.000, Predicted=1.511
Expected=0.000, Predicted=1.425
Expected=0.000, Predicted=0.652
Expected=0.000, Predicted=0.930
Expected=0.000, Predicted=1.336
Expected=0.000, Predicted=1.689
Expected=0.000, Predicted=1.954
Expected=0.000, Predicted=1.408
Expected=0.000, Predicted=1.589
Expected=0.000, Predicted=2.901
Expected=0.000, Predicted=1.211
Expected=0.000, Predicted=1.110
Expected=1.000, Predicted=9.086
Expected=1.000, Predicted=13.888
Expected=1.000, Predicted=20.729
Expected=1.000, Predicted=38.081
Expected=0.000, Predicted=-7.876
Expected=0.000, Predicted=-1.863
Expected=0.000, Predicted=-4.963
Expected=0.000, Predicted=-2.500
Expected=0.000, Predicted=-1.269
Expected=0.000, Predicted=-1.349
Expected=0.000, Predicted=0.239
Expected=0.000, Predicted=-0.432
Expected=0.000, Predicted=0.926
Expected=0.000, Predicted=1.246
Expected=0.000, Predicted=1.862
Expected=0.000, Predicted=-0.069
Expected=0.000, Predicted=0.861
Expected=0.000, Predicted=0.849
Expected=0.000, Predicted=0.538
Expected=0.000, Predicted=4.378
Expected=0.000, Predicted=5.172
Expected=0.000, Predicted=5.587
Expected=0.000, Predicted=4.397
```

```
Expected=0.000, Predicted=2.561
Expected=0.000, Predicted=1.778
Expected=0.000, Predicted=1.703
Expected=0.000, Predicted=4.285
Expected=0.000, Predicted=-4.798
Expected=0.000, Predicted=-0.056
Expected=0.000, Predicted=1.434
Expected=0.000, Predicted=-2.057
Expected=0.000, Predicted=1.172
Expected=0.000, Predicted=0.239
Expected=0.000, Predicted=-0.967
Expected=0.000, Predicted=0.239
Expected=0.000, Predicted=1.182
Expected=0.000, Predicted=0.544
Expected=0.000, Predicted=0.842
Expected=0.000, Predicted=0.239
Expected=0.000, Predicted=-0.056
Expected=0.000, Predicted=0.836
Expected=0.000, Predicted=1.996
Expected=0.000, Predicted=2.617
Expected=0.000, Predicted=1.964
Expected=0.000, Predicted=4.956
Expected=0.000, Predicted=4.919
Expected=1.000, Predicted=8.425
Expected=1.000, Predicted=8.898
Expected=1.000, Predicted=24.334
Expected=1.000, Predicted=20.843
Expected=1.000, Predicted=14.708
Expected=1.000, Predicted=30.485
Expected=1.000, Predicted=14.185
Expected=1.000, Predicted=25.697
Expected=1.000, Predicted=89.540
Expected=1.000, Predicted=412.108
Expected=1.000, Predicted=283.811
Expected=1.000, Predicted=798.083
Expected=1.000, Predicted=1238.108
Expected=1.000, Predicted=66.370
Expected=1.000, Predicted=32.314
Expected=1.000, Predicted=74.358
Expected=1.000, Predicted=97.283
Expected=1.000, Predicted=45.797
Expected=1.000, Predicted=32.751
Expected=1.000, Predicted=29.601
Expected=1.000, Predicted=13.245
Expected=1.000, Predicted=7.095
Expected=0.000, Predicted=4.212
Expected=0.000, Predicted=3.896
Expected=0.000, Predicted=3.961
Expected=0.000, Predicted=4.336
Expected=0.000, Predicted=4.563
Expected=0.000, Predicted=4.265
Expected=0.000, Predicted=3.310
Expected=0.000, Predicted=2.861
Expected=0.000, Predicted=2.418
Expected=0.000, Predicted=2.275
Expected=0.000, Predicted=4.051
Expected=0.000, Predicted=1.366
Expected=0.000, Predicted=3.496
Expected=0.000, Predicted=2.197
Expected=0.000, Predicted=2.536
Expected=0.000, Predicted=1.348
Expected=0.000, Predicted=3.241
Expected=0.000, Predicted=3.042
Expected=0.000, Predicted=1.059
Expected=0.000, Predicted=0.804
Expected=0.000, Predicted=1.675
Expected=0.000, Predicted=0.395
Expected=0.000, Predicted=1.354
Expected=1.000, Predicted=8.509
Expected=0.000, Predicted=-2.395
Expected=0.000, Predicted=1.925
Expected=0.000, Predicted=3.088
```

```
Expected=0.000, Predicted=5.025
Expected=0.000, Predicted=3.377
Expected=0.000, Predicted=3.566
Expected=0.000, Predicted=3.670
Expected=0.000, Predicted=2.989
Expected=0.000, Predicted=4.210
Expected=0.000, Predicted=4.616
Expected=0.000, Predicted=4.193
Expected=0.000, Predicted=4.586
Expected=0.000, Predicted=1.017
Expected=0.000, Predicted=3.361
Expected=0.000, Predicted=0.959
Expected=0.000, Predicted=-1.847
Expected=0.000, Predicted=-0.935
Expected=0.000, Predicted=0.432
Expected=0.000, Predicted=0.179
Expected=0.000, Predicted=-0.611
Expected=0.000, Predicted=0.013
Expected=0.000, Predicted=-0.629
Expected=1.000, Predicted=7.569
Expected=0.000, Predicted=5.972
Expected=0.000, Predicted=1.351
Expected=0.000, Predicted=0.716
Expected=0.000, Predicted=-0.319
Expected=0.000, Predicted=-0.183
Expected=0.000, Predicted=1.195
Expected=0.000, Predicted=1.387
Expected=0.000, Predicted=0.925
Expected=0.000, Predicted=1.539
Expected=0.000, Predicted=-0.431
Expected=0.000, Predicted=1.101
Expected=0.000, Predicted=2.238
Expected=0.000, Predicted=0.518
Expected=0.000, Predicted=3.005
Expected=0.000, Predicted=1.291
Expected=0.000, Predicted=0.684
Expected=0.000, Predicted=0.596
Expected=0.000, Predicted=1.993
Expected=0.000, Predicted=2.195
Expected=0.000, Predicted=3.698
Expected=0.000, Predicted=5.081
Expected=0.000, Predicted=0.934
Expected=0.000, Predicted=-0.439
Expected=0.000, Predicted=0.934
Expected=0.000, Predicted=1.596
Expected=0.000, Predicted=4.134
Expected=1.000, Predicted=6.555
Expected=0.000, Predicted=2.135
Expected=0.000, Predicted=-0.403
Expected=0.000, Predicted=4.548
Expected=0.000, Predicted=0.186
Expected=0.000, Predicted=2.843
Expected=0.000, Predicted=-0.707
Expected=0.000, Predicted=0.764
Expected=0.000, Predicted=1.632
Expected=0.000, Predicted=1.065
Expected=0.000, Predicted=2.552
Expected=0.000, Predicted=0.987
Expected=0.000, Predicted=2.650
Expected=0.000, Predicted=-0.081
Expected=0.000, Predicted=0.596
Expected=0.000, Predicted=4.101
Expected=0.000, Predicted=5.680
Expected=0.000, Predicted=3.409
Expected=0.000, Predicted=4.093
Expected=1.000, Predicted=7.549
Expected=0.000, Predicted=5.026
Expected=1.000, Predicted=6.736
Expected=0.000, Predicted=2.870
Expected=0.000, Predicted=2.831
Expected=0.000, Predicted=2.445
Expected=0.000, Predicted=1.995
```

```
Expected=0.000, Predicted=1.517
Expected=0.000, Predicted=0.763
Expected=0.000, Predicted=2.281
Expected=0.000, Predicted=2.323
Expected=0.000, Predicted=2.310
Expected=0.000, Predicted=0.534
Expected=0.000, Predicted=0.043
Expected=0.000, Predicted=0.710
Expected=0.000, Predicted=1.499
Expected=0.000, Predicted=0.875
Expected=1.000, Predicted=7.989
Expected=0.000, Predicted=4.449
Expected=0.000, Predicted=1.045
Expected=0.000, Predicted=2.121
Expected=0.000, Predicted=1.797
Expected=0.000, Predicted=0.514
Expected=0.000, Predicted=0.126
Expected=0.000, Predicted=1.540
Expected=0.000, Predicted=1.159
Expected=0.000, Predicted=1.224
Expected=0.000, Predicted=0.675
Expected=0.000, Predicted=1.399
Expected=0.000, Predicted=0.976
Expected=0.000, Predicted=0.805
Expected=0.000, Predicted=2.125
Expected=0.000, Predicted=0.541
Expected=0.000, Predicted=0.777
Expected=0.000, Predicted=1.567
Expected=0.000, Predicted=0.627
Expected=0.000, Predicted=1.011
Expected=0.000, Predicted=0.373
Expected=0.000, Predicted=2.014
Expected=0.000, Predicted=2.017
Expected=0.000, Predicted=2.013
Expected=0.000, Predicted=2.016
Expected=0.000, Predicted=2.016
Expected=0.000, Predicted=2.014
Expected=0.000, Predicted=2.013
Expected=0.000, Predicted=2.016
Expected=0.000, Predicted=2.016
Expected=0.000, Predicted=2.012
Expected=0.000, Predicted=2.017
Expected=0.000, Predicted=4.114
Expected=0.000, Predicted=-1.015
Expected=0.000, Predicted=-1.006
Expected=0.000, Predicted=0.704
Expected=0.000, Predicted=-2.307
Expected=0.000, Predicted=-3.007
Expected=0.000, Predicted=0.109
Expected=0.000, Predicted=0.086
Expected=0.000, Predicted=3.390
Expected=0.000, Predicted=0.772
Expected=0.000, Predicted=-3.465
Expected=0.000, Predicted=1.503
Expected=0.000, Predicted=-0.427
Expected=0.000, Predicted=0.097
Expected=0.000, Predicted=-1.683
Expected=0.000, Predicted=1.319
Expected=0.000, Predicted=-0.609
Expected=0.000, Predicted=-1.617
Expected=0.000, Predicted=-3.443
Expected=0.000, Predicted=2.613
Expected=0.000, Predicted=3.277
Expected=0.000, Predicted=0.824
Expected=0.000, Predicted=-3.918
Expected=0.000, Predicted=-1.240
Expected=0.000, Predicted=-1.426
Expected=0.000, Predicted=3.137
Expected=0.000, Predicted=1.089
Expected=0.000, Predicted=4.484
Expected=0.000, Predicted=-1.664
Expected=0.000, Predicted=2.445
```

```
Expected=0.000, Predicted=0.272
Expected=0.000, Predicted=0.075
Expected=0.000, Predicted=-2.010
Expected=0.000, Predicted=-0.549
Expected=0.000, Predicted=4.826
Expected=0.000, Predicted=3.060
Expected=0.000, Predicted=0.630
Expected=0.000, Predicted=2.888
Expected=0.000, Predicted=0.593
Expected=0.000, Predicted=-3.266
Expected=0.000, Predicted=0.625
Expected=0.000, Predicted=2.104
Expected=0.000, Predicted=3.644
Expected=0.000, Predicted=-0.100
Expected=0.000, Predicted=-4.538
Expected=0.000, Predicted=6.010
Expected=1.000, Predicted=10.832
Expected=0.000, Predicted=5.627
Expected=0.000, Predicted=3.874
Expected=1.000, Predicted=6.391
Expected=0.000, Predicted=-8.271
Expected=0.000, Predicted=-3.286
Expected=0.000, Predicted=-3.016
Expected=0.000, Predicted=2.389
Expected=0.000, Predicted=1.936
Expected=0.000, Predicted=-3.151
Expected=0.000, Predicted=-2.256
Expected=0.000, Predicted=1.585
Expected=0.000, Predicted=-0.758
Expected=0.000, Predicted=-1.828
Expected=0.000, Predicted=-1.260
Expected=0.000, Predicted=-3.000
Expected=0.000, Predicted=-2.685
Expected=1.000, Predicted=7.396
Expected=0.000, Predicted=1.926
Expected=0.000, Predicted=-1.925
Expected=0.000, Predicted=-0.066
Expected=0.000, Predicted=3.653
Expected=0.000, Predicted=-0.189
Expected=0.000, Predicted=3.339
Expected=1.000, Predicted=7.461
Expected=1.000, Predicted=12.411
Expected=0.000, Predicted=5.460
Expected=0.000, Predicted=3.637
Expected=0.000, Predicted=0.239
Expected=0.000, Predicted=0.872
Expected=0.000, Predicted=-0.690
Expected=0.000, Predicted=0.452
Expected=0.000, Predicted=0.239
Expected=0.000, Predicted=2.780
Expected=0.000, Predicted=2.508
Expected=0.000, Predicted=-2.823
Expected=0.000, Predicted=-0.045
Expected=0.000, Predicted=-0.622
Expected=0.000, Predicted=0.239
Expected=0.000, Predicted=1.422
Expected=0.000, Predicted=1.209
Expected=0.000, Predicted=-0.161
Expected=0.000, Predicted=0.447
Expected=0.000, Predicted=0.456
Expected=0.000, Predicted=0.591
Expected=0.000, Predicted=-1.016
Expected=0.000, Predicted=0.855
Expected=0.000, Predicted=2.575
Expected=0.000, Predicted=4.157
Expected=0.000, Predicted=2.238
Expected=0.000, Predicted=0.275
Expected=0.000, Predicted=0.375
Expected=0.000, Predicted=1.337
Expected=0.000, Predicted=1.245
Expected=0.000, Predicted=1.118
Expected=0.000, Predicted=0.963
```

```
Expected=0.000, Predicted=1.953
Expected=0.000, Predicted=3.416
Expected=0.000, Predicted=3.195
Expected=0.000, Predicted=3.312
Expected=0.000, Predicted=3.707
Expected=0.000, Predicted=3.604
Expected=0.000, Predicted=3.196
Expected=1.000, Predicted=6.360
Expected=0.000, Predicted=3.344
Expected=0.000, Predicted=4.688
Expected=0.000, Predicted=5.011
Expected=0.000, Predicted=5.333
Expected=0.000, Predicted=3.854
Expected=1.000, Predicted=7.375
Expected=1.000, Predicted=7.759
Expected=0.000, Predicted=4.773
Expected=1.000, Predicted=6.249
Expected=1.000, Predicted=6.575
Expected=0.000, Predicted=4.639
Expected=1.000, Predicted=6.552
Expected=0.000, Predicted=3.536
Expected=0.000, Predicted=2.940
Expected=0.000, Predicted=3.034
Expected=0.000, Predicted=2.358
Expected=0.000, Predicted=2.081
Expected=0.000, Predicted=1.744
Expected=0.000, Predicted=1.357
Expected=0.000, Predicted=1.090
Expected=0.000, Predicted=0.965
Expected=0.000, Predicted=1.198
Expected=0.000, Predicted=0.239
Expected=0.000, Predicted=2.659
Expected=0.000, Predicted=2.874
Expected=0.000, Predicted=1.494
Expected=0.000, Predicted=3.511
Expected=0.000, Predicted=3.732
Expected=0.000, Predicted=5.088
Expected=0.000, Predicted=3.730
Expected=0.000, Predicted=3.550
Expected=0.000, Predicted=2.822
Expected=0.000, Predicted=2.304
Expected=0.000, Predicted=3.255
Expected=0.000, Predicted=3.028
Expected=0.000, Predicted=1.291
Expected=0.000, Predicted=2.793
Expected=0.000, Predicted=3.487
Expected=0.000, Predicted=2.165
Expected=0.000, Predicted=1.145
Expected=0.000, Predicted=1.556
Expected=0.000, Predicted=2.141
Expected=0.000, Predicted=0.773
Expected=0.000, Predicted=1.025
Expected=0.000, Predicted=0.239
Expected=0.000, Predicted=0.495
Expected=0.000, Predicted=0.492
Expected=0.000, Predicted=0.992
Expected=0.000, Predicted=1.058
Expected=0.000, Predicted=0.239
Expected=0.000, Predicted=0.154
Expected=0.000, Predicted=1.960
Expected=0.000, Predicted=1.362
Expected=0.000, Predicted=0.400
Expected=0.000, Predicted=0.437
Expected=0.000, Predicted=0.200
Expected=0.000, Predicted=0.786
Expected=0.000, Predicted=1.916
Expected=0.000, Predicted=1.590
Expected=0.000, Predicted=4.020
Expected=0.000, Predicted=5.140
Expected=0.000, Predicted=6.131
Expected=0.000, Predicted=3.084
Expected=0.000, Predicted=5.617
```

```
Expected=0.000, Predicted=4.824
Expected=0.000, Predicted=2.652
Expected=0.000, Predicted=4.379
Expected=0.000, Predicted=2.596
Expected=0.000, Predicted=4.366
Expected=0.000, Predicted=3.704
Expected=1.000, Predicted=6.410
Expected=0.000, Predicted=3.642
Expected=0.000, Predicted=3.310
Expected=0.000, Predicted=4.123
Expected=0.000, Predicted=1.674
Expected=0.000, Predicted=2.513
Expected=0.000, Predicted=3.583
Expected=0.000, Predicted=5.943
Expected=1.000, Predicted=8.400
Expected=1.000, Predicted=13.967
Expected=1.000, Predicted=8.842
Expected=0.000, Predicted=0.703
Expected=0.000, Predicted=2.885
Expected=0.000, Predicted=3.799
Expected=0.000, Predicted=2.244
Expected=0.000, Predicted=1.957
Expected=0.000, Predicted=3.212
Expected=0.000, Predicted=1.978
Expected=0.000, Predicted=0.883
Expected=0.000, Predicted=2.026
Expected=0.000, Predicted=2.742
Expected=0.000, Predicted=2.575
Expected=0.000, Predicted=2.043
Expected=0.000, Predicted=1.513
Expected=0.000, Predicted=4.748
Expected=0.000, Predicted=3.390
Expected=0.000, Predicted=1.526
Expected=0.000, Predicted=4.426
Expected=0.000, Predicted=3.039
Expected=0.000, Predicted=1.946
Expected=0.000, Predicted=2.293
Expected=0.000, Predicted=-2.475
Expected=1.000, Predicted=9.793
Expected=0.000, Predicted=0.545
Expected=0.000, Predicted=1.704
Expected=0.000, Predicted=3.569
Expected=0.000, Predicted=0.819
Expected=0.000, Predicted=0.050
Expected=0.000, Predicted=0.525
Expected=0.000, Predicted=0.145
Expected=0.000, Predicted=1.092
Expected=0.000, Predicted=-0.037
Expected=0.000, Predicted=0.425
Expected=0.000, Predicted=0.055
Expected=0.000, Predicted=0.797
Expected=0.000, Predicted=0.056
Expected=0.000, Predicted=0.478
Expected=0.000, Predicted=0.763
Expected=0.000, Predicted=0.785
Expected=0.000, Predicted=0.805
Expected=0.000, Predicted=1.130
Expected=0.000, Predicted=1.116
Expected=0.000, Predicted=1.831
Expected=0.000, Predicted=1.129
Expected=0.000, Predicted=0.243
Expected=0.000, Predicted=0.802
Expected=0.000, Predicted=1.897
Expected=1.000, Predicted=6.456
Expected=1.000, Predicted=7.345
Expected=0.000, Predicted=4.779
Expected=0.000, Predicted=3.452
Expected=0.000, Predicted=3.164
Expected=0.000, Predicted=2.945
Expected=1.000, Predicted=7.171
Expected=1.000, Predicted=10.104
Expected=1.000, Predicted=8.136
```

```
Expected=0.000, Predicted=4.235
Expected=0.000, Predicted=2.464
Expected=0.000, Predicted=1.889
Expected=0.000, Predicted=2.724
Expected=0.000, Predicted=1.534
Expected=0.000, Predicted=0.454
Expected=0.000, Predicted=1.314
Expected=0.000, Predicted=4.303
Expected=0.000, Predicted=3.435
Expected=0.000, Predicted=4.066
Expected=0.000, Predicted=1.092
Expected=0.000, Predicted=4.804
Expected=0.000, Predicted=2.429
Expected=0.000, Predicted=2.038
Expected=0.000, Predicted=2.196
Expected=0.000, Predicted=2.279
Expected=0.000, Predicted=2.272
Expected=0.000, Predicted=2.291
Expected=0.000, Predicted=1.502
Expected=0.000, Predicted=1.849
Expected=0.000, Predicted=2.154
Expected=0.000, Predicted=1.413
Expected=0.000, Predicted=1.643
Expected=0.000, Predicted=1.709
Expected=0.000, Predicted=2.905
Expected=0.000, Predicted=2.875
Expected=0.000, Predicted=3.145
Expected=0.000, Predicted=0.990
Expected=0.000, Predicted=1.114
Expected=0.000, Predicted=2.188
Expected=0.000, Predicted=1.703
Expected=0.000, Predicted=1.473
Expected=0.000, Predicted=1.365
Expected=1.000, Predicted=7.703
Expected=1.000, Predicted=9.793
Expected=1.000, Predicted=11.643
Expected=1.000, Predicted=8.750
Expected=1.000, Predicted=10.633
Expected=0.000, Predicted=5.908
Expected=1.000, Predicted=17.185
Expected=1.000, Predicted=17.399
Expected=1.000, Predicted=12.226
Expected=1.000, Predicted=7.106
Expected=0.000, Predicted=0.725
Expected=1.000, Predicted=7.166
Expected=0.000, Predicted=1.790
Expected=0.000, Predicted=-0.129
Expected=0.000, Predicted=0.985
Expected=0.000, Predicted=1.990
Expected=0.000, Predicted=1.946
Expected=0.000, Predicted=0.931
Expected=0.000, Predicted=1.548
Expected=0.000, Predicted=-0.409
Expected=0.000, Predicted=0.902
Expected=0.000, Predicted=0.919
Expected=0.000, Predicted=1.854
Expected=0.000, Predicted=1.953
Expected=0.000, Predicted=1.503
Expected=0.000, Predicted=0.980
Expected=0.000, Predicted=0.169
Expected=0.000, Predicted=-0.097
Expected=0.000, Predicted=0.579
Expected=0.000, Predicted=0.744
Expected=0.000, Predicted=0.811
Expected=0.000, Predicted=1.552
Expected=0.000, Predicted=1.198
Expected=0.000, Predicted=2.387
Expected=0.000, Predicted=5.010
Expected=0.000, Predicted=2.331
Expected=0.000, Predicted=3.516
Expected=0.000, Predicted=3.468
Expected=0.000, Predicted=3.141
```

```
Expected=0.000, Predicted=2.822
Expected=0.000, Predicted=3.048
Expected=0.000, Predicted=3.969
Expected=0.000, Predicted=3.383
Expected=0.000, Predicted=2.093
Expected=0.000, Predicted=3.956
Expected=0.000, Predicted=2.547
Expected=0.000, Predicted=2.847
Expected=0.000, Predicted=1.045
Expected=0.000, Predicted=0.947
Expected=0.000, Predicted=1.176
Expected=0.000, Predicted=2.038
Expected=0.000, Predicted=2.924
Expected=0.000, Predicted=1.953
Expected=0.000, Predicted=1.787
Expected=0.000, Predicted=1.775
Expected=0.000, Predicted=2.068
Expected=0.000, Predicted=1.131
Expected=0.000, Predicted=0.550
Expected=0.000, Predicted=1.059
Expected=0.000, Predicted=0.445
Expected=0.000, Predicted=0.813
Expected=0.000, Predicted=0.422
Expected=0.000, Predicted=1.069
Expected=0.000, Predicted=0.587
Expected=0.000, Predicted=0.685
Expected=0.000, Predicted=0.533
Expected=0.000, Predicted=1.220
Expected=0.000, Predicted=0.847
Expected=0.000, Predicted=1.401
Expected=0.000, Predicted=0.529
Expected=0.000, Predicted=0.536
Expected=0.000, Predicted=0.510
Expected=0.000, Predicted=0.624
Expected=0.000, Predicted=0.801
Expected=0.000, Predicted=0.372
Expected=0.000, Predicted=1.166
Expected=0.000, Predicted=2.083
Expected=0.000, Predicted=2.288
Expected=0.000, Predicted=0.701
Expected=0.000, Predicted=-0.776
Expected=0.000, Predicted=1.645
Expected=0.000, Predicted=1.531
Expected=0.000, Predicted=2.096
Expected=0.000, Predicted=1.109
Expected=0.000, Predicted=-0.068
Expected=0.000, Predicted=0.519
Expected=0.000, Predicted=1.613
Expected=0.000, Predicted=2.670
Expected=0.000, Predicted=-1.120
Expected=0.000, Predicted=1.055
Expected=0.000, Predicted=3.524
Expected=0.000, Predicted=4.244
Expected=0.000, Predicted=4.861
Expected=0.000, Predicted=0.237
Expected=0.000, Predicted=3.725
Expected=0.000, Predicted=3.601
Expected=1.000, Predicted=11.735
Expected=0.000, Predicted=5.714
Expected=1.000, Predicted=6.914
Expected=0.000, Predicted=4.951
Expected=0.000, Predicted=3.245
Expected=0.000, Predicted=3.216
Expected=0.000, Predicted=6.376
Expected=1.000, Predicted=1.995
Expected=0.000, Predicted=6.874
Expected=1.000, Predicted=12.453
Expected=1.000, Predicted=1.202
Expected=0.000, Predicted=2.105
Expected=0.000, Predicted=3.752
Expected=0.000, Predicted=10.454
Expected=1.000, Predicted=14.875
```

```
Expected=1.000, Predicted=2.596
Expected=0.000, Predicted=13.545
Expected=1.000, Predicted=17.300
Expected=1.000, Predicted=17.382
Expected=1.000, Predicted=21.954
Expected=1.000, Predicted=8.985
Expected=1.000, Predicted=3.426
Expected=0.000, Predicted=2.587
Expected=0.000, Predicted=2.215
Expected=0.000, Predicted=2.311
Expected=0.000, Predicted=5.873
Expected=0.000, Predicted=4.086
Expected=0.000, Predicted=4.429
Expected=0.000, Predicted=4.718
Expected=0.000, Predicted=5.571
Expected=0.000, Predicted=2.693
Expected=0.000, Predicted=1.855
Expected=0.000, Predicted=3.697
Expected=0.000, Predicted=3.984
Expected=0.000, Predicted=4.336
Expected=0.000, Predicted=3.476
Expected=0.000, Predicted=3.889
Expected=0.000, Predicted=2.776
Expected=0.000, Predicted=2.642
Expected=0.000, Predicted=1.466
Expected=0.000, Predicted=2.987
Expected=0.000, Predicted=0.960
Expected=0.000, Predicted=-2.220
Expected=0.000, Predicted=-1.292
Expected=0.000, Predicted=-1.777
Expected=0.000, Predicted=-1.926
Expected=0.000, Predicted=-0.225
Expected=0.000, Predicted=-0.236
Expected=0.000, Predicted=1.684
Expected=0.000, Predicted=4.833
Expected=0.000, Predicted=1.433
Expected=0.000, Predicted=1.004
Expected=0.000, Predicted=0.613
Expected=0.000, Predicted=0.609
Expected=0.000, Predicted=1.694
Expected=0.000, Predicted=1.976
Expected=0.000, Predicted=3.192
Expected=0.000, Predicted=2.330
Expected=0.000, Predicted=0.517
Expected=1.000, Predicted=10.745
Expected=0.000, Predicted=-5.157
Expected=0.000, Predicted=-0.942
Expected=0.000, Predicted=0.466
Expected=0.000, Predicted=0.239
Expected=0.000, Predicted=-0.210
Expected=0.000, Predicted=-0.216
Expected=0.000, Predicted=0.933
Expected=0.000, Predicted=0.013
Expected=0.000, Predicted=-0.216
Expected=0.000, Predicted=-0.456
Expected=0.000, Predicted=-0.946
Expected=0.000, Predicted=-1.383
Expected=0.000, Predicted=0.956
Expected=0.000, Predicted=-0.195
Expected=0.000, Predicted=0.105
Expected=0.000, Predicted=0.510
Expected=0.000, Predicted=1.923
Expected=0.000, Predicted=0.143
Expected=0.000, Predicted=0.496
Expected=0.000, Predicted=1.411
Expected=0.000, Predicted=2.039
Expected=0.000, Predicted=2.828
Expected=0.000, Predicted=1.721
Expected=0.000, Predicted=1.399
Expected=0.000, Predicted=0.467
Expected=0.000, Predicted=2.049
Expected=0.000, Predicted=1.519
```

```
Expected=0.000, Predicted=2.284
Expected=0.000, Predicted=1.005
Expected=0.000, Predicted=2.292
Expected=0.000, Predicted=2.524
Expected=0.000, Predicted=2.213
Expected=0.000, Predicted=0.471
Expected=0.000, Predicted=1.387
Expected=0.000, Predicted=0.747
Expected=0.000, Predicted=0.803
Expected=0.000, Predicted=1.214
Expected=0.000, Predicted=1.183
Expected=0.000, Predicted=0.474
Expected=0.000, Predicted=0.712
Expected=0.000, Predicted=0.860
Expected=0.000, Predicted=0.635
Expected=0.000, Predicted=0.539
Expected=0.000, Predicted=1.476
Expected=0.000, Predicted=1.281
Expected=0.000, Predicted=1.326
Expected=0.000, Predicted=0.882
Expected=0.000, Predicted=1.149
Expected=0.000, Predicted=1.378
Expected=0.000, Predicted=1.697
Expected=0.000, Predicted=2.241
Expected=0.000, Predicted=2.757
Expected=0.000, Predicted=3.454
Expected=0.000, Predicted=4.363
Expected=0.000, Predicted=3.598
Expected=0.000, Predicted=3.618
Expected=0.000, Predicted=3.288
Expected=0.000, Predicted=3.892
Expected=0.000, Predicted=4.387
Expected=0.000, Predicted=4.490
Expected=0.000, Predicted=4.939
Expected=0.000, Predicted=4.539
Expected=0.000, Predicted=3.997
Expected=0.000, Predicted=3.577
Expected=0.000, Predicted=5.076
Expected=0.000, Predicted=5.837
Expected=0.000, Predicted=5.074
Expected=0.000, Predicted=4.088
Expected=0.000, Predicted=4.570
Expected=0.000, Predicted=4.505
Expected=0.000, Predicted=4.887
Expected=0.000, Predicted=4.322
Expected=0.000, Predicted=3.187
Expected=0.000, Predicted=2.874
Expected=0.000, Predicted=2.840
Expected=0.000, Predicted=2.425
Expected=0.000, Predicted=2.814
Expected=0.000, Predicted=2.291
Expected=0.000, Predicted=1.795
Expected=0.000, Predicted=1.844
Expected=0.000, Predicted=1.941
Expected=0.000, Predicted=2.979
Expected=0.000, Predicted=1.973
Expected=0.000, Predicted=0.655
Expected=0.000, Predicted=1.252
Expected=0.000, Predicted=1.639
Expected=0.000, Predicted=2.356
Expected=0.000, Predicted=3.684
Expected=0.000, Predicted=2.368
Expected=0.000, Predicted=1.510
Expected=0.000, Predicted=1.732
Expected=0.000, Predicted=1.927
Expected=0.000, Predicted=1.957
Expected=1.000, Predicted=6.812
Expected=1.000, Predicted=7.579
Expected=1.000, Predicted=7.702
Expected=1.000, Predicted=21.769
Expected=1.000, Predicted=11.469
Expected=0.000, Predicted=4.353
```

```
Expected=1.000, Predicted=15.052
Expected=1.000, Predicted=20.273
Expected=1.000, Predicted=20.883
Expected=0.000, Predicted=1.480
Expected=0.000, Predicted=1.285
Expected=0.000, Predicted=4.729
Expected=0.000, Predicted=2.331
Expected=0.000, Predicted=0.062
Expected=0.000, Predicted=1.491
Expected=0.000, Predicted=0.239
Expected=0.000, Predicted=1.675
Expected=0.000, Predicted=1.883
Expected=0.000, Predicted=1.797
Expected=0.000, Predicted=-2.228
Expected=0.000, Predicted=0.777
Expected=0.000, Predicted=-0.289
Expected=0.000, Predicted=1.966
Expected=0.000, Predicted=0.529
Expected=0.000, Predicted=2.013
Expected=0.000, Predicted=1.896
Expected=0.000, Predicted=1.594
Expected=0.000, Predicted=1.088
Expected=0.000, Predicted=0.985
Expected=0.000, Predicted=1.077
Expected=0.000, Predicted=0.706
Expected=0.000, Predicted=1.509
Expected=0.000, Predicted=1.050
Expected=0.000, Predicted=1.966
Expected=0.000, Predicted=1.638
Expected=0.000, Predicted=3.242
Expected=0.000, Predicted=1.222
Expected=0.000, Predicted=2.686
Expected=0.000, Predicted=1.622
Expected=0.000, Predicted=2.794
Expected=0.000, Predicted=3.228
Expected=0.000, Predicted=2.897
Expected=0.000, Predicted=4.322
Expected=0.000, Predicted=2.918
Expected=0.000, Predicted=2.806
Expected=0.000, Predicted=2.494
Expected=0.000, Predicted=2.078
Expected=0.000, Predicted=2.695
Expected=0.000, Predicted=2.376
Expected=0.000, Predicted=2.545
Expected=0.000, Predicted=2.181
Expected=0.000, Predicted=2.536
Expected=0.000, Predicted=1.887
Expected=0.000, Predicted=1.445
Expected=0.000, Predicted=1.858
Expected=0.000, Predicted=2.100
Expected=0.000, Predicted=1.354
Expected=0.000, Predicted=1.314
Expected=0.000, Predicted=1.165
Expected=0.000, Predicted=1.066
Expected=0.000, Predicted=1.066
Expected=0.000, Predicted=0.833
Expected=0.000, Predicted=1.047
Expected=0.000, Predicted=1.053
Expected=0.000, Predicted=1.320
Expected=0.000, Predicted=0.670
Expected=0.000, Predicted=1.476
Expected=0.000, Predicted=1.266
Expected=0.000, Predicted=1.706
Expected=0.000, Predicted=1.293
Expected=0.000, Predicted=1.556
Expected=0.000, Predicted=1.297
Expected=0.000, Predicted=1.774
Expected=0.000, Predicted=1.972
Expected=0.000, Predicted=1.709
Expected=0.000, Predicted=1.092
Expected=0.000, Predicted=1.069
Expected=0.000, Predicted=1.046
```

```
Expected=0.000, Predicted=2.596
Expected=0.000, Predicted=0.967
Expected=0.000, Predicted=1.661
Expected=0.000, Predicted=0.918
Expected=0.000, Predicted=1.566
Expected=0.000, Predicted=2.145
Expected=0.000, Predicted=1.434
Expected=0.000, Predicted=1.388
Expected=0.000, Predicted=1.345
Expected=0.000, Predicted=1.306
Expected=0.000, Predicted=1.166
Expected=0.000, Predicted=1.188
Expected=0.000, Predicted=0.578
Expected=0.000, Predicted=0.861
Expected=0.000, Predicted=1.270
Expected=0.000, Predicted=0.285
Expected=0.000, Predicted=0.646
Expected=0.000, Predicted=0.105
Expected=0.000, Predicted=1.136
Expected=0.000, Predicted=2.677
Expected=0.000, Predicted=3.297
Expected=0.000, Predicted=1.736
Expected=0.000, Predicted=3.437
Expected=0.000, Predicted=0.993
Expected=0.000, Predicted=1.005
Expected=0.000, Predicted=2.186
Expected=0.000, Predicted=1.661
Expected=0.000, Predicted=2.178
Expected=0.000, Predicted=2.606
Expected=0.000, Predicted=3.276
Expected=0.000, Predicted=5.904
Expected=0.000, Predicted=6.159
Expected=0.000, Predicted=5.072
Expected=0.000, Predicted=3.105
Expected=0.000, Predicted=3.741
Expected=0.000, Predicted=4.418
Expected=0.000, Predicted=3.970
Expected=0.000, Predicted=6.119
Expected=1.000, Predicted=6.216
Expected=1.000, Predicted=11.415
Expected=1.000, Predicted=16.601
Expected=1.000, Predicted=14.280
Expected=1.000, Predicted=16.375
Expected=1.000, Predicted=38.544
Expected=1.000, Predicted=32.950
Expected=1.000, Predicted=29.410
Expected=1.000, Predicted=49.720
Expected=1.000, Predicted=54.956
Expected=1.000, Predicted=16.545
Expected=1.000, Predicted=10.661
Expected=1.000, Predicted=13.106
Expected=1.000, Predicted=7.527
Expected=1.000, Predicted=7.541
Expected=1.000, Predicted=8.238
Expected=1.000, Predicted=8.032
Expected=1.000, Predicted=6.616
Expected=1.000, Predicted=6.879
Expected=1.000, Predicted=6.629
Expected=0.000, Predicted=5.604
Expected=0.000, Predicted=5.711
Expected=0.000, Predicted=2.931
Expected=0.000, Predicted=3.421
Expected=0.000, Predicted=3.956
Expected=0.000, Predicted=4.238
Expected=0.000, Predicted=2.777
Expected=0.000, Predicted=2.824
Expected=0.000, Predicted=2.202
Expected=0.000, Predicted=2.323
Expected=0.000, Predicted=2.571
Expected=0.000, Predicted=-4.908
Expected=0.000, Predicted=-3.908
Expected=0.000, Predicted=-0.965
```

```
Expected=0.000, Predicted=0.492
Expected=0.000, Predicted=0.490
Expected=0.000, Predicted=0.239
Expected=0.000, Predicted=-1.254
Expected=0.000, Predicted=-1.856
Expected=0.000, Predicted=-0.887
Expected=0.000, Predicted=-0.639
Expected=0.000, Predicted=-0.062
Expected=0.000, Predicted=0.239
Expected=0.000, Predicted=-0.065
Expected=0.000, Predicted=2.086
Expected=0.000, Predicted=-0.630
Expected=0.000, Predicted=0.510
Expected=0.000, Predicted=1.133
Expected=0.000, Predicted=1.368
Expected=0.000, Predicted=1.912
Expected=0.000, Predicted=2.061
Expected=0.000, Predicted=1.135
Expected=0.000, Predicted=1.036
Expected=0.000, Predicted=1.792
Expected=0.000, Predicted=1.111
Expected=0.000, Predicted=3.042
Expected=0.000, Predicted=1.372
Expected=0.000, Predicted=2.536
Expected=0.000, Predicted=2.319
Expected=0.000, Predicted=2.682
Expected=0.000, Predicted=1.069
Expected=0.000, Predicted=0.239
Expected=0.000, Predicted=0.732
Expected=0.000, Predicted=1.562
Expected=0.000, Predicted=1.126
Expected=0.000, Predicted=1.306
Expected=0.000, Predicted=1.031
Expected=0.000, Predicted=0.972
Expected=0.000, Predicted=1.105
Expected=0.000, Predicted=0.861
Expected=0.000, Predicted=0.562
Expected=0.000, Predicted=0.986
Expected=0.000, Predicted=0.986
Expected=0.000, Predicted=1.171
Expected=0.000, Predicted=0.945
Expected=0.000, Predicted=0.653
Expected=0.000, Predicted=0.348
Expected=0.000, Predicted=0.863
Expected=0.000, Predicted=1.142
Expected=0.000, Predicted=1.090
Expected=0.000, Predicted=1.167
Expected=0.000, Predicted=2.209
Expected=0.000, Predicted=3.227
Expected=0.000, Predicted=3.536
Expected=0.000, Predicted=3.302
Expected=0.000, Predicted=2.696
Expected=0.000, Predicted=3.987
Expected=0.000, Predicted=2.389
Expected=0.000, Predicted=4.180
Expected=0.000, Predicted=3.344
Expected=1.000, Predicted=7.166
Expected=1.000, Predicted=6.300
Expected=0.000, Predicted=2.717
Expected=0.000, Predicted=4.569
Expected=0.000, Predicted=3.971
Expected=0.000, Predicted=2.449
Expected=0.000, Predicted=4.061
Expected=0.000, Predicted=5.434
Expected=1.000, Predicted=7.405
Expected=1.000, Predicted=12.660
Expected=1.000, Predicted=8.659
Expected=1.000, Predicted=6.543
Expected=1.000, Predicted=7.945
Expected=0.000, Predicted=5.136
Expected=1.000, Predicted=6.230
Expected=1.000, Predicted=14.156
```

```
Expected=1.000, Predicted=17.235
Expected=1.000, Predicted=16.721
Expected=1.000, Predicted=33.714
Expected=1.000, Predicted=59.633
Expected=1.000, Predicted=179.006
Expected=1.000, Predicted=39.873
Expected=1.000, Predicted=175.154
Expected=1.000, Predicted=382.740
Expected=1.000, Predicted=19789.338
Expected=1.000, Predicted=6565434.256
Expected=0.000, Predicted=-2.051
Expected=0.000, Predicted=1.200
Expected=0.000, Predicted=1.708
Expected=0.000, Predicted=1.350
Expected=0.000, Predicted=0.727
```

Having y as the inflation_crises and x as inflation_annual_cpi we use the equation y = B0 + B1(x) Using 1 = 0.2986 + 0.2393(x). We have annual rate of inflation as 2.9310. 2.9310 is the annual rate of inflation at which inflation becomes a practical certainty.