```
# main.py
# main entrance of the program used to test all the modules
# import all the module
import dataLoader as dl
import MLETrainer as mt
import laplaceTrainer as lt
import logRegTrainer as lr

dataSet_1 = dl.dataSet()
# Load training data set from .txt
dataSet_1.loadFile("heart-train.txt")
dataSet_2 = dl.dataSet()
# Load testing data set from .txt
dataSet_2.loadFile("heart-test.txt")

trainer_MLE = mt.MLETrainer()
# Start MLE training
trainer_MLE.loadTrainDataSet(dataSet_1)
# Start MLE result verification
trainer_MLE.loadTestDataSet(dataSet_2)
# Print MLE P(Y = 1)
trainer_MLE.printPY(1)
# Print MLE P(Y = 1|X = 0)
for i in range(0, dataSet_2.inputNum):
    trainer_MLE.printPx_y(i, 1, 1)
# Print MLE P(Y = 1|X = 1)
for i in range(0, dataSet_2.inputNum):
    trainer_MLE.printPx_y(i, 1, 0)
# Print MLE Result Accuracy
trainer_MLE.printAccuracy()
# Repeat same procedure with Laplace Estimator
trainer_Laplace = lt.LaplaceTrainer()
trainer_Laplace.loadTrainDataSet(dataSet_1)
trainer_Laplace.loadTestDataSet(dataSet_2)
trainer_Laplace.printPY(1)
for i in range(0, dataSet_2.inputNum):
    trainer_Laplace.printPx_y(i, 1, 1)
for i in range(0, dataSet_2.inputNum):
    trainer_Laplace.printPx_y(i, 1, 0)
trainer_Laplace.printAccuracy()
t_diff = trainer_MLE.ty - trainer_Laplace.ty

# Print training parameters in LaTex table format
for i in range(0, dataSet_2.inputNum):
    print("%d & %.3f & %.3f & %.3f & %.3f \\\\" % (
        i, trainer_MLE.px_y[i][1][0], trainer_MLE.px_y[i][1][1], trainer_Laplace.px_y[i][
1][0],
        trainer_Laplace.px_y[i][1][1]))
# Print likelihood ratio
trainer_MLE.printR()

for i in range(0, dataSet_2.inputNum):
    print("%d & %.3f" % (i, trainer_MLE.r[i]))

trainer_logReg = lr.logRegTrainer()
# Start logistic regression training
trainer_logReg.loadTrainDataSet(dataSet_1)
# Start logistic verification
trainer_logReg.loadTestDataSet(dataSet_2)
trainer_logReg.printAccuracy()
# Print logistic verification results
trainer_logReg.printTheta()
trainer_logReg.printLL(dataSet_2)
print("Finished.")

```