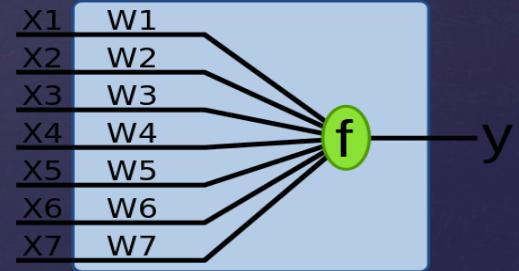
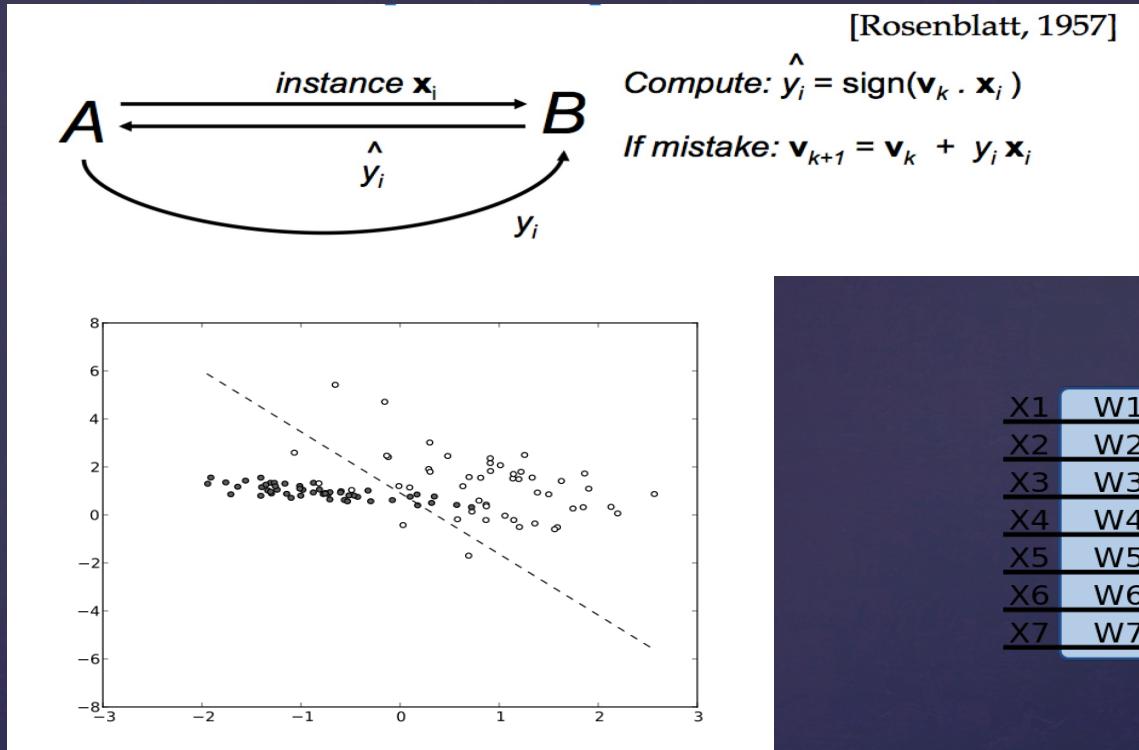


# Online Perceptron

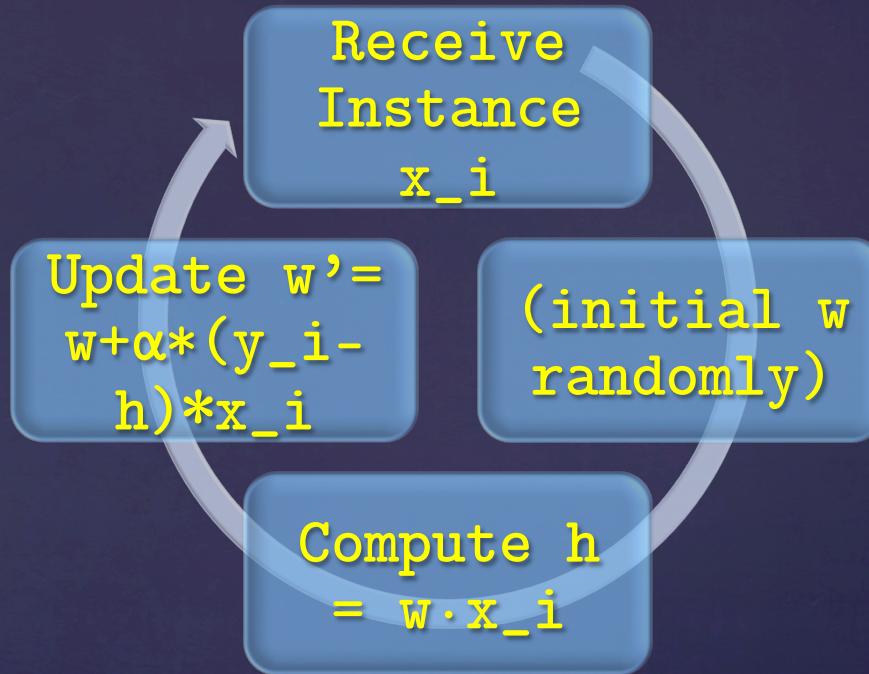
{ For the tic-tac-toe endgame

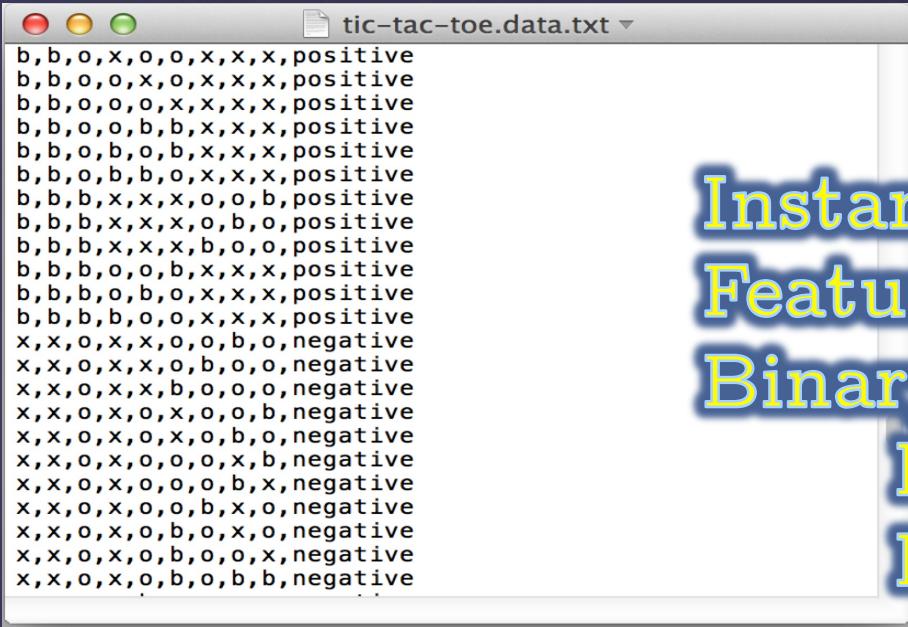
Long He



# Perceptron Algorithm

# Online Perceptron





A screenshot of a terminal window titled "tic-tac-toe.data.txt". The window displays a list of 958 data instances, each consisting of 9 features separated by commas and ending with a class label ("positive" or "negative"). The features are represented by lowercase letters 'b', 'x', and 'o'. The data is as follows:

```
b,b,o,x,o,o,x,x,x,positive  
b,b,o,o,x,o,x,x,x,positive  
b,b,o,o,o,x,x,x,x,positive  
b,b,o,o,b,b,x,x,x,positive  
b,b,o,b,o,b,x,x,x,positive  
b,b,o,b,b,o,x,x,x,positive  
b,b,b,x,x,x,o,o,b,positive  
b,b,b,x,x,x,o,b,o,positive  
b,b,b,x,x,x,b,o,o,positive  
b,b,b,o,o,b,x,x,x,positive  
b,b,b,o,b,o,x,x,x,positive  
b,b,b,b,o,o,x,x,x,positive  
x,x,o,x,x,o,o,b,o,negative  
x,x,o,x,x,o,b,o,o,negative  
x,x,o,x,x,b,o,o,o,negative  
x,x,o,x,x,o,o,b,negative  
x,x,o,x,o,x,o,b,o,negative  
x,x,o,x,o,o,o,x,b,negative  
x,x,o,x,o,o,o,b,x,negative  
x,x,o,x,o,o,b,x,o,negative  
x,x,o,x,o,b,o,x,o,negative  
x,x,o,x,o,b,o,o,x,negative  
x,x,o,x,o,b,o,b,b,negative
```

Instance Num: 958

Features Num: 9

Binary Label:

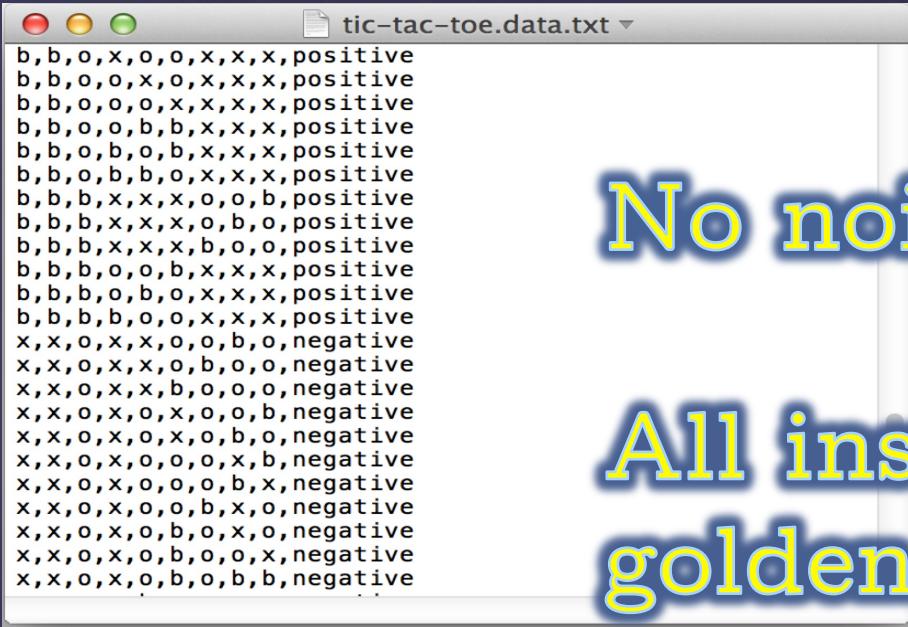
Positive(626)

Negative(332)

No duplicate

Not linearly separable

Data

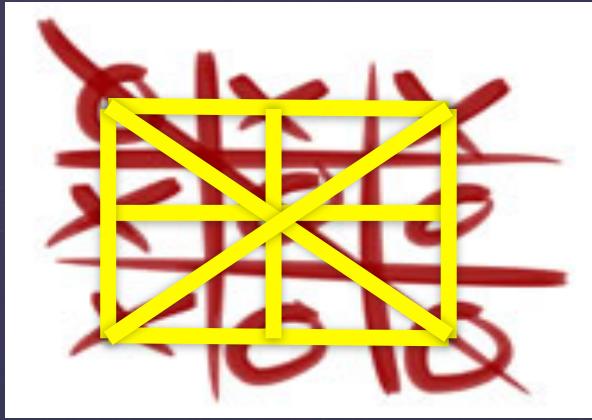


```
b,b,o,x,o,o,x,x,x,positive  
b,b,o,o,x,o,x,x,x,positive  
b,b,o,o,o,x,x,x,x,positive  
b,b,o,o,b,b,x,x,x,positive  
b,b,o,b,o,b,x,x,x,positive  
b,b,o,b,b,o,x,x,x,positive  
b,b,b,x,x,x,o,o,b,positive  
b,b,b,x,x,x,o,b,o,positive  
b,b,b,x,x,x,b,o,o,positive  
b,b,b,o,b,x,x,x,positive  
b,b,b,o,b,o,x,x,x,positive  
b,b,b,b,o,o,x,x,x,positive  
x,x,o,x,x,o,o,b,o,negative  
x,x,o,x,x,o,b,o,o,negative  
x,x,o,x,x,b,o,o,o,negative  
x,x,o,x,o,x,o,o,b,negative  
x,x,o,x,o,x,o,b,o,negative  
x,x,o,x,o,o,o,x,b,negative  
x,x,o,x,o,o,o,b,x,negative  
x,x,o,x,o,o,b,x,o,negative  
x,x,o,x,o,b,o,x,o,negative  
x,x,o,x,o,b,o,o,x,negative  
x,x,o,x,o,b,o,b,b,negative
```

No noise!

All instances are  
golden standards!

# What's More



One of 8  
lines has 3  
“X”

None of 8  
lines has 3  
“O”

Total  
number of  
“O” > “X”

Positive

# Data Analysis

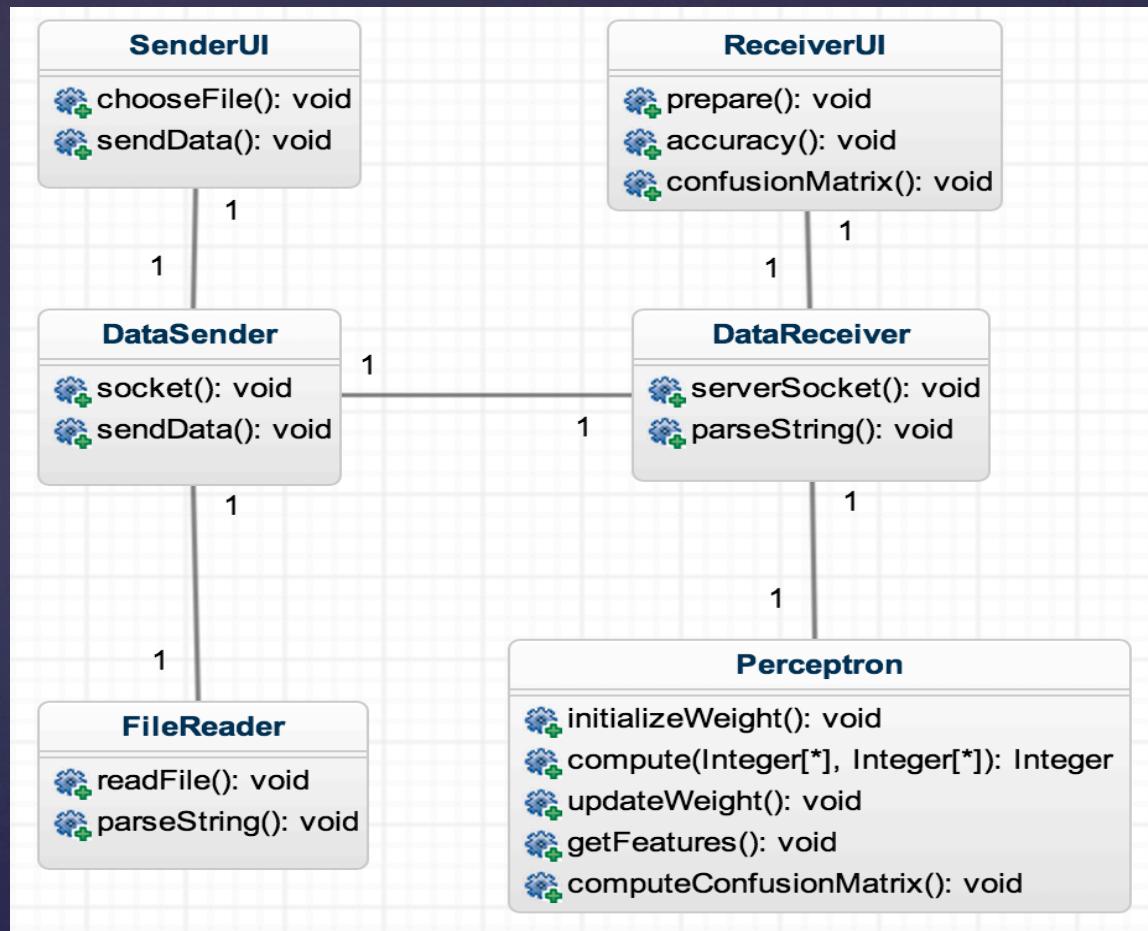
A: One of 8 lines has 3 “X”  
B: None of 8 lines has 3 “O”  
C: Total number of “O” > “X”  
If (A || (B && C)), Y == Positive.

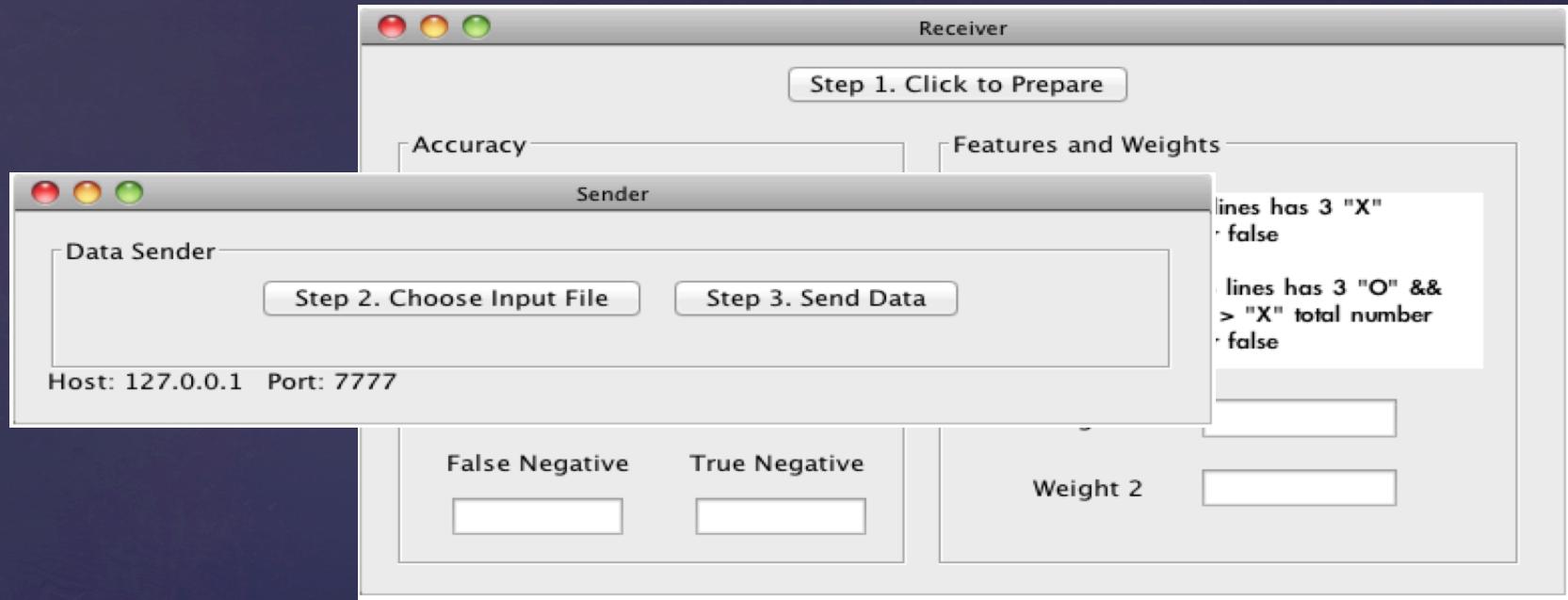
I chose A and (B && C) as two features:

F1 = A,  
F\_2 = (B && C),  
1 for true, -1 for false.

# Features Selection

# UML





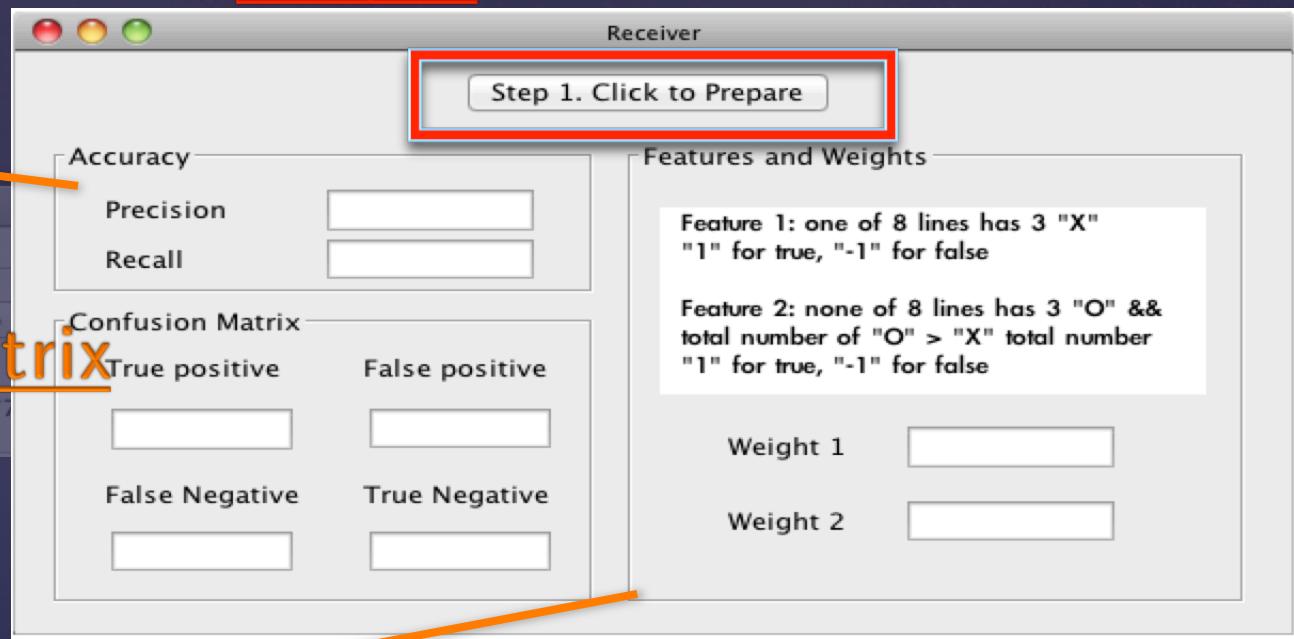
# User Interface

# Step 1

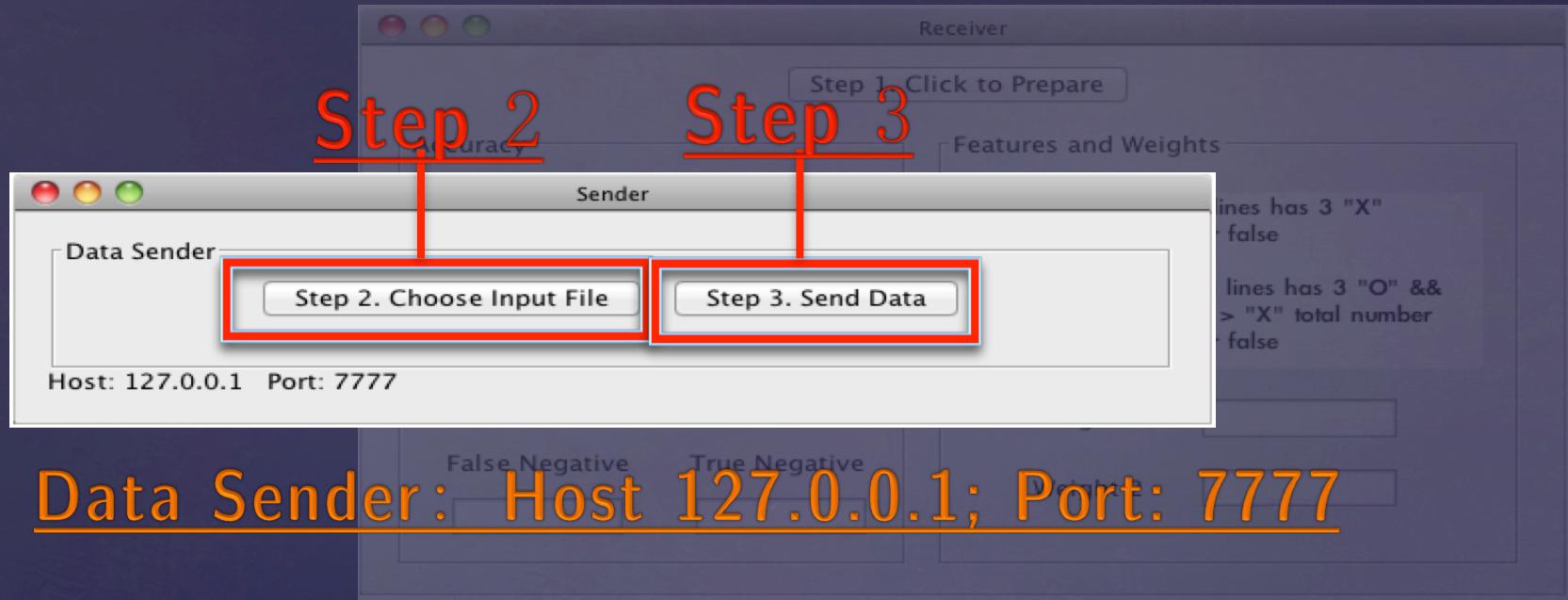
Accuracy

ConfusionMatrix

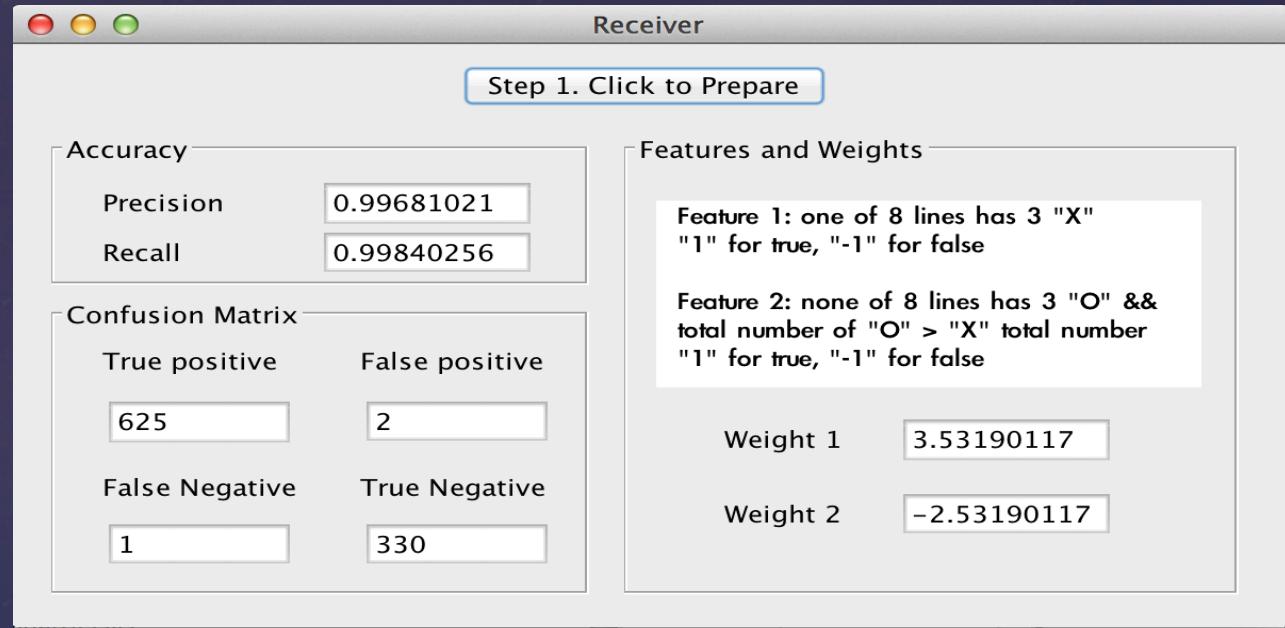
Features, Weights



# Receiver Interface



# Sender Interface



# Result

Precision: 0.997  
Recall: 0.998

```
Console X
ReceiverUI [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_20.jdk/Contents/Home/bin/ja
-0.4544764984146683 -1.1702505281819984
-0.31236351329833334 -1.241307020740166
-0.27683526701924965 -1.2590711438797078
-0.2679532054494787 -1.2635121746645932
-0.265732690057036 -1.2646224323608146
-0.26517756120892527 -1.26489999678487
-0.2650387789968976 -1.2649693878908836
-0.26500408344389065 -1.2649867356673872
-0.2649954095556389 -1.264991072611513
-0.26499324108357597 -1.2649921568475446
-0.26499269896556027 -1.2649924279065523
-0.2649925634360563 -1.2649924956713043
-0.2649925295536803 -1.2649925126124923
-0.2649925210830863 -1.2649925168477894
-0.2649925189654378 -1.2649925179066135
-0.2649925184360256 -1.2649925181713195
-0.2649925183036726 -1.264992518237496
-0.2649925182705843 -1.26499251825404
-0.2649925182623121 -1.264992518258176
-0.2649925182602441 -1.26499251825921
-0.2649925182597271 -1.2649925182594686
-0.26499251825959785 -1.2649925182595332
-0.26499251825956555 -1.2649925182595494
-0.26499251825955744 -1.2649925182595534
-0.2649925182595544 -1.2649925182595543
-0.2649925182595549 -1.2649925182595547
-0.2649925182595548 -1.2649925182595547
-0.2649925182595548 -1.2649925182595547
-0.2649925182595548 -1.2649925182595547
-0.2649925182595548 -1.2649925182595547
-0.2649925182595548 -1.2649925182595547
-0.2649925182595548 -1.2649925182595547
```

Weight( $w_1, w_2$ )  
converge  
at the very begining

Look into the Process

```
ReceiverUI [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_20.jdk/Contents/Home/bin/java
-0.1719469289516855
0.7070132677620782
0.9267533169405198
0.9816883292351299
0.9954220823087825
0.9988555205771963
0.9997138801442995
0.9999284700360747
0.9999821175090187
0.9999955293772542
0.9999988823443138
0.9999997205860787
0.9999999301465197
0.9999999825366306
0.999999956341581
0.9999999989085397
0.9999999997271356
0.999999999317835
0.999999999829461
0.999999999957367
0.999999999989342
0.999999999997335
0.999999999999334
0.99999999999984
0.999999999999956
0.999999999999991
1.0
1.0
1.0
1.0
1.0
1.0
```

$h = w \cdot x_i$  converges  
at the very beginning

Don't need decay  
function, optimized  $\alpha$   
and more training data.

# Look into the Process

Thank you.

Long He  
Nov. 10, 2014