Task I:

Terms used: True positive rate: the rate of correct(true) prediction for correct answer

True Negative rate: the rate of correct(false) prediction for false answer.

False Positive rate: the rate of false answers predicted as correct

False Negative rate: the rate of correct answers predicted as wrong

Precision: Fraction of output items that are correct versus incorrect, using GT

Recall: Fraction output items correct in GT

F-1 Measure: It is the harmonic mean of precision and recall. Used to tune and maximize both by adjusting certain thresholds.

I formatted the input into a list as: list1=

[1, 20, 80, 2, 98]
[5, 50, 50, 5, 95]
[10, 60, 40, 10, 90]
[15, 80, 20, 20, 80]
[20, 88, 12, 30, 70]
[25, 90, 10, 40, 60]
[30, 95, 5, 50, 50]
[35, 96, 4, 60, 40]
[40, 97, 3, 70, 30]
[50, 98, 2, 80, 20]

Python code to print all precisions for the respective thresholds

```
for var in list1:
    print("For threshold",var[0],"precision=",round(var[1]/(var[1]+var[3]),4),"
          Recall=",round((var[1]/(var[1]+var[2])),4))
    pr=var[1]/(var[1]+var[3])
    re=var[1]/(var[1]+var[2])
    f1=2*pr*re/(pr+re)
    #print (f1,"   ",var[0])
    if(f1>temp):
        temp=f1
        bestthreshold=var[0]
 print ("best f1 measure value is =",temp," threshold for this f1 is=",bestthreshold)
```

Output :

For threshold 1 precision= 0.9091 Recall= 0.2
For threshold 5 precision= 0.9091 Recall= 0.5
For threshold 10 precision= 0.8571 Recall= 0.6
For threshold 15 precision= 0.8 Recall= 0.8
For threshold 20 precision= 0.7458 Recall= 0.88
For threshold 25 precision= 0.6923 Recall= 0.9
For threshold 30 precision= 0.6552 Recall= 0.95
For threshold 35 precision= 0.6154 Recall= 0.96
For threshold 40 precision= 0.5808 Recall= 0.97
For threshold 50 precision= 0.5506 Recall= 0.98

best f1 measure value is = 0.8073394495412843 threshold for this f1 is= 20

Hence similarity threshold should be set as 20

Task II

ROC curve

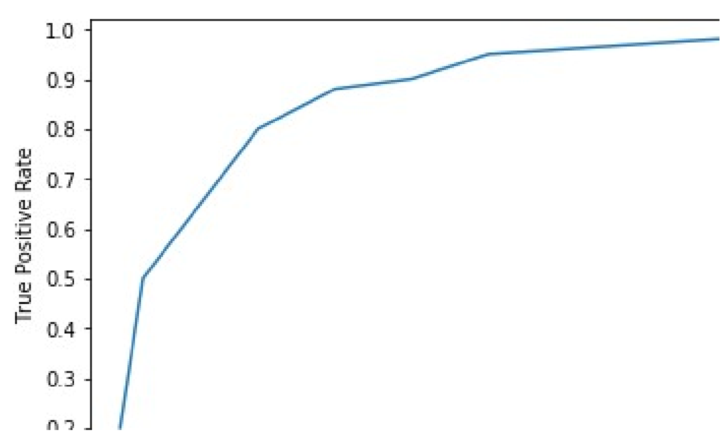
It is used to determine how increasing true positive rate affects false positive rate. A straight 45 degree line indicates random guesses. Any curve leaning towards the top left is better. The more it leans the more better the model is.

Code used for the plots:

```
for var in list1:
    print (var)
    print("For threshold",var[0],"precision=",round(var[1]/(var[1]+var[3]),4),"
Recall=",round((var[1]/(var[1]+var[2])),4))
    pr=var[1]/(var[1]+var[3])
    re=var[1]/(var[1]+var[2])
    Xaxis.append(var[3]/(var[3]+var[4]))
    misr=((var[3]/(var[3]+var[4])))
    fpr=(var[2]/(var[1]+var[2]))
    if (misr<0.5 and misr>0.1 and fpr>0.05 and fpr <0.40):
        Xaxis2.append(((var[3]/(var[3]+var[4]))))
        Yaxis2.append((var[2]/(var[1]+var[2])))
   Yaxis.append(re)
   f1=2*pr*re/(pr+re)
   if(f1>temp):
       temp=f1
       bestthreshold=var[0]
  print(Xaxis2)
  print(Yaxis2)

plt.figure(1)       #Roc curve
plt.plot(Xaxis,Yaxis)
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
```

ROC curve:



As we can see the curve for the given model is good due to it leaning towards the top left. Ideally in the model if the false positive rate is close to zero the true positive rate should be close to 1.

TaskIII
DET curve
ROC curve wastes a lot of space because we are only interested in optimizing the models near the top left corner(where false p rate is low and true pos rate is high). DET or Detection error tradeoff error curve is used. In this technique the ROC curve is flipped along the x-axis, log scale is chosen and we zoom in near the curve’s key parts.
For DET curve a better model leans to the bottom left.

DET curve code:

```
plt.figure(2)
plt.xlabel("False Positive Rate")
plt.ylabel("Missed Detection Rate")
plt.yscale('log')
plt.xscale('log')
plt.plot(Xaxis2,Yaxis2)
```

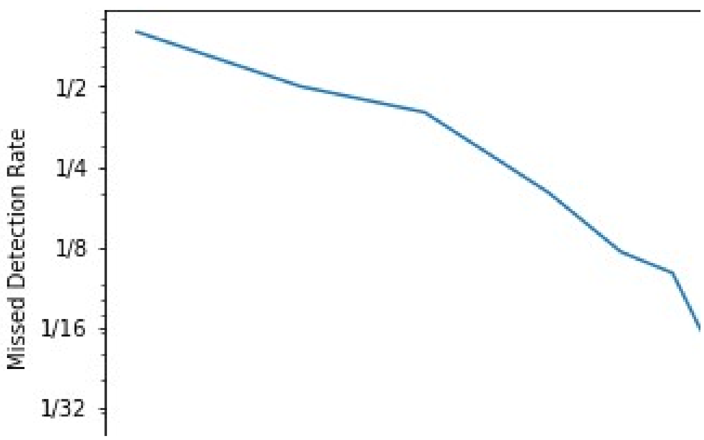


Figure 1: shows the DET curve for all thresholds.
DET curve(without zooming):

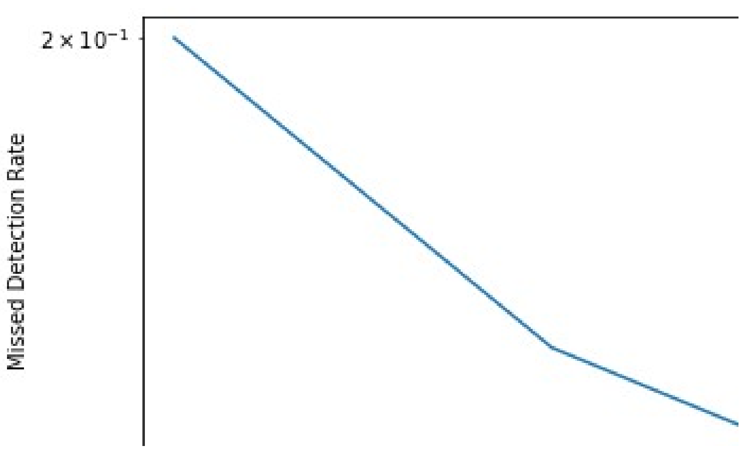


Figure 2
Choosing values near the best f1 measure at similarity threshold 20 and zooming into the DET curve.

As we can see in figure 2 the curve slightly tilts to the bottom left corner indicating good performance of the model. This is because false positive rate and missed detection rate are lower at bottom left.