Practical 8

1.) A table of the threshold, precision, and recall and f1 score can be seen below. Precision is the measure of how many of the true positives found by the predictive model are relevant. This is measured by TP/TP+FP (true positives over all positives). Recall is how many of the total relevant items are selected. It is represented by TP/TP+FN (true positives over all correct or relevant answers). The f1 measure is a combination of the two. It is also called the harmonic mean. It is found with the formula 2 * (precision*recall)/(precision + recall). It is an F1 measure because precision and recall are both equally weighted. Should this change, It would be referred to as an F2 or and F.5 score.

Threshold	Precision	Recall	F1 Measure
1	0.9091	.2	0.32786885245901637
5	0.9090909090909091	.5	0.6451612903225806
10	0.8571428571428571	.6	0.7058823529411765
15	0.8	.8	0.8
20	0.7457627118644068	.88	0.8073394495412844
25	0.6923076923076923	.9	0.782608695652174
30	0.6551724137931034	.95	0.7755102040816326
35	0.6153846153846154	.96	0.75
40	0.5808383233532934	.97	0.7265917602996255
50	0.550561797752809	.98	0.7050359712230215

The best threshold in terms of the best f1 score is threshold 20 with an f1 score of .807, followed closely by the second-best threshold 15 with an f1 score of .8.

2.) The ROC (receiver operative characteristic) curve of the data is a measure of the credibility and hence usefulness of a test. It is the plotting of the data of a binary classification model where the x value is the false positive rate or the 1 minus the specificity (true negative rate – measuring the proportion of correctly classified negatives and measuring the avoidance of false positives) and the y is the sensitivity (true positive rate or recall – quantifying the avoidance of false negatives). Because both the avoidance of false positives and false negatives are being quantified, the tests with values above 90% sensitivity (recall) and specificity about 90% are said to be credible. In terms of the graph, the greater the area under the curve, the more useful or credible a classifier.

Threshold	Sensitivity (tp/tp+fn)	Specificity = tn/tn+fp	1 – specificity (FP rate)
1	.2	.98	.02
5	.5	.95	.05

10	.6	.9	.1
15	.8	.8	.2
20	.88	.7	.3
25	.9	.6	.4
30	.95	.5	.5
35	.96	.4	.6
40	.97	.3	.7
50	.98	.2	.8

Plotting these points yields the below ROC plot. It may be noticed that the ROC doesn't continues to the ends of the plot. This could be prevented by clipping the x and y axes min and max. However, I wanted to show the entirety of the plot to emphasize how it exactly maps to the provided above values.

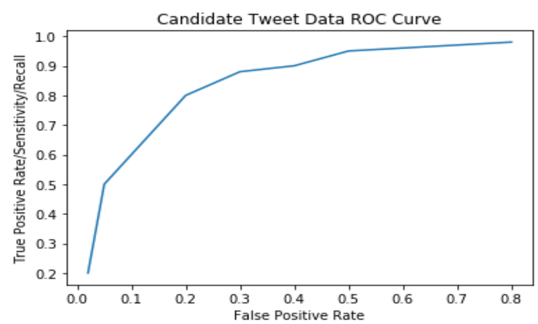
```
sensitivity = []
count = 0
for i in TPos:
    fn = FNeg[count]
    result = i/(i+fn)
    sensitivity.append(result)
    count +=1
print(sensitivity)
```

```
[0.2, 0.5, 0.6, 0.8, 0.88, 0.9, 0.95, 0.96, 0.97, 0.98]
```

```
TN = [98,95,90,80,70,60,50,40,30,20]
specificity = []
count = 0
for i in TN:
    fp = FPos[count]
    result = i/(i+fp)
    res = 1-result
    specificity.append(res)
    count +=1
print(specificity)
```

```
[0.020000000000000018, 0.050000000000000044, 0.0999999999999999
8, 0.19999999999999996, 0.30000000000000004, 0.4, 0.5, 0.6, 0.7,
0.8]
```

```
import numpy as np
import matplotlib.pyplot as plt
x = specificity
y = sensitivity
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate/Sensitivity/Recall')
plt.title('Candidate Tweet Data ROC Curve')
plt.plot(x,y)
plt.show()
auc = np.trapz(y,x)
```



3.) A DET (detection error tradeoff) graph is a graph of data from a binary classification model wherein the x axis is the false positive rate and the y is the false negative rate. False positive rate is calculated as FP/FP+TN. The false negative rate is = FN / (TP + FN) (MedScape, 2013).

Threshold	False Positive Rate	False Negative Rate
1	.02	.08
5	0.05	0.7142857142857143
10	.1	0.667
15	.2	.5
20	.3	.375
25	.4	.334
30	.5	.2
35	.6	.1667
40	.7	0.13043478260869565

50	.8	0.09090909090909091

Using the below code, my own knowledge of matplotlib and log graphs, and the skeleton of a previously written function (Karnowsky, 2015), I plotted the DET curve. Once again, the provided points did not reach zero, nor do any points reach 1. These numbers constitute the beginning and the end of the axes, and thus the line does not extend to either end. I attempted to ameliorate this through clamping the axes, but beginning an axis not at zero when dealing with such small values is misleading. Alternatively, I could try to find a way to extend the line to either end, but there will be no situation where an error rate would be 0%. However, if the classifier used were known, there could be a way of predicting values that might yield results closer to 0 and 1.

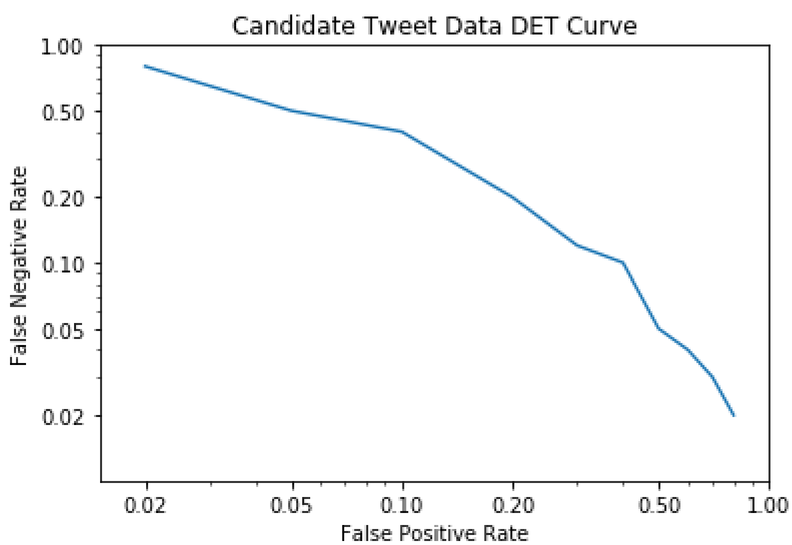
```
#find false pos
fp_rate = []
count = 0
for i in FPos:
    res = i/(i+TN[count])
    fp_rate.append(res)
    count+=1
print (fp_rate)
```

```
[0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8]
```

```
#find false neg
fn_rate = []
count = 0
for i in FNeg:
    res = i/(TPos[count]+i)
    fn_rate.append(res)
    count+=1
print (fn_rate)
```

```
[0.8, 0.5, 0.4, 0.2, 0.12, 0.1, 0.05, 0.04, 0.03, 0.02]
```

```
#plot DET for data
#x=fp rate
#y= fn rate
#log scale
#based on Jeremy karnowski's DETCurve function
#fps and fns, ticks, axis, use of pyplot modules determined by me
def DETCurve(fps,fns):
    axis_min = min(fps[0],fns[-1])
    fig,ax = plt.subplots()
    plt.plot(fps,fns)
    plt.yscale('log')
    plt.xscale('log')
    ticks_to_use = [0.02,0.05,0.1,0.2,0.5,1]
    ax.get_xaxis().set_major_formatter(matplotlib.ticker.ScalarFormatter())
    ax.get_yaxis().set_major_formatter(matplotlib.ticker.ScalarFormatter())
    ax.set_xticks(ticks_to_use)
    ax.set_yticks(ticks_to_use)
    plt.axis([0.015,1,0.01,1])
    plt.xlabel('False Positive Rate')
    plt.ylabel('False Negative Rate')
    plt.title('Candidate Tweet Data DET Curve')
    plt.show()
DETCurve(fp_rate,fn_rate)
```



Works Cited

Karnowsky, J. (2015, August 07). Detection Error Tradeoff (DET) curves. Retrieved November 10, 2017, from https://jeremykarnowski.wordpress.com/2015/08/07/detection-error-tradeoff-det-curves/

MedScape. (2013). False Negative Rate from Sensitivity and Prevalence. Retrieved November 09, 2017, from https://reference.medscape.com/calculator/false-negative-rate