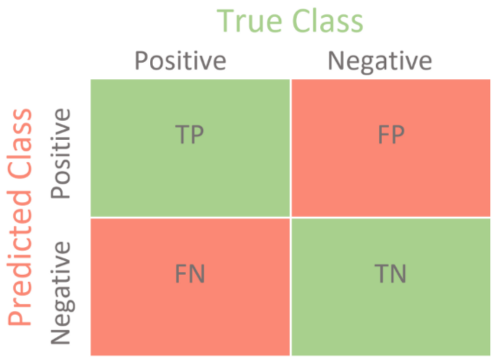
Jason Ballantyne
13432788

Text Analytics COMP47600
Practical Number 8

08/11/2021

Q1:

The data provided contains True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN). These values represent the confusion matrix. The confusion matrix is a performance measurement for machine learning classification problems where the output can be two or more classes. It is extremely useful for measuring Recall, Precision and F1 Measures which we will go on to discuss in detail. Please see an example of a confusion matrix below.



Precision and recall are two crucial topics in machine learning. They are two numbers which together are used to evaluate the performance of classification. Precision is defined as the fraction of relevant instances among all retrieved instances. It is the ratio between the True Positives and all the Positives. The formula for Precision can be seen below:

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Recall is sometimes referred to as 'sensitivity,' it is the fraction of retrieved instances among all relevant instances. Essentially, Recall is the ability of a model to find all the relevant cases within a data set and is the measure of our model correctly identifying True Positives. The formula for Recall can be seen below:

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Precision is how sure you are of your true positives whilst recall is how sure you are that you are not missing any positives. A perfect classifier has precision and recall both equal to 1.

We will now take a look at the F1-score. The F1-score takes both precision and recall into account to ultimately measure the accuracy of the model. It tries to take this into account, giving more weight to false negatives and false positives while not letting large numbers of true negatives influence the score. The F1-score performs best if there is some sort of balance between precision and recall. Essentially, we can say that the F1-score punishes extreme values. The formula for the F1-score can be seen below:

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Please see the snippet of the dataframe below containing the Precision, Recall and F1 measure for each threshold value.

	Threshold	TP	FN	FP	TN	Correct	Incorrect	Test set	Precision	Recall	F1 measure
0	1	20	80	2	98	100	100	200	0.909091	0.20	0.327869
1	5	50	50	5	95	100	100	200	0.909091	0.50	0.645161
2	10	60	40	10	90	100	100	200	0.857143	0.60	0.705882
3	15	80	20	20	80	100	100	200	0.800000	0.80	0.800000
4	20	88	12	30	70	100	100	200	0.745763	0.88	0.807339
5	25	90	10	40	60	100	100	200	0.692308	0.90	0.782609
6	30	95	5	50	50	100	100	200	0.655172	0.95	0.775510
7	35	96	4	60	40	100	100	200	0.615385	0.96	0.750000
8	40	97	3	70	30	100	100	200	0.580838	0.97	0.726592
9	50	98	2	80	20	100	100	200	0.550562	0.98	0.705036

Based on the results above, we can see that the **threshold value of 20** does the best as it gives the F1 measure of 0.8073, the highest among all the threshold values. An F1 score reaches its best value at 1 and worst value at 0. A low F1 score is an indication of both poor precision and poor recall. It is clear that the higher the F1 score the better.

Some other well performing threshold values in order include threshold 15 with an F1 measure of 0.8 (the second best). Threshold value 25 performed third best with an F1 measure of 0.7826. The worst performing F1 measure was threshold 1, with an F1 measure of 0.3279 this was due to it's poor recall score of 0.20.

Q2:

An ROC curve or receiver operating characteristic curve is a graph showing the performance of a classification model at all classification thresholds. It is a probability curve and essentially separates the 'signal' from the 'noise'. The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.

This curve plots two parameters:

- True Positive Rate
- False Positive Rate

The True Positive Rate is calculated as follows:

$$TPR = \frac{TP}{TP + FN}$$

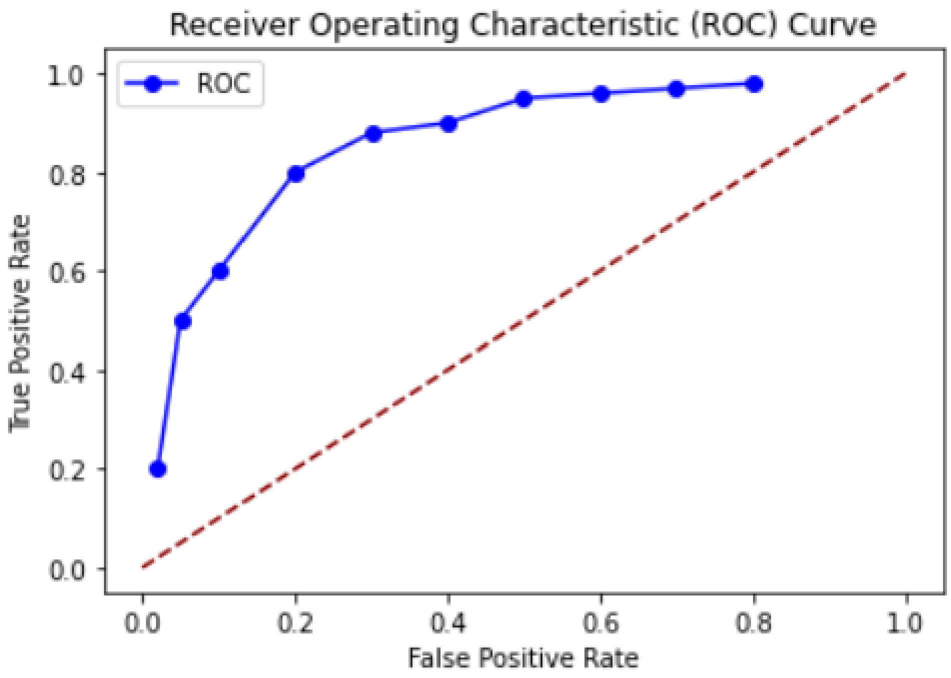
The False Positive Rate is calculated as follows:

$$FPR = \frac{FP}{FP + TN}$$

Please see the snippet of my dataframe below showing the True Positive Rate and the False Positive Rate for each threshold value.

	Threshold	TPR	FPR
0	1	0.20	0.02
1	5	0.50	0.05
2	10	0.60	0.10
3	15	0.80	0.20
4	20	0.88	0.30
5	25	0.90	0.40
6	30	0.95	0.50
7	35	0.96	0.60
8	40	0.97	0.70
9	50	0.98	0.80

Please see the plotted ROC curve below based on the above data.



The ROC curve shows the trade-off between sensitivity and specificity. Based on our above results, we can see that the curve is closer to the top left corner. This indicates a better performance. As a baseline, a random classifier is expected to give points lying along the diagonal (FPR = TPR). The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test. We can see that our curve is not close to the 45-degree diagonal indicating an accurate test. We can say from this curve that the system used to classify tweets about different candidates was accurate.

Q3:

The DET curve or Detection Error Trade-off curve is a graphical plot of error rates for binary classification systems, plotting the false rejection rate vs. false acceptance rate. It produces an output containing the false positive rate (FPR) and false negative rate (FNR) for all threshold values. DET curves have the property that if the underlying score distributions for the two types of trials are normal, the curve becomes a straight line. They have been widely used to present the performance characteristics of speaker recognition systems. Since DET curves plot error rates, they give uniform treatment to both types of errors, and they use scaling for both the axes which spreads out the plot and produces almost linear plots.

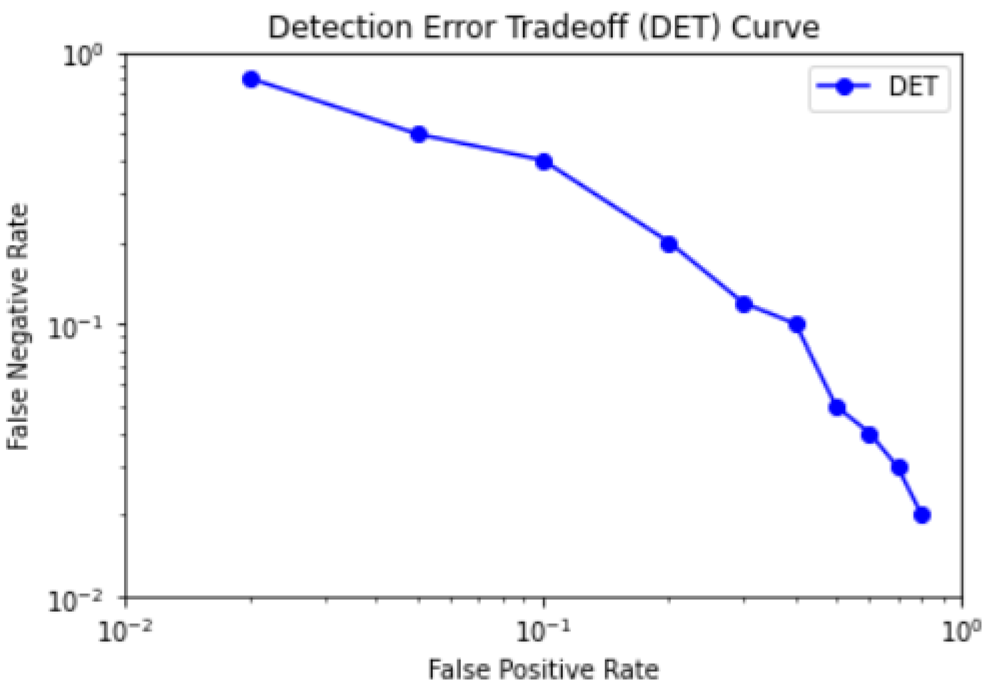
As stated, it uses the false positive rate (FPR) and false negative rate (FNR), we have already defined FPR in our previous question, however we need to calculate FNR, which is calculated using the below formula:

$$FNR = \frac{FN}{Actual\ Positive} = \frac{FN}{TP + FN}$$

Please see a snippet of our dataframe below showing the False Positive Rate and the False Negative Rate for each threshold value.

	Threshold	FPR	FNR
0	1	0.02	0.80
1	5	0.05	0.50
2	10	0.10	0.40
3	15	0.20	0.20
4	20	0.30	0.12
5	25	0.40	0.10
6	30	0.50	0.05
7	35	0.60	0.04
8	40	0.70	0.03
9	50	0.80	0.02

Please see the plotted DET curve below based on the above data.



DET curves give us direct feedback of the detection error trade-off to aid in operating point analysis. We can deduct directly from the DET-curve plot at which rate false-negative error rate will improve when willing to accept an increase in false-positive error rate. If the resulting curves are straight lines, then this provides a visual confirmation that the underlying likelihood distributions from the system are normal.

From our graph however, the curve moves towards the bottom left of our plot, this shows us that the DET curve is getting better at the task it is doing (i.e., It's getting closer to perfect classification). If we had a straighter line, the likelihoods we would be getting are more normal.