David Shvimer
Z23430635

Logistic Regression

Introduction

In this experiment a Logistic Regression model was used to evaluate the effectiveness of the General Classification Rule. The LR model will output the probability that some input belongs to one of two classes. This is typical in binary classification. A classification rule is a way to intelligently determine a threshold for the probability and assign it to one of two classes. It is possible to select a threshold for the probability naively but with little effort a more robust method is possible. The dataset used described software modules and the LR model was trained to predict whether the module was fault prone or not.

Method

1. The dataset contains the number of faults for various software modules. Convert this field to a binary class value of either fault prone or not fault prone, where 2 faults or more is considered fault prone.
2. Use Weka to train a Logistic Regression model to predict the probability that a module is fault prone.
3. Apply the model to both fit and test sets.
4. Apply the General Classification rule with various values of ‘c’ ranging from 0.1 to 50. This assigns each model a predicted class.
5. Using the predicted and know class, calculate a confusion matrix for each ‘c’ and evaluate the false negative rate (FNR) and false positive rates (FPR).
6. Find the strongest selection for ‘c’ by finding where the FNR and FPR are balanced, ideally minimizing FNR.

Logistic Regression Model

```
Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                  Class
Variable             fp
=====================
NUMUORS          0.1126
NUMUANDS         0.0244
TOTOTORS         0.0018
TOTOPANDS       -0.0109
VG              -0.1022
NLOGIC           0.1269
LOC              0.0013
ELOC             0.0591
Intercept        -6.773
```

General Classification Rule

$$Class(\mathbf{x}_i) = \begin{cases} G_1 & \text{if } \frac{f_1(\mathbf{x}_i)}{f_2(\mathbf{x}_i)} \geq c \\ G_2 & \text{otherwise} \end{cases}$$

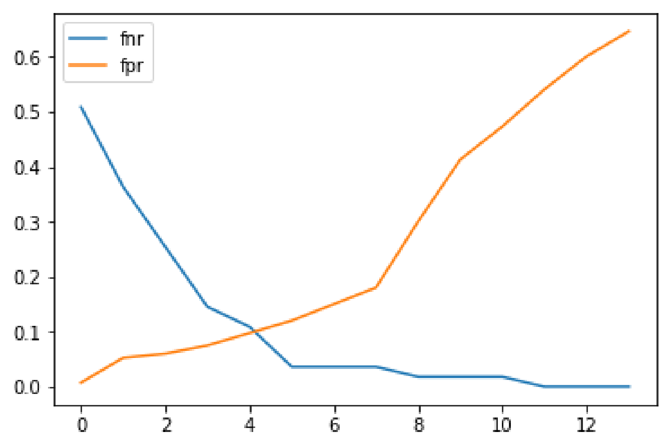
Results

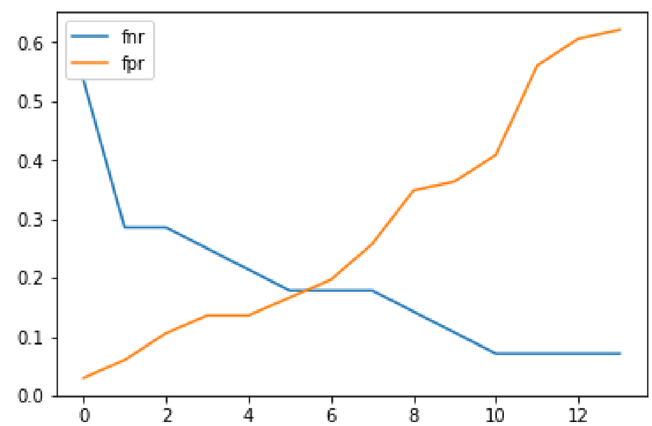
Fit

	accuracy	c	error_rate	fn	fnr	fp	fpr	tn	tnr	tp	tpr
0	0.845745	0.1	0.154255	28	0.509091	1	0.007519	132	0.992481	27	0.490909
1	0.856383	0.5	0.143617	20	0.363636	7	0.052632	126	0.947368	35	0.636364
2	0.882979	1.0	0.117021	14	0.254545	8	0.060150	125	0.939850	41	0.745455
3	0.904255	2.0	0.095745	8	0.145455	10	0.075188	123	0.924812	47	0.854545
4	0.898936	3.0	0.101064	6	0.109091	13	0.097744	120	0.902256	49	0.890909
5	0.904255	4.0	0.095745	2	0.036364	16	0.120301	117	0.879699	53	0.963636
6	0.882979	5.0	0.117021	2	0.036364	20	0.150376	113	0.849624	53	0.963636
7	0.861702	6.0	0.138298	2	0.036364	24	0.180451	109	0.819549	53	0.963636
8	0.781915	10.0	0.218085	1	0.018182	40	0.300752	93	0.699248	54	0.981818
9	0.702128	15.0	0.297872	1	0.018182	55	0.413534	78	0.586466	54	0.981818
10	0.659574	20.0	0.340426	1	0.018182	63	0.473684	70	0.526316	54	0.981818
11	0.617021	30.0	0.382979	0	0.000000	72	0.541353	61	0.458647	55	1.000000
12	0.574468	40.0	0.425532	0	0.000000	80	0.601504	53	0.398496	55	1.000000
13	0.542553	50.0	0.457447	0	0.000000	86	0.646617	47	0.353383	55	1.000000

Test

	accuracy	c	error_rate	fn	fnr	fp	fpr	tn	tnr	tp	tpr
0	0.819149	0.1	0.180851	15	0.535714	2	0.030303	64	0.969697	13	0.464286
1	0.872340	0.5	0.127660	8	0.285714	4	0.060606	62	0.939394	20	0.714286
2	0.840426	1.0	0.159574	8	0.285714	7	0.106061	59	0.893939	20	0.714286
3	0.829787	2.0	0.170213	7	0.250000	9	0.136364	57	0.863636	21	0.750000
4	0.840426	3.0	0.159574	6	0.214286	9	0.136364	57	0.863636	22	0.785714
5	0.829787	4.0	0.170213	5	0.178571	11	0.166667	55	0.833333	23	0.821429
6	0.808511	5.0	0.191489	5	0.178571	13	0.196970	53	0.803030	23	0.821429
7	0.765957	6.0	0.234043	5	0.178571	17	0.257576	49	0.742424	23	0.821429
8	0.712766	10.0	0.287234	4	0.142857	23	0.348485	43	0.651515	24	0.857143
9	0.712766	15.0	0.287234	3	0.107143	24	0.363636	42	0.636364	25	0.892857
10	0.691489	20.0	0.308511	2	0.071429	27	0.409091	39	0.590909	26	0.928571
11	0.585106	30.0	0.414894	2	0.071429	37	0.560606	29	0.439394	26	0.928571
12	0.553191	40.0	0.446809	2	0.071429	40	0.606061	26	0.393939	26	0.928571
13	0.542553	50.0	0.457447	2	0.071429	41	0.621212	25	0.378788	26	0.928571





Conclusion

In the results for the fit dataset we can visually see that the FNR and FPR are balanced from indices 3 to 7. These indices correspond to ‘c’ values from 2 to 6. In the test set we see a balance from indices 4 to 7. The optimal value for ‘c’ from the fit set would be 4 since this is where the rates are balanced and FNR is minimized. For the test set the same value of 4 seems to be ideal. Therefore the recommended value for ‘c’ when using the general classification rule is 4.