Sick dataset analysis

Bogdan Jastrzębski

26 kwietnia 2020 r.

Contents


```
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
```

1 Comparison between prediction accuracy of interpretable and non-interpretable models

I will compare five different interpretable models:

- naive biases
- logistic regression
- basic tree
- knn
- ctree

and back-box models:

- C50
- adabag boosting
- mlp
- ada boost
- gradient boosting machine
- glm boost
- random forest
- cforest

1.1 Interpretable models

In this section we will examine prediction accuracy (via measuring AUC and AUPRC) of interpretable models. The following is not only a summary of the previous work, but an extension. Namely, due to imbalance of our classes, we performed oversampling, which generally has a capacity of increasing performance.

1.1.1 Naive Bayes

iter	auc	auprc
1	0.8304931	0.1101079
2	0.7882103	0.0986908
3	0.8672120	0.1462317
4	0.8724165	0.0929076
5	0.8282906	0.1180700

With oversampling:

iter	auc	auprc
1	0.8362270	0.4155058
2	0.8449867	0.3461226
3	0.8603510	0.3287230
4	0.8336865	0.3797344
5	0.8412488	0.3698691

Here we can see, that oversampling incresed AUPRC significantly. We must remember, that it doesn't mean that the model is that much better.

1.1.2 Logistic Regression

iter	auc	auprc
1	0.9286859	0.7556515
2	0.9254187	0.6217183
3	0.9377056	0.6528932
4	0.9706809	0.7178206
5	0.8901906	0.6521577

With oversampling:

iter	auc	auprc
1	0.9720501	0.9108548
2	0.9529617	0.8594950
3	0.9588918	0.9208025
4	0.9719616	0.9231461
5	0.9542819	0.8998895

Again, oversampling gave better results.

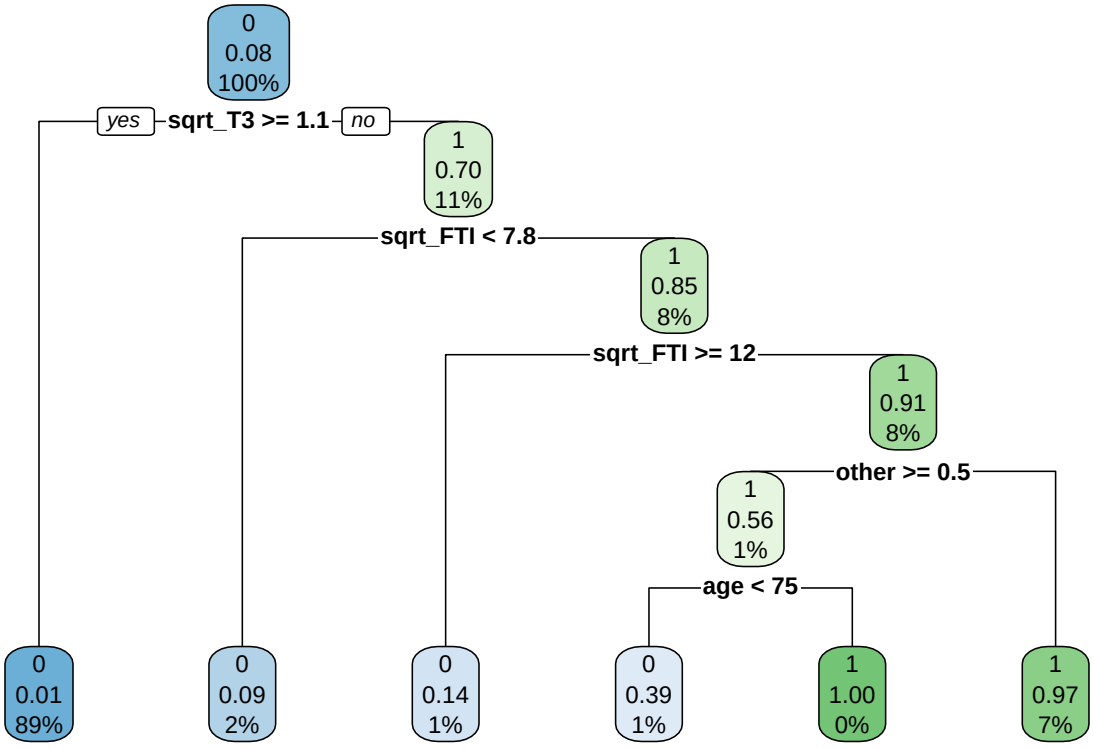
1.1.3 Tree

iter	auc	auprc
1	0.9967262	0.9179463
2	0.9599987	0.9194374
3	0.8867839	0.6035345
4	0.9648651	0.8280466
5	0.9814282	0.8762334

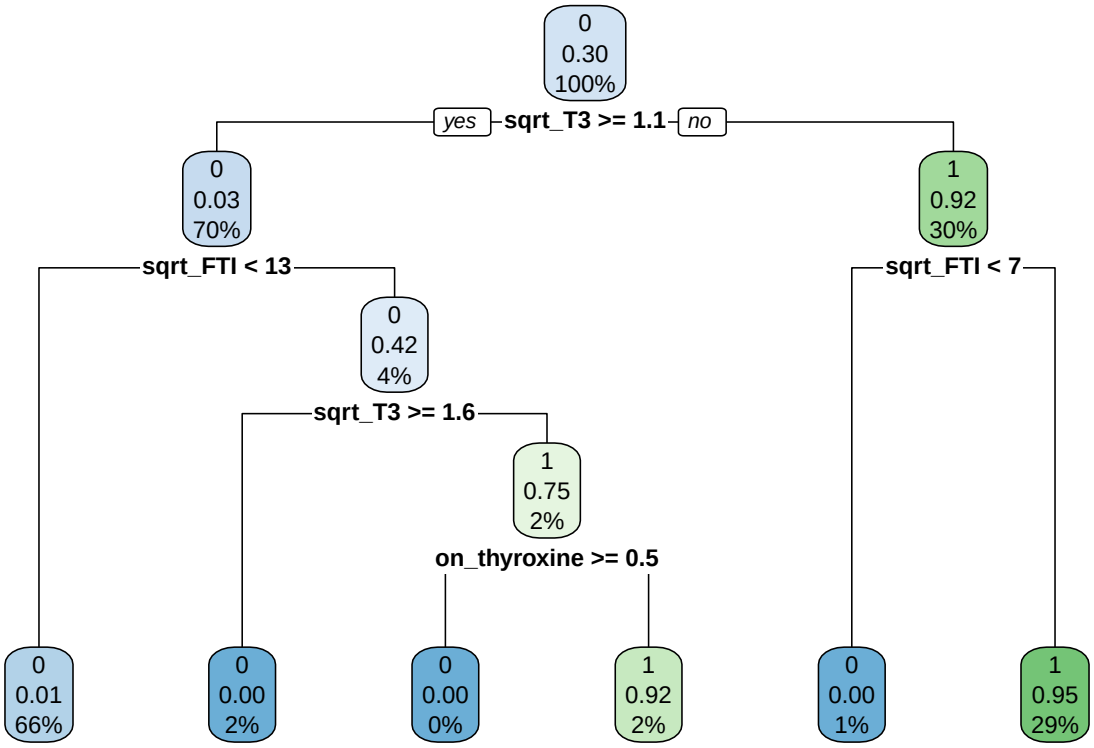
With oversampling:

iter	auc	auprc
1	0.9873023	0.9497708
2	0.9710237	0.9244497
3	0.9887780	0.9602050
4	0.9778690	0.9176867
5	0.9696256	0.9141573

Does the model changed? Let's see, this is a model without oversampling:



and with oversampling:



Yes, it has changed. Oversampling then does change the model structure.

1.1.4 The KNN

iter	auc	auprc
1	0.9881672	0.8619730
2	0.9756292	0.8235280
3	0.9757846	0.8376327
4	0.9781044	0.8900684
5	0.9909455	0.8667806

With oversampling:

iter	auc	auprc
1	0.9951570	0.9665361
2	0.9955008	0.9700725
3	0.9931792	0.9639269
4	0.9920948	0.9625218
5	0.9946253	0.9683712

The KNN works on a small subset of variables and it achieves very good results.

1.1.5 C-Tree

iter	auc	auprc
1	0.9190774	0.7717909
2	0.9733520	0.7959343
3	0.9950144	0.9138931
4	0.9769290	0.7984033
5	0.9582785	0.8559226

With oversampling:

iter	auc	auprc
1	0.9899359	0.9464697
2	0.9926569	0.9650465
3	0.9891570	0.9681651
4	0.9935207	0.9690091
5	0.9947102	0.9631525

It has very good results, but it's questionable if it's better than KNN. We would say, that it's not.

1.2 Black-box Models

In this section I will examine the performance of back-box models.

1.2.1 C50

iter	auc	auprc
1	0.9942010	0.8869136
2	0.9846397	0.9584543
3	0.9563858	0.7951213
4	0.9751569	0.7954296
5	0.9923878	0.8432126

With oversampling:

iter	auc	auprc
1	0.9983349	0.9876174
2	0.9977289	0.9878737
3	0.9968033	0.9844093
4	0.9968852	0.9817508
5	0.9942834	0.9585587

C50 shows comparable improvement with best interpretable classifiers.

1.2.2 Adabag Boosting

iter	auc	auprc
1	0.9937299	0.9250668
2	0.9980200	0.9475070
3	0.9989497	0.9656606
4	0.9962340	0.9306854
5	0.9985296	0.9630421

With oversampling:

iter	auc	auprc
1	1	0.9921029
2	1	0.9958073
3	1	0.9881446
4	1	0.9957347
5	1	0.9866360

Adabag has the best results so far. It even maxed out auc in oversampling, which will not be achieved by other classifiers.

1.2.3 Ada Boost

iter	auc	auprc
1	0.9937054	0.9425114
2	0.9982906	0.9524091
3	0.9880114	0.8886003
4	0.9976763	0.9415906
5	0.9984460	0.9515872

With oversampling:

iter	auc	auprc
1	0.9992589	0.9946971
2	0.9999029	0.9949991
3	0.9999847	0.9958075
4	0.9979454	0.9847321
5	0.9996317	0.9954494

Ada boost performed very well, scoring quite steady 0.95. It's not better than Adabag model, but it indicates that Ada/Adabag might be the way to go.

1.2.4 Gradient Boosting Machine

iter	auc	auprc
1	0.9276051	0.6415774
2	0.9337101	0.6092595
3	0.9599625	0.5846263
4	0.9580998	0.7152544
5	0.9441214	0.6945922

With oversampling:

iter	auc	auprc
1	0.9470620	0.8676216
2	0.9477396	0.8871754
3	0.9390531	0.8604404
4	0.9337172	0.8621906
5	0.9439801	0.8729915

Gradient Boosting Machine achieved very low scores, which is kind of surprise, it generally does well.

1.2.5 GLM Boost

iter	auc	auprc
1	0.9159207	0.6351267
2	0.9677110	0.6705510
3	0.9463078	0.6700301
4	0.9590243	0.6623087
5	0.9736615	0.6783640

With oversampling:

iter	auc	auprc
1	0.9531780	0.9032972
2	0.9396781	0.8399779
3	0.9647061	0.8951961
4	0.9677524	0.9457281
5	0.9410588	0.8491252

GLM boost didn't performed well.

1.2.6 Random Forest

iter	auc	auprc
1	0.9951247	0.9261055
2	0.9973677	0.9315830
3	0.9944853	0.9314102
4	0.9930784	0.9236272
5	0.9943840	0.9315555

With oversampling:

iter	auc	auprc
1	0.9999693	0.9955222
2	0.9992569	0.9947587
3	0.9998167	0.9941548
4	0.9998539	0.9925148
5	0.9996528	0.9946762

Random forest shows steady high auprc. It's an indicator of a good model.

1.2.7 Cforest

iter	auc	auprc
1	0.9971805	0.9355238
2	0.9929099	0.8885217
3	0.9902675	0.8988477
4	0.9913621	0.9190240
5	0.9872803	0.9009080

With oversampling:

iter	auc	auprc
1	0.9972927	0.9849820
2	0.9983141	0.9881283
3	0.9961964	0.9799128
4	0.9981982	0.9896381
5	0.9969619	0.9879105

It performed good, but not as good as other classifiers.

2 Conclusions

Ada boost/Adabag boost performed best. Here's comparison of the performance of all classifiers on the test data set:

	auprc	auprc oversampled
classif.glmboost	0.0445585	0.0444613
classif.gbm	0.0708848	0.0708848
classif.mlp	0.0828732	0.0828625
classif.naiveBayes	0.0956296	0.0929974
classif.binomial	0.6081044	0.6339371
classif.ctree	0.7007869	0.6821296
classif.rpart	0.7432471	0.7377270
classif.kknn	0.7879904	0.7385897
classif.cforest	0.8481901	0.8799941
classif.ranger	0.9152835	0.9190056
classif.ada	0.9238044	0.9217160
classif.C50	0.9292426	0.9413438
classif.boosting	0.9507964	0.9534268

As we can see, Adabag boosting classifier achieved the best results, along with C50 and Ada, as we predicted. C-forest performed particularly bad, the opposite of what we predicted. There can be seen a trend, that generally interpretable models perform worse than their back-box competitors, it's not a rule though.

The important thing about oversampling is that it did't change performance of the models. There is a bit of change in auprc in C50, however it just as well might be a statistical error.