Ensemble Methods for Predicting the Likelihood of Survival from Thoracic Surgery Data

Samuel Jackson, Aberystwyth University

***Abstract*—Abstract goes here...**

I. Introduction

Thoracic surgery is a major invasive surgery involving operating on the lungs of a patient. The authors of ref. [1] collected several pieces of possibly relevant data on a number patients who went on to have thoracic surgery. The data also includes a record of whether a given patient survived for longer than one year after the surgery. This paper looks at using a reduced subset of the features and patients from the dataset in [1] to classify patients based on whether or not they will survive for one year after the surgery. This paper compares three different classifiers: random forests [2], extremely randomised trees [3], and gradient boosting [4].

The format of the result of this paper is structured as follows: section II outlines the preprocessing steps performed on the dataset and describes the classifiers used. Section III presents the performance of the classifiers on the dataset. Section IV discusses the results and presents possible justification for the performance based on the properties of the classifier and dataset. Finally, a summary and discussion of possible future directions are discussed in section V.

TABLE I
Description of Columns in the Thoracic Surgery Dataset

Column	Type	Description
DGN	Nominal	Diagnosis: Specific combination of ICD-10 codes for primary and secondary as well multiple tumours if any (DGN3, DGN2, DGN4, DGN6, DGN5, DGN8, DGN1)
PRE4	Numeric	Forced vital capacity (FVC)
PRE5	Numeric	Volume that has been exhaled at the end of the first second of forced expiration (FEV1)
PRE6	Ordinal	Performance status - Zubrod scale (PRZ2, PRZ1, PRZ0)
PRE7	Nominal	Pain before surgery (T,F)
PRE8	Nominal	Haemoptysis before surgery (T,F)
PRE9	Nominal	Dyspnoea before surgery (T,F)
PRE10	Nominal	Cough before surgery (T,F)
PRE11	Nominal	Weakness before surgery (T,F)
PRE14	Ordinal	T in clinical TNM - size of the original tumour, from OC11 (smallest) to OC14 (largest) (OC11, OC14, OC12, OC13)
PRE17	Nominal	Type 2 DM - diabetes mellitus (T,F)
PRE25	Nominal	Peripheral arterial diseases (PAD) (T,F)
PRE30	Nominal	Smoking (T,F)
PRE32	Nominal	Asthma (T,F)
AGE	Numeric	Age at surgery
Risk1Y	Nominal	1 year survival period - (T)rue value if died (T,F) (Class Label)

II. Methods

A. Dataset and Preprocessing

The thoracic surgery dataset used consists of 16 predictors and 300 instances. Table I gives a description of each predictor derived from the original UCI dataset repository [5]. The dataset includes a mixture of both categorical (nominal and ordinal) and continuous data. The final (17^{th}) column of the dataset is the binary class label with value 0 if the patient survived and 1 if they died within one year of surgery.

Several initial observations can be made about dataset prior to any preprocessing steps. One key thing to note about the dataset as a whole is that there is a slight imbalance between the two classes. Only 28% of the dataset is of the positive class (28% of patients died). While this imbalance is not extreme, it can have repercussions for the performance of the classifiers. The accuracy paradox [6] states that a classifier with high accuracy can be built from highly imbalanced training by always predicting the negative class.

The predictor PRE32 is zero for all of the patients in the training dataset. This predictor therefore has zero variation and will not help to discriminate between instance. PRE32 is therefore discarded during preprocessing.

PRE5 appears to have some extreme values. PRE5 corresponds to the FEV1 measure. This would suggest that some patients have an unusually high forced expiration volume. Also, all of the outliers are of the same class. This could cause the classifiers to fit to noise rather than to properly generalise. These instances were therefore removed from the dataset. No reduction in performance was witnessed during cross validation for all classifiers after their removal.

The feature DGN is a nominal categorical predicator. This feature was transformed into series of new features via one hot encoding. Each new predictor is a binary feature which is one if the patient falls into the category and zero otherwise. The original DGN feature is drop after the 7 new binary features are created.

B. Classifiers

Four classifiers were chosen for use on the dataset. The four classifiers used are Random Forests, Gradient Boosting, AdaBoost, and Extra Trees. The implementations of all four classifiers are taken directly from the scikit-learn library [7]. All of these methods are known as ensemble methods. Ensemble methods compose together multiple weak learners a single strong classifier.

TABLE II
MEAN F1, F2 AND F0.5 SCORES FOR ALL FOUR CLASSIFIERS OVER 10 ROUNDS OF 5-FOLD CROSS VALIDATION.

	RandomForest	ExtraTrees	GradientBoost	AdaBoost
F1	0.557502	0.643371	0.582411	0.617863
F2	0.478619	0.589029	0.515569	0.562853
F0.5	0.681112	0.715311	0.676343	0.689950

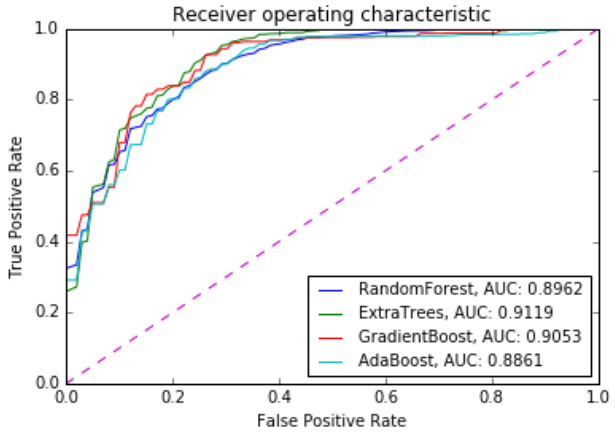

Fig. 1. Mean ROC curves and the mean AUC for all four classifiers over 10 rounds of 5-fold cross validation. All four classifiers perform similarly, with Extra Trees producing the best AUC. All four curves are slightly skewed towards the right of the plot, suggestive of poor recall. The F2 score (table II and figure 2 confirms this.

C. Hyperparameters & Tuning

III. RESULTS

A. Performance Evaluation

Each classifier in section II-B was trained using stratified 5-fold cross validation. Stratification was performed to ensure that there was a representative sample of positive classes in each fold. For all classifiers cross validation was repeated ten times, each with a new set of folds to ensure consistent results.

Figure 1 shows the mean ROC curve and mean AUC for each of the classifier after cross validation. The performance of each classifier appears to be very similar. Notably the ROC curve for each type of classifier is shifted to the right of the graph, suggesting that they all exhibit a low recall rate.

Table II and figure 2 confirm this indication. Table II shows the F measure with a β parameter of 1, 2, and 0.5. Figure 2 shows a bar chart of the F2 scores in table II. The performance of all classifiers measured with the F2 score (which weights recall more highly than precision) is much lower in comparison to the F0.5 and F1 scores. This further confirms that all classifiers have a problem with recall.

B. Engineered Features

In addition to the preprocessing steps outlined in II-A several combinations of new features were generated from the existing predictors. Firstly, as a large portion of the features are binary, a set of new features were created based on logical binary operators. The creation of the binary features is as follows: all pairs of binary features are enumerated. From each pair three new features are created by combining the pair using logical OR, AND and XOR.

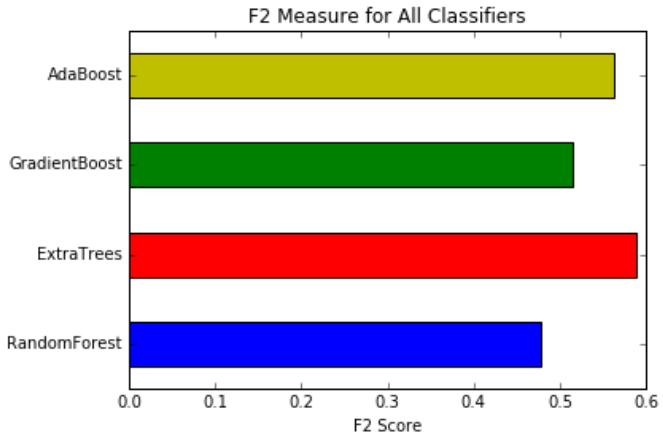

Fig. 2. Bar chart showing the F2 score for all classifiers taken from table II. ExtraTrees and AdaBoost are shown to perform the best. Random forests are shown to be less sensitive to recall.

The second modification to the original dataset is to create a couple of new features called FER and OBS. FER is the FEV1/FVC ratio which is spirometry measurement defined as $(FEV1/FVC) \cdot 100$ [8]. It is interpreted as the percentage of FVC expelled in the first second of a forced expiration. A ratio value below <70% can be suggestive of an obstructive disease. Using this information another feature (OBS) is generated from the ratio. OBS is a binary feature with value 1 when a patient has a ratio <70%.

Finally, a selection of new features are created from all order 2 polynomial combinations of the two best (numerical) predictors: PRE4 and PRE5. This means that the new features are of the form a^2, ab, b^2 where a and b are PRE4 and PRE5 respectively. This polynomial combination led to the best predictive results across all classifiers under cross validation.

C. Dataset Balancing

As mentioned in section II-A the thoracic surgery dataset is class imbalanced with only 28% of the dataset being of the positive class. One technique to combat class imbalance is to resample the dataset to put more emphasis on the known positive examples. A popular technique for resampling data is SMOTE [9]. SMOTE rebalances a dataset by creating new synthetic training to balance out the majority class. SMOTE is typically combined with under-sampling of the majority class to produce a final dataset that is re-weighted in favour of the minority class.

The results for the classifiers in part III-A shows that they have lower recall than precision. Rebalancing the dataset should show a decrease in precision and an increase in recall rate. This can be desirable in a dataset such as this where recall may be more important than precision. It is probably more desirable overestimate the number people who are likely to die from surgery than to achieve better precision.

SMOTE datasets cannot be validated using conventional k-fold cross validation. This is because the testing fold would

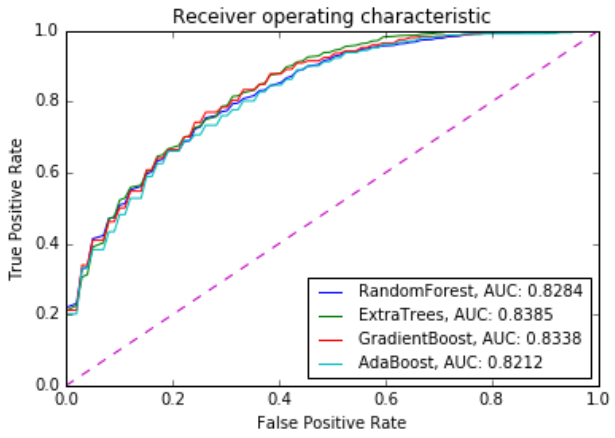

Fig. 3. ROC curves for all four classifiers with SMOTE oversampling with a ratio of 0.8. Each curve represents the average over 50 iterations of Monte Carlo validation. The ROC curves for all classifiers are less skewed compared to figure 1.

TABLE III
MEAN F1, F2 AND F0.5 SCORES FOR ALL CLASSIFIERS AFTER MONTE CARLO CROSS VALIDATION WITH SMOTE RESAMPLING WITH A RATIO OF 0.8

	RandomForest	ExtraTrees	GradientBoost	AdaBoost
F1	0.603586	0.624919	0.617296	0.601894
F2	0.620932	0.663461	0.647674	0.636108
F0.5	0.590391	0.593262	0.592385	0.574231

contain synthetically generated training examples which are obviously not representative of the ground truth. Instead, in order to achieve a representative sample of performance, "Monte Carlo" cross-validation [10] is used. Before a any resampling is applied, the data set is randomly split into a training and testing set. The split is stratified according to the class labels. All reported experiments use and 80/20 split. Resampling is then applied to the training dataset only, with the testing set remaining untouched. This process is then repeated for the desired number of iterations and the resulting performance measures are averaged. In all experiments the number of iterations performed was 50.

Figure 3 shows ROC curve and mean AUC scores for each of the classifiers using SMOTE with a resampling ratio of 0.8. Table III shows the F1, F2, and F0.5 scores for each of the classifiers. Comparing this table to the results of II shows a clear difference in the F2 score. Recall weighted performance is now better both than F1 and F0.5. This improvement comes at the cost of a decrease in both the AUC and F0.5 measures. Increasing the oversampling ratio or under-sampling the majority class accentuates this effect.

IV. Discussion

V. Conclusions

Appendix A
IPython Notebook and Additional Python Files

Appendix B
Predictions for Testing Dataset

References

[1] M. Zięba, J. M. Tomczak, M. Lubicz, and J. Świątek, "Boosted svm for extracting rules from imbalanced data in application to prediction of the post-operative life expectancy in the lung cancer patients," *Applied soft computing*, vol. 14, pp. 99–108, 2014.
[2] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
[3] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.
[4] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in neurorobotics*, vol. 7, 2013.
[5] "Thoracic Surgery Data Set," http://archive.ics.uci.edu/ml/datasets/Thoracic+Surgery+Data, accessed: 2016-04-29.
[6] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[8] "Spirometry, Patient Website," http://patient.info/doctor/spirometry-pro, accessed: 2016-05-04.
[9] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, pp. 321–357, 2002.
[10] W. Dubitzky, M. Granzow, and D. P. Berrar, *Fundamentals of data mining in genomics and proteomics*. Springer Science & Business Media, 2007.