Naïve Bayes and Random Forest applied to Crime Classification

Matthew Tregear and James Hooper

School of Mathematics, Computer Science and Engineering at City, University of London
Matthew.Tregear@city.ac.uk and James.Hooper@city.ac.uk

Introduction and Hypothesis

We compare and contrast the performance of Naive Bayes (NB) and Random Forest (RF) in a binary classification problem, predicting high and low crime rates in areas of London. This is similar to the approach in Iqbal (2013), except they compare the performance between NB and decision trees.[1]

Our initial hypothesis is that **RF will outperform NB on accuracy**. Though NB may be preferred over RF when other factors such as compuational efficiency, generalisability and sensitivity/specificity are taken into account.

Training Methodology

We trained our model on a dataset of c. 5000 rows. We chose to use **k-fold CV for NB** and **hold out CV for RF**.

- For NB, We decided not to go with hold out CV because there was limited data to train a model in our training set. We also chose K-fold CV over leave-one out CV because leave one out may results in high variance in the test error as noted in Kohavi (1995)[2] (which may make it difficult to compare the two models) .
- For RF we chose hold out over Leave-one out or K-fold CV because of the increased computing time. RF also already performs cross validation of a sort in the form of the out of bag error it calculates. And it requires less training data to train a model with good fit.

Naive Bayes

NB calculates the probability of training data being a certain class given the predictor values observed in the training data using bayes formula (ie the prior and the likelihood). It then assigns training data to the class with the highest conditional probability.

NB is a **high bias/low variance algorithm**. This means that it trains models that are consistent but lack accuracy.

The main advantage of NB is **its simplicity**. This means that it is **highly generalisable and less likely to overfit** its training data. It is also easy to interpret and computationally efficient.

The main drawback of NB, outside of its high bias, is its **strong assumption of conditional independence between predictors**. However NB can perform well in particular cases when predictors are functionally dependent. For example in Rish (2001) [3] they find that Naive Bayes can perform well when there are almost deterministic or low entropy dependencies in the data.

Random Forests

Random Forests create multiple decision trees, with the user deciding the amount , using different bagged subsets of the training data. [4]

RFs tend to have a higher accuracy than other classification methods. RFs can handle a larger amount of attributes more easily than other methods. RFs are quite difficult to interpret compared to other methods like Decision Trees and Naive Bayes. Computationally, Random Forests are not very fast.

Evaluation Methodology

We mainly evaluate the two algorithms by comparing **the accuracy of NB and RF**.

However, alongside this we also look at: **generalisability/simplicity** (including the degree of overfitting), **sensitivity and specicificity** (as these may be a more relevant measure of relative performance in certain contexts) and **computational efficiency**.

Lessons Learned and Future Work

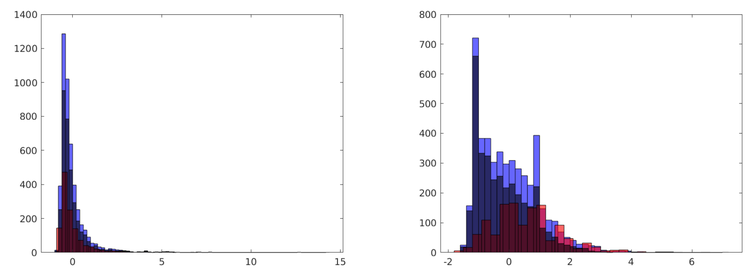
As expected by our hypothesis, RF has greater accuracy than NB. However, accuracy is not the only factor when choosing a machine learning model. Generalisability (ie. the degree of overfitting), computational efficiency and specificity/sensitivity are also important

Going forward we would like could explore a number of variations to our pre-processing/paramter tuning including how:

- NB performance could be improved through other alternative forms of binning (such as entropy-based binning or other methods that better manage discretisation bias and variance as described in Yang(2008) [8].
- RF performance could be improved through Smart Hyperparameter tuning, which uses much less computational time than Grid Search, and could allow us to search through a larger range of hyperparameter combinations.
- the relative performance of NB and RF varies when predictors are added/removed or when the dataset is smaller/larger.

Initial Analysis of the Data

- Our dataset comprises 4736 rows of LSOA level 2001 London Census data.
- We used this data to classify whether crime is high or low (ie above the 150 crime rate mean) using 11 continuous variables and 1 categorical variable (the borough of the LSOA)
- These variables are ones that we picked as we believe they should help predict crime, and align with Iqbal(2013).
- We applied a PCA to our continuous data to try and reduce the dimensions of our predictor data. But this was unsuccessful. (We found that each continuous variable had a large loading above 30 percent in the first few components)
- The covariance between variables was generally quite low. Only Economic Deprivation/Unemployment Rate and Economic Deprivation/Claimant Count had a covariance above 0.8. For this reason independence between predictors is probably not a bad assumption.



(Left - the distribution of Not UK(%), Right - the distribution of House Price)

- Most continuous variables fairly closely followed a normal distribution apart from BAME and Median House Prices. These two variable may be particularly good candidates for binning or kernelisation under NB.
- Summary statistics on the 11 continuous variables are set out below.

	Mean			Standard deviation			Skew		
Summary Stats	**Mean**	**Mean(crime=' high')**	**Mean(crime='low')**	**St.dev**	**St.dev(crime='high')**	**St.dev(crime='low')**	**kurtosis**	**kurtosis(crime='high')**	**kurtosis(crime='low')**
Population	1,717.20	1,916.80	1,645.00	323.94	468.39	209.82	34.54	22.59	6.50
Population Density	94.69	102.02	92.03	59.29	56.90	59.92	6.66	3.38	7.85
BAME(%)	39.15	44.82	37.11	20.40	18.29	20.73	2.36	2.47	2.42
Not UK(%)	35.86	43.00	33.28	14.48	12.08	14.42	2.33	2.95	2.21
House Price	396,020.00	419,640.00	387,490.00	356,580.00	418,850.00	330,860.00	50.61	39.23	56.16
Claimant Count May 2013	42.94	58.16	37.44	27.50	30.69	24.00	5.00	4.59	4.73
No qualification	241.95	259.53	235.60	96.58	107.34	91.58	3.24	3.62	2.74
Unemployed	7.42	8.23	7.12	3.39	3.29	3.38	3.24	3.07	3.66
Truancy	0.98	1.14	0.92	0.49	0.47	0.49	3.39	3.60	4.13
Economic Deprivation	25.16	31.70	22.80	13.19	12.22	12.72	3.82	2.42	2.60
Proportion 16-29	0.22	0.26	0.21	0.07	0.07	0.06	2.41	6.75	8.19

Choice of Parameters and Experimental Results

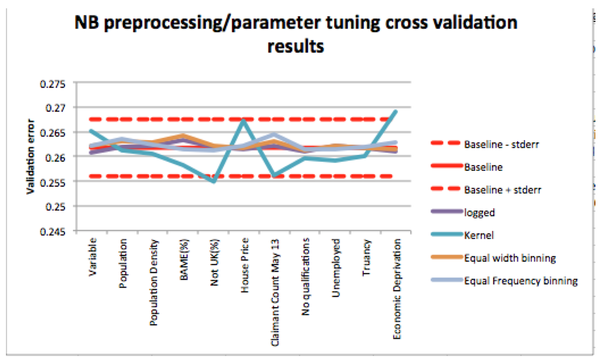
Naive Bayes Pre-processing/Parameters:

We use a gaussian parameterisation for the 11 continuous variables and a multinomial parameterisation for the discrete variable in our NB baseline algorithm

We looked at tuning NB by using **equal width and equal frequency binning**, using a **kernel distribution** [5] and **log transforming continuous variables**.

We found that there was only a small difference in performance for different intervals of binning (for both equal width and equal frequency). However we found that 4 interval binning generally provided the best accuracy(this is used in the diagram below).

More generally, we also found that parameter tuning/pre-processing did not significantly improve the baseline model. Although NB was more accurate than the baseline under certain specifications, it was not significantly above one standard error.So, as the baseline is simpler than parameter tuned variants, it is preferred according to the 'one standard error rule'. [6]



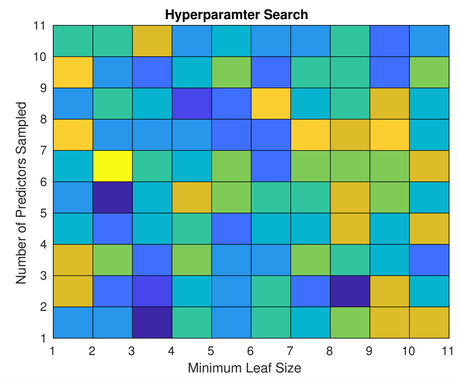
Random Forest Pre-processing/Parameters:

It was possible to improve the Random Forest model beyond the baseline model. For this the hyperparameters used are: **Tree Number of 85, Number of Predictors to Sample of 5 and Minimum Leaf Size of 2.**

We optimized the number of trees, chosen to be 85, by looking at the validation error of the model. Unexpectedly the error was quite unpredictable as the number of trees increased, probably due to the relatively small size of the dataset.

By using a **Grid Search** to search through all combinations of number of predictors to sample and minimum leaf size we chose 5 and 2 respectively, which can be seen below (dark blue the lowest error the light yellow the highest). The error used is also the validation error.

We managed to choose a set of hyperparameters that reduced the model error slightly, though it was not a significant gain given the amount of computing time it took.



Analysis and Critical Evaluation of Results

As shown in the table below, RF significantly outperforms NB on accuracy. NB misclassifies around 9 percent more of the test data than RF.

However, our RF model seems to overfit the data given that its training error is only 4 percent (13 percent less than its test error). NB on the other hand has training error of 31 percent that is much closer to its test error.

NB is significantly more computationally efficient than RF.

	Training Error	Test Error	Test sensitivity	Test specificity
Naïve Bayes	31.06%	26.52%	86.90%	44.88%
Random Forest	4.05%	17.42%	85.04%	75.14%

Both RF and NB have a similar model sensitivity of around 85 percent. This is relevant because certain applications of classifications algorithm may prioritise sensitivity/specificity over accuracy, as in Kazmierska(2008) [7]

For example, if the algorithm was used in an application that was only interested in correctly identifying areas of high crime - without very successfully ruling out areas of low crime - our results may suggest NB is the better choice over RF.

That said, we note in this case that it is unlikely that NB would be chosen over RF - given the loss in accuracy. RF is significantly more accurate than NB plus one standard error.

References

[1] Rizwan Iqbal, Masra Murad, Aida Mustapha, Payam Panahy, and Nasim Khanahmadliravi. An experimental study of classification algorithms for crime prediction. *Indian Journal of Science and Technology*, 6(3):4219–4225, 2013.

[2] George H. John and Pat Langley. A study of cross-validation and bootstrap of accuracy estimation and model selection. *Proc. 14 int. Joint Conf. Artificial Intelligence*, (338-45), 1995.

[3] Irina Rish. An empirical study of the naïve bayes classifier. *IJCAI 2001 workshop on empirical methods in artificial intelligence*, 3(22):41–46, 2001.

[4] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):17–22, 2002.

[5] George H. John and Pat Langley. Estimating continuous distributions in bayesian classifiers. *UAI'95 Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, (338-45), 1995.

[6] T Hastie, R Tibshirani, and J. Friedman. *The elements of Statistical Learning*, 2, 2009.

[7] Joanna Kazmierska and Julian Malicki. Application of the naïve bayesian classifier to optimize treatment decisions. *Radio Therapy and Oncology*, 86(2):211–6, 2008.

[8] Ying Yang and Geoffery I. Webb. Discretization for naïve-bayes learning: managing discretization bias and variance. *Machine Learning,2009*, 2008.