Predicting Credit Card Default: A Comparison of Naive Bayes and Random Forest Classifiers

Aparna Surendra and Zara Thomas

1.Description and Motivation of the problem

- We implement two machine learning methods, Naive Bayes and Random Forest, to predict credit card default (a binary classification problem). We compare the models to one another (using classification error rates, run-time and F1 scores) and simultaneously evaluate the use of two different cross-validation methods (k-fold and holdout).

2. Exploratory Analysis

- The UCI dataset has 30,000 observations on 25 variables related to credit card default. There are 23 predictor features and 1 response feature.
- We remove the 'ID' feature (the identity column), as it does not have any predictive value.
- We look at the summary statistics for each of the variables across the whole dataset, and then segmented by defaulters and non-defaulters.

(Age remains somewhat constant across the groups, but gender, limit balance, and pay amount and bill amounts have different statistical characteristics based on segment -- suggesting that they will be better predictors of default.)

	Limit Balance (μ, σ)	Age (μ, σ)	Gender (%)	Pay Amount 6 (μ, σ)	Bill Amount 6 (μ, σ)
Non-Defaulters (23,364)	178,099 (131,628.35)	35.417 (9.077)	61.4% - Female	5719.37 (18.792)	39,042 (59,547)
Defaulters (6636)	130,019.65 (115,378.54)	35.72 (9.69)	56.7% - Female	3441.482 (13,464)	38,271 (59,579.57)
Whole Dataset (30,000)	167,484.32 (129,747.66)	35.48 (9.21)	60.2% - Female	5215.50 (17,777.46)	38,871 (59,554)

Figure 1: A selection of summary statistics

- This can be considered an imbalanced dataset, as the defaulter class is only 22.12% of the dataset
- There are six months of bill payment and payment delay data. They are heavily correlated. can be seen with the more prominent regions shaded in blue of the correlation matrix (Figure 2).

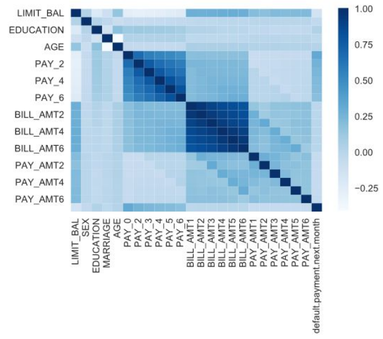

Figure 2: A correlation matrix of model features

3. Hypothesis Statement

Random Forest will be more computationally expensive (as measured by run-time), but will produce better results (as measured by classification error). This is based on Wainer et al.'s findings from comparing 115 binary datasets [1].

4. Description of choice of training and evaluation methodology

We use two forms of cross-validation methods.

- The Holdout method. We split the data 70%-30% into training and test sets. We report in-sample and out-of-sample classification error rates.
- K-fold Cross-Validation method (K = 10). We report average out-of-sample error rates.

The holdout method is less computationally expensive, but can have high variance [2].
The K-fold method is far more expensive, but it relies less on how the data is divided and can provide a lower variance error rate. We compare the error rates and the computational time for each method.

5. Brief summary of Naive Bayes and Random Forest models

I. Naive Bayes

Naive Bayes is a probabilistic classifier that assumes, for a given class, features are independent. The assumption creates a high bias, low variance model. While individual class density estimates may be biased due to the independence assumption, the posterior probabilities (especially near decision regions) are not necessarily much affected [1] .

Pros:

- Highly scalable (linear scaling)
- Fairly high-performing, given simplicity of independence assumption.
- Can be reach reasonable accuracy with fewer observations.

Cons:

- Does not incorporate interactions among variables.
- Naive Bayes can outperform more sophisticated models, but it is generally considered to perform less well than other models. [3]

II. Random Forest

Random Forests are an ensemble learning method for both classification and regression models. This method constructs a multitude of decision trees during training, where each tree gives a classification and votes for a particular class. The forest chooses the classification with the most votes out of all the trees.

Pros:

- Random forests do not overfit - there is no limit to the number of trees that can be run.
- Can be used when there are many more variables than observations
- Incorporates interactions among predictor variables
- It can handle thousands of input variables without variable deletion.
- It gives estimates of what variables are important in the classification.
- It generates an internal unbiased estimate of the generalization error as the forest building progresses.
- It has methods for balancing error in class population unbalanced data sets

Cons:

- Storage requirements increase with an increase in the number of trees.
- Slower to train models. [4]

6. Findings

I. Naive Bayes (NB)

Choice of parameters

- We manipulated the distributions for each of the continuous predictor features. (We assess 5 possible distributions for continuous data: normal distributions, and kernel density estimation with four different kernel options: Normal, Box, Epanechnikov, Triangle).
- Hyperparameter optimisation: We compared exhaustive grid search and bayesian optimisation methods to identify kernel distribution combination with the lowest classification error (if each continuous feature was modeled using kernel density estimation). Compared run-time and error rates.

Main Experimental Results

- The most accurate model was identified through exhaustive grid search. It has an out-of-sample classification error rate of 0.203 (10-fold cross validated). This modeled continuous features with kernel density estimates as follows: Balance Limit (Distribution - Kernel: Box), Age (Kernel: Box), Bill Amount (Kernel: Normal), Pay Amount (Kernel: Box).
- The biggest reduction in classification error comes from adjusting the distributions of the continuous features from 'Normal' (error rate: 0.34) to 'Kernel: Normal' (error rate: 0.211). This corresponds to previous research [5], which demonstrates that kernel estimation methods can better model complex data (for continuous features in Naive Bayes), and can generalise better than models that assume a Gaussian distribution.

Further Comments

- Exhaustive grid search did identify a more optimal combination of kernel distributions for continuous features than did bayesian optimisation (with 30 runs). However, it ran all 256 possible models as opposed to 30, and the corresponding error rate drop (from 0.211 to 0.203) did not warrant the excess computational time (3270.84s vs 18,461s).
- Naive Bayes can produce good performance, even if the independence assumption is violated (as is true in our dataset, as evidenced by Figure 2). This occurs if the dependence of features is evenly distributed across classes, and appears to be true of our dataset [4].

II. Random Forest (RF)

Choice of Parameters

- We evaluated the effect of varying the number of trees (from 1 to 500) on model performance indicator, including run time and accuracy. We also analysed the behaviour of feature importance rankings, and how that varied with the number of trees.

Main Experimental Results

- In comparing, the number of trees with in-sample error rates it was observed that after approximately 200 trees, the error rates plateau around the mean classification error of 0.215.
- The graph shows the relatively linear relationship between the number of trees and the run time. However, there is a spike in trees ranging from 276 to 343. These results demonstrate the trade-off between time and accuracy found in random forests. Based on this evidence, random forests with 200 trees were chosen in comparison with Naive Bayes model.

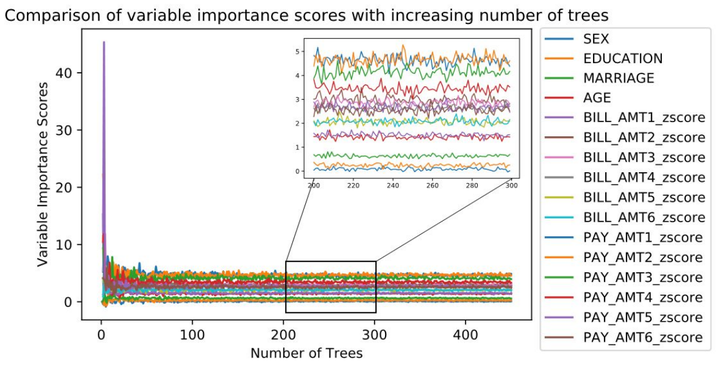

Figure 3

Further comments

- We measured feature importance (the measure used in variable selection) for 1 to 450 trees.
- The variable importance scores remain fairly steady after 100 trees.
- However, some variables fluctuated more than others (Figure 3). We observed that highly correlated features such as the bill amount variables (as displayed in the correlation matrix, Figure 2) fluctuated the most.
- This conforms to previous findings by Gregorutti et al [5] and Breiman [6], who demonstrate that random forests achieve lower generalisation errors if there is lower correlation between classifiers, and higher strength of the trees.

7. Analysis and Critical Evaluation of Results

Unless otherwise specified, analysis compares Naive Bayes ('kernel- normal' models) with Random Forest (200 trees).

	10 - Fold Cross-Validation		Holdout Cross-Validation (70-30 test/train split)				
	Run Time	**Avg. out-of-sample classification error**	**Run Time**	**Out-of-sample classification error**	**Precision**	**Recall**	**F1 Score**
NB: 'Normal'	2.72s	0.3441	0.79s	0.2861	0.3858	0.6614	0.48
NB : 'Kernel - Normal'	133.60s	0.2115	107.73s	0.2084	0.5174	0.4597	0.48
RF: 200 trees	1664.4s	0.2126	108s	0.2134	0.2460	0.5648	0.34

Figure 4: Comparison of models . 'Run time' includes time to build and report classification error for models.

- The 'kernel - normal' Naive Bayes model and Random Forest models produced identical error rates to two decimal places (0.21 out-of-sample classification error). To confirm, we plot the error rates for 10 10-folds models (100 values). As can be seen, the mean Naive Bayes error rate is similar to that of Random Forest, but has greater variance.
- As expected, the Naive Bayes models had the fastest runtimes, as they are less computationally intensive. However, the fastest Naive Bayes model (0.79s) used default Gaussian distribution setting, and had error rates that are ~8% higher than the Naive Bayes model with Kernel- Normal distributions (107.73s). This suggests that parameter tuning can reduce the computational gap between classifiers (Figure 6).
- The error rates for the holdout method were only marginally lower that those for the 10-fold cross-validation, suggesting that the ordering of observations in the dataset do not yield any major trends. However, out-of-sample classification errors were ~6% greater in the 10-fold cross-validated runs for the 'Normal' naive bayes model than for holdout (0.34 vs. 0.29).
  This suggests that the Gaussian distribution better fit continuous data in some of the 'folds' of the dataset than others. This variation is overcome by using the 'Kernel- Normal' Naive Bayes, as the kernel density estimates better fit the continuous data across the dataset (as evidenced by boxplots in Figure 5).
- As expected, 10-fold cross- validation run times were greater than holdout run times for all types of models, as it is a more computationally-intensive method.

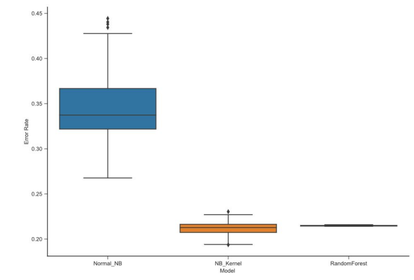

Figure 5: Box plots with 100 sample error means

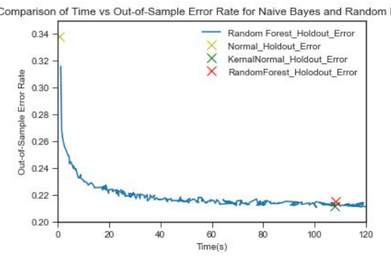

Figure 6

- With imbalanced classes, a classifier that always predicts the majority class can yield high classification accuracy. In these instances, an F1 score can be a better gauge of accuracy, as it weighs both precision and recall scores. Naive Bayes has x2 the precision of Random Forest, but lower recall. Both Naive Bayes models have a higher F1 score than Random Forest (0.48 compared to 0.34), suggesting that Naive Bayes is better suited to a dataset with unadjusted imbalanced classes.
- Ultimately, Naive Bayes exceeds accuracy expectations. This may be explained by the different classifiers' approaches to class imbalance and/or correlated data.

8. Lessons learned

- Naive Bayes - changing predictor distributions for continuous features, especially from Gaussian to kernel distributions, can have dramatic improvement on accuracy rates.
- Random forest- varying the number of trees moves the model along a trade-off curve between accuracy and computational cost.
- Cross-validation method - When the k-folds and holdout method do not reveal significant variation in error rates in the first phase of the analysis, it may suggest that the holdout method is an appropriate (and less computationally-intensive) solo method of providing validated results.

9. Future Work

- *Further hyperparameter optimisation.* We can use bayesian optimisation to adjust other parameters in the classifiers, in order to identify new optima. For Naive Bayes: optimising the prior, and the kernel 'width'. For Random Forest: optimising inner-node size, the maximum number of features, and the maximum depth of each tree.
- *Correlated data.* We can explicitly address issues of multi-collinearity in the data, for instance - through feature selection. Recursive Feature Elimination (RFE) has been found to reduce the effect of correlation on the feature importance measure in random forests [5]. Further work can be done on investigating the effect of RFE on both accuracy rates and feature importance (in Random Forest).
- *Class Imbalance.* We can experiment with mitigating the effect of unbalanced classes (currently 22.12% for defaulters vs. 77.88% for non-defaulters). Perhaps through random over-sampling or under-sampling, or combining Naive Bayes with a neural network (which research suggests can yield a boost of 10%) [8].

Works Cited

[1]Wainer, J. (2016), 'Comparison of 14 different families of classification algorithms on 115 binary datasets.', *CoRR*abs/1606.00930 .
[2] Friedman, J., Hastie, T. and Tibshirani, R., 2001. *The elements of statistical learning* (Vol. 1, pp. 241-249). New York: Springer series in statistics.
[3] Rennie, J.D., Shih, L., Teevan, J. and Karger, D.R., 2003. Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)* (pp. 616-623).
[4] Zhang, H., 2004. The optimality of naive Bayes. *AA*, *1*(2), p.3.
[5] Gregorutti, B., Michel, B. and Saint-Pierre, P. (2017) 'Correlation and variable importance in random forests', *Statistics and Computing*, 27(3), pp. 659–678. doi: 10.1007/s11222-016-9646-1.
[6] Breiman, L. (2001) 'Random Forests', *Machine Learning*, 45(1), pp. 5–32. doi: 10.1023/A:1010933404324.
[7] John, G.H. and Langley, P., 1995, August. Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence* (pp. 338-345). Morgan Kaufmann Publishers Inc.
[8] Adam, A., Shapiai, M.I., Ibrahim, Z. and Khalid, M., 2011, April. Artificial neural network—Naïve bayes fusion for solving classification problem of imbalanced dataset. In *Modeling, Simulation and Applied Optimization (ICMSAO), 2011 4th International Conference on* (pp. 1-5). IEEE.