A Comparison of Random Forests and Naïve Bayes Applied to Predicting Credit Default

Mike Culey and Andrew Gibbs-Bravo

CITY UNIVERSITY OF LONDON EST 1894

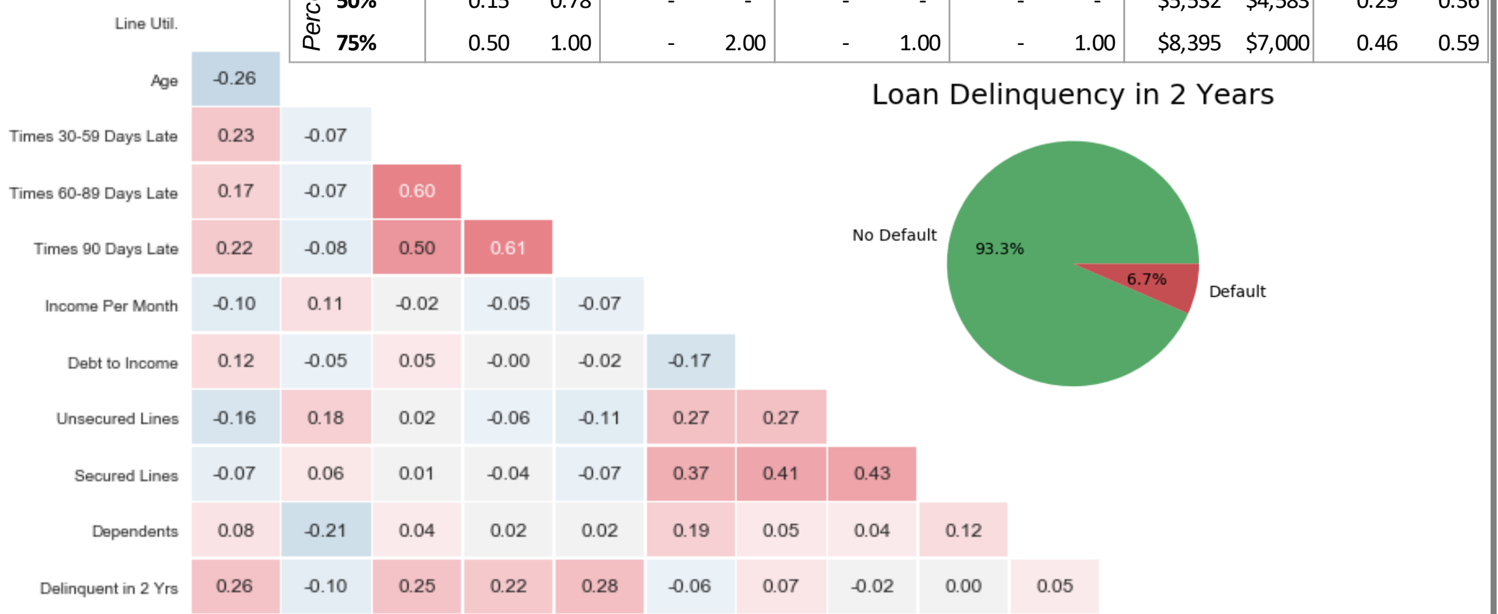
Description and Motivation of the Problem

- Predicting credit default has always been a priority for lending institutions and public awareness of this issue has greatly increased since the 2008 financial crisis
- We took our dataset from a 2011 Kaggle competition. [1] The competition aim was to build the best credit score model to identify potential loan applicants that will experience financial distress in the next two years
- Area Under (ROC) Curve was the evaluation criteria used by Kaggle. Some studies we reviewed also used this or a combination of measures. [6,7,8] We decided to construct a financial parameter closer to what is used in the credit industry for evaluation alongside AUC [5]

Initial Analysis of the Dataset Including Basic Statistics

- The training data set provided by Kaggle was 150k loan applicant records, comprising 10 continuous variable features plus the delinquency in 2 years indicator
    - The dataset is imbalanced with approximately 6.7% of records classed as "delinquent"
    - Initial review and cleaning of data set reduced records to ~117k
- A normal probability plot showed that some features followed a normal distribution over their semi-interquartile range (line utiliz., age, debt ratio, # of open lines)
- There is a relatively high correlation between the 3 late payment features, line utilisation, and the delinquency variable. All other features were within (plus or minus) +0.1 correlation with delinquency i.e. individually they were weak indicators
- Extreme outliers were identified for the "monthly income" and 3 "late days" features. Capping high outlier values for these features resulted in improved performance
- The summary table shows the different distribution characteristics between 'standard and 'delinquent' loans for a selection of features. Future delinquents have higher line utilisation, more late payments, lower income and higher debt to income ratios

		Line Utilization		Times 30-59 Days Late		Times 60-89 Days Late		Times 90 Days Late		Monthly Income		Debt Ratio	
		Standard	Default	Standard	Default	Standard	Default	Standard	Default	Standard	Default	Standard	Default
	Mean	0.30	0.66	0.21	1.11	0.05	0.47	0.05	0.69	$6,548	$5,579	0.35	0.45
	Std. dev	0.33	0.36	0.71	1.99	0.35	1.26	0.40	1.53	$4,329	$3,938	0.36	0.43
Percentile	25%	0.03	0.35	-	-	-	-	-	-	$3,539	$3,000	0.14	0.17
	50%	0.15	0.78	-	-	-	-	-	-	$5,532	$4,583	0.29	0.36
	75%	0.50	1.00	-	2.00	-	1.00	-	1.00	$8,395	$7,000	0.46	0.59



*Maximum values for certain features have been capped at threshold determined by visualization

Hypothesis Statement

- We expect random forests to perform better than naive bayes in terms of AUC and financial performance given that the winners of the competition and reference paper used tree-based approaches or neural networks which are able to form very complex decision boundaries [10, 2, 12]
- Of 7 papers we reviewed relating to credit scoring algorithms, 6 included random forests and 2 included naïve bayes [2, 3, 5, 6, 7, 8, 9]
- Prior Kaggle winners also used very elaborate ensembles of models to improve performance with a particular emphasis on gradient boosted decision trees therefore we expect to achieve lower performance [10, 2]
- Other drivers of performance are expected to include:
    - Creative feature engineering which is expected to improve results given the relatively few features
    - Addressing the imbalance between the majority and minority classes should increase performance for both models
    - Increasing the size of the training set is expected to have a greater benefit to the RF model

Evaluation Methodology

- In order to train the models we split the data into a training and test split with our baseline model containing 25k training samples and 15k test samples
- Used 5-fold cross validation on the training set to tune the model hyperparameters and examine prediction variance
- Retrained the best performing hyperparameters on the full training set and evaluated performance on the unseen holdout test set
- Lessman et al [5] highlight an important managerial question is the impact to the "financial bottom-line" so we developed a metric easily understood in the finance domain. Model performance evaluation is then a 3 stage process:
    - Area Under the Curve (AUC) is calculated to maintain comparability to the Kaggle competition and reference paper
    - A financial error metric (FEM) was then used to identify the optimal point on the ROC curve for which a confusion matrix was generated. FEM = (NPV defaulted loan * false negatives + NPV standard loan * false positives) / number of loans
    - The confusion matrix derived from the FEM is then used in an 'NPV per loan' calculation to estimate the financial impact of different scenarios on each model and allow comparison between models

Random Forests (RF)

Model Review Including Pros and Cons

- Random Forests are an ensemble of decision trees which improve on the performance of individual decision trees (which tend to overfit) by using bootstrap aggregation which builds many trees and then predicts based on the majority vote of trees for classification [27, 4, 13]
- Introduces two sources of randomness to decorrelate the trees and improve performance
    - Randomly samples a subset of training data
    - Randomly selects a subset of features to use in splitting

Pros

- Decision trees are capable of capturing complex non-linear relationships and have low bias with sufficient depth. Averaging large numbers of decorrelated decision trees is an effective way of reducing overfitting [13]
- Produces indication of feature importance based on occurrence and hierarchy of features used to split (depending on depth)
- Does not require substantial preprocessing of data (e.g. standardization) and is robust to noise
- Easy to parallelize which can significantly improve computation time [14, 15]
- Achieves performance similar to boosting while being easier to train and tune [16, 13]

Cons

- Can be relatively computationally expensive depending on hyperparameter configuration
- Offers less interpretability than other algorithms for individual decisions [17]
- Does not consider the relative strength of trees and assigns equal weighting [16]

Choice of Parameters and Experimental Results

Hyperparameters

- Used gridsearch with k-fold cross validation to determine the optimal tree depth
- Number of sampled predictors is equal to the square root of the total predictors [13]
- Number of trees in forest is set at the highest number computationally feasible [18, 12, 4]

Experimental Results

- Observed that the greater the tree depth the better the AUC performance tends to be across scenarios
- The cross-validation test performance experienced relatively low variance across hyperparameters
- Performance was not substantially improved by increasing size of training data

Naïve Bayes (NB)

Model Review Including Pros and Cons

- Naive Bayes classifies a new example by estimating the probability that the example belongs to each class (e.g.; case delinquent or non-delinquent) and selects the class with the highest probability
- NB selects the category with the highest probability, so ranking is important even if the actual calculated probability values themselves can be inaccurate [11]
- NB performs better with categorical input variables compared to numerical variables [11]. For numerical variables a distribution assumption is required (e.g. a normal distribution)

Pros

- Simple classifier taking all feature evidence into account. Efficient with respect to storage space and computation time [11] for large test data sets. It also performs well in multi-class prediction
- Results are good training with small data sets and NB is an 'incremental learner'
- When assumption of independence holds, a NB classifier performs better compared to other models like logistic regression

Cons

- NB assigns a new category (not seen in training data) as a zero probability. One workaround is to use a Laplace correction
- The underlying theoretical assumption of NB is that features/predictors are independent - e.g. this assumption often viewed as simplistic or "naïve". But even when assumption is not true, NB can perform 'surprisingly well' [11]
- Irrelevant features / redundant features can skew results so should be removed

Choice of Parameters and Experimental Results

Hyperparameters

- The Matlab NB function has 4 data distribution modes, 2 for multinomial, and 2 for continuous distributions: 'Normal' and 'Kernel'
- Kernel itself had 4 with different distribution settings (that could be specified at individual feature level)

Experimental Results

- Performance by Kernel setting in descending order was: 'Normal', 'Triangle', 'Epanechnikov', 'Box'
- However as each Kernel setting performed worse than Normal on the cross validation test set, Normal was used as the default distribution mode

Analysis and Critical Evaluation of Results

Naive Bayes Results

- Achieved its best AUC performance without engineered features and 0.25 oversampling (T2). Worst performance was with correlated features removed and outliers uncapped (T6 and T11). Degree of oversampling had little effect
- For NPV, the engineered features had a slight benefit (1 cent). Worst scenarios as per AUC above

Random Forests Results

- Best AUC and NPV when outliers were uncapped (T11) although performance was not significantly impacted when outliers were capped (T2) highlighting RFs robustness to outliers. Worst performance when correlated features were removed (T6)

Comparison of Models

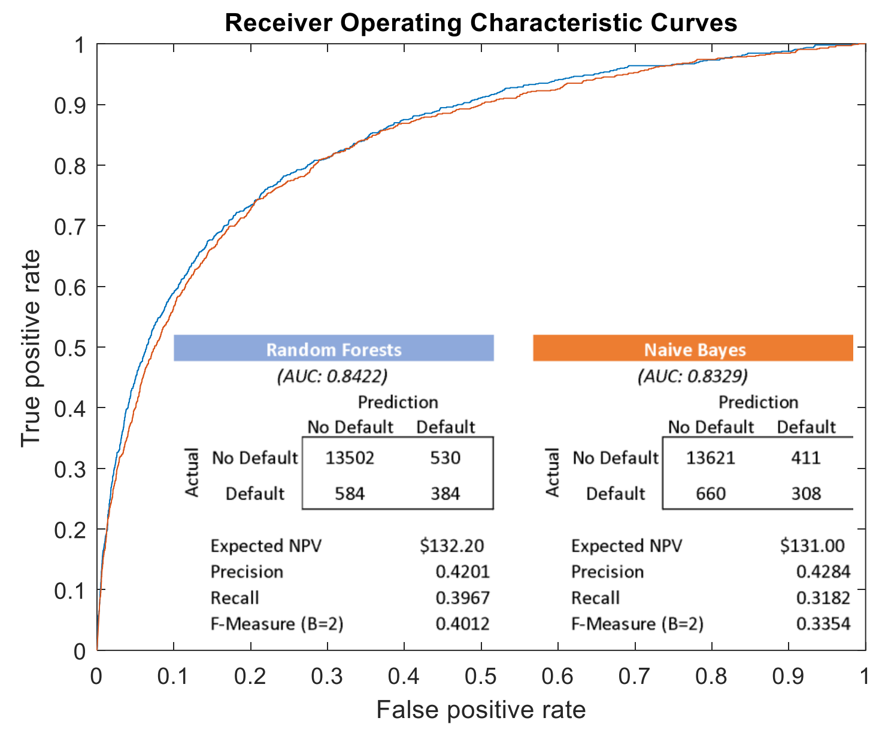
- RF outperforms NB in terms of both AUC and expected NPV in all cases except when training on 1000 training samples (T7)
    - RF's superior performance can be attributed to its ability to develop complex decision boundaries while not overfitting due to randomly sampling features and bootstrapping [4, 3]
    - The only case in which NB outperformed RF in terms of NPV was with a very small training sample size which is consistent with other studies demonstrating the ability of NB to perform well on small sample sizes [19, 20]
- RF AUC benefits slightly from increased oversampling although the relationship with NPV depends on cost of false positives and false negatives. (T1-T3) Given our financial NPV assumptions the optimal oversampling proportion for RF at a 0.50 threshold is around 0.25 which is shown in the chart to the right
    - The impact of oversampling on AUC appears to be somewhat dataset specific.[21, 22, 23, 24] However, oversampling increases recall at the expense of precision for a given threshold [25]
- RF and NB performance improves only slightly from 5,000 trainings samples vs 50,000 samples (T8 vs T9)
    - Once the critical information has been extracted from a dataset, incremental training observations are unlikely to improve performance [26]
- RF performance is constant when outliers are not capped although NB performance declines dramatically (T11)
    - RF relies on constructing multiple decision trees which are not impacted by the magnitude of the values on the sides of the decision boundary
    - Naive Bayes with a normal kernel evaluates the mean and variance of each continuous feature by class and is therefore heavily impacted by extreme values
- Both algorithms have the worst performance when engineered features and highly correlated features are excluded (T6)
    - While several features are correlated to one another they also contain significant signal and therefore weak performance results from removing correlated features and engineered features showing the importance of feature engineering [26]
- NPV per loan with 100% accuracy would be $157.21 versus if we had issued loans to everyone (dummy classifier) of $123.71

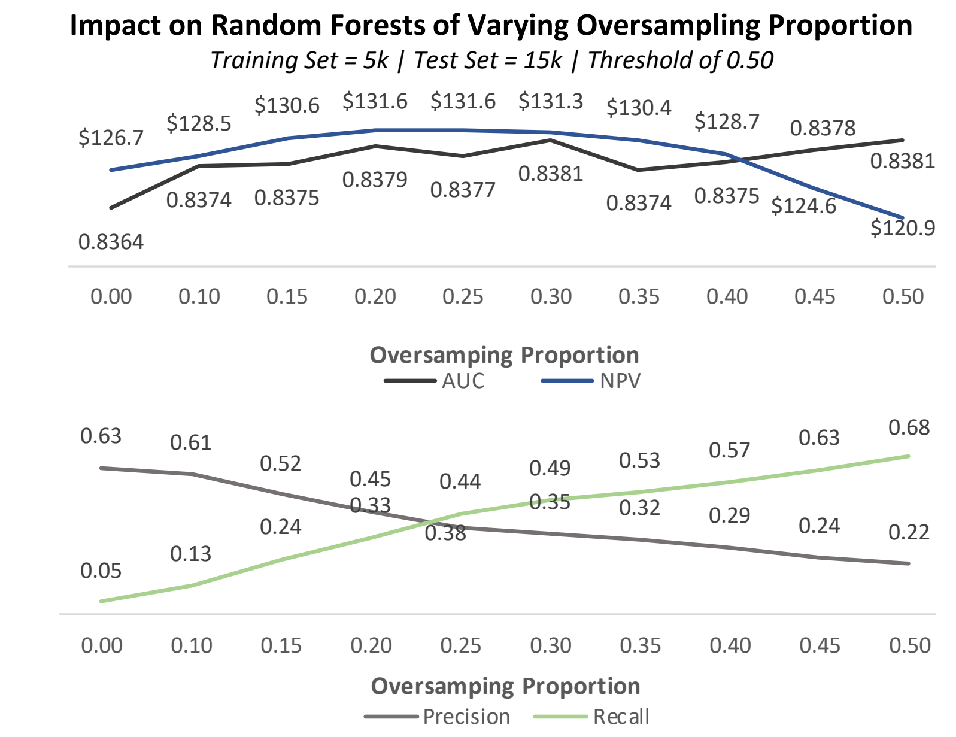
Hypothesis Summary

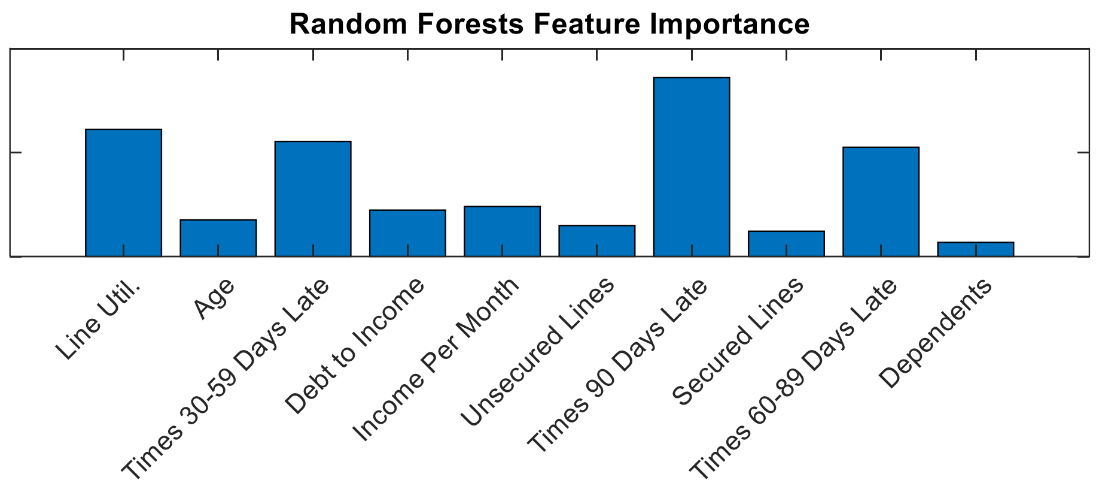
- Random forests did perform better than Naive Bayes for both AUC and financial performance
- These single models did not beat top 3 Kaggle winners using an ensemble of models
- The 3 engineered features we created did NOT improve results
- Increasing the oversampling proportion was beneficial up to 40% for AUC in general (no benefit found using higher %)
- Increasing the size of the training set was beneficial up to 25k records

Summary Scenario Evaluation Table

Test Number	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
N-train Samples (Test: 15k)	25,000	25,000	25,000	25,000	25,000	25,000	1,000	5,000	50,000	90,000	25,000
Oversample	0	0.25	0.4	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Incl Engineered Features				TRUE	TRUE						
Feature Selection				TRUE		TRUE					
Include Outliers											TRUE
RF AUC	0.8399	0.8422	0.8424	0.8393	0.8403	0.7961	0.8204	0.8377	0.8401	0.8384	0.8422
RF NPV per $1k loan	$131.95	$132.20	$132.07	$132.10	$132.23	$128.61	$129.81	$131.84	$131.85	$131.75	$132.25
NB AUC	0.8319	0.8329	0.8323	0.8174	0.8282	0.7856	0.8032	0.8286	0.8317	0.8317	0.7858
NB NPV per $1k loan	$130.97	$131.00	$130.99	$131.05	$131.29	$127.85	$130.92	$131.28	$131.01	$131.04	$127.07
RF AUC > NB AUC	0.0079	0.0093	0.0100	0.0219	0.0121	0.0105	0.0172	0.0091	0.0085	0.0067	0.0564
RF NPV > NB NPV	$0.98	$1.20	$1.08	$1.05	$0.94	$0.76	-$1.11	$0.56	$0.84	$0.71	$5.18







Lessons Learned and Future Work

- The NPV-per-loan criteria provided a comparison metric that was useful in this analysis and is more meaningful to credit officers than AUC. Given the dataset there was surprisingly little financial impact between models
- The lower number of features and their relatively high correlation made improving model performance very challenging. This highlights the importance of having high quality data and feature engineering
- In our analysis the best NB scenario scored AUC: 0.8329 whilst RF scored AUC: 0.8424. The winning Kaggle team scored AUC: 0.8696 after making 128 entries which we assume were iterative improvements
- The top 3 winners used an ensemble of between 5 to 15 models (each included random forests) and manufactured up to 35 new features [10] and 1 team had applied some real world credit rules to classify data (eg applicants over 60 years had more stable incomes through pensions of state assistance, lending institution decision criteria includes classification by debt income ratio thresholds). It was noted that some other credit-style datasets had more informative features
- Suggested future work is to investigate ensembling multiple decorrelated models which is expected to yield improved results. In particular adding a gradient boosted decision tree model is expected to improve performance
- Additional suggested future work would be engineering additional features which may improve performance although given small number of features and their correlation it will be challenging to improve performance dramatically

1. Kaggle link: https://www.kaggle.com/c/GiveMeSomeCredit
2. Jitendra Nath Pandey, Maheshwaran Srinivasan," Predicting Probability of Loan Default", Stanford University, CS229 Project report, 2011
3. Caruana, R. & Niculescu-Mizil, A. 2006, "An empirical comparison of supervised learning algorithms", ACM, , pp. 161.
4. Breiman, L. 2001, "Random Forests", Machine Learning, vol. 45, no. 1, pp. 5-32.
5. Lessmann, S., Baesens, B., Seow, H. & Thomas, L.C. 2015, "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research", European Journal of Operational Research, vol. 247, no. 1, pp. 124-136.
6. Brown, I. & Mues, C. 2012, "An experimental comparison of classification algorithms for imbalanced credit scoring data sets", Expert Systems With Applications, vol. 39, no. 3, pp. 3446-3453.
7. Kennedy, K., Namee, B.M. & Delany, S.J. 2013, "Using semi-supervised classifiers for credit scoring", The Journal of the Operational Research Society, vol. 64, no. 4, pp. 513.
8. Fitzpatrick, T. & Mues, C. 2016, "An empirical comparison of classification algorithms for mortgage default prediction: evidence from a distressed mortgage market", European Journal of Operational Research, vol. 249, no. 2, pp. 427-439.
9. Barboza, F., Kimura, H. & Altman, E. 2017, "Machine learning models and bankruptcy prediction", Expert Systems With Applications, vol. 83, pp. 405-417.
10. "No Free Hunch" - http://blog.kaggle.com/tag/give-me-some-credit/
11. Provost and Fawcett, 2013, "Data Science for Business", O'Reilly
12. Khoshgoftaar, T.M., Golawala, M. & Van Hulse, J. 2007, "An Empirical Study of Learning from Imbalanced Data Using Random Forest", IEEE, , pp. 310.
13. Hastie, T., Tibshirani, R. & Friedman, J. H. (Jerome H.) 2017, The elements of statistical learning: data mining, inference, and prediction, Second edn, Springer, New York.
14. Sharp, T. 2008, "Implementing Decision Trees and Forests on a GPU" in Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 595-608.
15. Van Essen, B., Macaraeg, C., Gokhale, M. & Prenger, R. 2012, "Accelerating a Random Forest Classifier: Multi-Core, GP-GPU, or FPGA?", IEEE, , pp. 232.
16. Robnik-Šikonja M. (2004) Improving Random Forests. In: Boulicaut JF., Esposito F., Giannotti F., Pedreschi D. (eds) Machine Learning: ECML 2004. ECML 2004. Lecture Notes in Computer Science, vol 3201. Springer, Berlin, Heidelberg
17. Palczewska, A., Palczewski, J., Robinson, R.M. & Neagu, D. 2013, "Interpreting random forest classification models using a feature contribution method", .
18. Probst, P. & Boulesteix, A. 2017, "To tune or not to tune the number of trees in random forest?", .
19. Forman G., Cohen I. (2004) Learning from Little: Comparison of Classifiers Given Little Training. In: Boulicaut JF., Esposito F., Giannotti F., Pedreschi D. (eds) Knowledge Discovery in Databases: PKDD 2004. PKDD 2004. Lecture Notes in Computer Science, vol 3202. Springer, Berlin, Heidelberg
20. Ng, A.Y. and Jordan, M.I., 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In Advances in neural information processing systems (pp. 841-848).
21. Tantithamthavorn, C., Hassan, A.E. & Matsumoto, K. 2018, "The Impact of Class Rebalancing Techniques on the Performance and Interpretation of Defect Prediction Models",
22. Xue, J. & Hall, P. 2015, "Why Does Rebalancing Class-Unbalanced Data Improve AUC for Linear Discriminant Analysis?", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 5, pp. 1109-1112.
23. Dal Pozzolo, Andrea, Olivier Caelen, Reid A. Johnson, and Gianluca Bontempi. "Calibrating probability with undersampling for unbalanced classification." In Computational Intelligence, 2015 IEEE Symposium Series on, pp. 159-166. IEEE, 2015.
24. Chawla, N.V., Bowyer, K.W., Hall, L.O. & Kegelmeyer, W.P. 2002;2011;, "SMOTE: Synthetic Minority Over-sampling Technique", Journal of Artificial Intelligence Research, vol. 16, pp. 321-357.
25. More, A. 2016, "Survey of resampling techniques for improving classification performance in unbalanced datasets", .
26. Domingos, P. 2012, A few useful things to know about machine learning, ACM, New York.
27. Artur Garcez. Lecture Notes 6. INM431 Machine Learning.