A Comparison of Naive Bayes and Tree Based Ensembles Applied to the Bank Marketing Dataset

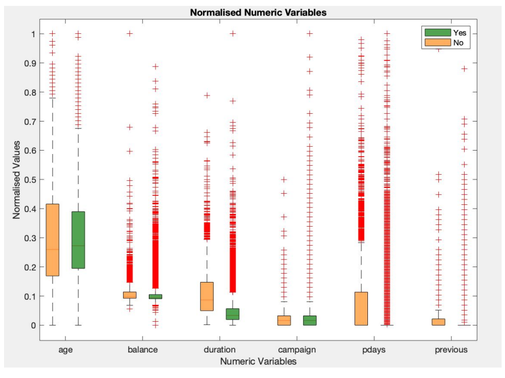
Matthew Galloway and Nuray Golde

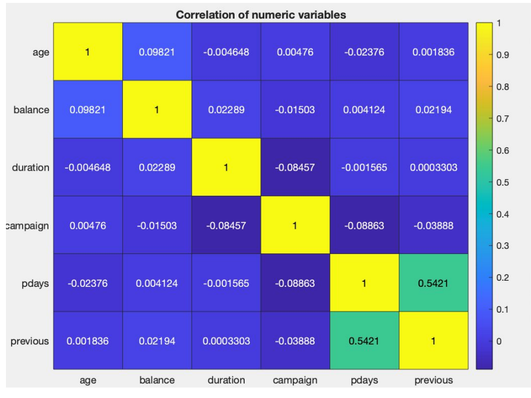
1.0 Motivation and problem description

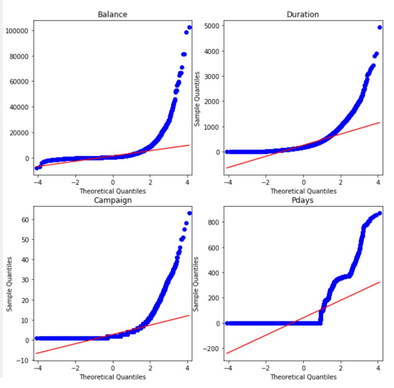
Data driven decision making is becoming more common as data sources and machine capabilities increase[1]. When engaging with clients these often presents binary classification problems, will a client buy a product? As the majority of customers do not respond to marketing, datasets are highly unbalanced leading models to take a majority vote. Neural nets, SVM, decision tree and naive bayes are all have been applied to this problem but random forest is yet to be explored[2][3][4]. The literature has focused on parameter selection to provide the best dataset, however comparing sampling methods across machine learning techniques and in depth hyperparameter optimisation to improve the ability to predict if a customer will respond yes (minority class) to marketing is unexplored.

2.0 Dataset Descriptions and preliminary analysis

- 45,211 rows of data were analysed in order to examine the variables that contributed to customers subscribing to a banks loan product.
- The data set was highly unbalanced 39,922 (88.3%) responded no and 5,289, (11.7%) responded yes.
- There were 9 categorical and 6 numerical variables - with no null values.
- Most variables showed a high degree of positive skew and deviated from normal distributions significantly.
- 1 outlier in the 'previous' column and 2 in the 'balance' column were replaced by their respective means to aid visualisation.
- Numeric values showed low degrees of correlation, with the exception of 'pdays', suggesting Naive Bayes is likely to perform well
- Numerical values were normalised using min max scaling to aid visualisation and allow k means resampling algorithms to work, it's noted min max scaling is not explicitly required for Random Forest or Naive Bayes.
- Min/max scaling was chosen over standard gaussian normalisation methods as distribution of data were not normal.







3.0 Two Machine Learning Models and their Pros and Cons

3.1 Naive Bayes

- Makes probabilistic predictions about a class using Bayes Theorem. Converts prior probabilities in to posterior probabilities and selects a class using a maximum a posteriori rule [4].
- Assumes that prior distribution exists. Unlike other machine learning models that make estimate priors from the data, naive bayes offers an easy way to adjust priors iteratively using domain knowledge.
- Assumes independence of features, but often performs well regardless of whether this condition is met. [4]

Pros:

- Although a relatively simple algorithm, it can still outperforms more sophisticated techniques on certain tasks [5].
- Interpretability of naive bayes is high when compared with RF and is often preferred in industry as a result.
- Assumed independence of features makes algorithm run times fast when compared with random forest.
- Allows the use of kernel density estimation, a non-parametric which eliminates the requirement to make assumptions on each variables distribution.

Cons:

- Dealing with continuous variables often leads to computational heavy solutions, eg kernel optimisation[6]
- Usually outperformed by more sophisticated machine learning techniques[5].
- Unable to see how the interaction of all features affect the outcome prediction

3.2 Random Forest

- Random forest classifier creates a forest of decorrelated decision trees, through ensemble methods, with each tree voting on the classification. [7]
- Decorrelation is achieved by limiting number of predictors on each decision tree, reducing variance of the ensemble classification as a result.[8]
- Each tree receives bootstrapped data which further reduces the variance of the ensemble predictions[3].

Pros:

- Performance of a random forest ensemble generally outperforms single decision trees and is competitive with state of the art machine learning techniques (neural networks) on some tasks without calibration[5].

Cons:

- Visualization and interpretability is significantly reduced when compared to single decision trees.
- Increasing forest size and tree depth whilst generally increase accuracy, make the algorithm computation expensive to run, when compared with simpler algorithm like Naive bayes.

4.0 Hypothesis

In general literature shows RF to outperform NB, however there are problem dependent exceptions [5]. For this particular problem it is expected that random forest will outperform naive bayes as decision trees have already been shown to have competitive performance with NB[2][4] We expect random undersampling to show the worst performance across both algorithms due to the loss of data and SMOTEEN to perform best as it is seen as and advancement on SMOTE based methods in its removal of outliers during oversampling. We expect kernel optimisation of NB to improve its predictive ability given the non gaussian distribution of continuous features.

5.0 Training and evaluation methodology

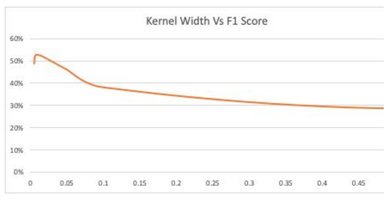
- 80 % of the data was used for training and 20% was set aside for testing. 10 fold cross validation for all training metrics were used.
- Due to the class imbalance present in the data (88.3% vs 11.7 %) over, under and over and under sampling was done on the training data. To ensure models were not simply performing a majority vote F1 scores were considered for optimisations.
- Categorical data was label encoded rather than one hot encoded to allow for Naive bayes kernel optimisations.
- Data was min/max scaled prior to applying sampling methods as both oversampling methods use KNN and relative positions in vector space affects results.
- Its noted that neither scaling nor encoding of categorical variables is explicitly required for NB or RF prior to optimisation.
- SMOTE data was used for algorithm optimisation (tree depth vs kernels) to ensure fair results.

6.0 Choice of parameters and experimental results

6.1 Naive Bayes

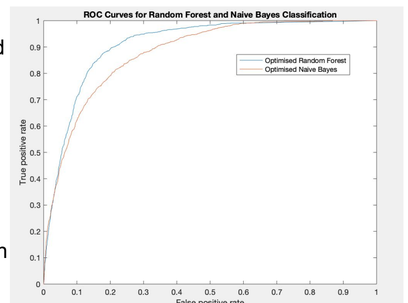
Parameters

- Sampling methods were used to investigate the class imbalance.
- As sampling was shown to have a non statistically significant effect on F1, SMOTE data was used for comparison with RF (where it had a significant effect).
- Kernel density estimates were made for continuous variable and grid search performed to find the optimal kernel widths.



Main Experimental Results

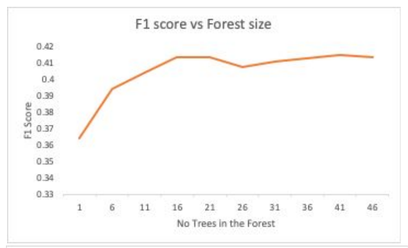
1. Naive Bayes F1 score was significantly better than RF on an unsampled dataset, contrary to our assumption that the effect on the priors of a Naive Bayes model would make it a worse predictor.
2. Counterintuitively, **log transformation** of the continuous variables decreased the F1 score for the gaussian naïve bayes model by 42%, although post logging the variables were closer to the normal distribution.
3. **Sampling methods:** had no positive effect on F1 score although the precision recall balance changed, as models got worse at predicting majority class.
4. **Kernel** optimisation of continuous variables improved the F1 score, with a grid search showing optimal values at width of 0.01.
5. Changing the **kernel smoothing type** from gaussian to box, triangle and epanechnikov had no effect on the results.

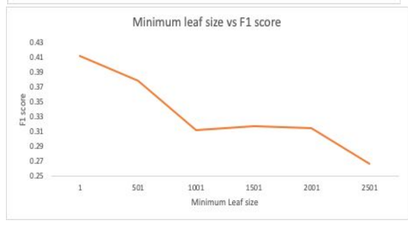


6.2 Random Forest

Parameters

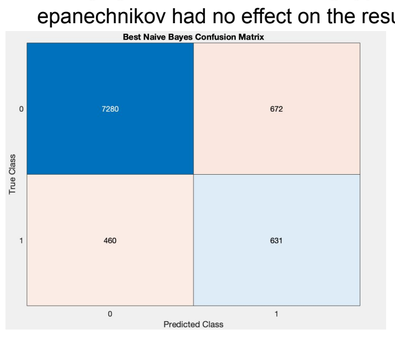
- Sampling methods were investigated to correct the data class imbalance, with size of the forest, min leaf size and max no of splits, at 100 (the default for the algorithm)
- The best sampling method was then used to investigate algorithm optimisations:
  - Forest Size/No of Splits/Min leaf size
- Grid search on the aforementioned was used to find the best result
- Boosting was applied to these optimisations



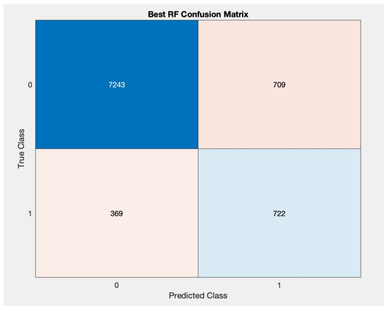


Main Experimental Results

1. Random forest recall rate of 25.98% before resampling data showed a poor ability to classifying the minority class.
2. **Sampling methods:** greatly improved F1 and recall scores, SMOTE was the best method.
3. **Forest size**: Increasing from a forest size of 2 trees to 16 showed increase in F1 score, however past this point changes in F1 score were negligible
4. **No of splits:** had no effect on F1 score
5. **Min leaf size:** F1 score got worse as min leaf size increased from 1
6. **Grid search** was performed on SMOTE data to show max no of splits at 7001 and min leaf size of 1 were the best model.
7. **Boosting** RUSBoost/AdaBoost/GentleBoost all increased F1 score



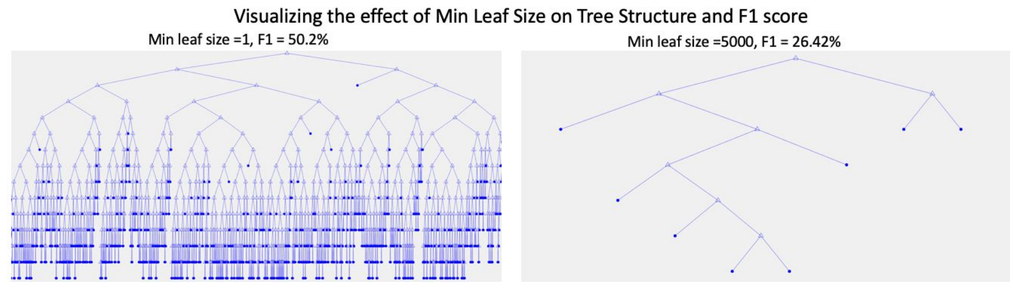
Naïve Bayes							Random Forest					
Test	**Train**	**F1**	**Precision**	**recall**	**AUC**	**Sampling method**	**Test**	**Train**	**F1**	**Precision**	**recall**	**AUC**
82.55%	83.40%	40.04%	33.95%	48.79%	0.81	**No method**	89.75%	90.28%	37.20%	65.58%	25.98%	0.90
74.74%	79.22%	39.73%	25.39%	90.69%	0.81	**SMOTE Oversampling**	78.69%	85.72%	48.57%	35.62%	76.32%	0.88
74.49%	74.80%	39.44%	26.18%	79.95%	0.80	**Random Undersampling**	79.37%	81.12%	45.88%	31.63%	83.53%	0.88
75.03%	80.65%	36.67%	23.14%	88.27%	0.82	**SMOTENN**	78.71%	85.58%	48.45%	35.44%	76.55%	0.88



7.0 Analysis and critical evaluation of results

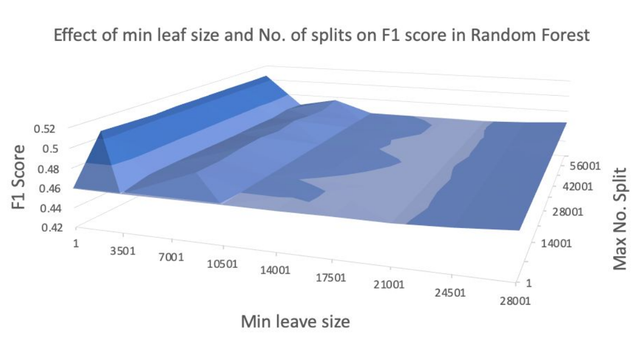
Data Level Optimisation:

- In contrast to our hypothesis Naive bayes outperformed Random Forest on unsampled data based on F1 score 40.04% vs 37.20%, this was due to RF taking a more significant majority class vote than naive bayes (RF recall = 25.98%).
- As expected the RF (AUC 0.9) outperformed the decision trees in literature (AUC 0.87), even though a larger number of features were used in literature so a direct comparison would require further work[3].
- Investigating sampling methods across the models showed improvements in RF F1 score but had no significant positive effect on NB. This was contrary to our hypothesis, as we expected rebalancing the dataset and changing the priors of NB would have a significant negative impact on F1.
- Its thought that undersampling increased F1 score of RF by balancing the classes; at the cost of the precision of model. The removal of data from the majority class is likely the reason the model was worse at classifying this class, evidenced in training and test scores dropping equal amounts.



- Both SMOTE based sampling methods for RF showed improvements in recall and F1 score when compared to random undersampling. Its thought that the retention of the data of the majority class is the reason for this. However precision was still significantly lower than when no sampling method was applied.
- Our initial hypothesis was SMOTENN would outperform SMOTE as the noise generated in oversampling would be removed, this was not the case however for either model [10]. It is thought that the algorithm was unable to correctly assign clusters to the dataset.
- Data level optimisations highlighted the importance of balancing accuracy, precision and recall. Rebalancing data sets changes the priors of the model and will typically lead to lower accuracy as the precision decreases and majority class predictions get worse. However this led to better recall scores as the ability to predict the minority class increased. These considerations would be important in industry where slightly more unconventional optimisations metrics like recall score could be used if the focus is only on understanding the minority class (if customers respond yes to marketing).
- A final consideration with sampling methods was the size of the data and how it affected run time of the models, with more computationally heavy models like random forest of large size and tree depth there was a factor of 4 difference in run time between the oversampled and undersampled data sets.

Best Naïve Bayes		**Best Random Forest**
85.75%	**Train**	98.80%
74.69%	**Test**	78.63%
52.72%	**F1**	57.26%
48.43%	**Precision**	50.45%
57.84%	**recall**	66.18%
0.88	**AUC**	0.91



Algorithm Optimisations on resampled SMOTE data:

- Evaluating the improvement in F1 score vs the size of the forest was consistent with literature, averaging more decision trees gives better results, attributed to law of large numbers. [7] A drop off in performance improvements was seen after a forest size of 16 and as this ran 6 times faster than the algorithm default (forest size of 100), 16 was chosen as the hyper parameter for investigation..
- Both minimum leaf size and no of split were evaluated as proxies for tree depth. It was shown that deeper trees improved the prediction of F1 score with minimum leaf size being the more significant optimisation parameter.
- Given the non gaussian distribution of continuous features kernel density estimation was applied within naive Bayes, where it improved results. This was in line with our hypothesis that non parametric assumption would be a better representation of the data.
- Kernel density estimation massively outperformed normal Naive bayes but the 600 fold increase in computational time relative to a unoptimised naive bayes decreased the value of these result. This was especially true given a random forest optimised for tree depth out performed the kernel naive bayes in every parameter and ran 283 times faster.

8.0 Lesson learned and future work

1. Despite the simplicity of Naive Bayes classifier its out the box performance is competitive with the more sophisticated ensemble methods and it runs much faster (prior to optimisation).
2. Resampling data changes the models priors, it was assumed this would have a large effect on NB F1 scores and not on RF, however the inverse was true. Further working investigating changing the priors while maintaining the data may provide insight on why.
3. Boosting was briefly investigated and showed promising improvements on a random forest, future investigations looking at how tree structures differ in a boosted model vs a traditional random forest potentially could highlight why.
4. SVM is reported within the literature as the most successful classifier for this problem, applying different resampling methods to SVM could further improve the models performance [2].

9.0 References

[1] Chen, Chiang, and Storey, 'Business Intelligence and Analytics: From Big Data to Big Impact', *MIS Q.*, vol. 36, no. 4, p. 1165, 2012.
[2] S. Moro, P. Cortez, and P. Rita, 'A data-driven approach to predict the success of bank telemarketing', *Decis. Support Syst.*, vol. 62, pp. 22–31, Jun. 2014.
[3] D. Pavlovic, M. Reljic, and S. Jacimovic, 'Application of data mining in direct marketing in banking sector', *Industrija*, vol. 42, no. 1, pp. 189–201, 2014.
[4] H. A.Elsalamony, 'Bank Direct Marketing Analysis of Data Mining Techniques', *Int. J. Comput. Appl.*, vol. 85, no. 7, pp. 12–22, Jan. 2014.
[5] R. Caruana and A. Niculescu-Mizil, 'An empirical comparison of supervised learning algorithms', in *Proceedings of the 23rd international conference on Machine learning - ICML '06*, Pittsburgh, Pennsylvania, 2006, pp. 161–168.
[6] A. Pérez, P. Larrañaga, and I. Inza, 'Bayesian classifiers based on kernel density estimation: Flexible classifiers', *Int. J. Approx. Reason.*, vol. 50, no. 2, pp. 341–362, Feb. 2009.
[7] G. James, D. Witten, T. Hastie, and R. Tibshirani, Eds., *An introduction to statistical learning: with applications in R*. New York: Springer, 2013.
[8] M. N. Murty and V. S. Devi, 'Bayes Classifier', in *Pattern Recognition*, vol. 0, London: Springer London, 2011, pp. 86–102.
[9] L. Breiman, 'Random Forests', *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[10] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, 'A study of the behavior of several methods for balancing machine learning training data', *ACM SIGKDD Explor. Newsl.*, vol. 6, no. 1, p. 20, Jun. 2004.