Comparing Naïve Bayes with Logistic Regression

Babak Hessamian

Description and motivation of the problem

→ Our classification problem is to predict the success of telemarketing calls for selling bank long-term deposits.
→ We use the logistic regression and naïve bayes methods to build machine learning models that can predict the outcome of a telemarketing call – we compare the performance of the two methods.
→ This predictive knowledge can help managerial decision making in the business to target those clients who are most likely to buy the long-term deposit product.

Exploratory Analysis

→ Dataset: Bank Marketing Data Set from UCI
→ It has 4521 instances, a 10% subset of the original dataset. This was chosen to reduce time when building models.
→ It has 7 numeric and 9 categorical predictors , plus one target variable.
→ There is a class imbalance: 88% said 'no' to signing for a long-term deposit product (Fig.1).
→ Extensive feature selection has already been carried out already [1] to reduce the number of variable from 150 to 17.
→ Numeric predictors are mostly independent which is important as one of the methods is Naïve Bayes (Fig.3).
→ Chi-square test was performed to rank predictors in order of importance/contribution [8]. Duration of the telemarketing call is by the far the biggest contributor (Fig.2).

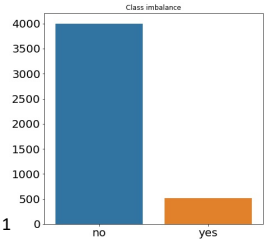


Fig. 1

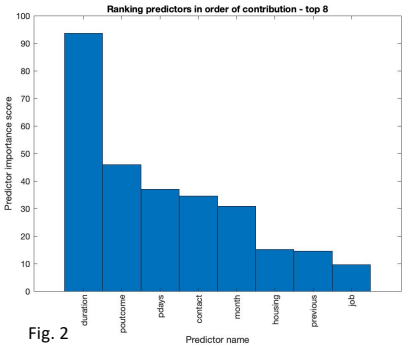


Fig. 2

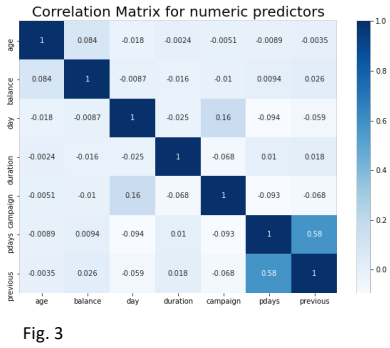


Fig. 3

Models

Naïve Bayes (NB)

→ A generative classifier since it specifies how to generate the observed features *x* for each class *y* [2].
→ It applies Bayes Theorem under different distributions to predict the probability that a given observation belongs to a particular class.
→ By using Bayes Theorem, the prior, our assumption about the model, is updated with each iteration and becomes our posterior.

Pros
✓ Simple to implement and immune to overfitting [2].
✓ Quick to train and predict.
✓ Easy to explain as based on the simple concept of Bayes Theorem.

Cons
✕ Assumes features are independent which limits its use.
✕ Underperforms compared to many other models including neural network and support vector machines.
✕ Choice of the density curve (Gaussian/Bernoulli, Multinoulli) depends on type of each feature. This limits hyperparameter tuning when the data set contains a mix of real-valued, binary, and categorical predictors [2].

Logistic Regression (LR)

→ Discriminative classifier since it discriminates between class labels by modelling the mapping from input *x* to output *y* [3].
→ A form of a Generalised Liner Model (GLM).
→ It calculates the probability of an event occurring (p) to it not occurring ($1 - p$) – or the *odds ratios*.
→ It uses the training data and a gradient descent algorithm to estimate the maximum likelihood estimation for new data.

Pros
✓ Simple to implement.
✓ Easy to extend to multi-class classification problems.
✓ Generally, performs well even with smaller datasets.

Cons
✕ A rigid model as no hyperparameters to tune.
✕ As a GLM, it does not adequately model complex non-linear relationships [1].
✕ Only relies on feature engineering to improve performance.
✕ Not as flexible as models such as support vector machines where there are no prior assumptions made about the data [1].

Hypothesis

→ Both models will perform better than random guess.
→ LR to perform better NB.
→ Expect regularisation to have little affect on the LR model performance as extensive feature selection has already been carried out [1].
→ Expect to have limited scope to tune Naïve Bayes hyperparameters as our data contains a mix of categorical and real-valued predictors [2].
→ Expect NB to be quicker than LR [4].

Methodology

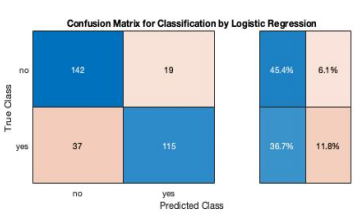
The general steps taken were as follows:

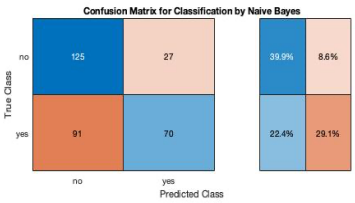
1. Preprocessed and did initial investigation in Python, including re-ordering columns, dummy coding, checking for missing values.
2. Created a Matlab file for each method and split the data into train and test sets using the same proportions and functions.
3. Trained LR model and try different regularisations (stepwise and lasso). Categorical predictors were dummy coded and the numeric predictors normalized.
4. Trained NB model and try cross validation and different distributions, priors and widths.
5. Saved models and use on test set.
6. Used ROC plots, confusion matrices, F1 scores to compare the two methods and link to reference papers.
7. Reflected on the results and shortcomings.

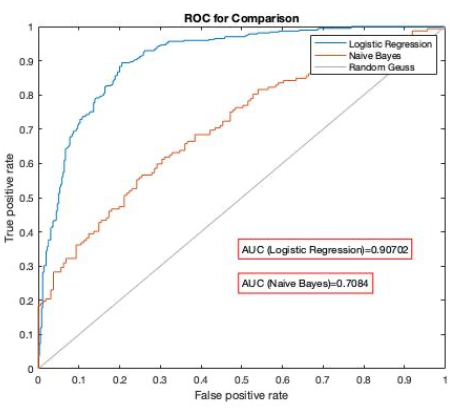
→ The reference paper uses LR but not NB. We compare these two as they are a popular Generative-Discriminative pair to compare [5] and are both simple and use probability distribution, yet are expected to perform differently.
→ We used exactly the same partitioning and data preparation methods for a fair comparison. This can be seen in the comparison.m file.
→ We had a class imbalance. This was dealt with in our reference paper by a rolling window evaluation procedure. This was too complex for the scope for this coursework. In our approach, we randomly under-sampled the over-represented class.

Results & Analysis

→ Both models performed better than random guessing.
→ LR preformed better than NB as demonstrated by the confusion matrices and ROC curves below.
→ For our NB model using a 10-fold cross validation with a prior set to 0.7/0.3 yielded very similar results to not using any cross validation. We got an unusually high rate of false negatives despite changing priors to default setting of 0.5.
→ Choice of distributions for hyperparameter tuning in our NB models was limited to normal and mvmn (Multivariant Multinomial) distributions.
→ Using Lasso regularisation for feature selection in our LR model did not improve the performance and added to computation time. Stepwise regularisation took too long to compute and was abandoned.
→ The ROC curve shows the cross validated NB model compared to a simple LR model.
→ NB was quicker than LR: up to five times in training and three times in predicting.







→ In general, Naive Bayes performance is dependent on the tuning of hyper-parameters and feature engineering, whereas logistic regression is mainly dependent on feature engineering. So we tried to improve model performance with each of these approaches.
→ Lasso regularisation is used for feature selection. In the context of a GLM it does this by penalizing the negative log-likelihood with l_1 – *norm* [7]. When we applied L1 Lasso regularisation to our LR method, it was suggested that numerous predictors were to be removed. We did not remove any predictor but used the lasso model coefficients to examine the performance of the model. We did not see any improvement in accuracy. This may be because extensive feature selection had already been carried out [1].
→ We tried several hyperparameter optimization methods for NB in a for loop, but we were limited to using the mvmn distribution in Matlab as our dataset contains a mix of real-valued non-normally distributed predictors and categorical predictors. This limited our options to optimise our NB models.
→ The reference paper uses a time ordered split to divide the dataset into training (2008 to 2012) and test sets (2012 to 2013) [1]. This was not possible as the dates are not provided with the dataset for data privacy reasons. Despite this, we achieved a similar ROC curve for LR to that in the paper. However due to the reduction of our training set by under sampling the over-represented class, we may have compromised on the bias-variance trade off and made the model less generalisable.
→ Vapnik [6] argues that classification problems should be solved directly rather than through solving intermediate problems, as it is the case with NB. Indeed, we find this to be true as our discriminative classifier (LR) has an AUC of 0.90 while our generative classifier (NB) has an AUC of 0.70. However, there may be computation time and simplicity benefits to using a cross-validated and optimised NB model.

Future Work

→ A more robust validation on the larger version of this dataset can be carried out with more holdout runs for both methods to try and improve accuracy.
→ Also, our random splitting of the training and test sets does not reflect the temporal dimension that a real prediction system would have to follow, i.e. using past patterns to fit the model in order to issue predictions for future client contacts.
→ Synthetic Minority Oversampling Technique (SMOTE) can be used to address the class imbalance and compare the results with our method of under-sampling the overrepresented class.

1. S. Moro, *et al*. "A data-driven approach to predict the success of bank telemarketing." *Decision Support Systems*, vol 62 , pp 22-23, June, 2014.
2. : K. Murphy. "Generative models for discrete data " in *Machine Learning, Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.
3. K Murphy. "Logistic regression" in *Machine Learning, Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.
4. A.Y Ng, and M.I Jordan. "On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes." NIPS, 2001.
5. G.J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. Wiley, New York, 1992.
6. V.N Vapnik. *Statistical Leaning Theory*. John Wiley & Sons, 1998.
7. V Fonti and E Belitser. "Feature Selection using LASSO". VU Amsterdam. March, 2017.
8. T Parlar and A.K Acaravci. "Using Data Mining Techniques for Detecting the Important Features of the Bank Direct Marketing Data". *International Journal of Economics and Financial Issues*, vol. 7(2), pp. 692-696, 2017.