Decision Tree and Naïve Bayes application to Bank Marketing Data

Humza Khan

humza.khan.2@city.ac.uk

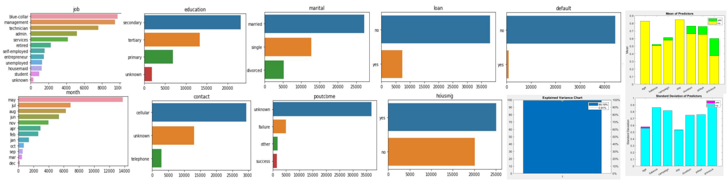
Student Number: 040008336

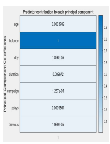
Brief Description and motivation of the problem

- The data is comprising of direct marketing campaigns (phone calls) of a Portuguese Bank. The data file reveals a Portuguese bank which has had a decline in revenue due to infrequent depositing by their clients and would like to assess future actions that can be undertaken. Term deposits allows banks to invest in higher earning financial products and this is combined with cross selling further products to their clients to increase revenue. The aim is to use a classification approach to predict if existing clients will subscribe (yes/no) to a term deposit ('y') and hence the bank can focus their efforts on these clients.

Initial analysis of the data set including basic statistics

- The Bank Marketing Dataset is collected from UCI website with 11,162 rows and 17 attributes in total including 1 dependent variable (age, job, marital, education, default, balance, housing, loan, contact, day, month, duration, campaign, pdays, previous, poutcome & deposit)
- Our data contains 45,211 observations of 17 features, where 7 are numerical and 10 categorical features. 16 predictor/independent attributes and 1 dependent. We use a holdout of 35% which brings the training set to 29,388*17 and test set to 15,823 *17.
- Dataset was checked for missing values and categorical and numerical features identified.
- The dataset does contain unknown values, and these were kept as this information is not known when the call is performed.
- Categorical features: ['job', 'marital', 'education', 'default', 'housing', 'loan', 'contact', 'month', 'poutcome']
- Numerical features: ['age', 'balance', 'day', 'duration', 'campaign', 'pdays', 'previous']
- The target variable was transformed to binary (1 for yes and 0 for now) using Label encoding and the predictor columns were re-arranged to show the numerical columns followed by categorical columns.
- A statistical summary was produced to show the count, min, max and median values. We observed differences in the mean/standard deviation in the two classes (yes/no). The normalized mean and standard deviation was calculated for each numerical column for each classes as discussed above.
- Principal Component Analysis was conducted to transform features with the least residual variance into planes. First principal component analysis explained 99.19% of the variance.
- The contribution of each variable in the PCA was visualized and we found that the contribution to first principal component is most with 'Balance' and least with 'Campaign'.





	age	balance	day	duration	campaign	pdays	previous
count	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000
mean	40.936210	1362.272058	15.806419	258.163080	2.763841	40.197828	0.580323
std	10.618762	3044.765829	8.322476	257.527812	3.098021	100.128746	2.303441
min	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.000000	0.000000
25%	33.000000	72.000000	8.000000	103.000000	1.000000	-1.000000	0.000000
50%	39.000000	448.000000	16.000000	180.000000	2.000000	-1.000000	0.000000
75%	48.000000	1428.000000	21.000000	319.000000	3.000000	-1.000000	0.000000
max	95.000000	102127.000000	31.000000	4918.000000	63.000000	871.000000	275.000000

Two ML models with their pros and cons

Decision Tree:
- Decision Trees are a supervised learning technique used for classification and regression problems. The model attempts to predict the value of the target variable by learning simple decision rules taken from the data's features. The complexity of these rules increases the deeper the tree grows. A decision tree consists of nodes which shows each feature, branch which shows a decision rule and the leaf nodes displays an outcome.

Advantages :
- It is able to handle both categorical and numerical data where it automatically bins categorical variables.
- It requires less data preparation, simple to understand, and performs well even if its assumptions are somewhat not held.
- It outperforms other models in terms of Accuracy, Execution Time and Precision.

Disadvantages:
- Overfitting is the main problem where the tree keeps generating new nodes to fit the data hence becoming overly complex to interpret. This also leads to a higher variance.
- An Ensemble Tree provides a higher precision but takes longer to computer.

Naive Bayes :
- Naïve Bayes is a classifier model that separates data into different classes based on Bayes Theorem which is an extension of conditional probability. Naive Bayes assumes that all predictors are independent of each other. i.e features are independent of a given class and hence a simplified learning process. Naïve Bayes predicts based on an objects probability which makes it a probabilistic classifier.

Advantages :
- It requires only a small amount of training data to estimate the test data hence it works well with small dataset works providing better AUC and Accuracy values.
- It performs better than other classifier models if the assumptions of independent predictors are true.

Disadvantages:
- It assigns a zero probability "Zero Frequency Problem" to categorical data present in test set but not during training dataset and hence will fail to provide predictions in this regard.
- It makes unrealistic assumption that all features are independent.
- The performance (Accuracy, Execution Time, Precision and F-Score) deteriorates as the dataset size increases.

Hypothesis Statement

- Optimised models can be created by tuning parameters for both models such as binning, splitting etc.
- Sérgio Moro and Raul M. S. Laureano found Naive Bayes Model (reported AUC 0.87) to outperform Decision Tree Models (reported AUC 0.86) in certain situations.
- We will run the final models on the full datasets and a subset datafile for the failed models provided in the supplementary page.
- We will compute model performance results such as Accuracy, Recall, Precision, F-Score, ROC/AUC, Confusiong Matrices, Misclassification Errors an Execution Time.
- Both models are expected to produce good Accuracy and Precision Scores although it is anticipated that Decision Tree will outperform Naïve Bayes as it performs better, is more flexble with larger datasets and automatically carries out feature selection.

Description of choice of training and evaluation methodology

- Naïve Bayes and Decision Tree binary classification algorithms were created which consists of 16 inputted attributes.
- The data was split into approx. two thirds training and a third for testing.
- We experimented the K-Fold and holdout methodologies on large datasets.
- After carrying out an error analysis, a holdout of 35% was chosen to carry out for our final models.
- The models were tested using both training and test datasets.
- Performance metrics such as Accuracy, Precision, Recall, F-Score, AUC, ROC Curves and confusion matrices were calculated on both training and testing datasets.

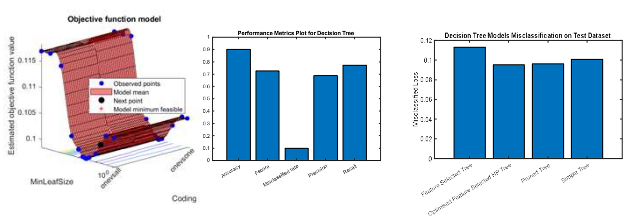
Choice of parameters and experimental result

Decision Tree

Parameters
- A holdout of 35% split the data into training and testing datasets.
- MATLAB in-built function was used to feature select most relevant predictors.
- An importance score of 0.006 was defined and predictors that exceeded this score were taken as inputs for our model.
- This importance score was visualized using the graph 'Predictor importance on response'.
- Binning was experimented with to find the most efficient bins required to improve the performance metrics and AUC values.
- Hyperparameter Optimisation was used to find the best model giving the best performance metrics.
- A multi class model was used to train fully and return the multi class error correcting model.
- Multi – class model was used to return fully trained multi –class error correcting code model.
- In order to speed up the computations, parallel computing was used.

Main Experimental Results
- From the graph for importance score, we find that the greatest model was created using the input predictors duration, month and poutcome.
- The minimum life size as **49,** bins as **52** and coding method of '**onevsone**'

Further Results
- As we note from the graph, "Duration of Call" was the most important predictor in predicting on our response variable.
- The unknown function is estimated using Bayesian optimization technique on the training dataset. The estimated function is then used to predict the testing dataset stats. (Fig 8)
- ECOE model integrated the tree template function which enabled it to optimize the model faster.
- Naïve Bayes model are represented in the graph titled "Performance Metrics for Naïve Bayes". We can see high values of these metrics and a low misclassification error.

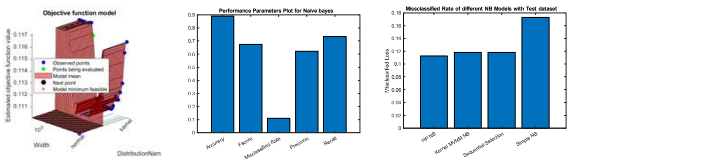


Naive Bayes

Parameters
- The data was split two thirds and a third for training and testing datasets respectively.
- The features that mattered most in predicting the value of response variable were separated from those that were not good at predicting using sequential feature selection. This is shown in the graph that shows the best predictors on the response variable.
- A loss function was utilised to select these predictors that gave the least loss.
- To avoid outliers in our model, we normalised (using z-scores) our predictors before inputting into our model.
- The predictors used Kernel distribution for numerical data and Multi-Variate Multi-Nomial distribution for categorical data.
- HyperParameter Optimisation was carried out in our models to find the final optimized model. The performance metrics for the final Naïve Bayes model are represented in the graph titled "Performance Metrics for Naïve Bayes". We can see high values of these metrics and a low misclassification error.

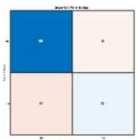
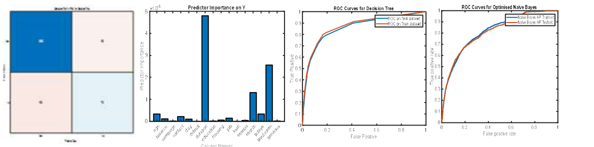
Main Experimental Results
- The distribution selected was gaussian and the best predictors that were selected for our model were (age, job, marital and call duration) In order to speed up the computations, parallel Bayesian optimization was carried out.

Further Results
- We found that feature selection reduced the loss for NB Models and experimenting with different distributions.
- The performance of the model is affected by type of distribution and the width. Using z-scores improved the width of the model and made computation easier. Using z-scores hence improved our performance metrics (accuracy and precision values.)



Analysis and critical evaluation of results

- We analysed our two models by comparing their ROC Curves which is a useful tool when predicting the probability of a binary outcome. The plot is of True and False Positive Rates with a higher ROC plot being better as it identifies the level of class discrimination. AUC show sus how much the models are capable of distringuishing between classes.The Higher the better. AUC of close to 1 represents good separability and a poor model with an AUC close to 0. Sérgio Moro and Raul M. S. Laureano mentioned similar methods in reading the AUC calculations with 0.5 representing a random classifier. Our results concluded that we were able to tune our model in order to increase our prediction techniques and thus showing a improvement in the results.
- After running our final models in Matlab we found the test AUC values to be 0.8592 for decision tree model and 0.8108 for naïve bayes. Overall we found decision tree model to perform better than naïve bayes as we can see from the small differences in performance metrics (Accuracy, Precision, Recall etc).
- Decision tree performed the evaluation in 56.785317 seconds whereas Naive bayes took a total of 276.874130 seconds. The higher elapsed time with Naïve Bayes may be due to the larger dataset which works better with decision tree modelling.
- The Decision Tree model was first tuned and optimized by varying the min leaf size, bins and error correcting codes. We calculated the loss on different models which were created by varying parameter values. This was further observed by analysing the performance metrics such as accuracy, precision, recall and F-Score. At first the models did not perform as well as we wanted and gave a high error values. Tuning the models by finding the optimal bins and finding the most important predictors by feature selection improved the model. We found the performance results to have improved however the evaluation time was larger for our training data over the testing dataset. To reduce this effect, we decided to use parallel computing.
- Tuning our model (increased binning) gave us a better performing ECOC model with higher AUC values on our test dataset. We found OnevsOne coding worked better to our hyper optimised ECOC model where OnevsOne predicts one class label and the one with the most predictions is voted for. We also showed the confusion matrix to show the TN, TP, FN and FP values by looking at the actual and predicted classes.
- Looking at the graph for most important predicted feature, we found this to be 'Call Duration' which implies that the longer the call, the better chance of success. Other features which were found to be important are 'Previous Year Outcome', i.e. repeated customer interaction from previous year and also the 'Contacted Month'. Some important correlating factors were found to be between 'Previous Year Outcome' and success rate as there is a higher chance that a customer re-subscribes each year. The months (Mar, Jun, Sep and Dec) has a higher success rate most likely because of seasonality and customer behaviours. Paulo Cortez observed a similar outcome. This information can prove useful for managers looking to improve in their marketing techniques.
- Naïve Bayes Classifier performance is linked to its distribution width and name. We found different distributions worked better in certain cases, for example Kernal worked well with numerical data. We found that the training dataset has a influence on the model if the dataset consists of both numerical and categorical. Gaussian Distribution worked best for our dataset when we compared against Multi-Variant Multi-Nomial and normal distributions.
- We used sequential feature selection to select the most relevant columns/predictors to be used in the model. The quality of the model was improved by following this process in order the select the most important predictors and ignoring the least relevant. This reduced the errors and the model sensitive improved vastly, enhanced the model and thus provided better performance metrics for Naïve Bayes Model.
- Hyper-Parameter Optimisation provided us the most optimised model and consisted of low variance as shown by the marginal difference between the test and training performance values for Naïve Bayes Model. We found that the Naïve Bayes Model performance is highly dependent on the dataset. For example we found Decision Trees to have performed better with a larger dataset than Naïve Bayes Model, which works better with smaller data.
- We can see from the Naïve Bayes Loss graph that the misclassified errors can be reduced by changing parameters such as distribution name. Hence this would have a positive effect on our performance metrics and higher/better values can be obtained.
- The author I.Rishi tries to understand the data characteristics which affect the performance of the Naïve Bayes Model. Monte Carlo Simulations are used to analyse the classification accuracy for several classes. The impact of distribution entropy on the classification error, low entropy features yielded the best results. Filtering the predictors to only the good ones was used here similar to our own work.
- To conclude we found using predictive classifier models such as Decision Trees and Naïve Bayes can be a key tool for marketing campaigns to reduce costs from Labor force/Time etc and to have a targeted approach which works better. For example we can have better success targeting our campaign during certain months, increasing duration of call and comparing to previous years responses from clients.

	Decision Tree		Naïve Bayes	
	Training	Testing	Training	Testing
AUC	0.8707	0.8592	0.8147	0.8094
Accuracy	90.60%	90.17%	88.98%	88.95%
Precision	70.32%	68.58%	62.32%	62.32%
Recall	78.66%	77.37%	73.46%	73.28%
Misclassification	9.40%	9.83%	11.02%	11.05%
F-Score	74.25%	72.71%	67.43%	67.35%

Lessons Learned & Future Work

- We found that Decision Trees is less computational heavy in comparison to Naive Bayes. As the dataset increases, Naïve Bayes becomes more complex and computational time increases.
- To improve on this model in the future, we could look at the FN and FP and try to compare them to columns correctly predicted. We can therefore identify features that are influence the FN and FP and do further feature engineering.
- We could also provide probability estimates for our predictors so the marketing campaign can focus on clients individually and attain information such as how long to spend on the call with them using their own judgements or if they should skip clients overall.
- We could carry out undersampling techniques before fitting the data to the model as the we found the data to be slightly biased.
- We could further look into missing values and how each model approaches when the dataset size is increased.
- We can tune the tree depth further for Decision Tree Models and the width for Naïve Bayes Models to see if it improves performance. Also we can see how adding extra client information will improve the model further and how each model handles this extra information.

References

- [1] 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, 2014.
- [2] Shamala, Palaniappan & Mustapha, Aida & Mohd Foozy, Cik Feresa & Atan, Rodziah. (2017). Customer Profiling using Classification Approach for Bank Telemarketing. JOIV : International Journal on Informatics Visualization. 1. 214. 10.30630/joiv.1.4-2.68.
- [3] A. Y. Ng and M. I. Jordan, "On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes." pp. 841–848, 2002.
- [4] J. J. Chen, C. A. Tsai, H. Moon, H. Ahn, J. J. Young, and C. H. Chen, "Decision threshold adjustment in class prediction," SAR QSAR Environ. Res., vol. 17, no. 3, pp. 337–352, 2006. [5] F. Kaya, "Discretizing Continuous Features for Naive Bayes and C4.5 Classifiers," Univ. Maryl. Publ., 2008.
- [5] Rish, Irina. (2001). An Empirical Study of the Naïve Bayes Classifier. IJCAI 2001 Work Empir Methods Artif Intell. 3.
- [6] https://archive.ics.uci.edu/ml/datasets/bank+marketing