A Critical Evaluation of Bank Marketing Data through Multilayer Perceptrons and Support Vector Machines

Bernardo Saab Martiniano de Azevedo Bernardo.Azevedo.2@city.ac.uk

Abstract

In this paper we selected two algorithms, Multilayer Perceptron (MLP) and Support Vector Machines (SVM), to perform a classification task of term deposit subscription for clients. For both algorithms, over-sampling and under-sampling techniques are combined to explain the difference between highly imbalanced classes given a 1:9 approximate ratio. A randomized search method is implemented to solve a hyperparameters optimization problem and a stratified cross-validation is used to strengthen the reliability of results and avoid overfitting. The results from the best evaluated models are compared for both classes using ROC-AUC and PR-AUC as well as Confusion Matrices to identify the effects of rare samples in evaluation sets. It was concluded that despite the implementation of a novel optimizer for MLP called AdaBound, the performance of SVM was better and better classification scores were obtained for the minority class and a reduced probability of making a Type I error.

1. Introduction and Problem Statement

Predicting future outcomes of customer behavior from marketing selling campaigns based on historical data is a complex task and considered to be an NP-Hard problem [1]. Companies invest a considerable amount of time and money in statistical analysis and several service channels are utilized to achieve targets of customer related products and most of the times the actual results are not aligned with the expectation. Understanding which features are the most important and to what extent each instance affect a customer profile will determine the probability of long-term product subscription observations and their impact on the continuity of companies banking services over time.

The objective of this paper is to address a binary classification problem of bank marketing data to distinguish two classes which are represented by a client choice to subscribe a term deposit. Given the problem we propose a critical evaluation of two classes of models to optimize the classification task. We will perform the comparison on a Multi-Layer Perceptron (MLP) with backpropagation algorithm and a Support Vector Machine (SVM) to train on data and perform a model selection process to choose a unique best model of each class. In summary, we shall apply data processing and sampling methods followed by a controlled evaluation process over hyperparameters space optimized with a randomized search method. The dataset properties and processing prior to the implementation of models, training and testing are discussed in Section 2. In section 3, we introduce the characteristics, advantages and disadvantages of the previous learning methods. Subsequently, we present a hypothesis given the dataset, models and task and then we discuss algorithm configurations, critical evaluation and conclusions throughout sections 5, 6, and 7.

2. Dataset Description and Analysis

The dataset selected for the experiments can be accessed in the UCI repository [2] and is related to marketing campaigns from a specific bank and each campaign is targeted to customers via phone communication based on cellular or telephone calls. The data contains 41,888 observations and 21 attributes that are distributed in 20 inputs and 1 output variable (target) responsible for a client subscription of a term deposit product. Each client has been manually assigned a target value (yes/no), which can be converted to a binary class, being 0 if a client has not subscribed and 1 for a subscription. As per description of the dataset, we removed one continuous variable (duration) due to a high correlation with the target. Thus, a duration of 0 implies there was no contact yet and therefore no subscription was made. Multicollinearity is present between emp.var.rate, nr.employed and euribor3m variables, but despite possible effects on network convergence we kept the variables to avoid an unfair comparison. The remaining variables are split in 10 categorical and 9 continuous and there is a high imbalance between the classes as showed in Table 1 (scaled). Analysis was performed to find duplicates or missing values to justify the application of appropriate techniques of cleaning or imputation, however no inconsistency was observed. Apart from the presence of outliers, we decided against removing them because SVM and MLP algorithms are supposed to be robust to noise [3], nevertheless some effect might be expected as a result of a limitation of comprehensive studies over all possible parameters and kernels. Both methods consider as input to each example a vector of real numbers, therefore it is important to encode categorical attributes. For each categorical variable we employed encoding methods aimed at ordinal

variables (e.g. month names to ordinal) and nominal variables were encoded to sparse columns differently for training and test sets to avoid data leakage.

	36548 Subscription to term deposit (No)										4640 Subscription to term deposit (Yes)									
Variable name	(0, '25%')	(0, '50%')	(0, '75%')	(0, 'count')	(0, 'freq')	(0, 'max')	(0, 'mean')	(0, 'min')	(0, 'std')	(0, 'top')	(0, 'unique')	(1, '25%')	(1, '50%')	(1, '75%')	(1, 'count')	(1, 'freq')	(1, 'max')	(1, 'mean')	(1, 'min')	(1, 'std')	(1, 'top')	(1, 'unique')
age	-0.77	-0.19	0.67	36548.0	nan	5.28	-0.01	-2.21	0.95	nan	nan	-0.87	-0.29	0.96	4640.0	nan	5.56	0.09	-2.21	1.33	nan	nan
campaign	-0.57	-0.2	0.16	36548.0	nan	19.29	0.02	-0.57	1.04	nan	nan	-0.57	-0.2	-0.2	4640.0	nan	7.38	-0.19	-0.57	0.6	nan	nan
cons.conf.idx	-0.47	-0.28	0.89	36548.0	nan	2.94	-0.02	-2.22	0.95	nan	nan	-1.23	0.02	0.95	4640.0	nan	2.94	0.16	-2.22	1.33	nan	nan
cons.price.idx	-0.86	0.59	0.72	36548.0	nan	2.06	0.05	-2.37	0.97	nan	nan	-1.18	-0.65	0.59	4640.0	nan	2.06	-0.38	-2.37	1.17	nan	nan
contact	nan	nan	nan	36548	22291	nan	nan	nan	nan	cellular	2	nan	nan	nan	4640	3853	nan	nan	nan	nan	cellular	2
day_of_week	nan	nan	nan	36548	7667	nan	nan	nan	nan	mon	5	nan	nan	nan	4640	1045	nan	nan	nan	nan	thu	5
default	nan	nan	nan	36548	28391	nan	nan	nan	nan	no	3	nan	nan	nan	4640	4197	nan	nan	nan	nan	no	2
education	nan	nan	nan	36548	10498	nan	nan	nan	nan	university.degree	8	nan	nan	nan	4640	1670	nan	nan	nan	nan	university.degree	8
emp.var.rate	-1.2	0.65	0.84	36548.0	nan	0.84	0.11	-2.22	0.95	nan	nan	-1.2	-1.2	-0.12	4640.0	nan	0.84	-0.84	-2.22	1.03	nan	nan
euribor3m	-1.28	0.71	0.77	36548.0	nan	0.82	0.11	-1.72	0.94	nan	nan	-1.6	-1.36	0.45	4640.0	nan	0.82	-0.86	-1.72	1.0	nan	nan
housing	nan	nan	nan	36548	19069	nan	nan	nan	nan	yes	3	nan	nan	nan	4640	2507	nan	nan	nan	nan	yes	3
job	nan	nan	nan	36548	9070	nan	nan	nan	nan	admin.	12	nan	nan	nan	4640	1352	nan	nan	nan	nan	admin.	12
loan	nan	nan	nan	36548	30100	nan	nan	nan	nan	no	3	nan	nan	nan	4640	3850	nan	nan	nan	nan	no	3
marital	nan	nan	nan	36548	22396	nan	nan	nan	nan	married	4	nan	nan	nan	4640	2532	nan	nan	nan	nan	married	4
month	nan	nan	nan	36548	12883	nan	nan	nan	nan	may	10	nan	nan	nan	4640	886	nan	nan	nan	nan	may	10
nr.employed	-0.94	0.4	0.85	36548.0	nan	0.85	0.13	-2.82	0.9	nan	nan	-2.07	-0.94	0.33	4640.0	nan	0.85	-1.0	-2.82	1.21	nan	nan
pdays	0.2	0.2	0.2	36548.0	nan	0.2	0.12	-5.15	0.65	nan	nan	0.2	0.2	0.2	4640.0	nan	0.2	-0.91	-5.15	2.16	nan	nan
poutcome	nan	nan	nan	36548	32422	nan	nan	nan	nan	nonexistent	3	nan	nan	nan	4640	3141	nan	nan	nan	nan	nonexistent	3
previous	-0.35	-0.35	-0.35	36548.0	nan	13.79	-0.08	-0.35	0.83	nan	nan	-0.35	-0.35	1.67	4640.0	nan	11.77	0.65	-0.35	1.74	nan	nan

Table 1: Summary statistics of features separated by each of the two classes.

Considering the imbalance class ratio of approximately 12:88 we must account for the difference and select adequate metrics to avoid inaccuracy and decreased predictive performance. Several over-sampling and under-sampling techniques were experimented to overcome the imbalance during training and validation. Studies have shown that both methods can be applied in conjunction to the minority class (over-sampling) and majority class (under-sampling) [4]. SMOTE and its variations are algorithms that generate data using the classes distribution by creating synthetic data variations from the minority class combined with k nearest neighbors, consequently we chose SMOTE as a result of its performance against over-sampling with replacement of minority class method. After encoding variables to a numerical dataset, we converted into arrays and applied a stratified split of the data into training and test sets followed by scaling on the training distribution before applying to test data. This process was only performed to generate the test set considering both models were implemented in modeling pipelines [5] to enable scalability and reproducibility of results given a dataset and random seed.

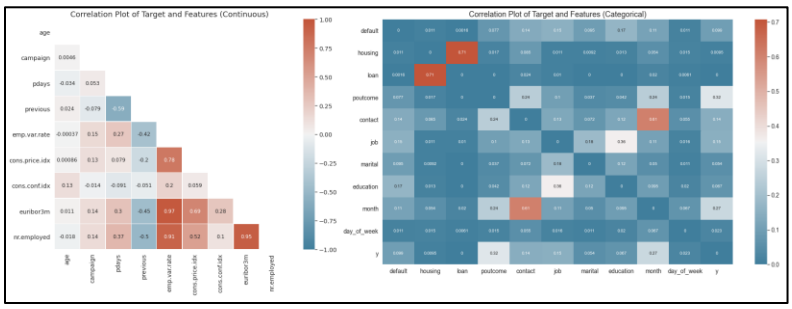

Figure 1:
Correlation plot of features and target by type.

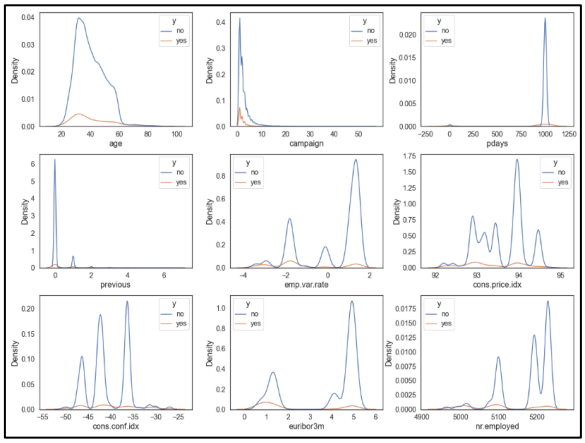

Figure 2: Continuous variable distribution.

3. Learning Methods

3.1. Multilayer Perceptron (MLP) and Backpropagation Algorithm

Multilayer Perceptron is a feedforward class of artificial neural network, which can be utilized for classification of linearly inseparable patterns and generalize and approximate functions. As per the definition MLPs have fully connected layers determined by synaptic weights [6], where each node is connected to all the nodes in the following layers, and the network is consisted of an input layer, one or more hidden layers and one output layer. Apart from the input layer each node from the hidden and output layers contains differentiable activation functions to perform transformations and add non-linearity to neural networks. Technically, the network learns a task based on the application of an optimization algorithm, as for example stochastic gradient descent (SGD), and a supervised learning technique formally defined as backpropagation algorithm for computation of gradients. To find the correct parameters for the network a gradient descent algorithm is used. The

backpropagation algorithm is divided in two phases being the first one a forward pass, which is a propagation of input signals through the network until the output generation. After that, an error function is calculated from the difference between the desired target and the network output. This error is propagated through each layer of the network in a backwards direction and the weights are modified proportionally to the contribution of each weight to the error. When the neural network has achieved the objective of learning the mapping to approximate output labels, we can predict the probability of each sample to the correspondent class. One advantage of MLP is their property of approximating any continuous functions from a compact set of real values with a unique hidden layer [7]. On the other side, the cost can increase exponentially to tune hyperparameters for more complex tasks.

3.2. Support Vector Machine (SVM)

Support Vector Machine is a non-parametric supervised learning model proposed as an application of a constrained optimization problem and was idealized by Vapnik and others [8] to find a solution to classification of patterns and regression problems. Typically, the algorithm determines one or more hyperplanes in a high dimensional space and the classifier is known to have a lower generalization error when the distance between the data points of different classes is maximized. This distance is also called as the margin of separation and the points are support vectors. Assuming a non-linear function to map the input space to a higher dimensional space, we should expect to obtain a linear classification decision boundary to separate the distinct classes, however this is not achievable in the input space but in the feature space through a kernel function, being the linear, RBF, polynomial and sigmoid the most common. Therefore, the kernel trick is defined by the calculation of inner products in higher dimensional spaces to simplify the problem into a function of data in a lower dimensional space and is one of the advantages of SVM. However, the algorithm tends to have poor performance for large datasets because of the Sequential Minimal Optimization implementation with a cubic time complexity of the training set size [9].

3. Hypothesis statement

SVM trained models perform better for higher dimensional problems and are capable to solve a convex problem and find a unique and global solution given the objective function, however this is not guaranteed for MLP. Experiments with imbalanced data showed that MLP can outperform SVM [10]. Based on the literature and considering the limitations of SVM when noise is introduced as well as the difficulty to select an appropriate kernel function and parameters we formulate the hypothesis that the best MLP model will outperform the best SVM model for classification metrics recommended to imbalanced problems. Moreover, to support our hypothesis we adopted a implementation of a novel optimization method called AdaBound [11], which combines properties of faster training and learning from adaptive methods and generalization characteristics of non-adaptive algorithms and exploits dynamic bounds on learning rate to achieve improvements in neural networks convergence.

4. Choice of training and evaluation methodology

The complete processed dataset was shuffled and divided using 80% for training and selection of best models, and 20% for the test set, that should be used for the algorithm comparison. The split used a stratified sampling strategy to maintain the imbalance of both splits and ensure the test set is a decent representation of the whole data, thus we avoid the risk of not having a minority class sample in the test set and produce inconsistent high accuracies. A pipeline was composed of scaling and a combination of SMOTE and random under-sampling, which were configured with a sampling strategy to generate only double the amount of minority and reduce the majority class until data is balanced. However, the sampling is performed over training samples exceptionally and we apply only scaling to validation set because it should replicate the unseen data. Typically, it is not feasible to predict which parameters are best for a generic task through predefined rules derived from distinct problems. Based on the literature an exhaustive grid search is suggested as a form of optimization of the search space, but more recent works as in [12] compared this method against random search and genetic algorithms. Results showed improvements in accuracy and training time for random search, particularly because it is not bound to a discrete space and if the distribution of optimal parameters is concentrated in a restricted space that can be situated between fixed set of values in the grid procedure, hence we applied a randomized search for both methods. The search was conducted with a stratified 5-fold cross-validation for a reliable estimate of evaluation with each iteration composed of numerous scoring metrics. To evaluate the test set and due to the highly skewed distribution we did not use accuracy. Thus, to evaluate the performance of classifiers we preferred to use ROC-AUC as the main metric, followed by the PR-AUC, F1-scores and

confusion matrices generated for both classes. The accuracy was partially considered and must be used with caution. Because of the accuracy paradox it can be a deceptive metric if there is an unequal number of observations in each class or if we have a multi-classification problem with more than two classes.

5. Choice of parameters and experimental results

MLP

The neural network topology was initially defined with 4 hidden layers and was further reduced to 2 hidden layers after experiments showing similar results using an identical set of hyperparameters. According to studies it can be enough to apply a unique hidden layer to approximate a function for inputs, but some works stated improved efficiency using an extra layer [13]. Each of the 40 input nodes maps one input variable to the next layers and each hidden layer is composed by an ELU activation function, which are known to be computationally less expensive and have a smoother contribution to the dead neuron phenomenon compared to ReLU [14]. Moreover, the function can converge faster because of negative values that enable shift mean activations near zero and decrease the bias [14], and avoid the vanishing gradient problem common to sigmoid. Batch normalization was applied to every hidden layer to normalize each mini-batch and to reduce the training time, however it is debatable the improvement in shallow networks. An initial set of random weights was assigned to a uniform distribution and the bias vector was set to zero. The network has 2 output nodes to represent both classes and the performance of the model is calculated with a cross-entropy loss function that internally implements a softmax activation to predict the outputs as class probabilities and cross-entropy loss that measure the separation between the predicted and actual distributions. As for the number of nodes in each hidden layer we proposed a decremental factor of 0.5 between the first and second layers, which are also used as a hyperparameter varying from 2/3 and twice the of the input nodes, a range situated close to thresholds proposed by some authors [15]. Given the flexibility of the hidden nodes which can induce slow training or overfitting we used a dropout hyperparameter that helped to reduce the effect by removing nodes in a random fashion and part of the noise from training data [16]. Some authors proved that for unbalanced problems we may introduce weights to the loss function to produce a regularization such that minority class examples would have a greater bias and better accuracy, however we did not see benefits as seen with sampling. Learning curves are important to understand the performance while training, hence underfit, overfit or data properties are commonly identified comparing training and validation losses over epochs.

SVM

Given the large training set and time complexity of SVM we proposed 2 approaches for model selection. The first requires a dimensionality reduction of training set if there is an independent component between the dynamics of learning and dimensions of the feature space. Another, which was implemented in this work as an adaptation of a work proposed in [17], contemplates training using linear kernel over the dataset and if results are not appropriate then apply a 3-fold cross-validation using a random and stratified portion of 70% from original training. The methodology employs a random search over a reduced number of 32 hyperparameters iterations due to a lower parametrization compared to MLP. We examined the classifier performance using only linear and RBF kernels, as the latter has an exponential function which is more flexible than other kernels within its function space. Moreover, poly is not suitable to random search and sigmoid generally provides worse results than RBF in most studies [18]. From the initial training subset, the choice of the best model was done using ROC score and a final training was performed over the complete training set.

For binary classification problems the ROC curve describes the ability to distinguish two classes and it tends to be a better metric than accuracy for imbalanced problems. We also evaluated precision and recall scores, useful for very imbalanced classes and a decision threshold of 0.5 is employed to output as negative or positive. However, the task has the objective to maximize prediction of subscribed clients and favor precision over recall. Therefore, we can vary this threshold and evaluate the area under the Precision-Recall curve, which is preferable to ROC [19] because it does not incorporate correctly predicted negatives and is therefore less susceptible to inflated performance for unbalanced data. Hence, we measure the PR area to find the relation between precision and recall at distinct thresholds, consequently all the positive class examples can be correctly identified without involuntarily predicting large number of negative examples as positive. If we believe a true unsubscribed client is subscribed (False Positive), he would not be contacted again, and this is more damaging than if we incorrectly predict a subscribed client as not subscribed (False Negative) as the client has already accepted the product and generated income.

6. Analysis and critical evaluation of results

Model selection

From Figure 4 we note the best MLP model obtained slightly higher scores for ROC-AUC and accuracy even with a higher initial learning rate compared to others, thus the use of number of epochs also as a hyperparameter together with dropout was useful to avoid overfitting and generalize better. Early stopping criteria was also successful to prevent overfitting. In MLP learning curves we observed that on average the model started to overfit after 200 epochs, albeit the gap between train and valid losses firstly showed signs of underfit and started to increase after 100 epochs but in opposite directions as normally is, indicating an easier to predict validation set. In Figure 3, we see that AdaBound outperformed Adam despite of a small difference, so if we consider the randomness from hyperparameters search it is logical that AdaBound could be compared alone for better results. As opposed to common belief in respect of hidden nodes we found that models with near twice the input nodes were better, which might be an indicative of a complex model for the architecture. Our dropout is interpreted contrary to theory as they represent a probability of units to be dropped and not the opposite [16]. Unsurprisingly, models with greater performance had dropout values between [0,0.5] and are aligned to the work in [20] which recommended values between [0.5,0.8], however instead of two we used a single parameter value for input and hidden layers. Except from the best model most best ROC scores in MLP had initial learning rates of less than 0.01, such exception can be explained from the capacity of the algorithm to adapt the learning. Unsurprisingly, for SVM we observe that linear kernel obtained smaller results compared to RBF and zero standard deviation based on the regularization parameter C, which for the approximate range of 10 to 400 could mean the values were not small enough to produce more misclassified points and a larger margin of separating hyperplane. As for the RBM parameters it is noticeable that a high C value combined with gamma close to 0.0003 yielded an optimal path. It is most likely that C contributed more to the result because of its influence for misclassification on the objective function, however we could not perform more experiments to prove this effectively.

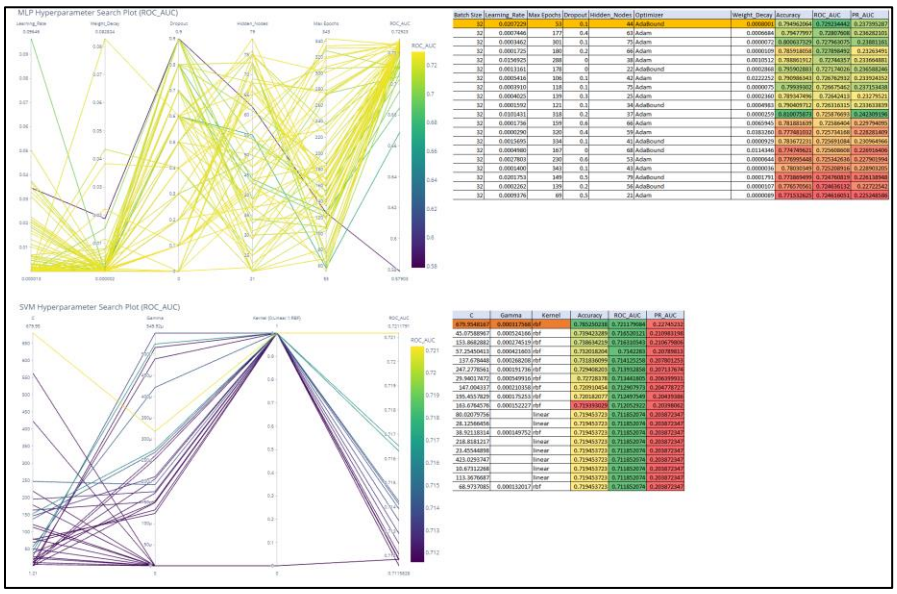

Figure 3: Hyperparameter Random Search Plots and Tables.

Algorithm comparison

Although not displayed, metrics for the best models using all training data were closer to when using test set (Figure 4 and 5) with accuracies of 76.8% vs 77.8% for MLP and 78% vs 80% for SVM, respectively. Better ROC-AUC results in test set against the training data were also observed indicating the cross-validation process was effective. In Figure 4 the ROC indicates the quality of best models at different probability thresholds, therefore both have similar areas with MLP performing slightly better than SVM (80% vs 79%) and both results are above the baseline for a random classifier. This means they are classifying well if we consider both classes equally with a probability for a randomly selected positive sample being higher than a negative. However, both are not seen as close to perfect classifiers for problems with more balanced data. A small gap is noted for F1-scores with SVM scoring better results, mostly due to a higher precision. We also calculated a F0.5-score, a variation of F1 that employs a beta parameter to give more weight to precision than recall, and SVM also outperformed MLP (35.8% vs 33.1%). The confusion matrices show that MLP model was more

accurate to classify subscription predictions with both having close to 70% of Recall (TPR), whereas for actual non-subscriptions over the sum of all the non-subscriptions (FPR), SVM outperformed MLP with 18.6% against 21.5% for MLP. When looking at the PR-AUC for test data we see that the ratio between TPR and FPR display better results for MLP which means a better average for precision vs recall overall. The confusion matrices show similar numbers for predictions for clients that subscribed but there is a large difference in favor of SVM for clients that have no subscription and the model predicted a subscription. The difference of 213 (1572-1359) of False Positives is seen as a large advantage for SVM, therefore based on the above assumptions we confirm the SVM model outperformed MLP as a more sensitive classifier to the minority class and less biased.

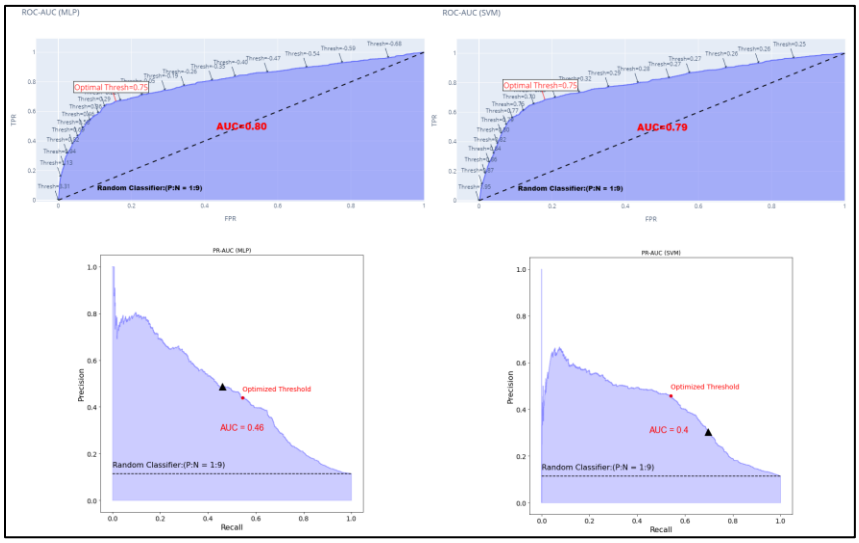

Figure 4: PR-AUC and ROC-AUC Curves (Test set).

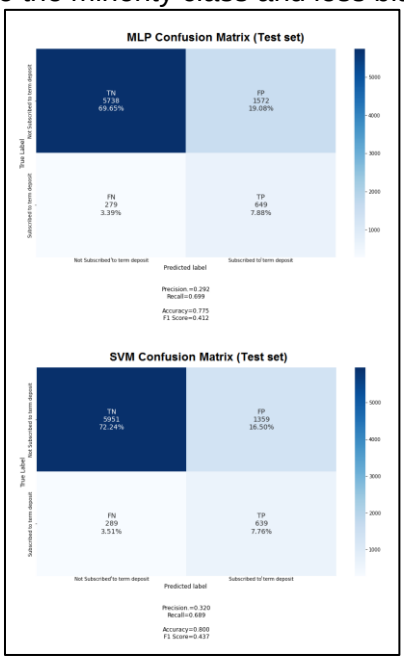

Figure 5: Confusion Matrices.

7. Conclusions, lessons learned, references and future work

As anticipated, both models had similar performance and achieved a high accuracy as expected if predictions were done entirely based on the majority class, but with better prediction for the rare class. The MLP model selection was 3 times faster in training compared to SVM, despite the use of GPU device in opposition to SVM models, which proved to be computationally more expensive even with adaptations to training methodology. Application of nested cross-validation approaches [21] using the entire dataset can be beneficial to avoid overfitting and optimistic bias from standard cross-validation, with external loops responsible for the hyperparameters optimization for model selection and internal loops manage each model and apply the condition to the entire training data. Furthermore, we would suggest exploration of approaches to control the classification error of minority class, such as mixed-ensemble models [22], which considers a boosting algorithm using weights or a combination of outputs from other algorithms.

Methods of dimensionality reduction such as PCA [23] are useful before adding noise for MLP to select features and generalize better if high correlation is present, nonetheless it can be harmful if feature and targets correlation is not considered. Moreover, unsupervised methods as for example SOM [24] can be utilized to produce a better representation of a dataset and provide robust predictions. For future works we would be encouraged to explore other approaches for SVM implementation with subsets of data and kernel approximations using Fourier or Nystrom methods to minimize the effect of training time, which should be considered as part of reporting results despite of classification metrics used in the algorithm comparison.

8. 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, Jun. 2014, doi: 10.1016/j.dss.2014.03.001.

[2] “UCI Machine Learning Repository: Bank Marketing Data Set.” https://archive.ics.uci.edu/ml/datasets/bank+marketing (accessed Feb. 19, 2021).

[3] J. Hoak, “The Effects of Outliers on Support Vector Machines,” p. 9.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.

[5] P. Sugimura and F. Hartl, “Building a Reproducible Machine Learning Pipeline,” p. 4.

[6] S. S. Haykin and S. S. Haykin, *Neural networks and learning machines*, 3rd ed. New York: Prentice Hall, 2009.

[7] “THE APPROXIMATION OF CONTINUOUS FUNCTIONS BY MULTILAYER PERCEPTRONS - IAN GLOVER.pdf.” .

[8] “A Training Algorithm for Optimal Margin Classifiers - Bernhard Boser.pdf.” .

[9] “Large data sets classification using convex–concave hull.pdf.” .

[10] J. Ren, “ANN vs. SVM: Which one performs better in classification of MCCs in mammogram imaging,” *Knowl.-Based Syst.*, vol. 26, pp. 144–153, Feb. 2012, doi: 10.1016/j.knosys.2011.07.016.

[11] L. Luo, Y. Xiong, Y. Liu, and X. Sun, “ADAPTIVE GRADIENT METHODS WITH DYNAMIC BOUND OF LEARNING RATE,” p. 21, 2019.

[12] P. Liashchynskyi and P. Liashchynskyi, “Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS,” *ArXiv191206059 Cs Stat*, Dec. 2019, Accessed: Apr. 08, 2021. [Online]. Available: http://arxiv.org/abs/1912.06059.

[13] V. Korkov and C. Sciences, “Kolmogorov’s Theorem and Multilayer Neural Networks,” p. 6.

[14] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs),” *ArXiv151107289 Cs*, Feb. 2016, Accessed: Apr. 05, 2021. [Online]. Available: http://arxiv.org/abs/1511.07289.

[15] “The Number of Hidden Layers | Heaton Research.” https://www.heatonresearch.com/2017/06/01/hidden-layers.html (accessed Apr. 09, 2021).

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” p. 30.

[17] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, “A Practical Guide to Support Vector Classification,” p. 16.

[18] H.-T. Lin and C.-J. Lin, “A Study on Sigmoid Kernels for SVM and the Training of non-PSD Kernels by SMO-type Methods,” p. 32.

[19] T. Saito and M. Rehmsmeier, “The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets,” *PLOS ONE*, vol. 10, no. 3, p. e0118432, Mar. 2015, doi: 10.1371/journal.pone.0118432.

[20] C. Garbin, “Dropout vs. batch normalization: an empirical study of their impact to deep learning,” *Multimed. Tools Appl.*, p. 39, 2020.

[21] G. C. Cawley and N. L. C. Talbot, “On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation,” p. 29.

[22] L. Turgeman and J. H. May, “A mixed-ensemble model for hospital readmission,” *Artif. Intell. Med.*, vol. 72, pp. 72–82, Sep. 2016, doi: 10.1016/j.artmed.2016.08.005.

[23] I. B. V. da Silva and P. J. L. Adeodato, “PCA and Gaussian noise in MLP neural network training improve generalization in problems with small and unbalanced data sets,” in *The 2011 International Joint Conference on Neural Networks*, San Jose, CA, USA, Jul. 2011, pp. 2664–2669, doi: 10.1109/IJCNN.2011.6033567.

[24] F. D. Mwale, A. J. Adeloye, and R. Rustum, “Application of self-organising maps and multi-layer perceptron-artificial neural networks for streamflow and water level forecasting in data-poor catchments: the case of the Lower Shire floodplain, Malawi,” *Hydrol. Res.*, vol. 45, no. 6, pp. 838–854, Dec. 2014, doi: 10.2166/nh.2014.168.

Appendix 1 – glossary
Include definitions of all technical terms such as *perceptron*, *softmax*, *Bayesian regulation*, *generalizability*, etc.

- Accuracy: Ratio of correctly predicted instances over all input samples used for evaluation.
- Batch size: Number of training examples in one forward/backward pass.
- Customer/Client: Terms used interchangeably in this paper for convenience.
- C: controls the tradeoff between classification of training points accurately and a smooth decision boundary.
- Confusion matrix: Considering k response classes it is a matrix of k x k dimensions with quantities of correct and incorrect predictions for true known values and is considered a measure of classifiers performance.
- Cross-validation: **T**echnique to evaluate predictive models by partitioning the original sample into a training set to train the model and a validation set to evaluate it.
- Dropout: Technique to randomly omit neurons in a neural network.
- Early Stopping: Early stopping is a form of regularization used to avoid overfitting when training a learner.
- ELU: Exponential Linear Unit activation function.
- Gamma: A hyperparameter in SVM which controls the value of curvature in a decision boundary (e.g. a high gamma value characterizes more curvature, therefore only near points have high influence).
- F1-Score: Harmonic mean of the precision and recall.
- FN: False Negatives.
- FP: False Positives.
- FPR: False Positive Rate.
- Kernel: Mathematical function to transform input data into a required format.
- Imbalanced classes: Unequal distribution of classes with a common domination of the majority class examples over the minority class. It can lead to classification errors because most machine learning algorithms are designed with the assumption of an equal number of examples for each class.
- Multicollinearity: When more than two explanatory variables are highly correlated linearly.
- Perceptron: Represents a single layer neural network.
- Precision: Number of positive class predictions that belong to the positive class.
- PR-AUC: It is a curve that combines precision and recall in a single visualization.
- Randomized Search: Technique where random combinations of the hyperparameters are used to find the best solution for the built model.
- Recall: Number of positive class predictions made from all positive examples in the dataset.
- ReLU: Rectified Linear Unit activation function.
- ROC- AUC: Curve plot of sensitivity (TPR) versus 1-specificity (FPR) considering all possible values of thresholds.
- SOM: Self-Organizing Maps.
- Softmax: Softmax is a mathematical function that converts a vector of numbers into a vector of probabilities.
- TN: True Negatives.
- TP: True Positives.
- TPR: True Positive Rate.

Appendix 2 – Implementation details

Include any relevant implementation details such as negative results, changes made which improved results, training regimes and how to train the models to reproduce results.

Initially we defined a range of 64 hyperparameter space and we further reduced to 32 to SVM only as a response of the dataset size. However, if we investigate the parameter combination for both models it is enough to confirm that the MLP space account for a large variety of possible parameters. We conducted experiments with oversampling with SMOTE and variants SMOTEEN, ADASYN, and SMOTE with Tomek Links, but results proved to be similar or even worse and training process was longer and we decided for a combination between oversampling and undersampling after that. Experiments were also done using class weights for MLP loss function and weighted samples to emphasize the choice of harder training samples, but no improvement was seen, and we only used balanced weights for SVM. However, if not limited by time constraints we thought it would be fundamental to try to set weights as prior hyperparameters in a grid search before applying hyperparameter optimization to the remaining variables.

Fortunately, we had access to use GPU for training with MLP models during the final stages of the work, however training with SVM proved to be a challenge and changes given the dataset size, complexity of the model and choice of hyperparameters space and iterations. Therefore, we decided to scale down the number of iterations as well as to select a smaller dataset. However, for both final models we used the entire training data to provide similar environment conditions for the algorithms before applying to the test set.

Reproducibility:

To reproduce the results achieved it is highly recommended to train the MLP model using a GPU. Training times reduced from more than 10 hours using CPU cores to fewer than 4 hours when a Tesla P100 GPU was used.

We created a function in the Models Reproducibility (Section 7) of the INM427_Train_Coursework.ipynb file called pre_process and other functions responsible for the hyperparameter optimization (model selection) and final training (best model) for both models.

First, we should run the uncommented functions under section 7. After that, we uncomment the last cells from the file depending on which model we would like to train. Each cell will run the pre_process function and after that MMM_train_and_model_selection and MMM_train_best_model where MMM can be mlp or svm, depending on the training. With regards to the model selection we output the results from each algorithm in a csv file and for the best model generation the output is a pkl file that contains the models parameters for testing.