School of Computer Science

Data Mining in Fulfilment of DATA9910

Maksymilian Drzezdzon

C15311966

Degree: TU060/1

Module Coordinator: Brendan Tierney

Declaration of Ownership: I declare that the attached work is entirely my own and that all sources have been acknowledged: ☑

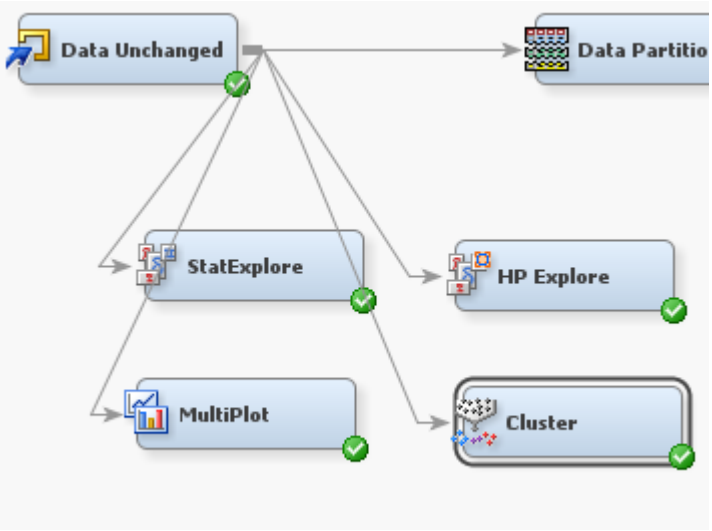
Date: 2020/11/09

Definition of problem - Business Understanding

<u>The purpose of this data mining project is to identify customers who are most likely to subscribe to a term deposit account based on previous marketing campaigns.</u>

This project aims to employ the CRISP-DM (cross industry standard process for data mining) methodology. The dataset that is being worked on consists of 41188 rows and 20 columns.
The motivation for such a model is to reduce cost and identify characteristics for people that are more likely to make purchases.

Data Exploration/Understanding



Data was explored using the following setup.

Some of the results are appended below.

Statistics Table

Label	Scale	Missing	Percent Missing	Non Missing	Minimum	Mean	Maximum	Standard Deviation	Skewness	Kurtosis	Coefficient of Variation
age	VAR	0	0	45211	18	40.93621	95	10.61876	0.684818	0.31957	0.259398
balance	VAR	0	0	45211	-8019	1362.272	102127	3044.766	8.360308	140.7515	2.235064
campaign	VAR	0	0	45211	1	2.763841	63	3.098021	4.89865	39.24965	1.120912
contact	CLASS	0	0	45211	.	.	.	.	.	.	.
day	VAR	0	0	45211	1	15.80642	31	8.322476	0.093079	-1.0599	0.526525
default	CLASS	0	0	45211	.	.	.	.	.	.	.
duration	VAR	0	0	45211	0	258.1631	4918	257.5278	3.144318	18.15392	0.997539
education	CLASS	0	0	45211	.	.	.	.	.	.	.
housing	CLASS	0	0	45211	.	.	.	.	.	.	.
job	CLASS	0	0	45211	.	.	.	.	.	.	.
loan	CLASS	0	0	45211	.	.	.	.	.	.	.
marital	CLASS	0	0	45211	.	.	.	.	.	.	.
month	CLASS	0	0	45211	.	.	.	.	.	.	.
pdays	VAR	0	0	45211	-1	40.19783	871	100.1287	2.615715	6.935195	2.490899
poutcome	CLASS	0	0	45211	.	.	.	.	.	.	.
previous	VAR	0	0	45211	0	0.580323	275	2.303441	41.84645	4506.861	3.969237
y	CLASS	0	0	45211	.	.	.	.	.	.	.

Table 1: Summary statistics for bank dataset

Not a great deal of info can be drawn from some of these nodes, Table 1 is added as a demonstration of the outputs received.

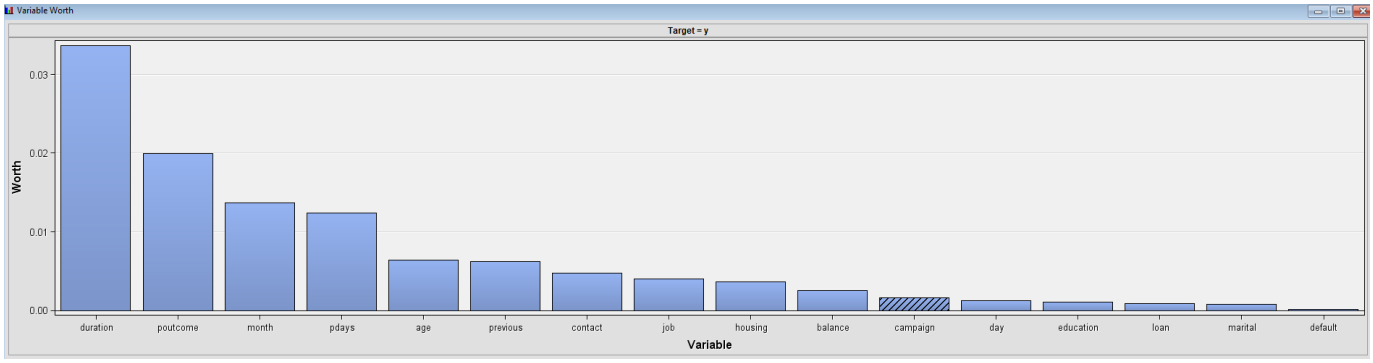
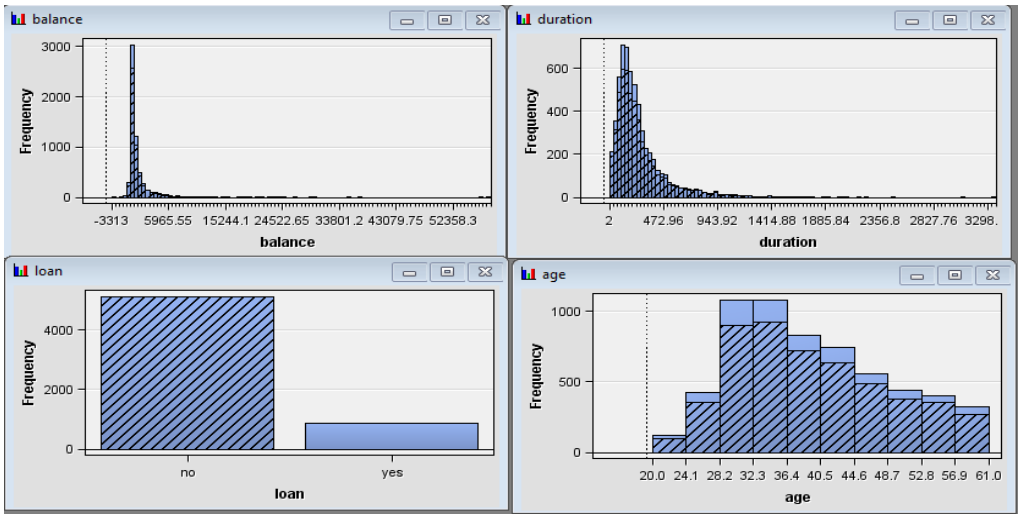
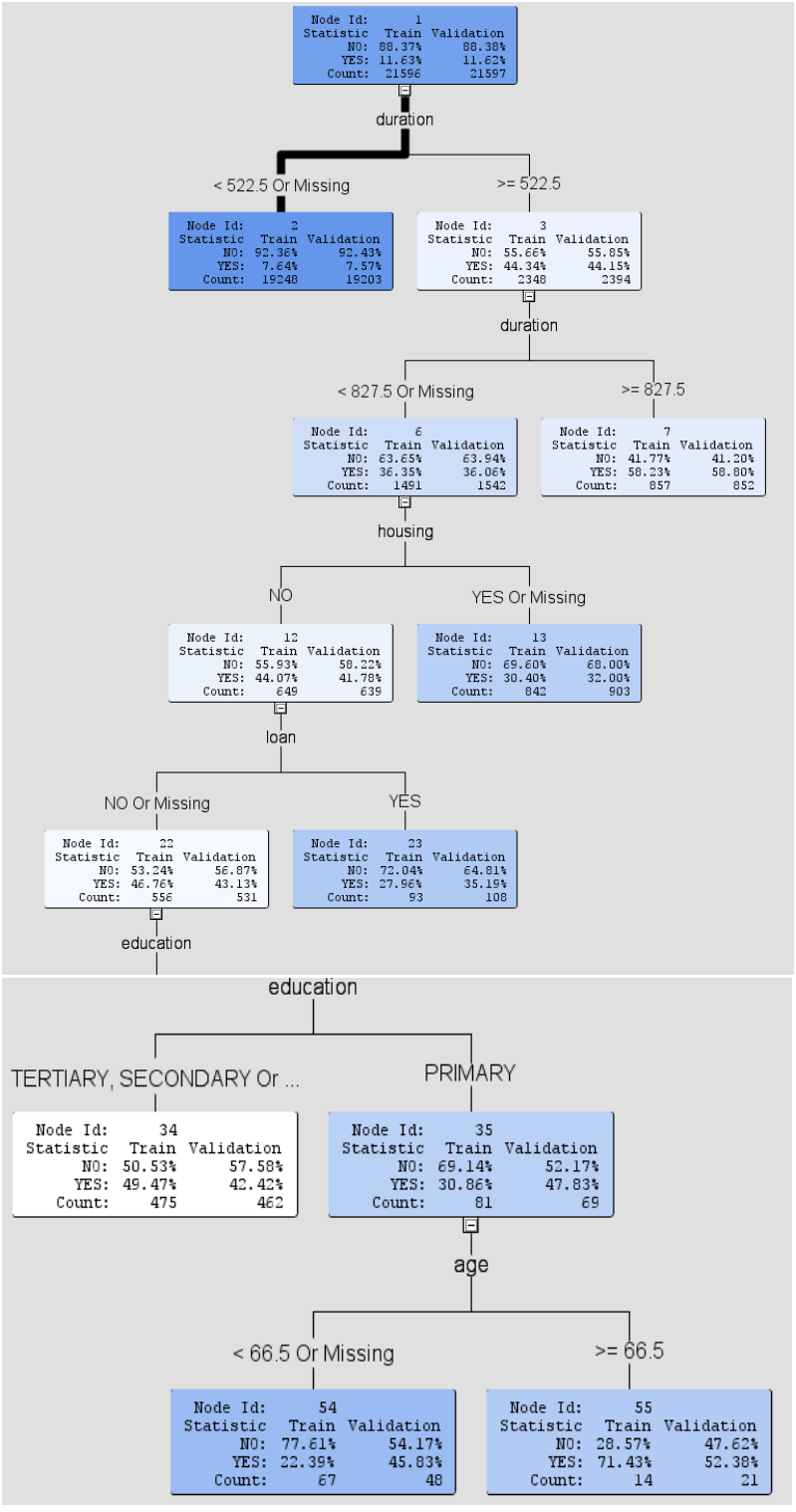


Figure 1: campaign predictors the end with a customer subscribing to a long-term deposit account

This is an interesting graph; the target y is whether a customer subscribed to a term deposit account. The highest predictors are in descending order. Poutcome is the outcome of the previous campaign and pday is the number of days that have passed since a potential client was last contacted.



You can see the representation of all other participants that don't have a loan across the age, balance and duration bar charts. Another augmentation that was made to what SaS already provides was the target variable **Y** was renamed to "subscribed_term_deposit" as its more descriptive in the R code and all yes or no values were converted to 1 or 0 as neural networks perform better on binary data. Bin sizes were increased for duration and balance to get a better read for the representation of clumped up values. After that the data was passed through a decision tree for more analysis and the results are



The thicker the line the more data has been passed in that direction/leaf. The lighter the shade the more impure/bad the data is at predicting the target variable, in this case it being **Y** which is whether the customer subscribed to a term deposit account. According to the decision tree, the best predicators are a person doesn't have a house, doesn't have a loan has a higher education status of above primary school, with longer conversations not materialising into good costumers as one would expect. This is indicated by lighter shades of blue across the graph.

As depicted in Figure 1 duration is one of the highest predictors, however a large portion of it is void and discarded by the decision tree.

Data Preparations

First data was explored in R and SaS enterprise miner. Values that can be dropped are **poutcome** because it's all unknown, **previous** all values are 0, **pdays** are all -1, **months** isn't needed unless one wishes to see if there is a seasonal trend in sales, **contact** is mostly unknown, this was done with the code below.

```
df = read.table("bank/bank-full.csv", header = T, sep = ";")

drop = c("poutcome", "previous", "pdays", "months", "contact")

df = df[,!(names(df) %in% drop)]

unique(df$job , incomparables = FALSE)

unique(df$education , incomparables = FALSE)

names(df)[names(df) == "y"] = "subscribed_term_deposit"

df$subscribed_term_deposit[df$subscribed_term_deposit == "no"] = 0

df$subscribed_term_deposit[df$subscribed_term_deposit == "yes"] = 1

df$job[df$job == "unknown"] = NA

df$education[df$education == "unknown"] = NA

df = na.omit(df)

write.csv(df, "bank-full-updated.csv", row.names = F)
```

This decision which variables to keep was made by selecting all variables in SaS and clicking explore which generates visualisations, when you select a column all other graphs reflect this action and show the representation of that column in other graphs, **no** in the loan bar chart was selected.

Data Mining Models Used - Modelling

Data Models Setup:

Data Node – Holds Dataset.

Data Unchanged – Holds the original Dataset with 0 changes apart from setting the y variable as target and type as binary for data mining models.

Data Partition Node – Holds 50% of data for training and 50% for test.

Import Data Node – Renamed to Data and changed variable y to **binary** type from nominal and set it as the target., likewise for the unmodified dataset.

Model Comparison Node – Selection Statistic is set to Misclassification Rate with the HP selection Statistic also being set to the same as all models are configured this way unless specified in the LIFT or AUC-ROC sections.

These are the settings used for all evaluations unless specified otherwise in said evaluation.

Logistic Regression

Configuration used: All defaults except in the model selection the selection model was changed to stepwise, the default setting is logistic regression, linear regression has to be configured in the class targets.

This was done to have SaS enterprise miner determine the most useful inputs for the best result. [1] Another change was in the same table the selection criterion was set to validation misclassification similar to the decision tree above to get the best result possible.

Model workflow used: Usually, the workflow for a regression model would look like this Data Node – Replacement Node – Data Partition Node – Impute Node – Regression Node. This process was not followed in this case because there is no data to be replaced, instead the workflow was Data Node – Data Partition Node – Regression Node.

Logistic regression is used to explain the relationship between a dependant binary variable, in this case a yes/no – 1/0 value and one or more other variables.

Neural Network

Configuration used: All defaults are in place; in the Train table the model selection criterion is set to misclassification to pick out the best model.

Model workflow used: Data Node – Data Partition Node – Neural Network Node – Model Comparison Node.

Neural networks create a net of interconnected nodes that try to look for patterns of variable depth, meaning there could be one, two, five or ten columns of nodes that data will pass through before a correct pattern is found, this is done on a yes/no basis, where each correct correlation lights up a node with then moves to the next layer of nodes that do the same thing until a path is found from a to z

Decision Tree

Configuration used: Selection statistic is set to Misclassification Rate. All other fields are left as they were.

Model workflow used: Data Node – Data Partition Node – Decision Tree Node – Model Comparison Node.

A decision tree is a diagram that uses probability for one outcome or another to occur, this then splits into branches until similar to the neural net the end is reached, however unlike the neural net, a decision tree is more visual and can be interpreted by a person whereas a neural net cannot, there is no way to see how the neural net came to said conclusion/result. Models that are interpretable by humans are called white-box models. [3]

Gradient boost

Configuration used: Selection statistic is set to Misclassification Rate. All other fields are left as they were.

Model workflow used: Data Node – Data Partition Node – Gradient Boost Node – Model Comparison Node.

Gradient boosting combines the results of a model with the previous one in order to minimize the error when making predictions. [4]

HP SVM

Configuration used: Everything is left in default as it was created.

Model workflow used: Data Node – Data Partition Node – HP SVM Node – Model Comparison Node.

Support Vector Machine is a statistical technique that splits data into two groups and then uses a line (a hyperplane) between these two groups, then it calculates the distance between points and the line moving it in order to adjust the distance until the maximum distance/optimal distance is found. This is done to create as big of a distinction between two groups as possible allowing for better classification. [5]

Auto Neural Network

Configuration used: Selection statistic is set to Misclassification Rate. All other fields are left as they were. 5 minutes were allowed for training.

Model workflow used: Data Node – Data Partition Node – Auto Neural Network Node – Model Comparison Node.

This model was removed from later testing as its underperformance and time needed to train was becoming an inconvenience when tweaking other models with a model comparison node

MBR

Configuration used: Selection statistic is set to Misclassification Rate. All other fields are left as they were.

Model workflow used: Data Node – Data Partition Node – MBR Node – Model Comparison Node.

Memory Based Reasoning uses known data to then ‘label’ new instances and finds neighbours similar to the new data, this is then used for prediction and classification.

Model Performance – Evaluation

Comparing Models

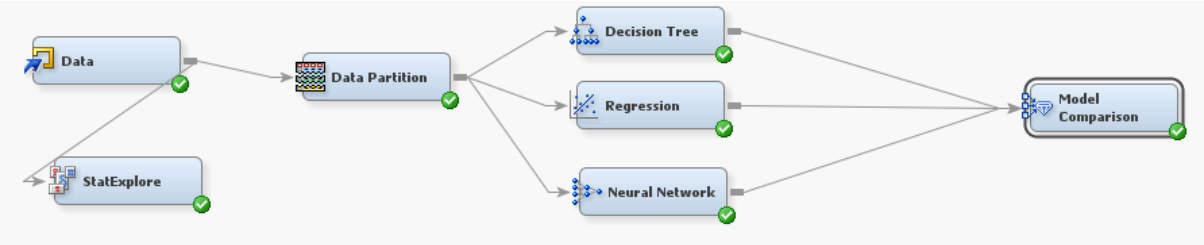
For testing purposes, a modified dataset was used along with an identical but unmodified dataset.

Configuration: All settings left on default except just like before selection statistic is set to misclassification rate so the comparison node will also have the same setting and it will use the validation data in the selection table to compare the results of each model.

The data type of the target variable y was changed to binary to be compatible with the SVM algorithm.

This general layout was used for both modified and unmodified datasets, all algorithms listed in the results appended below were attached in a similar fashion and run concurrently to be compared in the model comparison node.

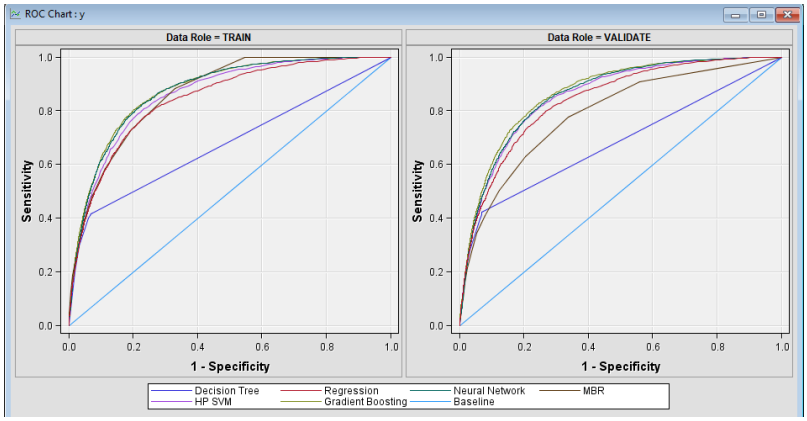
An educated guess before these experiments are run is that the neural network should perform the best because of how many free parameter its has when finding patterns [6], albeit it being a black box model, it wouldn't be preferred for deployment especially if a white box model performs close enough to it.



This is workflow for how models were assessed. A Data Node – Data Partition Node – Model being compared connected to the Model Comparison Node.

Models that overlap with the ones used in the original research paper are marked with a blue dot.

Results for modified bank dataset.



Interpreting ROC curve, the specificity x-axis is the sample size that would fall under which level of sensitivity y-axis which is the percent chance that the reading will be correct.

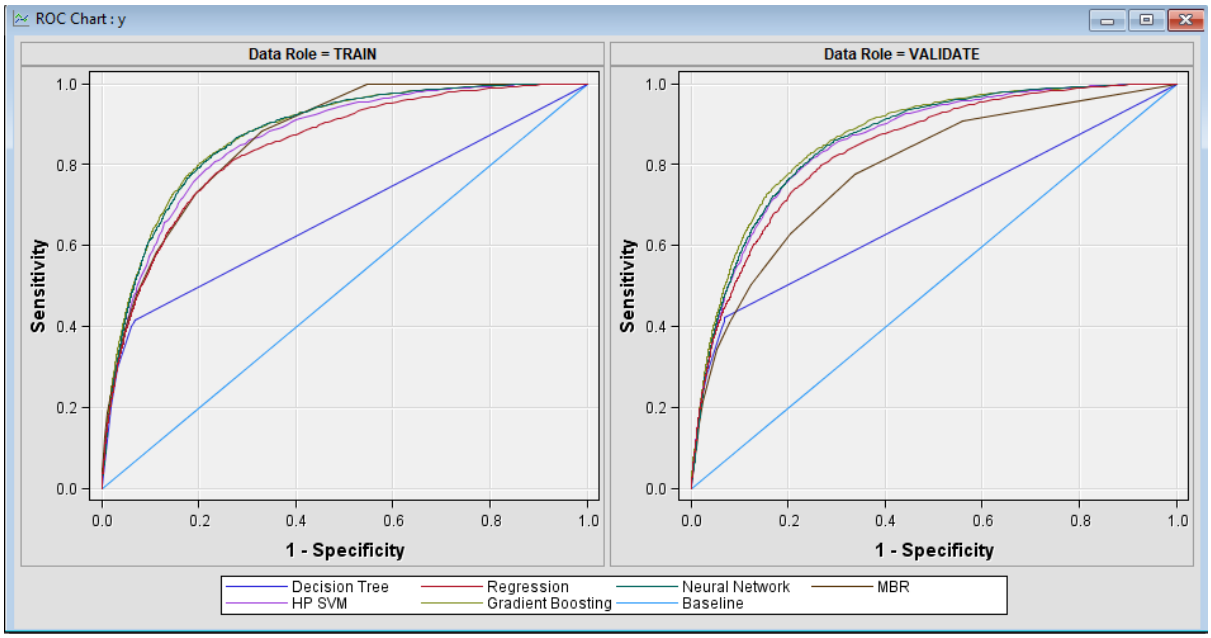
E.G a sample size of 0.4 (40%) would cover a little above 0.60 (60%) for a decision tree.

Assessed using Misclassification rate.

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Misclassification Rate
Y	Boost	Boost	Gradient Bo...	y		0.108904
	Tree	Tree	Decision Tr...	y	●	0.109228
	Reg	Reg	Regression	y	●	0.109367
	Neural	Neural	Neural Net...	y	●	0.110062
	AutoNeural	AutoNeural	AutoNeural	y		0.111867
	MBR	MBR	MBR	y		0.113581
	HPSVM	HPSVM	HP SVM	y	●	0.114692

The best performing algorithm is the gradient boost and from the algorithms used in the research paper there is a close result between the decision tree and linear regression models of 0.1092 and 0.1093. According to these results my original hypothesis is wrong, the neural network did not perform well in comparison to the other models.

With a ROC curve of

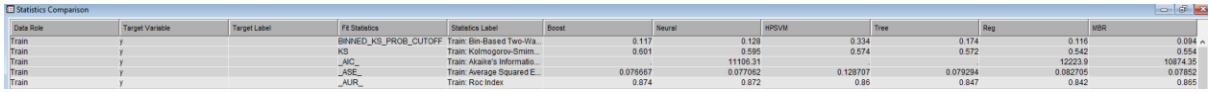


Assessed using LIFT.

An adjustment made for this run of the models was that the subtree assessment was set to LIFT from misclassification, this greatly increased the performance of the decision subtree model.

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Lift
Selected Model						
Y	Boost	Boost	Gradient Bo...	y		3.784335
	Neural	Neural	Neural Net...	y		3.760434
	HPSVM	HPSVM	HP SVM	y		3.561258
	Tree	Tree	Decision Tr...	y		3.502405
	AutoNeural	AutoNeural	AutoNeural	y		3.44972
	Reg	Reg	Regression	y		3.441753
	MBR	MBR	MBR	y		2.986755

With appropriate ROC indexes for train and test below.

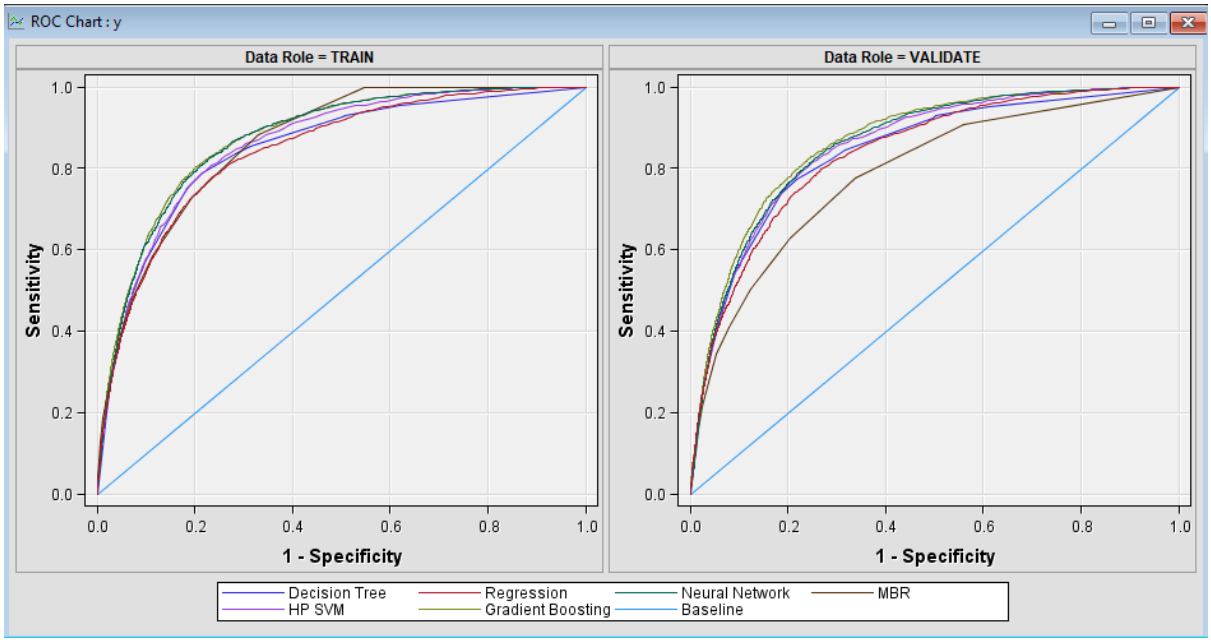


With a close up, you can view the original by zooming in to see it's the same table.

Boost	Neural	HPSVM	Tree	Reg	MBR
0.117	0.128	0.334	0.174	0.116	0.094
0.601	0.595	0.574	0.572	0.542	0.554
.	11106.31	.	.	12223.9	10874.35
0.076667	0.077062	0.128707	0.079294	0.082705	0.07852
0.874	0.872	0.86	0.847	0.842	0.865

Metric	LR	DT	SVM	NN
ALIFT	0.842	0.847	0.86	0.872

With a ROC curve of

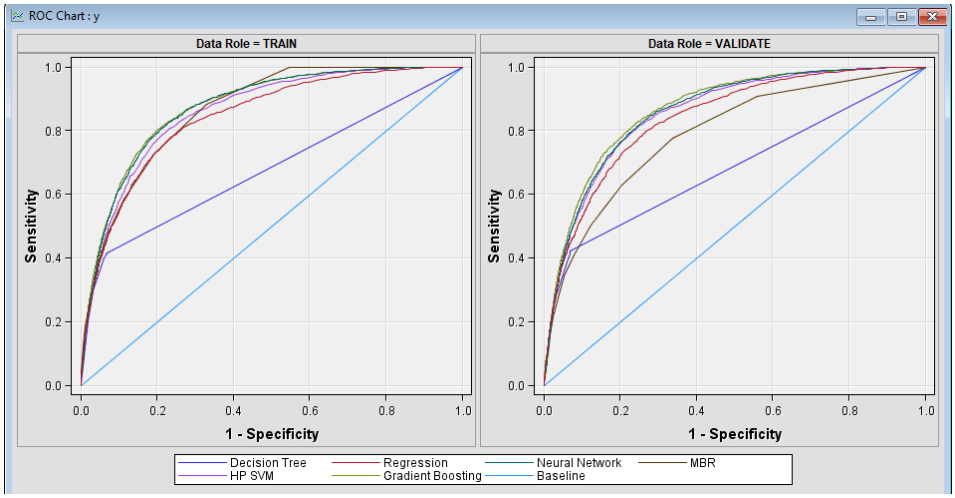


Assessed using AUC-ROC.

The model selection statistic was changed to ROC, however fort the HP Selection Statistic for the HP SVM model no such option exists; therefore, it was left as misclassification rate.

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Roc Index
Y	Boost	Boost	Gradient Bo...	y		0.867
	Neural	Neural	Neural Net...	y	●	0.859
	HPSVM	HPSVM	HP SVM	y	●	0.855
	AutoNeural	AutoNeural	AutoNeural	y		0.85
	Reg	Reg	Regression	y	●	0.838
	MBR	MBR	MBR	y		0.787
	Tree	Tree	Decision Tr...	y	●	0.679

With a ROC curve of



Results for the unmodified bank dataset.

Assessed using Misclassification rate.

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Misclassification Rate
Y	Neural	Neural	Neural Net...	y	●	0.094227
	Tree	Tree	Decision Tr...	y	●	0.095643
	Boost	Boost	Gradient Bo...	y		0.09697
	Reg	Reg	Regression	y	●	0.097501
	HPSVM	HPSVM	HP SVM	y	●	0.10104
	MBR	MBR	MBR	y		0.113294
	AutoNeural	AutoNeural	AutoNeural	y		0.119973

In this case the neural network did perform the best with a score of 0.0942 with the decision tree model behind it with 0.0956.

Assessed using LIFT.

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Lift
Y	HPSVM	HPSVM	HP SVM	y	●	4.615242
	Reg	Reg	Regression	y	●	4.532016
	Neural	Neural	Neural Net...	y	●	4.471488
	Boost	Boost	Gradient Bo...	y		4.439333
	Tree	Tree	Decision Tr...	y	●	4.192626
	AutoNeural	AutoNeural	AutoNeural	y		3.866211
	MBR	MBR	MBR	y		3.480199

Assessed using AUC-ROC.

The model selection statistic was changed to ROC, however fort the HP Selection Statistic for the HP SVM model no such option exists; therefore, it was left as misclassification rate.

AUC - ROC is a performance measurement for classification problems. ROC is a probability curve and AUC is the degree of separability. [2]

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Roc Index
Y	Neural	Neural	Neural Net...	y	●	0.92
	Boost	Boost	Gradient Bo...	y		0.913
	Reg	Reg	Regression	y	●	0.907
	HPSVM	HPSVM	HP SVM	y	●	0.906
	AutoNeural	AutoNeural	AutoNeural	y		0.889
	Tree	Tree	Decision Tr...	y	●	0.813
	MBR	MBR	MBR	y		0.812

When assessing these models on the unchanged data.

According to these results the best preforming models are: (if an algorithm used in the paper is not first in a category then the first model from the paper on the results table will be added for the sake of comparison with its place on the list)

LIFT is the ratio of target response divided by average response, the higher the resulting number the more times the average response divides the response.

AUC-ROC is a value t hat ranges from 0 to 1, 0.5 results are agnostic, with 0.7-0.8 is considered acceptable, 0.8-0.9 is considered excellent, anything above 0.9 is considered outstanding. [3]

Modified Dataset:

Misclassification – Gradient Boost misclassification of 10.8% with the Decision Tree misclassification of 10.9% on 2nd place.

LIFT - Gradient Boost with a ratio of 3.78 and the Neural Network on 2nd place with 3.76.

AUC-ROC – Gradient Boost with a value of 0.867 and the Neural Network on 2nd place with a value of 0.859.

Unmodified Dataset:

Misclassification – Neural Network misclassification of 9.4% and Decision Tree misclassification of 9.56%.

LIFT – Support Vector Machine with a ratio of 4.61, Logistic Regression with a ratio of 4.53 and Neural Network with a ratio of 4.47, all of these models preform very close to each other.

AUC-ROC – Neural Network with a value of 0.92 and Logistic Regression with 0.907 on 3rd place.

For deployment white box models would be preferred over black box models, especially when the results are close. This is because white box models are easier to interpret such as a decision tree provides visualisation whereas a neural net does not, it just spits out a result.

As tempting as it may be to use the unmodified dataset models because the above results suggest better model performance, it seems like they might have been over fitted on the data that was used, for that reason the ROC curves were omitted for the unmodified dataset and for the sake of saving some space and keeping the report shorter.

Models that will be used for comparison will be the ones trained on the modified dataset.

How Did Results Compare To Original Research?

In the original paper titled A Data-Driven Approach to Predict the Success of Bank Telemarketing, logistic regression, decision trees, neural networks and support vector machine were used. Note original findings

Discussion of how your results compared to the research paper and any conclusions that you can draw from this comparison

Original findings, final results for AUC and ALIFT

Metric	LR	DT	SVM	NN
AUC	0.715	0.757	0.767	**0.794**
ALFIT	0.626	0.651	0.656	**0.672**

Results for this project

ALIFT was taken from the **_AUR_** fit statistic, when results pop up from the model comparison node, simply click view -> model -> statistics comparison. Under the AUR column in fit statistics you can see the results that were taken to compare LIFT using the ALIFT metric.

Metric	LR	DT	SVM	NN
ROC-AUC	0.838	0.679	0.855	**0.859**
LIFT	3.44	3.50	3.56	**3.76**
ALIFT	0.842	0.847	0.86	**0.872**

As can be seen above my decision tree implementation is underperforming when compared to the original research paper, however it seems al other models seem to be over preforming. Which could mean a very good model or an overfit. Preforming A/B testing in a production environment would a good test to see how these models actually preform.

The performance gains my models have on the academic models could stem from an inexperienced analysis meaning that my results are overfitted as compared to the ones from the research paper. The approach I took was also different from the one taken in the paper.

I modified my dataset and removed what felt like redundant variables, this wasn't done in the paper. Like mentioned above I'd like to test such a model on users by deploying it to a real-world environment in order to monitor its performance.

Overall, I found this to be a good introduction to testing models and researching how to use LIFT in a business environment how much do I need to invest in a campaign (sample size) in order to get as much profit for my investment (the ratio that ranges from 0-1).

Much like in the original research the neural network preformed the best, however like mentioned many times throughout the project, a Whitebox model would be preferred when deploying to production as its easier to monitor and interpret.

Appendix

https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5 [2]

https://blog.dataiku.com/white-box-vs-black-box-models-balancing-interpretability-and-accuracy#:~:text=On%20the%20other%20hand%2C%20white,accuracy%2C%20but%20higher%20explainability [3]

https://www.siliconrepublic.com/enterprise/white-box-machine-learning

https://www.datascience-pm.com/crisp-dm-2/

https://www.youtube.com/watch?v=meZ5qhr3nV0&ab_channel=RapidMiner%2CInc.

https://communities.sas.com/t5/SAS-Data-Mining-and-Machine/AUC-value-Area-Under-Curve-or-ROC-Index-in-SAS-Miner-9-3/td-p/263271

https://documentation.sas.com/?docsetId=emref&docsetTarget=n1fevkin0iu4cxn1khv8ow9r8pmj.htm&docsetVersion=14.3&locale=en#:~:text=Memory%2Dbased%20reasoning%20is%20a,to%20categorize%20or%20predict%20observations.

https://www.youtube.com/watch?v=tSg5W4vR6Bg&ab_channel=EasyEngineeringClasses

References

https://documentation.sas.com/?docsetId=emref&docsetTarget=n1jqzz8cssr9m2n1ktx2iyv87q56.htm&docsetVersion=14.3&locale=en#n1m5w9deopojaqn1jykkfp0x5fcy [1]

Jayawant N. Mandrekar, Receiver Operating Characteristic Curve in Diagnostic Test Assessment, Journal of Thoracic Oncology, September 2010 20https://www.sciencedirect.com/science/article/pii/S1556086415306043#:~:text=AREA%20UNDER%20THE%20ROC%20CURVE,-AUC%20is%20an&text=In%20general%2C%20an%20AUC%20of,than%200.9%20is%20considered%20outstanding. [3]

Jake Hoare, https://www.displayr.com/gradient-boosting-the-coolest-kid-on-the-machine-learning-block/ [4]

Bruno Stecanella, June 22, 2017 https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/ [5]

https://stackoverflow.com/questions/38595451/why-do-neural-networks-work-so-well#:~:text=Neural%20Networks%20can%20have%20a,are%20too%20simple%20to%20fit.&text=The%20input%20to%20a%20NN,output%20hidden%20inside%20of%20it. [6]