Original Research Article

A Stacking Approach to Direct Marketing Response Modeling

***Abstract* –In this work, we investigate the viability of the stacked generalization approach in predictive modeling of a direct marketing problem. We compare the performance of individual models created using different classification algorithms, and stacked ensembles of these models. The base algorithms we investigate and use to create stacked models are Neural Networks, Logistic Regression, Support Vector Machines (SVM), Naïve Bayes and Decision Tree (CART). These algorithms were selected for their popularity and good performance on similar tasks in previous studies. Using a benchmark experiment and statistical tests, we compared five single algorithm classifiers and 26 stacked ensembles of combinations these algorithms on two popular metrics: Area Under ROC Curve (AUC) and lift. We will demonstrate a significant improvement in the AUC and lift values when the stacked generalization approach is used viz a viz the single-algorithm approach. We conclude that despite its relative obscurity in marketing applications, stacking holds great promise as an ensembling technique for direct marketing problems.**

Keywords – response modeling, stacked generalization, AUC, Lift

I. INTRODUCTION

We are living in the information age. Vast amounts of data and information are stored by companies and businesses about their clients and their habits. This data is a valued resource and in the application of data mining, businesses seek to leverage it in order to improve their competitive position in the market, reduce costs of operations and consequently improve their profit.

Data mining has been defined as the process of selection, exploration and modeling of large databases in order to discover models and patterns that are unknown apriori [1]. Data mining techniques provides companies with opportunities of learning from the data held in their data warehouses in order to inform future decisions and strategic actions. A popular framework for data mining activities is the CRISP-DM (CRoss Industry Standard Process for Data Mining) [2] framework. This framework outlines six phases for a data mining process. These are *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, and *deployment*. These phases can be undertaken iteratively, that is one can go back and forth between phases in a data mining project as its activities are refined and improved.

Direct marketing on the other hand has been described as the process of identifying likely buyers of certain products and promoting the products accordingly [3]. A direct marketing effort seeks to acquire and retain customers by contacting them directly with the objective of achieving a direct response which is usually the purchase or uptake of a product or service. In a direct marketing campaign, a business or its agents reach out to individual customers to sell a product or service through interactive communications in ways that allows response to be measured. It has the advantage of allowing the customization of messages for individuals [4], therefore making it possible to reach individual customers in ways most convenient to them.

Considering the amount of data held by companies, data mining can be a useful tool for making direct marketing efforts more effective and less costly for a business. Data mining in a direct marketing operation can be used to predict the most likely clients to purchase the product or service or take up an offer being marketed. In this approach, a machine learning model is trained on past customers data and then the model is applied to current prospects to predict those most likely to respond positively to a direct marketing effort. Only the most likely customers can then be contacted. Applying such a model to a direct marketing effort leads to a more effective campaign with a better response rate for fewer resources used. More of the best prospects will be reached while fewer resources are expended in the effort. This results in better profits for the business.

This rest of this paper is organized in the following manner: Section II presents a brief discussion of related work in response modelling in direct marketing. The proposed stacked models approach is discussed in Section III. Section IV describes the experiments carried out and the analysis of the results obtained. Section V presents a discussion of the results obtained in comparison to previous studies. We finally make conclusions and recommendations for future work in Section VI.

II. RELATED WORK

Both statistical and machine learning methods have been employed in modelling of direct marketing problems. Statistical methods such as logistic regression have traditionally been used to model response in marketing [5], [6]. Studies employing logistic regression include [7], [8]. Machine learning methods have of late also become popular in modelling these kinds of problems. Some of these algorithms that have been applied in studies include decision trees (DT) [9]–[15], Support Vector Machines (SVM) [10], [13], [16], neural networks (NN) [8], [10], [14], [17]–[20] and Naïve Bayes (NB) [11], [21]. These studies have demonstrated the capabilities of these algorithms to create simple learners that can select the most likely respondents to a marketing campaign.

In [13], Moro et al. compared Decision Trees, Naïve Bayes and Support Vector Machines (SVM) models in predictive modelling of the bank telemarketing problem. They applied the CRoss-Industry Standard Process for Data Mining (CRISP-DM) and used AUC and lift analysis to compare the models. In that work, SVM performed better than the other models with an AUC of 0.938 and area under lift curve (ALIFT) of 0.887. In other work [11], they used a novel rolling windows evaluation scheme, and compared Logistic Regression (LR), Decision Trees (DT), Neural Network (NN) and Support Vector Machine (SVM) Models. In this work, the neural network model outperformed the other models with an AUC 0.794.

Sing'oei and Wang in [12] proposed a five phased data mining framework for direct marketing. They applied C5.0 decision tree to model the bank telemarketing problem. Their C5.0 model achieved an accuracy of 93%.

Nachev in [8] undertook a case study of data mining modelling techniques for direct marketing. He compared five models: Neural Network (NN), Logistic Regression (LR), Naïve Bayes (NB), Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) all tested at different levels of data saturation. The neural network model produced the best overall performance at 98% data saturation with an average AUC of 0.915.

III. THE STACKING APPROACH

The proposed stacking approach implements stacked ensembles [22], [23] in response modelling of direct marketing problems. Stacked models have been shown to generate classifiers with superior predictive performance than the constituent base classifiers. This is especially true when the stacked learners employ dissimilar approaches to learning.

Selective ensembles [24] is where after generating a set of base learners, selecting some base learners instead of using all of them to compose an ensemble is a better choice. The framework we propose for customer response modelling applies this approach. In the proposed approach, candidate algorithms for modeling are identified beforehand. These are typically algorithms that have been shown to perform well in the given domain (in this case, customer response marketing). Once these have been identified, learners are created and tuned for each algorithm.

A second level meta-learner is identified and stacked ensembles are then created of all possible combinations of these learners. These stacked models, together with the base algorithms are executed in a benchmark experiment. The result of the benchmark experiment is the comparative performance of each of the benchmarked models – both single algorithm and stacked. Out of this result, the best stacked model is then selected for application.

This approach is illustrated in Figure 1, below.

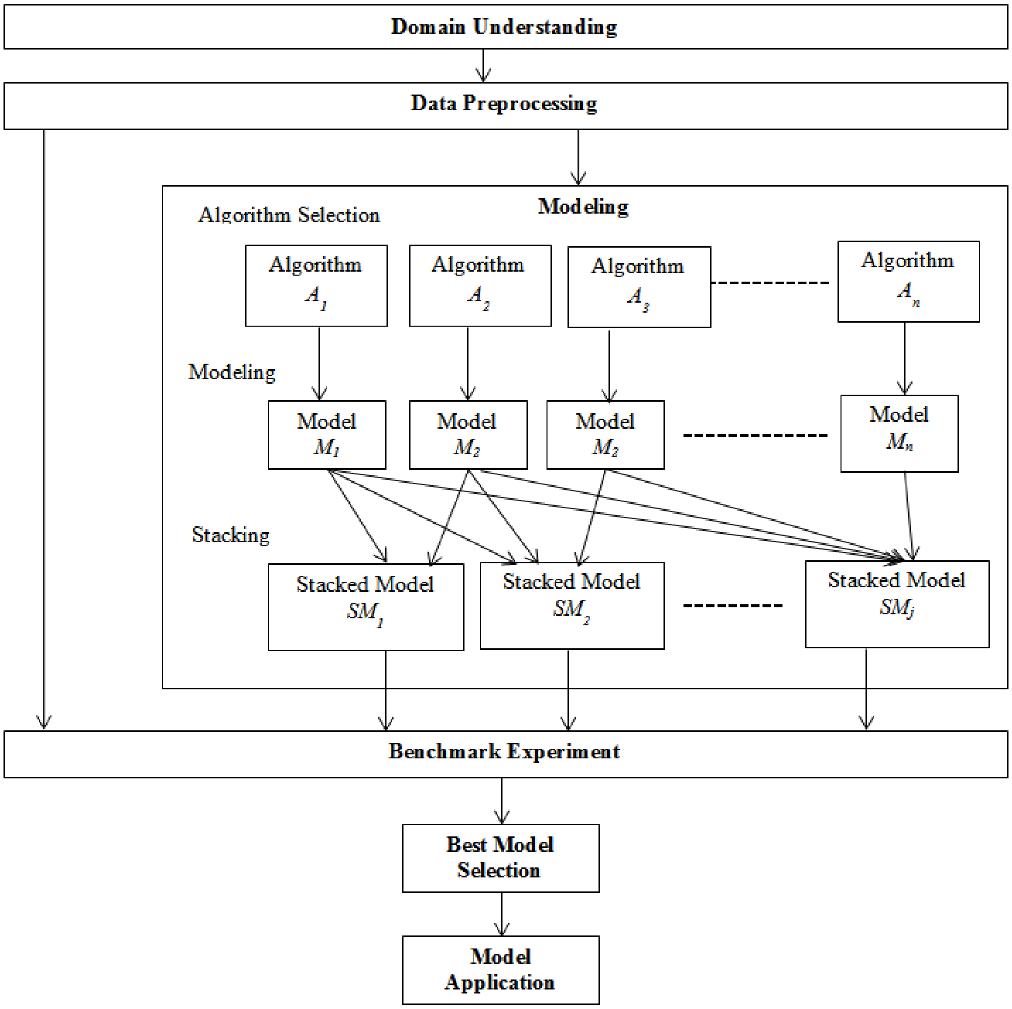

Figure 1 The proposed stacked models approach

The proposed stacked models approach is a five phase process as described below.

Domain understanding

The goal of this phase is for the analyst to familiarize oneself with the relevant domain including prior knowledge, requirements and goals of the application.

Modeling

Stacked generalization involves the application of multiple heterogeneous models. At this stage, the candidate algorithms are identified and selected. Prior knowledge of successful algorithms for the domain/dataset, is useful at this stage. An analyst may select algorithms that have shown good performance for the domain/dataset in prior work. Once these have been identified, they are used to create models. These single algorithm models are then combined together in stacked ensembles using a chosen meta-learner in readiness of the next stage.

Data Pre-processing

In data processing, the final dataset for modelling is created from the raw data. Activities such as feature selection, data cleaning, and creation of new attributes are undertaken at this stage. Strategies for handling of missing data are applied at this stage. Dimensionality reduction could also be undertaken at this stage to reduce the number of variables.

Benchmark Experiment

This is the stage that is used to select the best stacked model from among those constructed in the Algorithm Selection phase. A benchmark experiments is an empirical experiments with the aim of comparing and ranking algorithms with respect to certain performance measures [25]. In this phase, a benchmark experiment of all the models constructed in the Algorithm Selection and Stacking phase is executed. Out of the benchmark experiment, the best stacked model is identified.

Best Model Selection

From the results of the benchmark experiment, the best stacked model is identified for the problem. This is the model that is then applied in production in the next phase.

Model Application

This is the final phase of the process. Here, the best model selected from the benchmark experiment is applied by the organization in response modelling for their direct marketing campaigns.

IV. EXPERIMENTS AND RESULTS

Data Exploration and Preprocessing

A key process in the CRISP-DM (CRoss Industry Standard Process for Data Mining) framework for data mining is data preparation. In this stage, data to be used is identified, selected and prepared for inclusion in the data mining model. This involves the acquisition, integration, and formatting of the data according to the project's needs. The data is then be cleaned up and transformed according to the requirements of the algorithm(s) that will be applied.

The dataset used in this work is the Bank Marketing dataset from the UCI repository. This data was collected by a Portuguese banking institution during direct marketing campaigns. The campaigns ran from May 2008 to November 2010. Telephone calls were the primary marketing channel but internet online banking channels were also employed to contact customers. The goals of these campaigns were to sell an attractive long term deposit product with good interest rates. Data was collected for every contact made including whether the contact resulted in a positive response (a yes) or a negative response (a no) from the contacted client. In the dataset this is encoded in a target variable *"y"* with the possible values of *"yes",* if the customer subscribed to the long term deposit product offered or *"no"* if the customer did not subscribe to the offer. This will be the target variable in the models we build in this work. The classification aim is to predict this variable, given yet unseen data for a client.

The dataset consists of 4119 records of customer contacts during the marketing campaign. Each record has 21 attributes, 5 of which are integer, 5 of which are continuous and 11 categorical. The target variable is a categorical variable "y" with two possible levels "yes" and "no" indicating the outcome of a contact. The dataset is described in detail in [11].

Some algorithms such as SVMs and Neural Networks are based on the assumption of a well distributed dataset. For algorithms, it is necessary to scale and center the data before application to the model. In the data preparation stage, skewness of attribute values was investigated and high levels of skew detected in some of the attributes. This called for normalization and scaling of such variables such that it is ensured that all data attributes have equal weights regardless of nature of the data or measurement units used.

There were no missing values in the dataset.

Computational Environment

The experiments performed in this work were conducted using the R[26] language and the RStudio editor for R. The MLR [27] R package for machine learning was used for the creation of creation of learners, training and testing. MLR provides a standardized interface to most machine learning algorithms and a host of data mining tools. The Rattle [28] graphical interface package was used for data exploration and analysis.

Experiments were run on an HP EliteBook 8770w Workstation laptop running Ubuntu 16.04 and equipped with 8GB RAM and 500 GB hard disk.

The statistical tests were run using the XLSTAT [29] statistical add-in for Microsoft Excel. The significance level used in the *t-test* is 5%. This value is the most commonly used in literature [30].

Experiment

After data preprocessing and preparation, we proceeded to the modeling stage. 25% of the dataset was set aside as a validation set. The rest was used to train and test the model in repeated 10 fold cross validation. 10x10 cross validation was used in our experiments. This is because it is known to provide better replicability than a simple 10-fold CV [30]. 10 repeats were selected to guarantee more robust estimates of model performances.

To statistically evaluate the model performances, we used the k-fold Cross-validated Paired *t-Test* [31]. This test was chosen because of its power (i.e. the ability to detect a difference in classifier performance when one actually exists).

Of the entire dataset of 4119 records, 3090 were randomly selected to be used as the validation set. For all the five basic algorithms (Neural Network –NN, Logistic Regression – LR, Support Vector Machines – SVM, Naïve Bayes – NB, and Decision Tree –DT(CART)), machine learning models were created and tuned for AUC. These models were then combined into stacked ensemble models. Using a logistic regression meta-learner, 26 stacked models were created from of the five basic models. The 26 stacked models, together with the five basic models (making a total of 31 models) were executed in a repeated 10x10 fold cross validation benchmark experiment.

The models executed in the benchmark experiment are shown in the Table 1 below.

Table 1Models applied in the benchmark experiment

1	Neural Network (NN)	17	Stack(NN+LR+SVM)
2	Logistic Regression(LR)	18	Stack(NN+LR+DT)
3	Naïve Bayes(NB)	19	Stack(NN+NB+SVM)
4	Support Vector Machine(SVM)	20	Stack(NN+NB+DT)
5	Decision Tree (CART)	21	Stack(NN+SVM+DT)
6	Stack(DT+NN+LR+)	22	Stack(LR+NB+SVM)
7	Stack(NN+NB)	23	Stack(LR+NB+DT)
8	Stack(NN+SVM)	24	Stack(LR+SVM+DT)
9	Stack(NN+DT)	25	Stack(NB+SVM+DT)
10	Stack(LR+NB)	26	Stack(NN+LR+NB+SVM)
11	Stack(LR+SVM)	27	Stack(NN+LR+NB+DT+)
12	Stack(LR+DT)	28	Stack(NN+LR+SVM+DT)
13	Stack(NB+SVM)	29	Stack(NN+NB+SVM+DT)
14	Stack(NB+DT)	30	Stack(LR+NB+SVM+DT)
15	Stack(SVM+DT)	31	Stack(NN+LR+NB+SVM+DT)
16	Stack(NN+LR+NB)		

Results

The averaged AUC results for the benchmark result are as shown below.

Table 2: Average AUC valued for benchmarked models (ordered from best to worst)

	Model	**Average AUC**
1	NN+DT Stacked Model	0.942077
2	NN+LR+DT Stacked Model	0.940922
3	NN	0.940629
4	NN+LR Stacked Model	0.940316
5	NN+SVM+DT Stacked Model	0.938759
6	NN+LR+SVM+DT Stacked Model	0.938407
7	NN+LR+NB+DT Stacked Model	0.938221
8	NN+NB+DT Stacked Model	0.938157
9	NN+LR+NB+SVM+DT Stacked Model	0.938019
10	NN+NB+SVM+DT Stacked Model	0.937874
11	NN+LR+NB Stacked Model	0.937736
12	NN+NB Stacked Model	0.937568
13	NN+SVM Stacked Model	0.937524
14	NN+LR+SVM Stacked Model	0.937327
15	NN+NB+SVM Stacked Model	0.936981
16	SVM+DT Stacked Model	0.936747
17	NN+LR+NB+SVM Stacked Model	0.936655
18	LR+DT Stacked Model	0.936527
19	LR+SVM Stacked Model	0.935935
20	NB+SVM+DT Stacked Model	0.934908
21	LR	0.934789
22	LR+SVM+DT Stacked Model	0.934481
23	SVM	0.933964
24	LR+NB+DT Stacked Model	0.933417
25	NB+SVM Stacked Model	0.932716
26	LR+NB+SVM Stacked Model	0.93232
27	LR+NB+SVM+DT Stacked Model	0.932122
28	LR+NB Stacked Model	0.930842
29	NB+DT Stacked Model	0.912452
30	NB	0.887083
31	DT	0.850268

The results obtained after the benchmark experiment is as shown in the table above. The stacked model comprising of a Neural Network model and a Decision Tree (NN+DT) model achieved the best score in terms of AUC (0.9421) followed by another stacked model (Neural Network + Logistic Regression + Decision Tree which achieved an average AUC of 0.9409).

Neural network, Logistic Regression, Support Vector Machine, Naïve Bayes and Decision tree models achieved average AUCs of 0.9406, 0.9348, 0.9340, 0.8871, and 0.8503 respectively.

From the results, two stacked models have performed better than any model single-algorithm model. But how significant is performance advantage? We use statistical tests to answer this question next.

Statistical Tests for AUC

A cross-validated *t-test* [31] was done to test if the improved performance of the Neural Network and Decision Tree stacked model over the five basic models (NN, LR, SVM, NB and DT) was statistically significant.

As the tests seek to determine if it is in fact the case that the stacked model has a superior AUC than the single-algorithm models, we chose to undertake the paired samples one-tailed *t-test*. In these tests, the null hypothesis (H_o) states that there is no difference in the average AUC of the single-algorithm models and the stacked model whereas the alternative hypothesis H_a states that the average AUC of the stacked model is greater than the average AUC of a single algorithm model.

Since there are five models whose performance we want to compare with the best stacked model (NN+DT) picked by the benchmark, six paired *t-tests* were carried out.

The tests were as follows:

i. ***Stacked Model (NN+DT) & Neural Network Model paired t-test***

Hypothesis:
H_0: The difference between the AUC means is equal to 0.
H_a: The difference between the AUC means is greater than 0

ii. ***Stacked Model (NN+DT) Logistic Regression Model***

paired t-test

Hypothesis:

H_0: The difference between the AUC means is equal to 0.

H_a: The difference between the AUC means is greater than 0

iii. *Stacked Model (NN+DT) & SVM Model paired t-test*

Hypothesis:

H_0: The difference between the AUC means is equal to 0.

H_a: The difference between the AUC means is greater than 0

iv. *Stacked Model (NN+DT) & Naïve Bayes Model paired t-test*

Hypothesis:

H_0: The difference between the AUC means is equal to 0.

H_a: The difference between the AUC means is greater than 0

v. *Stacked Model (NN+DT) & Decision Tree Model paired t-test*

Hypothesis:

H_0: The difference between the AUC means is equal to 0.

H_a: The difference between the AUC means is greater than 0

The results for the statistical tests are summarized in the Table 3 below:

Table 3: Best stacked model AUC summary statistics

	Stacked Model Summary Statistics (AUC)				
Model	**Observations**	**Minimum**	**Maximum**	**Mean**	**Std. Deviation**
Stacked Model (NN+DT)	100	0.977	0.942	0.015	0.908

Table 4: Cross-validated base models summary statistics and *t-test* results (AUC)

	Summary Statistics (AUC)					**Paired (Stacked Ensemble –NN+DT & Model) *t-test* Results**					
Model	**Observations**	**Minimum**	**Maximum**	**Mean**	**Std. Deviation**	**Difference**	**t-value (observed)**	**t value (Critical value)**	**DF**	**p-value**	**alpha α**
Neural Network	100	0.905	0.979	0.941	0.015	0.001	4.886	1.660	99	<0.0001	0.05
Logistic Regression	100	0.894	0.981	0.935	0.017	0.007	6.904	1.660	99	<0.0001	0.05
SVM	100	0.897	0.967	0.934	0.016	0.008	12.675	1.660	99	<0.0001	0.05
Naïve Bayes	100	0.805	0.955	0.887	0.027	0.055	29.302	1.660	99	<0.0001	0.05
Decision Tree	100	0.728	0.946	0.850	0.039	0.092	29.111	1.660	99	<0.0001	0.05

Interpretation:

From the Table 3 and Table 4, we can conclude that the stacked model (NN+DT) achieved better AUC (mean=0.942, SD=0.015) than the five single algorithm models i.e. the Neural Network Model (mean = 0.941, SD=0.015), Logistic Regression Model (mean = 0.935, SD=0.017), SVM model (mean = 0.934, SD=0.016), Naïve Bayes Model (mean = 0.887, SD=0.027) and the Decision Tree Model (mean = 0.850, SD=0.039).

The paired t-tests found the difference in AUC performance between the stacked model and the rest of the models to be significant. The t-test results are summarized in Table 5 below:

Table 5 t-test results summary (AUC)

Paired t-test	**Test of Significance (α=0.05)**
Stacked model –NN model	t(99) =4.886, p<0.0001
Stacked model –LR model	t(99) =6.904, p<0.0001
Stacked model –SVM model	t(99) =12.675, p<0.0001
Stacked model –NB model	t(99) =29.302, p<0.0001
Stacked model –DT model	t(99) =29.111, p<0.0001

From Table 5, we observe that all the five the t-tests are significant. This is since the computed *p-value* for all the tests is much less than the significance level *α=0.05*. This suggests that the AUC performance advantage of the stacked model over the single-algorithm models is significant. We therefore, should reject the null hypotheses H_0, that there is no difference in AUC means between the stacked model and the single algorithm models and accept the alternative hypotheses H_a that the difference between the Stacked model AUC and the single algorithm model's AUC's is greater than zero.

Lift Analysis

In the previous section, we investigated the stacked model performance *viz a viz* the single algorithm model from a machine learning point of view using the AUC scores. The problem at hand is a marketing problem. It would also be convenient to a marketer to see how the stack model performs against the single algorithm model using the metrics most often used by marketing professionals i.e. Lift and Gains/Cumulative lift.

We performed lift analysis for all the five models together with the best stack model. The average lift and gain/cumulative lift values for the six models are shown in the Table 6.

The 10th decile lift is a popular metric used in the marketing domain[32]. We also used lift to compare the performance of the best stacked model with the five basic models. As with the AUC values, the average lifts across the 10x10 CV runs were computed and compared. The *t-test* was also used to validate the significance of the difference in performance between the models.

Lift Analysis Results

The average 10th decile lifts for the five single-algorithm models plus the best stacked model is shown in Table 6 below.

Table 6 Average top (10th) decile lift

	Model	**10th Decile Lift**
1	NN+DT Stacked Model	5.9542
2	Neural Network Model	5.8003
3	Decision Tree Model	5.6880
4	Logistic Regression Model	5.6674
5	SVM Model	5.4940
6	Naïve Bayes Model	4.6056

For the lift metric, the stacked model has also outperformed the single algorithm models as shown in Table 6 above.

To check the significance of these results, the significance of the difference in mean lift values were tested using the paired one-tailed t-test. The results are discussed below.

Statistical Tests for Lift

Six paired statistical tests were done to test the significance of the differences between the average lifts of the Stacked models and the single-algorithm models.

The tests were as follows:

i. ***Stacked Model (NN+DT) & Neural Network Model***
paired t-test

Hypothesis:

H_0: The difference between the 10th decile lift means is equal to 0.

H_a: The difference between the 10th decile lift means is greater than 0

ii. ***Stacked Model (NN+DT) & Logistic Regression Model***
paired t-test

Hypothesis:

H_0: The difference between the 10th decile lift means is equal to 0.

H_a: The difference between the 10th decile lift means is greater than 0

iii. ***Stacked Model (NN+DT) & SVM Model***
paired t-test

Hypothesis:

H_0: The difference between the 10th decile lift means is equal to 0.

H_a: The difference between the 10th decile lift means is greater than 0

iv. ***Stacked Model (NN+DT) & Naïve Bayes Model***
paired t-test

Hypothesis:

H_0: The difference between the 10th decile lift means is equal to 0.

H_a: The difference between the 10th decile lift means is greater than 0

v. ***Stacked Model (NN+DT) & Decision Tree Model***
paired t-test

Hypothesis:

H_0: The difference between the 10th decile lift means is equal to 0.

H_a: The difference between the 10th decile lift means is greater than 0.

A summary of the results of lift analysis is presented in Table 7 below.

Table 7: Calculated Average lift and Cumulative lift/Gain for the models

Decile	**Stacked Model**	**NN Model**	**LR Model**	**SVM Model**	**NB Model**	**DT Model**

	Lift	Gains	Lift	Gains	Lift	Gains	Lift	Gains	Lift	Gains	Lift	Gains
10	5.95	59.54	5.80	58.00	5.67	56.67	5.49	54.94	4.61	46.06	5.69	56.88
20	4.27	85.49	4.25	84.99	4.25	85.05	4.21	84.25	3.69	73.84	3.88	77.64
30	3.20	95.93	3.19	95.81	3.16	94.93	3.15	94.37	2.88	86.43	2.70	81.10
40	2.46	98.23	2.46	98.23	2.43	97.31	2.45	97.90	2.31	92.42	2.08	83.37
50	1.99	99.65	1.99	99.62	1.98	98.88	1.98	99.08	1.92	95.81	1.72	85.87
60	1.67	99.94	1.67	99.94	1.66	99.50	1.67	99.91	1.63	97.64	1.47	87.91
70	1.43	100.00	1.43	100.00	1.42	99.68	1.43	100.00	1.41	98.67	1.30	91.33
80	1.25	100.00	1.25	100.00	1.25	99.71	1.25	100.00	1.24	99.14	1.19	95.40
90	1.11	100.00	1.11	100.00	1.11	99.71	1.11	100.00	1.11	99.76	1.09	97.73
100	1.00	100.00	1.00	100.00	1.00	100.00	1.00	100.00	1.00	100.00	1.00	100.00

Table 8: Best stacked model Lift summary statistics

	Stacked Model Summary Statistics (Lift)				
Model	**Observations**	**Minimum**	**Maximum**	**Mean**	**Std. Deviation**
Stacked Model (NN+DT)	100	3.824	7.353	5.954	0.650

Table 9: Cross validated t-test results (Lift)

	Summary Statistics (Top Decile Lift)					**Paired (Stack & Model) t-test Results**					
Model	**Observations**	**Minimum**	**Maximum**	**Mean**	**Std. Deviation**	**Difference**	**t-value (observed)**	**t value (Critical value)**	**DF**	**p-value**	**alpha α**
Neural Network	100	3.824	7.941	5.800	0.669	0.154	3.995	1.660	99	<0.0001	0.05
Logistic Regression	100	4.118	8.235	5.667	0.653	0.287	6.035	1.660	99	<0.0001	0.05
SVM	100	3.235	7.353	5.494	0.674	0.460	10.248	1.660	99	<0.0001	0.05
Naïve Bayes	100	2.941	6.765	4.606	0.678	1.349	17.217	1.660	99	<0.0001	0.05
Decision Tree	100	4.118	7.059	5.688	0.597	0.266	5.616	1.660	99	<0.0001	0.05

Interpretation:

From the Table 8 and Table 9, we can conclude that the best stacked model (NN+DT) achieved better top decile lift (mean=5.954, SD=0.650) than the all the single model algorithms i.e. Neural Network Model (mean = 5.800, SD=0.669), Logistic Regression Model (mean = 5.667, SD=0.653), SVM model (mean = 5.494, SD=0.674), Naïve Bayes Model (mean = 4.606, SD=0.678) and the Decision Tree Model (mean = 5.688, SD=0.597)

The paired t-tests found the difference in lift performance between the stacked model and the rest of the models to be significant. The t-test results are summarized in Table 10 below:

Table 10 t-test results summary (10th decile lift)

Paired t-test (Lift)	**Test of Significance (α=0.05)**
Stacked model –NN model	t(99) =3.995, p<0.0001
Stacked model –LR model	t(99) =6.035, p<0.0001
Stacked model –SVM model	t(99) =10.248, p<0.0001
Stacked model –NB model	t(99) =17.217, p<0.0001
Stacked model –DT model	t(99) =5.616, p<0.0001

From Table 10 we observe that all the five the *t-tests* are significant as the computed *p-value* is lower than the significance level *α=0.05*. This suggests that the top decile lift performance advantage of the stacked over the single-algorithm models is significant. For all the 5 tests, we therefore reject the null hypotheses H_0, that there is no difference in lift means between the stacked model and the single algorithm models and accept the alternative hypotheses H_a that the difference between the Stacked model lift and the single algorithm models lifts is greater than zero.

Lift Chart

For visual comparison, a lift chart was plotted to illustrate the lift performances of the six models. This is shown next.

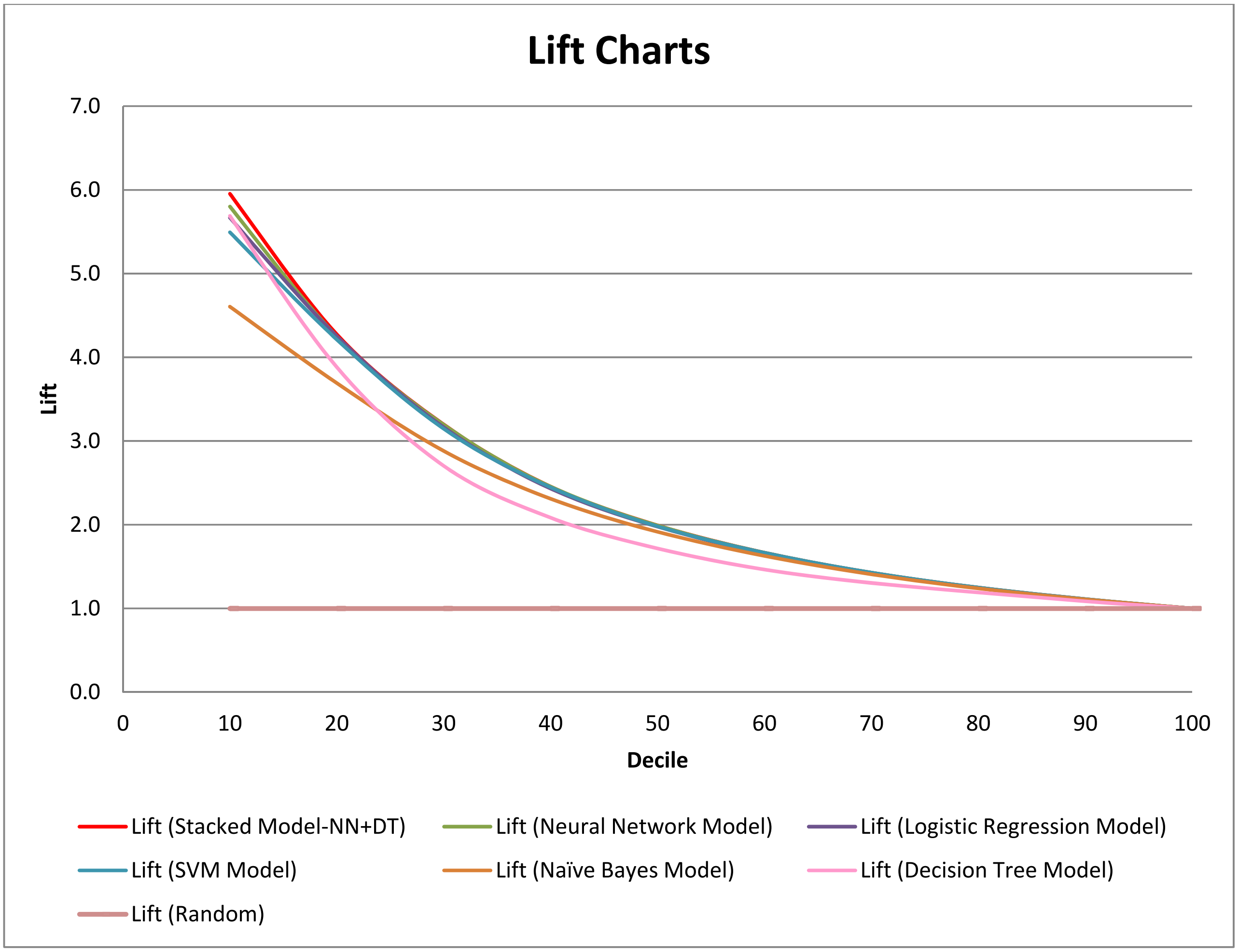

Figure 2 Lift plots for the six models

From the lift chart in Figure 2 above and Table 7, we can see the dominance of the stacked model (red plot) at both the 10th decile and 20th decile.

Comparison with Results from Previous Work

In [33], [21], Moro et al. applied a Naïve Bayes, Decision Tree, and Support Vector Machine models to the bank telemarketing response modeling problem. In that study, the SVM model outperformed both the Naïve Bayes and decision tree model. The SVM model achieved AUC of 0.938 while the Naïve Bayes and Decision Tree models achieved 0.870 and 0.868 respectively. Comparing these results with the results obtained by the best stacked model (NN-DT) in this study, it is evident that the best stacked model achieved does much better (AUC = 0.942) than the best model obtained in that study SVM with AUC of 0.938.

Moro et al. in [11] compared four DM models in response modeling a bank telemarketing problem. The models they studied were logistic regression, decision trees (CART), neural network (NN) and support vector machine. They evaluated the models using the latest data of a marketing campaign using a rolling windows scheme. The Neural Network achieved the best result with and AUC of 0.794 while the logistic regression, decision tree and SVM achieved 0.715, 0.757 and 0.767 respectively. With an AUC of 0.942, the stacked model selected in this work compares favorably with the results obtained in that study.

In [8] a comparative analysis of neural nets (NN), logistic regression (LR), Naive Bayes (NB), linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) in was done. In that study, the Neural Network produced the best average AUC of 0.915. The LR, NB, LDA and QDA models achieved average AUCs of 0.902, 0.852, 0.900, and 0.838 respectively. These AUC values are much less than the AUC of 0.942 achieved by the stacked model introduced in this work.

V. CONCLUSION AND FUTURE WORK

We have demonstrated in this paper that in response modeling of a direct marketing problem, a stacked model can have a better performance than single algorithm models. 31 models – 26 stacked and 5 single algorithm models – were constructed and evaluated using the two key metrics of AUC and Lift. We showed that stacked models can provide superior predictive performance to single-algorithm models in selecting the top prospects in directed marketing campaign. The results for the experiments were confirmed to be significant through statistical testing (*t-tests* using the 5% significance level)

The stacked models approach has now been shown to have the capacity to improve a models predictive performance. Better predictive performance through the use of stacked models as shown in this work would guarantee better response models which translate to lower costs and efficient marketing hence better return on investment (ROI) per campaign.

Investigating the suitability of the stacking approach in other domains provides promising avenues for future research. It would also be interesting to study how stacking would fare against the other ensembling techniques such as bagging and boosting in the context of response modeling problems.

REFERENCES

[1] P. Giudici, *Applied data mining: statistical methods for business and industry*. New York: J. Wiley, 2003.

[2] R. Wirth and J. Hipp, “CRISP-DM: Towards a standard process model for data mining,” presented at the Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining, 2000, pp. 29–39.

[3] C. X. Ling and C. Li, “Data mining for direct marketing: Problems and solutions.,” in *KDD*, 1998, vol. 98, pp. 73–79.

[4] M. J. Berry and G. Linoff, *Data mining techniques: for marketing, sales, and customer support*. John Wiley & Sons, Inc., 1997.

[5] D. A. Aaker, V. Kumar, and G. S. Day, *Marketing research*. John Wiley & Sons, 2008.

[6] P. Berger and T. Magliozzi, “The effect of sample size and proportion of buyers in the sample on the performance of list segmentation equations generated by regression analysis,” *J. Direct Mark.*, vol. 6, no. 1, pp. 13–22, 1992.

[7] V. S. Lo, “The true lift model: a novel data mining approach to response modeling in database marketing,” *ACM SIGKDD Explor. Newsl.*, vol. 4, no. 2, pp. 78–86, 2002.

[8] A. Nachev, “Application of data mining techniques for direct marketing,” *Comput. Models Bus. Eng. Domains*, 2015.

[9] D. Haughton and S. Oulabi, “Direct marketing modeling with CART and CHAID,” *J. Interact. Mark.*, vol. 11, no. 4, pp. 42–52, 1997.

[10] S. F. Crone, S. Lessmann, and R. Stahlbock, “The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing,” *Eur. J. Oper. Res.*, vol. 173, no. 3, pp. 781–800, 2006.

[11] 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.

[12] L. Sing’oei and J. Wang, “Data mining framework for direct marketing: A case study of bank marketing,” *Int. J. Comput. Sci. Issues IJCSI*, vol. 10, no. 2, pp. 198–203, 2013.

[13] S. Moro, P. Cortez, and R. Laureano, “A data mining approach for bank telemarketing using the rminer package and r tool,” 2013.

[14] H. A. Elsalamony and A. M. Elsayad, “Bank Direct Marketing Based on Neural Network and C5. 0 Models,” *Int. J. Eng. Adv. Technol. IJEAT*, vol. 2, no. 6, 2013.

[15] S. Abbas, “Deposit subscribe Prediction using Data Mining Techniques based Real Marketing Dataset,” *ArXiv Prepr. ArXiv150304344*, 2015.

[16] S. Viaene *et al.*, “Knowledge discovery in a direct marketing case using least squares support vector machines,” *Int. J. Intell. Syst.*, vol. 16, no. 9, pp. 1023–1036, 2001.

[17] N. Levin and J. Zahavi, “Data mining for target marketing,” *Data Min. Knowl. Discov. Handb.*, pp. 1261–1301, 2005.

[18] B. Baesens, S. Viaene, D. Van den Poel, J. Vanthienen, and G. Dedene, “Bayesian neural network learning for repeat purchase modelling in direct marketing,” *Eur. J. Oper. Res.*, vol. 138, no. 1, pp. 191–211, 2002.

[19] K. Ha, S. Cho, and D. MacLachlan, “Response models based on bagging neural networks,” *J. Interact. Mark.*, vol. 19, no. 1, pp. 17–30, 2005.

[20] B. Curry and L. Moutinho, “Neural networks in marketing: modelling consumer responses to advertising stimuli,” *Eur. J. Mark.*, vol. 27, no. 7, pp. 5–20, 1993.

[21] S. Moro, P. Cortez, and R. Laureano, “A data mining approach for bank telemarketing using the rminer package and r tool,” 2013.

[22] D. H. Wolpert, “Stacked generalization,” *Neural Netw.*, vol. 5, no. 2, pp. 241–259, 1992.

[23] L. Breiman, “Stacked regressions,” *Mach. Learn.*, vol. 24, no. 1, pp. 49–64, 1996.

[24] Z.-H. Zhou, J. Wu, and W. Tang, “Ensembling neural networks: many could be better than all,” *Artif. Intell.*, vol. 137, no. 1–2, pp. 239–263, 2002.

[25] M. J. Eugster and F. Leisch, “Exploratory analysis of benchmark experiments an interactive approach,” *Comput. Stat.*, vol. 26, no. 4, pp. 699–710, 2011.

[26] R. C. Team, “R language definition,” *Vienna Austria R Found. Stat. Comput.*, 2000.

[27] B. Bischl *et al.*, “mlr: Machine learning in R,” *J. Mach. Learn. Res.*, vol. 17, no. 170, pp. 1–5, 2016.

[28] G. J. Williams, “Rattle: a data mining GUI for R,” *R J.*, vol. 1, no. 2, pp. 45–55, 2009.

[29] S. Addinsoft, “XLstat 2012: Leading data analysis and statistical solution for microsoft excel,” *Addinsoft SRL*, 2012.

[30] R. R. Bouckaert, “Choosing between two learning algorithms based on calibrated tests,” in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, pp. 51–58.

[31] T. Dietterich, “Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms,” *Neural Comput.*, vol. 10, no. 7, pp. 1895–1923, Oct. 1998.

[32] R. Blattberg, K. Byung-Do, and S. Neslin, *Database Marketing: Analyzing and Managing Customers*. 2008.

[33] S. Moro, R. Laureano, and P. Cortez, “Using data mining for bank direct marketing: An application of the crisp-dm methodology,” in *Proceedings of European Simulation and Modelling Conference-ESM’2011*, 2011, pp. 117–121.