UNDERSTANDING THE APPLICABILITY OF LINEAR & NON-LINEAR MODELS USING A CASE-BASED STUDY

Gaurav Singh Thakur[1],Anubhav Gupta [2], Ankur Bhardwaj[3] and Biju R Mohan[4]

Department of Information Technology, National Institute of Technology Karnataka, Surathkal

Abstract

This paper uses a case based study – "product sales estimation" on real-time data to help us understand the applicability of linear and non-linear models in machine learning and data mining. A systematic approach has been used here to address the given problem statement of sales estimation for a particular set of products in multiple categories by applying both linear and non-linear machine learning techniques on a data set of selected features from the original data set. Feature selection is a process that reduces the dimensionality of the data set by excluding those features which contribute minimal to the prediction of the dependent variable. The next step in this process is training the model that is done using multiple techniques from linear & non-linear domains, one of the best ones in their respective areas. Data Re-modeling has then been done to extract new features from the data set by changing the structure of the dataset & the performance of the models is checked again. Data Remodeling often plays a very crucial and important role in boosting classifier accuracies by changing the properties of the given dataset. We then try to explore and analyze the various reasons due to which one model performs better than the other & hence try and develop an understanding about the applicability of linear & non-linear machine learning models. The target mentioned above being our primary goal, we also aim to find the classifier with the best possible accuracy for product sales estimation in the given scenario.

Keywords

Machine Learning, Prediction, Linear and Non-linear models, Linear Regression, Random Forest, Dimensionality Reduction,Feature Selection, Homoscedasticity, Overfitting, Regularization, Occam Razor Hypothesis, Elastic Net

1. Introduction

According to Arthur Samuel (1959), **Machine learning** is a field of study that gives computers the ability to learn without being explicitly programmed. For example, given a purchase history for a customer and a large inventory of products, a machine learning algorithm can be used to identify those products in which that customer will be interested and likely to purchase. There are various types of Machine Learning Algorithms; two of the main types are Supervised Learning (where we provide the algorithm with a training dataset in which the right answers are given i.e. for each training example, the right output is given), and Unsupervised Learning (where we do

DOI : 10.5121/ijaia.2014.5603

not provide the algorithm with the right answers). In this paper, we discuss the techniques which belong to Supervised learning[3].

Predicting the future sales of a new product in the market has intrigued many scholars and industry leaders as a difficult and challenging problem. It involves customer sciences and helps the company by analyzing data and applying insights from a large number of customers across the globe to predict the sales in the upcoming time in near future. The success or failure of a new product launch is often evident within the first few weeks of sales. Therefore, it is possible to forecast the sales of the product in the near future by analyzing its sales in the first few weeks. We propose to predict the success or failure of each of product launches 26 weeks after the launch, by estimating their sales in the 26th week based only on information up to the 13th week after launch. We intend to do so by combining data analysis with machine learning techniques and use the results for forecasting.

We have divided the work into following phases:

i) Dimensionality reduction (Feature selection)
ii) Application of Linear & Non-Linear Learning Models
iii) Data Re-modeling
iv) Re-application of learning models.
v) Evaluation of the performance of the learning models through comparative study & Normality tests.
vi) Boosting the accuracy of the model that better suits the problem based on their evaluation.

To develop a forecasting system for this problem statement, we gathered 26 weeks information for nearly 2000 Products belonging to 198 categories to train our model. Various attributes such as units_sold_that_week, Stores_selling_in_the_nth_week, Cumulative units sold to a number of different customer groups etc are used as independent variables to train & predict the dependent variable- “Sales_in_the_nth_week”. However, our task here is only to predict their sales in the 26th week.

In Section 2, we discuss about the methodology and work done in each of the phases, followed by the results & discussion in Section 3.Finally, we draw a conclusion in Section 4 along with its applications, followed by the references.

2. Methodology and Work Done

A basic block diagram to explain the entire process of Machine Learning is given below.

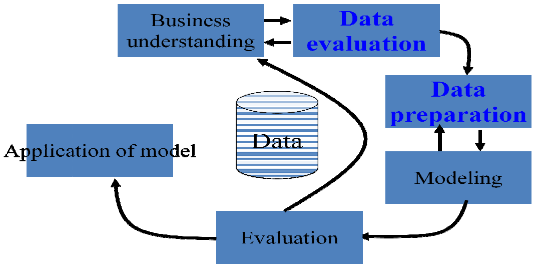


Figure 1 : Machine Learning Life Cycle

2.1) Feature selection

We use Greedy Stepwise[2]mechanism for feature selection[2]. The process of feature selection gives us a list of important features from the original feature set. Here stores_selling_in_the_nth_weekand weeks_since_launch have been the two most important features with maximum sales predicting power in the original data set.

The results from this procedure can be backed up using the scatter plots. The scatter plots are used for the feature "Total Units sold in nth week" plotted against other features. Their variations are then studied and can be used as a reference to justify the results from feature selection. Those scatter plots with random nature can be easily identified and discarded.

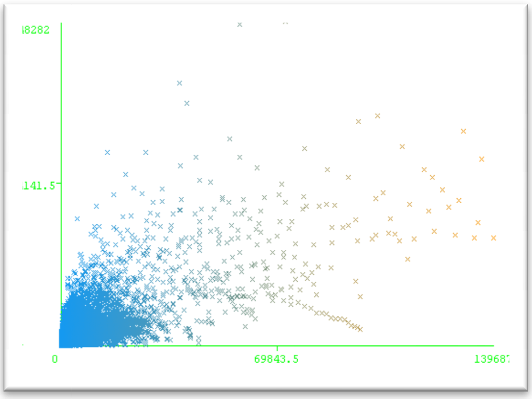


Figure 2: Cumulative Units sold to very price sensitive customers vs. Total units Sold

The above scatter plot is between:

y-axis	Total Units sold
x-axis	Units sold to Very Price sensitive customers

We can clearly see, the scatter plot between the two features does not show any trend as it is completely random in nature. A similar scatter plot was seen for most of the features, except for those obtained from feature selection. For the ones obtained from the feature selection processthese scatter plots showed some relation between them which confirms the fact that they are the necessary features for regression.

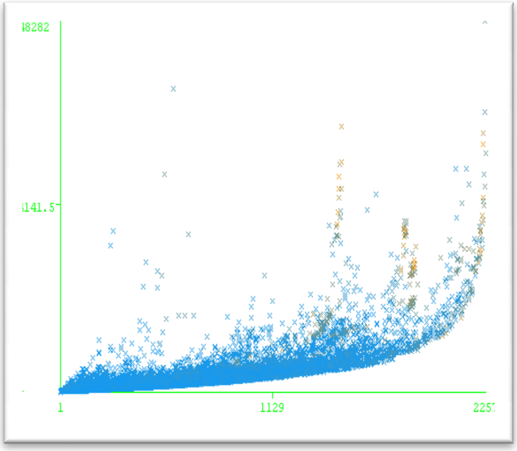


Figure 3 : Total Units Sold Vs Stores Selling

y-axis	Total Units sold
x-axis	Stores Selling

Hence, this allows us to reduce the number of features that must be used to train our model.

2.2) Linear Model

We used Multiple Linear Regression[1] for our linear learning model. The equation for Multiple Linear Regression is given below.

$$\mathbf{F_{\Theta}(X) = \Theta + \sum \Theta_i X} \quad \text{(Eq.. 1)}$$

Where, X is the set of input vector with coefficients/weights Θ_i and constant value of Θ called the bias. $F_{\Theta}(X)$ is the approximated Linear function to be used for regression.

This model needs to be optimized by minimizing the Mean Square Error produced by the model. The cost function in this case is:

$$\mathbf{J(\Theta)=(1/2m) \sum (F_{\Theta}\ (x_i)- y_i)\hat{}2} \quad \text{(Eq.. 2)}$$

Where, F_{Θ} (x_i) is the predicted value, y_i is the actual value, and 'm' is the number of tuples used for training. This is the cost function which has been optimized using Gradient Descent Algorithm [4].

We have applied this linear learning model on the data set of selected features. The results obtained have been mentioned in the next section.

2.3) Non-Linear Model

We use Random Forest, a bagging based ensemble learning technique for non-linear training. A Random Forest[9] consists of a collection or ensemble of basedecision tree predictors/classifiers, each capable of producing a response when presented with a set of predictor input values. For classification problems, this response takes the form of a class membership, which associates, or classifies, a set of independent predictor values with one of the categories present in the dependent variable. Alternatively, for regression problems, the tree response is an estimate of the dependent variable given the predictors, taking the average from each tree as the net output of the model.

Each tree is grown as follows[3]:

1. Firstly, create n_tree bootstrap samples allowing selection with replacement, where each sample will be used to create a tree.

2. If there are M input variables, a number m<M is specified such that at every node, m variables are selected at random out of the M and the best splitting attribute off these m is selected. The value of m is kept constant during the process.

3. Each tree is grown completely without pruning.

This technique was implemented in R tool, with the parameter values as n_trees = 500 and m (variables tried as each split) = sqrt(Number of features).

The accuracy of Random forests is calculated from the out-of-bag MSE which provides an unbiased result and eliminates the need for cross-validation.

$$\mathbf{MSE = (1/n) \sum (y_i - F_{\Theta\_oob})^\wedge 2} \qquad \text{(Eq.. 3)}$$

2.4) Data Remodeling

Data Remodeling is a phase that requires some domain specific knowledge and use of problem specific information to restructure the data. Another approach could be the Brute Force technique which is not a good practice. We have made use of certain basic assumptions related to the market activities to make changes in this stage. We progressively make changes to the data set and analyze their results with the aim to improve them further.

2.4.1) Stage 1

The Data set provided for the problem statement originally had the following structure.

- Independent Variables – product id, product category, weeks since launch, stores selling in that week and various sales data to categorical customers.
- Dependent Variables – The total sales in the nth week.

The current approach is basically dividing the dataset based on the “product_category” and training the model for each one of them separately. This goes by the intuition that the market sales patterns and demands-supply varies differently for different categories. And hence we regress separately for each category and use that model to predict the sales of a product for the 26th week.

After some study, we had already identified in the initial phase that, for a particular category of product, only the weeks since launch and number of stores selling have a major effect in predicting the total sales for that week. But this model was not exactly suitable as:

- Firstly, the sales in the 26th week apart from stores selling are also dependent on the sales in the previous weeks which were not being considered in the previous data model.
- Secondly, since in the test cases, the data provided is only for 13 weeks, the training must also not include any consumer specific sales data from beyond 13 weeks.
- The independent variable to be predicted must be the “Total Sales in the 26th week” and not the “Total Sales in the nth week”.

Hence, we have modified the data set such that we use the sales in every week upto 13 weeks along with the stores selling in week 26 as a feature set to estimate the sales in week 26. This way we can also measure the predictive power of sales in each week and how do they affect the sales in the later stages. This has been analyzed using feature selection on the new set and also through performing an Autocorrelation analysis on the ‘sales_in_nth_week’ to find the correlation between the sales series with itself for a given lag. The acf value for lag 1 was 0.8 and for lag 2 was 0.6 showing that with so much persistence, there is a lot of predictive power in the Total sales in a given week that can help us predict the sales for atleast two more weeks.Finally, the new structure of the dataset for this problem statement is as follows:

- Independent Variables – Sales in week1, Sales in week2, Sales in week3… Sales in week13, stores_selling_in_the_13th_week
- Dependent variable - Total Sales in the 26th week.

This dataset was then subjected to both Linear & Non-linear learning models. The one which performs better would then be used to train on the next phase of Data Remodeling.

2.4.2) Stage 2

Weneeded to further modify the data structure to improve results and also find a method by which we could regress the data set for all the categories together. This meant trying to find a model that allowed us to train a single model that could work on all the categories together. To do this, we used the following strategy:

1. Let ‘usn’ represent Units_sold_in_week_n and ‘ssn’ represent Stores_selling_in_week_n.

2. Now, as we had previously obtained the hypothesis from Linear Regression (Eq.. 1), in the form of:

Sales_n = (α) x Stores_n (Eq.. 4)

Whereα is the co-efficient of stores_n. Note that α is the only factor which would vary from category to category.

3. Therefore, from Eq.. 4, we get

Sales_26 = (α) x Stores_26 (Eq.. 5)

Sales_13 = (α) x Stores_13 (Eq.. 6)
=>Sales_26/Sales_13 = Stores_26/Stores_13 (Eq.. 7)
=>Sales_26= Stores_26/ Stores_13 * Sales_13 (Eq.. 8)

4.In this way, we remove the need for finding the need of the Coefficient of **Stores_n** for each category and simply keep an additional attribute '**Stores_26/ Stores_13 * Sales_13'.**

5. Also, instead of keeping only the "Stores_selling" in week 26, we decided to keep "Stores_selling" from week 14 to 26 to further incorporate the trend (if any) of the Stores_selling against Units_sold_in_week_26. This was the only useful feature provided beyond 13th week. The number of stores could help us identify the trends in the sales of the product hence further improving the accuracy of our predictions in the 26th week.

Hence the structure of our new dataset was as follows:

1. For each of weeks 1 to 13, the ratio of stores in week 26 to stores in week 13, multiplied by the sales in that week.
2. The raw sales in weeks 1 through 13
3. The number of stores in weeks 14 through 26.

The results obtained from these changes are explained later in the next section.

2.5) Understanding the Applicability of Models

Any Linear model can only be applied on a given dataset assuming that it encompasses the following properties, else it performs poorly.

1. **Linearity** of the relationship between dependent and independent variables.
2. **Independence** of the errors (no serial correlation).
3. **Homoscedasticity**[11]means that the residuals are not related to the variable plotted on X-axis.
4. **Normality** of the error distribution.

In this case we test these properties to understand and justify the performance of Linear Models against a Non-Linear Models in this domain. These tests are conducted by:

1. Linear relationship among the features is a domain based question. For example does the "sales to price sensitive customer" affect its "stores selling in nth week". Such errors can be fixed only by applying transformations that take into account the interactions between the features.

2. Independence of errors is tested by plotting the Autocorrelation graph for the residuals. Serial correlation in the residuals implies scope for improvement and extreme serial correlation is a symptom of a bad model.

3. If the variance of the errors increases with time, confidence intervals for out of-sample predictions tend to be unrealistically narrow. To test this we look at plots of *residuals vs.*

time and *residuals vs. predicted value*, and look for residuals thatincrease (i.e., more spread-out) either as a function of time or the predicted value.

4. The best test for normally distributed errors is a *normal probability plot* of the residuals. This is a plot of the fractiles of error distribution versus the fractiles of a normal distribution having the same mean and variance. If the distribution is normal, the points on this plot fall close to the diagonal line.

The results obtained in these tests are given in the next section.

2.6) Regularized Linear Regression

As we have already performed the normality tests on the data, and it is clear that the Linear Models are most likely to perform better than any other model, our next aim here is to further improve upon the accuracy obtained from the application of Linear Regression Model on this problem. To achieve this we use a concept of Regularization.

In supervised learning problems, it has been found that when we have a large number of input features as compared to the training examples, algorithm can suffer from the problem of *overfitting*(i.e., if we have too many input features, the learned hypothesis may fit the training set very well($J(\theta)$ is almost equal to 0), but fail to generalize to new examples).

Regularization is one of the most effective mechanisms to avoid overfitting problem. It basically introduces an additional term in the cost function which helps penalize modelwith extreme parameter values. A theoretical justification for regularization mechanism is that it strives to impose *Occam's razor hypothesis*(which states that "a simple hypothesis generalizes better") on the solution.

We apply the two standard regularization mechanisms in machine learning i.e., *L1 regularization*and *L2 regularization*on the data. L1 regularization uses a penalty term that makes the sum of the absolute values of the parameters small. L2 regularization makes the sum of the squares of the parameters small. Linear least-squares regression with L1 regularization is called the *Lasso*algorithm[14] and linear regression with L2 regularization is called the *Ridge* regression. Modified cost function is as follows:

$$\mathbf{J(\theta) = (1/2m) \sum (F_{\Theta}\ (x_i)\text{- } y_i)\text{^}2\ \ + \lambda|w|} \qquad \text{(Eq.. 9)}$$

Where, F_{Θ} (x_i) is the predicted value, y_i is the actual value, 'm' is the number of tuples used for training, λ is regularization parameter that has to be tuned empirically and w is model's weight vector, $\|.\|$ is either the L1 norm or the squared L2 norm.

The regularization parameter, λ, controls a trade-offbetween two different goals. The first goal is captured by the first term($\mathbf{(1/2m) \sum (F_{\Theta}\ (x_i)\text{- } y_i)\text{^}2}$) in the cost function(Eq..9)which takes care of the training set to fit well. The second goal, captured by the second term($\mathbf{\lambda|w|}$), is that parameters need to be kept small and therefore keeping the hypothesis relatively simple to avoid overfitting. Cost function for Lasso Regression is as follows:

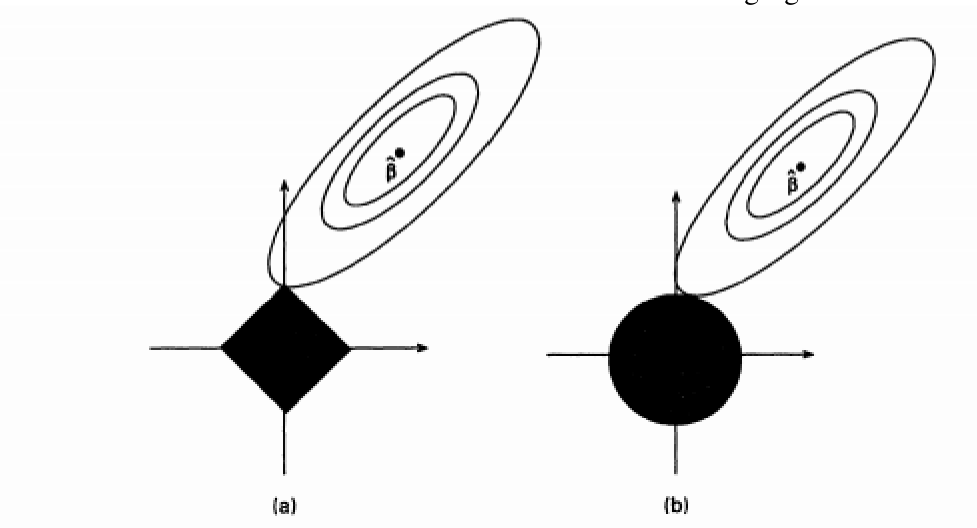
$$\mathbf{J(\theta) = (1/2m) \sum_{i=1\ to\ m} (F_{\Theta}\ (x_i)\text{- } y_i)\text{^}2 + \lambda\sum_{j=1\ to\ n} |\theta_j\ |} \qquad \text{(Eq.. 10)}$$

Cost function for Ridge Regression is as follows:

$$\mathbf{J(\theta) = (1/2m) \sum_{i=1\ to\ m} (F_{\Theta}\ (x_i)-\ y_i)\hat{}2 + \lambda\sum_{j=1\ to\ n}\theta^2_j} \qquad \text{(Eq.. 11)}$$

where n is number of input features.

Regularized regression can also be depicted as a constrained regression problem.The difference between both the mechanisms can be more described with the following figure:



Estimation picture for (a) the lasso and (b) ridge regression

From the figure, we can say that regularized regression is nothing but optimizing a data term and regularization term which balances MSE with the regularization cost. We can visualize this as the balance of two different losses. The square (in graph (a))and the circle (in graph (b)) represent the regularization costs of Lasso and Ridge regressions ,respectively, which are minimum at the center i.e. origin where all the parameters are zero (which ensures hypothesis is relatively simple and thus avoids overfitting).The contours in the plots represent different cost function values/data loss which are minimum at those dots in the figurewhich determine how well we can fit the model through the data(for the unconstrained regression model). So, the combination of the data term and the regularization term will be minimized at some point that touches both surfaces.

As mentioned earlier, L1 regularization results in sparse parameters i.e., most of the parameters are zero. A sparse vector is one that lies exactly on some coordinate axis.

L2 optimum in graph (b) will be sparse only when the minimum MSE point also lies exactly on axis which occurs with zero probability. But, in Lasso regression, L1 optimum can be on the axis even when the minimum MSE point is not on axis. Because the contour of L1 regularization cost is sharp at that point, it is possible for it to intersect with one of the data loss contours even when the minimum MSE point is not located on the axis. So, L1 regularized solutions result in incurring some level of sparsenesswhich makes the model relatively simple and saves a lot of computations. Thus, data analysts tend to choose L1 regularization over L2 regularization.

Andrew Ng has empirically compared L1 and L2 regularization. As per *Andrew Ng's Feature selection, L1 vs. L2 regularization, and rotational invariance* paper [15], L1 regularization is anticipated to perform better than L2 regularization if you have a less number of training examples as compared to the number of input features.Conversely, if your features are generated from algorithms like Principal Component Analysis (PCA), Singular Value Decomposition (SVD), or any other algorithm that assumes rotational invariance, or you have ampletraining examples, L2 regularization is anticipated to perform better because it is directly related to minimizing the VC dimension of the learned hypothesis, while L1 regularization does not have this property.

We have used R(a free software programming language and software environment for statistical computing and graphics) to implement Lasso and Ridge regressions on our data. The packages used to implement Lasso and Ridge are *lasso2* and *ridge* respectively and the functions used are *l1ce()* and *linearRidge()* respectively.

One of the important parameters of l1ce() is "bound" (a constraint(s) that is/are put onto the L1 norm of the parameters). We manually choose the optimal value of bound (i.e. 0.24) to obtain the minimum RMSE. There is no algorithm implemented in the package - lasso2 for automatically choosing an optimal value of the bound. However, there exists a number of algorithms to automate this selection process and may be implemented in the subsequent libraries in the time to come. One of those algorithms is found in the paper (*On Tuning Parameter Selection of Lasso-TypeMethods - A Monte Carlo Study*)[16]

One of the important parameters of linearRidge() is "lambda" (a ridge regression parameter). We kept its default value ,i.e. automatic, while implementing the function on the data which causes ridge regression parameter to be chosen automatically using the method (*a semi-automatic method to guide the choice of ridge parameter in ridge regression*)[17]

The results obtained after applying these methods, are given in the next section.
Every algorithm has its own demerits. Lasso fails to utilize the correlation between the input features. If there is a group of highly correlated features, then the Lasso has tendency to select one feature from the group and discard the others.

The results can be further improved upon by using another regularization method called *Elastic Net*[18] regularization (basically overcomes the limitations of Lasso regression)which linearly combines the L1 and L2 penalties of the Lasso and Ridge regressions.By incorporating ridge regression as well, it allows the L2-tendency of shrinking coefficients for correlated predictors towards each other, while keeping the feature selection property provided by the Lasso regression.

The cost function for Elastic Net is as follows:

$$\mathbf{J(\theta) = (1/2m) \sum_{i=1\ to\ m} (F_{\Theta}\ (x_i)\text{- } y_i)\text{^}2 + \lambda_1 \sum_{j=1\ to\ n} |\theta_j\ | + \lambda_2 \sum_{j=1\ to\ n} \theta^2{}_j} \qquad \text{(Eq.. 12)}$$

As we see, elastic net overcomes the limitations of Lasso by adding the quadratic part to the penalty ($\boldsymbol{\lambda_2 \sum_{j=1\ to\ n} \theta^2{}_j}$) which is used alone in ridge regression.Although this algorithm has not been used yet in our project, it can later be used to further optimize the results and adapt to any given system according to the requirements and specifications.

3. Results And Discussion

3.1 Feature selection

The list of features obtained from Greedy Stepwise feature selection [2] showed that “Stores selling in nth week” and “weeks since launch” were the most important features contributing to the prediction of sales. The variance of these features with the dependent variable, together is greater than 0.94 showing that they contribute the maximum to the prediction of sales.

3.2 Application of Linear Regression and Random Forest

Linear Regression Considering the top 6 features obtained from feature selection procedure based on their variances:

Correlation Coefficient	0.9339
Mean Absolute Error	28.37
RMSE	69.9397

Random Forests considering all the features:

OOB-RMSE	46.26

As we currently see, the non-linear model is working better than the linear model. This may lead to a jumpy conclusion that non-linear model is probably better in this scenario. Moreover the accuracy of the classifiers is also not great due to the high RMSE values of both the models.

3.3 Application of Learning Models after Data Re-modeling Phase-1

Linear Regression Results:

Correlation Coefficient	0.9972
Mean Absolute Error	0.4589
RMSE	0.9012

Random Forest Results:

OOB-RMSE	7.19

As we see here, the performance of both the models have improved drastically, however, we find that the linear model outperforms random forest. This finding compelled us to inquire about the properties of the dataset that satisfied the assumptions of the linear model. We found that:

i) The *Franke’s Anscombeexperiment*[10] to test the normality of data distribution came out inconclusive leading us to use the Normal Q-Q plot[12].
ii) The Normal Q-Q plot in R[13] concluded that the dataset follows the normal distribution.

iii) The residuals also follow the normal distribution curve under the Normal Q-Q plot just like the actual data conforming the second assumption of linearity.
iv) We check the Homoscedasticity[11] property by plotting the residuals againstfitted values. The graph was completely random in nature.
v) Lastly, the linear relationship between features is a domain specific question. The data collected mostly contains the sales data from local stores, from local manufactures of items of daily consumption types like – bread, milk_packets, airbags, etc. Since these types of products belong to a class of items where the stochastic component is negligible, it makes it easy for us to assume that the linear model can be easily applied to this problem. This is the reason why linear model is working better compared to the non-linear model due to negligible interaction of the features.

3.4 Application of Linear Models after Data Re-modeling Phase-2

Linear Regression Results considering all the new features:

Correlation Coefficient	0.9994
Mean Absolute Error	0.3306
RMSE	0.4365

Linear Regression Results considering only top 6 new features after applying feature selection on the new dataset:

Correlation Coefficient	0.99983
Mean Absolute Error	0.408
RMSE	0.7021

As we see, the results have improved further, with the accuracy of the classifier going up from RMSE value of approximately 65 to 0.43. With the final model, we were able to predict the Total sales of any given product in the test set with an error < 1 unit for any category, our best RMSE achieved being 0.43.

3.5 Application of Lassoand Ridge Regression

We have applied these algorithms on a subset of the actual training dataset with approximately 900 products belonging to 52 categories. Hence, the RMSE value for Simple Linear Regression for this subset is slightly different than that obtained for the entire dataset.

Lasso, Ridge and Simple linear regression results considering all the new features are as follows:

	Lasso Regression	Ridge Regression	Simple Linear Regression
RMSE	0.44	0.66	0.73
Mean Absolute Error	0.35	0.52	0.59

As we see clearly, Lasso regression outperforms Ridge and simple linear regressions.

It has been found frequently that L1 regularization (or Lasso regression when regularization applied to linear regression) in many models penalizes many parameters equal to zero resulting in the parameter vector being sparse. So, this becomes an obvious choice in feature selection scenarios, where we believe many input features should be eliminated.Theoretically speaking, Occam Razor hypothesis justifies that sparse or simple solutions (in case of L1 regularization) are preferable.

4. Conclusion

The primary target in machine learning is to produce the best learning models which can provide accurate results that assist in decision making, forecasting, etc. This brings us to the essential question of finding the best suitable model that can be applied to any given problem statement. We have performed a case based study here to understand on how to decide whether a linear or a non-linear model is best suited for a given application.

We initially follow a basic approach by adopting two leading classifiers from each domain and evaluate their performances. We then try to boost the accuracies of both the learning models using data re-structuring. The results obtained from this process help us derive an important empirical proof that the accuracy of a classifier not just depends on its algorithm. There is no such certainty that a more complex algorithm will perform better than a simple one. As we see in this case, Random Forests, which belong to the class of ensemble classifiers bagging based is known to perform well and produce high accuracies. However, here the simple Multiple Linear Regression model outperforms the previous one. The accuracy of the model largely depends on the problem domain where it is being applied and the data set, as the domain decides the properties that the data set would inherit and this greatly determines the applicability of any machine learning technique. Hence holding a prejudice for/against any algorithm may not provide optimal results in machine learning.

The framework developed here has been tested on real-time data and has provided accurate results. This framework can be used for the forecasting of daily use products, items of everyday consumption, etc. from local manufacturers, as it follows the assumption that the features have minimum interaction with each other. Branded products from big manufacturers include many more market variables, like the effect of political and economic factors, business policies, government policies, etc. which increase the stochastic factor in the product sales& also increase the interaction among the independent features. This feature interaction is very minimal for local products. Extending this framework to the “branded” scenario will require significant changes.However, the current model is well suited to small scale local products and can be easily used with minimal modifications, for accurate predictions.

5. Acknowledgements

We take this opportunity to express and gratitude and deep regards to our mentor – Mr. Biju R Mohan for his guidance and constant encouragement. Without his support it would not have been possible to materialize this project. His blessing, help and shall carry us a long way in the journey of life on which weare about to embark.

We are obliged to staff members of NITK IT department, for the valuable information provided by them. We are grateful for their cooperation during the project.Lastly, we thank almighty, our parents, friends and everyone involved directly or indirectly for their constant encouragement without which this project would not be possible.

References

[1] Jacky Baltes, Machine Learning Linear Regression , University of Manitoba , Canada

[2] Isabelle Guyon and Andrẻ Elisseeff ,An Introduction to Variable and Feature Selection, Journal of Machine Learning Research 3 (2003) 1157-1182

[3] Jiawei Han and Micheline Kamber, Data Mining –Conepts and Techniques , Second Edition Page-79-80

[4] Kris Hauser , Document B553, Gradient Descent, January 24,2012

[5] Luis Carlos Molina, Lluís Belanche, Àngela Nebot ,Feature Selection Algorithms: A Survey and Experimental Evaluation, University Politècnica de Catalunya

[6] Quang Nhat Nyugen, Machine Learning Algorithms and applications, University of Bozen-Bolzano, 2008-2009

[7] Jan Ivar Larsen, Predicting Stock Prices Using Technical Analysis and Machine Learning, Norwegian University of Science and Technology

[8] Classification And Regression By Random Forest by Andy Liaw And Matthew Wiener, Vol 2/3, December 2002

[9] Lecture 22,Classification And Regression Trees, 36-350, http://www.stat.cmu.edu/, November 2009

[10] F. J. Anscombe. Graphs in Statistical Analysis. The American Statistician, Vol. 27, No. 1. (Feb., 1973), pp. 17-21

[11] [Online] Homoscedasticity-http://en.wikipedia.org/wiki/Homoscedasticity

[12] [Online] Normal Q-Q Plot-http://en.wikipedia.org/wiki/Q–Q_plot

[13] [Online] Quick R-http://www.statmethods.net/advgraphs/probability.html

[14] Robert Tibshirani(1996). Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society. Society B(Methodological), Volume 58, Issue 1 (1996), 267-288.

[15] Andrew Y. Ng.Feature selection, L1 vs. L2 regularization,and rotational invariance.ICML '04 Proceedings of the twenty-first international conference on Machine learning, Page 78

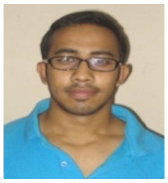
[16] Sohail Chand. On Tuning Parameter Selection of Lasso-TypeMethods - A Monte Carlo Study. Proceedings of 2012 9th International Bhurban Conference on Applied Sciences & Technology (IBCAST) Islamabad, Pakistan, 9th - 12th January, 2012

[17] Erika Cule and Maria De Iorio. A semi-automatic method to guide the choice of ridge parameter in ridge regression.arXiv: 1205.0686v1 [stat.AP] 3 May 2012

[18] Hui Zou and Trevor Hastie. Regularization and variable selection via theelastic net.J. R. Statist. Soc. B (2005) 67, Part 2, pp. 301–320

Authors

Gaurav Singh Thakur



Gaurav Singh Thakur has completed his B.Tech in 2014 in Information Technology from National Institute of Technology Karnataka, Surathkal and is currently working as a Software Engineer at Cisco Systems, Inc. Bangalore. His technical areas of interest include Machine learning, Networking & Security, Application Development and Algorithms.

Anubhav Gupta

Anubhav Gupta has pursued his Bachelors, at National Institute of Technology Karnataka, Surathkal in the Field of Information Technology (IT) and graduated in 2014. His technical areas of interest include Machine learning, Information Security, Web Development and Algorithms.Currently, he is working as Software Developer Engineer at Commonfloor (MaxHeap Technologies).

Ankur Bhardwaj

Ankur Bhardwaj completed his B.Tech in Information Technology from National Institute of Technology Karnataka, Surathkal in 2014. His technical areas of interest include Machine Learning, Statistics and Algorithms. Currently, he is working as Associate Professional at Computer Sciences Corporation.

Biju R Mohan

Mr. Biju R Mohan is an Assistant Professor at National Institute of Technology Karnataka, Surathkal at the Dept. of Information Technology. His technical areas of interest include Software Patterns, Software Aging, Virtualization, Software Engineering, Software Architecture & Requirements Engineering.