LEARNING NONLINEAR FEATURES TO IMPROVE LINEAR FORECASTING APPROACHES

by

Mert Öz

B.S., Aeronautical Engineering, Istanbul Technical University, 2014

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering
Boğaziçi University

2019

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my advisor Assist. Prof. Mustafa Gökçe Baydoğan for his continuous support, patience, and encouragement from the beginning until the end. I am grateful to study with him. I would never have succeeded without his support and faith. He was always solution-oriented and supportive.

I also would like to thank to Prof. Refik Güllü and Assoc. Prof. Özden Gür Ali for their participation in my thesis jury and for their valuable comments and suggestions.

Especially thanks to my family; my mother and brother for their continuous encouragement, patience, and belief in me. Their endless love and support helped me a lot to achieve graduation from Boğaziçi University. I would like to dedicate my thesis to the memory of my father who always believed in me and supported me every time.

Finally, I would like to thanks to all of my friends who always supported me and helped me to increase my spirits when I feel down. I am lucky to have them.

ABSTRACT

LEARNING NONLINEAR FEATURES TO IMPROVE LINEAR FORECASTING APPROACHES

Forecasting of future events is critical for decision making in many industries. Especially in the retail industry, forecasting of future sales has critical importance for management of the company. General time series models are a widely used method for forecasting. However, since general time series models mostly consider linear relation between response and explanatory variables, they can miss nonlinear relations which can have a critical effect on the response variable. We propose an iterative approach that starts with a base model and explain the residuals by tree-based regression. The path leading to the highest error is added to the base model as a new variable. Proposed algorithm is an improvement on general time series models since it adds nonlinear variables by residuals explanation to the linear models in the second stage. Proposed model consists of two-stage; first stage is a general time series model where Autoregressive Integrated Moving Average with regressor version (ARIMAX), Linear Regression and Penalized Regression models were used as base learner in this study, second stage is a residual explanation by regression tree to find new explanatory variables that cause the highest error on the first stage by considering linear and nonlinear relations. New regressors which are found on the second stage are added to the first model and new model continues for forecasting until it optimizes itself. Implementation of proposed algorithm on ARIMAX outperformed on regular ARIMA model and ARIMAX model with same regressors on the proposed model. Also, proposed algorithm was implemented on Linear Regression and Penalized Regression methods and when compared with regular Linear Regression and Penalized Regression respectively, proposed algorithm achieved better results.

ÖZET

DOĞRUSAL OLMAYAN ÖZELLİKLERİN ÖĞRENİLEREK DOĞRUSAL TAHMİNLEME YÖNTEMLERİNİN GELİŞTİRİLMESİ

Birçok endüstride gelecekte karşılaşılacak olayların düzgün tahmin edilmesi kritik rol oynar. Özellikle perakende sektöründe, gelecek satışların tahmini, şirketin yönetiminde alınacak kararlara ışık tutar. Tahminleme için kullanılan en yaygın geleneksel zaman serisi modelleri değişkenler arasındaki sadece lineer ilişkileri açıklayabildiğinden, lineer olmayan ilişkilerin yarattığı kritik etkileri kaçırabiliyorlar. Önerilen algoritmanın ikinci aşamasında hatalar analiz edilerek değişkenler arasındaki lineer olmayan ilişkilerin de hesaba katılıp, modele yeni değişkenler ekleyerek genel zaman serisi modelleri üzerinde geliştirme sağlıyor. Önerilen yaklaşım iki aşamadan oluşuyor; birinci aşamada genel zaman serisi modeli kullanılarak tahminleme yapılıyor, ikinci aşamada ise ilk modeldeki tahminleme sonucunda çıkan hatalar regresyon ağacı altında lineer ve lineer olmayan ilişkiler göz önünde bulundurularak incelenerek en çok hata veren değişkenler birinci modele ekleniyor. Model bu şekilde belirlenen limitleri sağladığı sürece kendini güncelleyerek geliştiriyor. Bu çalışmanın birinci aşamasında ARIMAX, Lineer Regresyon ve Cezalı Regresyon (Penalized Regression) modelleri kullanılarak bu modeller üzerinde gelişim sağlandı. Algoritmanın uygulandığı ARIMAX modeli, hiçbir bağımsız değişkenin eklenmediği ve önerilen algoritmada kullanılan aynı bağımsız değişkenlerin eklendiği iki modelden de büyük oranda daha iyi sonuçlar verdi. Ayrıca, algoritmanın test edildiği lineer regresyon ve cezalı regresyon modelleri de aynı değişkenlerin olduğu klasik lineer regresyon ve cezalı regresyon ile kıyaslandığında daha iyi sonuçlar verdi.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

b	Trend factor
B	Lag operator
c	Constant
d	Non-seasonal differencing order
D	Seasonal differencing order
E_t	Residual matrix
I	Seasonal index
J_β	Elastic net penalty
$\hat{L}$	Maximum likelihood value
M	Moving average result
n	Number of observation of response variable
N	Number of observation used for moving average
o	Number of independently adjusted parameters
p	Autoregressive lag order
q	Moving averages order
R^2	Coefficient of determination
R^2_{adj}	Adjusted coefficient of determination
R_m	Sum of divided parts in decision tree
s	Optimal splitting point in decision tree
S	Seasonal length
S_t	Smoothed observation at time t
X	Model matrix
x_i	i_th regressor of response variable
y_t	Variable of interest at time t
$\hat{y}$	Predicted value of response variable
y_{t-p}	Lags for variable of interest until time $t-p$
$y_t{}'$	ARMA(p,q) process

α,ω,γ	Smoothing weights
β	Regression coefficient
$\hat{\beta}$	Elastic net estimator
λ	Regularization coefficient
μ	Mean of the data
ε_t	Random error
φ_t	Autoregressive model parameters
θ_j	Moving Average model parameters
$\theta_q(B)$	Non-seasonal MA operator with order q
$\Theta_Q(B^S)$	Seasonal MA operator with order Q
$\phi_p(B)$	Non-seasonal AR operator with order p
$\Phi_P B^S$	Seasonal AR operator with order P

LIST OF ACRONYMS/ABBREVIATIONS

ADL	Autoregressive Distributed Lags
AIC	Akaike Information Criterion
ANN	Artificial Neural Networks
AR	Autoregressive Model
ARIMA	Autoregressive Integrated Moving Average
ARIMAX	Autoregressive Integrated Moving Average with Exogeneous Input
ARMA	Autoregressive Moving Average
CART	Classification and Regression Trees
ERNN	Elman's Recurrent Neural Networks
FNN	Feedforward Neural Network
MA	Moving Averages Model
MLP	Multi-Layer Perception
MLPNN	Multi-Layered Perception Neural Network
MSE	Mean Squared Error
MSFE	Mean Squared Forecast Error
RMSE	Root Mean Squared Error
RMSPE	Root Mean Squared Percentage Error
SARIMA	Seasonal Autoregressive Integrated Moving Average
SARIMAX	Seasonal Autoregressive Integrated Moving Average with Exogeneous Input
SKU	Stock Keeping Unit

1. INTRODUCTION

Forecasting is the use of historical data to predict future events with the most appropriate model or judgement. Forecasting has usage in many areas such as environmental and social sciences, economics and finance, politics, business and more. Prediction of future is important to make decision about future in all of those areas. Especially in retail business which is the subject of this work, marketing, procurement, finance, sales, production, accounting and human resource management departments need forecasts to create necessary information for future decisions (Mentzer *et al.*,, 1998). On the operations management side, sales or demand forecasts of products is necessary for production planning, inventory control, supply chain management, labor force and production locations. On the marketing side, sales forecasts are important for advertising, promotion and price decisions. Sales forecasts are also important feature for finance and risk management. Investors can predict returns of their investments by analyzing demand or sales forecasts (Montgomery *et al.*, 2015). Effective sales forecasting applications can increase profit and decrease costs in retail industry by developing business strategy quality such as increasing customers' contentment, decreasing failure of products and clear cut production plans (Aye *et al.*, 2015).

Forecasting problems can be divided in three categories (Montgomery *et al.*, 2015);

- Short-term: Hourly, daily, weekly or monthly forecast
- Medium-term: Up to two years
- Long-term: More than two years

Short-term forecasting problems are generally needed for operational reasons such as production planning, stock control, manpower scheduling, logistic requirements, etc. Short-term forecasts are made for a period up to one year for the current operation. Medium-term forecasts are needed for tactical decisions to decide future resources such as personnel hiring, equipment purchase, etc. Medium-term forecasts are critical

for managing the budget of the company. Long-term forecasts for major strategic decisions such as investments. It requires consideration of exogenous factors such as government policy, competitors behaviors, economic conditions, etc (Montgomery *et al.*, 2015). Different than long-term forecasting problems, solution of short and medium-term forecasting problems are usually depends on historical data. Therefore, statistical methods are highly useful for short and medium-term forecasts by analyzing and deducing historical data (Montgomery *et al.*, 2015).

Generally, there are two main approaches to perform forecasting; qualitative and quantitative techniques. Introducing new goods without any historical data need qualitative forecasting techniques which require expertise interpretation. On the other hand, quantitative models extrapolate historical data with a forecasting model (Montgomery *et al.*, 2015). In both cases, time series data which is equally spaced time-oriented observations are generally used. Especially in retail sales forecasting, univariate time series are involved in forecasting problems (Gur Ali *et al.*, 2016). In univariate time series, single output is observed in equal time intervals. On the other hand, in multivariate time series, more than one output is observed in each time. This thesis will focus on univariate time series in forecasting tasks.

In general, quantitative forecasting models can be divided in two categories; regression based models and general time series models. Regression models analyze relation between independent variables and dependent variable. On the other hand, general time series models create a model from statistical information in the past data (Montgomery *et al.*, 2015). Widely used general time series model is smoothing techniques in which forecasting is provided with a function applied to previous observations. Smoothing techniques can be categorized as Moving Average (MA) techniques and Exponential Smoothing Techniques. Both of those smoothing techniques are easy to understand and apply. If data has not a trend, smoothing techniques generate accurate forecasts. Also, there are some smoothing techniques as Holt-Winter's method which can overcome trend and seasonality effect. Smoothing techniques gives more importance on recent observations and update itself when new data become available. However, when data do not show same historical pattern, smoothing techniques have

less accurate forecasts. Since smoothing techniques use historical data, there should be sufficient number of data available to make accurate forecasts.

Another widely used general time series model is Box and Jenkins (1976) Autoregressive Integrated Moving Average (ARIMA) model. ARIMA model combines Autoregressive (AR) and Moving Average (MA) models. AR model makes forecasts with a linear regression between current value and lags of time series. MA model also makes forecast with a linear regression between the current value and white noise of the lags of the time series. When time series has a seasonal pattern, another version of ARIMA model called Seasonal Autoregressive Integrated Moving Average (SARIMA) is used for more successful forecasts. Another version of ARIMA which is also used as a base model in the first stage of proposed approach is Autoregressive Integrated Moving Average with regressor version (ARIMAX) which is used when independent variables are available. In ARIMAX model, response variable is analyzed for forecasting with its past observations and independent variables. Similar to ARIMAX, Seasonal Autoregressive Integrated Moving Average with Exogeneous Input (SARIMAX) model is useful when seasonality occurs in data. Unlike the smoothing techniques, ARIMA model cannot update itself when new data available. Main advantage of ARIMA model is that it works well on short-term forecasting. Also, ARIMA model is less sensitive to data fluctuations regarding to other general time series models. Main weakness of ARIMA models is assumption of linear relationship between input variables and output variables. In other words, ARIMA model cannot explain nonlinear relations (Aburto and Weber, 2007). In this study, we propose an algorithm that adds nonlinear relations into linear models to create more accurate forecast.

ARIMA and Smoothing techniques are basic time series models which only use past data. Both of the approaches use lags to create a model and make predictions. However, in many cases there are other variables which affect response variable. In such cases, only past observations of response variable will not explain exact statistical pattern behind the data. As explained above, regression approaches analyze relationship between input variables and response variable and create a model to forecast future response variables when input variables are available. Linear regression model which

is extremely simple method is mostly used in this area. However, linear regression is limited to analyze linear relationship. Also, linear regression is sensitive to outliers. Regression models try to find best explanatory variables i.e. regressors which have effects on response variable. Although this is the main purpose of regression models, in some cases they may fail to find the best variables. In this study, we analyze residuals of a basic forecasting method to find best suitable regressors which affects response variables most.

There are some hybrid models which uses classical time series models as predictors. For example, Aburto and Weber (2007) combine ARIMA models and neural networks and create a hybrid model for a retail sales forecast. Firstly original time series is modelled by an ARIMA model and residuals of ARIMA model is considered as another time series and modelled by a neural network model. Hybrid model's forecast is sum of ARIMA forecast and residual forecast of neural network model. Gur Ali *et al.*, (2016) also provides an two stage method where in the first stage seasonality, marketing and calendar effects estimated using a panel-regression model and residuals of first stage model is extrapolated in second stage. General forecast is combination of marketing, calendar and seasonality parameters which estimated in first stage plus estimated residuals adjusted in second stage. In general, both of those models focus on the prediction of residuals.

Khashei and Bijari (2011) made a research to find an optimal methodology for combining linear and nonlinear approaches. Since many of real-life forecasting problems mostly have nonlinear relations, linear models cannot capture pattern in those data. On the other hand, although some nonlinear models have accurate forecasting results in particular situations, they have poor prediction performance in other cases. For example Artificial Neural Networks (ANN) can capture nonlinear relations, however, it has not consistent performance on some linear problems. Therefore, hybrid models with both linear and nonlinear capabilities are better for real problems compared with the individual model. According to Zhang (2007), forecasting accuracy is increased when the difference between combined methodologies is more.

In the retail industry, forecasting of future sales is difficult since there are many variables such as; historical sales data, promotions, competitors' behaviors, advertises, seasonality, weather, holidays, prices, demographical and geopolitical situations that effect sales (Aburto and Weber, 2007). There are also some unusual variables which can have critical effect on sales such as Christmas week, national holidays, match games etc. or instantaneous sales variations due to special days such as elections, new movie release, etc. Among those variables, especially promotions have an important effect in sales. According to Gur Ali *et al.*, (2009) study, characteristics of promotions should be analyzed very carefully to obtain successful forecasts. Marketing studies' main goal is to demonstrate which factors cause sales increment not to find exact number of sales. In many cases, nonlinear relationships among independent variables also have a huge effect on sales. For example, sales are increased almost every time when retail has a promotion. Also, sales are increased on holidays since people go shopping on holidays more. However, when it is a holiday and retail has a promotion, sales can be increased more than a holiday time plus a promotion time sales. Therefore, interaction effects between predictors should also be taking into account to forecast accurate sales. Basic time-series and regression based models do not consider nonlinear relationship.

In this study, we propose a two-stage model with error exploration to find best regressors of the original time series. Firstly, sales are forecasted with a classical time series method or a regression based method. In the second stage, residuals of first model is analyzed with a regression tree with all original input variables and new variable which generates highest error in first stage added to first model. The model tries to add new variables to first model iteratively until first model cannot be improved anymore. Model's stopping algorithm works when regression tree is not able to create a meaningful tree or Mean Square Error (MSE) cannot be improved enough. The second stage of proposed model is similar with boosting algorithms. Boosting algortihms fit a decision tree on residuals of fitted models and tries to improve fitted models by adding new trees to update residuals. Since each time weight of the current residuals are increased, model tries to avoid making same errors and improves itself (James *et al.*, 2013). In our approach, highest residual source is detected in second stage and base model improved with adding new variables. Therefore, previous residual

time series changed by adding new variables. Basic time series models analyze trends, seasonality and cyclic components of the time series and leave unpredicted components as noise. Proposed approach can identify interactions between the predictor variables with second phase's regression tree by learning from residuals of first phase to model unpredicted components in classical time series models or regression based models. For example; if there is an interaction between a holiday and a promotion day variables which creates more than their single effect and causes highest errors in the base model, a new variable which states a promotion in holiday is included in base model. Thus, nonlinear relations are added to a linear model. We evaluated proposed approach on both synthetic data with 120 days of observation and a retailer data with 1115 stores and two years seven months daily sales. In this thesis, ARIMAX, Linear Regression and Penalized Regression models have been used in the first stage as base learner and both models have been improved with proposed algorithm.

Our approach has some differences from existing forecasting methods with residual explanation. Mainly, we focus on sources of current residuals to find new regressors for a base model, while, as Gur Ali *et al.*, (2016) and Aburto and Weber (2007) studies, they focused on forecasting future residuals. Our model starts with a general time series or regression-based model with least possible independent variables and develops itself by explaining source of residuals. In another words, proposed approach develop itself from a simple model to more complex model. On the other hand, similar works start with fully developed base model. In literature, as explained on the above works, generally future residuals are forecasted by some methods. In our study, residual explanation performed with regression tree. However, base learner models can be selected from different models. Different than those models, we focused on current residuals to improve our base learner in the first stage by considering linear and nonlinear relations between independent variables. We analyze nonlinear relationship in second phase by learning from residuals. Main philosophy in the proposed algorithm is that it improves itself by learning from its past errors.

This thesis is organized as follows; Chapter 2 summarizes related literature about general time series and regression-based models, Chapter 3 explains general formulation and properties of general time series and regression-based models which were used in our approach, Chapter 4 details proposed algorithm's methodology, Chapter 5 and Chapter 6 are analytical chapters which show implementation of proposed model on synthetic and real dataset respectively, as final, Chapter 7 provides conclusions for thesis and comments for possible future works.

2. LITERATURE REVIEW

Forecasting methods have been used in many areas as explained in Section 1. Therefore forecasting methods have been studied in many aspects for years. Research on classical time series methods have been overviewed in this study which is main subject of this work.

Alon (1997) compared general exponential smoothing and Holt's models with Winters' exponential smoothing on aggregate retail sales and found that Winters' method has more accurate forecasts. After four years, Alon *et al.*, (2001) compared classical time series models such as Winters' exponential smoothing, ARIMA models and multivariate regression with Artificial Neural Networks (ANN). Due to capability of explanation of nonlinear relations and seasonality, ANN has better performance on forecasting than those methods.

Duncan *et al.*, (2001) examine Bayesian pooling models and reveal advantages of pooling time series on short term forecasting. According to their study on Bayesian pooling models, pooling time series decrease parameter requirement for estimation while increasing forecasting precision. Also, depending on the pooling technique, pooling time series can be built on general time series models for stationary and nonstationary time series, overcome outlier observations and flexible to rapid changes in time series. They also suggest that pooling data can help explanation of forecast due to similar trend or seasonality effects on different groups.

Chintagunta *et al.*, (2003) focuses on pricing and customer welfare effects on profits for a supermarket chain. They suggest a pricing policy based on store location. According to their study, location based pricing policy rather than fixed-pricing increases retails profits based on product category. Although some customers segments are affected negatively on location based pricing, retail's profit still increases in general.

Gur Ali *et al.*, (2009) examine ways of improvements in forecasting accuracy with increasing model complexity and data preparation cost by changing forecasting technique, model scope and input variables on SKU-store level sales and promotion time series. They worked on 76 weeks long SKU-store level sales and promotion time series dataset from a grocery store in Europe with sales of a single category with 4 sub-categories in 4 stores. According to their study, pooling data increases forecasting accuracy in almost every case. They also demonstrate that classical time series models have best performance on time series without promotion. However, when promotion is involved, regression tree with basic input variables has up to 65% better performance. More complicated input variables have benefits for model performance if chosen model can handle them. Otherwise, sophisticated input variables only increase model cost and have no positive effect on model performance (e.g with linear regression).

There are some hybrid models which explain nonlinear relations between input and output variable. Khashei and Bijari (2011) compared three different hybrid models; generalized hybrid ANNs/ARIMA model, Zhang's hybrid ANNs/ARIMA and Artificial Neural Network (p, d, q) models which are a combination of linear and nonlinear models. According to their literature view, both empirical and theoretical results shows that combining different models, especially linear and nonlinear models have better performance than their individual results. Zhang (2003) proposed a hybrid model with ARIMA and ANN. Zhang assumes that time series are a combination of linear and nonlinear components. In his model, firstly ARIMA analyzes linear structure of the time series. Residual of the ARIMA model is modelled by ANN and nonlinear relations extracted. Final forecast is a combination of linear component's forecast by ARIMA and nonlinear component's forecast by ANN. However, Zhang's hybrid model has some disadvantages due to its assumption. Firstly, he assumes that residuals of ARIMA model will only have a nonlinear structure. Secondly, he assumes that linear and nonlinear components can be divided and analyzed separately by ARIMA and ANN and then combined. Thirdly, there is an additive relationship between linear and nonlinear components of a time series. Since those assumptions are not valid for every case, Zhang's hybrid model can fail in other applications (Khashei and Bijari, 2011). Another hybrid model with a combination of ARIMA and multilayer perceptron is proposed to handle

the above assumptions in their work is artificial neural network (p, d, q). It is similar to Zhang's hybrid model since residuals of ARIMA model is modelled with a neural network. The difference is in the second stage; residual of ARIMA model is modelled by ANN with original observations to find also nonlinear and linear relations. The main advantage is that no assumption is made in ANN (p, d, q) as in Zhang's hybrid model. Also, it has better performance than using ARIMA and ANN separately. However, it is not robust in all cases compared with Zhang's hybrid model. Lastly, a more accurate hybrid model is proposed as the generalized hybrid ANNs/ARIMA model. This model also has no assumption as in Zhang's hybrid model. First stage is similar to the previous one; aiming to find linear relations. Second stage's main goal is to find nonlinear relations. ANN is applied in the second stage to find both linear and nonlinear relations in the residuals of the first ARIMA model and in the original data. Different that ANN (p, d, q) approach, generalized hybrid ANNs/ARIMA model uses linear component and original data together to analyze linear structure of the model. They implemented all three hybrid models on three different data sets; the Canadian lynx data, United States dollar vs British Pound exchange rate data and Wolf's sunspot data. According to their study, generalized hybrid ANNs/ARIMA model have better forecasting results than other two hybrid models. On the other hand, Taskaya and Casey (2005) made a comparative study on ARIMA and ANN hybrid models in nine different data sets and found that five out of nine data sets, components of hybrid models outperform than those hybrid models. They have commented that a single model can also outperform ARIMA and ANN hybrid models due to assumption's explained above. Therefore, model selection has a critical role in hybrid models while combining two different approaches.

Aladag *etal.* (2009) improved Zhang's hybrid model (2003) by using combining ARIMA with Elman's Recurrent Neural Networks (ERNN) instead of Feedforward Neural Network (FNN). Proposed model is applied on Canadian Lynx data as Zhang did and forecasting accuracy have been improved with proposed model.

Aburto and Weber (2007) also combined ARIMA models and neural networks which uses forecasting residuals and propose a hybrid intelligent system to forecast demand for a Chilean supermarket. They choose a seasonal autoregressive integrated moving average model with exogenous inputs (SARIMAX) which incorporates seasonality and outcome best performance for this dataset among other ARIMA models and multi-layer perception (MLP) method among neural networks for their hybrid model. Their hybrid model increase forecasting accuracy and decrease inventory level for the several SKUs in the Chilean supermarket. In their hybrid model, firstly original time series is modelled with SARIMAX and forecasting error is considered as another time series and modelled by MLP. Hybrid forecast is sum of original time series forecast with SARIMAX and error term forecast by MLP. They use special days to reflect promotion effect. There are nine special day variables used as input variable such as monthly and two-weekly payment days, New Year, summer days and etc. In their study, it has been proved that ARIMA models have worse performance compared to neural networks while their hybrid model has the best performance among basic neural networks and ARIMA models. Their study is similar to our approach by considering residual exploration. However, our approach differ by method on analyzing residuals. Aburto and Weber tried to predict possible errors, on the other hand, our approach is like a boosting algorithm which tries to find new variables that cause the highest errors on general forecasting models and add those variables to our model.

There are also other hybrid models in literature based on SARIMA and ARIMA. Cools *et al.*, (2009) forecasted daily traffic counts with both ARIMAX and SARIMAX techniques. Seasonality and holiday effects included in the model by ARIMAX and SARIMAX methodology. Their work deduced that ARIMAX and SARIMAX models are important to identify possible variables which have an impact on traffic counts. Cornelsen and Normand (2012) forecasted the affect of smoking ban on sales in bars located in Ireland with ARIMAX model. They found out that smoking ban has negative impact on sales by also considering general econometric trends and prices. Chikobvu and Sigauke (2012) developed SARIMA and regression-SARIMA models (regression with SARIMA errors) for demand of daily peak electricity in South Africa. Calendar effects have been included in regression-SARIMA model. They compared developed

SARIMA, regression-SARIMA approaches with Winter's triple exponential smoothing model and revealed SARIMA models had better results in short-term predictions. On the other hand, they demonstrated that regression-SARIMA model catches important features on demand data and results can be improved by adding other independent variables such as weather effects. Therefore, they considered regression-SARIMA model has robust results relatively SARIMA and Winter's triple exponential smoothing model. Kongcharoen and Kruangpradit (2013) compared integrated autoregressive moving average (ARIMA) model and ARIMA with explanatory variables (ARIMAX) model on Thailand exports forecasting. In their study, they use Mean Squares Forecast Errors (MSFE) to compare forecasting performance of both models. Due to addition of explanatory variables, ARIMAX performs betters results in country-level data. For commodity-level data, they found that both models performance do not have a critical difference. As a result of their study, they suggests ARIMAX models for export data forecasting due to better performance. Arunraj and Ahrens (2015) combined SARIMA and quantile regression models for food sales forecasting. To overcome overfitting or underfitting possibility created by SARIMAX, they proposed a hybrid model with SARIMA and Quantile regression (SARIMA-QR) which also forecasts the quantiles instead of point forecasts. When compared with traditional SARIMA, seasonal naive forecasting and Multi-layered Perceptron Neural Network (MLPNN), their hybrid model has better performance on daily sales prediction of a banana retailer.

Gur Ali *et al.*, (2016) also create a model with residual extrapolation for a major Turkish retailer with 363 stores and seven product categories. Their multi-period sales forecasting method called as "Two-Stage Information Sharing Model", the first stage model category/store level sales with seasonality, calendar and marketing variables, while second stage extrapolate residuals of the first stage with information sharing between stores. Multi-ahead aggregated sales forecasting is generated with first stage regression model and arranged with second stage model. Forecasting accuracy has been increased with information sharing between stores by categories and categories by stores. In the Stage 2, differently from AR models which assume errors are correlated serially, residuals extrapolation is performed without any assumptions. Second stage has improved model's forecasting accuracy in comparison to model with only first

stage forecasts. Also, they proved that their "Two-Stage Information Sharing Model" has better performance than mixed models (panel regression models), autoregressive distributed lags (ADL) and Winters' exponential smoothing methods. The methodology in their approach has some similarities with ours in terms of a two-stage algorithm and residual-based model developments. However, while we are evaluating residuals of the first model to add new variables, they estimate future residuals by extrapolating first stage model residuals. Also, while they use store category and assortment variables and make category-based forecasts, we exclude store type and assortment type variables from our model.

According to our literature review, hybrid models are popular on time series forecasting since they use each capability of its components. Generally, ARIMA and ANN hybrid models have been proposed in many different types of time series. Our approach has some differences in current hybrid models. Firstly, we did not use ANN on our approach as a nonlinear model. We have tested our approach with three different linear models; ARIMAX, Linear Regression, and Penalized regression and we have proved all three hybrid models have better performance than base learners individually in general. Also, our algorithm can be used with different linear or nonlinear models. We analyzed residual of the linear models in the second stage with a regression tree to find the path for the highest error. As a result of the second stage, linear models improve itself by adding new variables which can be linear and nonlinear. Different than Zhang's hybrid model, we do not have any assumption on residuals to be nonlinear only. Also, our approach has some differences from existing forecasting methods by analyzing linear model's residuals. In many existing models (as in Gur Ali and Pinar (2016) and Aburto and Weber (2007) studies), residuals are extrapolated. On the other hand, our aim is to explain residuals of linear models by searching the path for the highest errors.

3. BACKGROUND

3.1. General Time Series Models

In this section, general properties and formulations of time series models which were used in our model will be explained. Linear time series models are generally used in time series forecasting. Most popular models in this area are Autoregressive (AR) models, Moving Average (MA) models, Autoregressive Moving Average (ARMA) which is the combination of AR and MA models (Fan and Yao, 2003).

Time series have four main components; trend, seasonality, cyclical components and irregular/random components which are defined as noise. Main focus of the general time series models such as Smoothing techniques, AR, MA, ARMA etc. is to model all other components correctly except random component which can't be estimated and leave random components as possible as predictable. Components of a time series are explained by Coghlan (2018) in the Figure 3.1 below;

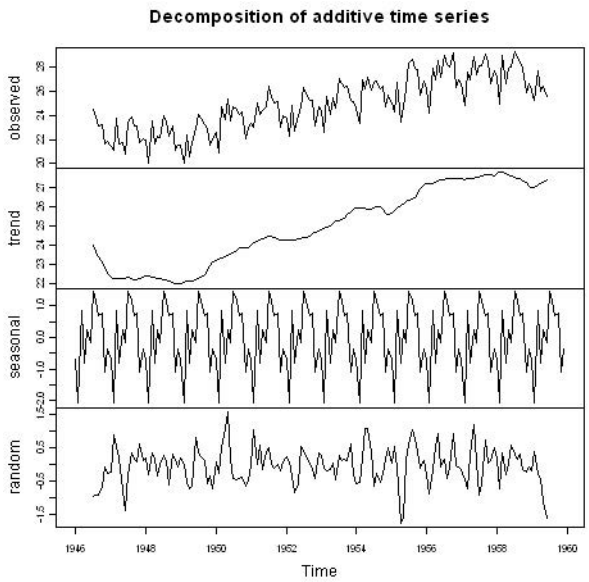

Figure 3.1. Components of a Time Series (from Coghlan, 2018, p. 25)

Smoothing techniques are other widely used method in time series forecasting. Smoothing techniques demonstrate trend, seasonality and cycling in the time series clearly. Smoothing methods generally divided into two groups; Moving Averages and Exponential Smoothing Methods. If data has no trend, simple average of all the past observation is a simple, quick and easy to use estimate for forecasting. However, real world time series data are generally have trend, seasonality or cycling. Therefore, moving average methods which only average smaller set of past observation is generally give better results. Single Moving Average process is formulated Eq. 3.1 below;

$$M_t = \frac{y_t + y_{t-1}.....y_{t-N+1}}{N} \tag{3.1}$$

where y_t is variable of interest, M is moving average result, N is number of observation. However, Single Moving Average method is not useful when data has a significant trend. Another form of Moving Average method, called as Double Moving Average has better performance when data has trend. In Double Moving Average method, second moving average is calculated from the first moving average. In Single and Double Moving Average methods, averaging past observations weighted equally.

There is another smoothing technique where weights assigned exponentially decreasing is called as Exponential Smoothing Technique. In this method, recent observations geometrically declined weighted over time where last observations have highest importance to forecast future events. Exponential smoothing method provided by Holt (1957) for time series that have no trend and seasonality. Then Winters improved the model in 1965 for seasonal time series and called "Holt-Winter Method". Winters' model is also called Triple Exponential Smoothing where model formulate simple smoothing, trend effect and seasonality effect (Alon *et al.*, 2001).

General formulation for Holt-Winter Method is explained in Eq. 3.2;

$$\begin{aligned} S_t &= \alpha\frac{y_t}{I_{t-L}} + (1-\alpha)(S_{t-1}+b_{t-1}) && OverallSmoothing, \\ b_t &= \gamma(S_t - S_{t-1}) + (1-\gamma)b_{t-1} && TrendSmoothing, \\ I_t &= \omega\frac{y_t}{S_t} + (1-\omega)I_{t-L} && SeasonalSmoothing, \\ y_{t+m} &= (S_t + m(b_t))I_{t-L+m} && Forecast \end{aligned} \tag{3.2}$$

where y is the observation, S is the smoothed observation, b is trend factor, α, γ and β are weights that need to be determined before for minimum MSE and I is the seasonal index. Smoothing techniques decrease outliers' effects and understand trends with parameter adjustments (Duncan *et al.*, 2001).

AR(p) model is linear regression of output variable with its p order past observations. AR models assume a linear relationship between variable of interest and its past observations (Adhikari and Agrawal, 2013). These models are mostly used linear time series model due to adaptation of different time series patterns. (Fan and Yao, 2003). Mathematical formulation of AR(p) model can be expressed as Eq. 3.3 (Adhikari and Agrawal, 2013);

$$y_t = c + \sum_{i=1}^{p} \varphi_i y_{t-i} + \varepsilon_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} ..... + \varphi_p y_{t-p} + \varepsilon_t \tag{3.3}$$

where y_t is variable of interest at time t, y_{t-p} are lags, c is a consant, ε_t is random error and φ_t (i = 1, 2,.., p) are model parameters. Due to linearity and stationarity requirements, AR(p) models are not effective in real life problems where non-linearity and non-stationarity data are observed (Cheng *et al.*,, 2015).

MA(q) model is linear regression of output variable with its q order past forecasting errors. MA models assumes random shocks (random errors) are a white noise. Application of MA model is more complicated than AR model due to the white noise is unobservable (Fan and Yao, 2003). Mathematical formulation of MA(q) process is

given in Eq. 3.4 (Adhikari and Agrawal, 2013);

$$y_t = \mu + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t = \mu + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} ..... + \theta_p \varepsilon_{t-p} + \varepsilon_t \tag{3.4}$$

where μ is the mean of the data, θ_j (j = 1, 2, ..., q) are model parameters, ε_{t-j} are random errors which assumed to be white noise and y_t is variable of interest at time t.

ARMA (p, q) models which are the combination of AR(p) and MA(q) models are one of the widely used time series model due to adaptation of different stationary time series. ARMA (p, q) model assumes linearity and stationarity in lags and output variable as AR(p) and MA(q) models (Fan and Yao, 2003). General expression of ARMA(p, q) model is addressed in Eq. 3.5 below (Adhikari and Agrawal, 2013);

$$y_t = c + \sum_{i=1}^{p} \varphi_i y_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \tag{3.5}$$

where y_t is variable of interest at time t, y_{t-p} are lags, ε_{t-j} are random errors, c is constant, p and q are orders, φ_i (i = 1, 2,.., p) and θ_j (j = 1, 2, ..., q) are model parameters for autoregressive and moving average models respectively. However, ARMA models fail nonlinear and nonstationary datasets forecasting especially trend and/or seasonality occurs as in many real life cases (Cheng *et al.*, 2015). There are many improvements studied on ARMA model to overcome nonlinearity and nonstationary processes existence. One of the mainly used models is Autoregressive Integrated Moving Average (ARIMA) model which is the form of Autoregressive Moving Average (ARMA) model to overcome nonstationary processes such as trends and seasonality. In ARIMA models, stationary time series is created by differencing original nonstationary observations more than one times if necessary (Adhikari and Agrawal, 2013). However, high-frequency noise occurs by differencing the original time series and order of ARIMA model is not easy to determine. Also, ARIMA models generally overcome only first order non-stationary (Cheng *et al.*, 2015). Mathematical formulation of ARIMA(p, d, q) model can be summarized as in Eq. 3.6 below;

- If d=0: $y_t' = y_t$
- If d=1: $y_t' = y_t$ - y_{t-1}
- If d=2: $y_t' = (y_t$ - $y_{t-1})$ - (y_{t-1} - $y_{t-2}) = y_t - 2y_{t-1} + y_{t-2}$

$$\hat{y_t}' = c + \sum_{i=1}^{p} \varphi_i y'_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \tag{3.6}$$

where y_t' is a ARMA(p, q) process in Eq. 3.5 above, p is order of autoregressive terms, d is the number of non-seasonal differences needed for stationarity, q is the order of moving average terms and ε_t is a random noise. In ARIMA(p, d, q) model, when d=0, the model is equal to ARMA(p, q) model. When $d = q = 0$, model is equal to AR(p) and when $d = p = 0$, model is equal to MA(q). ARIMA$(0, 1, 0)$ is known as Random Walk model (Adhikari and Agrawal, 2013). This above equation is for non-seasonal differences. When seasonality occurs, seasonal difference is applied. A seasonal difference is differencing the time series from one season to the next season. For example; if there are 12 periods in a monthly dataset, the seasonal difference of y at period t is $y_t - y_{t-12}$. ARIMA model also fails when there are nonlinear relationships in time series. There are other forms of ARIMA model to overcome different situations. Earlier, Box and Jenkings (1976) introduce Seasonal ARIMA (SARIMA) model for time series where seasonality occurs. SARIMA notation is explained as ARIMA$(p, d, q)(P, D, Q)s$ where p, d, q are non seasonal part, $(P, D, Q)s$ seasonal part and s is number of periods per season. Eq. 3.7 shows general formulation of SARIMA model below (Arunraj and Ahrens 2015);

$$\phi_p(B)\Phi_P B^S (1-B)^d (1-B^S)^D y_t = c + \theta_q(B)\Theta_Q(B^S)\varepsilon_t \tag{3.7}$$

where, S is seasonal length, B is lag operator, y_t is time series with seasonal effect, $\phi_p(B)$ is non-seasonal AR operator with order p, $\Phi_P B^S$ is seasonal AR operator with order P, $(1-B)^d$ is non-seasonal differencing operator with order d, $(1-B^S)^D$ is seasonal differencing operator with order D, $\theta_q(B)$ non-seasonal MA operator with order q, $\Theta_Q(B^S)$ is seasonal MA operator with order Q, ε_t is white noise and c is

constant.

Another widely used form of ARIMA model is Autoregressive Fractionally Integrated Moving Average (ARFIMA) model which is used when time series have long range dependence and non-integer values is used for differencing parameter d. There is an another extended version of ARIMA method, in which forecasts not only based on past observations but also includes other explanatory variables into the model. This model is called ARIMAX where X stands for explanatory variables. ARIMAX model has advantage of using autocorrelation in residuals to increase forecasting accuracy. One disadvantage of the ARIMAX model is interpretation of coefficient of explanatory variable. General formulation of ARIMAX model is just explanatory variable addition to basic ARIMA model.

3.2. Regression Based Models

3.2.1. Decision Tree Models

Decision tree is a tree shaped decision algorithm diagram which splits either by true or false statements to internal nodes. Decision algorithm starts with the root node and ends with leaf nodes. Decision trees which originally created for classification problems are also capable of solving regression problems (Geurts *et al.*, 2009). The main objective of tree-based models is to divide the sample space repetitively into smaller parts (Breheny, 1984). Classification trees aim to classify categorical variables. Regression trees are used for prediction of target variables which are continuous. Decision tree methods have three main constituents;

- Data split rules for each node
- Stopping rules
- Prediction for the target variable in each leaf node

General form of the regression trees for a linear regression model where $y_t = f(x_j) + \epsilon_t$, starts with dividing space by $x_j \leq s$ into two parts and modeling the output using the mean of the y in the produced two parts. Splitting process is repeated for divided parts until a stopping rule performed (Breheny, 1984).
The final result is the form of Eq.3.8 below;

$$f(x) = \sum_m c_m I \quad (x \in R_m) \tag{3.8}$$

where R_m represents the sum of the divided parts, c_m is the constant (i.e. the mean of y_i for those observations $x_i \in R_m$) for the mth part. Eq. 3.9 shows estimated cm when objective is least squares methodology where optimal splitting variable j and split point s shows the highest decrease in the residual sum of square (Breheny, 1984):

$$\hat{c}_m = \frac{\sum_i y_i I \quad (x \in R_m)}{\sum_i I \quad (x \in R_m)} \tag{3.9}$$

For each of the splitting variable x_j, optimal splitting point (s) can be found in minimizing Eq. 3.10;

$$\sum_{i:x_j \leq s} (y_i - \hat{c}_1)^2 + \sum_{i:x_j > s} (y_i - \hat{c}_2)^2 \tag{3.10}$$

Eq. 3.10 is repeated for each variable j to find best pair of (j, s) according to objective of the tree (i.e. residual sum of square).

Classification and Regression Trees (CART) is one of the widely used decision tree method. CART is a tree structured sequence of questions where each node divided into two children and those children divided into grandchildren until no more questions available. When maximal sized tree occurred, algorithm starts to prune tree by eliminating least contributed split. CART algorithm tries to find best tree with 10-fold cross validation pruning algorithm (Breiman *et al.*, 1984).

3.2.2. Penalized Regression

Ordinary Least Squares (OLS) is a widely used method in linear regression models for parameter estimating. OLS minimizes the residual sum of squares by a linear function applied to original data set. Although it is easy to apply, OLS is not good in prediction accuracy and model interpretation in some cases. The performance of OLS has been improved by penalized regression techniques such as ridge regression (Hoerly and Kennard, 1970), lasso (Tibshirani, 1996) and elastic net (Zou and Hastie, 2005).

Elastic net is a regularization and variable selection method proposed by Zou and Hastie in 2005. Elastic net makes variable selection and continuous shrinkage. Rather than Lasso model, Elastic net can select correlated variables in a group together. It has great performance when number of predictors are quite large than number of observations. Elastic net method is a combination of lasso and ridge regression. It applies both $L2$ and $L1$ norms to regression coefficients while minimizing residual sum of squares. Eq. 3.11 shows general formulation of elastic net regularization;

$$\hat{\beta} = \arg\min_{\beta} \|y - X\beta\|^2 + \lambda_2\|\beta\|^2 + \lambda_1\|\beta\|_1 \tag{3.11}$$

where X is the model matrix, y is the response vector, β is regression coefficients and λ are regularization coefficients and $\hat{\beta}$ is elastic net estimator. Elastic net penalty is shown in Eq. 3.12 below;

$$J(\beta) = \alpha\|\beta\|^2 + (1-\alpha)\|\beta\|_1 \quad \textit{where } \alpha = \frac{\lambda_2}{\lambda_1 + \lambda_2} \tag{3.12}$$

Zou and Hastie proved that elastic net has better performance than both Lasso and ridge regression methods.

Ridge regression (also known as Tikhonov regression) minimizes the residual sum of squares and sum of squares of regression coefficients. In other words, ridge regression applies $L2$-norm for both residuals and regression coefficients. Ridge regression has better prediction performance than OLS model. However, it keeps all predictors in the model.

Lasso is also minimizes the residual sum of squares and apply variable selection by minimizing the number of regression coefficients. In other words, Lasso applies $L2$-norm to residuals and $L1$-norm to regression coefficients. Although Lasso is good for variable selection, it performs poorly when data has group of correlated variables. Lasso randomly selects only one variable and ignores others from the model. Also when numbers of observations (n) are quite less than numbers of predictors (p), Lasso selects only n variables in the model.

3.3. Performance Parameters of Model Accuracy

One of the key element of solving a forecasting problem is to determine the most suitable model. Since there can be plenty of options for a particular forecasting problem, it requires an assessment parameter to select the most applicable approach. There are several methods used for assessing model accuracy such as Akaike Information Criterion (AIC), Mean Square Error (MSE), Root Mean Square Error (RMSE), Root Mean Square Percentage Error (RMSPE), R-squared etc.

Mean Square Error (MSE) is one of the widely used forecasting accuracy parameter (James *et al.*, 2013). Since MSE measures variability of forecasts, it is expected to be as small as possible. General formulation of MSE is given in Eq. 3.13 below;

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.13}$$

where, n is number of observations, y_i is true values of response variable, $\hat{y}_i$ are predicted values of response variables. Root Mean Percentage Error (RMSE) is just the square root of MSE. RMSE and MSE have a direct relationship since square root is a non-decreasing function. RMSE just adjusts the scale of the accuracy in terms of the scale of the target. On the other hand, Root Mean Squared Percentage Error (RMSPE) is just same as RMSE, only gives percentage type error. Formulation of RMSPE is given in Eq. 3.16 below;

$$RMSPE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\frac{y_i - \hat{y}_i}{y_i})^2} \tag{3.14}$$

Coefficient of determination a.k.a. R-squared (R^2) is another criteria which is used for linear regression methods. General formulation is given in Eq. 3.15 below,

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3.15}$$

where, $\bar{y}$ is the mean of observations. Since the denominator of the R^2 formula is constant, maximizing R-squared needs minimizing the sum of squared residuals. Therefore, MSE and R-squared have an inverse ratio between each other. The more value of R^2 shows the better forecasting. Although adding new variables into the model generally decreases the sum of squared residuals, it can lead to overfitting and creates complexity on model. Therefore, a large value of R-squared should not be the only parameter while assessment of mode performance (Montgomery *et al.*, 2015). Therefore, R-squared is adjusted as follows;

$$R^2_{Adj} = 1 - \frac{(1 - R^2)(n - 1)}{n - o - 1} \tag{3.16}$$

where, o is the number of independently adjusted parameters. Adjusted R-squared penalizes useless variables in the model.

Akaike Information Criterion (AIC) is another widely used performance criteria for model selection. AIC is defined by Akaike (Akaike, H., 1974) with Eq. 3.17 below;

$$AIC = -2log(\hat{L}) + 2o \tag{3.17}$$

where, o is the number of independently adjusted parameters and $\hat{L}$ is the value of maximum likelihood. While calculating maximum likelihood to obtain a better model, AIC also penalizes more parameter to overcome overfitting and provide simplicity. A model can be defined as good to fit when AIC value is small.

4. METHODOLOGY

Classical time series models fit a model with respect to trends, seasonality and cyclic components of the time series and leave unpredicted components as noise. In our algorithm, exogenous regressors which explain residuals are included to model according to their effect on base learners forecasts. Interactions between regressors are also considered while regressor addition to base learner model. In other words, linear model becomes nonlinear by including nonlinear relationships between independent variables into the model. Proposed algorithm consists of two phases. In the first phase, a linear forecasting model is fitted. In this study, ARIMAX model, Linear Regression model and Penalized Regression models are used as base learners in the first phase. In the second phase residuals of the first model is analyzed with a regression tree with all other regressors to model unpredicted components in the base learner model. The exogenous regressor which generates the highest residual is detected in the second phase and included to first model. Proposed approach iteratively spans and each time new input variable added to model until stopping algorithm starts.

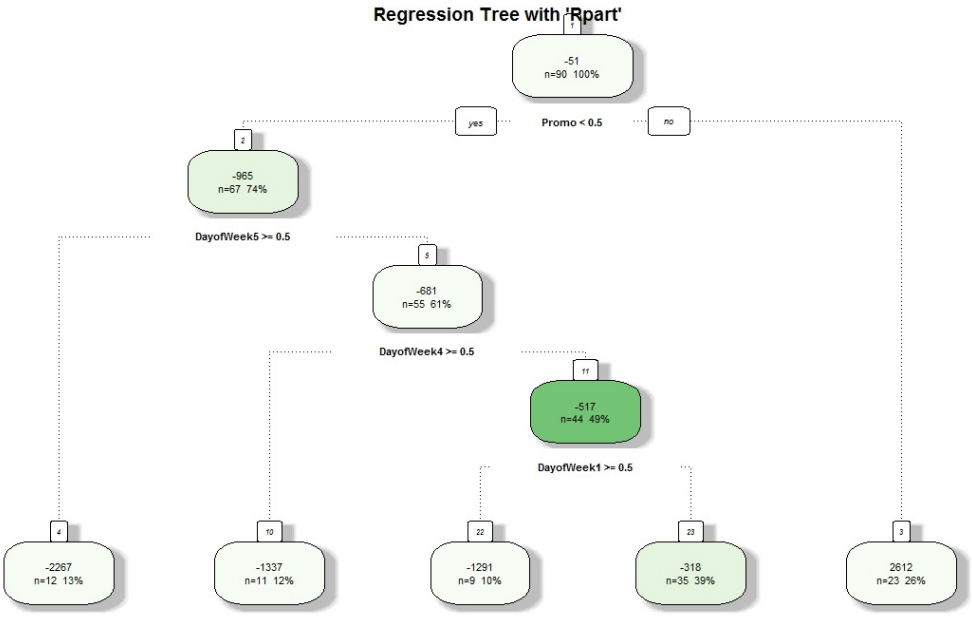

Figure 4.1. An Example for Regression Tree

As an example for new binary variable addition in second stage, Figure 4.1 shows an output of the regression tree from Section 5.1 with a sales data time series with regressors of promotion as "$Promo$" and day of weeks as "$DayofWeek$". It can be seen from Figure 4.1 that highest error from the first autoregressive model is in terminal node 3 with the value of 2612 and occurred when "$Promo$" is greater than 0.5 which represents "Promo" $= 1$ since "$Promo$" is a binary variable (0,1). Since new variable is determined as "$Promo = 1$", a new binary variable is added to first autoregressive model which shows when "$Promo$" is equal to 1. New variable can also be nonlinear as a combination of two regressors. For example, highest error can be found when promotion occurs on Saturdays. In this case new binary variable added when "$DayofWeek6 = 1$ and $Promo = 1$".

Figure 4.2 shows pseudo-code for proposed algorithm below;

fit $f(x_i + y_{t-1}...y_{t-p})$
 where $f()$ can be a linear or nonlinear model with regressed time series
  $y_{t-1}...y_{t-p}$ are lags for p
  x_i are regressors
  stop if MSE between last two models are less than *5%*
find e_t
 where e_t are residuals
Define residual matrix E_t with e_t and x_i
build tree(E_t)
  stop if tree has only one root and complexity parameter is equal to zero
    then $f(x_i)$ is final model
  else Find the node which maximize MSE
    Extract relevant path for the node
    Define x_{i+1} regressor according to path
add x_{i+1} to first model

Figure 4.2. Pseudo-code for Proposed Algorithm

Figure 4.3 shows flow chart of the proposed algorithm below;

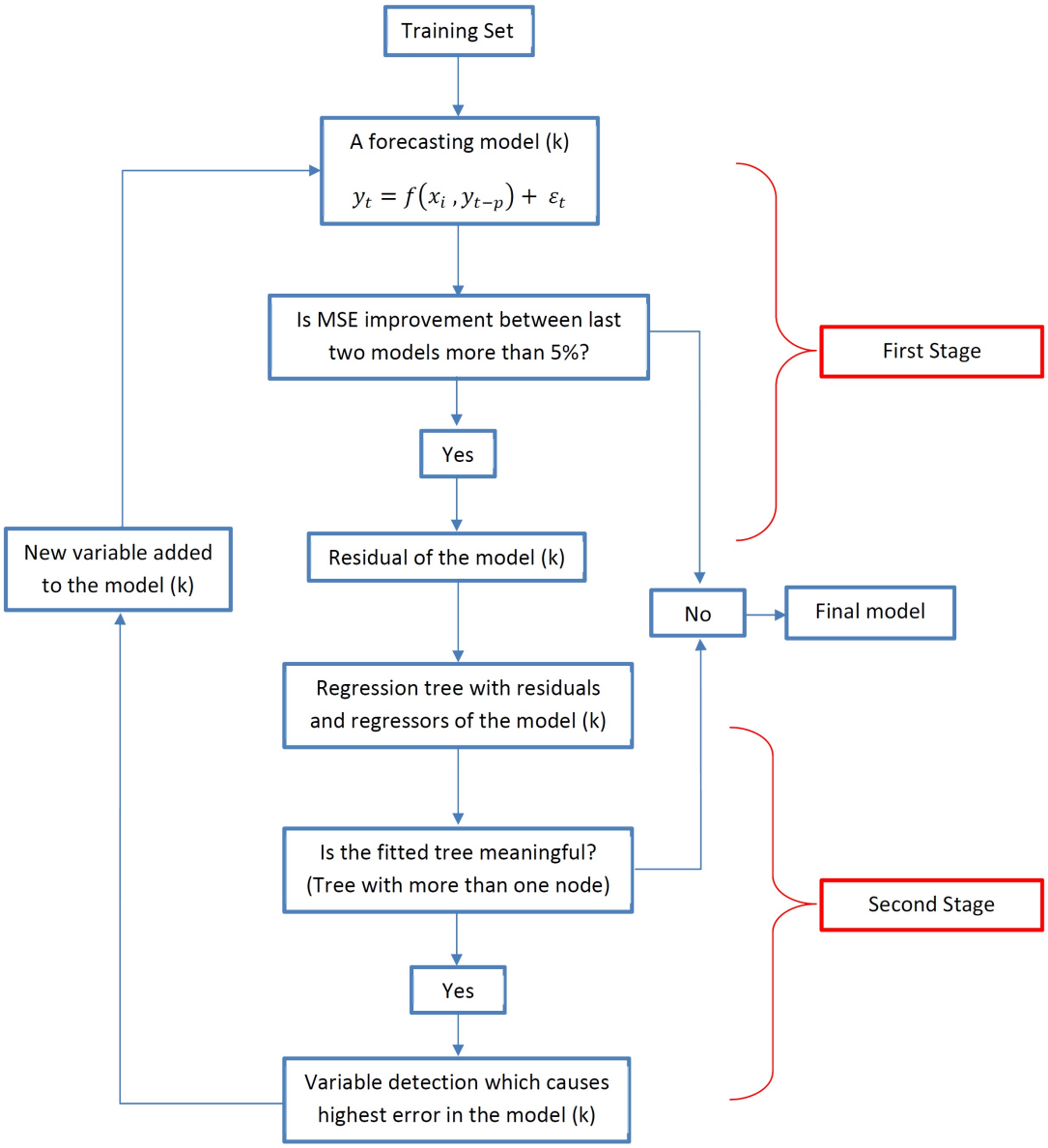

Figure 4.3. Flow Chart of the Proposed Model

4.1. First Stage of Proposed Approach

In the first stage a linear forecasting model is implemented to time series data as formulated in Eq. 4.1;

$$y_t = f(x_i + y_{t-1}...y_{t-p}) + \varepsilon_t \tag{4.1}$$

where y_t is variable of interest at time t, x_i are regressors for $i = 1, 2, \ldots .n$, $y_{t-1}...y_{t-p}$ are lags for $p = 1, 2, ..t-1$, ε_t is random error. Residuals of the first model ($e_t, t = 1, 2, \ldots .n$) are extracted (Eq. 4.2) and residual time series is constituted with original regressors of the main time series. Base model f can be a linear or nonlinear model for a time series with regressors. In this thesis, we used three different models; ARIMAX, Linear regression and Penalized regression. General formulation of those models are explained in followed Section 4.1.1, 4.1.2 and 4.1.3.

4.1.1. Base Learner 1: ARIMAX model

ARIMAX model is another version of ARIMA with regressors used for time series with independent variables. In our approach, ARIMAX models starts with previous observations (lags) of the response variable and adds regressors which are generated in the second stage. Figure 4.4 shows pseudo-code of ARIMAX model for the first stage below;

fit $f(x_i + y_{t-1},...y_{t-p})$
where $f()$ is ARIMAX,
$y_{t-1}..y_{t-p}$ are previous observations of response variable
x_i are new regressors added from Second Stage
find e_t

Figure 4.4. Pseudo-code for ARIMAX in Stage 1

4.1.2. Base Learner 2: Linear Regression model

Linear Regression is mostly used simple regression method that aims to fit linear models on time series. In this study, proposed algorithm is also tested on linear regression model. Firstly, some initial independent variables has been added to linear regression model and proposed algorithm has been implemented. New binary variables has been added to base linear regression model in each step with second phase of proposed algorithm. Figure 4.5 shows pseudo-code of Linear Regression model for the first stage below;

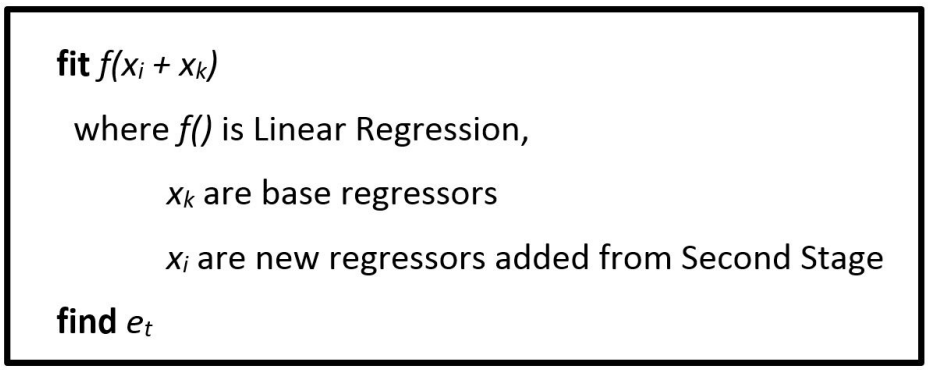

Figure 4.5. Pseudo-code for Linear Regression in Stage 1

4.1.3. Base Learner 3: Penalized Regression model

Penalized regression models are used for variable selection and regularization. In our approach, penalized regression model starts with two lags of the response variable; last day and last week observations, and adds new regressors which are generated from second stage. Figure 4.6 shows pseudo-code of Penalized Regression model for the first stage below;

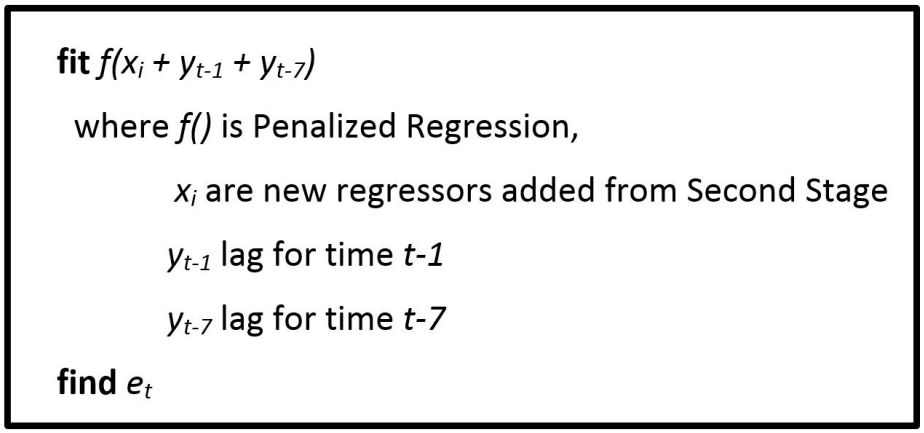

Figure 4.6. Pseudo-code for Penalized Regression in Stage 1

4.2. Second Stage of Proposed Approach

In the second stage, linear regression model implemented to residual time series resulted from first stage with Eq. 4.2 for regression tree as explained in Eq. 4.3;

$$e_t = y_t - \hat{y}_t \quad where\ t = 1, 2, \ldots, n \tag{4.2}$$

$$e_t = \beta_0 + \sum_{i=1}^{p} \beta_i * x_i + \varepsilon_t \tag{4.3}$$

where e_t is the residuals of the first model, x_p are the regressors of the first time series data,y_t is real response variable at time t, $\hat{y}_t$ is predicted response variable at time t. Result of the regression tree shows where residuals are occurred in accordance with regressors' behavior. The path for the terminal node which shows highest residual is extracted to be added first model as a new binary variable. New variable can be both linear or nonlinear which can be combination of two regressors. After the variable which cause highest error detected, model continues with first stage with lags or original regressors and new binary variable and new model is implemented. Then, model continues with second stage with second model's residuals and regression tree implemented to residuals time series. This loop repeated tr times until regression tree

is not able to create a meaningful tree or MSE improvement from the past model is less than 5%. When tree stops or loop stops, final model created in the first stage with new variables. In this study, ARIMAX, Linear Regression and Penalized Regression models are used in first stage and both models' performance have been improved on the final model.

The proposed algorithm has been implemented to Rossmann stores sales data. Details about Rossmann data is explained in Section 5.2. In addition to real life data, an autoregressive synthetic data is generated and the proposed algorithm tested.

5. EXPERIMENTS AND RESULTS

The proposed algorithm has been tested on both real data and synthetic data. Part of Rossmann Stores daily sales data has been used as real data. Three different models has been tested on real data as base learner; ARIMAX, Linear Regression and Penalized Regression. In this study, R program has been used for data analysis and model implementation. ARIMAX analysis has been made with "*auto.arima*" package. Hyndman (2008), proposed an automatic ARIMA algorithm with "*auto.arima*" function for automatic forecasting of univariate time series. Proposed automatic ARIMA forecasting algorithm uses unit root test and the Akaike Information Criterio (AIC) to select best model order which is the main objective. In this thesis, proposed approach is also implemented to linear regression model with "*lm*" package. In the first step of the proposed model, response variable and regressors added to "*lm*" function and new regressors added from second stage. Lastly, "*glmnet*" package is used in R program for penalized regression analysis on proposed approach in the first stage. "*glmnet*" fits generalized linear models such as regression, two-class logistic regression and multinominal regression with elastic-net penalties (Friedman *etal*., 2010).

In the next chapter, an example of the proposed algorithm with ARIMAX model as base learner is implemented to synthetic data with two independent variables.

5.1. Model Implementation on Synthetic Dataset

5.1.1. Synthetic Data Generation

We first evaluate the proposed approach on a synthetic examples. An autoregressive sales dataset with two independent variables and 120 observations is generated. General properties of the synthetic data are detailed in Table 5.1 below;

Table 5.1. Synthetic Data Variable Information.

Name	**Range**
Sales	See Eq. 5.1
Day of Week	1:7
Promo	0,1
Date	01.01.2013 : 30.04.2013

Sales time series is formulated in Eq. 5.1;

$$\begin{aligned} y_t &= -0.4y_{t-1} + 0.5y_{t-2} + \varepsilon_t \\ where\ \ \varepsilon_t &\sim 5000 + N(0, 150)\ \ and\ \ t = 1, 2, \ldots 120 \end{aligned} \tag{5.1}$$

where, y_t is the autoregressive sales data at time t, ε_t is innovations.

Independent variables effects on sales data added manually on synthetic data. Also nonlinear relationship between independent variables added to synthetic data. Table 5.2, Figure 5.1 and 5.2 below summarizes independent variables effects;

Table 5.2. Effects of Independent Variables.

Promo (Y/N)	**Weekend (Y/N)**	**Effect on Sales**
N	Y	+3000
Y	Y	+10000
Y	N	+1500

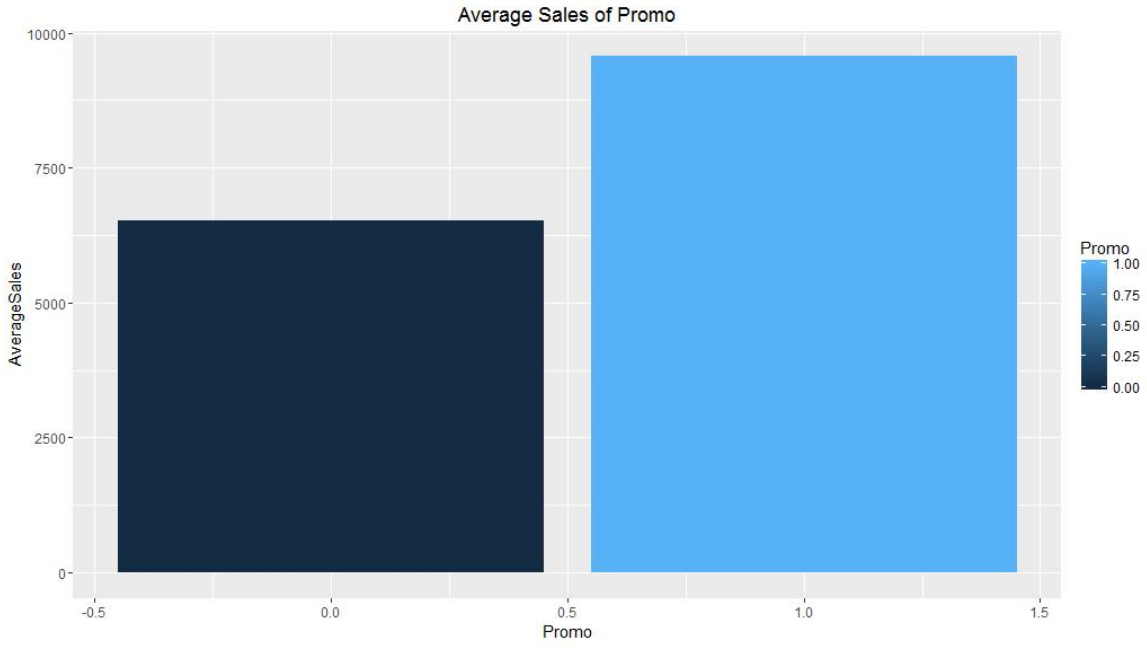

Figure 5.1. Effect of Promo on Sales

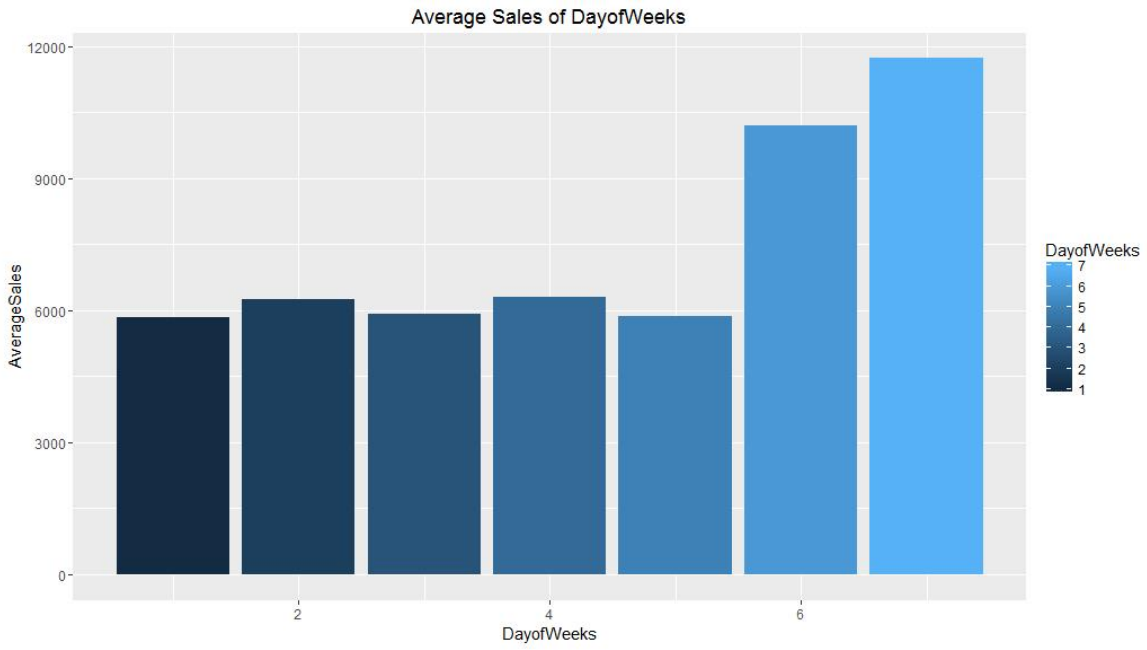

Figure 5.2. Effect of Day of Weeks on Sales

Figure 5.3 shows final form of the sales data with independent sales effects;

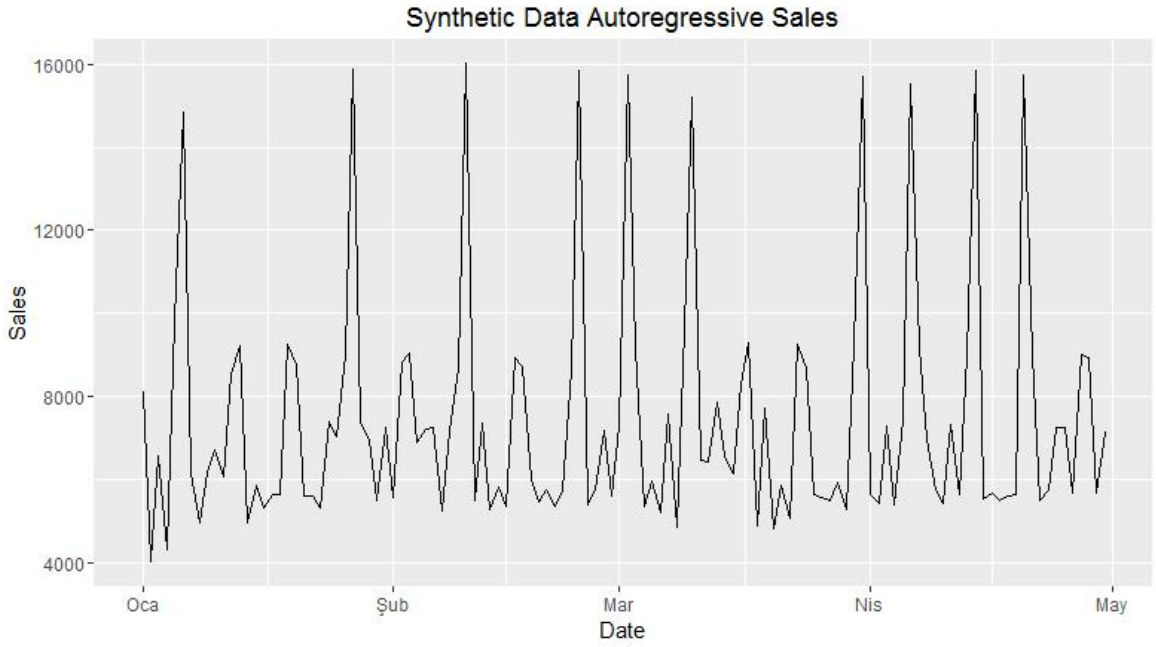

Figure 5.3. Final form of the Sales in Synthetic Data

5.1.2. Analysis on Synthetic Data

Synthetic data is divided two parts; train dataset with 90 observations and test dataset with 30 observations. ARIMAX model is used in first stage as base learner to add explanatory variable effects. Model starts with previous sales observations and each time new variable added to model from second phase until mean square error (MSE) does not improve more than 5% or model could not find any new variable due to lack of meaningful decision tree. 6 new variables added to model for the first dataset and model stopped since MSE did not improve enough.

Figure 5.4 shows first regression tree from first model. As it can be seen from the regression tree, highest errors occurred in Terminal Node 3 where "$Promo$" is greater than 0.5. Since "$Promo$" is a binary variable (0,1), regression tree points out "$Promo = 1$" for highest errors. Therefore, model adds "$Promo = 1$" variable to the initial model.

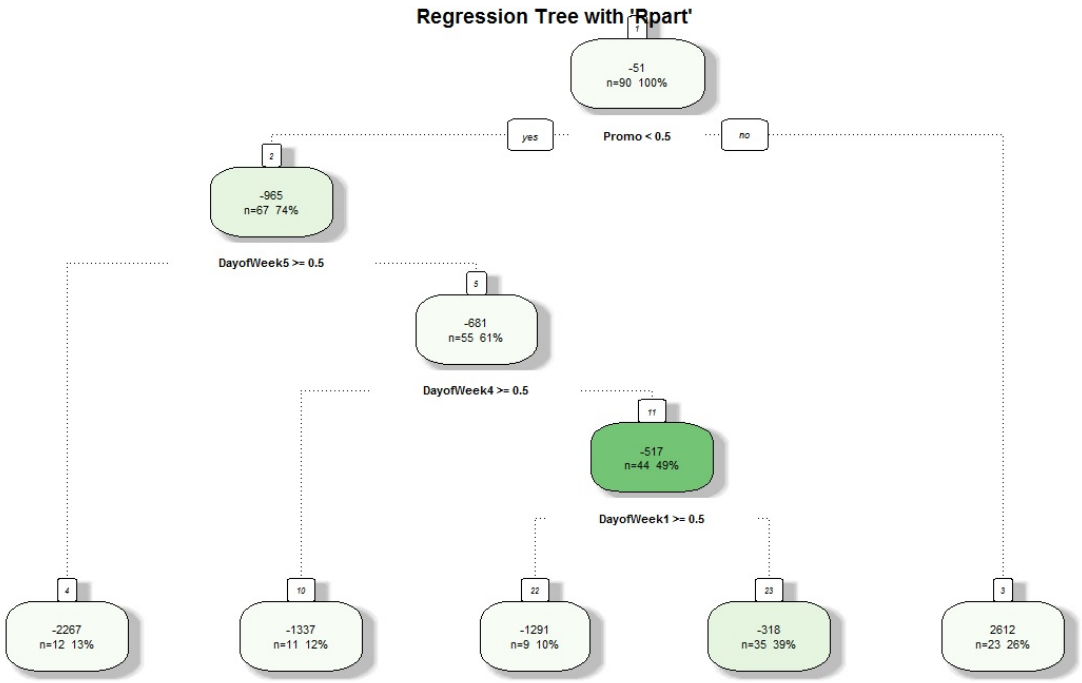

Figure 5.4. First Regression Tree Results

Since new explanatory variable detected in second stage, model continues with first stage with previous obversations + new variable which indicate whether there is promo or not. Model continues to add new variables until MSE doesn't improve more than 5%.

Table 5.3 shows results error improvements on train dataset by Akaike Information Criterion (AIC), Root Mean Square Error (RMSE) and Root Mean Square Percentage Error (RMSPE).

Table 5.3. Error Improvements on Train Dataset.

Steps	**AIC**	**RMSE**	**RMSPE**
1 (without any regressor)	1676.503	2493.038	0.310
2	1651.554	2223.707	0.265
3	1618.248	1857.310	0.209
4	1538.890	1168.623	0.139
5	1485.190	867.359	0.101
6	1186.751	154.995	0.025

Figure 5.5 shows prediction improvements on training dataset, while Figure 5.6 shows improvements on test dataset after each new variable addition in second stage of our model.

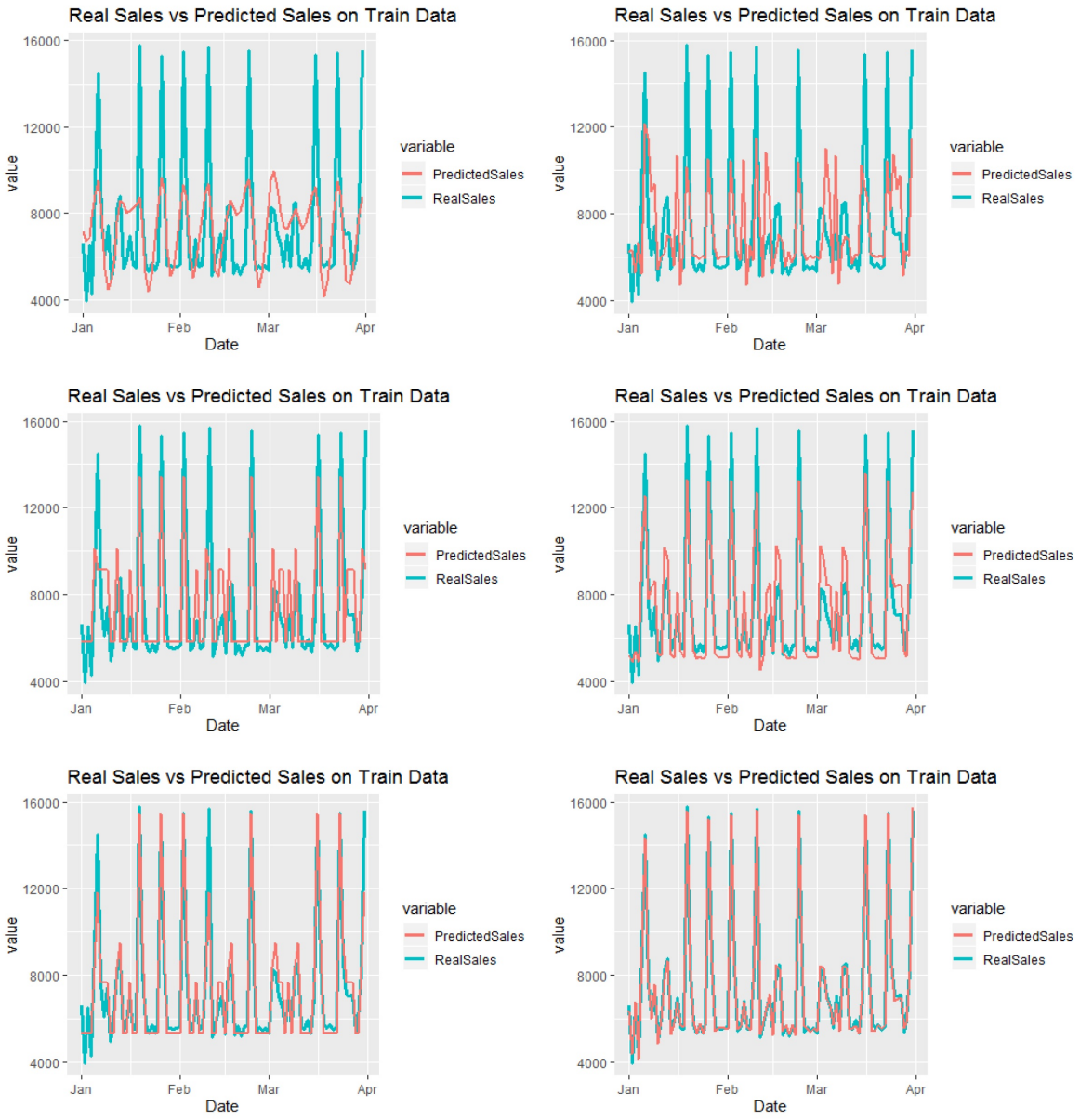

Figure 5.5. Improvements on Train Dataset

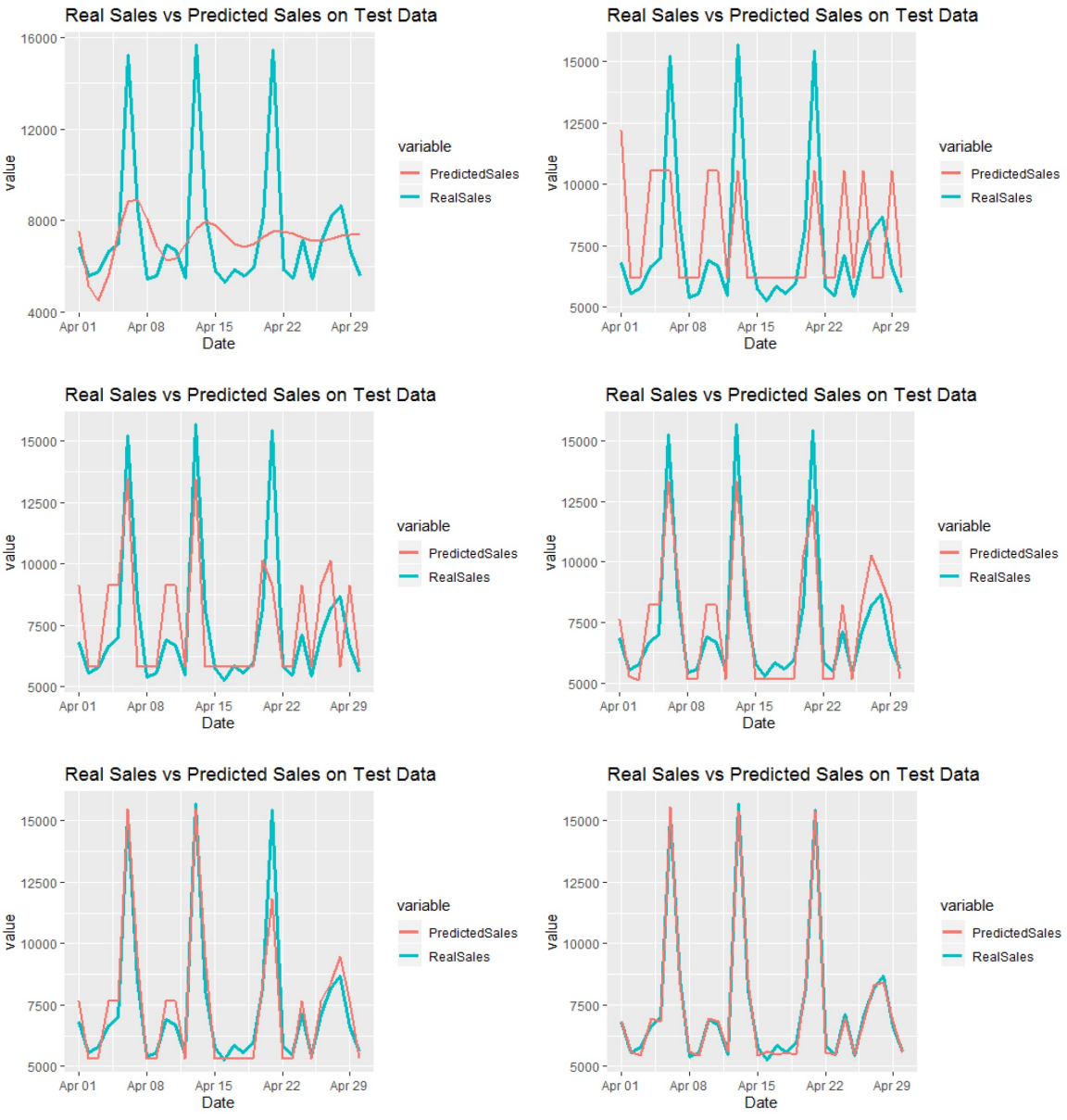

Figure 5.6. Improvements on Test Data Sales Predictions

In this experiment, synthetic data generated ten times and each time models improved with variable addition. Table 5.4 shows results of the final models in each synthetic data. Figure 5.8 and 5.7 shows average AIC, RMSE and RMSPE improvements on training data and test data after each iteration respectively. Figure 5.9 and 5.10 shows error improvements in each synthetic data set for test data after each iteration with respect to RMSE and RMSPE respectively. According to below figures, it can be concluded that forecasting accuracy is increased by adding new variables in second stage. Therefore, model performance is increased with proposed approach.

Table 5.4. Final Model Results.

Steps	**AIC Train**	**RMSE Train**	**RMSE Test**	**RMSPE Train**	**RMSPE Test**
1	1186.751	154.995	210.609	0.025	0.032
2	1216.928	185.721	331.678	0.035	0.049
3	1205.083	172.187	219.534	0.030	0.035
4	1184.306	154.949	406.328	0.029	0.060
5	1203.968	171.457	122.217	0.029	0.019
6	1195.801	162.999	183.044	0.026	0.028
7	1185.074	151.369	273.004	0.025	0.043
8	1199.705	167.166	460.789	0.030	0.068
9	1192.118	162.051	300.039	0.029	0.046
10	1200.829	169.963	170.482	0.031	0.026

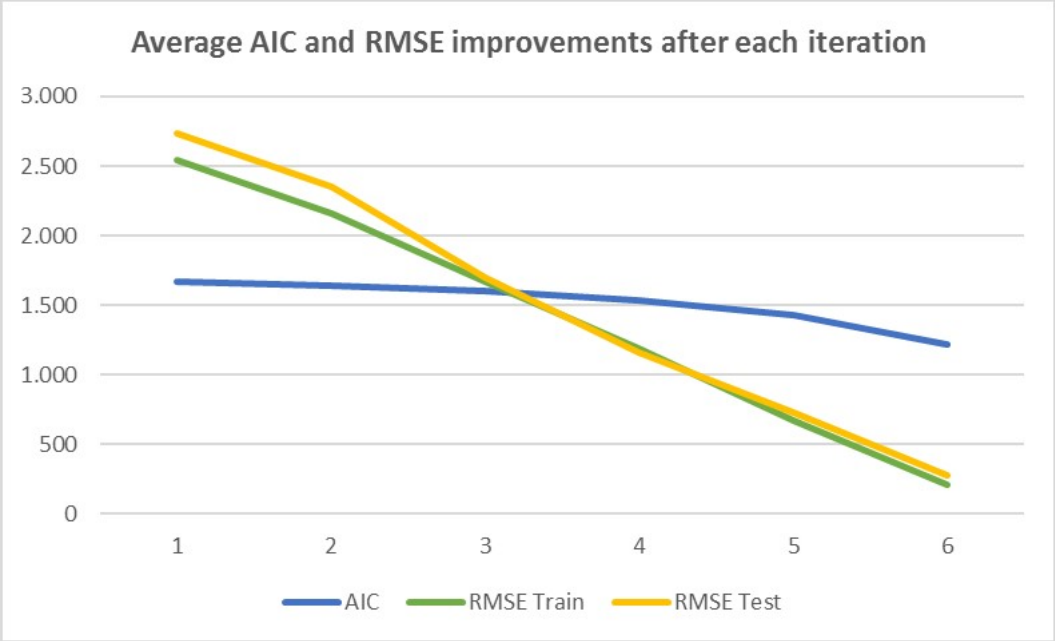

Figure 5.7. Average AIC and RMSE Improvements After Each Iteration

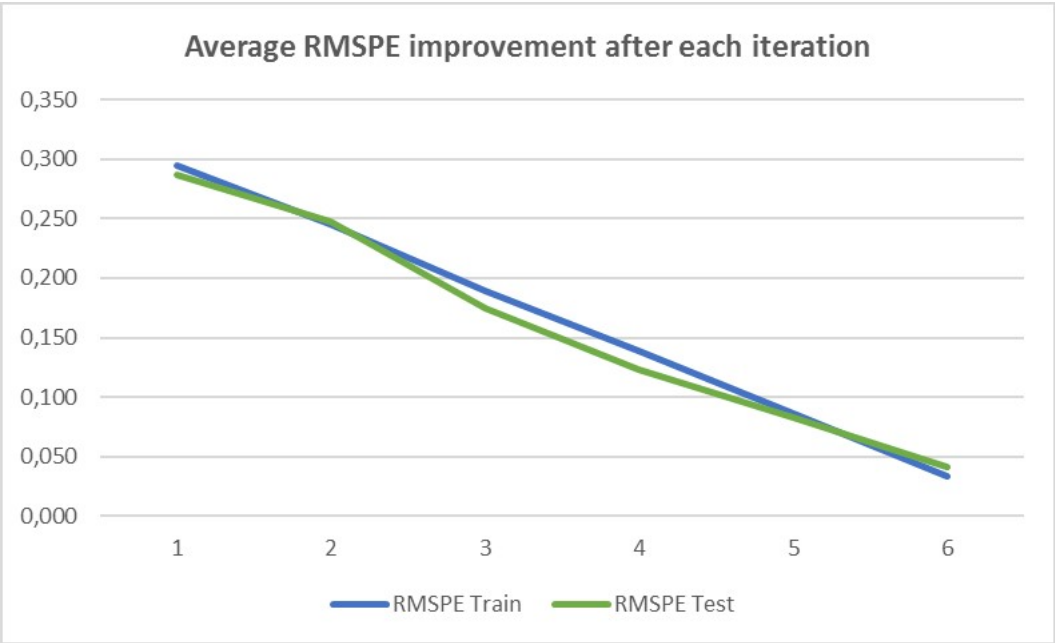

Figure 5.8. Average RMSPE Improvements After Each Iteration

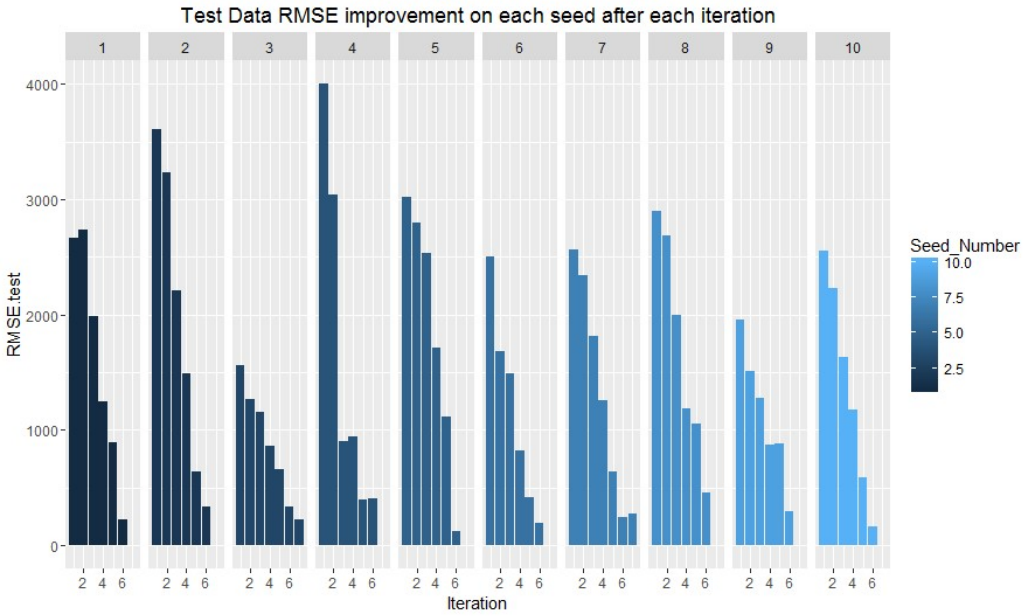

Figure 5.9. Test Data RMSE improvements on each seed after each Iteration

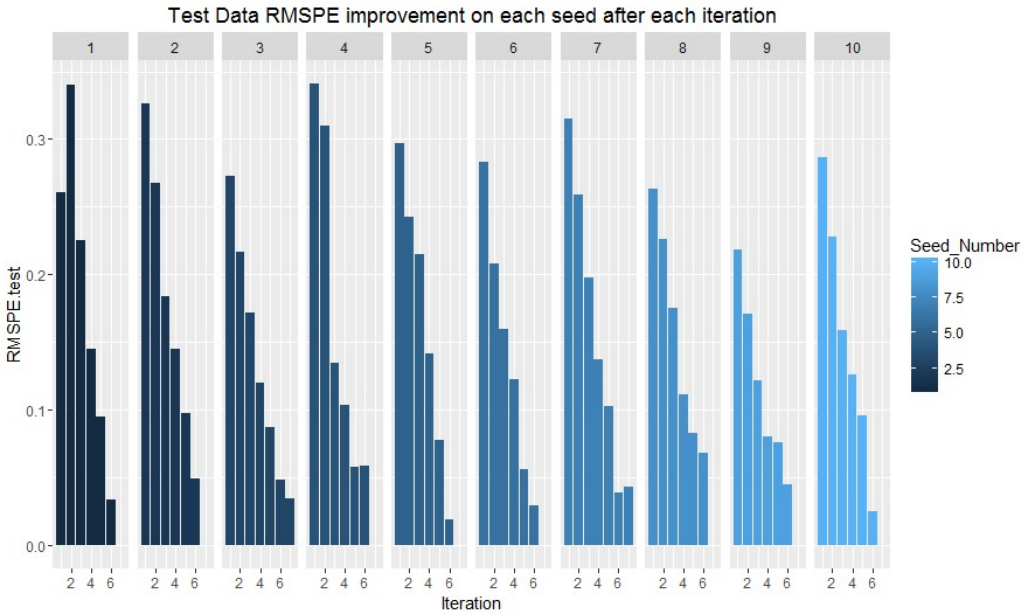

Figure 5.10. Test Data RMSPE Improvements on each Seed after each Iteration

5.2. Model Implementation on Real Dataset

5.2.1. Data Analysis

Rossmann is a Germany based drug store with more than 3,600 stores over the Europe which was founded in 1972. Rossmann store managers tasked to forecast daily sales for a period of six weeks. In this study, Rossmann data which contains three parts as train data, test data and store data is used for forecasting 1115 Rossmann stores' 6 weeks daily sales. General properties of the sales data and store data can be seen in Table 5.5 and Table 5.6 below;

Table 5.5. Sales Data Variable Analysis.

Name	**Range**
Store	1 : 1115
Day of Week	1 : 7
Date	01.01.2013 : 31.07.2015
Sales	0 : 41551
Customers	0 : 7338
Open	0 : 1
Promo	0 : 1
State Holiday	0, a, b, c
School Holiday	0 : 1

Table 5.6. Store Data Variable Analysis.

Name	**Range**
Store	1 : 1115
Store Type	a, b, c, d
Assortment	a, b, c
Competition Distance	20 : 75860
Competition Open Since Month	1 : 12
Competition Open Since Year	1900 : 2015
Long Term Promo	0 : 1
Long Term Promo Since Week	1 : 50
Long Term Promo Since Year	2009 : 2015
Promo Interval	1.Jan, Apr, Jul, Oct, 2.Feb, May, Aug, Nov, 3.Mar, Jun, Sep, Dec

Train data consists of 9 different variables where some of them are categorical and some of them are numerical. As mentioned before, there are 1115 different stores data provided for daily sales of 2 years and 7 months. Test data is just same as train data except with Sales information which is to be predicted in this study.

1115 stores have different type of properties such as assortment, type, competitor distance, competitor open date, long term promotion start date and long term promotion interval as mentioned in Table 5.6 above. Each of these properties makes these stores as unique. Each variable and their effects on sales are analyzed in the next parts.

There are 4 different type of store category and 3 types of product assortment are provided in the data. Number and percentage of store types and assortments can be seen in Figure 5.11 and Figure 5.12 respectively below.

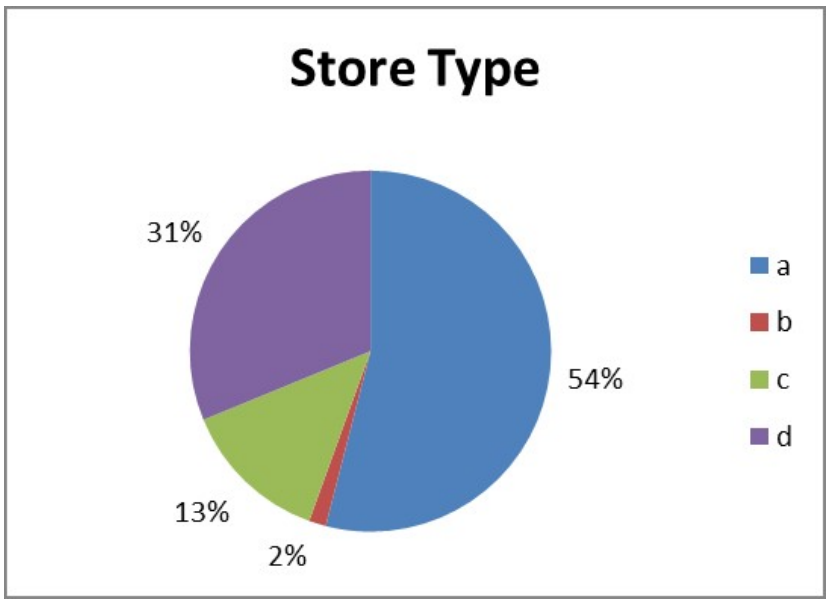

Figure 5.11. Percentage of Store Types

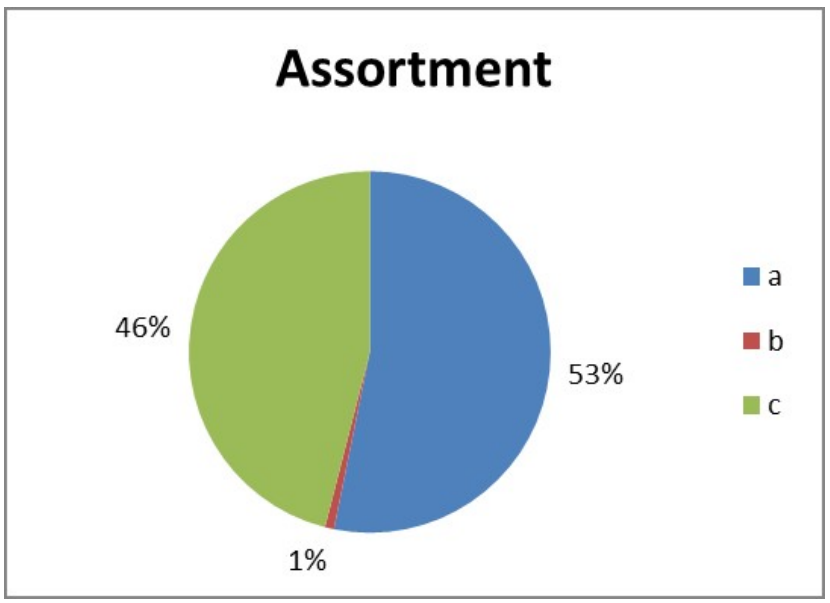

Figure 5.12. Percentage of Assortments

As it can be seen from Figure 5.11 and Figure 5.12, there are very few number of Type "b" Stores and Assortments included in the provided data. It will be hard to find and model assortment and store type "b" effects since only 1% and 2% of train data have type "b" assortment and store respectively. Also Figure 5.13 shows general

overview for average sales of each store types by each assortment. As it can be seen from Figure 5.13, only "b" type stores and assortments have a critic different on average sales than other types. Since there are few numbers of observation on "b" type stores and assortment, these variables are not included in the proposed model.

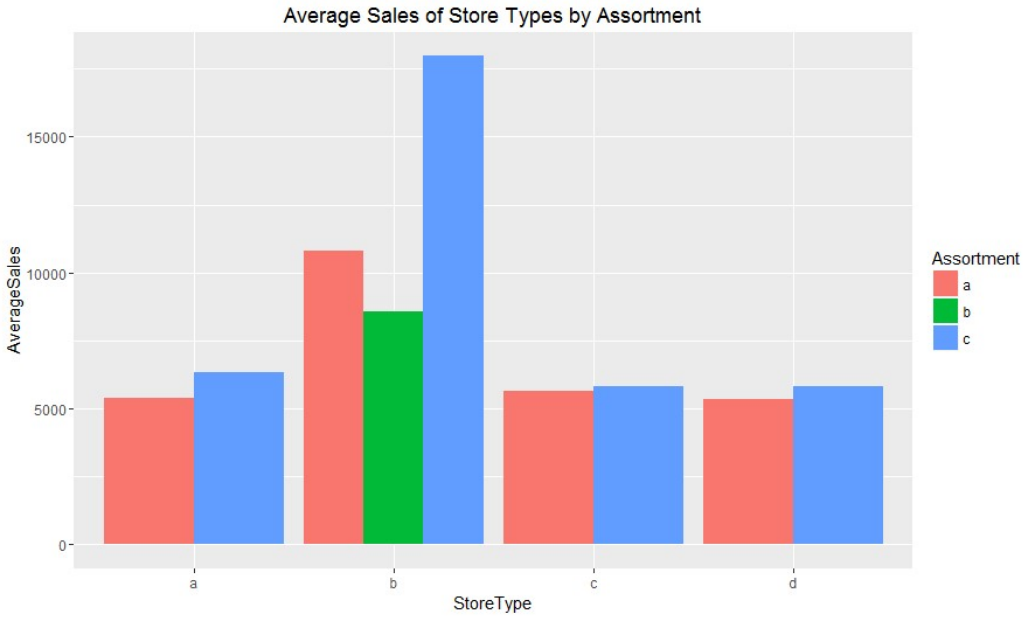

Figure 5.13. Average Sales of Store Types by Assortment

In the proposed data, there are 4 different time components; day, day of week, month and year. We also added seasons (winter, spring, summer and fall) to data analysis. Average sales by each time component are analyzed in the next chapter. Firstly, average sales by each month are demonstrated in Figure 5.14 below. Sales are increased in July and highest sales are occurred in November and December. Since there is not a critical difference on average sales in each month, this variable is also excluded from the model to decrease complexity and run time of the model.

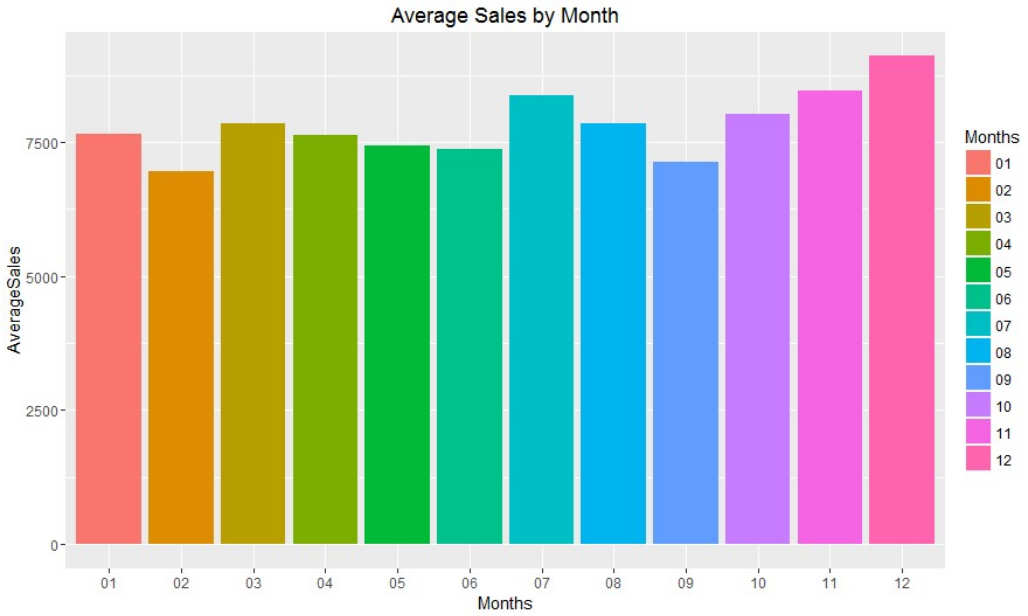

Figure 5.14. Average Sales by Month

Average sales by each day can be seen in Figure 5.15 below. Sales are increased at the beginning, middle and end of the month. Due to decrease number of variable in the model, this variable is added to model as *DaySplit* with three values; first ten days of the month, mid of the month and last ten days of the month.

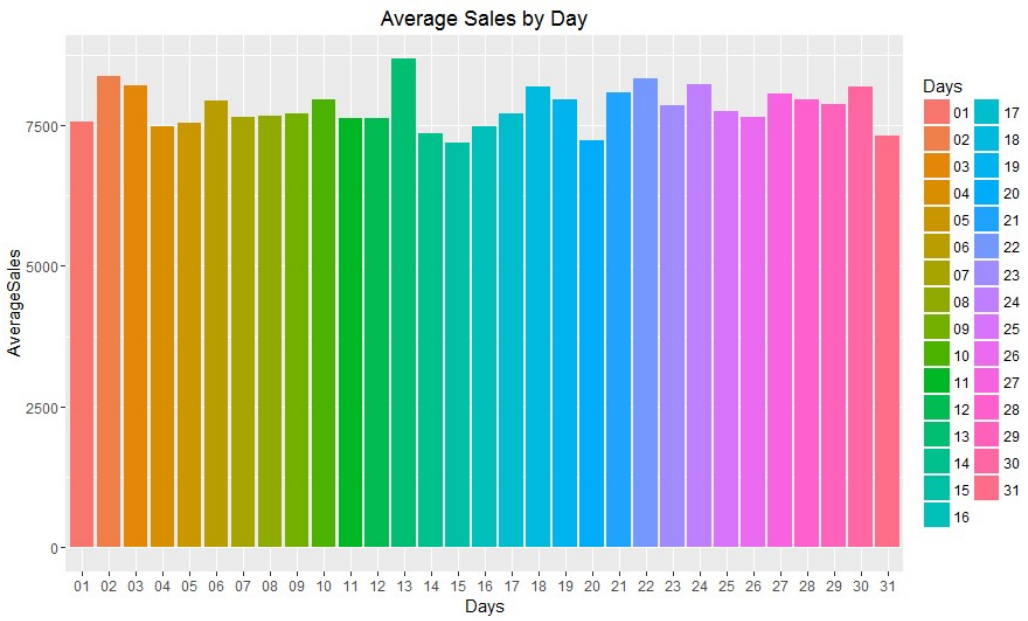

Figure 5.15. Average Sales by Day

There is a positive trend in yearly sales as it can be seen from Figure 5.16. However, this variable is also excluded from the model since there is not a critical difference between each year. Also, Figure 5.17 below shows average sales by seasons. Higher sales are observed in winter and summer. Since the difference is negligible, this variable is also excluded from the model.

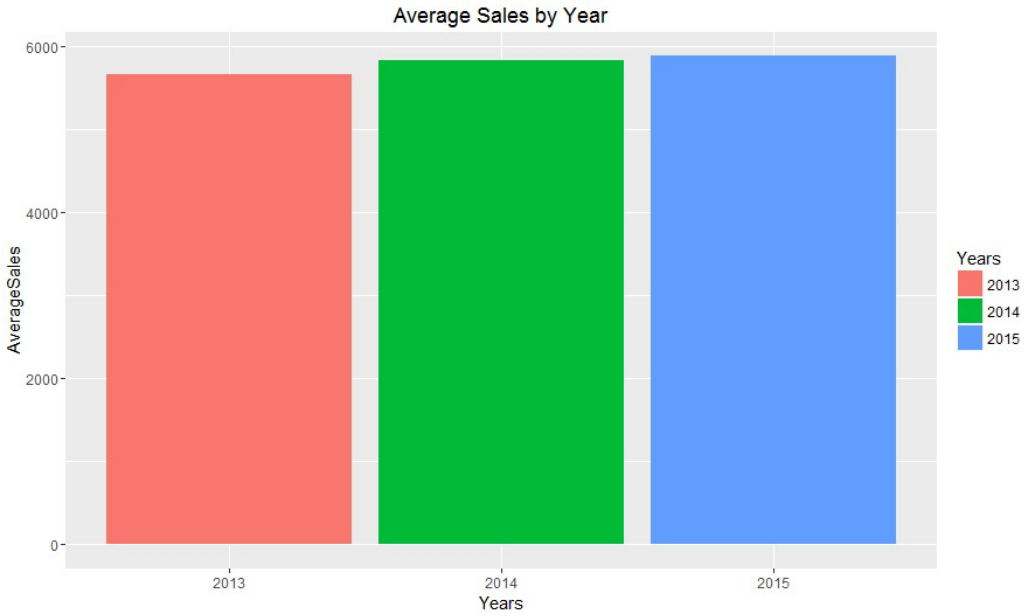

Figure 5.16. Average Sales by Year

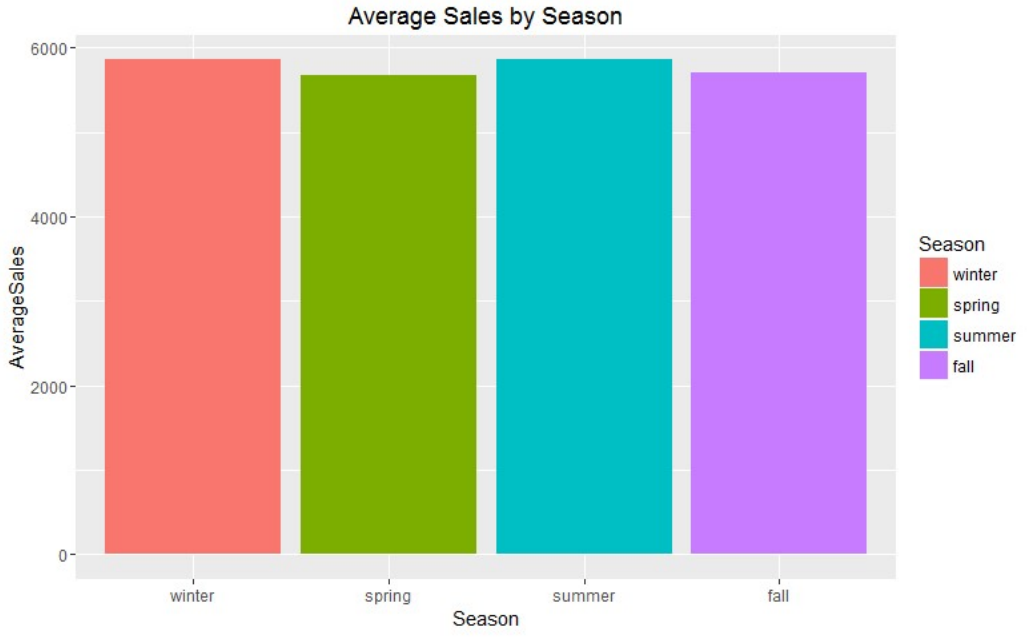

Figure 5.17. Average Sales by Season

Figure 5.18 shows average sales by each day of week. Highest sales are observed at the beginning of the week (Monday) and decreasing until Friday. Sales are increased on Friday also. On Sundays, only "b" type stores are open and they make the highest sales on Sundays (Figure 5.19). Also, only "b" type assortments have highest sales on Sundays (Figure 5.20). Since the number of type "b" stores is really few, average sales on Sundays in Figure 5.18 is quite low regarding other days. Day of Week variable is included to model since it has an important effect on sales.

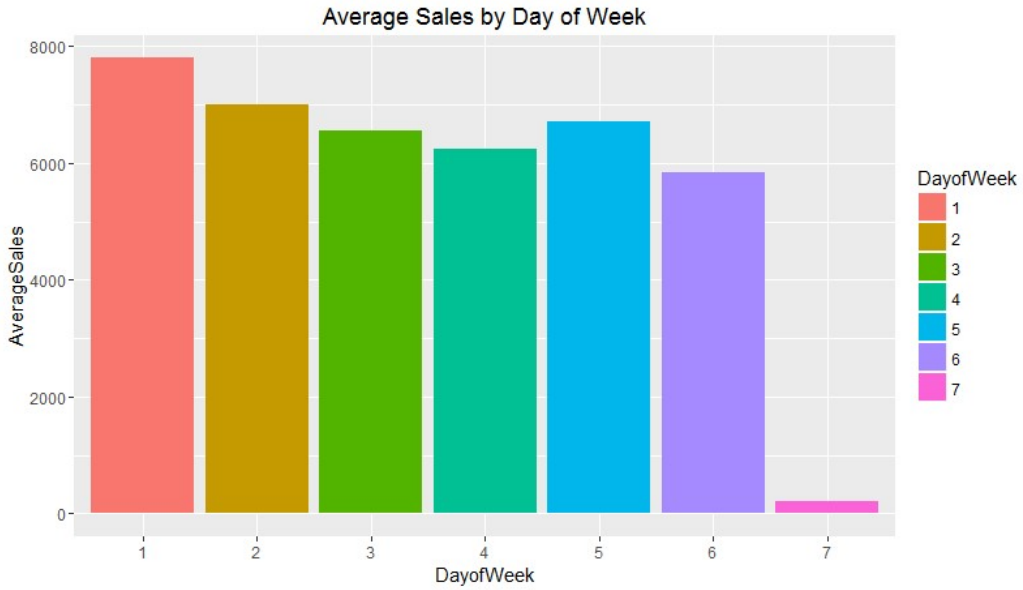

Figure 5.18. Average Sales by Day of Week

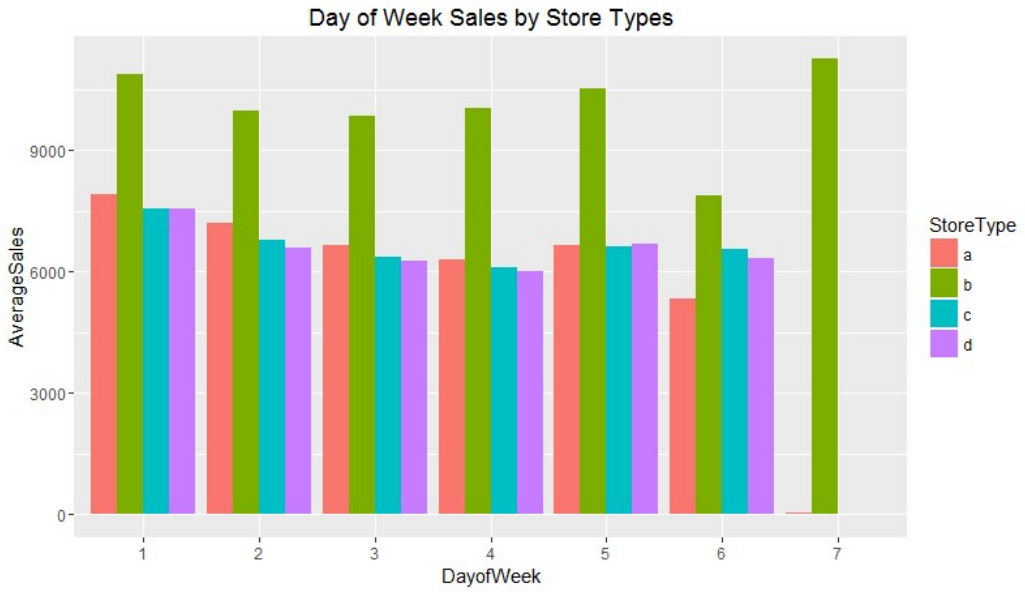

Figure 5.19. Day of Week Sales by Store Types

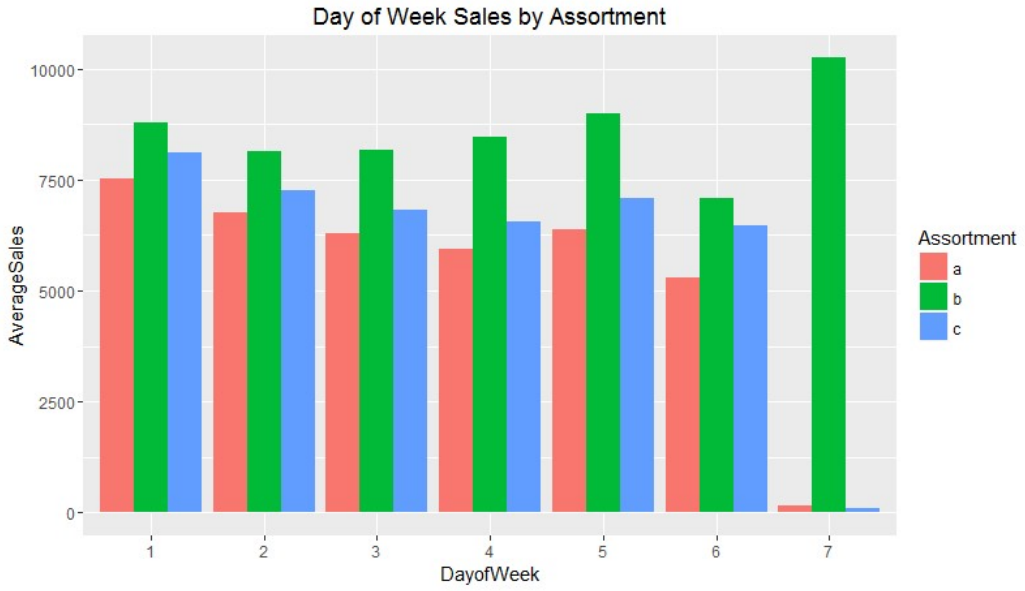

Figure 5.20. Day of Week Sales by Assortment

Figure 5.21 shows promotion effect on sales and Figure 5.22 shows average sales by promotion. As it can be seen by Figure 5.21, highest sales are generally observed when a store has a promotion. Since, Figure 5.21 and Figure 5.22 reveal that sales are increased by promotion in accordance with average sales of each store, variable for the promotion is critical in forecasting sales and included in the proposed model.

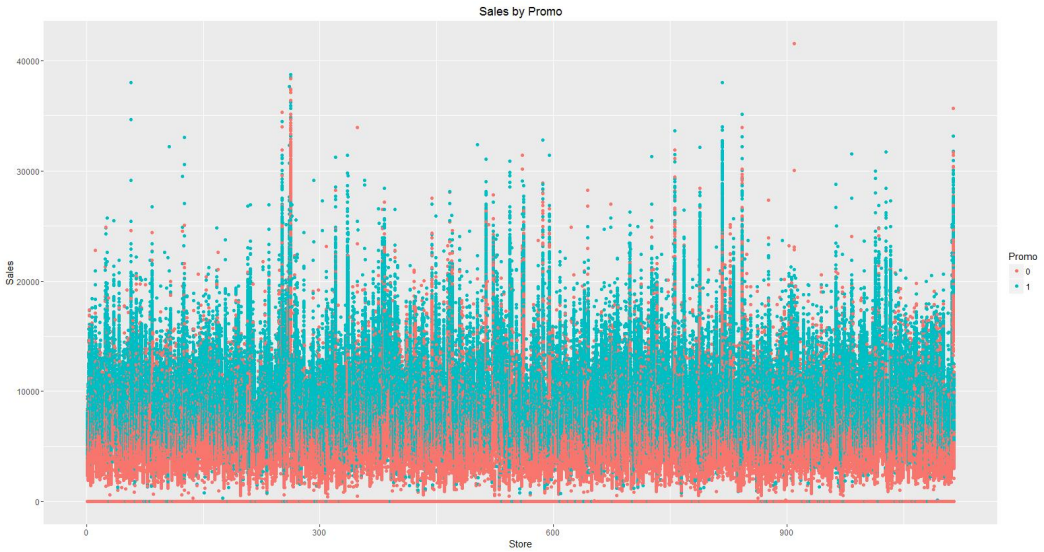

Figure 5.21. Sales by Promo

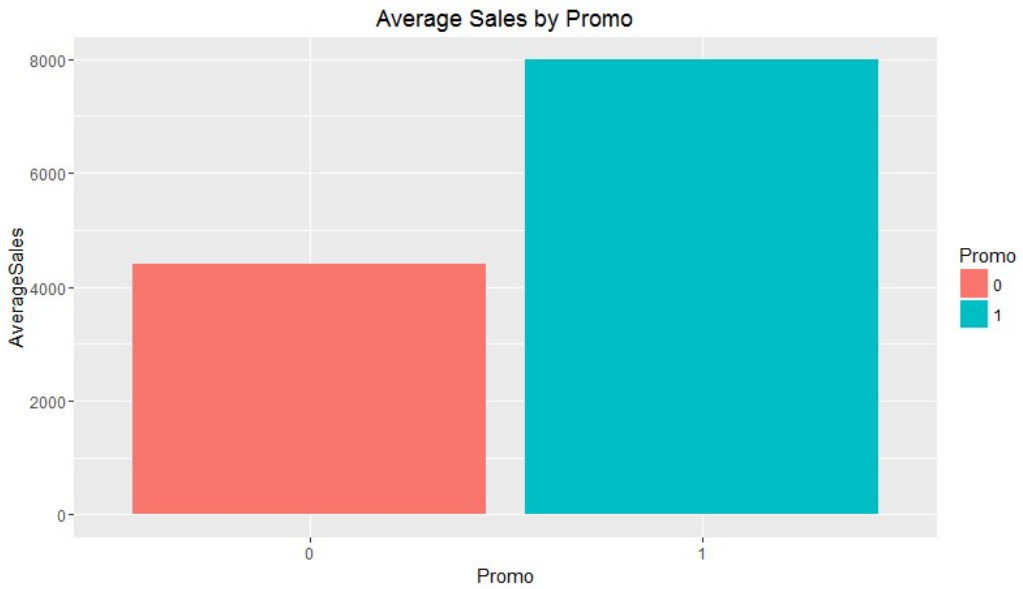

Figure 5.22. Average Sales by Promo

Figure 5.23 show average sales by State Holidays. In general, only “b” type stores and “b” type assortments are open in state holidays. Therefore, average sales are not high in state holidays. Therefore, this variable is not included in the model.

Figure 5.23. Average Sales by State Holiday

Figure 5.24 shows School Holiday effect on sales. Average sales increase when school holiday occurs. Firstly, this variable was added to the model since average sales increase when there is a school holiday. However, results did not change as expected with school holiday. For simplicity, this model also excluded from the model after.

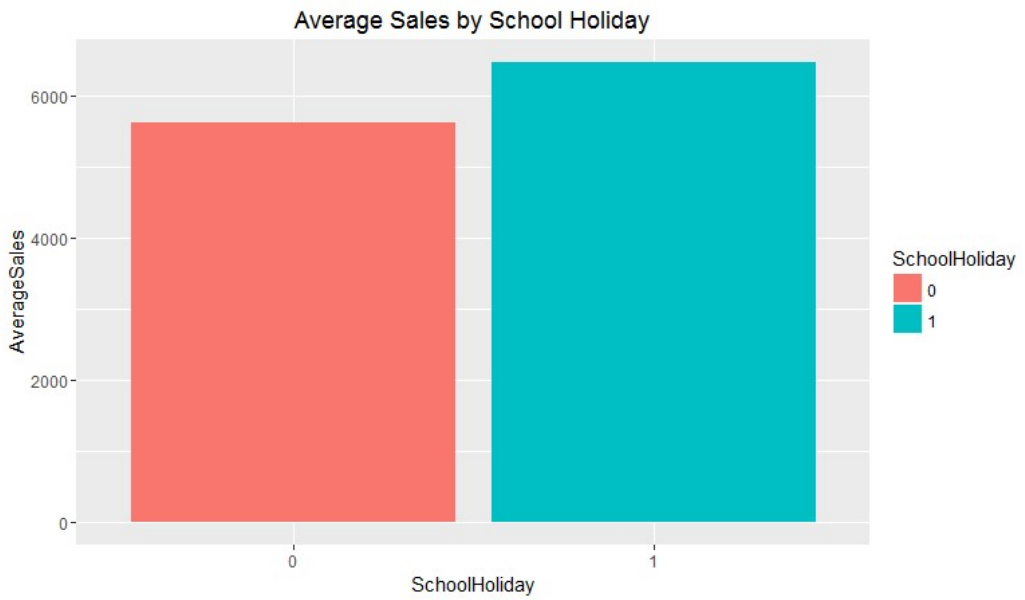

Figure 5.24. Average Sales by School Holiday

Figure 5.25 show continuous promotions (called as “promo2” in dataset) effect on average sales. Average sales decreased when continuous promotion occurs on a store.

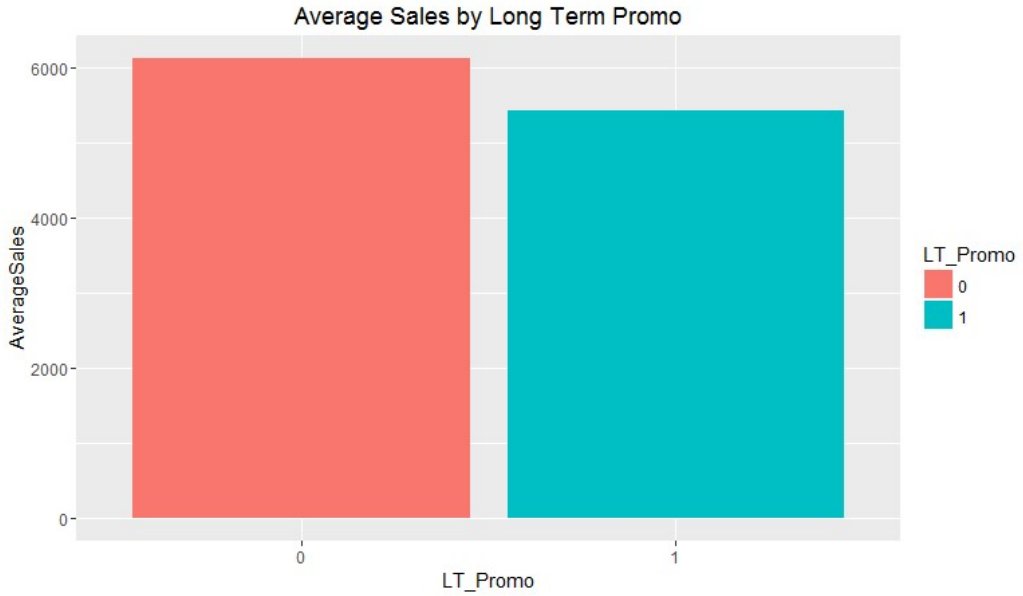

Figure 5.25. Average Sales by Long Term Promo

Figure 5.26 shows monthly average sales by continuous promotion types. As it can be seen from Figure 5.26, continuous promotions has not a critical effect on average sales, therefore exluded from the model.

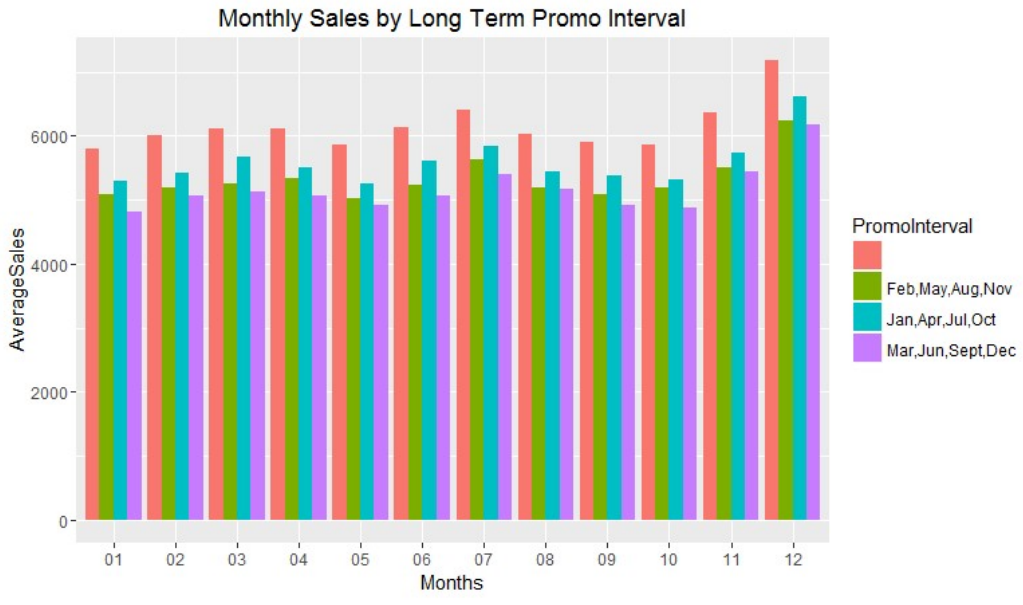

Figure 5.26. Monthly Sales by Long Term Promo

Also, there is another variable for each stores in the store information data; Competition Distance. This variable shows the distance of the closest competitor to that store. Figure 5.27 and Figure 5.28 show average sales of each store type and assortment respectively by competition distance. In general, it is expected that average sales increases when competition distance increases since if there is a competitor near that store customers can choose that store also. However, Figure 5.27 and Figure 5.28 show that competition distance have not any critical effect on sales since when competition distance increases, sales do not increase as expected. Therefore, this model also excluded from the model.

Figure 5.27. Average Sales by Competition Distance for Store Types

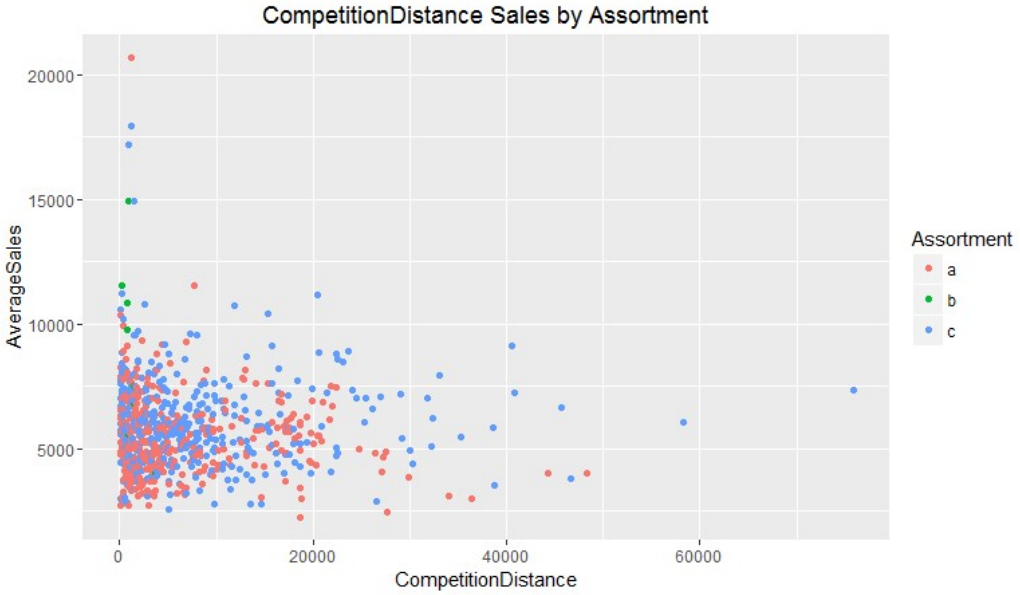

Figure 5.28. Average Sales by Competition Distance for Assortment

5.2.2. Model Implementation

As mentioned in Section 5.2.1, 1115 different stores data provided for daily sales of 2 years and 7 months in Rossmann Sales data. Data divided into two parts; train dataset with 2 years observations (until 01-01-2015) and test data set with 7 months observations (until 31-07-2015). Some observations are excluded when stores are closed since their sales are equal to 0. Model is implemented for step-by-step predictions to predict next month and expand train data each time with a month until test data ends. Figure 5.29 explains time slice. Error parameters calculated by the means of each test data results.

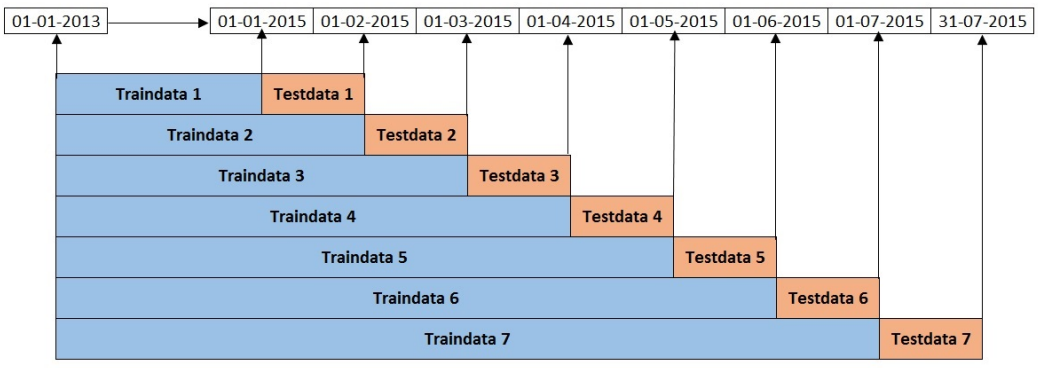

Figure 5.29. Time Slice Summary on Real Data

Three models have been used as base learners to measure performance of proposed algorithm; ARIMAX, Linear Regression and Penalized Regression. Each model consists of 3 regressors as follow;

- Day of Week: Indicates day of the week
- Promo: Indicates whether store have a promotion on the day or not
- Day Split: Indicates day of the month (beginning, mid or end of the month)

Variable selection performed by considering the effects on the sales in accordance with Section 5.2.1 results and optimizing model effectivity by decreasing running time and complexity. More trials can be performed as explained in the next Section 6 Future work.

Each model is implemented for 90 out of 1115 stores, which are selected by their average sales values. 30 stores have lowest average sales, 30 stores have around mean average sales and 30 stores have highest average sales. Figure 5.30 shows sales graphics of those 90 stores.

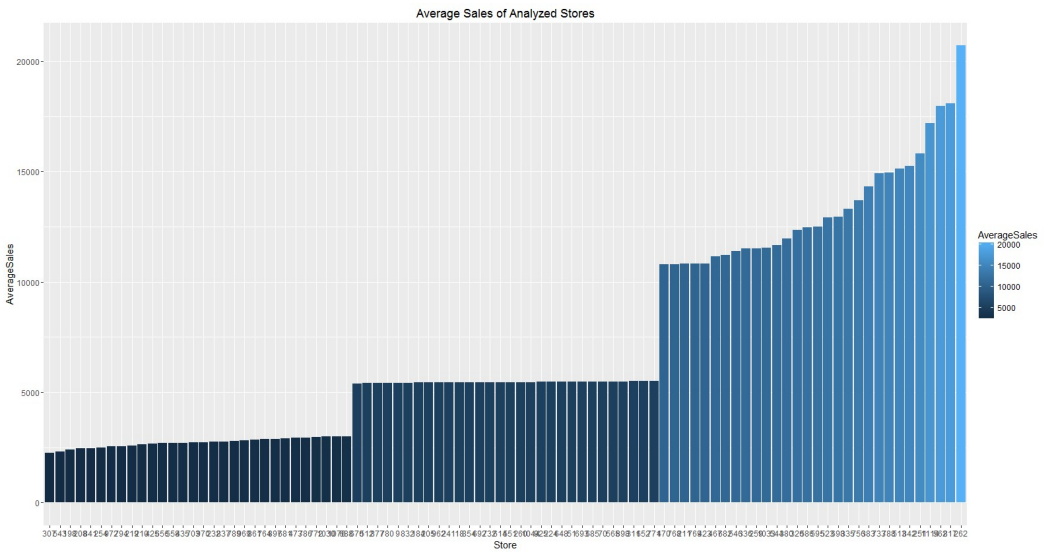

Figure 5.30. Average Sales of Analyzed Stores

Figure 5.31 and Figure 5.32 show characteristic of selected 90 stores with respect to type of store and assorment respectively.

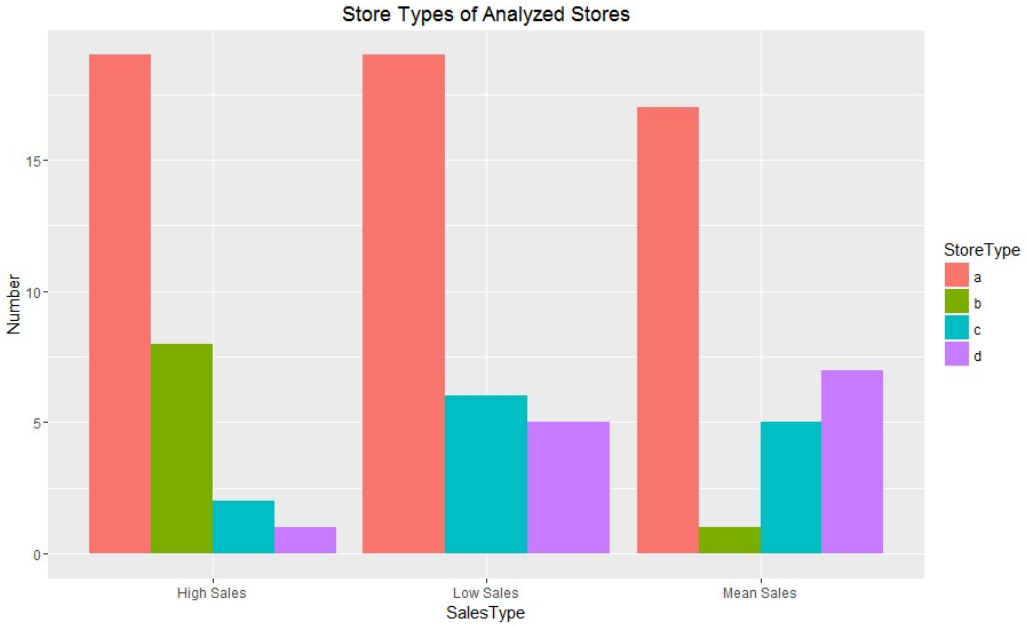

Figure 5.31. Store Types of Analyzed Stores

Figure 5.32. Assortments of Analyzed Stores

5.2.2.1. ARIMA with Residual Learning Model. ARIMAX model starts forecasting only with Sales data. First model built with only previous sales information. Since forecasting performed only with previous lags, residuals are higher in first models of each store. In the second step of our algorithm, residuals are added to regression tree with regressors and efffect of regressors on residuals revealed by below Eq. 5.2;

$$Residuals \sim DayOfWeek + Promo + DaySplit \tag{5.2}$$

Highest residual's source is detected with regression tree and the source is added as a regressor to the model. In this case, model optimizes itself each time by adding new regressors each "tr" iteration until improvement of MSE is less than 5% or regression tree stops to generate more trees. Final model for the store is built when model stops to add new regressors and predictions performed on test data. Results gathered and train data is expanded one more month each time until the end of all data. Each time when train data is expanded one more month, model starts from the beginning for that train data and add new regressors each time. At the end of 7 months test data, we have different models for each one-month test data and therefore different results for each test data. Model's performance is calculated by the mean of each model on each test data. Proposed ARIMAX model is compared with two different ARIMAX model, one is without any regressors (ARIMA model) and the other is with all three regressors. Results is demonstrated in Section 5.2.4 Results

5.2.2.2. Linear Regression with Residual Learning Model. Linear regression model starts with three regressors. Algorithm is exactly the same as applied on ARIMAX model. Model starts forecasting, residuals are added to regression three with three regressors, the highest error source is added to model and model continues to improve itself by adding new regressors until thresholds explained in Section 5.2.2.1. Nonlinear regressors are added to the linear regression model by stage two since all regressors are already provided to model and stage two only creates nonlinear regressors. Created model compared with the same linear regression model with three regressors. Results is demonstrated in Section 5.2.4 Results.

5.2.2.3. Penalized Regression with Residual Learning Model. Penalized regression model was also used as base learner in Stage 1 to evaluate proposed algorithm's performance. Different than previous models, two lags; last day and last week observations, were added as regressors in the beginning model when penalized regression is used for base learner. Ten-fold cross validation is applied in the first stage for model implementation. Same three regressors were to model in Stage 2 regression tree to find highest error source. Created model compared with penalized regression model with all five regressors added. Results is demonstrated in Section 5.2.4 Results.

5.2.3. Model Interpretation

In this section, model interpretation on a single store will be explained. Store 817 is one of the highest sales store in the real data set. ARIMAX model with residual explanation is implemented to Store 817. In the first stage of proposed approach for Store 817, ARIMA model started with only previous observartions. Figure 5.33 shows prediction results for initial model below;

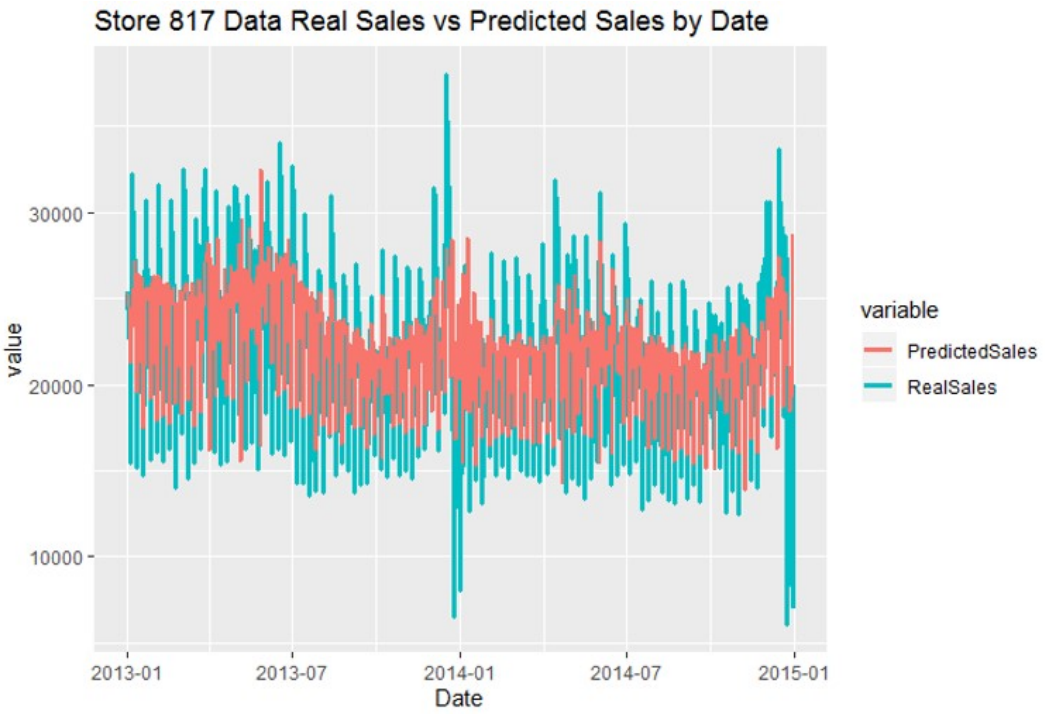

Figure 5.33. Predictions for initial model on Store 817

Results from initial predictions are collected and added to regression tree with three previously defined regressors. Result for regression three is shown in below Figure 5.34;

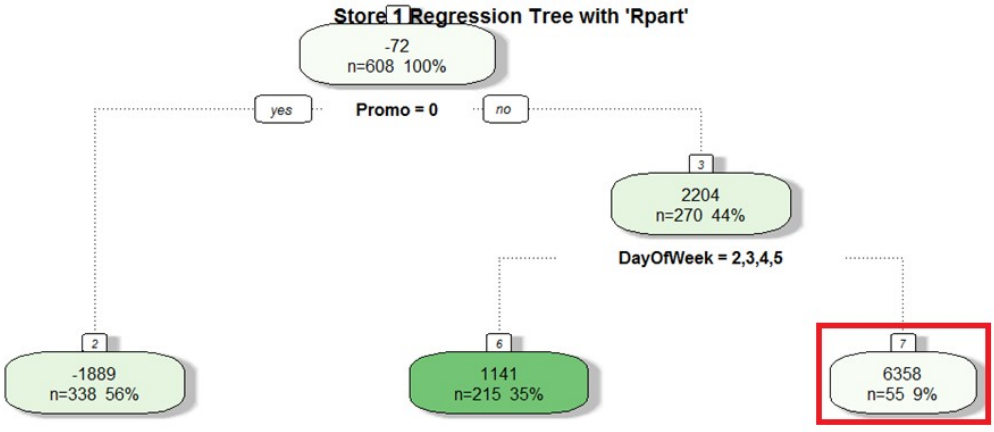

Figure 5.34. Initial regression tree results for Store 817

As it can be seen from Figure 5.34, highest errors occurred on $Node7$ which is $Promo = 1$ and $DayofWeek = (1, 6, 7)$ observations. This means that inital ARIMA model made highest errors when it is Monday, Saturday or Sunday and there is a promotion on those days. According to results from Stage 2 regression tree, model continues with Stage 1 with previous observations and new binary variable which indicate whether it is Monday, Saturday or Sunday with a promotion. Figure 5.35 shows prediction results for second model which has a new binary variable.

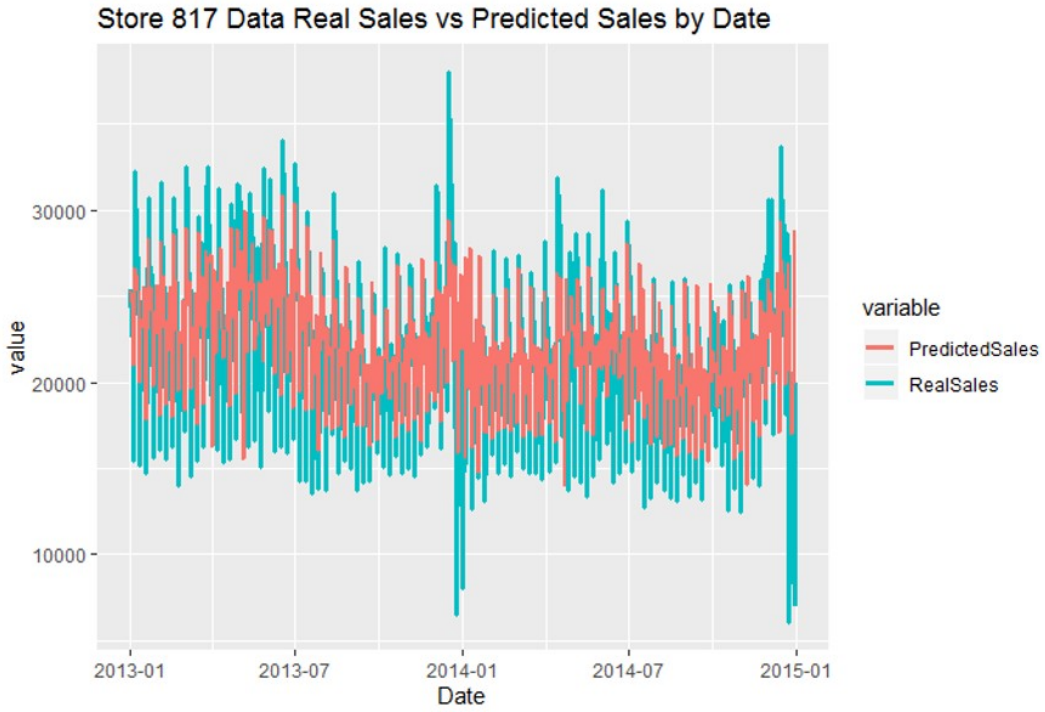

Figure 5.35. Predictions for second model on Store 817

Figure 5.35 clearly shows that model started to catch some higher sales with its new binary variable. Then model iteratively continues with new residuals and find new regressors for original models. It adds new regressors until conditions on stopping rules explained before provided.

5.2.4. Results

5.2.4.1. Results for ARIMA with Residual Learning Model. Proposed algorithm converged after 4th iteration on average when ARIMA is used as base learner in the first stage. Figure 5.36 shows average AIC improvement in training data after each iteration for final models of three types of store sets. Moreover, Figure 5.37 and 5.38 show error improvements of the proposed model after each iteration on the final models with respect to RMSE and RMSPE respectively. As it can be seen from Figures 5.37 and 5.38, model is improving itself dramatically by adding new variables. The increase on RMSPE value from third to forth iteration is caused by two stores which have very small sales in one observation (<50) although their average sales are high (>4500).

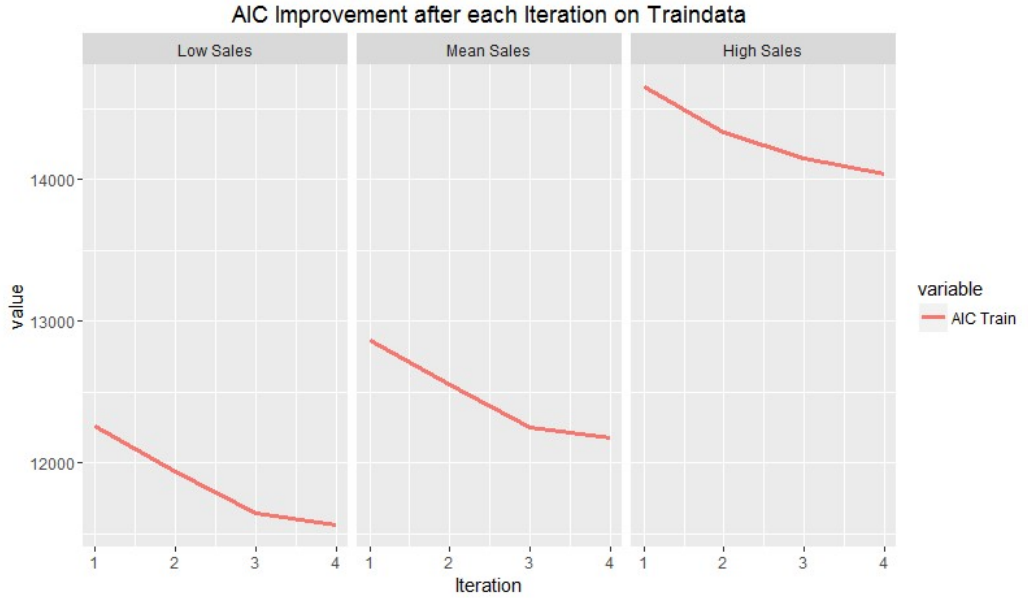

Figure 5.36. AIC Improvement after each Iteration on Traindata for Proposed Algorithm on ARIMAX

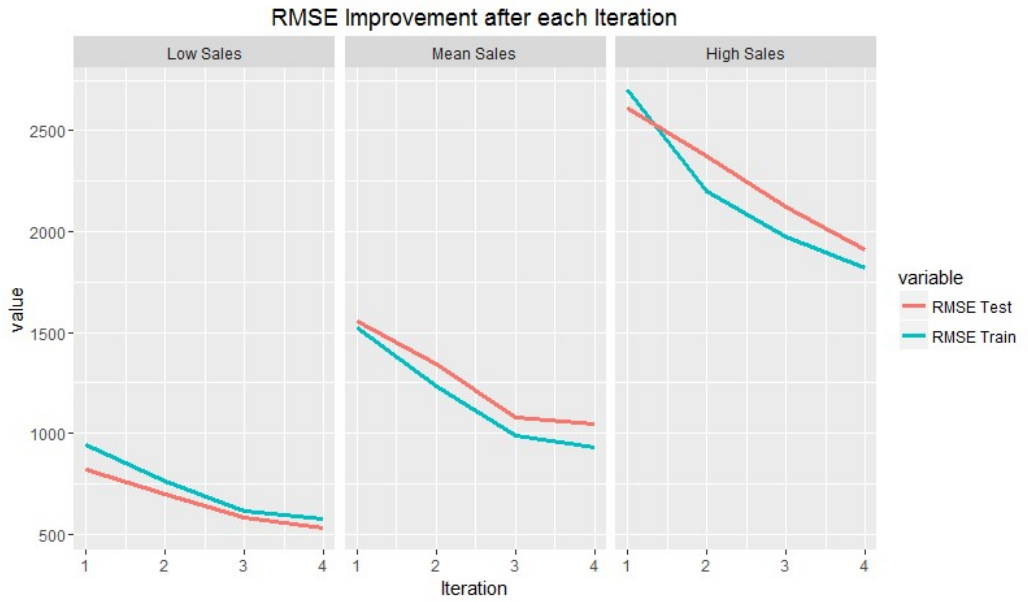

Figure 5.37. RMSE Improvement after each Iteration for Proposed Algorithm on ARIMAX

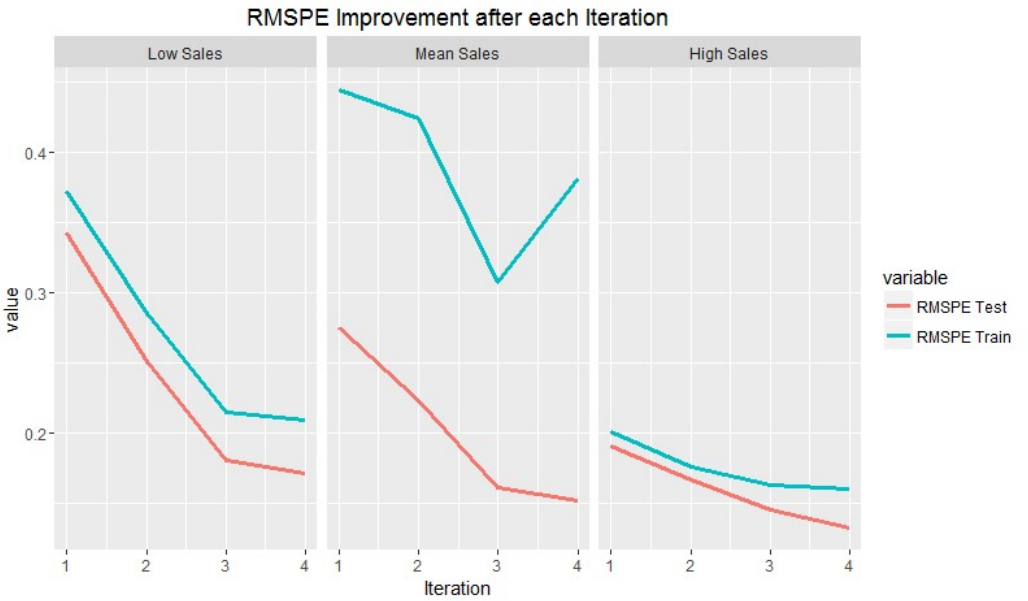

Figure 5.38. RMSPE Improvement after each Iteration for Proposed Algorithm on ARIMAX

In next pages, proposed model results and comparison with single ARIMA model and ARIMAX model with 3 regressors are demonstrated in several figures. Notation is as follow;

- Model 1: Proposed ARIMA with residual learning model
- Model 2: General ARIMA model without any regressor
- Model 3: ARIMA model with regressors (ARIMAX)

Figure 5.39 and 5.40 show comparison between each model of 30 stores with lowest average sales for RMSE and RMSPE respectively;

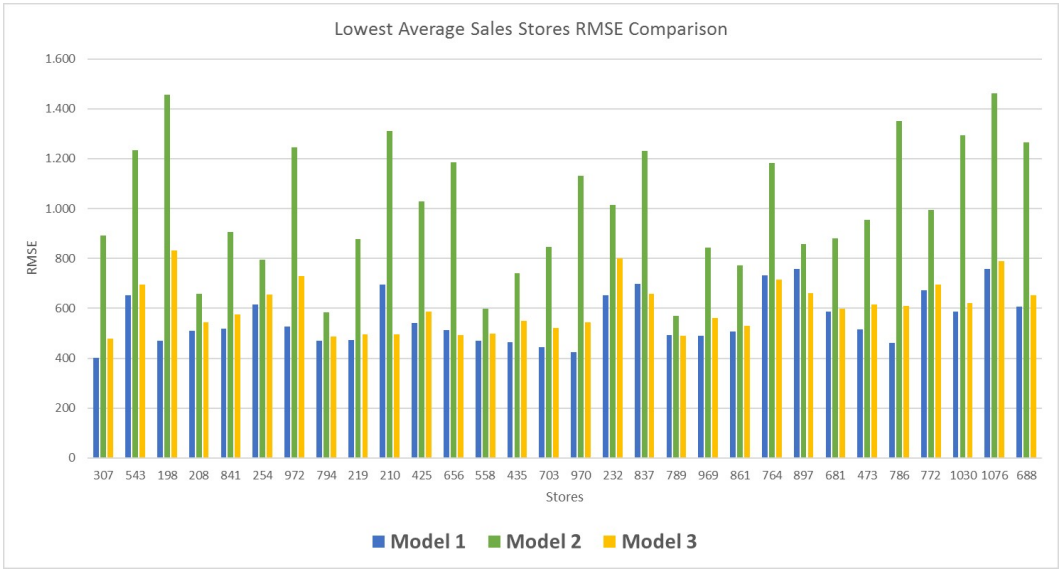

Figure 5.39. Comparison of 30 Stores with Lowest Average Sales as per RMSE for Proposed Algorithm on ARIMAX

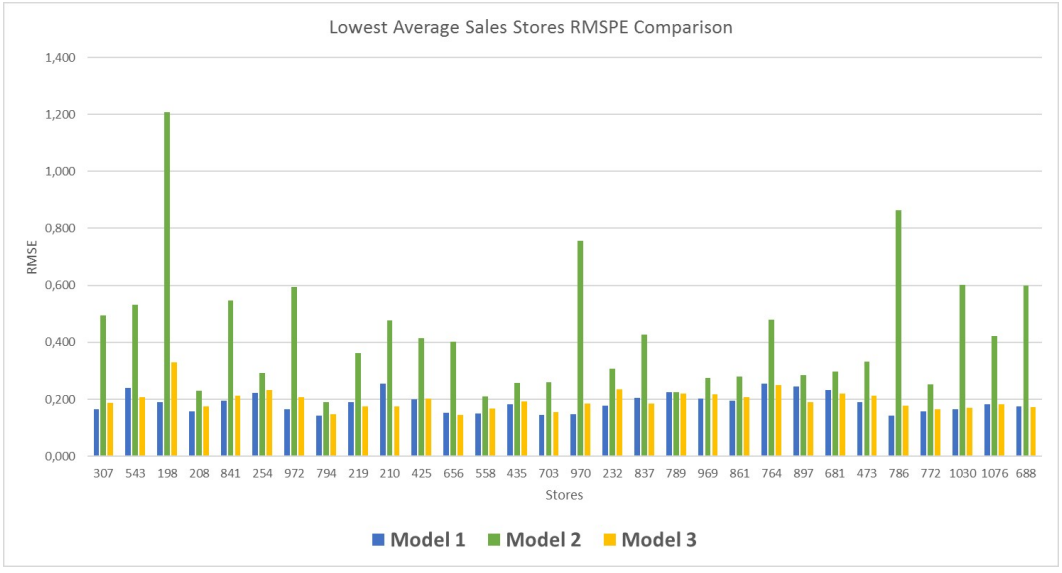

Figure 5.40. Comparison of 30 Stores with Lowest Average Sales as per RMSPE for Proposed Algorithm on ARIMAX

Figure 5.41 and 5.42 show comparison between each model of 30 stores with mean average sales for RMSE and RMSPE respectively;

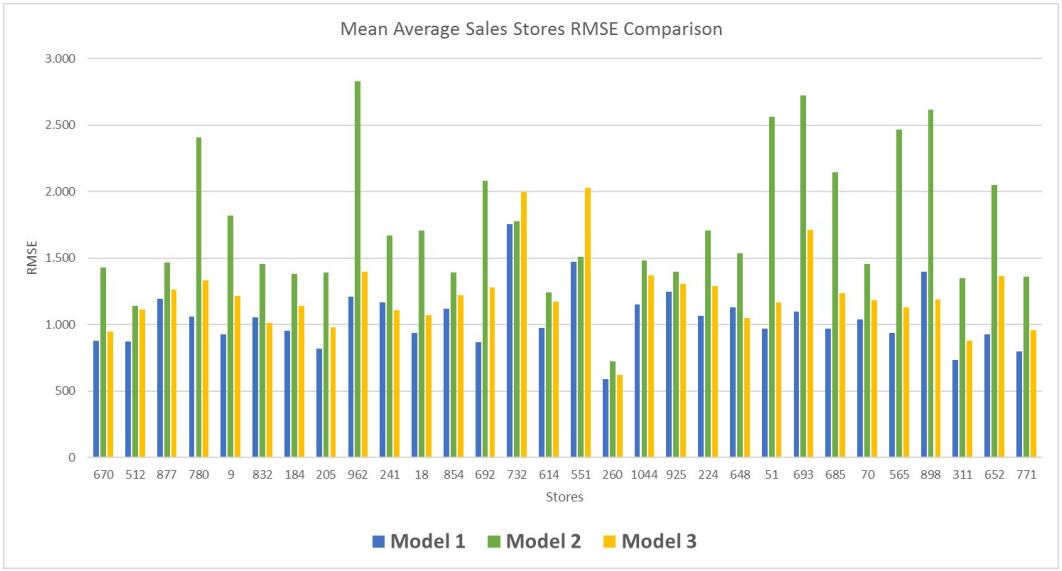

Figure 5.41. Comparison of 30 Stores with Mean Average Sales as per RMSE for Proposed Algorithm on ARIMAX

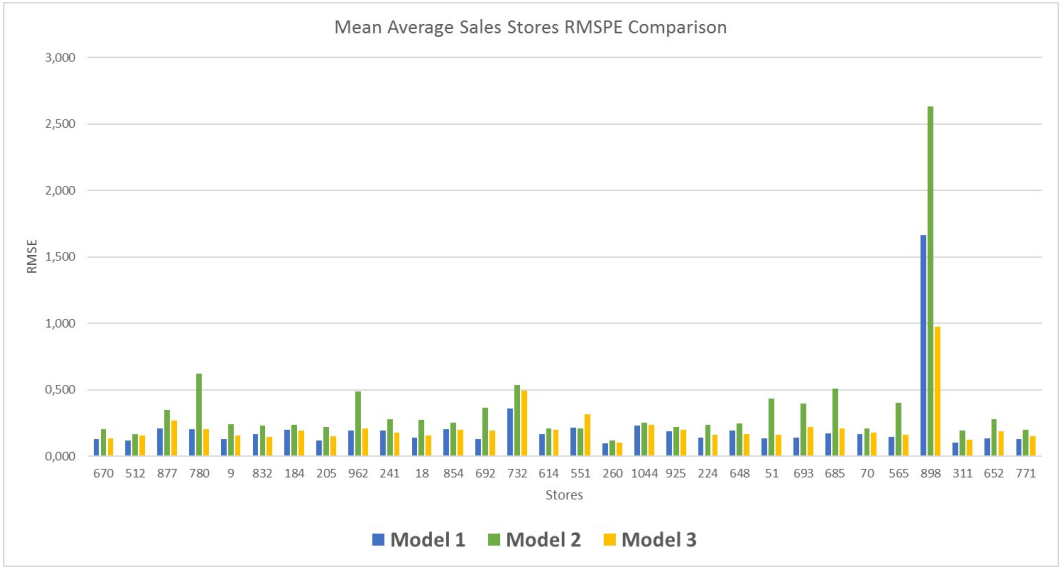

Figure 5.42. Comparison of 30 Stores with Mean Average Sales as per RMSPE for Proposed Algorithm on ARIMAX

Figure 5.43 and 5.44 show comparison between each model of 30 stores with highest average sales for RMSE and RMSPE respectively;

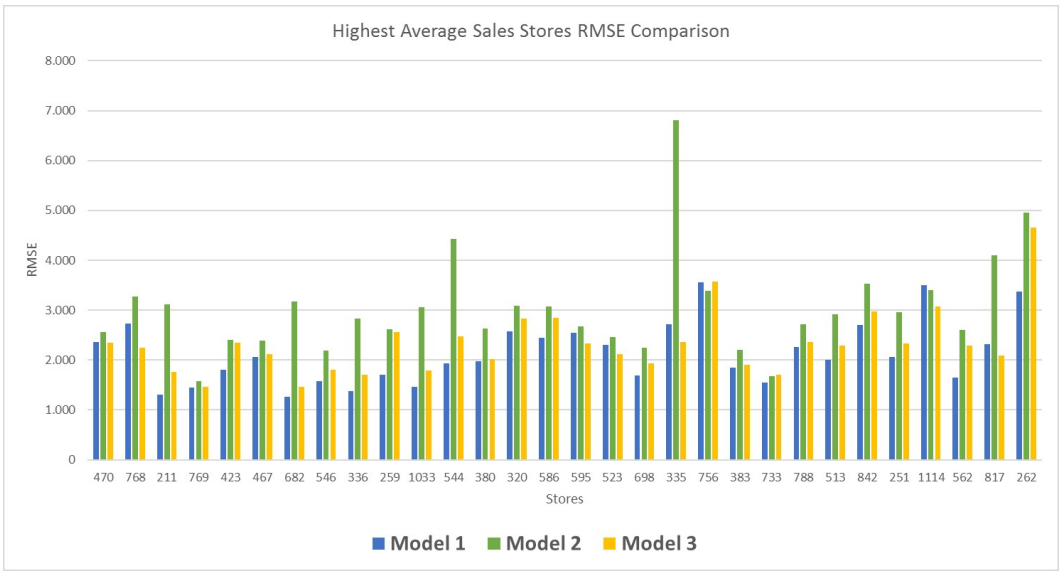

Figure 5.43. Comparison of 30 Stores with Highest Average Sales as per RMSE for Proposed Algorithm on ARIMAX

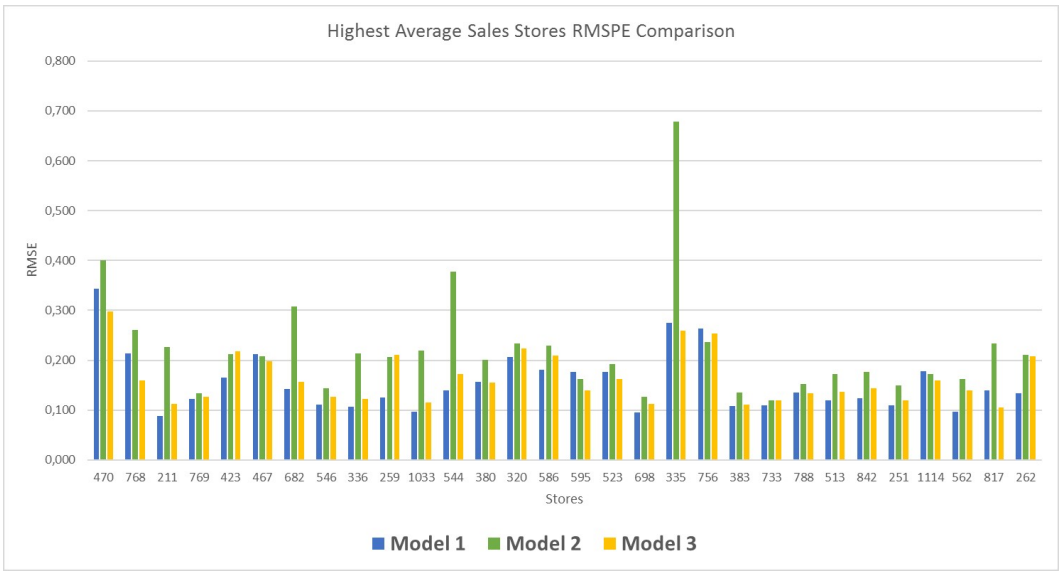

Figure 5.44. Comparison of 30 Stores with Highest Average Sales as per RMSPE for Proposed Algorithm on ARIMAX

As it can be seen from figures, proposed ARIMAX model outperformed single ARIMA model clearly. Although, difference between performance of proposed model and ARIMAX model in some stores is not critical, however, proposed model clearly has better performance. The reason behind small difference can be lack of nonlinear relation between added variables on Sales. Also, when we compare mean result for each type of stores of the three model, proposed algorithm's performance seem better as seen in Table 5.7

Table 5.7. Average Results of Each Model with ARIMA/ARIMAX.

Store Type	**RMSE Model 1**	**RMSE Model 2**	**RMSE Model 3**	**RMSPE Model 1**	**RMSPE Model 2**	**RMSPE Model 3**
Lowest	557	1005	605	0.188	0.429	0.197
Mean	1044	1743	1224	0.217	0.374	0.220
Highest	2138	3034	2325	0.155	0.222	0.164

5.2.4.2. Results for Linear Regression with Residual Learning Model. Proposed algorithm on Linear Regression is also converged after 4th iteration on average. Figure 5.45 shows average R-squared improvement in training data after each iteration for final models of three types of store sets. Moreover, Figure 5.46 and 5.47 show error improvements of the proposed model after each iteration on the final models with respect to RMSE and RMSPE respectively. Below figures reveals that proposed approach is improving itself cleary by adding new variables. The increase on RMSPE value from third to forth iteration is caused same reason as explained in Section 5.2.4.1.Two stores have very small sales in one observation (<50) although their average sales are high (>4500).

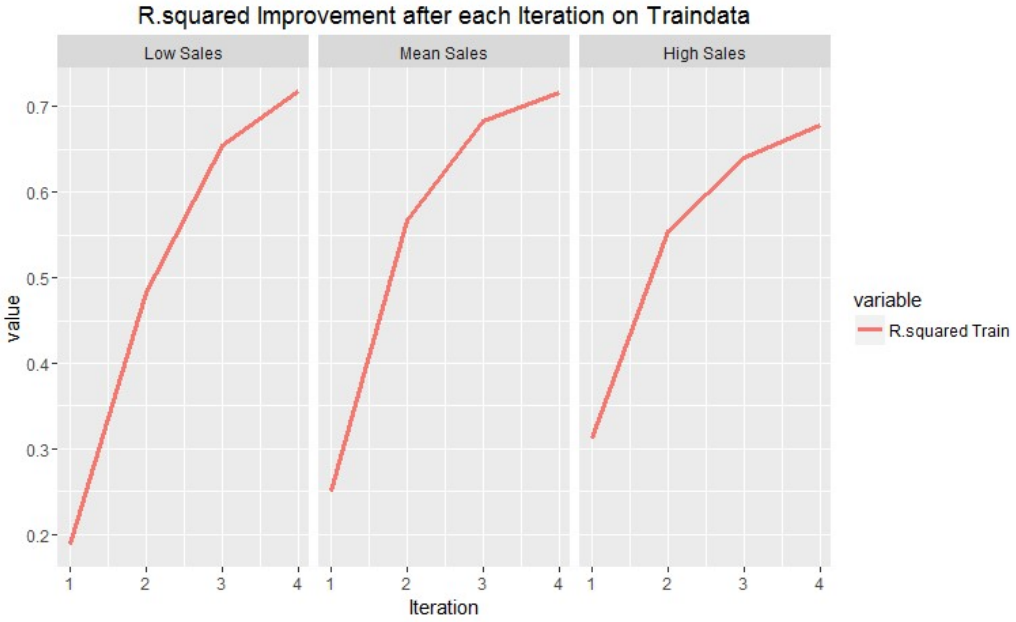

Figure 5.45. R-squared Improvement after each Iteration on Traindata for Proposed Model on Linear Regression

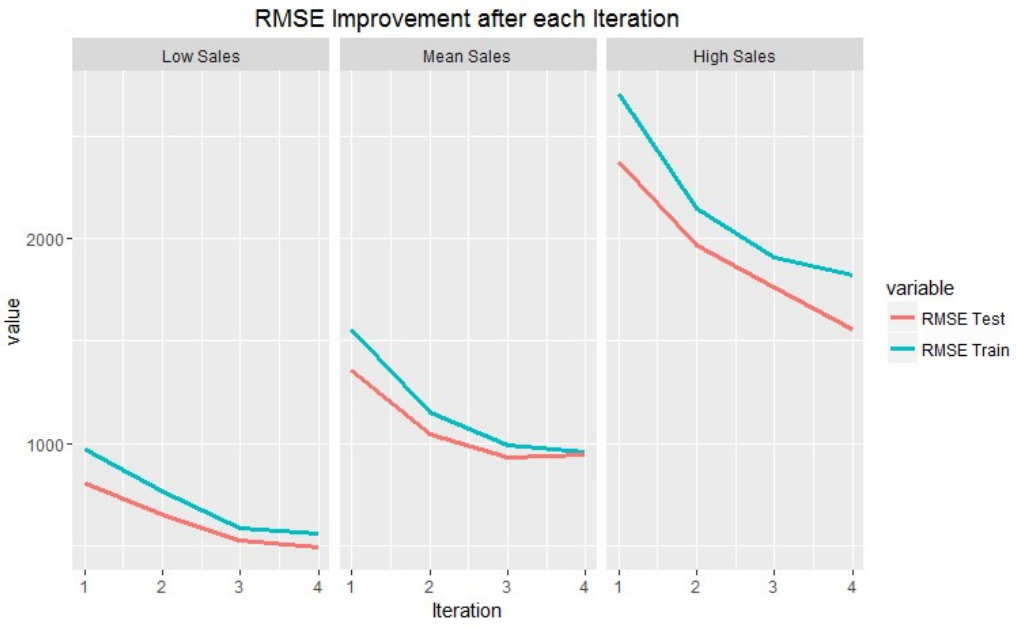

Figure 5.46. RMSE Improvement after each Iteration for Proposed Model on Linear Regression

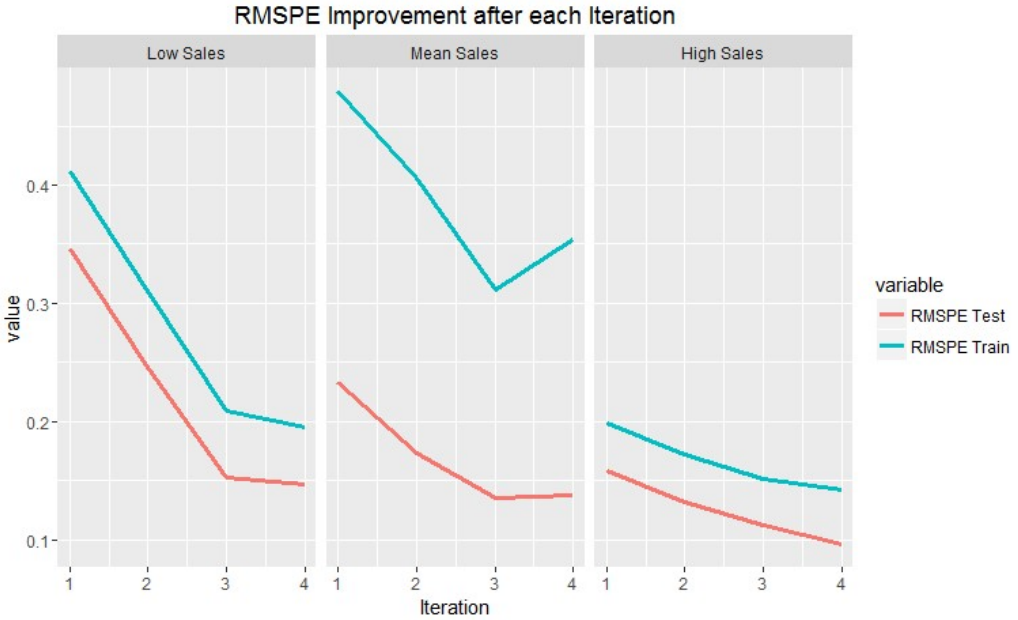

Figure 5.47. RMSPE Improvement after each Iteration for Proposed Model on Linear Regression

In next pages, proposed approach with general linear regression as base learner results and comparison with General Linear Regression model with same regressors are demonstrated in several Figures. Notation is as follow;

- Model 4: Proposed Linear Regression model with Residual Learning Model.
- Model 5: General Linear Regression model with same regressors.

Figure 5.48 and 5.49 show comparison between each model of 30 stores with lowest average sales for RMSE and RMSPE respectively;

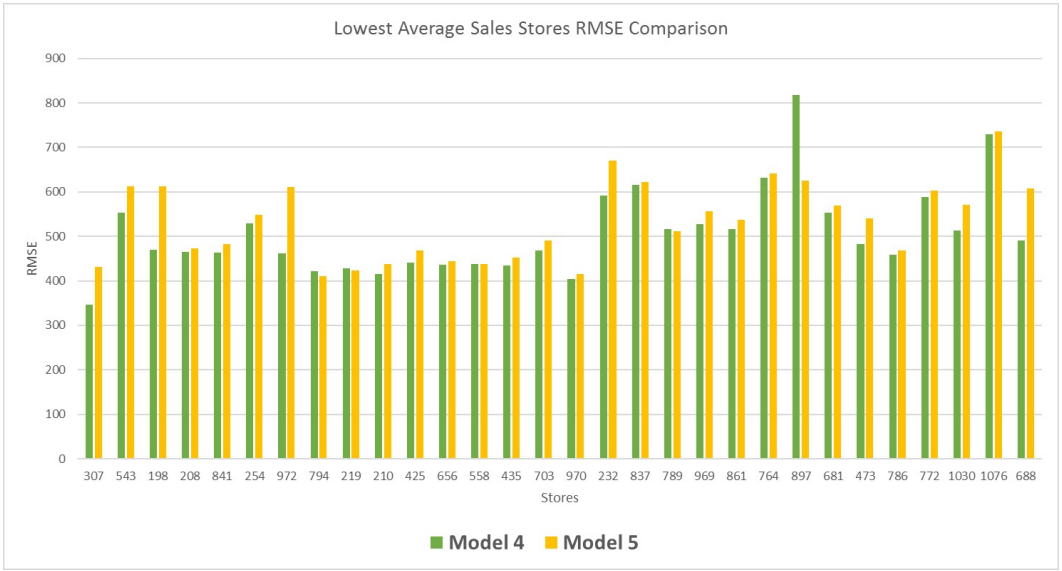

Figure 5.48. Comparison of 30 Stores with Lowest Average Sales as per RMSE for Proposed Model on Linear Regression

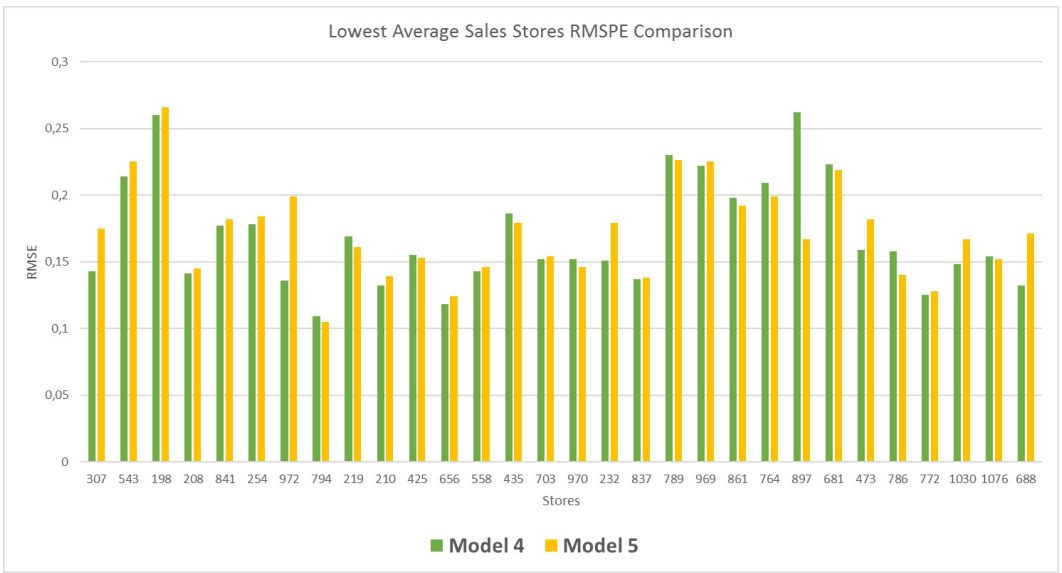

Figure 5.49. Comparison of 30 Stores with Lowest Average Sales as per RMSPE for Proposed Model on Linear Regression

Figure 5.50 and 5.51 show comparison between each model of 30 stores with mean average sales for RMSE and RMSPE respectively;

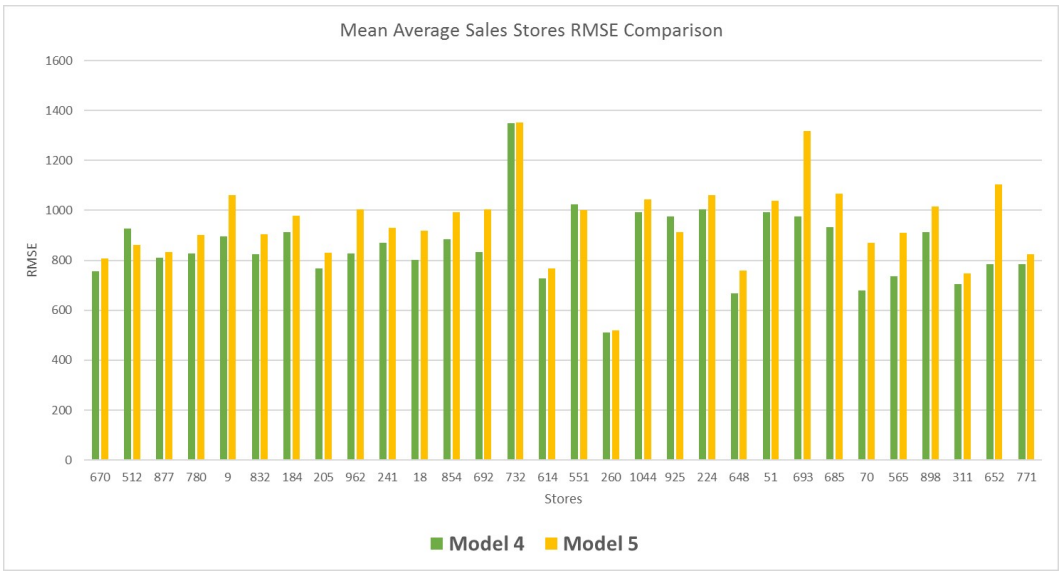

Figure 5.50. Comparison of 30 Stores with Mean Average Sales as per RMSE for Proposed Model on Linear Regression

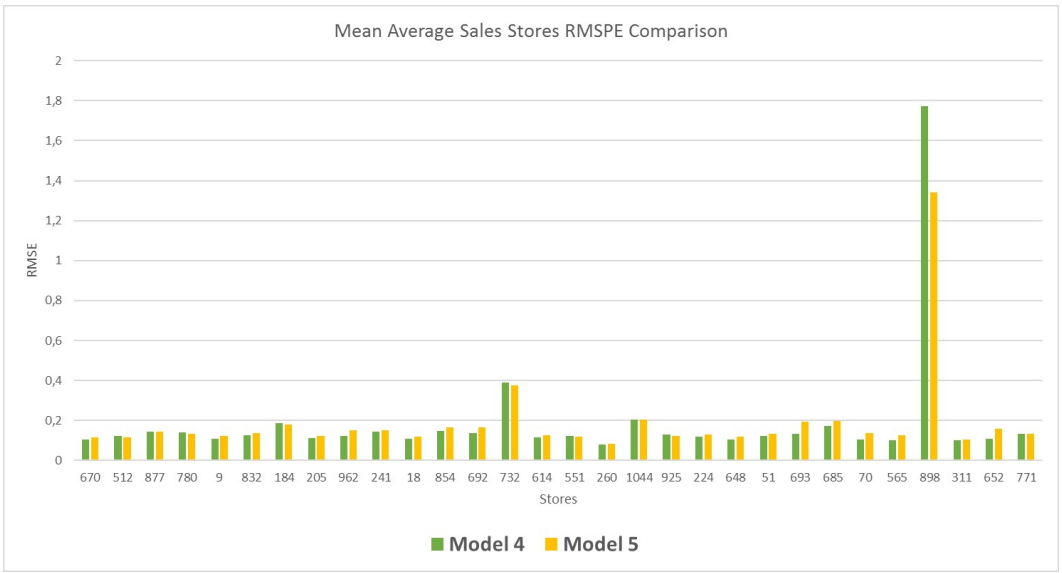

Figure 5.51. Comparison of 30 Stores with Mean Average Sales as per RMSPE for Proposed Model on Linear Regression

Figure 5.52 and 5.53 show comparison between each model of 30 stores with highest average sales for RMSE and RMSPE respectively;

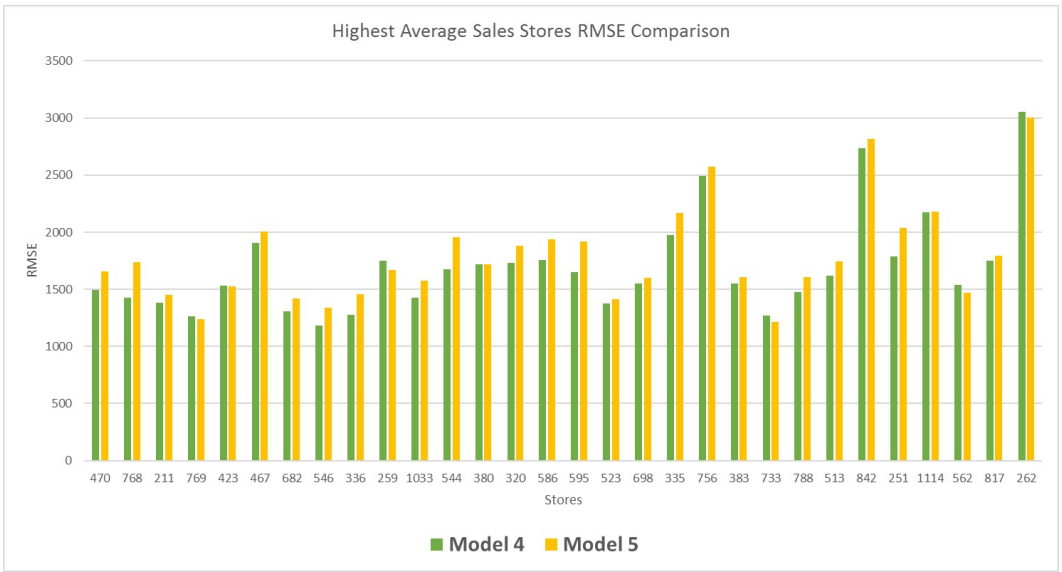

Figure 5.52. Comparison of 30 Stores with Highest Average Sales as per RMSE for Proposed Model on Linear Regression

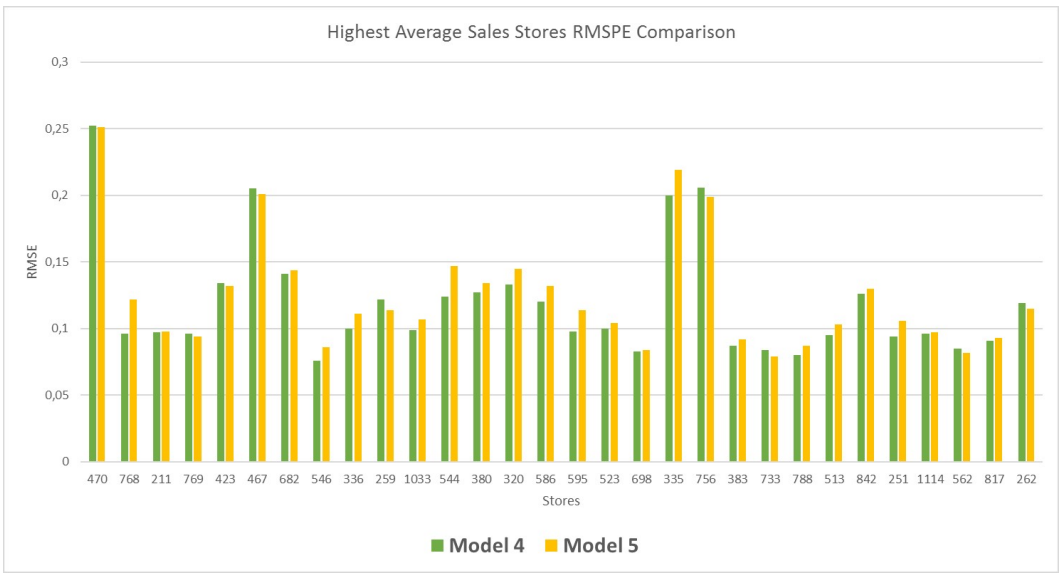

Figure 5.53. Comparison of 30 Stores with Highest Average Sales as per RMSPE for Proposed Model on Linear Regression

As it can be seen from figures, proposed approach with linear regression as base learner has better performance than general regression model. Although, difference between performance of proposed model and general regression model in some stores is not critical, however, proposed model clearly has better performance. However, as it can be seen in Table 5.8 mean result for each type of stores of the proposed approach is better than general regression model.

Table 5.8. Average Results of Each Model with Linear Regression.

Store Type	RMSE Model 4	RMSE Model 5	RMSPE Model 4	RMSPE Model 5
Lowest	507	534	0.169	0.172
Mean	856	944	0.191	0.187
Highest	1694	1791	0.119	0.124

5.2.4.3. Results for Penalized Regression with Residual Learning Model. Figure 5.54, 5.55 and 5.56 show average error improvements of the proposed model for each iteration for R.squared, RMSE and RMSPE respectively. As it can be seen from below figures, model is improving itself by adding new variables. In Figure 5.56, there is an improvement in RMSPE value of training data after third iteration. The reason behind this increment is that the number of the stores which continued after third iteration is few and mean of their RMSPE value is high. Although there is an increment after third iteration on training data RMSPE value, each stores RMSPE values have been decreased after each iteration. Model converged after 5th iteration for proposed algorithm on penalized regression.

Figure 5.54. R-squared Improvement after each Iteration on Traindata for Proposed Model on Penalized Regression

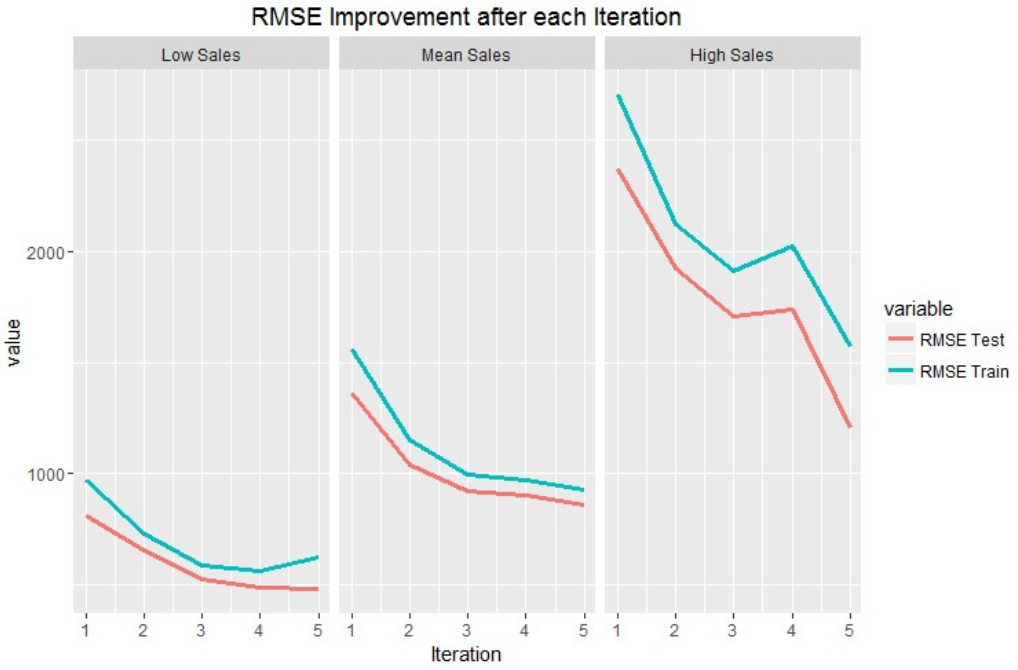

Figure 5.55. RMSE Improvement after each Iteration for Proposed Model on Penalized Regression

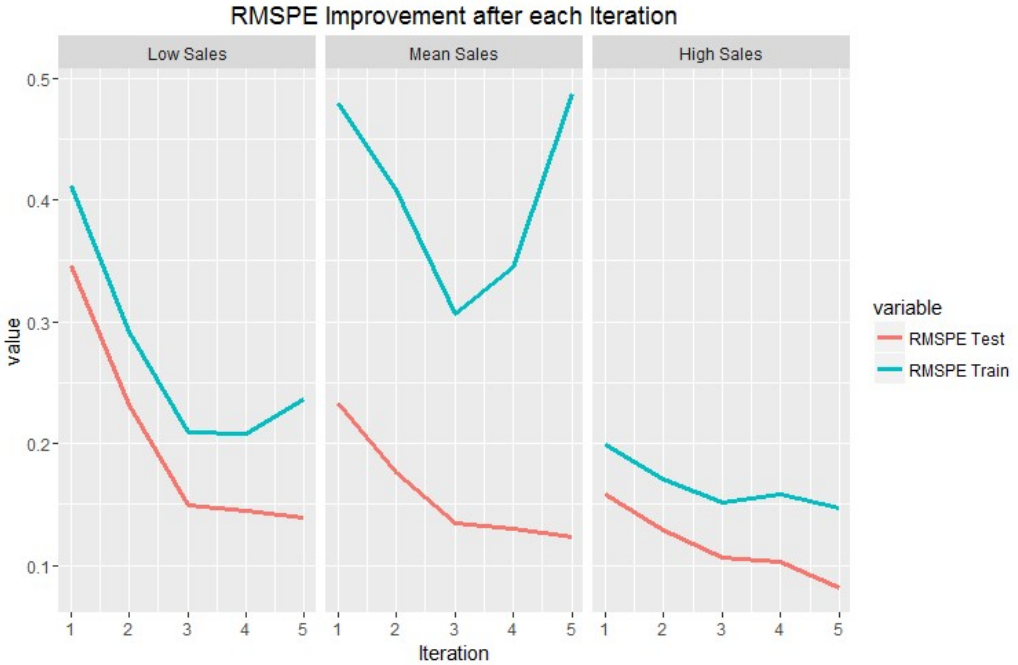

Figure 5.56. RMSPE Improvement after each Iteration for Proposed Model on Penalized Regression

In next pages, Proposed Penalized Regression model results and comparison with Penalized Regression with 3 regressors are demonstrated in several figures. Notation is as follow;

- Model 6: Proposed model with penalized regression as base learner
- Model 7: Penalized Regression model with same regressors

Figure 5.57 and Figure 5.58 shows comparison between each model of low sales stores for RMSE and RMSPE respectively.

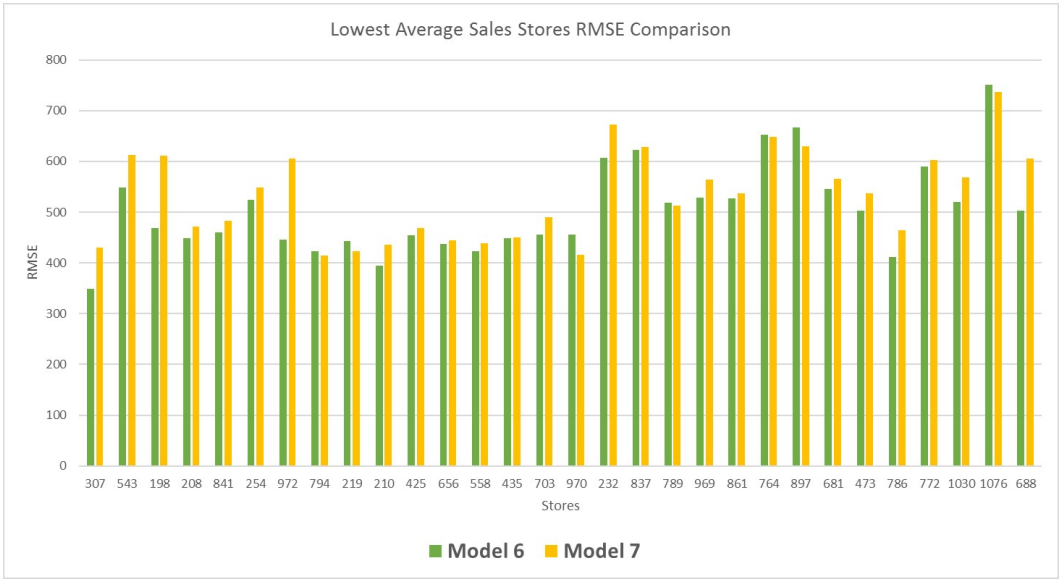

Figure 5.57. Comparison of 30 Stores with Lowest Average Sales as per RMSE for Proposed Model on Penalized Regression

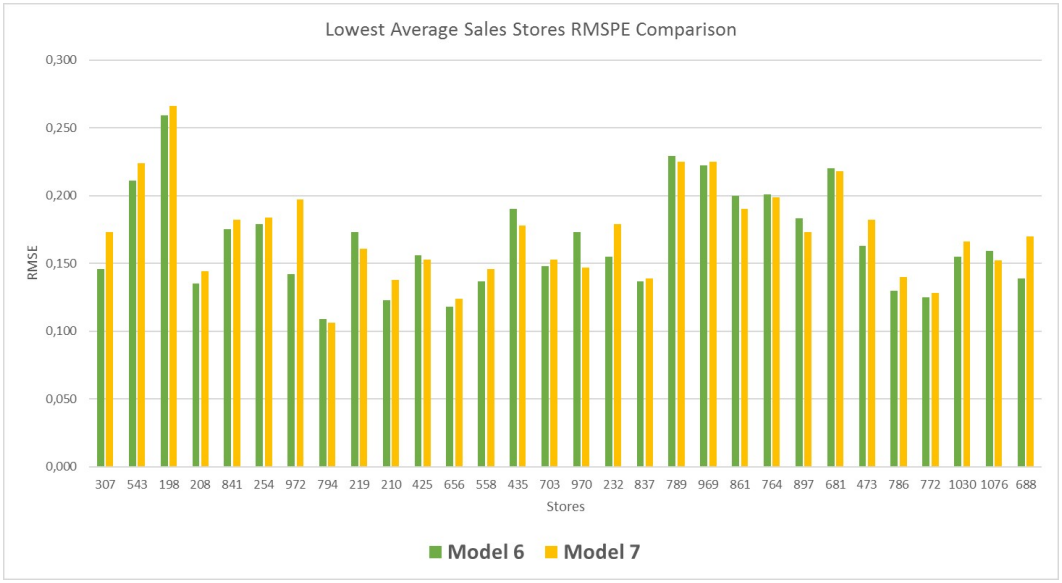

Figure 5.58. Comparison of 30 Stores with Lowest Average Sales as per RMSPE for Proposed Model on Penalized Regression

Figure 5.59 and 5.60 show comparison between each model of 30 stores with mean average sales for RMSE and RMSPE respectively;

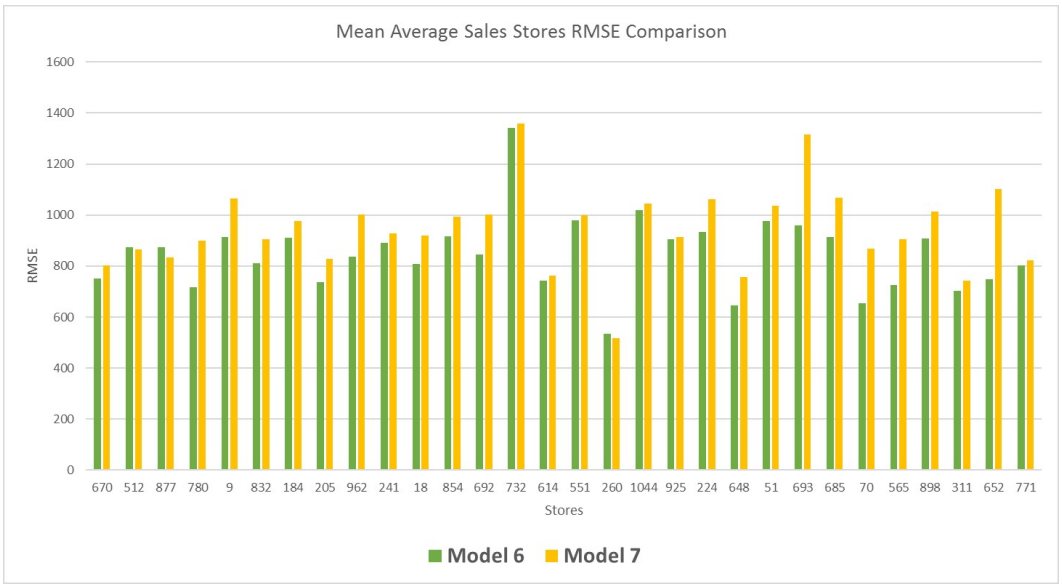

Figure 5.59. Comparison of 30 Stores with Mean Average Sales as per RMSE for Proposed Model on Penalized Regression

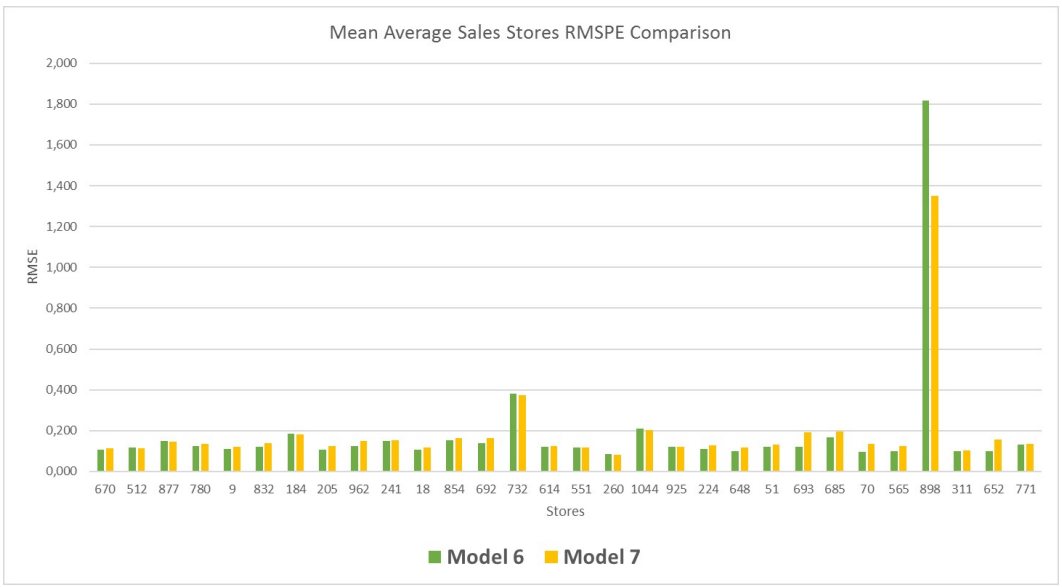

Figure 5.60. Comparison of 30 Stores with Mean Average Sales as per RMSPE for Proposed Model on Penalized Regression

Figure 5.61 and 5.62 show comparison between each model of 30 stores with highest average sales for RMSE and RMSPE respectively;

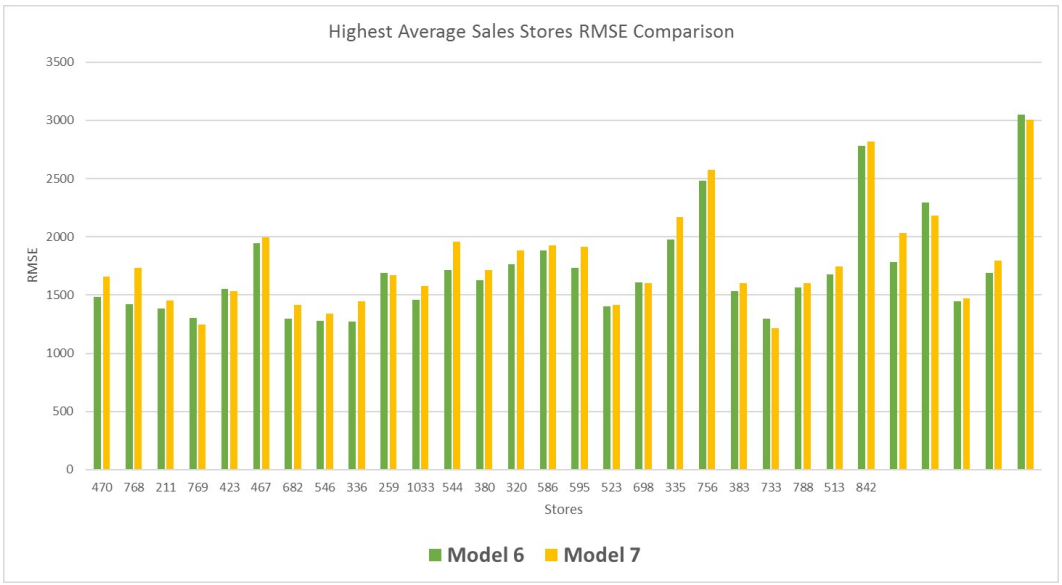

Figure 5.61. Comparison of 30 Stores with High Average Sales as per RMSPE for Proposed Model on Penalized Regression

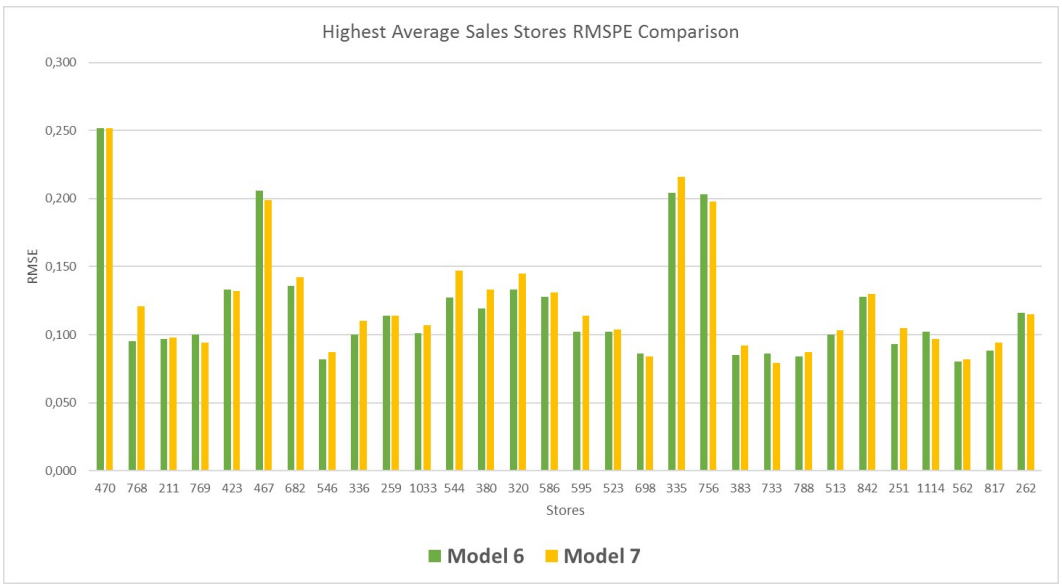

Figure 5.62. Comparison of 30 Stores with High Average Sales as per RMSPE for Proposed Model on Penalized Regression

As it can be seen from figures, proposed model with penalized regression with residual learning slightly have better performance than general penalized regression method. Although, difference between performance of proposed model and general model in some stores is not critical, proposed approach clearly have better performance in general. The reason behind small difference can be lack of nonlinear relation between added variables on Sales. Also, when we compare mean result for each type of stores of the three model, proposed algorithm's performance seem better as seen in Table 5.9

Table 5.9. Average Results of Each Model with Penalized Regression.

Store Type	**RMSE Model 6**	**RMSE Model 7**	**RMSPE Model 6**	**RMSPE Model 7**
Lowest	504	534	0.166	0.172
Mean	845	943	0.189	0.187
Highest	1712	1789	0.119	0.124

5.2.5. Summary of Results and Discussion

In this section, summary results for each proposed model and general models is combined in the following tables. Firstly, model numbers and their definitions are listed below;

- Model 1: Proposed ARIMA with Residual Learning Model
- Model 2: General ARIMA model without any regressor
- Model 3: ARIMA model with regressors (ARIMAX)
- Model 4: Proposed Linear Regression model with Residual Learning Model
- Model 5: General Linear Regression model with same regressors
- Model 6: Proposed Penalized Regression model with Residual Learning Model
- Model 7: General Penalized Regression model with same regressors

Table 5.10 shows average RMSE results of each model for each store types. According to Table 5.10, proposed model on ARIMAX (Model 1) has better average performance compared to regular ARIMA (Model 2) and ARIMAX (Model 3) on each type of stores. Also, proposed model on Linear Regression (Model 4) and Penalized Regression (Model 6) models have better average performance than regular Linear Regression (Model 5) and regular Penalized Regression (Model 7) models with respect to RMSE respectively. If we compare all seven models in accordance with their RMSE results, Model 6 has better performance on Low and Mean Sales Stores and Model 4 has better performance on High Sales Stores.

Table 5.10. Summary Results for RMSE.

	Low Sale Stores	**Mean Sale Stores**	**High Sale Stores**
Model 1: Proposed ARIMA model with Residual Learning	557	1044	2138
Model 2: General ARIMA model without any regressor	1005	1743	3034
Model 3: ARIMA model with regressors (ARIMAX)	605	1225	2325
Model 4: Proposed Linear Regression model with Residual Learning	507	856	1694
Model 5: General Linear Regression Model	534	944	1791
Model 6: Proposed Penalized Regression model with Residual Learning	504	845	1712
Model 7: General Penalized Regression Model	533	943	1789

Table 5.11 shows average RMSPE results of each model for each store types. According to Table 5.11, proposed model on ARIMAX (Model 1) has better average performance compared to regular ARIMA (Model 2) and ARIMAX (Model 3) on each type of stores for average RMSPE results. Also, proposed model on Linear Regression (Model 4) and Penalized Regression (Model 6) models have better average performance than regular Linear Regression (Model 5) and regular Penalized Regression (Model 7) models with respect to RMSPE respectively for Low and High Sales Stores. On the other hand, regular models have better performance on Mean Average Stores, however, difference is small. When all seven models compared in accordance with their RMSPE results, Model 6 has better performance on Low Sales Stores, Model 5 and Model 7 have equal results and better then other five models on Mean Sales Stores. For High Sales Stores, Model 4 and Model 6 has equal and best performance.

Comparison for each three type base learner models with their regular competitors in accordance with the number of stores is listed below;

- Proposed algorithm on ARIMAX model (Model 1) had better results than Model 2 for 88 out of 90 stores compared with RMSE result and 85 of 90 stores compared with RMSPE results.
- Proposed algorithm on ARIMAX model (Model 1) had better results than Model 3 for 73 out of 90 stores compared with RMSE result and 63 out of 90 stores compared with RMSPE results
- Proposed algorithm on Linear Regression model (Model 4) had better results than Model 5 for 77 out of 90 stores compared with RMSE result and 61 of 90 stores compared with RMSPE results.
- Proposed algorithm on Penalized Regression model (Model 6) had better results than Model 7 for 73 out of 90 stores compared with RMSE result and 64 of 90 stores compared with RMSPE results.

Difference between some stores are critical. The reason behind this can be nonlinear relations effects on those stores. On the other hand, there is not a critical difference between some stores which can be explained with opposite of the same reason.

Table 5.11. Summary Results for RMSPE.

	Low Sale Stores	**Mean Sale Stores**	**High Sale Stores**
Model 1: Proposed ARIMA model with Residual Learning	0.188	0.217	0.155
Model 2: General ARIMA model without any regressor	0.429	0.374	0.222
Model 3: ARIMA model with regressors (ARIMAX)	0.199	0.220	0.164
Model 4: Proposed Linear Regression model with Residual Learning	0.169	0.191	0.119
Model 5: General Linear Regression Model	0.172	0.187	0.124
Model 6: Proposed Penalized Regression model with Residual Learning	0.166	0.189	0.119
Model 7: General Penalized Regression Model	0.172	0.187	0.124

We also made Friedman test (Friedman, 1940) to see if applied seven models have a difference and then Nemenyi test (Nemenyi, 1962) to observe if there is a critical difference between seven models each other. Nemenyi test relies on the ranks of models according to their RMSE results on 90 stores. According to Friedman test result, p-value is almost zero, which means that seven models have differences. Figure 5.63 shows Nemenyi test results in which critical difference value found as 0.95257. This means that if rank difference is greater than 0.95257, models have a significant difference.

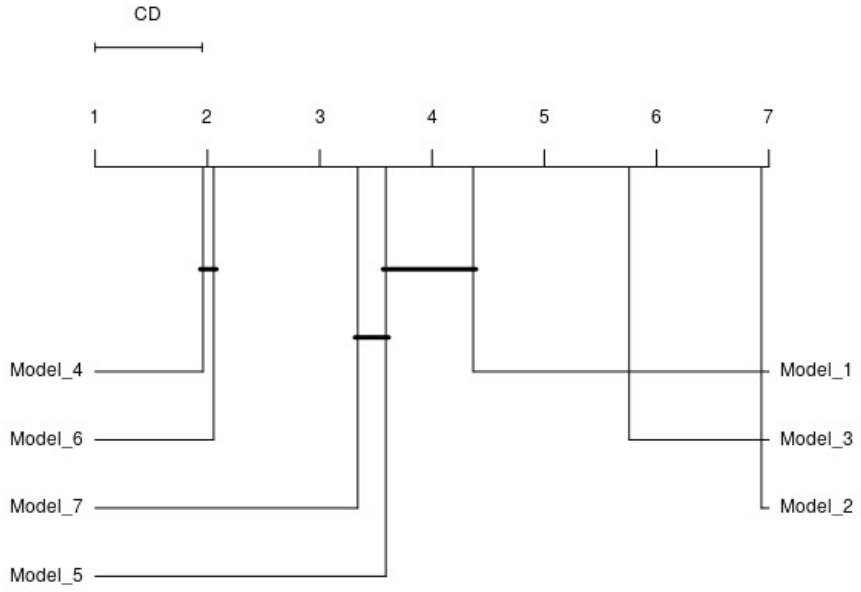

Figure 5.63. Nemenyi Test Results for All Models

According to Figure 5.63 results, Model 4 (Proposed Linear Regression model with Residual Learning Model) and Model 6 (Proposed Penalized Regression model with Residual Learning Model) have better results than other four models, however they do not have a significant difference with each other. Also, Model 5 (General Linear Regression model with same regressors) and Model 7 (General Penalized Regression model with same regressors) do not have a significant difference. The reason behind this can be explained that Penalized regression and Linear Regression models used same regressors and Penalized Regression model did not perform variable exclusion.

6. CONCLUSION AND FUTURE WORK

In this study, an algorithm for general time series modeling with residual explanation by regression tree is proposed. Proposed algorithm consists of two-stage; first stage is a classical time series model and second stage is new variable selection for the first stage by analyzing residuals of the first model in a regression tree. In this study, ARIMAX, Linear Regression and Penalized Regression models were used separately to analyze contribution of the proposed algorithm. Main contribution of the proposed algorithm is the second stage where residuals are interpreted with a regression tree to find new variables which cause highest prediction errors in the first model. Residual explanation in the second stage is also the main difference from the many works in literature since there are a few models focus on residuals interpretation. In the second stage, not only linear relations but also nonlinear relations can also be added to general time series model used in the first stage. This is the main advantage of the proposed model comparing to general time series models. Since, general time series models only focus on linear relations between regressors and response variable, proposed algorithm have an advantage of adding nonlinear relations. In some cases, as seen in the dataset from Rossmann Stores used in this study, nonlinear relations have a huge impact on response variable. Experimental results in Section 5.2.4.1, Section 5.2.4.2 and Section 5.2.4.3 show the contribution of proposed algorithm on ARIMAX, Linear Regression and Penalized Regression methods in accordance with error improvements.

In addition to its contribution to forecasting results, proposed model is also simple to implement on any general time series models where regressors are available in dataset. When a general forecast is performed and residuals are revealed, proposed algorithm automatically put residuals and regressors in a regression tree and find the best available regressor which causes the highest error in the first forecast and adds it to model and starts to forecast with new regressor continuously.

Moreover, proposed algorithm can be used as a variable selection method for general time series. Since new regressors that cause the highest errors are detected in the second stage, those new regressors are generally the ones that have main contributions on forecasting.

Finally, this research can be a guide for possible researches in the future. Different general regression models can be used in the first stage of the model to analyze the effect of the proposed algorithm. Also, variables used in the second stage to find the highest errors of the first model is selected from data analysis in Section 5.2.1 in this study. The aim of this approach is to decrease run time, model complexity and possible errors as rank-deficiency. However, in the future works, all available and possible new externals variables can be added to the second stage for variable selection. Also, pooling of all available store data can be performed.

The proposed algorithm in this study is a two-stage general time series forecasting technique with a residual explanation to analyze linear and nonlinear relations between regressors and response variable to add most effective variable into the model. Second stage of the proposed algorithm makes it a unique approach within literature due to revealing and adding nonlinear relations into the model by residual explanation since general time series models consider only linear relations and there are few models focuses on residual explanation in different ways. Proposed algorithm in this study possibly may lead to new ways to more researches on the general time series models.

REFERENCES

Aburto, L. and R. Weber, "Improved supply chain management based on hybrid demand forecasts", *Applied Soft Computing Journal*, Vol. 7, No. 1, pp. 136–144, 2007.

Adhikari, R. and R. K. Agrawal, *An Introductory Study on Time series Modeling and Forecasting*, LAP Lambert Academic Publishing, Germany, 2013.

Akaike, H., *Information Theory and an Extension of the Maximum Likelihood Principle*, pp. 199–213, Springer New York, New York, 1973.

Aladag, C. H., E. Egrioglu and C. Kadilar, "Forecasting nonlinear time series with a hybrid methodology", *Applied Mathematics Letters*, Vol. 22, No. 9, pp. 1467 – 1470, 2009.

Alon, I., "Forecasting aggregate retail sales: the Winters' model revisited", *The 1997 Annual Proceedings. Midwest Decision Science Institute*, 1997.

Alon, I., M. Qi and R. J. Sadowski, "Forecasting aggregate retail sales:: a comparison of artificial neural networks and traditional methods", *Journal of Retailing and Consumer Services*, Vol. 8, No. 3, pp. 147–156, 2001.

Arunraj, N. S. and D. Ahrens, "A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting", *International Journal of Production Economics*, Vol. 170, pp. 321–335, 2015.

Aye, G. C., M. Balcilar, R. Gupta and A. Majumdar, "Forecasting aggregate retail sales: The case of South Africa", *International Journal of Production Economics*, Vol. 160, pp. 66–79, 2015.

Box, G. and G. Jenkins, *Time series analysis: forecasting and control*, Holden-Day series in time series analysis and digital processing, Holden-Day, San Francisco, 1976.

Breheny, P., *Classification and Regression Trees*, 1984, https://web.as.uky.edu/statistics/users/pbreheny/621/F10/notes/12-9.pdf, accessed in April 2018.

Breiman, L., J. Friedman, C. Stone and R. Olshen, *Classification and Regression Trees*, Taylor & Francis, 1984.

Cheng, C., A. Sa-Ngasoongsong, O. Beyca, T. Le, H. Yang, Z. Kong and S. T. Bukkapatnam, "Time series forecasting for nonlinear and non-stationary processes: a review and comparative study", *IIE Transactions*, Vol. 47, No. 10, pp. 1053–1071, 2015.

Chikobvu, D. and C. Sigauke, "Regression-SARIMA modelling of daily peak electricity demand in South Africa", *Journal of Energy in Southern Africa*, Vol. 23, No. 3, pp. 23–30, 2012.

Chintagunta, P. K., J.-P. Dubé and V. Singh, "Balancing Profitability and Customer Welfare in a Supermarket Chain", *Quantitative Marketing and Economics*, Vol. 1, No. 1, pp. 111–147, 2003.

Coghlan, A., *A Llittle Book of R for Time Series*, 2018, https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/, accessed in January 2019.

Cools, M., E. Moons and G. Wets, "Investigating the variability in daily traffic counts through use of ARIMAX and SARIMAX models", *Transportation Research Record*, Vol. 1, No. 2136, pp. 57–66, 2009.

Cornelsen, L. and C. Normand, "Impact of the smoking ban on the volume of bar sales in Ireland - Evidence from time series analysis", *Health economics*, Vol. 21, pp. 551–61, 2012.

Duncan, G. T., W. L. Gorr and J. Szczypula, *Forecasting Analogous Time Series*, pp. 195–213, Springer US, Boston, MA, 2001.

Fan, J. and Q. Yao, *Nonlinear Time Series : Nonparametric and Parametric Methods*, Springer, New York, 2008;2003.

Friedman, M., "A Comparison of Alternative Tests of Significance for the Problem of m Rankings", *The Annals of Mathematical Statistics*, Vol. 11, No. 1, pp. 86–92, 1940.

Friedman, J., R. Tibshirani and T. Hastie, "Regularized Paths for Generalized Linear Models Via Coordinate Descent", *Journal of Statistical Software*, Vol. 33, pp. 1–22, 2010.

Geurts, P., A. Irrthum and L. Wehenkel, "Supervised learning with decision tree-based methods in computational and systems biology", *Molecular BioSystems*, Vol. 5, No. 12, pp. 1593–1605, 2009.

Gur Ali, O., S. Sayın, T. van Woensel and J. Fransoo, "SKU demand forecasting in the presence of promotions", *Expert Systems With Applications*, Vol. 36, No. 10, pp. 12340–12348, 2009.

Gur Ali, O. and E. Pinar, "Multi-period-ahead forecasting with residual extrapolation and information sharing — Utilizing a multitude of retail series", *International Journal of Forecasting*, Vol. 32, No. 2, pp. 502–517, 2016.

Holt, C. C., "Forecasting seasonals and trends by exponentially weighted moving averages", *International Journal of Forecasting*, Vol. 20, No. 1, pp. 5–10, 2004.

Hoerl, A. E. and R. W. Kennard, "Ridge Regression: Biased Estimation for Nonorthogonal Problems", *Technometrics*, Vol. 12, No. 1, pp. 55–67, 1970.

Hyndman, R. J. and Y. Khandakar, "Automatic time series forecasting: The forecast package for R", *Journal of Statistical Software*, Vol. 27, No. 3, pp. 1–22, 2008.

James, G., D. Witten, T. Hastie and R. Tibshirani, *An introduction to statistical learning: With applications in R*, Vol. 103, Springer, New York, 2013.

Khashei, M. and M. Bijari, "Which Methodology is Better for Combining Linear and Nonlinear Models for Time Series Forecasting?", *Journal of Industrial and Systems Engineering*, Vol. 4, pp. 265–285, 2011.

Kongcharoen, C. and T. Kruangpradit, "Autoregressive Integrated Moving Average with Explanatory Variable (ARIMAX) Model for Thailand Export",*33rd International Symposium on Forecasting, South Korea*, 2013.

Mentzer, J. and C. Bienstock, *Sales Forecasting Management: Understanding the Techniques, Systems and Management of the Sales Forecasting Process*, SAGE Publications, 1998.

Montgomery, D. C., C. L. Jennings and M. Kulahci, *Introduction to Time Series Analysis and Forecasting*, John Wiley & Sons, Incorporated, Somerset, 2015.

Nemenyi, P., "Distribution-free multiple comparisons", *Biometrics*, Vol. 18, p. 263, *International Biometric Soc 1441 I ST, NW, SUITE 700*, WASHINGTON, DC 20005-2210, 1962.

Taskaya-Temizel, T. and M. C. Casey, "A comparative study of autoregressive neural network hybrids", *Neural Networks*, Vol. 18, No. 5, pp. 781–789, 2005.

Tibshirani, R., "Regression Shrinkage and Selection via the Lasso", *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 58, No. 1, pp. 267–288, 1996.

Zhang, G. P., "Time series forecasting using a hybrid ARIMA and neural network model", *Neurocomputing*, Vol. 50, pp. 159–175, 2003.

Zhang, G. P., "A neural network ensemble method with jittered training data for time series forecasting", *Information Sciences*, Vol. 177, No. 23, pp. 5329–5346, 2007.

Zou, H. and T. Hastie, "Regularization and Variable Selection via the Elastic Net", *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, Vol. 67, No. 2, pp. 301–320, 2005.