**A REPORT ON**

‘Feature Selection

&

Predictive Analysis

On a given Sales data along with 90 External Economic Indicators.’

**BY:**

SURABHI AGARWAL

GREAT LAKES INSTITUTE OF MANAGEMENT

**COMPANY:**



ROBERT BOSCH ENGINEERING AND BUSINESS SOLUTIONS PRIVATE LIMITED

A REPORT ON

‘Feature Selection

&

Predictive Analysis

On a given Sales data along with 90 External Economic Indicators.’

BY

SURABHI AGARWAL

DM19152

PGDM 2017-2019

GREAT LAKES INSTITUTE OF MANAGEMENT

Company Guide

Robert Bosch Engineering and Business Solutions Pvt. Ltd.

Mr. Balasubramanyam Pisupati

Principal Data Scientist (EDS)

Faculty Guide

Great Lakes Institute of Management, Chennai.

Dr. Anuradha M V

Professor

**DECLARATION**

I hereby declare that the Project Report ‘Feature Selection & Predictive Analysis on a given Sales data along with 90 External Economic Indicators’ is my own work to the best of my knowledge and belief. It contains no material previously published or written by another person or material which to substantial extent has been accepted for the award of any other degree, diploma or program of any other institute, except where due acknowledgement has been made in text.

Name: Surabhi Agarwal

Date:

Roll No.: DM19152

Great Lakes Institute of Management, Chennai

**CERTIFICATE**

This is to certify that Project Work entitled ‘Feature Selection & Predictive Analysis on a given Sales data along with 90 External Economic Indicators’, is a piece of work done by Surabhi Agarwal under my guidance and supervision for the partial fulfilment of Post Graduate Diploma in Management, a Programme offered by Great Lakes

To the best of my knowledge and belief the Project Report:

1. Embodies the work of the candidate himself / herself
2. Has duly been completed
3. Fulfills the requirements of the Rules & Regulations relating to the Summer Internship of the Institute.
4. Is up to the standard both in respect to contents and language for being referred to the examiner

Signature of the Faculty Guide

Date:

Dr. Anuradha MV

Professor, Great Lakes Institute of Management, Chennai.

**CONTENTS**

[**TABLE OF FIGURES** 6](#_Toc517866936)

[**ACKNOWLEDGEMENT** 7](#_Toc517866937)

[**COMPANY PROFILE** 8](#_Toc517866938)

[**MISSION STATEMENT** 10](#_Toc517866939)

[**About Bosch Culture** 11](#_Toc517866940)

[**About Data Science at Bosch** 12](#_Toc517866941)

[**Executive Summary** 14](#_Toc517866942)

[**About the Product, High Pressure Pump for Common-Rail systems** 16](#_Toc517866943)

[**PROBLEM STATEMENT FOR THE SUMMER INTERNSHIP PROJECT** 17](#_Toc517866944)

[**Methodology** 17](#_Toc517866945)

[STEP 1. Data Processing 17](#_Toc517866946)

[1.1 Create Lagged variables 17](#_Toc517866947)

[1.2 Omitting the NAs. 19](#_Toc517866948)

[1.3 Normalizing the Data. 20](#_Toc517866949)

[STEP 2. Checking for multi-collinearity. 21](#_Toc517866950)

[2.1 CORRPLOT in R 21](#_Toc517866951)

[2.2 CLUSTER ANALYSIS 23](#_Toc517866952)

[STEP 3. Feature Selection 25](#_Toc517866953)

[3.1 Random Forest Algorithm 25](#_Toc517866954)

[3.2 Boruta Algorithm 30](#_Toc517866955)

[3.3 LASSO Regression 32](#_Toc517866956)

[STEP 4. Model Formation & Prediction 34](#_Toc517866957)

[STEP 5. Function Creation 36](#_Toc517866958)

[STEP 6. ANALYSIS 38](#_Toc517866959)

[6.1 PARTIAL RESIDUAL PLOTS 39](#_Toc517866960)

[6.2 ADDED VARIABLE PLOTS 39](#_Toc517866961)

[**OVERALL LEARNING AND KEY TAKEAWAYS FROM INTERNSHIP** 41](#_Toc517866962)

# **TABLE OF FIGURES**

[Figure 1. Operational Domains 10](#_Toc517865122)

[Figure 2. Solutions Offered 10](#_Toc517865123)

[Figure 3. Mission Statement 11](#_Toc517865124)

[Figure 4. We are Bosch 12](#_Toc517865125)

[Figure 5. Pillars of Bosch Culture 12](#_Toc517865126)

[Figure 6. Major Focus Areas of Bosch 13](#_Toc517865127)

[Figure 7. Offerings Mode 14](#_Toc517865128)

[Figure 8. Common Rail Pump 17](#_Toc517865129)

[Figure 9. Code for Laged Variables 19](#_Toc517865130)

[Figure 10. Output for Laged Variables 20](#_Toc517865131)

[Figure 11. Omitting the NAs. 20](#_Toc517865132)

[Figure 12. Formula for Mean Normalization. 21](#_Toc517865133)

[Figure 13. Code for Normalization 21](#_Toc517865134)

[Figure 14. Output for Normalization. 22](#_Toc517865135)

[Figure 15. Code for CorrPlot 23](#_Toc517865136)

[Figure 16. Output for CorrPlot 23](#_Toc517865137)

[Figure 17. Code for Hierarchical Cluster Analysis. 24](#_Toc517865138)

[Figure 18. Dendrogram output. 25](#_Toc517865139)

[Figure 19. Formula 26](#_Toc517865140)

[Figure 20. Nodes and Leaf Diagram 27](#_Toc517865141)

[Figure 21. Random Forest Algorithm 28](#_Toc517865142)

[Figure 22. Code for Random Forest 29](#_Toc517865143)

[Figure 23. Output for Random Forest 29](#_Toc517865144)

[Figure 24. Variable Importance Plot 30](#_Toc517865145)

[Figure 25. Boruta Code 32](#_Toc517865146)

[Figure 26. Output for Boruta Algorithm 32](#_Toc517865147)

[Figure 27. Variable sorting through Boruta 33](#_Toc517865148)

[Figure 28. LASSO formula 33](#_Toc517865149)

[Figure 29. Code for LASSO Regression 34](#_Toc517865150)

[Figure 30. Output for LASSO Regression 35](#_Toc517865151)

[Figure 31. MAPE formula 36](#_Toc517865152)

[Figure 32. Code for Model Formation 36](#_Toc517865153)

[Figure 33. Output for Model summary 37](#_Toc517865154)

[Figure 34. Predicted Sales 37](#_Toc517865155)

[Figure 35. Code for reading Excel entries 38](#_Toc517865156)

[Figure 36. Code for dividing data into Train and Test 38](#_Toc517865157)

[Figure 37. Deleting the unknown indicators 38](#_Toc517865158)

[Figure 38. Code for extracting the top ten indicators 39](#_Toc517865159)

[Figure 39. Code for Added Variable Plots. 41](#_Toc517865160)

[Figure 40. Added Variable Plots 41](#_Toc517865161)

# **ACKNOWLEDGEMENT**

I take this opportunity to extend my sincere thanks to Robert Bosch Engineering and Business Solutions Pvt. Ltd. for offering a unique platform to earn exposure and gain knowledge in the field of Analytics.

I would like to extend my sincere and heartfelt gratitude to my mentor Mr. BalaSubramanyam Pisupati, Principal Data Scientist and Mr. Prince Rajan, Senior Architect for welcoming me to the department and guiding me throughout my three months of internship.

To Ms. Shanu Agrawal, for giving me the opportunity of working on the project and for guiding, supporting and encouraging me during the entire tenure of the project. I’d like to thank my fellow colleagues and interns, for helping me get past all the roadblocks I faced during my internship.

I would also like to thank Dr. Sheela Siddappa and Mr. Bharat Kakaiah for giving me the opportunity to work in RBEI. I am able to say with conviction that I have benefited extremely from the prestigious association as a summer intern with Robert Bosch Engineering and Business Solutions Pvt. Ltd.

I would also like to thank Dr. Anuradha MV, my faculty guide, who was a constant support throughout the three months and who motivated me to give my best to the internship during our very first meet.

Lastly, I would like to thank Dr. Suresh Srinivasan and Ms. Kiruba Muthuraj for their guidance and motivation, and for creating a curriculum that has helped me in successfully completing my internship.

# **COMPANY PROFILE**

Robert Bosch Gmbh was founded in 1886 by Robert Bosch in Stuttgart. Bosch’s core businesses are divided into four business sectors, namely, mobility solutions, industrial technology (drive and control), consumer goods (power tools and household electrical appliances) and energy & technology building.

Robert Bosch, first entered India in 1922 and set up its Sales office in Calcutta. Bosch India grew to have a turnover of over $2 Billion and employees over 26000 employees across several locations and application develop centers. It is a publically traded company and has a market capitalization of over $12 billion.

About 84% of Bosch India’s revenues come from its automotive business and that is the domain in which I carried out my Summer Internship Project, Predictive Analysis and Sales Forecasting for an automotive part sold in the European Markets. Also, the remaining portion of its revenues is derived from energy and technology building solutions, power tools and consumer retail.

Robert Bosch Engineering and Business Solutions Private Limited is a 100% owned subsidiary of Robert Bosch GmbH. The company offers end-to-end Engineering, IT and Business Solutions. It is considered the largest software development center of Bosch, outside Germany. It stands as a Technological Powerhouse of Bosch in India. The company has a global footprint and presence in Us, Europe and Asia Pacific region.

RBEI was established in 1981, over the years it has established development centers at Mexico, Bangalore, Coimbatore and Vietnam. 20,000 employees are associated with RBEI under the leadership of its Managing Director and President Mr. Vijay Ratnaparkhe. RBEI alone as filed around a 1000 patents. Its primary domains of operations are: Manufacturing, Automotive, Healthcare and Hi-Tech.



Figure 1. Operational Domains

The domain of operations help contribute in offering a wide range of solutions broadly categorized as below:



Figure 2. Solutions Offered

RBEI has a Data Analytics division that offers end-to-end analytics solutions. It stands for Innovation and Incubation and contributes towards providing manufacturing, Engineering and Enterprise Solutions, and Business Services.

I interned as a Data Scientist at Robert Bosch Engineering and Business Solutions Private Limited (RBEI), Bangalore. The Data Science department is branched into two division, namely, EDS1 and EDS2. EDS stands for Engineering Data Science. These departments are branched on the basis of the types of clients they serve, the Bosch World or outside clients. EDS 1, the department I interned at, dealt with clients from the Bosch World, under the guidance and mentorship of Balasubramanyam Pisupati who leads the department. The department employees 18 associates, specializing in different tools and provide specialized solutions in the field of Machine Learning and Data Science.

# **MISSION STATEMENT**

Bosch’s mission statement consists of five critical questions.

Figure 3. Mission Statement

# **About Bosch Culture**

*“I would rather lose money than Trust” –Robert BOSCH*

This belief is imbedded in the company’s core values and culture. 

Figure 4. We are Bosch

The pillars on which Bosch’s culture stands:



Figure 5. Pillars of Bosch Culture

1. Future and result focus: The Company is extremely result-oriented. This also forms the basis for all its social initiatives and its foundation.
2. Responsibility and Sustainability: All the employees act responsibly in the interest of the company, especially taking into consideration the social and ecological impact of their actions.
3. Initiative and Determination: Employees are encouraged to take initiatives, entrepreneurial responsibility and achieve their goals with determination.
4. Openness and Trust: Open discussions and problem solving sessions are held on a regular basis.
5. Fairness: Fairness is viewed as a cornerstone of the company’s corporate success.
6. Reliability, Credibility and Legality: The main motto, accept only what can be delivered, agreements are seen as binding and, respect and observe the law in all the business transactions.
7. Diversity: Diversity is seen as an essential element for success.

# **About Data Science at Bosch**

With the increasing adoption of Internet of Things (IoT), there is a constant stream of data that is being generated exponentially across various business processes. They say Data is the new Intel and the ability to take data, understand it, process it, extract value from it, visualize it and finally, communicate it is what constitutes Data Science.

It is not just about the existence of data or making guesses about what that data might mean, it is also about testing hypothesis and making sure that the conclusions you are drawing from the data are valid.

At Bosch, they have four major Focus Areas:



Figure 6. Major Focus Areas of Bosch

They operate via three mode of offerings, namely, Consulting, Solutions and Services.

Figure 7. Offerings Mode

# **Executive Summary**

During my 3 months of internship, I worked as a Management Trainee in the Data Science department at Robert Bosch Engineering and Business Solutions Private Limited. I carried out my work extensively on R. All the analysis conducted by me, was done so with the use of packages and functions readily found in R.

During my time there, I worked on a single project. I worked on the Sales dataset of a common rail pump, manufactured and sold by Bosch as an automotive part in the European market. The Sales of this component was believed to have been affected by 90 External European Economic indicators, like German stocks, UK IP value, Spain’s quarterly GDP and so on. I was given the monthly sales value and the values of all the 90 economic indicators from a period of Jan 2006 to Oct 2016.

My main task was to conduct Feature Engineering on the Indicators, Predictive Analysis and predict the Sales value and finally, forecast the Sales for the future periods. Due to time constraints I could not conduct Sales Forecasting.

Before starting with Feature Engineering, some amount of data processing and manipulation was required. The first step was to create lagged variables for Sales and all the 90 external independent variables. This was done to investigate whether the variables (the dependent and the independent variables) in the previous periods were more important in understanding the outcome in the current period.

Next step was to normalize the data in order to clearly understand the output obtained from the feature engineering process.

Predictive analysis involved carrying out Cluster Analysis, Principal Component Analysis, finding the Correlation amongst the 90 indicators, creating a linear and non-linear model by running Regression using different algorithms, plotting Scatter Plots and studying and verifying the existence of the relationship and finally, predicting the Sales value and finding the Mean Absolute Percentage Error (MAPE) for the predictions.

For Feature Selection and Model formation (Regression), I used Boruta Algorithm, LASSO Regression (L1 Regularization) and RandomForest Algorithm.

Lastly, I created a function in R which took inputs from Excel of the available data, divided the data into test and train, and after data cleansing (Lagged Variables and Normalizing), ran Lasso Regression for Feature Selection. It would then extract the top 10 important indicators, form a model and predict the sales value. This it did individually for all the rows in the test data. This function is useful in real life where not all External Economic Indicators’ values would be known all the time. So the Indicator whose value is not known for a particular month, is removed from the data set before feature selection and prediction.

# **About the Product, High Pressure Pump for Common-Rail systems**

These are used for High motor performance with low consumption.

The Common-Rail pump provides better fuel efficiency and reduced emissions. It achieves this by atomizing the fuel by injecting at a high pressure ranging from 1800 to 200 bar. This system is used for passenger vehicles and light and medium duty vehicles.

The high-pressure pump compresses the fuel and supplies it only in the quantity that is needed. It maintains system pressure by continuously feeding fuel to the highly pressurized rail. Pressure generation is not linked to the engine speed and so the required pressure is available even at low speeds.

The common-rail pump has been very successful on the market and over 40 million units have been sold globally.

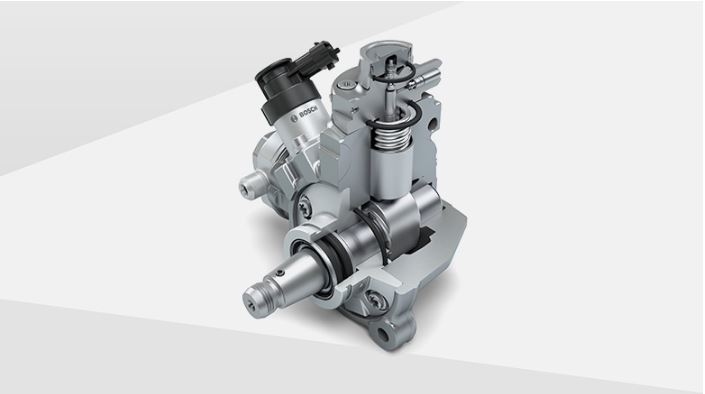
****

Figure 8. Common Rail Pump

**What do we aim to achieve?**

The main aim of undertaking this project, the business idea, behind it is to improve the predictive ability of the Model.

We have 90 external economic indicators but not all are improving the predictive accuracy. Some indicators act as noise. So through this project, I aimed at finding out the top indicators which contribute majorly to the product’s Sales and then tried predicting Sales. So in the future we can accurately predict Sales and avoid extra costs by studying the economic environment of that particular period.

# **PROBLEM STATEMENT FOR THE SUMMER INTERNSHIP PROJECT**

Given, 90 economic indicators affecting the sales of the **automotive part sold** by Bosch in the European market. The Data is from Jan 2006 to Oct 2016. Process the data, conduct predictive analysis on it and then finally, carry out Sales Forecasting for the next 12 months.

# **Methodology**

## STEP 1. Data Processing

Data in its raw form is not useful. Hence it is imperative that we convert the collected data into a valuable and desirable form. In simple terms, it is the manipulation of data to produce meaningful information.

The following are the processing steps I followed to convert the data:

* 1. Create Lagged variables for the dependent variable, Sales and the 90 external economic indicators which are believed to have an effect on Sales.

A lagged variable is a variable whose value comes from an earlier point in time. Also known as a delayed variable, they are created to investigate whether the variables in the previous periods are important in understanding the outcome in the current period.

Consider a sequence of n observations with X= x0, x1, x2,….., xn taken at T= 0,1,2,…., n. We can build a lagged variable, Y by taking the values of X from time, T =1,2,3…..n.

Another variable, Z can be taken from time, T= 2,3,….n and is said to have a lag of 2.

For my project I created a lag of 12 periods, denoting the number of months in a year, for each external economic indicator and also for the dependent target variable, Sales.

Lagged variables are created because of the fact that the value of an indicator in a certain moment of time, will have influence on its future values. For example, a high Sales value at time T, will cause a higher Sales in time T+1.

Rarely will the dependence of variable Y on another independent variable X, be instantaneous. Y often corresponds to different values if X with a time lapse, called a lag.

In our case, for example, the value of Germany Stocks has an influence over Sales. But Sales in the current period not only depends on value of Germany Stocks from the current period but also, on the value of Germany Stocks from the previous periods.

**CODE in R**

To create a lag of 12 periods in all the 90 variables I used the SHIFT function along with FOR loop and CBIND functions.

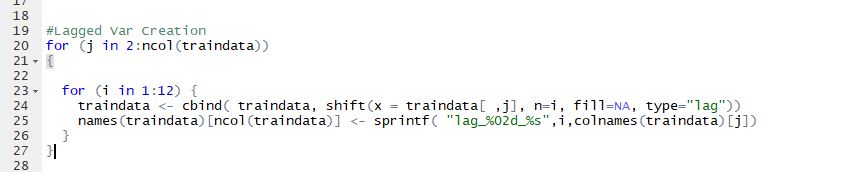


Figure 9. Code for Laged Variables

**CODE EXPLANATION**

This function takes an input vector X, which in our case is the different columns of the dataset consisting of the 90 external indicators and Sales. Along with that it takes a specified integer and shifts the input vector by that amount.

I wanted to create a lag of 1 to 12 periods for each of the variables and so have used two For Loops. The first FOR loop runs for the input variable selection and the second for the amount of shift needed.

NAs are added to pad the shifted vector so that it is of the same length as its input.

The *cbind* function is used to recombine all the columns into a single dataset again.

**OUTPUT**

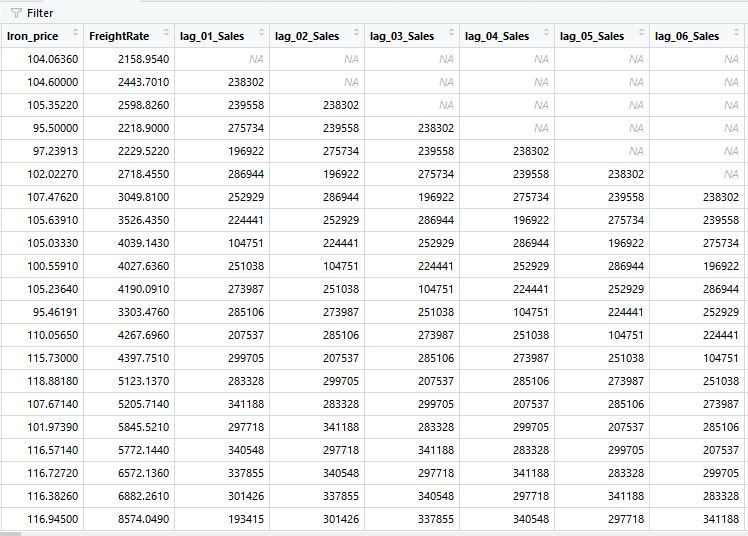


Figure 10. Output for Laged Variables

### 1.2 Omitting the NAs.

The rows containing the NAs will have to be deleted in order to run a successful feature selection and avoid any errors.

**Code in R**



Figure 11. Omitting the NAs.

**CODE EXPLANATION**

NA.OMIT function is from the *package data.table.* It takes as input the column, row or data table from which the NAs have to be deleted.

As an output it deletes all the rows which have NAs in them and then returns the same.

In my project, the first 12 rows were deleted due to the creation of lagged variables for a period of 12. So the data for the year 2006 was deleted from my dataset.

### 1.3 Normalizing the Data.

Normalization makes the data less sensitive to the scale of the indicators. In the given dataset, one feature has values ranging between 1 to 100, and another feature whose value ranges from 1 to 1000. In such cases, due to the greater numeric range of one feature, its impact on our target variable will be greater than the other features having a relatively lower numeric range, despite it being irrelevant to our target variable. Hence, normalizing the data before feature selection is necessary.

I have used mean normalization where I made the data mean zero and the variance as one.

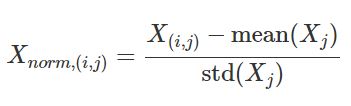


Figure 12. Formula for Mean Normalization.

Where, X is the feature. i represents rows of observations and j represents the columns containing the different features.

**Code in R**

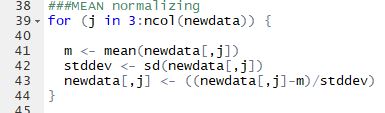
****

Figure 13. Code for Normalization

**CODE EXPLANATION**

For normalizing the data, I have generated a *for* loop, which will run separately for individual feature and mean normalize the observations for all the features, giving a scaled output.

**OUTPUT**

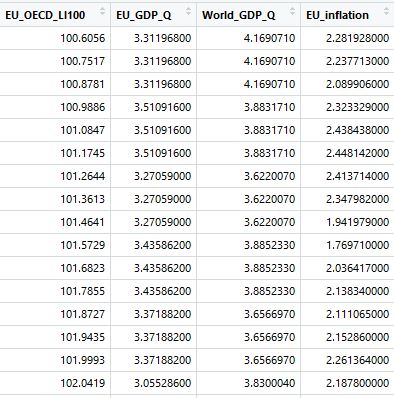
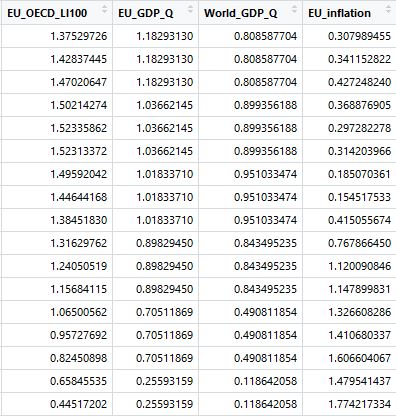


Figure 14. Output for Normalization.

## STEP 2. Checking for multi-collinearity.

Multi- collinearity occurs when the independent variables in a multiple regression model are correlated to one another. That is the change in one variable not only affects our target variable, but also brings about a change in another variable.

Methods to test the presence of multi-collinearity:

* Cluster Analysis
* Corrplot in R

### 2.1 CORRPLOT in R

It graphically displays the correlation matrix and gives us the correlation coefficient for each variable.

**Code in R**

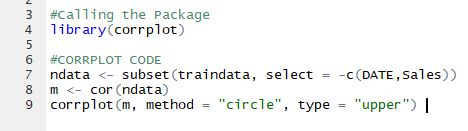
****

Figure 15. Code for CorrPlot

**CODE EXPLANATION**

* *Corrplot* function is part of the corrplot package.
* Firstly, defined the data frame for which the correlation has to be found.
* Secondly, used the function *cor* to find the correlation coefficient.
* Lastly, used *corrplot* function to plot the correlation.

**OUTPUT**

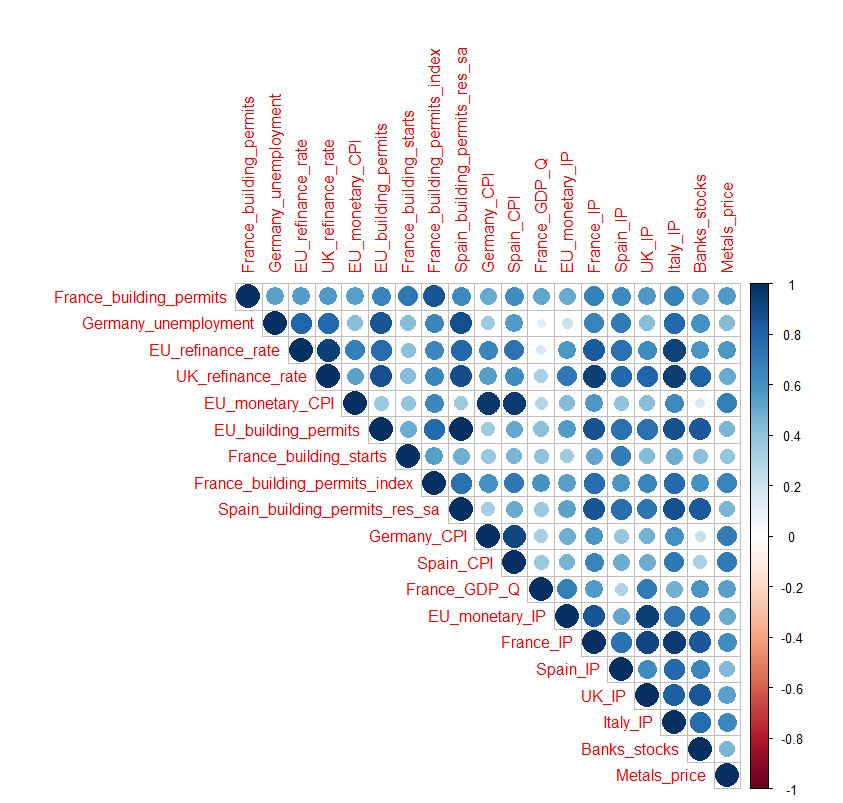
****

Figure 16. Output for CorrPlot

This corrplot is for the variable France\_building\_permits and shows that it is highly correlated to France\_building\_permits\_index.

This helps us in analyzing the result when one of the variable value is unknown and the model replaces that variable with another variable to which it is highly correlated to.

### 2.2 CLUSTER ANALYSIS

It is the grouping of similar variables in such a way that they form a part of the same group, called cluster. The members of a cluster are more similar to each other than they are to members belonging to another cluster.

There are several methods to perform cluster analysis, and I chose Hierarchical Clustering for my purpose.

Hierarchical Clustering seeks to build a hierarchy of clusters, giving us tree-like structures, called dendrograms. There are two ways to do this, the

* Bottoms Up Approach: This treats each feature as a single cluster at the start and merges two clusters at a time on the basis of some similarity. This continues until all the items merge into one single cluster.
* Top Down Approach: The opposite to the bottoms up approach. It starts as a single cluster and keeps splitting into distinct parts, based on the degree of similarity.

R uses Bottom-Up Hierarchical Clustering.

**Code in R**

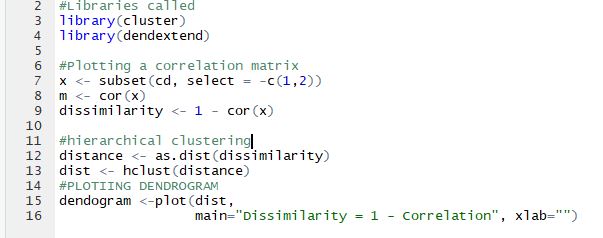
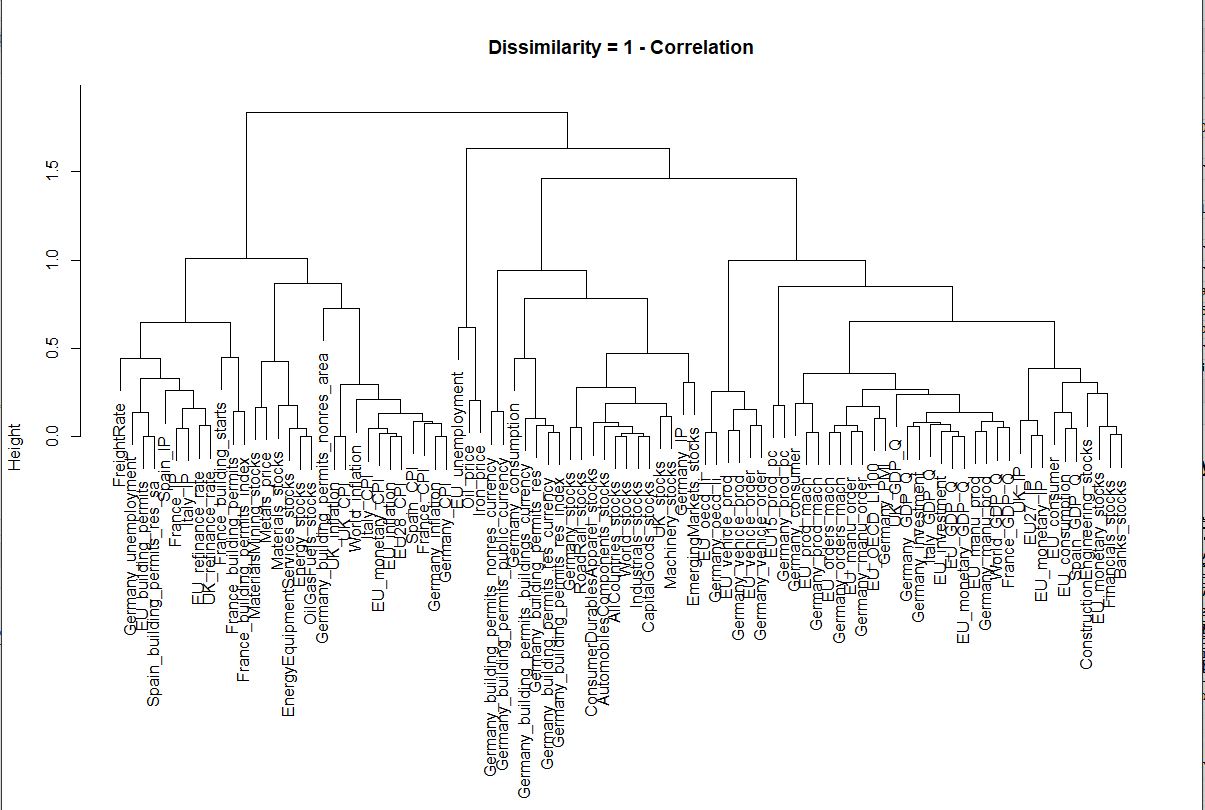
****

Figure 17. Code for Hierarchical Cluster Analysis.

**CODE EXPLANATION**

* The packages *cluster* and *dendextend* are called for hierarchical clustering.
* The function used in R for clustering is *hclust*. This function takes inputs in the form of the distance matrix.
* To determine how close two clusters are, we can use various methods but the default is complete linkage clustering, that is, finding the maximum possible distance between points from two separate clusters. Hence, we calculate the dissimilarity matrix and feed it into R.
* Finally, we use the function *plot* function to plot our dendrogram.

**OUTPUT**

****

**CLADES**

Figure 18. Dendrogram output.

Each node in the cluster tree contains a group of similar features. The total number of clusters produced is not predetermined.

The clusters similar to each other have clades of similar heights. The greater the difference in the heights of the clades of two clusters, the more dissimilar these clusters will be.

For example, EU27\_IP and EU\_monetary\_IP are most similar as they belong to the same cluster. Between these two will also be similar to UK\_IP since their clades are almost of the same height.

## STEP 3. Feature Selection

The terms “features”, “variables” and “attributes” all mean the same thing and can be used interchangeably.

Feature Selection is carried out to build accurate predictive models, free from correlated variables, biases and unwanted noise.

Often, data collected is described with large number of variables, most of which are irrelevant to classification. Dealing with too many variables is time consuming and uses too many resources. Also, dealing with large number of variables significantly decreases the model accuracy. To build a robust predictive model, it is not necessary to include all the variables, only an optimal number of important variables can be used to give us the desirable result. Why? Because too many variables result in a model that over fits the training data and hence is not able to generalize to the test data. Our goal is to get a model with a low variance.

To carry out Feature Selection, I implemented three different Algorithms, namely,

* Boruta
* Lasso Regression
* Random Forest

I will explain all three of them along with which method worked best for me and why.

### 3.1 Random Forest Algorithm

Random Forest is an algorithm that follows supervised learning. **Supervised learning** is when we have a dependent variable (Sales, Y) and independent variables (External Economic indicators, X) and we use an algorithm to map a function from the independent input variables to the dependent output variable.



Figure 19. Formula

It is called a supervised learning because of the process of the algorithm learning from the training data and approximating the mapping function accurately such that we can predict the output variable (Y) for a new dataset (Test).

Random Forest in simple terms maybe called as a collection of decision trees.

The **Decision Tree** Concept is a rule based system. It takes the training dataset with the Target variable and Features as input and then comes up with a set of rules. When you pass the training data through the decision tree classifier, it will start building the rules and create parent nodes (Nodes) and Leaf Nodes (Child Nodes). By considering the path from the parent to the child node, we can get the rules.

The same set of rules can be used to perform the predictions on the test data.

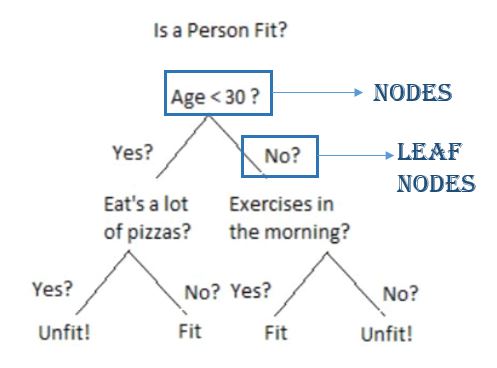


Figure 20. Nodes and Leaf Diagram

In Random Forest algorithm the process of finding the root nodes and splitting leaf nodes happens automatically. The following are the steps in brief:

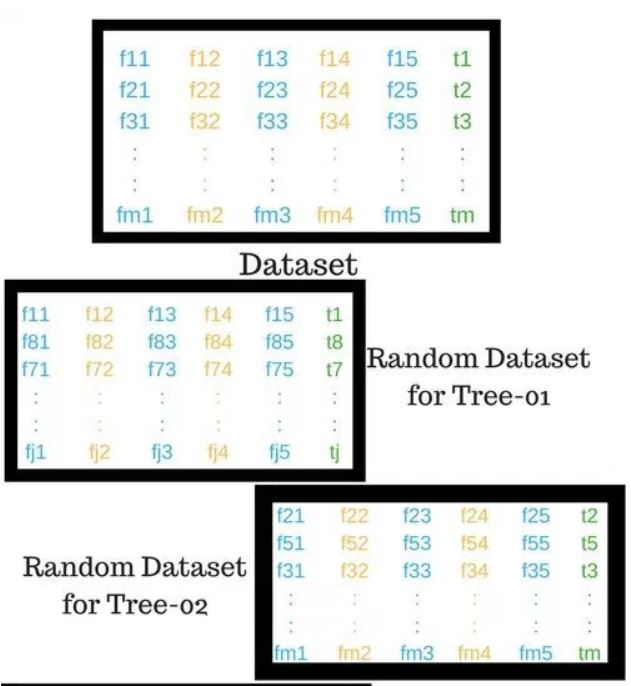


Figure 21. Random Forest Algorithm

* Randomly selecting K features out of a total of M features, where **K <<< M.** (Random selection of features and observations)
* Use the randomly selected K features to find the Root Node by using the best split approach.
* We calculate the daughter nodes, again using the same best split approach.
* We will repeat the first three steps till we get a tree with a root node and the target variable as the leaf node.
* Finally, we repeat all the previous four steps until we have n randomly created decision trees which together form the random forest.

The random forest algorithm brings extra randomness into the model and instead of searching for the best feature while splitting a node, it searches for the best feature among the randomly selected subset of features. Hence, when you are creating a tree, only a random subset of features is considered while splitting a node.

Advantages

* Does not over fit the data due to the large number of trees it generates.
* Does not require normalization of the data.

Disadvantages

* Random Forest is fast to train but slow to perform real-time predictions on test. The large number of trees make the algorithm slow and ineffective, resulting in a slower model.

**Code in R**

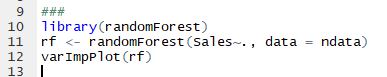


Figure 22. Code for Random Forest

**CODE EXPLANATION**

* In the first step we call the package randomForest.
* Next, we tell the function randomForest that the target variable, Sales is dependent on the variables, excluding the date column and that it should consider data from the data table named ndata.
* Lastly, we call upon the function varImpPlot, which gives you the importance of each variable selected by the algorithm.

**OUTPUT**

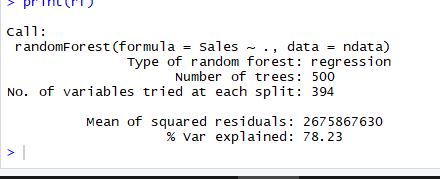


Figure 23. Output for Random Forest

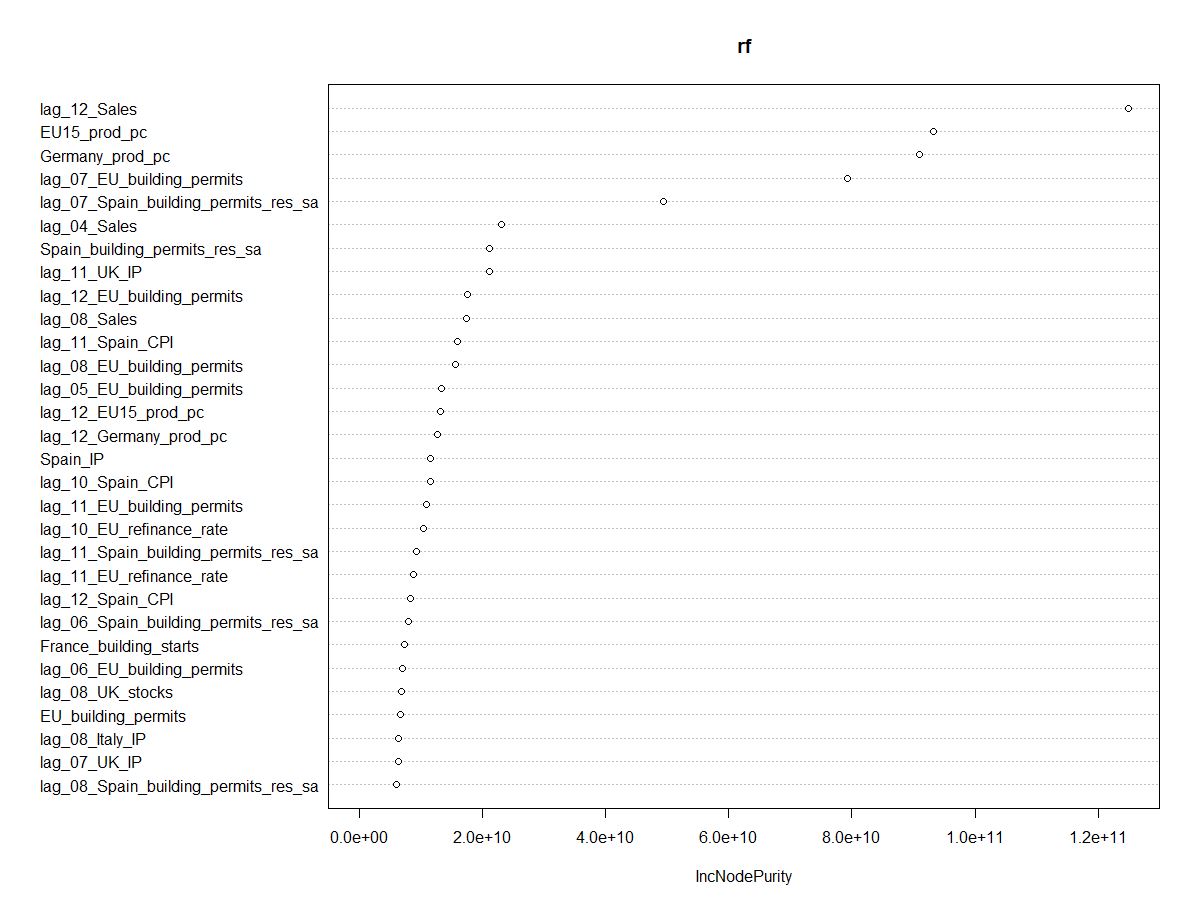


Figure 24. Variable Importance Plot

This graph represents the features, the random forest algorithm deems important in the decreasing order of importance. If we drop the topmost variable, lag\_12\_Sales from our model, the accuracy of the model will decrease highly.

However, if we drop the lowermost variable, lag\_08\_Spain\_building\_permits\_res\_sa from our model, the accuracy would not be affected much.

The IncNodePurity refers to the loss function, gini-impurity. Gini impurity measures how frequently a randomly chosen element would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset. In other words, determines how poorly our model performed.

### 3.2 Boruta Algorithm

This Algorithm works as a wrapper algorithm around Random Forest and is useful when a data set comprises of large number of variables.

The Algorithm is based on an **“all-relevant selection method”,** that is, it considers all features that are relevant to the dependent variable.

What does a wrapper algorithm do?

* Boruta algorithm starts by creating shadow features, that is, it duplicates each variable and instead of making a row to row copy, it permutes the order of the values of each variable. This is done so that there is no relationship between the target variable and the shadow features.
* It then creates all possible subsets from the set of features. These **“all possible subsets”** are randomly chosen in Boruta algorithm.
* Boruta uses a classification algorithm, Random Forest, to predict the target variable based on the original features and the shadow features.
* It then compares the variable importance scores for every variable, the original variable is compared to its shadow variable. And selects the feature whose importance is significantly greater than the importance of its shadow features.
* Finally, gives the subset of features with which the random forest algorithm performs the best.

Advantages

* Since it uses Random Forest as a classifier, there is no need for data pruning or normalization of data.
* Can be used with a large data set of variables.

Disadvantages

* Boruta Algorithm does not give an importance score for the important features which makes it impossible to extract the top ten features.

**CODE in R**

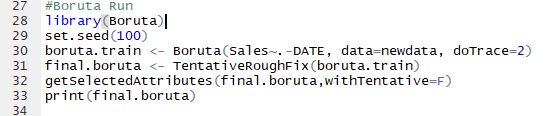


Figure 25. Boruta Code

**CODE EXPLANATION**

* The first step is to install and call the package Boruta.
* *set.seed()* is used to create simulations that can be reproduced.
* In the third step, we created a function that shows that the target variable, Sales is dependent on the variables, excluding the date column. Next we order Boruta to take data from the dataset named, newdata and finally *doTrace* refers to the verbosity level.
* *TentativeRoughFix()*: compares the Z scores of each of the variable to their respective shadow features and decides on whether to reject or confirm a certain attribute.
* *getSelectedAttribute():* saves a final list of important variables in a vector.

**OUTPUT**

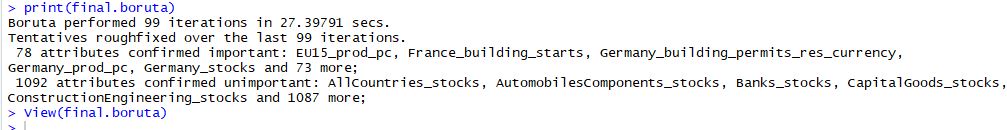


Figure 26. Output for Boruta Algorithm

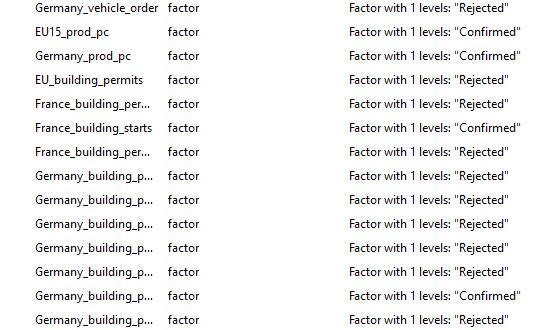


Figure 27. Variable sorting through Boruta

Boruta gives a clear picture on the significance of the features. In this case, out of 1172 variables, 78 attributes were confirmed.

### 3.3 LASSO Regression

**Lasso is short for Least Absolute Shrinkage and Selection Operator.**

Lasso Regression performs variable selection and regularization to improve the model’s prediction accuracy. It shrinks the data values towards a central point, mean. Lasso Regression is best suited for a dataset having highly correlated variables.

It performs L1 Regularization for feature selection, that is, it adds a penalty equivalent to the absolute value of the magnitude of the coefficient. In this way, few coefficients are reduced to zero and are eliminated from the model.

Goal of the algorithm: To minimize

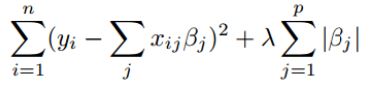


Figure 28. LASSO formula

Where I represents the number of observations and j represents the total number of features present in the dataset.

β represents the penalty added to the indicator’s coefficient.

λ is the tuning parameter, controlling the strength of the penalty. As λ increases, more and more indicators are eliminated from the model along with a decrease in the variance.

**CODE in R**

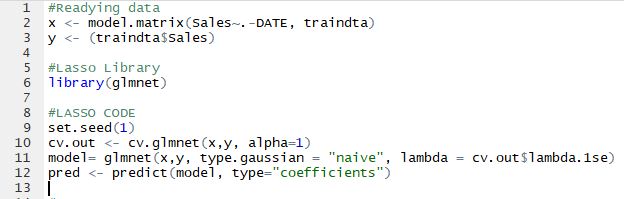
****

Figure 29. Code for LASSO Regression

**CODE EXPLANATION**

* The first step is to input the indicators as a matrix as Lasso does not have a formula interface.
* Next is the call to the package *glmnet.*
* Thirdly, we use the functions *cv.glmnet and glmnet* which will give us an optimal value for lambda. We set the alpha value to 1 to show that it is a lasso regression.
* Finally, we use the *predict* function to predict the coefficients of the features.

**OUTPUT**

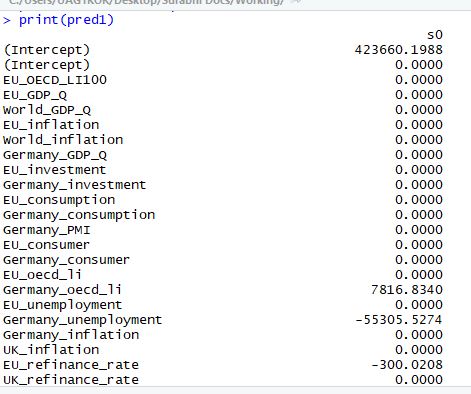


Figure 30. Output for LASSO Regression

In this way the variables with the largest coefficients can be said to e the most important variables directly affecting our target variable, Sales.

Out of the above mentioned three methods for feature selection, I chose LASSO Regression.

* It is a model building and variable selection algorithm that can be applied to many types of regressions.
* It addresses the multicollinearity amongst the indicators.
* It provides us with variable coeeficients and reduces the unimportant variables to zero.

## STEP 4. Model Formation & Prediction

After Feature Selection, we extract the top ten features for model formation and prediction and calculate the Mean Absolute Percentage Error (MAPE) for the predictions.

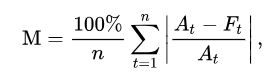


Figure 31. MAPE formula

Here *At* is the actual value of the Sales and *Ft* represents the predicted Sales value.

The MAPE represents the accuracy of the predicting model.

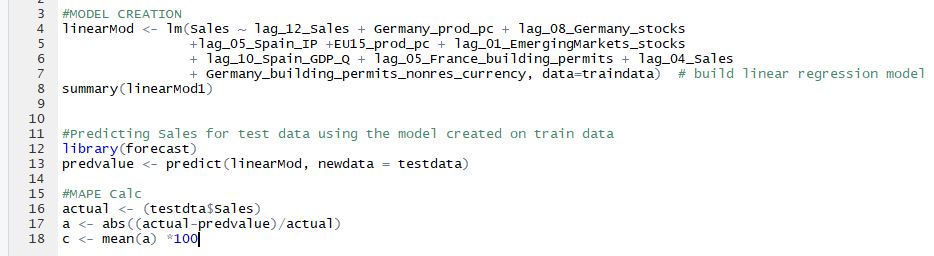
**CODE in R**

Figure 32. Code for Model Formation

**CODE EXPLANATION**

* The *lm* function is used to run a linear regression on the train data.
* The *summary* function gives us a summary of the fit of the model.
* *Forecast* package is called to make predictions. It has a function *predict* which uses the model created on the train to make predictions on the test data.
* Finally, we calculate the MAPE for the test data which comes to be 2.13%

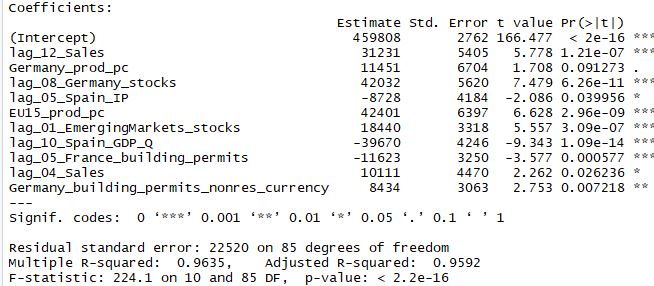
**OUTPUT**

Figure 33. Output for Model summary

This gives us an Adjusted R2 of 95.92% making it a very good model. And the indicators shown in the screenshot are the top ten indicators of our model.

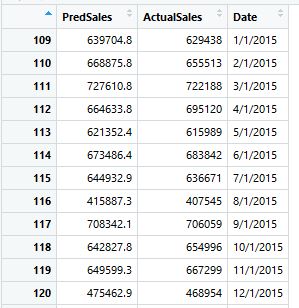


Figure 34. Predicted Sales

## STEP 5. Function Creation

1. The first step in the function creation is reading the input from an Excel sheet into R.



Figure 35. Code for reading Excel entries

The package *xlsx* contains the function *read.xlsx*, which reads data from excel into R. The first input is the Excel address and filename, the second is the specification of the sheet number from which the data has to be imported. The *header* is set to *TRUE* as we want R to read the first row as the column names.

1. Lagged variable creation.
2. Next, we divide the data into test and train. The test data will also contain the latest entries made by the user to the Excel dataset.



Figure 36. Code for dividing data into Train and Test

1. Now we check for the NA values in the test data. In real life situations, often, we wouldn’t know the values of all the economic indicators and so, we should not consider those indicators. Once we check the missing values, we will delete those features from our train data.

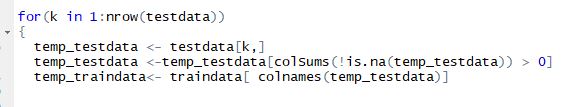


Figure 37. Deleting the unknown indicators

Here, the *is.na* function checks for missing values in the test data and deletes those columns from the dataset. It then compares all the columns of the test and train datasets and deletes all those columns from the train dataset which do not exist in the test data.

Also, we have created a *for* loop which runs for individual rows of the test data. This is done so that when a latest entry is made (for the recent month), we can run regression for that particular month and make a prediction for a single month.

1. Next, we delete the NAs formed due to lagged variable creations and normalize the train data.
2. We run the LASSO regression and get the coefficients of all the features for further model formation.

In this step, we use the function *abs* to get the absolute values of the coefficients and then the function *top* to arrange the indicators in the decreasing order of their coefficient’s magnitude.

We then select the first ten indicators and store them in a data table called *top*.

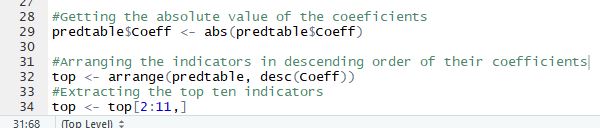


Figure 38. Code for extracting the top ten indicators

1. We now form a function using the indicators in the table top and predict the sales value for the latest entry.

This function will keep changing the top ten indicators based on the feature values available for that month and hence, accurately predict Sales.

## STEP 6. ANALYSIS

The top 10 indicators when the values for all the features are known are:

* **lag\_12\_Sales**
* **Germany\_prod\_pc**
* **lag\_08\_Germany\_stocks**
* **lag\_05\_Spain\_IP**
* **EU15\_prod\_pc**
* **lag\_01\_EmergingMarkets\_stocks**
* **lag\_10\_Spain\_GDP\_Q**
* **lag\_05\_France\_building\_permits**
* **lag\_04\_Sales**
* **Germany\_building\_permits\_nonres\_currency**

The first variable, lag\_12\_Sales, that is the Sales from the last month, is the most important variable and largely affects the current Sales value. When we remove it from the model, the MAPE increases from 2.13 to 5.44%.

### 6.1 PARTIAL RESIDUAL PLOTS

When we perform a linear regression with a single independent variable, a scatter plot of response variable against the independent variable provides a good reflection of the nature of their relationship.

But when a model has more than one independent variable affecting out response variable, scatter plots become inefficient because they do not take into account the effect of the independent variables on each other. So, partial residual plots are plotted to identify the nature of the relationship between the target variable and a single independent variable, given the effect of other variables in the model.

### 6.2 ADDED VARIABLE PLOTS

These are also known as Partial Regression Plots, Adjusted Variable Plots and Individual Coefficients Plots.

These plots are based on **Frisch-Waugh Theorem** which states that the multiple regression coefficient of any single variable can also be obtained by netting out the effect of other variables in the regression model from both the dependent and independent variables.

When we perform a linear regression with a single independent variable, a scatter plot of response variable against the independent variable provides a good reflection of the nature of their relationship.

But when a model has more than one independent variable affecting out response variable, scatter plots become inefficient because they do not take into account the effect of the independent variables on each other. So, partial regression plots are plotted to identify the nature of the relationship between the target variable and a single independent variable, given the effect of other variables in the model.

Steps to plot an added variable plot:

1. Perform a regression of the target variable, Y, against all the independent variables except one, say X. Compute the residuals for this model.
2. Perform a regression of the independent variable, X against the other independent variables. Compute the residuals for this model.
3. Plot the residuals obtained from step 1 against the residuals obtained from step 2.

Residuals are the difference between the Observed values and the Predicted values.

The vertical distances between the data points and the regression line are measured. These explain the amount of variability in Y that is left over after accounting for the variability explained by the other indicators. This leftover variability is due to the independent variable, X.

**CODE in R**

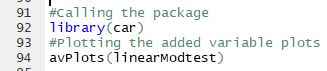


Figure 39. Code for Added Variable Plots.

**CODE EXPLAINED**

We call the package *car* which houses the function *avPlots* used for plotting an Added Variable Plot.

**OUTPUT**

The regression line has a positive slope, implying a positive relationship between the target and the independent variable.

The slope of the line is the coefficient of lag\_12\_Sales in the full regression.

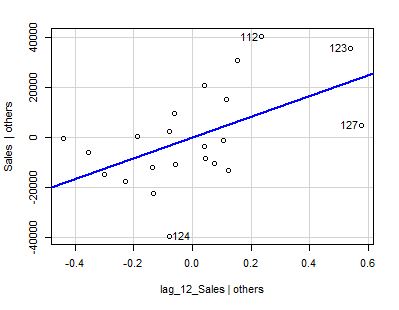


Figure 40. Added Variable Plots

# **OVERALL LEARNING AND KEY TAKEAWAYS FROM INTERNSHIP**

The internship was a great learning experience for me in the field of Data Science. Having no prior knowledge and experience in the field, I learnt the fundamentals of how a big corporate such as Bosch handles their clients, acquires new projects and successfully completes them on time. The project I was given, provided me with an insight as to how a seemingly simple process, Sales Forecasting, requires immense domain knowledge and that there are several steps to be followed and completed, before one can start with Forecasting. This project gave me an experience of how intricate data can be. I learnt that every data point tells a story and how we interpret it makes all the difference.

In addition to learning about the data science processes, I worked extensively on R and have now gained experience in navigating my way through any roadblocks I might encounter in the future.

Internship at Bosch has also taught me the importance of brainstorming and seeking help. We had weekly update meetings where the department shared with each other, their projects, the progress they had made so far and roadblocks that they faced and openly asked for help. I was pleasantly surprised at Bosch’s open culture where employees can openly admit their limitations, ask and accept help from others. This, I feel forms the foundation of ‘the willingness to learn’, something I found imbued in all my colleagues.

Lastly, from my department mentor, Mr Balasubramanyam Pisupati, I learnt leadership skills. A calm personality, he greets everyone with a happy smile and never loses his cool. This aspect of his personality inspires others to be proactive and be open in their communication with him.