Develop a data mining pipeline using R language comparing two models precision

Sushmitha Tharla & Shravya Rani Damarapelli

2022-11-08

36.2.1 Introduction

In the running decades trade market has gained immense attention due to economical growth around the World. There have been several retail companies competing in the market for enhancing their sale, but there are several controlling factors such as economic condition of the area where that retail shop belongs, holidays,weather of the area etc. Thus for a retail company it is very important to study the areal factors and predict expected business for a better profit and service, and developing a suitable model including all the required parameter in most convenient and easy way to accomplish. Here in this project we will compare performance of multiple linear regression(MLR) approach and Random forest(RF) approach for predicting our desired outcome, Walmart weekly sale using predictors like Holiday flag, Fuel-price, prevailing consumer price index, prevailing unemployment rate of the area and temperature . Multiple linear regression is a simple way to build a relationship between dependent and independent variables. Random forest is a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of overcoming over-fitting problem of individual decision tree.

###36.2.2 Related Works

However huge number of studies (Zhang et al.,2017; Xuefeng et al.,2021) revealed that RF model perform better than MLR showing lower error indices (MAE and RMSE) and higher R^2, possible reason may be advantages over other statistical modeling methods, such as the ability to model highly nonlinear dimensional relationships(Huang et al.,2022).

36.2.3 Methods

For accomplishing my objective I will manually fit MLR model using matrix method then build RF regression model by using R in-built package "randomForest".

Library loading and data preparation.

```
library(hms)
library(lubridate)
library(tidyverse)
library(reshape2)
library(ggplot2)
library(matlib)
library(dplyr)
library(tidymodels)
library(corrplot)
library(RColorBrewer)
data<-read.csv("Walmart_Store_sales.csv")
#str(data)
```

```
data$Date<-dmy(data$Date)
# Pre-processing Data
#Converting Holiday_Flag variable to a factor variable.
data$Holiday_Flag<-as.factor(data$Holiday_Flag)
#data[,c(1,3:6)]<-as.numeric(data[,c(1,3:6)])
#data$Store<-as.factor(data$Store)
#str(data)
dat = data %>%
   select(-Store,-Date)
dat<- dat %>% mutate_at(c('Weekly_Sales','Temperature','Fuel_Price','CPI','Unemployment'), as.numeric)

#Centralizing each independent continuous variables to avoid problems of Multicollinearity.
#dat$Temperature<-dat$Temperature-mean(dat$Temperature)
#dat$Fuel_Price<-dat$Fuel_Price-mean(dat$Fuel_Price)
#dat$CPI<-dat$CPI-mean(dat$CPI)
#dat$Unemployment<-dat$Unemployment-mean(dat$Unemployment)
```

Checking whether there are NA values.

```
anyNA(dat)
```

```
## [1] FALSE
```

```
colSums(is.na(dat))
```

```
## Weekly_Sales Holiday_Flag  Temperature   Fuel_Price          CPI Unemployment
##            0            0            0            0            0            0
```

Thus no data is missing in the data.

Checking for any inter correlation between numerical independent variables.

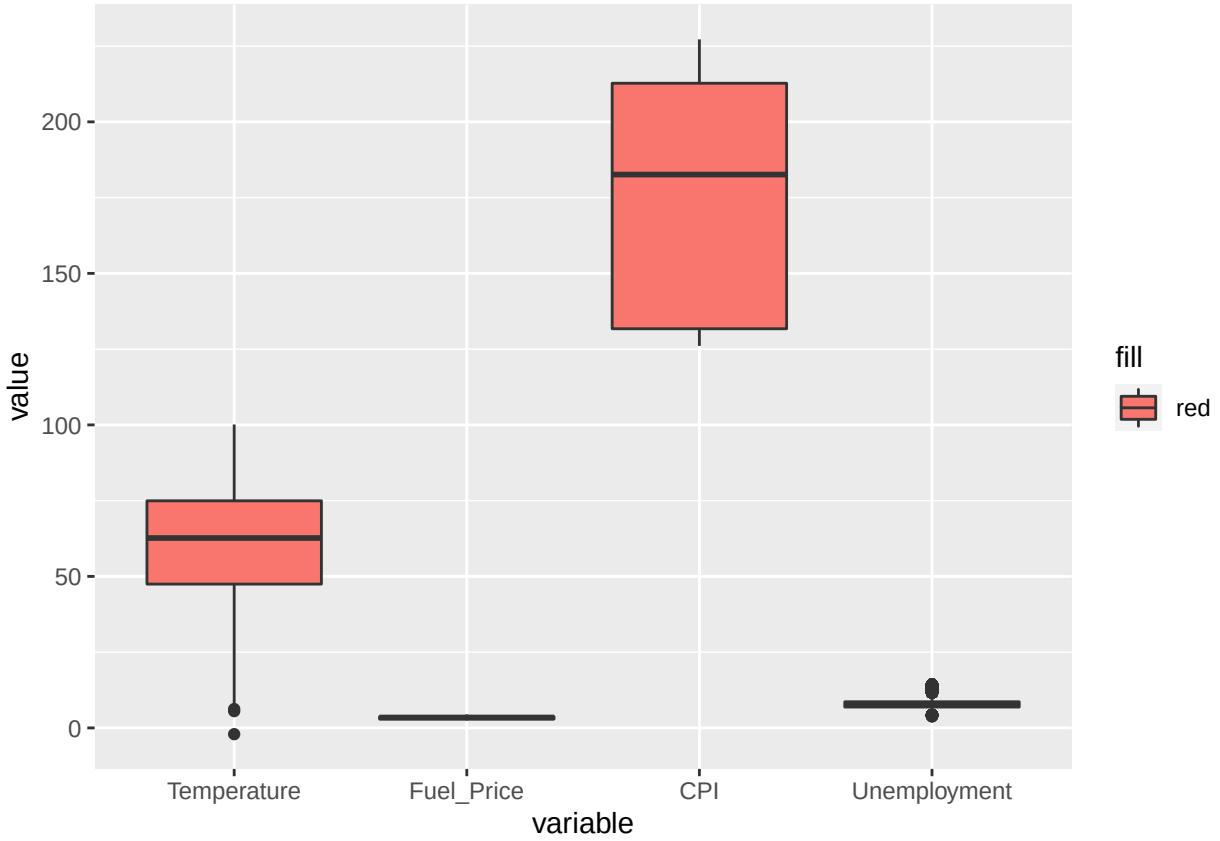
```
cor(dat[,c(3,4,5,6)])
```

```
##              Temperature  Fuel_Price        CPI Unemployment
## Temperature    1.0000000  0.14498181  0.1768877   0.10115786
## Fuel_Price     0.1449818  1.00000000 -0.1706418  -0.03468374
## CPI            0.1768877 -0.17064180  1.0000000  -0.30202006
## Unemployment   0.1011579 -0.03468374 -0.3020201   1.00000000
```

No significant correlation between the independent variables.

Checking for outlines in dependent and independent variables.

```
dat_long <- melt(dat[,c(2:6)], id = "Holiday_Flag")
ggplot(dat_long, aes(x = variable, y = value,fill="red")) +  # ggplot function
  geom_boxplot()
```



Variable employment has several outliers. Now lets remove the outliers.

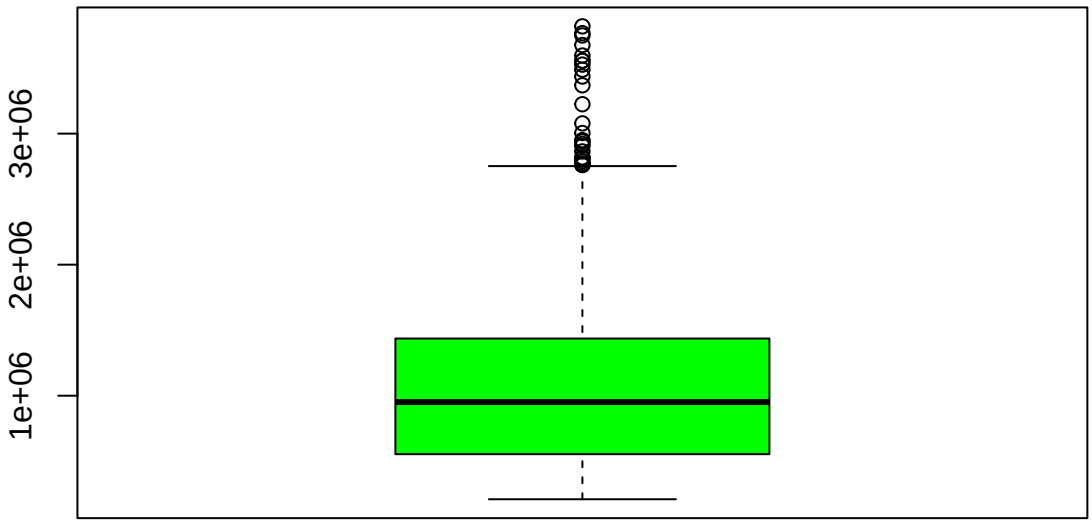
```
quartiles <- quantile(dat$Unemployment, probs=c(.25, .75), na.rm = FALSE)
IQR <- IQR(dat$Unemployment)

Lower <- quartiles[1] - 1.5*IQR
Upper <- quartiles[2] + 1.5*IQR

data_no_outlier <- subset(dat, dat$Unemployment > Lower & dat$Unemployment < Upper)
```

Checking outliers in the dependent variable.

```
boxplot(data_no_outlier$Weekly_Sales,col="green")
```



Here the dependent variable is upper skewed , thus it is very important to remove the outliers for getting a less error model.

```
quartiles <- quantile(data_no_outlier$Weekly_Sales, probs=c(.25, .75), na.rm = FALSE)
IQR <- IQR(data_no_outlier$Weekly_Sales)

Lower <- quartiles[1] - 1.5*IQR
Upper <- quartiles[2] + 1.5*IQR
data_no_outlier<-subset(data_no_outlier,data_no_outlier$Weekly_Sales>Lower &data_no_outlier$Weekly_Sales
dat<-as.data.frame(data_no_outlier)
nrow(dat)
```

```
## [1] 5925
```

Fitting Multiple linear regression model

The most common Multiple regression model equation is :

$Y = \beta_0 + \beta_1 \times X_1 + \beta_2 \times X_2 + ... + \beta_k \times X_k + \varepsilon$

where k=number of regressors or predictors=5 for our model. p=number of parameters. p=k+1=6 n=total number of observations=total number of rows in the data=5954. ε = error terms or uncertainty in predictions.

This equation can be written in matrix form as:

$Y = \beta X + \varepsilon$

where,

$$Y = \begin{bmatrix} y1 \\ y2 \\ . \\ . \\ . \\ y_n \end{bmatrix}$$

Observed outcomes.

and

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} & . & . & .x_{1k} \\ 1 & x_{21} & x_{22} & x_{23} & . & . & .x_{2k} \\ . & & & & & & \\ . & & & & & & \\ . & & & & & & \\ 1 & x_{n1} & x_{n2} & x_{n3} & . & . & .x_{nk} \end{bmatrix}$$

predictor matrix.

random error matrix is

$$\varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ . \\ . \\ \varepsilon_n \end{bmatrix}$$

and our coefficient matrix is

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ . \\ . \\ \beta_k \end{bmatrix}$$

Coefficient matrix

Here noticeable part in matrix X is that first column is containing all 1s.The importance of this is that when we multiply our coefficient matrix β with predictor matrix X then first term of each equation formed will be β_0. Therefore the coefficient of X we multiply with β_0 is simply 1. So it is crucially important that we create a column of 1s as first column in model matrix X.

In order to get a best fit multiple linear regression model we must minimize the sum of squared error.

Here the sum of squared error L is:

$$L = \sum_{i=1}^{n} {\varepsilon_i}^2 = (y - X\beta)'(y - X\beta)$$

The least square estimator $\hat{\beta}$ is the solution for β in the equation:

$\frac{\partial L}{\partial \beta} = 0.$

Here $\hat{\beta}$ is a vector that contains all our estimates for the parameters in our model.

The minimization of L in calculus leads to the normalization equation: $X'X\hat{\beta} = X'Y$

Solution of this normal equation leads to solution for $\hat{\beta}$ as:

$\hat{\beta} = (X'X)^{-1}X'Y$

36.2.4. Results and Discussion:

Calculating B-estimates of MLR

```
df<-dat %>% mutate(D=1)
#creating Y-vector.
Y_train<-matrix(df$Weekly_Sales,ncol = 1)

#creating X-vector.
X_train<-matrix(c(df$D,df$Holiday_Flag,df$Temperature,df$Fuel_Price,df$CPI,df$Unemployment),ncol = 6)
k<-ncol(dat[,c(-1)])
p<-k+1
n<-nrow(Y_train)

#taking transpose of X-vactor.
XT<-t(X_train)
#XT[,1:6]
#head(XT)

#multiply X-transpose by x.
XTX<-XT %*% X_train

#take inverse of multiple.
XTXinv<-inv(XTX)

#calculation XTY
XTY<-XT %*% Y_train

#calculating Beta.
Beta<-XTXinv %*% XTY
data <- c(1, 2, 7, 2, 8, 4, 3, 0, 9)
A <- matrix(data, nrow = 3, ncol = 3)

A_T <- t(A)
```

$\hat{\beta}$ estimates for multiple linear model are:

```
Beta
```

```
##               [,1]
## [1,] 1606805.50303
## [2,]   44065.48933
## [3,]     715.73962
## [4,]   -2613.74345
## [5,]      66.54202
## [6,]  -39945.86093
```

Now we will calculate the statistical properties of the model and estimates.

residuals $e = (y - \hat{y})$

Squared Standard error(SSE)

$$\hat{\sigma}_2 = \frac{\sum_{i=1}^{n} e_i^2}{n-p} = \frac{SS_E}{n-p}$$

Root Mean Square Error(RMSE)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} e_i^2}{n}}$$

and

Mean Absolute Error(MAE)

$$MAE = \frac{\sum_{i=1}^{n} |(Y_i - \hat{Y}_i)|}{n} = \frac{\sum e_i}{n}$$

```
e<-(Y_train-X_train %*% Beta)^2
#residuals
#e

SSE<-sum(e^2)/(n-p)
RMSE<-sqrt(sum(e)/n)
MAE<-(sum(abs(sqrt(e))))/n
cbind(RMSE,MAE)
```

```
##         RMSE      MAE
## [1,] 655051 561616.5
```

Now our regression sum of squares

$$SSR = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

and the total sum of squares is

$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

```
SSR<-sum((X_train %*% Beta-mean(Y_train))^2)
SST<-sum((Y_train-mean(Y_train))^2)
data.frame(SSR,SST)
```

```
##             SSR          SST
## 1 7.460291e+14 1.816981e+15
```

Now we will run a F-statistics test for obtaining significance of our overall model.

F-statistics is $\frac{\frac{SSR}{k}}{SSE}$

```
F0<-(SSR)/(SSE)
pf(F0,k,n-p)
```

```
## [1] 6.508509e-22
```

Here the p-value from f-statistics test indicates that model is statistically significant. Now to evaluate model performance we will calculate R^2 value by using the formula.

```
R2<-1-SSR/SST
error<-sqrt(SST/n)
data.frame(R2=c(0.0167),error=c(550890))
```

```
##       R2  error
## 1 0.0167 550890
```

The r-value here indicates that the independent variables can describe only 1.7 % variations of the dependent variable. Whereas associated standard error is 550890 which is quite high. Although the fitted model is not a good fit still as my main goal is to compare performance of two model so I will use this model.

Now I shall estimate the same model parameters using random forest regression method.

Random forest

```
library(randomForest)
rf<-randomForest(Weekly_Sales~.,dat,ntree = 500)
print(rf)
```

```
##
## Call:
##  randomForest(formula = Weekly_Sales ~ ., data = dat, ntree = 500)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 265026858649
##                     % Var explained: 13.58
```

Here also we will derive the statistical values as RMSE, MAE and R^2 for Random Forest Model.

```
y_pred = predict(rf, newdata = dat[,c(-1)])
e_rf<-(Y_train-y_pred)^2
#residuals
#e

SSE_rf<-sum((e_rf)^2)/(n-p)
RMSE<-sqrt(sum((e_rf))/n)
MAE<-(sum(sqrt(e_rf)))/n
cbind(RMSE,MAE)
```

```
##          RMSE      MAE
## [1,] 470838.8 399713.6
```

```
SSR_rf<-sum((y_pred-(Y_train))^2)
SST_rf<-sum((Y_train-mean(Y_train))^2)
```

p-value from F-statistics in random forest model is,

```
F0_rf<-(SSR_rf/k)/(SSE_rf)
pf(F0_rf,5,n-p)
```

```
## [1] 7.911227e-22
```

Thus the model is significant.

Thus the standard error for random forest model is square root of mean of squared residuals which is

```
error_rf<-sqrt(rf$mse[500])
error_rf
```

```
## [1] 514807.6
```

R^2 for random regression model.

```
R2<-rf$rsq[500]
R2
```

```
## [1] 0.1357727
```

R^2 value of 0.136 also indicate that in contrary to multiple linear regression model 13% of variances in the dependent variables are explained by independent predictors in random forest model.

###36.2.5 Conclusion:

From my whole study a comparison matrix can be built as:

$$\begin{bmatrix} Names & RMSE & MAE & R^2 \\ MLR & 655051 & 561616.5 & 0.0167 \\ RF & 488663.3 & 409985 & 0.1312953 \end{bmatrix}$$

From comparison matrix one can easily depict that all the error statistics are better for RF model than MLR. Also R^2 value 0.13 is greater for RF model than 0.0167 which means for RF model can explain 13% variation in the dependent variable whereas MLR can only 1.7%. For any study related to model selection for predicting any retail outcome RF model can give better result than MLR. Although MLR is advanced with less run time than RF as RF runs on creating several modeling trees and summarize them.

36.2.6 Data and software availability:

Here Walmart retail data (https://www.kaggle.com/datasets/aditya6196/retail-analysis-with-walmart-data?resource=download) was used for the analysis.

and all the analyses were conducted using R-Studio. Link to download the software is (https://cran.r-project.org/bin/windows/base/R-4.2.2-win.exe).

36.2.7 References

1) Huan Zhang, Pengbao Wu, Aijing Yin, Xiaohui Yang, Ming Zhang, Chao Gao, Prediction of soil organic carbon in an intensively managed reclamation zone of eastern China: A comparison of multiple linear regressions and the random forest model, Science of The Total Environment, Volume 592,2017,Pages 704-713,ISSN 0048-9697,https://doi.org/10.1016/j.scitotenv.2017.02.146.

2) Xuefeng Xie, Tao Wu, Ming Zhu, Guojun Jiang, Yan Xu, Xiaohan Wang, Lijie Pu, Comparison of random forest and multiple linear regression models for estimation of soil extracellular enzyme activities in agricultural reclaimed coastal saline land, Ecological Indicators, Volume 120,2021,106925,ISSN 1470-160X,https://doi.org/10.1016/j.ecolind.2020.106925.

3) Huang, Li-Ying, Fang-Yu Chen, Mao-Jhen Jhou, Chun-Heng Kuo, Chung-Ze Wu, Chieh-Hua Lu, Yen-Lin Chen, Dee Pei, Yu-Fang Cheng, and Chi-Jie Lu. 2022. "Comparing Multiple Linear Regression and Machine Learning in Predicting Diabetic Urine Albumin–Creatinine Ratio in a 4-Year Follow-Up Study" Journal of Clinical Medicine 11, no. 13: 3661. https://doi.org/10.3390/jcm11133661