Multiple Linear Regression

Cross Validation - I

Contents

Cross Validation in Predictive Modeling

Cross Validation is a
process of evaluating a model on
'Out of Sample' data

- **Model performance measures** such as R-squared or Root Mean Squared Error (RMSE) **tend to be optimistic on 'In Sample Data'**
- **Model performance on Out of sample data** gives a more realistic picture of model performance.

Cross validation is important because although a model is built on historical data, ultimately it is to be used on future data. However good the model, if it fails on out of sample data then it defeats the purpose of predictive modelling.

Cross Validation in Predictive Modeling

There are different approaches to cross validation. The five most significant are:

- Hold-Out Validation
- K-Fold Cross Validation
- Repeated K-Fold Cross Validation
- Leave-One-Out Cross Validation (LOOCV)
- Resampling Validation Method (Bootstrap Method)

Introduction To Package Caret in R

- The **caret** package (short for **C**lassification **A**nd **Re**gression **T**raining) is a **set of functions that attempt to streamline the process for creating predictive models**
- The package contains tools for:
  - Data splitting
  - Pre-processing
  - Feature selection
  - Model tuning using re-sampling
  - Variable importance estimation

Case Study – Modelling Motor Insurance Claims

Background

- A car insurance company collects range of information from its customers at the time of buying and claiming insurance. The company wishes to check if any of this information can be used to model and predict claim amount

Objective

- To model motor insurance claim amount based on vehicle related information collected at the time of registering and claiming insurance

Available Information

- **Sample size is 1000**
- Independent Variables: Vehicle Information – **Vehicle Age, Engine Capacity, Length and Weight of the Vehicle**
- Dependent Variable: **Claim Amount**

Data Snapshot

Motor_Claims

Independent variables

Dependent variable

Observations

vehage	CC	Length	Weight	claimamt
4	1495	4250	1023	72000
2	1061	3495	875	72000
2	1405	3675	980	50400
7	1298	4090	930	39960
2	1495	4250	1023	106800
1	1086	3565	854	69592.8

Columns	Description	Type	Measurement	Possible values
vehage	Age of the vehicle at the time of claim	integer	Years	positive values
CC	Engine capacity	numeric	cc	positive values
Length	Length of the vehicle	numeric	mm	positive values
Weight	Weight of the vehicle	numeric	kg	positive values
claimamt	Claim amount	numeric	INR	positive values

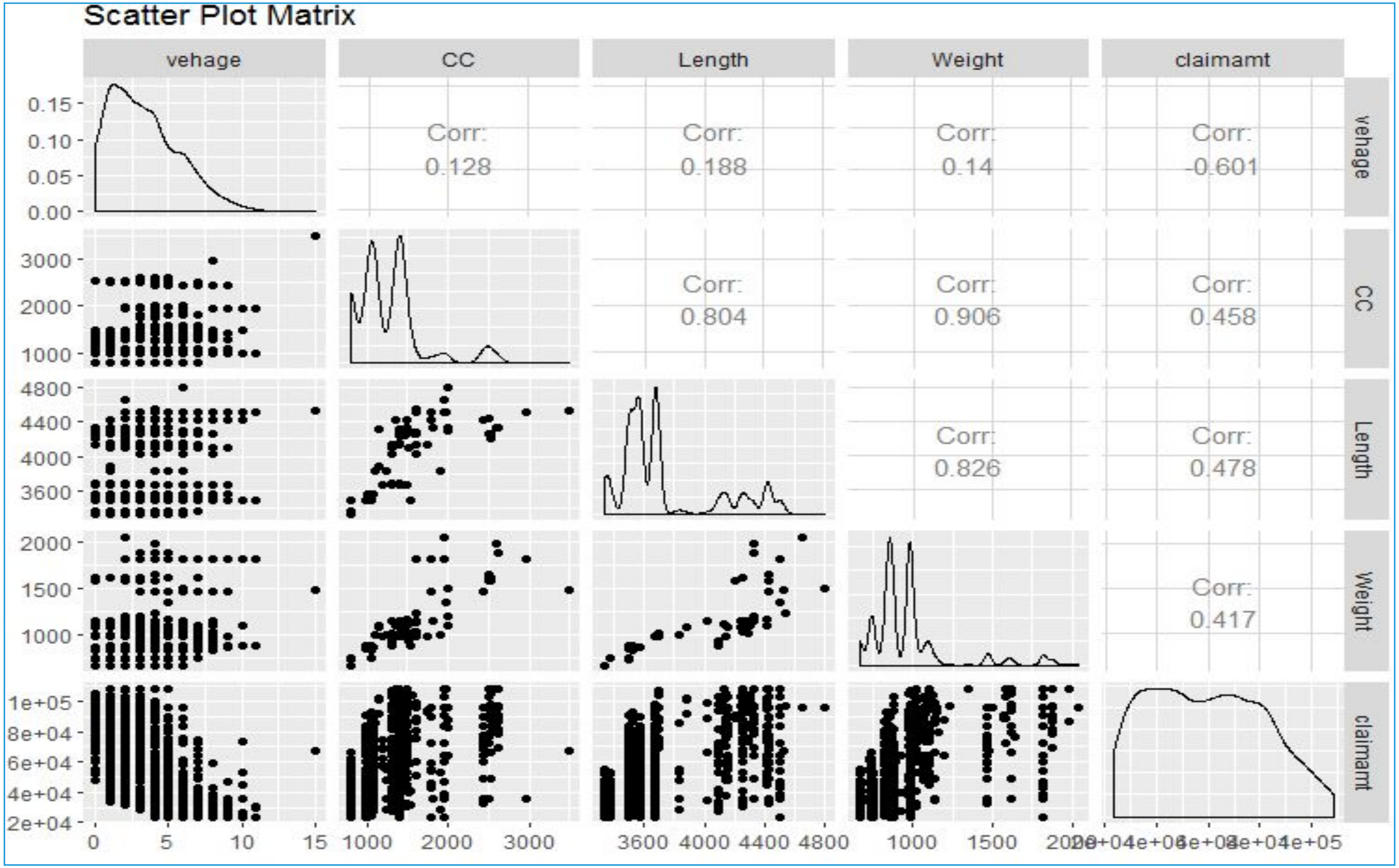
Data Visualization

```
#Importing the Data
motor<-read.csv("Motor_Claims.csv",header=TRUE)

# Install package “GGally”, if not installed
previously
# plotting correlation matrix
library(GGally)

ggpairs(motor[,c("vehage","CC","Length","Weight","claimamt")],
title="Scatter Plot Matrix" ,
columnLabels = c("vehage","CC","Length","Weight","claimamt"))
```

Using ggpairs in library GGally to get a scatter plot of the variables in the data set

Scatter Plot

```
# Output
```



Interpretation : Correlation between some of the independent variables are high suggesting a chance of multicollinearity.

Detecting Multicollinearity

Linear regression model

```
motor_model<-lm(claimamt~Length+CC+vehage+Weight, data=motor)
summary(motor_model)
```

Output

```
Residuals:
   Min     1Q Median     3Q    Max 
-45577  -8007     39   7852  40561 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -54765.128   5569.375  -9.833  < 2e-16 ***
Length          35.461      1.990  17.824  < 2e-16 ***
CC              15.413      2.114   7.292 6.23e-13 ***
vehage       -6637.213    154.098 -43.071  < 2e-16 ***
Weight         -16.255      3.678  -4.420 1.10e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11360 on 995 degrees of freedom
Multiple R-squared:  0.7379,	Adjusted R-squared:  0.7368 
F-statistic: 700.3 on 4 and 995 DF,  p-value: < 2.2e-16
```

Interpretation:
All the independent variables in the model are significant.

Detecting Multicollinearity

```
# Obtaining vif
```

```r
library(car)
vif(motor_model)
```

vif in library car gives the VIFs of the independent variables in the regression model.

```
# Output showing VIF
```

```
  Length       CC     vehage    Weight
3.396171 5.881428 1.038357 6.552811
```

Interpretation:
CC and Weight have VIF >5

Re- Modelling

```
# New model
```

```r
motor_model1<-lm(claimamt~Length+CC+vehage,data=motor)
summary(motor_model1)
```

New model after removing weight to adjust for multicollinearity

```
# Output of the new model
```

```
Residuals:
   Min      1Q Median     3Q    Max 
-47069  -7673    -14   7783  40447 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -49195.196   5475.151  -8.985  < 2e-16 ***
Length          32.065      1.852  17.312  < 2e-16 ***
CC               8.689      1.481   5.867 6.02e-09 ***
vehage       -6638.076    155.525 -42.682  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11470 on 996 degrees of freedom
Multiple R-squared:  0.7327,    Adjusted R-squared:  0.7319 
F-statistic: 910.3 on 3 and 996 DF,  p-value: < 2.2e-16
```

Interpretation:
All the independent variables in the model are significant.

* **Dropping one independent variable is one of the remedial measures to adjust for multicollinearity(when not many variables are multicollinear). As weight had the maximum VIF value, it is excluded from the model to adjust for multicollinearity.**

VIF of New Model

```
# VIF

vif(motor_model1)
```

Getting VIFs of the independent variables in the new model

```
# VIFs of variables in the new model
```

Length	CC	vehage
2.889718	2.833931	1.038355

Interpretation:
All VIF s are <5.

RMSE of the Model

```
# RMSE of the model
```

```
motor$res<-residuals(motor_model1)
RMSEmotor<-sqrt(mean(motor$res**2))
RMSEmotor
```

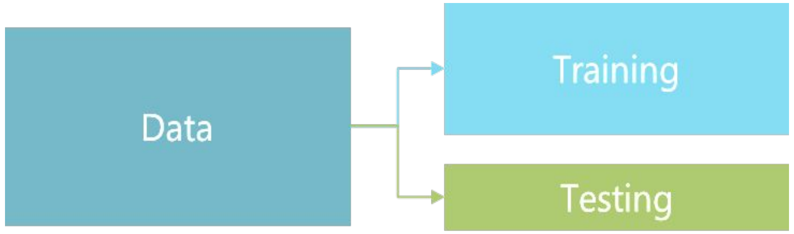
```
# Output
```

```
[1] 11444.51
```

Interpretation :
RMSE for the model.

* RMSE of this model will be used later to measure the performance of the model on cross validation

Hold-Out Validation



In Hold-Out validation method, **available data is split into two non-overlapped parts: 'Training Data' and 'Testing Data'**

- The model is,
  - Developed using training data
  - Evaluated using testing data

- Training data should have more sample size. Typically 70%-80% data is used for model development

* **Note : This ppt is in continuation of previous ppt. So the data considered is the same, "Motor_Claims" data.**

Hold Out Validation in R

```
# Creation of Datasets for Validation
# Install & load package “caret”
```

```
motor <- read.csv("Motor_Claims.csv", header=TRUE)

install.packages("caret")
library(caret)

Index <- createDataPartition(motor$claimamt,p=0.8,list=FALSE)

head(index)
dim(index)
```

- *createDataPartition() generates list of observation numbers to be included in training data.*
- *p= is the percentage of data that goes into training data.*
- *list= specifies if results should be in a list format or matrix.*

```
# Output of first 6 rows of index
```

```
     Resample1
[1,]         1
[2,]         2
[3,]         3
[4,]         5
[5,]         6
[6,]         7
```

```
# Output dim(index)
```

```
[1] 800   1
```

* Note : While splitting data, observations are selected randomly, so the output will vary.

Hold Out Validation in R

```
traindata<-motor[index,]
testdata<-motor[-index,]

dim(traindata)
dim(testdata)
```

training and testing data sets for the Motor Insurance Data

```
# Output

[1] 800    5

[1] 200    5
```

Dimension of training set

Dimension of testing set

Hold Out Validation in R

```
# RMSE of training data
```

```
motor_trn_model<-lm(claimamt~Length+CC+vehage,data=traindata)

traindata$res<-residuals(motor_trn_model)
head(traindata)

RMSEtrain<-sqrt(mean(traindata$res**2))
RMSEtrain
```

```
# RMSE for testing data
```

```
testdata$pred<-predict(motor_trn_model,testdata)

testdata$res<-(testdata$claimamt-testdata$pred)

RMSEtest<-sqrt(mean(testdata$res**2))
RMSEtest
```

* For testing data, since we work with predictors, we calculate residuals as observed values minus the predicted values

Hold Out Validation in R

```
# Output
```

	vehage	CC	Length	Weight	claimamt	res
1	4	1495	4250	1023	72000.0	-1603.964
2	2	1061	3495	875	72000.0	12821.441
3	2	1405	3675	980	50400.0	-17651.055
5	2	1495	4250	1023	106800.0	19996.145
6	1	1086	3565	854	69592.8	1397.857
7	4	796	3495	740	38400.0	-5059.737

res column gives the residual for the training set data

```
[1] 11444.51
```

RMSE for the original data with all data points

```
[1] 11345.8
```

RMSE for the training data

```
[1] 11868.11
```

RMSE for testing data

Interpretations :
Comparing RMSE of training and testing data shows not much difference between the two and also are in line with the RMSE of the original model. Thus we can say that the model is stable.

* **There is no thumb rule for comparison of model performance. The only criterion is performance measure for training and testing data should not be too diverse**

Quick Recap

Cross Validation – Meaning and Need	• Process of evaluating the model on 'Out of Sample' data • Important because although a model is built on historical data, ultimately it is to be used on future data