# Big mart sales prediction based on generalized linear regression and Lasso regression

William Cheng

# Abstract

The generalized linear regression is a flexible generalization of ordinary linear regression that allows for response variables that have error distribution models other than a normal distribution. And the unknown parameters are typically estimated with maximum likelihood.  Lasso regression which is the generalized linear regression via penalty item can avoid the overfitting issue and help in selecting variables.  Lasso regression uses shrinkage, where data values are shrunk towards a central point.

I'm going to demonstrate how generalized linear regression and generalized Lasso regression perform in big mart sales prediction.

# Materials and Methodology

For this research project, R is the programming language that has been used. In the coding it is data analysis libraries that makes the coding efficient.

- Outliers
- Stats
- Caret
- Extremevalues
- Corrplot
- MASS
- CaretEnsemble

# Data Overview

The data contains  2013 sales data for 1559 products across 10 stores in different cities. Also, certain attributes of each product and store have been defined. The goal is to build a predictive model and find out the sales of each product at a particular store.

There are total 14204 data observations  and I randomly separated the dataset into training data (8523) and testing data(5681).

# Data Cleaning

# Missing Data

**Item Weight category**

- Find the missing weight from another store for the specific product.
- The second solution is there is only one record for some products, so we can use the average weight from the whole column to impute the missing values.

**Outlet Size**

Use  mode which is “Medium” and replaced the empty value by mode

**Item visibility**

Use average item visibility from the same kind of product to replace the zero values

# Data Correction

**Item Fat Content Section**

- Find typos that marked “Low Fat” as “LF” and “low fat”; and marked “Regular” as “reg”.
- Find “LF/low fat”and “reg” and replace them with “Low Fat” and “Regular”.
- Replace the item fact content for item type “Health and Hygiene,Household, and Others“ with “NA” since they are not food and drink.

In order to avoid overfitting issue, we can combine the item type into “Food”, “Drink”, and “Non-Consumable” these three categories.

# Data Transformation

**Establishment Year**

We can use the year operation instead of establishment year

Formula:  2013 -Outlet_Establishment_Year

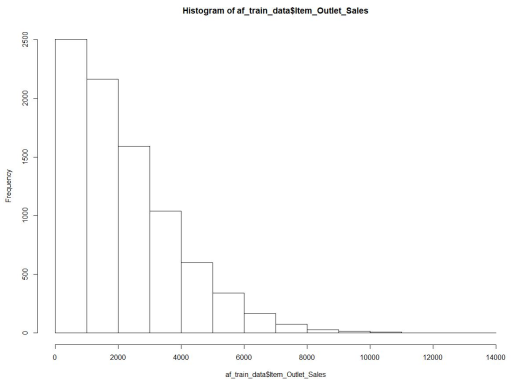
| Outlet_Establishment_Year |
|---|
| 1999 |
| 2009 |
| 1999 |
| 1998 |
| 1987 |
| 2009 |
| 1987 |
| 1985 |
| 2002 |
| 2007 |
| 1999 |
| 1997 |
| 1999 |
| 1997 |
| 1987 |
| 1997 |
| 2009 |

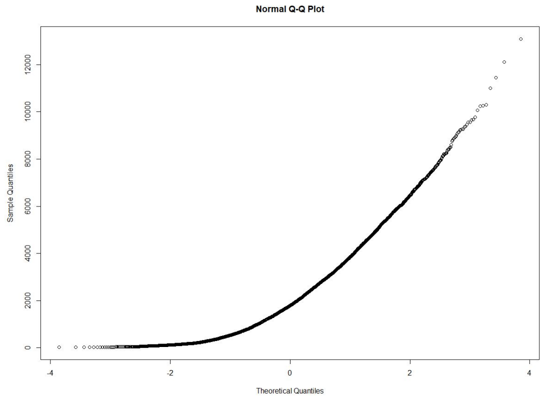
| Years_Of_Operation |
|---|
| 14 |
| 4 |
| 14 |
| 15 |
| 26 |
| 4 |
| 26 |
| 28 |
| 11 |
| 6 |
| 14 |
| 16 |
| 14 |
| 16 |
| 26 |
| 16 |
| 4 |

# Outlier

# Outlier

- Use histogram and qq diagram to check whether the Item outlet sales are normally distributed.

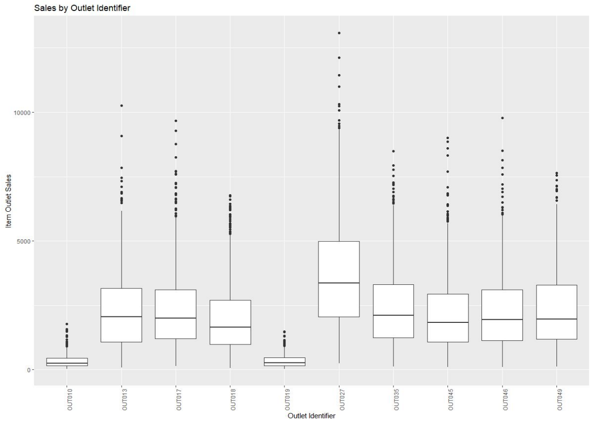


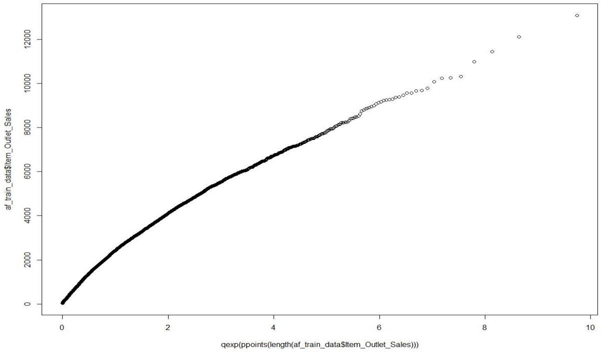




- Use boxplot to detect the potential outliers which outside the 90% percentile of data.
- Use qq plot  to test whether it is an exponential distribution as below.

# Outlier

- Each outlet have few potential outliers. And OUT027 has much higher potential sales outlier data. Also OUT027 has higher average sales compared to other outlets. From the graph, we can see it’s an approximate exponential distribution.



# Outlier

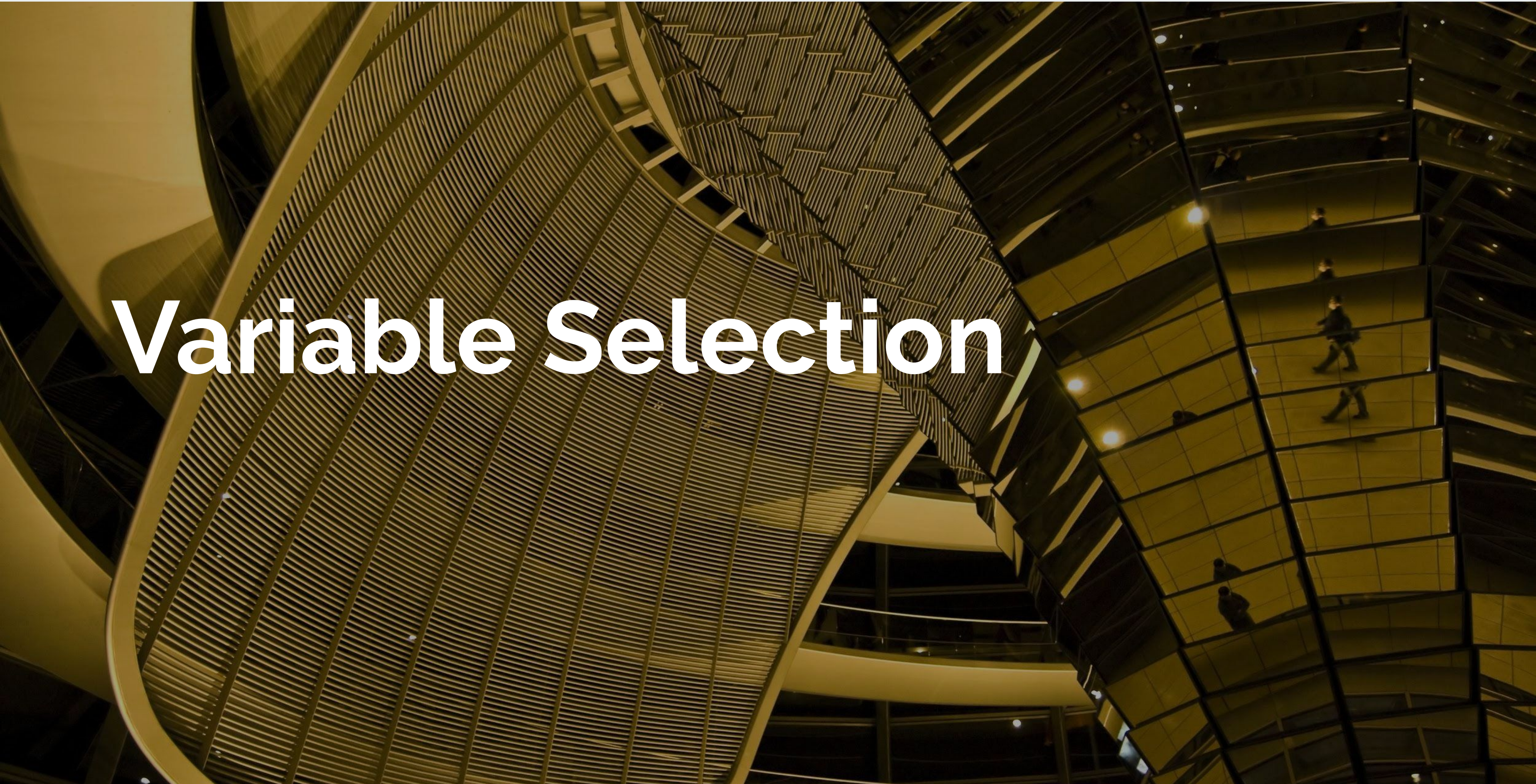
I used "getOutlier" in R to test the outliers which give use the results saying there is no outlier. So we don’t need to remove any outliers here according to getOutliers tool.

```
getOutliers(af_train_data$Item_Outlet_Sales, distribution = 'exponential')
```

Below are the results from the getOutlier function performed above.

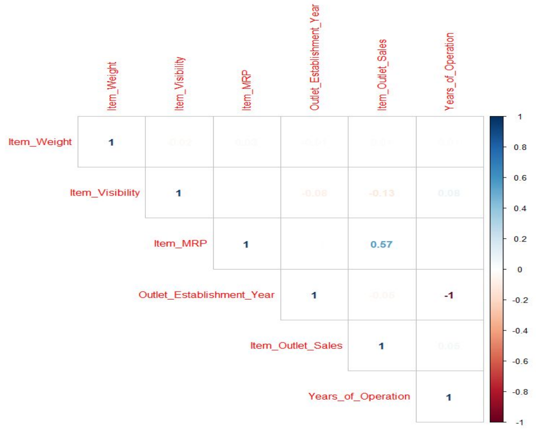
```
getOutliers:
 Left Right
    0     0
```

# Variable Selection



# Variable Selection

First, I checked the correlation between variables According to the correlation matrix, there are no correlat between variables.




I used "stepwise" variable selection for linear model.

```
v_selection <- stepAIC(model1, direction = "both", trace = FALSE)
```

I found out that variable - Item MRP, Location Type Tier, Outlet Size, Outlet Type and years of operation are significant.

Besides I added combined item type in the model to distinguish the product type.

# Modeling

# Generalized Linear Regression vs LASSO

With the selection of the variables with high significance, I built a generalized linear regression model and Lasso regression in the training dataset with 3 fold cross -validation.

The functions used in R is "glm" for generalized linear regression, "glmnet" for Lasso regression.

```
control <- trainControl(method = "repeatedcv", number
= 10, repeats = 3, savePredictions = TRUE, classProbs
= TRUE)
mList <- c('glm','glmnet')

fit_models <- caretList(Item_Outlet_Sales~
                    Combined_Item_Type
                    +Item_MRP
                    +Outlet_Identifier
                    +Outlet_Location_Type
                    +Outlet_Size
                    +Outlet_Type
                    +Years_of_Operation,
                    data = af_train_data,
                    trControl = control,
                    methodList = mList)
```

# Prediction results for training dataset

The RMSE(Root Mean Square Error) for generalized linear regression model is 1128.372.

And for the glmnet model we have the optimal value for the lowest RMSE is alpha = 1 and lambda = 1.937, and I have the RMSE of Lasso model equal to 1128.335 which is slightly lower than generalized linear regression model. So the Lasso regression performed better in the training set

$glm
Generalized Linear Model

| RMSE | Rsquared | MAE |
|---|---|---|
| 1128.372 | 0.563201 | 837.2549 |

$glmnet

| alpha | lambda | RMSE | Rsquared | MAE |
|---|---|---|---|---|
| 0.10 | 1.937018 | 1128.384 | 0.5631983 | 836.9739 |
| 0.10 | 19.370175 | 1128.694 | 0.5630865 | 836.5560 |
| 0.10 | 193.701751 | 1152.765 | 0.5537720 | 852.0912 |
| 0.55 | 1.937018 | 1128.352 | 0.5632209 | 836.9141 |
| 0.55 | 19.370175 | 1129.686 | 0.5626180 | 836.2742 |
| 0.55 | 193.701751 | 1209.027 | 0.5178554 | 895.9940 |
| 1.00 | 1.937018 | 1128.335 | 0.5632334 | 836.8594 |
| 1.00 | 19.370175 | 1131.297 | 0.5617536 | 836.6575 |
| 1.00 | 193.701751 | 1252.429 | 0.4958680 | 927.8833 |

RMSE was used to select the optimal model using the smallest value.
The final values used for the model were alpha = 1 and lambda = 1.937018.

# Prediction

# Prediction results for testing dataset

Based on the generalized linear regression and Lasso regression built above, I used "predict" function to predict the sales on testing data.

I have the RMSE for glm is 1202.0354, and for the glmnet model, I have RMSE equal to 1202.3587. The fitness of two models are very close. However, Lasso model helps to reduce the model complexity and minimize the error for the quantitative response variables. Also it avoids the overfitting issue. So I would still suggest Lasso as the better prediction model.

```
glmnet_model <- caretStack(fit_models, methodList =
"glmnet", trControl = trainControl(method = "repeatedcv",
number = 10, repeats = 3, savePredictions = TRUE))
glmnet_model

predict_on_test <- predict(glmnet_model, newdata =
af_test_data )
Predict_on_test

glm_model <- caretStack(fit_models, method = "glm",
trControl = trainControl(method = "repeatedcv", number = 10,
repeats = 3, savePredictions = TRUE))
glm_model

predict_on_test2 <- predict(glm_model, newdata =
af_test_data )
predict_on_test2
```

# Future Work

Outliers: more sophisticated way

Variable selection: further simplify the model

# Thank you!