

## DATA 621 Homework 3

### Binary Logistic Regression

Trishita Nath

## Contents

DATA EXPLORATION.....	2
Missing Values .....	3
Boxplots .....	4
Density Plot .....	6
DATA PREPARATION .....	7
Data Splitting.....	7
Log Transformation.....	7
BoxCox Transformation .....	8
BUILD MODELS.....	10
Model 1. Backward Elimination on Log Transformed data .....	10
Model 2. Backward Elimination on BoxCox Transformed data .....	11
Model 3. Using Stepwise Regression .....	12
Model 4. Using glmulti .....	13
SELECT MODELS .....	14
Model Performance .....	14
Prediction Accuracy .....	14
Prediction the evaluation data set.....	18
References .....	19
Appendix .....	19
Description of the variables .....	19
R Code .....	19












## DATA EXPLORATION

This section explores the given data to see the data type, data structure, the correlation among the variables as well as if there are missing values in the data.

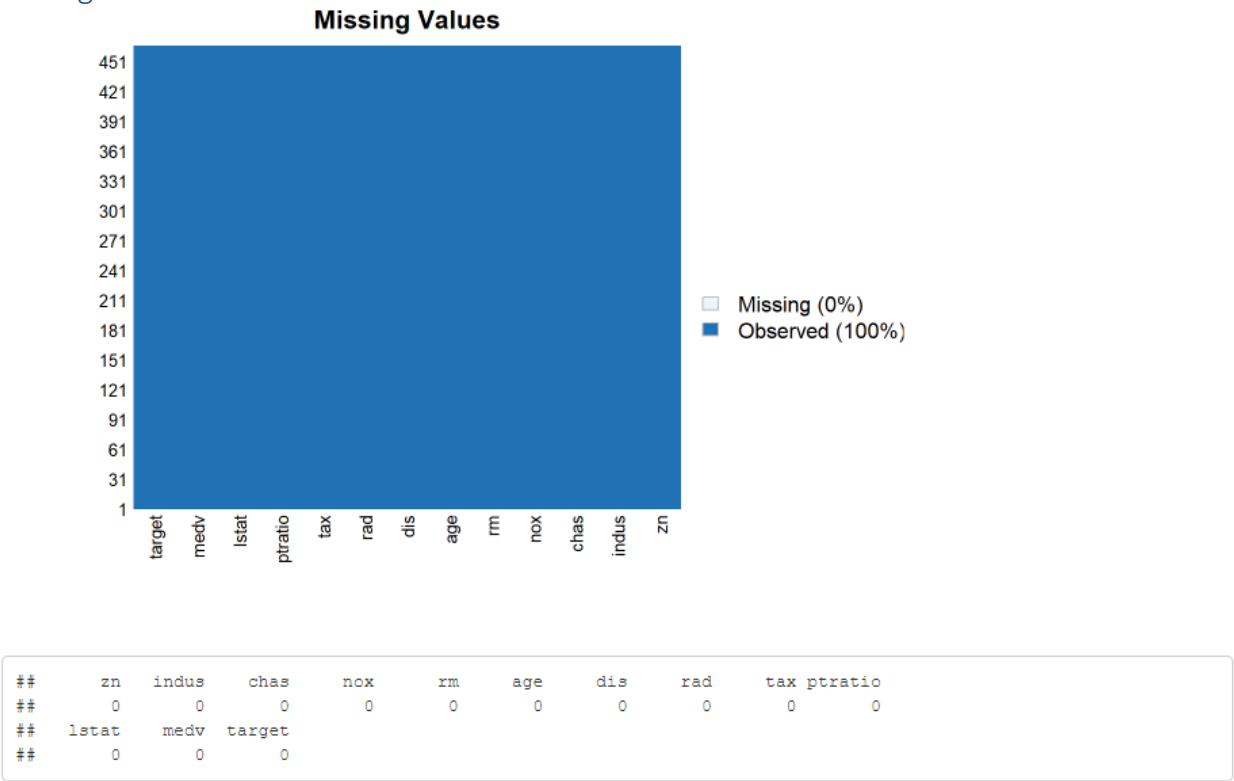
zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	target
0	19.58	0	0.605	7.929	96.2	2.0459	5	403	14.7	3.70	50.0	1
0	19.58	1	0.871	5.403	100.0	1.3216	5	403	14.7	26.82	13.4	1
0	18.10	0	0.740	6.485	100.0	1.9784	24	666	20.2	18.85	15.4	1
30	4.93	0	0.428	6.393	7.8	7.0355	6	300	16.6	5.19	23.7	0
0	2.46	0	0.488	7.155	92.2	2.7006	3	193	17.8	4.82	37.9	0
0	8.56	0	0.520	6.781	71.3	2.8561	5	384	20.9	7.67	26.5	0

From variables documentation, `target` and `chas` are the factor variables hence should be converted them into factors to aid in data exploration. The `skim` function from `skimr` package builds histogram for each numeric variable and shows number of missing values and quantiles. Double check the number of missing values with `colSums` and `misomap` functions.

skim_type	skim_variable	n_missing	complete_rate	factor.ordered	factor.n_unique	factor.top_counts	numeric.mean
factor	chas	0	1	FALSE	2	0: 433, 1: 33	NA
factor	target	0	1	FALSE	2	0: 237, 1: 229	NA
numeric	zn	0	1	NA	NA	NA	11.5772532
numeric	indus	0	1	NA	NA	NA	11.1050215
numeric	nox	0	1	NA	NA	NA	0.5543105
numeric	rm	0	1	NA	NA	NA	6.2906738
numeric	age	0	1	NA	NA	NA	68.3675966
numeric	dis	0	1	NA	NA	NA	3.7956929
numeric	rad	0	1	NA	NA	NA	9.5300429
numeric	tax	0	1	NA	NA	NA	409.5021459
numeric	ptratio	0	1	NA	NA	NA	18.3984979
numeric	lstat	0	1	NA	NA	NA	12.6314592
numeric	medv	0	1	NA	NA	NA	22.5892704

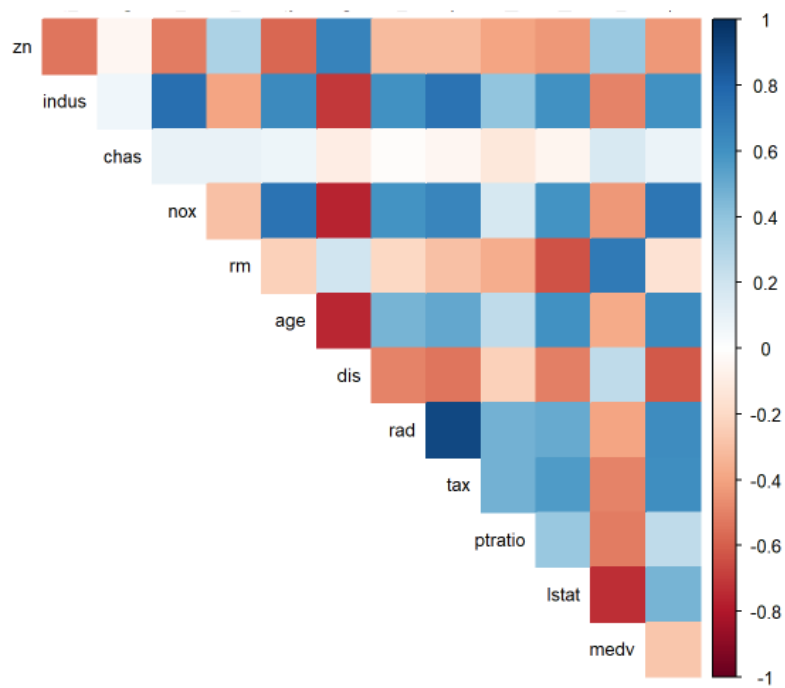
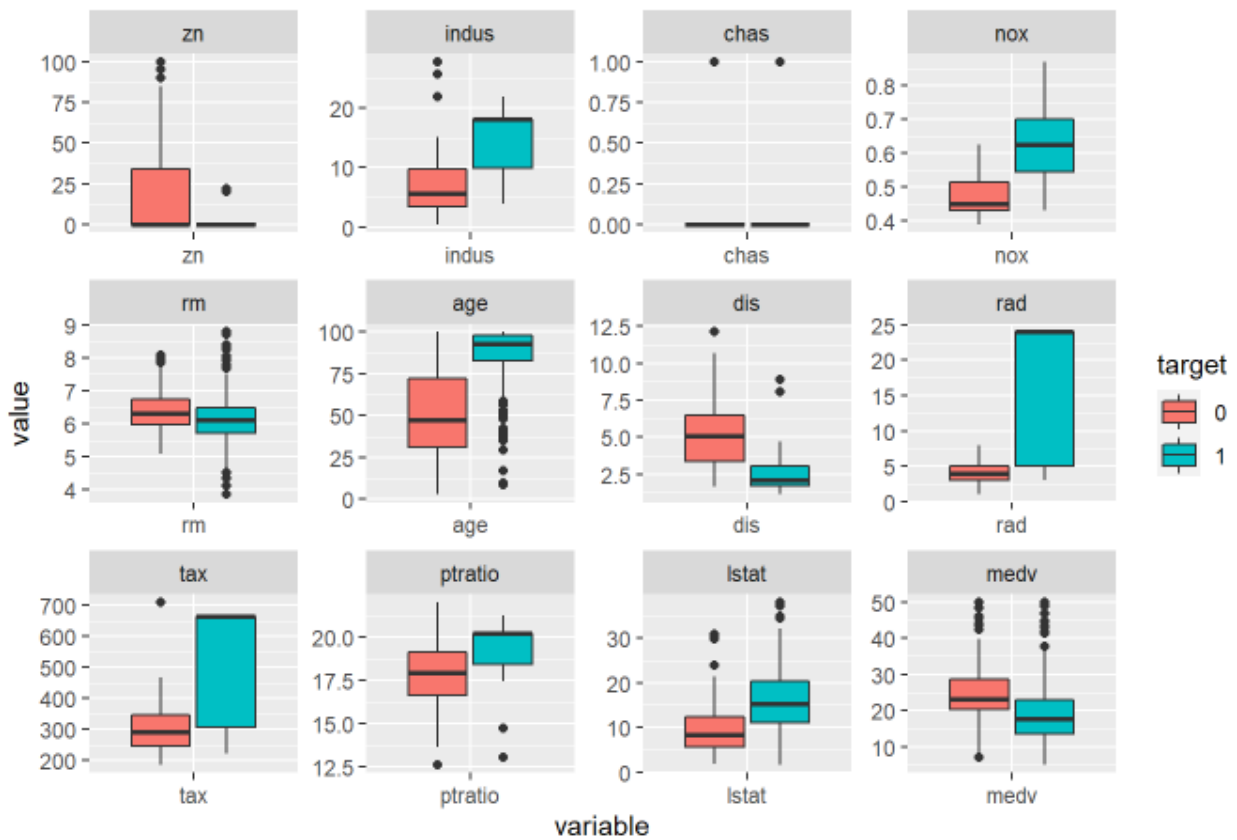
numeric.mean	numeric.sd	numeric.p0	numeric.p25	numeric.p50	numeric.p75	numeric.p100	numeric.hist
NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA
11.5772532	23.3646511	0.0000	0.000000	0.00000	16.25000	100.0000	
11.1050215	6.8458549	0.4600	5.145000	9.69000	18.10000	27.7400	
0.5543105	0.1166667	0.3890	0.448000	0.53800	0.62400	0.8710	
6.2906738	0.7048513	3.8630	5.887250	6.21000	6.62975	8.7800	
68.3675966	28.3213784	2.9000	43.875000	77.15000	94.10000	100.0000	
3.7956929	2.1069496	1.1296	2.101425	3.19095	5.21460	12.1265	
9.5300429	8.6859272	1.0000	4.000000	5.00000	24.00000	24.0000	
409.5021459	167.9000887	187.0000	281.000000	334.50000	666.00000	711.0000	
18.3984979	2.1968447	12.6000	16.900000	18.90000	20.20000	22.0000	
12.6314592	7.1018907	1.7300	7.042500	11.35000	16.93000	37.9700	
22.5892704	9.2396814	5.0000	17.025000	21.20000	25.00000	50.0000	

Missing Values



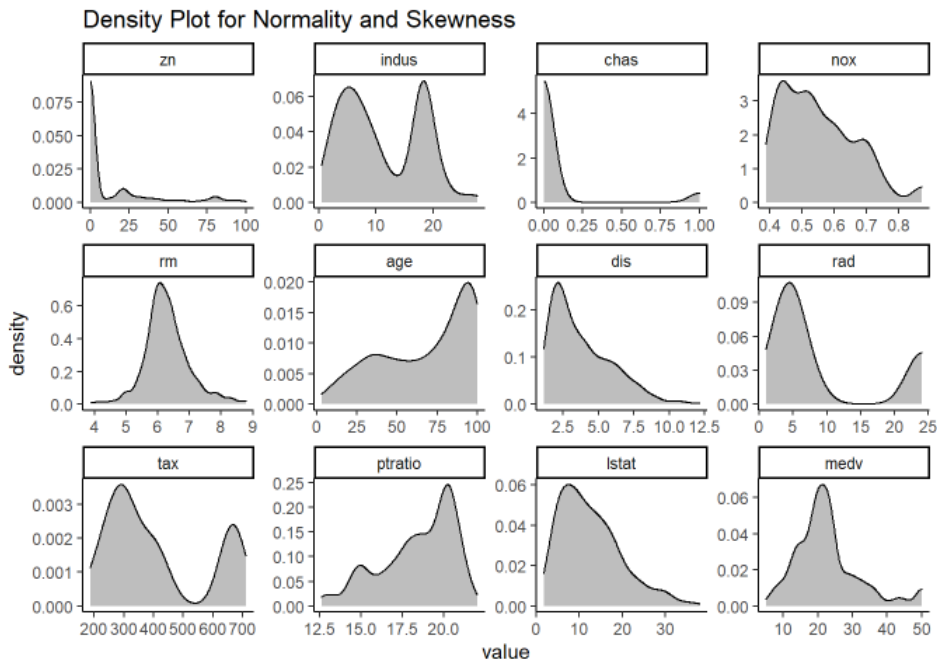
Below is a boxplot to check the correlation of predictors among themselves and with the target variable.

### Predictor Distribution with Target Variable



rowname	Variable	Correlation
target	target	1.0000000
nox	target	0.7261062
age	target	0.6301062
rad	target	0.6281049
tax	target	0.6111133
indus	target	0.6048507
lstat	target	0.4691270
ptratio	target	0.2508489
chas	target	0.0800419
rm	target	-0.1525533
medv	target	-0.2705507
zn	target	-0.4316818
dis	target	-0.6186731

## Density Plot



```
##      zn      indus      chas      nox      rm      age
## 2.17681518 0.28854503 3.33548988 0.74632807 0.47932023 -0.57770755
##      dis      rad      tax      ptratio      lstat      medv
## 0.99889262 1.01027875 0.65931363 -0.75426808 0.90558642 1.07669198
##      target
## 0.03422935
```

Observations from the correlation plot and matrix:

- nox, age, rad, tax and indus are positively correlated with target.
- lstat, ptratio and chas have weak correlations with target variable.
- dis have good negative correlation followed by zn medv and rm which do not seem to have strong correlation with target variable.

## DATA PREPARATION

From the above data exploration steps, the following issues are noted:

- Most of the variables seem to be skewed and not normally distributed.
- Outliers are seen in some variables

### Data Splitting

Split the training dataset into train and test datasets to check the accuracy of our model. Split the dataset with 70 percent train and 30 percent test data using the `createDataPartition` function from `caret` library.

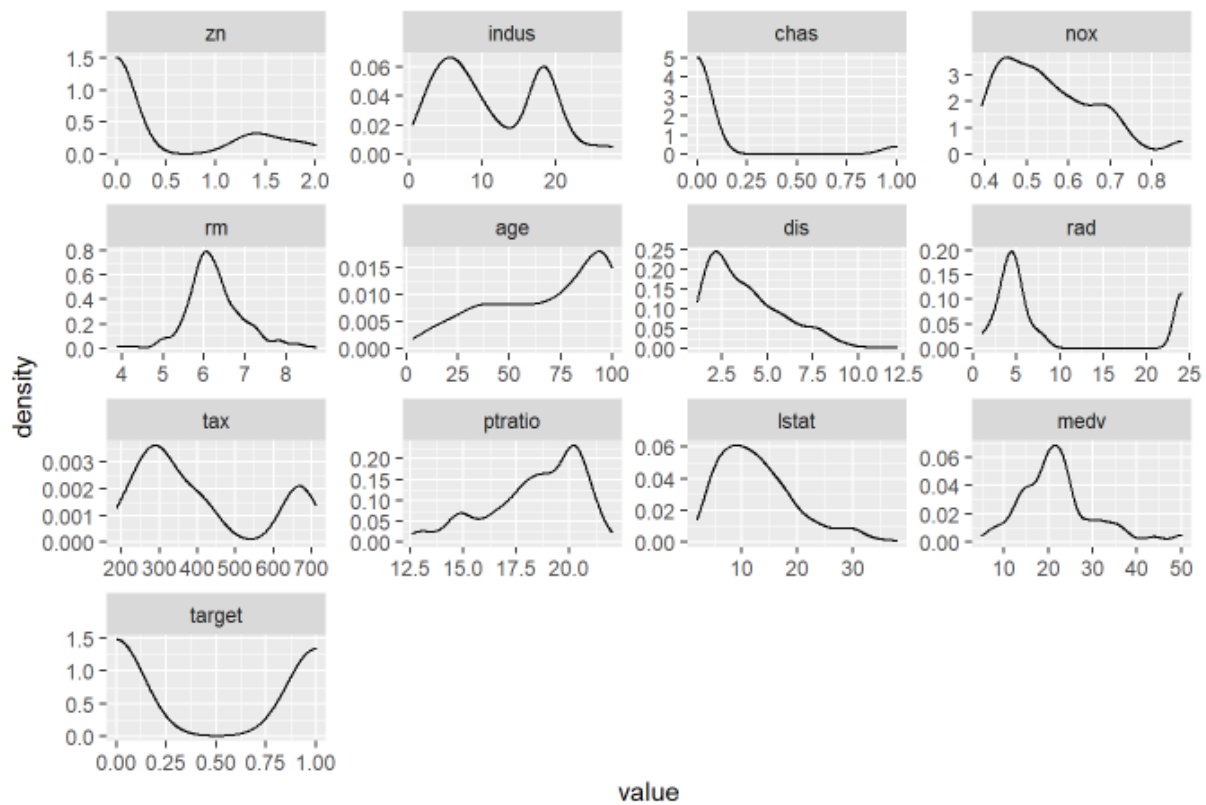
```
##      zn      indus      chas      nox      rm      age
## 2.17681518 0.28854503 3.33548988 0.74632807 0.47932023 -0.57770755
##      dis      rad      tax      ptratio      lstat      medv
## 0.99889262 1.01027875 0.65931363 -0.75426808 0.90558642 1.07669198
##      target
## 0.03422935
```

From statistics, skewness values within the range +2 and -2 are considered acceptable. From the above results, `zn` and `chas` are not symmetric; hence we'll use log transformation to make them symmetric. However, we won't consider transforming `chas` because it is a categorical data.

### Log Transformation

```
##      zn      indus      chas      nox      rm      age      dis
## 1.1954317 0.3708873 3.2567321 0.8108244 0.4843153 -0.5322325 0.9721716
##      rad      tax      ptratio      lstat      medv      target
## 1.1317640 0.7385414 -0.8218609 0.9700911 0.9700927 0.1036391
```

## Log Transformation

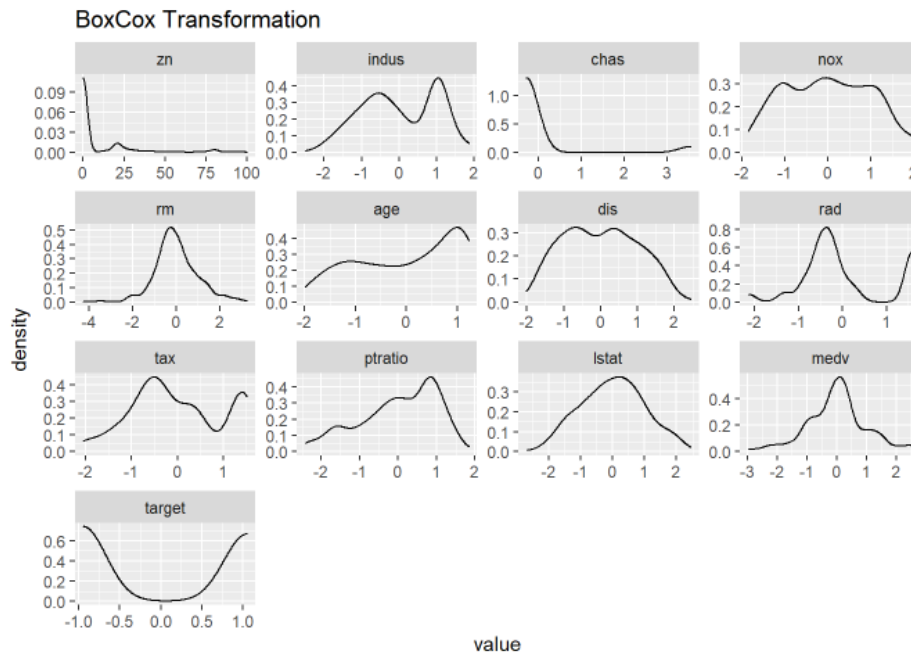


Skewness value of `zn` is now close to 1 hence safer for model building than before. The issue with extreme outliers is also resolved after this transformation.

## BoxCox Transformation

Apply BoxCox transformation on both the train and test datasets to see if it will give better results in terms of accuracy as compared with log transformed data or not.





##	zn	indus	chas	nox	rm	age
##	2.353729370	-0.120195838	3.256732134	0.068418489	0.026658478	-0.366340589
##	dis	rad	tax	ptratio	lstat	medv
##	0.106811243	0.402503618	0.069764777	-0.613098264	-0.002592159	-0.039697233
##	target					
##	0.103639097					

The skewness for `zn` did not improve much but in overall it seems to be symmetrical compared with one for log transformation. These transformed datasets will form separate models for model building. For model building, we'll use backward elimination model and another model that will be result of cumulative variables that have collinearity.

## BUILD MODELS

### Model 1. Backward Elimination on Log Transformed data

Use the log transformed data and remove the least insignificant variables one at a time until the model becomes completely significant. We have already checked the skewness for the variables and adjusted `zn` to `log + 1` which also addressed missing values. We remove `chas`, `lstat`, `rm`, `indus`, `ptratio`, `tax` and `dis` one by one to keep only the significant variables in the model. At the end only `age`, `nox`, `rad` and `medv` have significant impact on target with adjusted R-Square of 0.59. Only `zn` was adjusted to log but since it was removed the model is normal and not transformed.

Observations	327
Dependent variable	target
Type	OLS linear regression

F(4,322)	118.76
R2	0.60
Adj. R2	0.59

	Est.	S.E.	t val.	p
(Intercept)	-1.12	0.13	-8.62	0.00
nox	1.81	0.24	7.50	0.00
age	0.00	0.00	3.90	0.00
rad	0.02	0.00	7.57	0.00
medv	0.01	0.00	2.90	0.00

Standard errors: OLS

### Multicollinearity

Term	VIF	SE_factor
nox	2.760468	1.661466
age	2.344391	1.531141
rad	1.623970	1.274351
medv	1.433166	1.197149

## Model 2. Backward Elimination on BoxCox Transformed data

Using the BoxCox transformed data, we eliminate the insignificant variables (that have highest p-value) one-by-one. The model seems slightly better as compared with Model1:

- R-Square improved from 0.60 to 0.63
- Adjusted R-Square improved from 0.59 to 0.61.
- `dis` variable is significant unlike in Model1.

`lstat`, `rm`, `indus`, `chas`, `ptratio`, `zn` and `tax` are insignificant in this model hence removed.

Observations	327
Dependent variable	target
Type	OLS linear regression

F(5,321)	104.74
R2	0.62
Adj. R2	0.61

	Est.	S.E.	t val.	p
(Intercept)	-0.00	0.03	-0.00	1.00
nox	0.65	0.09	7.61	0.00
age	0.20	0.06	3.37	0.00
dis	0.23	0.08	2.93	0.00
rad	0.33	0.04	7.55	0.00
medv	0.12	0.04	2.75	0.01

Standard errors: OLS

### Multicollinearity

Term	VIF	SE_factor
nox	6.140573	2.478018
age	3.082054	1.755578
dis	5.293362	2.300731
rad	1.592763	1.262047
medv	1.508302	1.228129

### Model 3. Using Stepwise Regression

This method is used to verify the result of Model2 by eliminating all the insignificant variables one at a time under the hood and brings the significant variables.

```
## Start: AIC=-305.39
## target ~ nox + age + dis + rad + medv
##
##      Df Sum of Sq  RSS   AIC
## <none>            123.88 -305.39
## - medv  1      2.9275 126.81 -299.75
## - dis   1      3.3095 127.19 -298.77
## - age   1      4.3841 128.27 -296.02
## - rad   1     22.0257 145.91 -253.88
## - nox   1     22.3745 146.26 -253.10
```

Observations	327
Dependent variable	target
Type	OLS linear regression

F(5,321)	104.74
----------	--------

R2	0.62
----	------

Adj. R2	0.61
---------	------

	Est.	S.E.	t val.	p
(Intercept)	-0.00	0.03	-0.00	1.00
nox	0.65	0.09	7.61	0.00
age	0.20	0.06	3.37	0.00
dis	0.23	0.08	2.93	0.00
rad	0.33	0.04	7.55	0.00
medv	0.12	0.04	2.75	0.01

Standard errors: OLS

## Model 4. Using glmulti

```
##  
## Call:  
## fitfunc(formula = as.formula(x), family = ..1, data = data)  
##  
## Deviance Residuals:  
##      Min       1Q   Median       3Q      Max  
## -1.7621  -0.2870  -0.0080   0.0057   3.2397  
##  
## Coefficients:  
##              Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -36.411735   6.701072  -5.434 5.52e-08 ***  
## zn          -0.052983   0.034824  -1.521  0.12815  
## indus       -0.069985   0.048870  -1.432  0.15213  
## nox         44.712077   8.392153   5.328 9.94e-08 ***  
## age         0.032535   0.012388   2.626  0.00863 **  
## dis         0.749006   0.262842   2.850  0.00438 **  
## rad         0.569547   0.159955   3.561  0.00037 ***  
## tax        -0.005851   0.003072  -1.905  0.05683 .  
## ptratio     0.268150   0.129009   2.079  0.03766 *  
## medv        0.089608   0.039255   2.283  0.02245 *  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## (Dispersion parameter for binomial family taken to be 1)  
##  
##      Null deviance: 452.43  on 326  degrees of freedom  
## Residual deviance: 152.03  on 317  degrees of freedom  
## AIC: 172.03  
##  
## Number of Fisher Scoring iterations: 8
```

The `glmulti()` function optimizes the best performing model through simulating all possible models under the hood and finds the best performing model. It takes less time to optimize the model though.

## SELECT MODELS

### Model Performance

Lets select the best model using `compare_performance` and `model_performance` functions from performance package. It calculates AIC, BIC, R2 & adjusted r-sq, RMSE, BF and Performance\_Score. From the first three models, model1 is performs best since the values of AIC and BIC are lower. RMSE is also lower as compared with model2 and model3.

Looking at Model4, AIC and BIC values are lower than in model1. R2 has also increased to 0.71 but RMSE has increased slightly. Since RMSE values in all models are very low so I won't be base performance on RMSE.

In terms of AIC, BIC and R2, Model4 is the best performing model and hence I will select Model4.

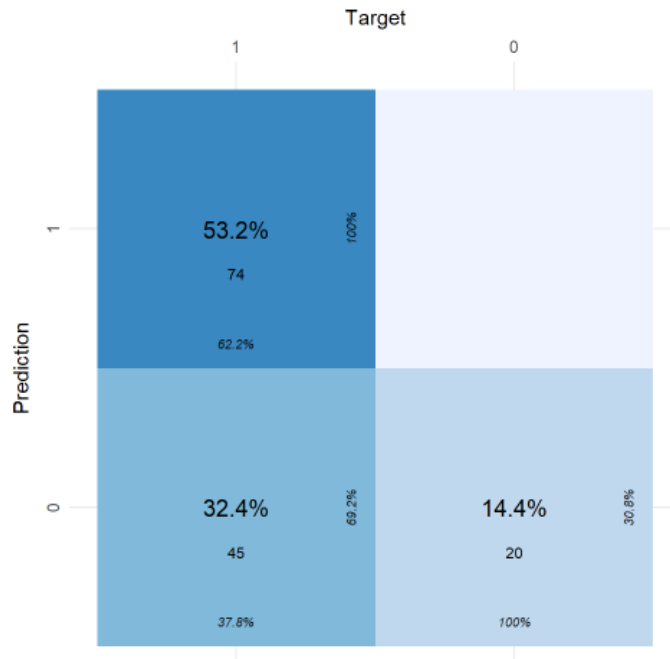
Name	Model	R2	R2_adjusted	RMSE	Sigma	AIC_wt	BIC_wt	Performance_Score
model1	lm	0.5960003	0.5909817	0.3173751	0.3198297	1	1	0.6666667
model2	lm	0.6199863	0.6140671	0.6155092	0.6212350	0	0	0.3333333
model3	lm	0.6199863	0.6140671	0.6155092	0.6212350	0	0	0.3333333

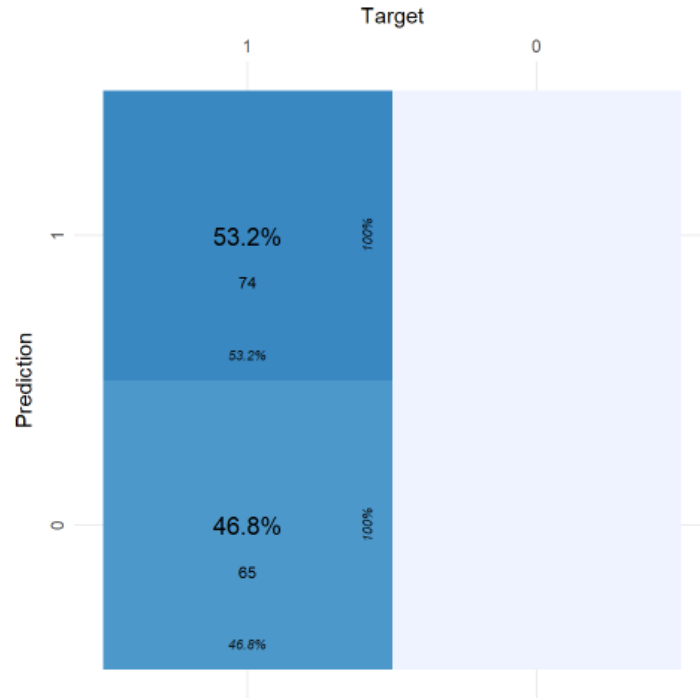
AIC	BIC	R2_Tjur	RMSE	Sigma	Log_loss	Score_log	Score_spherical	PCP
172.0349	209.9345	0.7092821	0.2638491	0.6925355	0.2324693	-Inf	0.024527	0.8550339

### Prediction Accuracy

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##           0 20  0
##           1 45 74
##
##           Accuracy : 0.6763
##           95% CI : (0.5917, 0.7531)
##           No Information Rate : 0.5324
##           P-Value [Acc > NIR] : 0.0003971
##
##           Kappa : 0.3212
##
##           Mcnemar's Test P-Value : 5.412e-11
##
##           Sensitivity : 0.3077
##           Specificity : 1.0000
##           Pos Pred Value : 1.0000
##           Neg Pred Value : 0.6218
##           Precision : 1.0000
##           Recall : 0.3077
##           F1 : 0.4706
##           Prevalence : 0.4676
##           Detection Rate : 0.1439
##           Detection Prevalence : 0.1439
##           Balanced Accuracy : 0.6538
##
##           'Positive' Class : 0
##
```

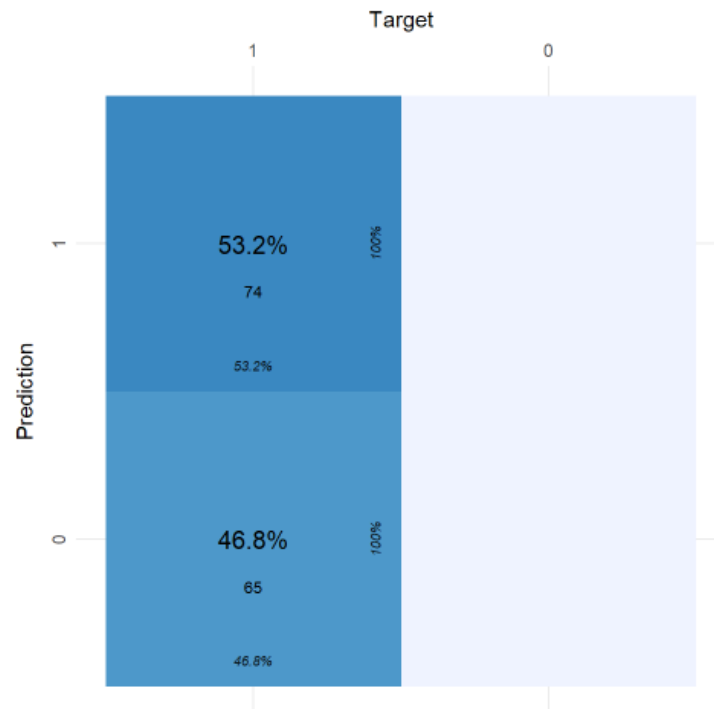


```
## Confusion Matrix and Statistics
##
##      Reference
## Prediction  0  1
##      0  0  0
##      1 65 74
##
##      Accuracy : 0.5324
##      95% CI : (0.4459, 0.6174)
##      No Information Rate : 0.5324
##      P-Value [Acc > NIR] : 0.5346
##
##      Kappa : 0
##
##      Mcnemar's Test P-Value : 2.051e-15
##
##      Sensitivity : 0.0000
##      Specificity : 1.0000
##      Pos Pred Value :  NaN
##      Neg Pred Value : 0.5324
##      Precision :    NA
##      Recall : 0.0000
##      F1 :    NA
##      Prevalence : 0.4676
##      Detection Rate : 0.0000
##      Detection Prevalence : 0.0000
##      Balanced Accuracy : 0.5000
##
##      'Positive' Class : 0
##
```



```
## Confusion Matrix and Statistics
##
##      Reference
## Prediction  0  1
##      0  0  0
##      1 65 74
##
##      Accuracy : 0.5324
##      95% CI : (0.4459, 0.6174)
##      No Information Rate : 0.5324
##      P-Value [Acc > NIR] : 0.5346
##
##      Kappa : 0
##
##  Mcnemar's Test P-Value : 2.051e-15
##
##      Sensitivity : 0.0000
##      Specificity : 1.0000
##      Pos Pred Value :  NaN
##      Neg Pred Value : 0.5324
##      Precision :  NA
##      Recall : 0.0000
##      F1 :  NA
##      Prevalence : 0.4676
##      Detection Rate : 0.0000
##      Detection Prevalence : 0.0000
##      Balanced Accuracy : 0.5000
##
##      'Positive' Class : 0
##
```





```
## Confusion Matrix and Statistics
##
##      Reference
## Prediction  0  1
##      0  60  5
##      1   5 69
##
##      Accuracy : 0.9281
##      95% CI : (0.8717, 0.965)
##      No Information Rate : 0.5324
##      P-Value [Acc > NIR] : <2e-16
##
##      Kappa : 0.8555
##
##  Mcnemar's Test P-Value : 1
##
##      Sensitivity : 0.9231
##      Specificity : 0.9324
##      Pos Pred Value : 0.9231
##      Neg Pred Value : 0.9324
##      Precision : 0.9231
##      Recall : 0.9231
##      F1 : 0.9231
##      Prevalence : 0.4676
##      Detection Rate : 0.4317
##      Detection Prevalence : 0.4676
##      Balanced Accuracy : 0.9278
##
##      'Positive' Class : 0
##
```



Model 1 has accuracy of 67.63% as compared to Model 2 and 3 which have 53.24%. Model4 has accuracy of 92.81% which was achieved using `glmulti()` function.

Looking at F1 scores there are no F1 values in Model 2 and model 3. This implies these models did not have precision and hence it did not identify anything positively hence are poor models.

We are now left with Model 1 and Model 4. Model 1 has F1 value of 0.47 with precision is 1 and recall is 0.3. Model 4 has the highest Precision, recall and F1 values.

Basing on overall RMSE, AIC, BIC, F1, Precision and Recall values Model 4 is the best out of all the models.

### Prediction the evaluation data set

Let's use the selected model (Model4) to predict the test set.

```
##      zn indus chas   nox    rm  age    dis rad tax ptratio lstat medv target
## 1  0   7.07    0 0.469 7.185 61.1 4.9671 2 242   17.8  4.03 34.7      0
## 2  0   8.14    0 0.538 6.096 84.5 4.4619 4 307   21.0 10.26 18.2      1
## 3  0   8.14    0 0.538 6.495 94.4 4.4547 4 307   21.0 12.80 18.4      1
## 4  0   8.14    0 0.538 5.950 82.0 3.9900 4 307   21.0 27.71 13.2      0
## 5  0   5.96    0 0.499 5.850 41.5 3.9342 5 279   19.2  8.77 21.0      0
## 6 25   5.13    0 0.453 5.741 66.2 7.2254 8 284   19.7 13.15 18.7      0
```

## References

- A Modern Approach to Regression with R: Simon Sheather
- Linear Models with R: Julian Faraway.
- [Detecting Multicollinearity with VIF](#)

## Appendix

### Description of the variables

- zn: proportion of residential land zoned for large lots (over 25000 square feet) (predictor variable)
- indus: proportion of non-retail business acres per suburb (predictor variable)
- chas: a dummy var. for whether the suburb borders the Charles River (1) or not (0) (predictor variable)
- nox: nitrogen oxides concentration (parts per 10 million) (predictor variable)
- rm: average number of rooms per dwelling (predictor variable)
- age: proportion of owner-occupied units built prior to 1940 (predictor variable)
- dis: weighted mean of distances to five Boston employment centers (predictor variable)
- rad: index of accessibility to radial highways (predictor variable)
- tax: full-value property-tax rate per \$10,000 (predictor variable)
- ptratio: pupil-teacher ratio by town (predictor variable)
- black:  $1000(B_k - 0.63)^2$  where  $B_k$  is the proportion of blacks by town (predictor variable)
- lstat: lower status of the population (percent) (predictor variable)
- medv: median value of owner-occupied homes in \$1000s (predictor variable)
- target: whether the crime rate is above the median crime rate (1) or not (0) (response variable)

### R Code

GitHub: <https://github.com/nathtrish334/Data-621/blob/main/HW3/Homework03.rmd>

RPubs: <https://rpubs.com/trishitanath/833136>