Critical Thinking Group 4: DATA621 Homework 3

Table of Contents

TEAM Members: 2

Overview 2

Deliverables 3

Data Exploration 3

Missing Values & Data Type Check 4

Data Statistics Summary 7

Consolidated Data Dictionary 17

Data Preparation 20

Re-scale Data 20

Build Models 22

Model 1: Full Model 22

Model 2: Removing Predictors Seemed Unnecessary 23

Model 3: Removing Highest VIF Values 24

Model 4: Removing Poor Predictors 25

Model 5: Stepwise Based on AIC 26

Model 6: Stepwise Based on BIC 33

Model 7: Best Subset Based on AIC 37

Model 8: Best Subset Based on BIC 38

Select Models 40

Fourfold Plots 41

Summary Statistics 42

ROC / AUC 44

R^2, AIC, AICc & BIC 45

Model Selection 47

Odds Ratio 49

Make Predictions 50

Appendix 53

TEAM Members:

Rajwant Mishra
Priya Shaji
Debabrata Kabiraj
Isabel Ramesar
Sin Ying Wong
Fan Xu

Overview

In this homework assignment, you will explore, analyze and model a dataset containing information on crime for various neighborhoods of a major city. Each record has a response variable indicating whether or not the crime rate is above the median crime rate (1) or not (0).

Your objective is to build a binary logistic regression model on the training dataset to predict whether the neighborhood will be at risk for high crime levels. You will provide classifications and probabilities for the evaluation dataset using your binary logistic regression model. You can only use the variables given to you (or variables that you derive from the variables provided). Below is a short description of the variables of interest in the dataset:

Variable Name	Definition	Variable Type
zn	proportion of residential land zoned for large lots (over 25000 square feet)	predictor
indus	proportion of non-retail business acres per suburb	predictor
chas	a dummy var. for whether the suburb borders the Charles River (1) or not (0)	predictor
nox	nitrogen oxides concentration (parts per 10 million)	predictor
rm	average number of rooms per dwelling	predictor
age	proportion of owner-occupied units built prior to 1940	predictor
dis	weighted mean of distances to five Boston employment centers	predictor
rad	index of accessibility to radial highways	predictor
tax	full-value property-tax rate per $10,000	predictor

Variable Name	Definition	Variable Type
ptratio	pupil-teacher ratio by town	predictor
black	1000(Bk - 0.63)2 where Bk is the proportion of blacks by town	predictor
lstat	lower status of the population (percent)	predictor
medv	median value of owner-occupied homes in $1000s	predictor
target	whether the crime rate is above the median crime rate (1) or not (0)	response

Deliverables

A write-up of your solutions submitted in PDF format. Assigned prediction (probabilities, classifications) for the evaluation dataset. Use 0.5 threshold.

Data Exploration

We have two datasets. One is the `training set`, which includes 12 candidate predictors, 1 response variable, and 466 observations. The other one is the `evaluation set`, which includes 12 candidate predictors only, 40 observations.

We are going to study their missing values, data types and data statistics.

Hide

```
data_t <- read_csv("https://raw.githubusercontent.com/Rajwantmishra/DATA621_CR4/master/HW3/crime
  dplyr::select(target, everything())

data_e <- read_csv('https://raw.githubusercontent.com/Rajwantmishra/DATA621_CR4/master/HW3/crime
```

Missing Values & Data Type Check

In the `training set`, there are 12 candidate predictors and 1 response variable with 466 observations. In the `evaluation set`, there are 12 candidate predictors with 40 observations. Both datasets have no missing values (eg: NA, NULL or ''). However, the variable `black`, which is described in the overview section, is not presented in both datasets.

Among the 12 candidate predictors, 1 is categorical (`chas`), the other 11 are continuous numerical. The response variable `target` is categorical.

Hide

```
glimpse(data_t)
```

```
## Observations: 466
## Variables: 13
## $ target  <dbl> 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1,...
## $ zn      <dbl> 0, 0, 0, 30, 0, 0, 0, 0, 0, 80, 22, 0, 0, 22, 0, 0, 100, 20...
## $ indus   <dbl> 19.58, 19.58, 18.10, 4.93, 2.46, 8.56, 18.10, 18.10, 5.19, ...
## $ chas    <dbl> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ nox     <dbl> 0.605, 0.871, 0.740, 0.428, 0.488, 0.520, 0.693, 0.693, 0.5...
## $ rm      <dbl> 7.929, 5.403, 6.485, 6.393, 7.155, 6.781, 5.453, 4.519, 6.3...
## $ age     <dbl> 96.2, 100.0, 100.0, 7.8, 92.2, 71.3, 100.0, 100.0, 38.1, 19...
## $ dis     <dbl> 2.0459, 1.3216, 1.9784, 7.0355, 2.7006, 2.8561, 1.4896, 1.6...
## $ rad     <dbl> 5, 5, 24, 6, 3, 5, 24, 24, 5, 1, 7, 5, 24, 7, 3, 3, 5, 5, 2...
## $ tax     <dbl> 403, 403, 666, 300, 193, 384, 666, 666, 224, 315, 330, 398,...
## $ ptratio <dbl> 14.7, 14.7, 20.2, 16.6, 17.8, 20.9, 20.2, 20.2, 20.2, 16.4,...
## $ lstat   <dbl> 3.70, 26.82, 18.85, 5.19, 4.82, 7.67, 30.59, 36.98, 5.68, 9...
## $ medv    <dbl> 50.0, 13.4, 15.4, 23.7, 37.9, 26.5, 5.0, 7.0, 22.2, 20.9, 2...
```

Hide

```
glimpse(data_e)
```

```
## Observations: 40
## Variables: 12
## $ zn      <dbl> 0, 0, 0, 0, 0, 25, 25, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 22,...
## $ indus   <dbl> 7.07, 8.14, 8.14, 8.14, 5.96, 5.13, 5.13, 4.49, 4.49, 2.89,...
## $ chas    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,...
## $ nox     <dbl> 0.469, 0.538, 0.538, 0.538, 0.499, 0.453, 0.453, 0.449, 0.4...
## $ rm      <dbl> 7.185, 6.096, 6.495, 5.950, 5.850, 5.741, 5.966, 6.630, 6.1...
## $ age     <dbl> 61.1, 84.5, 94.4, 82.0, 41.5, 66.2, 93.4, 56.1, 56.8, 69.6,...
## $ dis     <dbl> 4.9671, 4.4619, 4.4547, 3.9900, 3.9342, 7.2254, 6.8185, 4.4...
## $ rad     <dbl> 2, 4, 4, 4, 5, 8, 8, 3, 3, 2, 2, 2, 4, 5, 5, 4, 8, 8, 7, 1,...
## $ tax     <dbl> 242, 307, 307, 307, 279, 284, 284, 247, 247, 276, 188, 188,...
## $ ptratio <dbl> 17.8, 21.0, 21.0, 21.0, 19.2, 19.7, 19.7, 18.5, 18.5, 18.0,...
## $ lstat   <dbl> 4.03, 10.26, 12.80, 27.71, 8.77, 13.15, 14.44, 6.53, 8.44, ...
## $ medv    <dbl> 34.7, 18.2, 18.4, 13.2, 21.0, 18.7, 16.0, 26.6, 22.2, 21.4,...
```

Hide

```
library(gridExtra)
p_t_dt <- vis_dat(data_t)
p_t_m <- vis_miss(data_t)
p_e_dt <- vis_dat(data_e)
p_e_m <- vis_miss(data_e)
grid.arrange(p_t_m,p_e_m, p_t_dt,p_e_dt,ncol = 2,
             widths = c(1,1),
             heights = c(1.5,1),
             top = 'Missing Values & Data Type Check
             Training Set                                                  Evaluation Set')
```

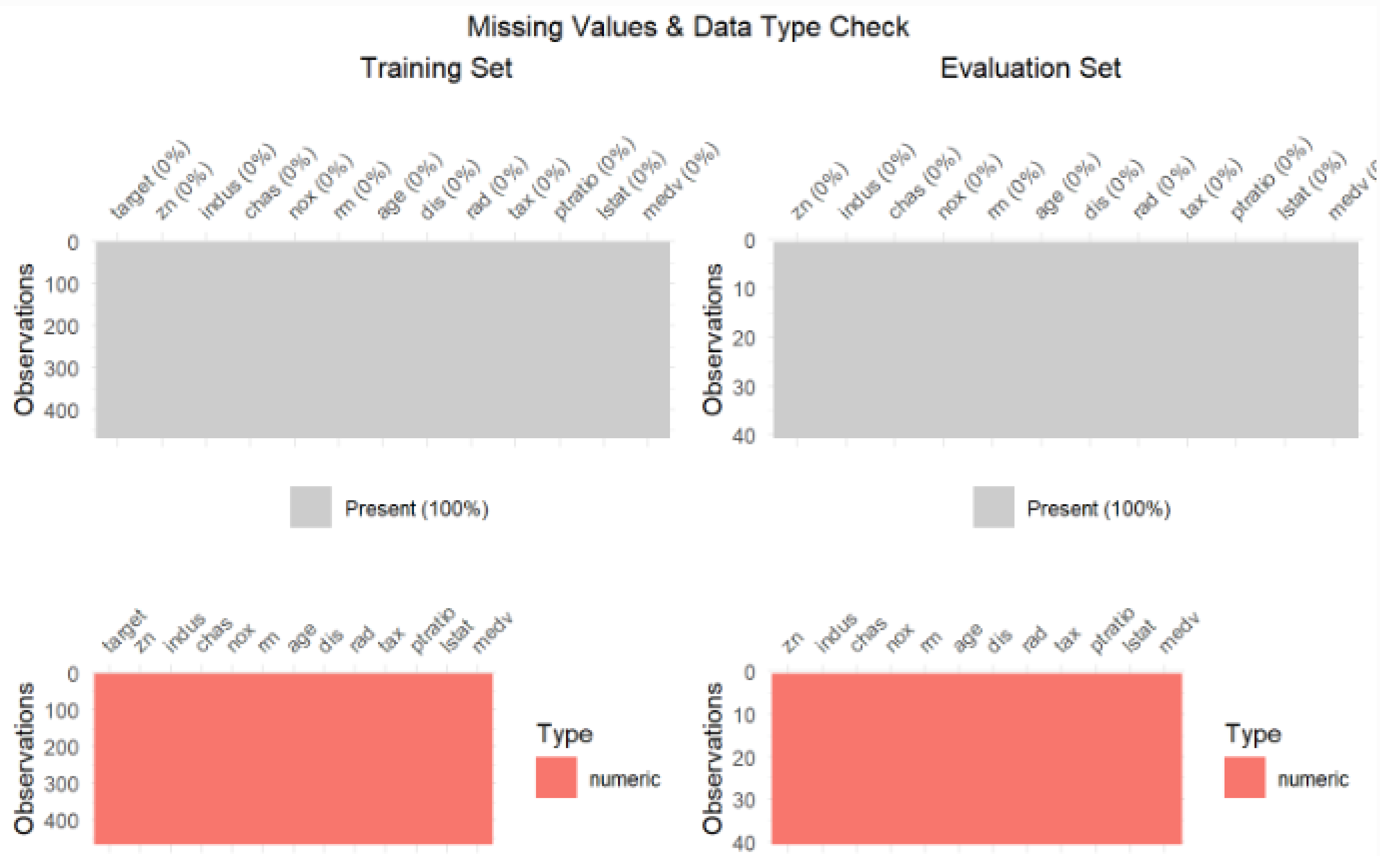


Figure 1: Missing Values & Data Type Check

Below is the summary of the datasets and some inference of it.

1. It seems there are no Null values in the predictor and response variables.
2. Each variables are in different scale.
3. Categorical variables are `chas` and `target`.
4. There are a total of 466 observations and 12 predictor variables and 1 response variable.

Data Statistics Summary

A binary logistic regression model is built using the `training set`, therefore the `training set` is used for the following data exploration.

The data types in the raw dataset are all 'doubles', however the candidate predictor `chas` and the response variable `target` are categorical, therefore, we update the data types of these two variables to 'factors'.

Hide

```
data_t_mod <- data_t %>%
  mutate(chas = as.factor(chas),
         target = as.factor(target)) %>%
  dplyr::select(target, everything())
DT::datatable(data_t_mod)
```

Show 10 entries Search:

	target	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv
1	1	0	19.58	0	0.605	7.929	96.2	2.0459	5	403	14.7	3.7	50
2	1	0	19.58	1	0.871	5.403	100	1.3216	5	403	14.7	26.82	13.4
3	1	0	18.1	0	0.74	6.485	100	1.9784	24	666	20.2	18.85	15.4
4	0	30	4.93	0	0.428	6.393	7.8	7.0355	6	300	16.6	5.19	23.7
5	0	0	2.46	0	0.488	7.155	92.2	2.7006	3	193	17.8	4.82	37.9
6	0	0	8.56	0	0.52	6.781	71.3	2.8561	5	384	20.9	7.67	26.5
7	1	0	18.1	0	0.693	5.453	100	1.4896	24	666	20.2	30.59	5
8	1	0	18.1	0	0.693	4.519	100	1.6582	24	666	20.2	36.98	7
9	0	0	5.19	0	0.515	6.316	38.1	6.4584	5	224	20.2	5.68	22.2
10	0	80	3.64	0	0.392	5.876	19.1	9.2203	1	315	16.4	9.25	20.9

Showing 1 to 10 of 466 entries Previous 1 2 3 4 5 … 47 Next

Table 3: `training set`

The statistics of all variables are list below:

Hide

```
summary(data_t_mod)
```

```
##  target        zn               indus          chas         nox        
##  0:237   Min.   :  0.00   Min.   : 0.460   0:433   Min.   :0.3890  
##  1:229   1st Qu.:  0.00   1st Qu.: 5.145   1: 33   1st Qu.:0.4480  
##          Median :  0.00   Median : 9.690           Median :0.5380  
##          Mean   : 11.58   Mean   :11.105           Mean   :0.5543  
##          3rd Qu.: 16.25   3rd Qu.:18.100           3rd Qu.:0.6240  
##          Max.   :100.00   Max.   :27.740           Max.   :0.8710  
##        rm             age              dis              rad       
##  Min.   :3.863   Min.   :  2.90   Min.   : 1.130   Min.   : 1.00  
##  1st Qu.:5.887   1st Qu.: 43.88   1st Qu.: 2.101   1st Qu.: 4.00  
##  Median :6.210   Median : 77.15   Median : 3.191   Median : 5.00  
##  Mean   :6.291   Mean   : 68.37   Mean   : 3.796   Mean   : 9.53  
##  3rd Qu.:6.630   3rd Qu.: 94.10   3rd Qu.: 5.215   3rd Qu.:24.00  
##  Max.   :8.780   Max.   :100.00   Max.   :12.127   Max.   :24.00  
##       tax           ptratio         lstat             medv      
##  Min.   :187.0   Min.   :12.6   Min.   : 1.730   Min.   : 5.00  
##  1st Qu.:281.0   1st Qu.:16.9   1st Qu.: 7.043   1st Qu.:17.02  
##  Median :334.5   Median :18.9   Median :11.350   Median :21.20  
##  Mean   :409.5   Mean   :18.4   Mean   :12.631   Mean   :22.59  
##  3rd Qu.:666.0   3rd Qu.:20.2   3rd Qu.:16.930   3rd Qu.:25.00  
##  Max.   :711.0   Max.   :22.0   Max.   :37.970   Max.   :50.00  
```

The box plot below shows that outliners exist in variables `zn`, `rm`, `dis`, `istat`, `medv`. We use scaled training set to draw the box plot to show the corresponding outliers by ratio.

Hide

```
data_t %>%
  scale() %>%
  as.data.frame() %>%
  stack() %>%
  ggplot(aes(x = ind, y = values)) +
  geom_boxplot(fill = 'deeppink4') +
  labs(title = 'Boxplot: Scaled Training Set',
       x = 'Variables',
       y = 'Normalized_Values')+
  theme(panel.background = element_rect(fill = 'grey'))
```

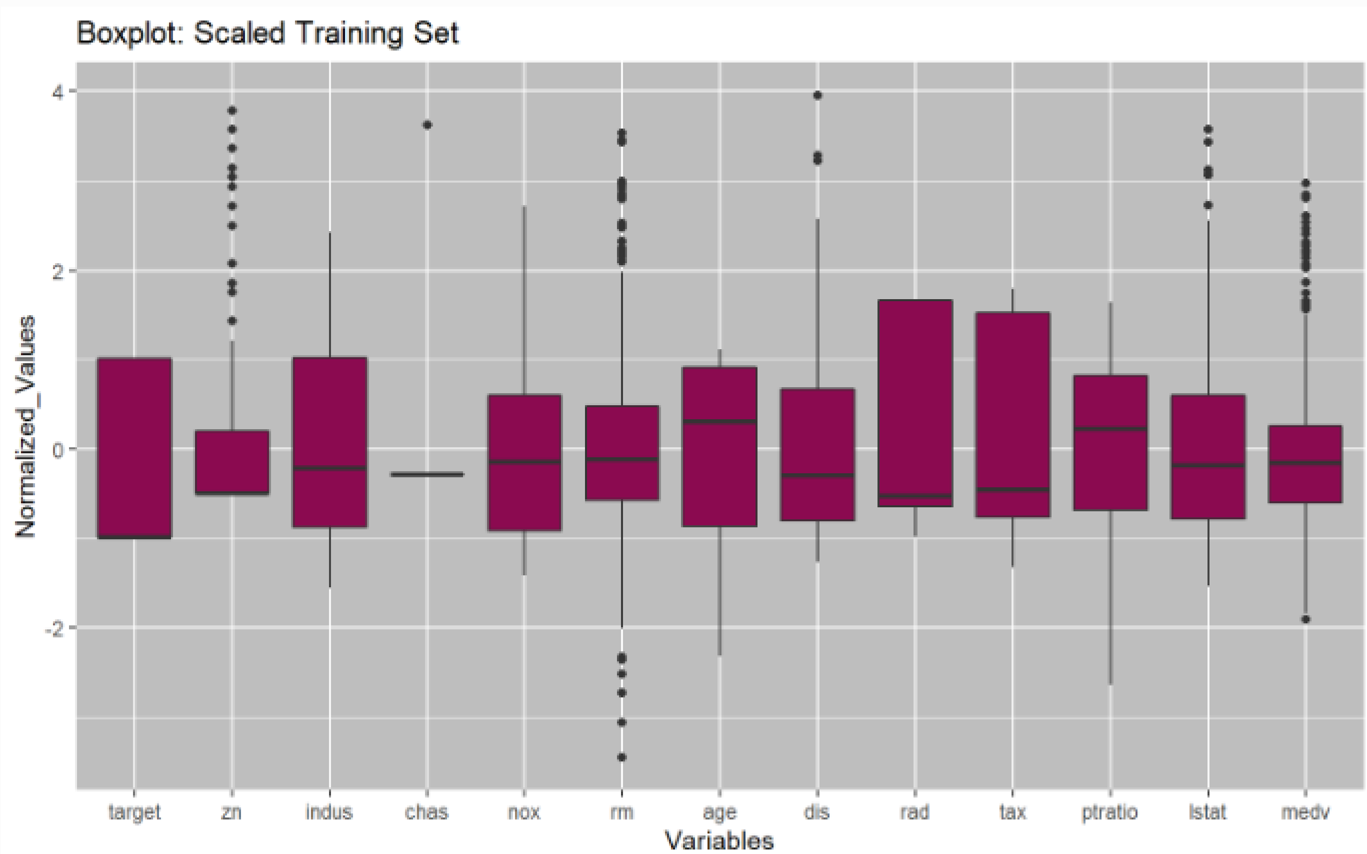


Figure 2: Boxplot: Scaled Training Set

The scaled histogram and density plot show that variables `zn`,`nox`, `dis`, `lstat`, `medv` are right skewed; `age`, `ptratio` are left skewed; `rad`, `tax` are bimodal; `target`, `chas` are categorical however `target` is close to unbias while `chas` is highly biased; the rest are close to normal.

Hide

```
data_t %>%
  scale() %>%
  as.data.frame() %>%
  stack() %>%
  ggplot(aes(x = values)) +
  geom_histogram(fill = 'deeppink4', color = 'black') +
  labs(title = 'Histogram: Training Set')+
  theme(panel.background = element_rect(fill = 'grey'))+
  facet_wrap(~ind, scale='free', ncol = 4)
```

Figure 3: Histogram: Training Set

Hide

```
data_t %>%
  select_if(is.numeric) %>%
  keep(is.numeric) %>%                    # Keep only numeric columns
  gather() %>%                            # Convert to key-value pairs
  ggplot(aes(x=value)) +                  # Plot the values
    facet_wrap(~key, scales = "free") +   # In separate panels
    geom_density()
```

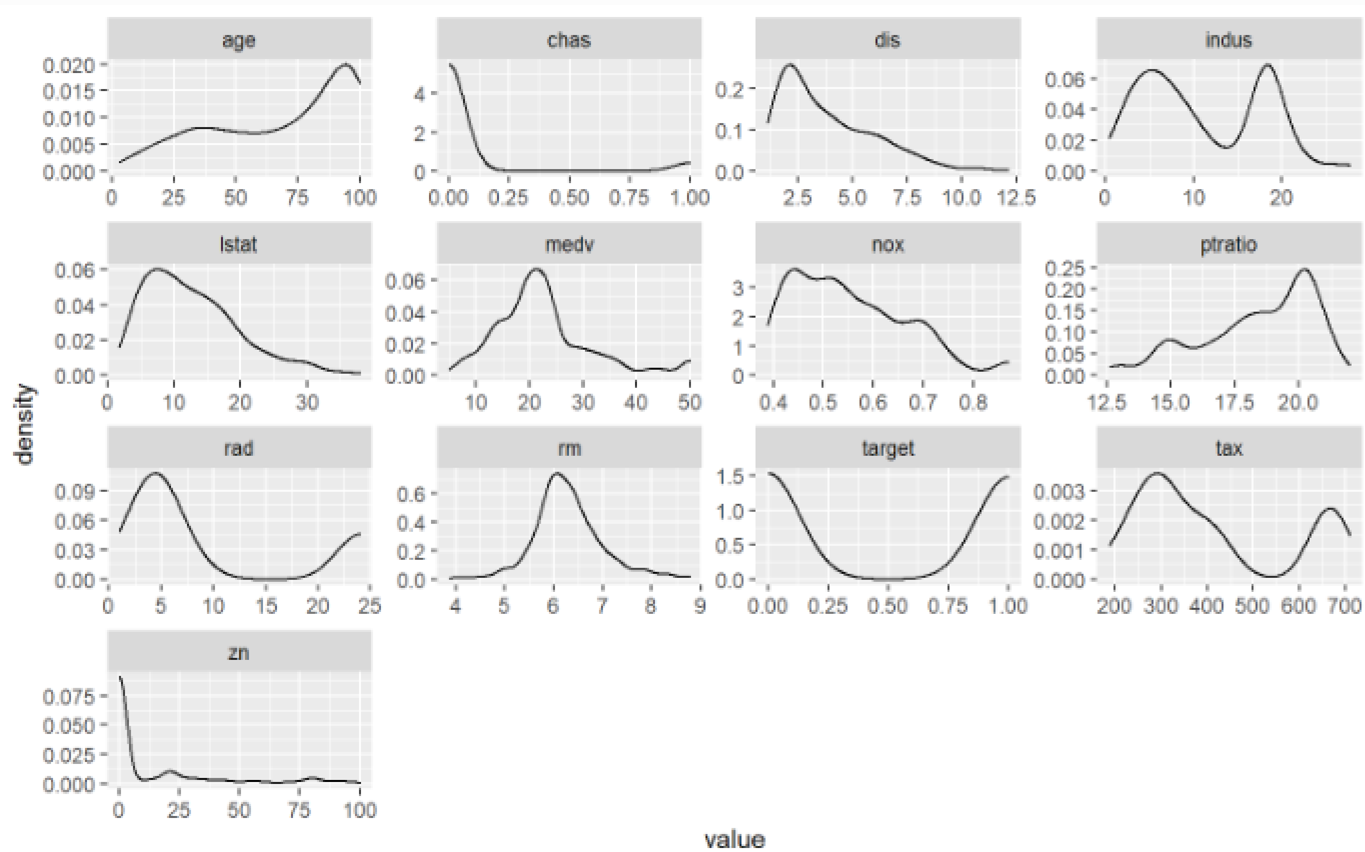


Figure 4: Density Plot: Training Set

The correlation matrix below shows that the response variable `target` has strong positive relationship (>=0.6) with variables `rad`,`tax`,`age`,`indus`,`nox`, and strong negative relationship (<=-0.6) with variable `dis`.

Meanwhile, it worths notice that some pairs of candidate predictors have strong correlationship, such as `rad` and `tax` (0.92), `indus` and `nox` (0.76), `nox` and `dis` (-0.77), etc.

Hide

```
data_t %>%
  cor() %>%
  #corrplot(method = "square", type = "upper", order = 'hclust', tl.col = "black", diag = FALSE, bg= 'whit
  corrplot.mixed(upper = 'pie', lower = 'number', order = 'hclust', tl.col = "black")
```

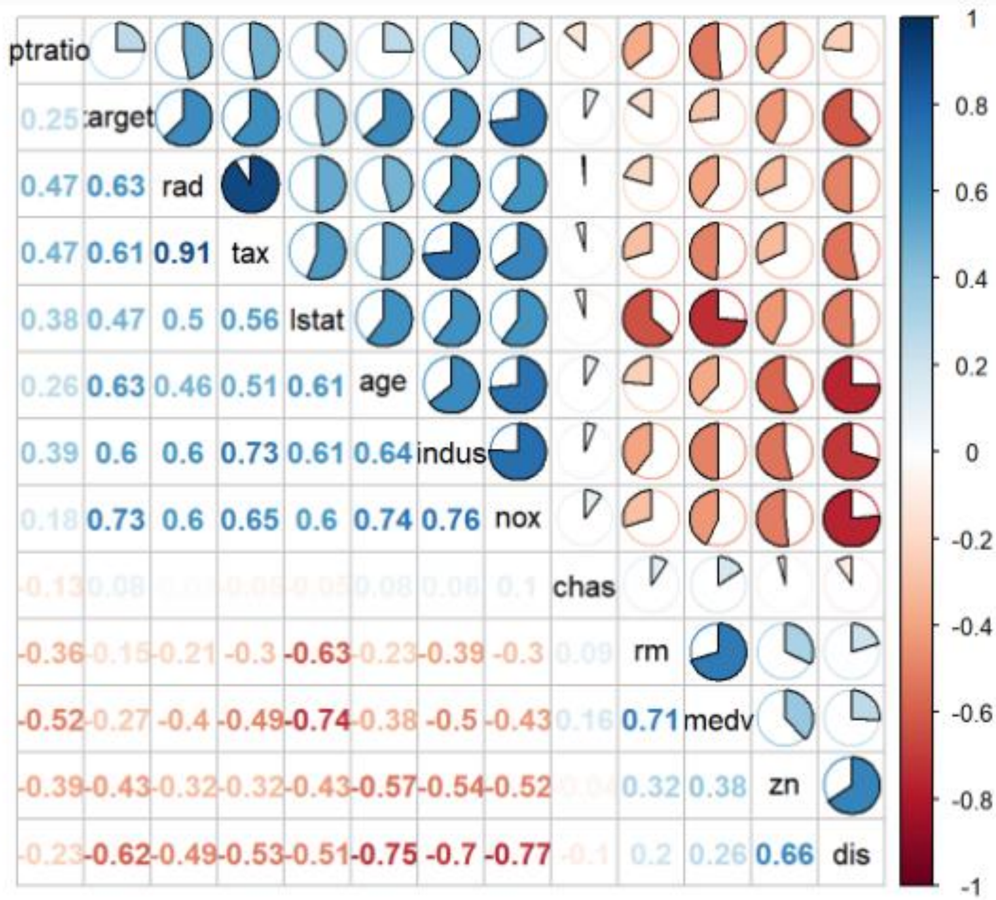


Figure 5: Correlation Pie Chart: Training Set

We implement a correlation matrix to better understand the correlation between variables in the dataset. The below matrix is the results and we noticed a few interesting correlations.

- `nox` : High nitrogen oxides concentration (parts per 10 million) ("nox") is positively correlated with higher than median crime rates. As defined by the EPA - "NOx pollution is emitted by automobiles, trucks and various non-road vehicles (e.g., construction equipment, boats, etc.) as well as industrial sources such as power plants, industrial boilers, cement kilns, and turbines". It is clear to see that nox is concentrated in areas of high road traffic and possible high industrial use which would be neighborhoods of low value and may attract crime.
- `dis`: The weighted mean of distances is negatively correlated with a city with higher than median crime rate. This is intuitive in that employment centers would be more closely located in cities of high crime due to high unemployment being positively correlated with higher crimes rates.
- `tax` : It is also counterintuitive how the crime rate has a positive correlation with the property tax. It would be anticipated that if the property tax increases, the crime rate would decrease due to the money that home occupants and owners would spend on "promised" security systems. However, when the crime rate starts to increase, the housing prices would decrease due to the fact that the home occupants and owners would not want to risk their safety.

Hide

```
PerformanceAnalytics::chart.Correlation(data_t, histogram=TRUE, pch=19)
```

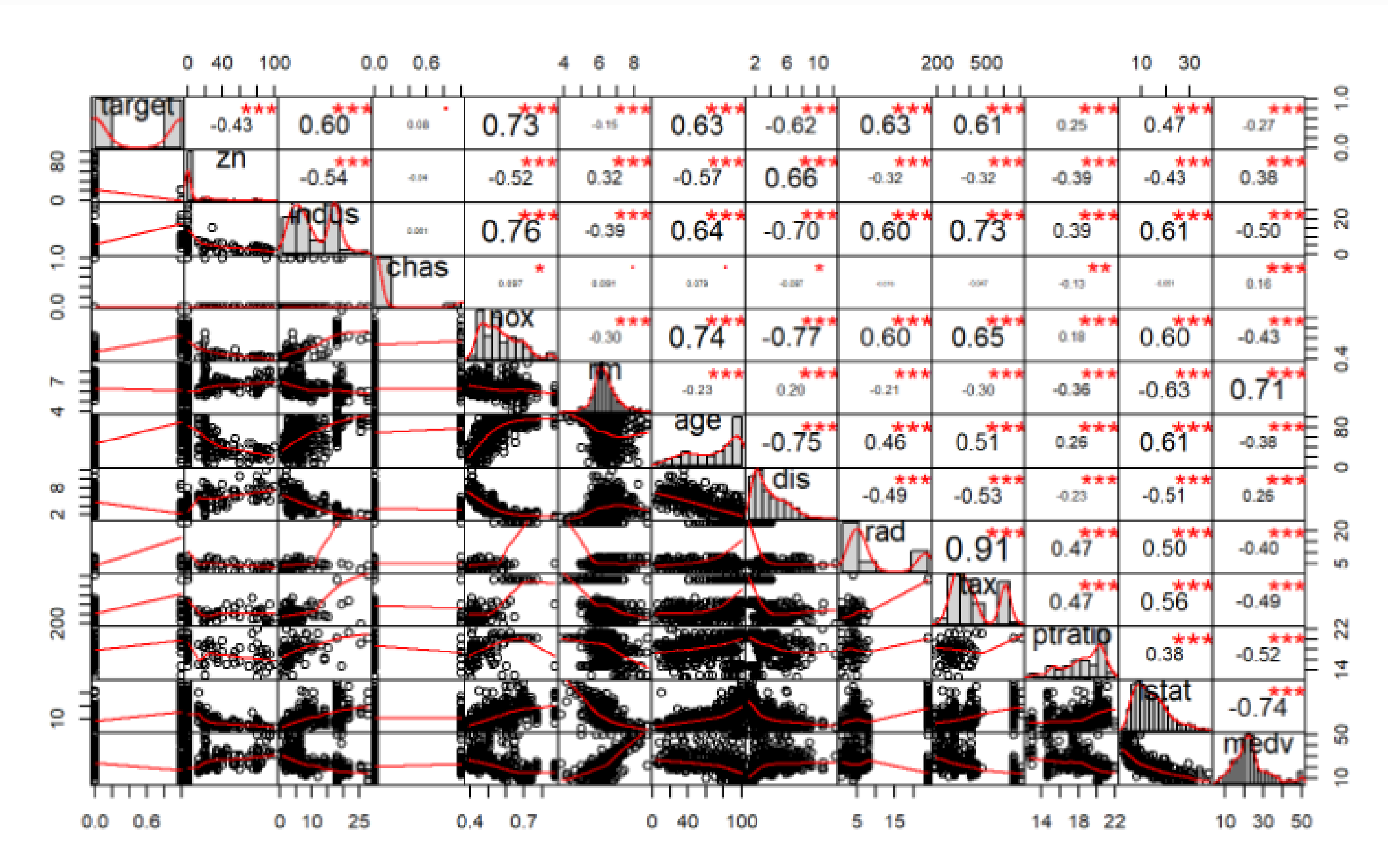


Figure 6: Correlation Chart: Training Set

Hide

```
data_t %>%
  cor() %>%
  as.data.frame() %>%
  rownames_to_column('Variable') %>%
  dplyr::rename(Correlation_vs_Response = target)
```

Variable <chr>	Correlation_vs_Response <dbl>	zn <dbl>	indus <dbl>	chas <dbl>	nox <dbl>
target	1.00000000	-0.43168176	0.60485074	0.08004187	0.72610622
zn	-0.43168176	1.00000000	-0.53826643	-0.04016203	-0.51704518
indus	0.60485074	-0.53826643	1.00000000	0.06118317	0.75963008
chas	0.08004187	-0.04016203	0.06118317	1.00000000	0.09745577
nox	0.72610622	-0.51704518	0.75963008	0.09745577	1.00000000
rm	-0.15255334	0.31981410	-0.39271181	0.09050979	-0.29548972
age	0.63010625	-0.57258054	0.63958182	0.07888366	0.73512782
dis	-0.61867312	0.66012434	-0.70361886	-0.09657711	-0.76888404
rad	0.62810492	-0.31548119	0.60062839	-0.01590037	0.59582984
tax	0.61111331	-0.31928408	0.73222922	-0.04676476	0.65387804

1-10 of 13 rows | 1-6 of 14 columns Previous 1 2 Next

Consolidated Data Dictionary

As a summary of the data exploration process, a data dictionary is created below:

Hide

```
data_stat <- data_t %>%
  dplyr::select(-target,-chas) %>%
  gather() %>%
  group_by(key) %>%
  summarise(Mean = mean(value),
            Median = median(value),
            Max = max(value),
            Min = min(value),
            SD = sd(value))

data_cor <- data_t %>%
  cor() %>%
  as.data.frame() %>%
  dplyr::select(target) %>%
  rownames_to_column('Variable') %>%
  dplyr::rename(Correlation_vs_Response = target)

data_t %>%
  gather() %>%
  dplyr::select(key) %>%
  unique() %>%
  dplyr::rename(Variable = key) %>%
  mutate(Description = c('whether the crime rate is above the median crime rate (1) or not (0)',
                         'proportion of residential land zoned for large lots (over 25000 square
                         'proportion of non-retail business acres per suburb',
                         'a dummy var. for whether the suburb borders the Charles River (1) or r
                         'nitrogen oxides concentration (parts per 10 million)',
                         'average number of rooms per dwelling',
                         'proportion of owner-occupied units built prior to 1940',
                         'weighted mean of distances to five Boston employment centers',
                         'index of accessibility to radial highways',
                         'full-value property-tax rate per $10,000',
                         'pupil-teacher ratio by town',
                         'lower status of the population (percent)',
                         'median value of owner-occupied homes in $1000s'),
```

```
        Var_Type_1 = case_when(Variable %in% c('target','chas') ~ 'categorical',
                               Variable %in% c('rad','tax') ~ 'discrete numerical',
                               TRUE ~ 'continuous numerical'),
        Var_Type_2 = if_else(Variable == 'target', 'response', 'predictor'),
        Missing_Value = 'No') %>%
  left_join(data_stat, by = c('Variable'='key')) %>%
  left_join(data_cor, by = 'Variable') %>%
  mutate_if(is.numeric,round,2) %>%
  kable() %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"),full_width
```

Variable	Description	Var_Type_1	Var_Type_2	Missing_Value	Mean	Median	Max	Min	SD	Correlation_vs Response
target	whether the crime rate is above the median crime rate (1) or not (0)	categorical	response	No	NA	NA	NA	NA	NA	1.00
zn	proportion of residential land zoned for large lots (over 25000 square feet)	continuous numerical	predictor	No	11.58	0.00	100.00	0.00	23.36	-0.43
indus	proportion of non-retail business acres per suburb	continuous numerical	predictor	No	11.11	9.69	27.74	0.46	6.85	0.60
chas	a dummy var. for whether the suburb borders the Charles River (1) or not (0)	categorical	predictor	No	NA	NA	NA	NA	NA	0.08
nox	nitrogen oxides concentration (parts per 10 million)	continuous numerical	predictor	No	0.55	0.54	0.87	0.39	0.12	0.73
rm	average number of	continuous numerical	predictor	No	6.29	6.21	8.78	3.86	0.70	-0.15

Variable	Description	Var_Type_1	Var_Type_2	Missing_Value	Mean	Median	Max	Min	SD	Correlation_vs Response
	rooms per dwelling									
age	proportion of owner-occupied units built prior to 1940	continuous numerical	predictor	No	68.37	77.15	100.00	2.90	28.32	0.63
dis	weighted mean of distances to five Boston employment centers	continuous numerical	predictor	No	3.80	3.19	12.13	1.13	2.11	-0.62
rad	index of accessibility to radial highways	discrete numerical	predictor	No	9.53	5.00	24.00	1.00	8.69	0.63
tax	full-value property-tax rate per $10,000	discrete numerical	predictor	No	409.50	334.50	711.00	187.00	167.90	0.61
ptratio	pupil-teacher ratio by town	continuous numerical	predictor	No	18.40	18.90	22.00	12.60	2.20	0.25
lstat	lower status of the population (percent)	continuous numerical	predictor	No	12.63	11.35	37.97	1.73	7.10	0.47
medv	median value of owner-occupied homes in $1000s	continuous numerical	predictor	No	22.59	21.20	50.00	5.00	9.24	-0.27

Data Preparation

Re-scale Data

The dataset contains variables of different measurements, such as percentage, distance, money values, etc. To put all the predictors and the response on a comparable scale, they are all normalized with mean = 0 and SD = 1.

Hide

```
data_rescaled <- scale(data_t_mod[c(2,3,5:13)]) %>%
  as.data.frame() %>%
  cbind(data_t_mod[c(1,4)]) %>%
  dplyr::select(target, zn, indus, chas, everything())

DT::datatable(data_rescaled)
```

Show 10 entries

	target	zn	indus	chas	nox	rm	age	dis
1	1	-0.495502935416321	1.23797226807758	0	0.434481303529558	2.32435721915126	0.982734775035177	-0.83048638445262
2	1	-0.495502935416321	1.23797226807758	1	2.71448132301663	-1.25937745244706	1.11690903540976	-1.17425351364536
3	1	-0.495502935416321	1.02178305320934	0	1.59162417056247	0.275698127390383	1.11690903540976	-0.862523221703082
4	0	0.788487988980374	-0.902008811530363	0	-1.08266156658026	0.145174140934703	-2.13858222946826	1.53767662510154
5	0	-0.495502935416321	-1.2628110822902	0	-0.568375847898965	1.2262532461437	0.841498711482986	-0.519752794113694
6	0	-0.495502935416321	-0.371760939927849	0	-0.294090131268941	0.695644866421697	0.103540279422784	-0.445949413484851
7	1	-0.495502935416321	1.02178305320934	0	1.18876702426212	-1.18844050328636	1.11690903540976	-1.09451738537754
8	1	-0.495502935416321	1.02178305320934	0	1.18876702426212	-2.51354271360815	1.11690903540976	-1.01449648522305
9	0	-0.495502935416321	-0.864029625134591	0	-0.336947274492382	0.035931239227233	-1.06871904806041	1.26377353210537
10	0	2.92847286297487	-1.09044400557093	0	-1.39123299778903	-0.588313913386883	-1.73959034993332	2.57462598843205

Showing 1 to 10 of 466 entries

Table 5: Rescaled `training set`

Search:

rad	tax	ptratio	lstat	medv
-0.521538206872743	-0.0387262804545447	-1.68354995796081	-1.25761710967871	2.96663146629856
-0.521538206872743	-0.0387262804545447	-1.68354995796081	1.99785401046603	-0.994544073229029
1.66590816150777	1.52768146870325	0.820040725098986	0.875617642492262	-0.778086393473423
-0.40640945064219	-0.652186349516341	-0.818673176540154	-1.04781382382163	0.120212977512341
-0.75179571933385	-1.28947011058054	-0.272435209327107	-1.0999126263499	1.65706250377714
-0.521538206872743	-0.151888817465944	1.1386795393066	-0.698611039307832	0.423253729170189
1.66590816150777	1.52768146870325	0.820040725098986	2.52869856595676	-1.90366632820257
1.66590816150777	1.52768146870325	0.820040725098986	3.42845896637739	-1.68720864844697
-0.521538206872743	-1.10483649756194	0.820040725098986	-0.978818112365273	-0.0421302823043636
-0.982053231794957	-0.562847504507341	-0.909712837742329	-0.476135071754688	-0.182827774145508

Previous 1 2 3 4 5 … 47 Next

Build Models

Because we have a small number of observations to train over, we will use k-fold Cross Validation to train, with k = 10. We'll hold out 15% of the data for validation while doing the initial modeling, but once we select our model, we will retrain over the full training set.

Each of our logistic regression models will use binomial regression with a logit link function.

Model 1: Full Model

The first model includes all the variables. A review of the VIF output of the model suggests some points that are highly colinear and a number of variables that may not be necessary. Model 1 uses the formula:

target ~ .

Hide

```
set.seed(121)
split <- caret::createDataPartition(data_rescaled$target, p=0.85, list=FALSE)
partial_train <- data_rescaled[split, ]
validation <- data_rescaled[ -split, ]
mod1 <- caret::train(target ~., data = partial_train,
              method = "glm", family = "binomial",
              trControl = trainControl(
                  method = "cv", number = 10,
                  savePredictions = TRUE),
              tuneLength = 5,
              preProcess = c("center", "scale"))
knitr::kable(vif(mod1$finalModel))
```

	x
zn	1.775536
indus	2.615682
chas1	1.289891
nox	4.090926
rm	6.680172
age	2.408913
dis	3.574289
rad	2.078134
tax	2.209580
ptratio	2.433736
lstat	2.735861
medv	9.246747

Model 2: Removing Predictors Seemed Unnecessary

Our second model ignores the colinear issues but removes models that seemed unnecessary in Model #1. Model 2 uses the formula:

target ~ zn + nox + age + dis + rad + ptratio + medv

Hide

```
# remove low p-values
mod2 <- train(target ~ zn + nox + age + dis + rad + ptratio + medv,
            data = partial_train,
            method = "glm", family = "binomial",
            trControl = trainControl(
                  method = "cv", number = 10,
                  savePredictions = TRUE),
            tuneLength = 5,
            preProcess = c("center", "scale"))
knitr::kable(vif(mod2$finalModel))
```

	x
zn	1.801287
nox	3.049522
age	1.685178
dis	3.659469
rad	1.235992
ptratio	1.826575
medv	2.094548

Model 3: Removing Highest VIF Values

Model #3 removes the variables with the 2 highest VIF values from model1. The model formula is:

target ~ indus + rm + age + dis + tax + ptratio + lstat + medv

Hide

```
## Reduce Collinearity by removing high VIFs
mod3 <- train(target ~ indus + rm + age + dis + tax + ptratio + lstat + medv, data = partial_train,
              method = "glm", family = "binomial",
              trControl = trainControl(
                  method = "cv", number = 10,
                  savePredictions = TRUE),
              tuneLength = 5,
              preProcess = c("center", "scale"))
knitr::kable(vif(mod3$finalModel))
```

	x
indus	2.190206
rm	4.462813
age	2.097140
dis	1.956005
tax	1.749705
ptratio	1.423980
lstat	2.765737
medv	5.782926

Model 4: Removing Poor Predictors

Model #4 takes the advances in model #3 and removes those values shown to be poor predictors.

target ~ age + dis + tax + medv

Hide

```
## reduce collinearity, and remove low values
mod4 <- train(target ~ age + dis + tax + medv,
            data = partial_train,
            method = "glm", family = "binomial",
            trControl = trainControl(
                  method = "cv", number = 10,
                  savePredictions = TRUE),
            tuneLength = 5,
            preProcess = c("center", "scale"))
knitr::kable(vif(mod4$finalModel))
```

	x
age	1.733106
dis	1.715677
tax	1.386751
medv	1.413739

Model 5: Stepwise Based on AIC

Model #5: We use stepwise function based on AIC criterion in both direction and get Model #5 in 10 steps.

target ~ nox + rad + tax + ptratio + medv + lstat + dis + zn + age

Hide

```
full_model <- glm(target ~ ., data = partial_train, family = "binomial")
model_5 <- stepwise(full_model, criterion = 'AIC', direction = 'forward/backward', trace = TRUE)
```

```
##
## Direction:  forward/backward
## Criterion:  AIC
##
## Start:  AIC=552.24
## target ~ 1
##
##           Df Deviance    AIC
## + nox      1   263.04 267.04
## + rad      1   353.20 357.20
## + dis      1   362.95 366.95
## + age      1   374.91 378.91
## + tax      1   389.61 393.61
## + indus    1   394.18 398.18
## + zn       1   445.94 449.94
## + lstat    1   458.04 462.04
## + ptratio  1   527.19 531.19
## + medv     1   527.24 531.24
## + rm       1   543.46 547.46
## + chas     1   544.87 548.87
## <none>         550.24 552.24
##
```

```
## Step:  AIC=267.04
## target ~ nox
##
##           Df Deviance    AIC
## + rad      1   218.02 224.02
## + rm       1   257.59 263.59
## + medv     1   257.84 263.84
## + chas     1   259.12 265.12
## + indus    1   259.98 265.98
## + tax      1   260.04 266.04
## + zn       1   260.29 266.29
## <none>         263.04 267.04
## + ptratio  1   261.57 267.57
## + dis      1   261.93 267.93
## + age      1   262.05 268.05
## + lstat    1   263.04 269.04
## - nox      1   550.24 552.24
##
## Step:  AIC=224.02
## target ~ nox + rad
##
##           Df Deviance    AIC
## + tax      1   205.03 213.03
## + indus    1   213.29 221.29
## + zn       1   214.62 222.62
## + ptratio  1   215.77 223.77
## + medv     1   215.87 223.87
## <none>         218.02 224.02
## + dis      1   216.04 224.04
## + rm       1   216.20 224.20
## + chas     1   216.21 224.21
## + age      1   216.26 224.26
## + lstat    1   217.98 225.98
## - rad      1   263.04 267.04
## - nox      1   353.20 357.20
##
```

```
## Step:  AIC=213.03
## target ~ nox + rad + tax
##
##           Df Deviance    AIC
## + ptratio  1   199.20 209.20
## + zn       1   201.48 211.48
## + age      1   202.17 212.17
## <none>         205.03 213.03
## + lstat    1   203.38 213.38
## + dis      1   203.44 213.44
## + chas     1   204.25 214.25
## + indus    1   204.50 214.50
## + rm       1   204.73 214.73
## + medv     1   204.93 214.93
## - tax      1   218.02 224.02
## - rad      1   260.04 266.04
## - nox      1   347.31 353.31
##
## Step:  AIC=209.2
## target ~ nox + rad + tax + ptratio
##
##           Df Deviance    AIC
## + medv     1   196.13 208.13
## + age      1   196.43 208.43
## <none>         199.20 209.20
## + zn       1   197.55 209.55
## + rm       1   197.59 209.59
## + chas     1   197.61 209.61
## + dis      1   197.88 209.88
## + lstat    1   198.62 210.62
## + indus    1   198.85 210.85
## - ptratio  1   205.03 213.03
## - tax      1   215.77 223.77
## - rad      1   259.41 267.41
## - nox      1   346.63 354.63
##
```

```
## Step:  AIC=208.13
## target ~ nox + rad + tax + ptratio + medv
##
##           Df Deviance    AIC
## + lstat    1   190.72 204.72
## + age      1   192.40 206.40
## + dis      1   193.46 207.46
## <none>         196.13 208.13
## + zn       1   194.18 208.18
## + chas     1   194.38 208.38
## - medv     1   199.20 209.20
## + indus    1   195.82 209.82
## + rm       1   195.89 209.89
## - ptratio  1   204.93 214.93
## - tax      1   208.44 218.44
## - rad      1   246.62 256.62
## - nox      1   346.52 356.52
##
## Step:  AIC=204.72
## target ~ nox + rad + tax + ptratio + medv + lstat
##
##           Df Deviance    AIC
## + dis      1   187.06 203.06
## <none>         190.72 204.72
## + zn       1   188.81 204.81
## + age      1   189.03 205.03
## + chas     1   189.96 205.96
## + indus    1   190.09 206.09
## + rm       1   190.72 206.72
## - lstat    1   196.13 208.13
## - medv     1   198.62 210.62
## - ptratio  1   201.36 213.36
## - tax      1   205.13 217.13
## - rad      1   243.00 255.00
## - nox      1   310.16 322.16
##
```

```
## Step:  AIC=203.06
## target ~ nox + rad + tax + ptratio + medv + lstat + dis
##
##           Df Deviance    AIC
## + zn       1   182.11 200.11
## + age      1   183.27 201.27
## <none>         187.06 203.06
## + chas     1   185.67 203.67
## + indus    1   186.31 204.31
## - dis      1   190.72 204.72
## + rm       1   187.06 205.06
## - lstat    1   193.46 207.46
## - medv     1   197.45 211.45
## - ptratio  1   198.90 212.90
## - tax      1   199.70 213.70
## - rad      1   237.35 251.35
## - nox      1   265.54 279.54
##
## Step:  AIC=200.11
## target ~ nox + rad + tax + ptratio + medv + lstat + dis + zn
##
##           Df Deviance    AIC
## + age      1   178.21 198.21
## <none>         182.11 200.11
## + indus    1   181.38 201.38
## + chas     1   181.44 201.44
## + rm       1   182.06 202.06
## - zn       1   187.06 203.06
## - dis      1   188.81 204.81
## - lstat    1   189.00 205.00
## - ptratio  1   190.53 206.53
## - tax      1   192.52 208.52
## - medv     1   194.47 210.47
## - rad      1   229.73 245.73
## - nox      1   258.55 274.55
##
```

```
## Step:  AIC=198.21
## target ~ nox + rad + tax + ptratio + medv + lstat + dis + zn +
##     age
##
##           Df Deviance    AIC
## <none>         178.21 198.21
## + rm       1   177.42 199.42
## + indus    1   177.46 199.46
## + chas     1   177.73 199.73
## - lstat    1   181.88 199.88
## - age      1   182.11 200.11
## - zn       1   183.27 201.27
## - dis      1   187.43 205.43
## - ptratio  1   187.51 205.51
## - tax      1   188.85 206.85
## - medv     1   191.07 209.07
## - rad      1   226.81 244.81
## - nox      1   245.63 263.63
```

Hide

```r
summary(model_5)
```

```
##
## Call:
## glm(formula = target ~ nox + rad + tax + ptratio + medv + lstat +
##     dis + zn + age, family = "binomial", data = partial_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9810  -0.2345  -0.0020   0.0038   3.2862
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.6310     0.7350   3.580 0.000344 ***
## nox           5.0016     0.8106   6.170 6.82e-10 ***
## rad           6.5437     1.4025   4.666 3.07e-06 ***
## tax          -1.4398     0.4858  -2.964 0.003037 **
## ptratio       0.7573     0.2569   2.948 0.003199 **
## medv          1.2908     0.3889   3.319 0.000904 ***
## lstat         0.6877     0.3597   1.912 0.055883 .
## dis           1.3092     0.4517   2.898 0.003752 **
## zn           -1.4975     0.7628  -1.963 0.049617 *
## age           0.6307     0.3307   1.907 0.056531 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 550.24  on 396  degrees of freedom
## Residual deviance: 178.21  on 387  degrees of freedom
## AIC: 198.21
##
## Number of Fisher Scoring iterations: 9
```

Model 6: Stepwise Based on BIC

Model #6: We use stepwise function based on BIC criterion in both direction and get Model #6 in 4 steps.

target ~ nox + rad + tax

Hide

```
model_6 <- stepwise(full_model, criterion = 'BIC', direction = 'forward/backward', trace = TRUE)
```

```
##
## Direction:  forward/backward
## Criterion:  BIC
##
## Start:  AIC=556.22
## target ~ 1
##
##           Df Deviance    AIC
## + nox      1   263.04 275.00
## + rad      1   353.20 365.17
## + dis      1   362.95 374.91
## + age      1   374.91 386.88
## + tax      1   389.61 401.58
## + indus    1   394.18 406.15
## + zn       1   445.94 457.90
## + lstat    1   458.04 470.01
## + ptratio  1   527.19 539.15
## + medv     1   527.24 539.21
## + rm       1   543.46 555.42
## <none>         550.24 556.22
## + chas     1   544.87 556.84
##
```

```
## Step:  AIC=275
## target ~ nox
##
##            Df Deviance    AIC
## + rad      1   218.02 235.98
## <none>          263.04 275.00
## + rm       1   257.59 275.54
## + medv     1   257.84 275.79
## + chas     1   259.12 277.07
## + indus    1   259.98 277.93
## + tax      1   260.04 278.00
## + zn       1   260.29 278.24
## + ptratio  1   261.57 279.52
## + dis      1   261.93 279.88
## + age      1   262.05 280.00
## + lstat    1   263.04 280.99
## - nox      1   550.24 556.22
##
## Step:  AIC=235.98
## target ~ nox + rad
##
##            Df Deviance    AIC
## + tax      1   205.03 228.97
## <none>          218.02 235.98
## + indus    1   213.29 237.23
## + zn       1   214.62 238.56
## + ptratio  1   215.77 239.71
## + medv     1   215.87 239.80
## + dis      1   216.04 239.98
## + rm       1   216.20 240.14
## + chas     1   216.21 240.14
## + age      1   216.26 240.19
## + lstat    1   217.98 241.91
## - rad      1   263.04 275.00
## - nox      1   353.20 365.17
##
```

```
## Step:  AIC=228.97
## target ~ nox + rad + tax
##
##           Df Deviance    AIC
## <none>          205.03 228.97
## + ptratio  1    199.20 229.12
## + zn       1    201.48 231.40
## + age      1    202.17 232.09
## + lstat    1    203.38 233.30
## + dis      1    203.44 233.36
## + chas     1    204.25 234.17
## + indus    1    204.50 234.42
## + rm       1    204.73 234.65
## + medv     1    204.93 234.84
## - tax      1    218.02 235.98
## - rad      1    260.04 278.00
## - nox      1    347.31 365.26
```

Hide

```r
summary(model_6)
```

```
## 
## Call:
## glm(formula = target ~ nox + rad + tax, family = "binomial", 
##     data = partial_train)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -1.87592  -0.37435  -0.04307   0.00727   2.51162  
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)   2.5664     0.6048   4.243 2.20e-05 ***
## nox           4.0277     0.5539   7.272 3.54e-13 ***
## rad           5.4287     1.0918   4.972 6.62e-07 ***
## tax          -1.3703     0.4225  -3.244  0.00118 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 550.24  on 396  degrees of freedom
## Residual deviance: 205.03  on 393  degrees of freedom
## AIC: 213.03
## 
## Number of Fisher Scoring iterations: 8
```

Model 7: Best Subset Based on AIC

Model #7: We use best subset method based on AIC criterion to find Model #7.

target ~ zn + nox + age + dis + rad + tax + ptratio + lstat + medv (Same as Model 5)

Hide

```
Xy <- partial_train %>% dplyr::select(-target,everything())
model_7 <- bestglm(Xy = Xy, family = binomial, IC = 'AIC', method = 'exhaustive')
```

Top 5 models among all the subsets:

Hide

```
model_7$BestModels %>%
  mutate(model_rank = row_number()) %>%
  dplyr::select(model_rank, everything()) %>%
  kable() %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"),full_width = F
```

model_rank	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	Criterion
1	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	196.2099
2	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	197.4200
3	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	197.4567
4	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	197.5250
5	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	197.7333

The rank 1 model is selected as model 7.

Hide

```
model_7$BestModel
```

```
## 
## Call:  glm(formula = y ~ ., family = family, data = Xi, weights = weights)
## 
## Coefficients:
## (Intercept)           zn          nox          age          dis          rad  
##      2.6310      -1.4975       5.0016       0.6307       1.3092       6.5437  
##         tax      ptratio        lstat         medv  
##     -1.4398       0.7573       0.6877       1.2908  
## 
## Degrees of Freedom: 396 Total (i.e. Null);  387 Residual
## Null Deviance:       550.2 
## Residual Deviance: 178.2     AIC: 198.2
```

Model 8: Best Subset Based on BIC

Model #8: We use best subset method based on BIC criterion to find Model #8.

target ~ nox + rad + tax (Same as Model 6)

Hide

```
model_8 <- bestglm(Xy = Xy, family = binomial, IC = 'BIC', method = 'exhaustive')
```

Top 5 models among all the subsets:

Hide

```
model_8$BestModels %>%
  mutate(model_rank = row_number()) %>%
  dplyr::select(model_rank, everything()) %>%
  kable() %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"),full_width = F
```

model_rank	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	Criterion
1	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	222.9816
2	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	223.1394
3	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	225.4175
4	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	226.0475
5	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	226.1033

The rank 1 model is selected as model 8.

Hide

```
model_8$BestModel
```

```
##
## Call:  glm(formula = y ~ ., family = family, data = Xi, weights = weights)
##
## Coefficients:
## (Intercept)          nox          rad          tax
##       2.566        4.028        5.429       -1.370
##
## Degrees of Freedom: 396 Total (i.e. Null);  393 Residual
## Null Deviance:       550.2
## Residual Deviance: 205   AIC: 213
```

Hide

```
#re-train data using train() function
model_5 <- train(target ~ nox + rad + tax + ptratio + age + medv + dis + zn + age,
            data = partial_train,
            method = "glm", family = "binomial",
            trControl = trainControl(
                  method = "cv", number = 10,
                  savePredictions = TRUE),
            tuneLength = 5,
            preProcess = c("center", "scale"))

model_6 <- train(target ~ nox + rad + tax,
            data = partial_train,
            method = "glm", family = "binomial",
            trControl = trainControl(
                  method = "cv", number = 10,
                  savePredictions = TRUE),
            tuneLength = 5,
            preProcess = c("center", "scale"))
```

Select Models

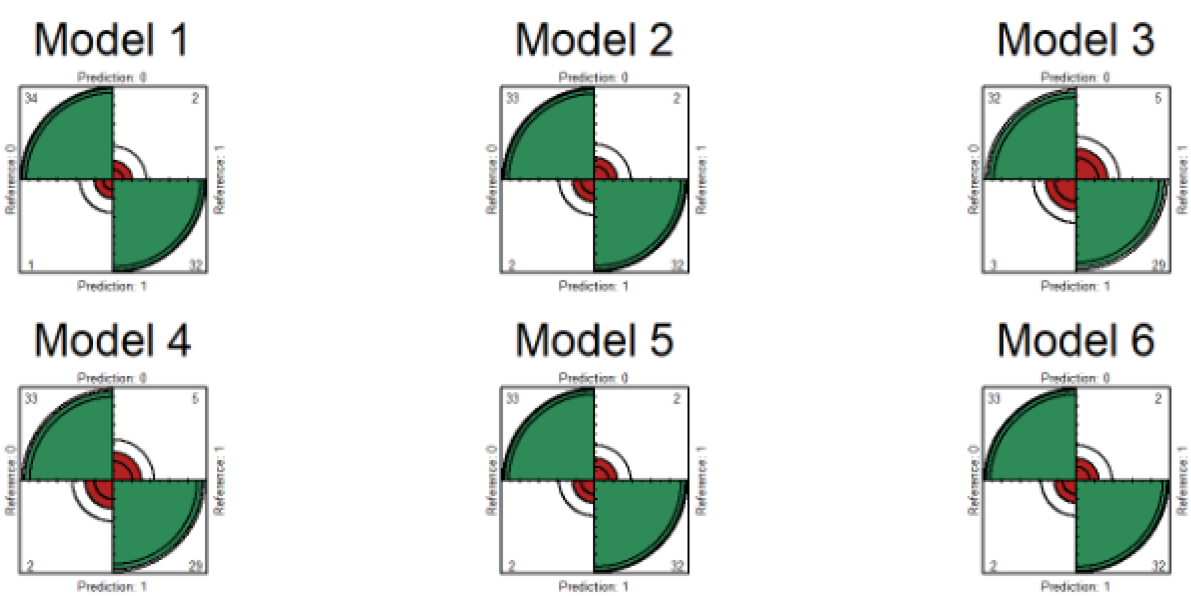
To help aid in model selection, we will review their accuracy by making predictions on our holdout validation set, and comparing their performance using a variety of confusion matrix adjacent functions like fourfold plots, summary statistics, and ROC / AUC plots.

Fourfold Plots

Hide

```
preds1 <- predict(mod1, newdata = validation)
preds2 <- predict(mod2, newdata = validation)
preds3 <- predict(mod3, newdata = validation)
preds4 <- predict(mod4, newdata = validation)
preds5 <- predict(model_5, newdata = validation)
preds6 <- predict(model_6, newdata = validation)
m1cM <- confusionMatrix(preds1, validation$target, mode = "everything")
m2cM <- confusionMatrix(preds2, validation$target, mode = "everything")
m3cM <- confusionMatrix(preds3, validation$target, mode = "everything")
m4cM <- confusionMatrix(preds4, validation$target, mode = "everything")
m5cM <- confusionMatrix(preds5, validation$target, mode = "everything")
m6cM <- confusionMatrix(preds6, validation$target, mode = "everything")

par(mfrow=c(3,3))
fourfoldplot(m1cM$table, color = c("#B22222", "#2E8B57"), main="Model 1")
fourfoldplot(m2cM$table, color = c("#B22222", "#2E8B57"), main="Model 2")
fourfoldplot(m3cM$table, color = c("#B22222", "#2E8B57"), main="Model 3")
fourfoldplot(m4cM$table, color = c("#B22222", "#2E8B57"), main="Model 4")
fourfoldplot(m5cM$table, color = c("#B22222", "#2E8B57"), main="Model 5")
fourfoldplot(m5cM$table, color = c("#B22222", "#2E8B57"), main="Model 6")
```



Summary Statistics

Model 1, Model 2 and Model 5 have best performance in at least one category.

Hide

```
temp <- data.frame(m1cM$overall,
                   m2cM$overall,
                   m3cM$overall,
                   m4cM$overall,
                   m5cM$overall,
                   m6cM$overall) %>%
  t() %>%
  data.frame() %>%
  dplyr::select(Accuracy) %>%
  mutate(Classification_Error_Rate = 1-Accuracy)

Summ_Stat <-data.frame(m1cM$byClass,
                   m2cM$byClass,
                   m3cM$byClass,
                   m4cM$byClass,
                   m5cM$byClass,
                   m6cM$byClass) %>%
  t() %>%
  data.frame() %>%
  cbind(temp) %>%
```

```
# manipulate results DF
  mutate(Model = c("Model 1", "Model 2", "Model 3", "Model 4", "Model 5", "Model 6")) %>%
  dplyr::select(Model, Accuracy, Classification_Error_Rate, Precision, Sensitivity, Specificity, F1
  add_row(Model = 'Model 7 (Same as Model 5)') %>%
  add_row(Model = 'Model 8 (Same as Model 6)') %>%
  mutate_if(is.numeric, round,3) %>%
  mutate_at(c('Accuracy', 'Precision', 'Sensitivity', 'Specificity', 'F1'), function(x) {
  cell_spec(x,
            bold = if_else(x == max(x, na.rm = TRUE),TRUE, FALSE),
            font_size = if_else(x == max(x, na.rm = TRUE),14, 12))}) %>%
  mutate(Classification_Error_Rate = cell_spec(Classification_Error_Rate,
                                               bold = if_else(Classification_Error_Rate == min(Clas
                                               font_size = if_else(Classification_Error_Rate == min
  mutate(Model = cell_spec(Model,
                           bold = if_else(Model %in% c('Model 1', 'Model 2', 'Model 5'), TRUE, FALS
                           font_size = if_else(Model %in% c('Model 1', 'Model 2', 'Model 5'), 14, 1
  kable('html', escape = F) %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"),full_width = F

Summ_Stat
```

Model	Accuracy	Classification_Error_Rate	Precision	Sensitivity	Specificity	F1
Model 1	**0.957**	**0.043**	**0.944**	**0.971**	**0.941**	**0.958**
Model 2	0.942	0.058	0.943	0.943	**0.941**	0.943
Model 3	0.884	0.116	0.865	0.914	0.853	0.889
Model 4	0.899	0.101	0.868	0.943	0.853	0.904
Model 5	0.942	0.058	0.943	0.943	**0.941**	0.943
Model 6	0.928	0.072	0.917	0.943	0.912	0.93
Model 7 (Same as Model 5)	NA	NA	NA	NA	NA	NA
Model 8 (Same as Model 6)	NA	NA	NA	NA	NA	NA

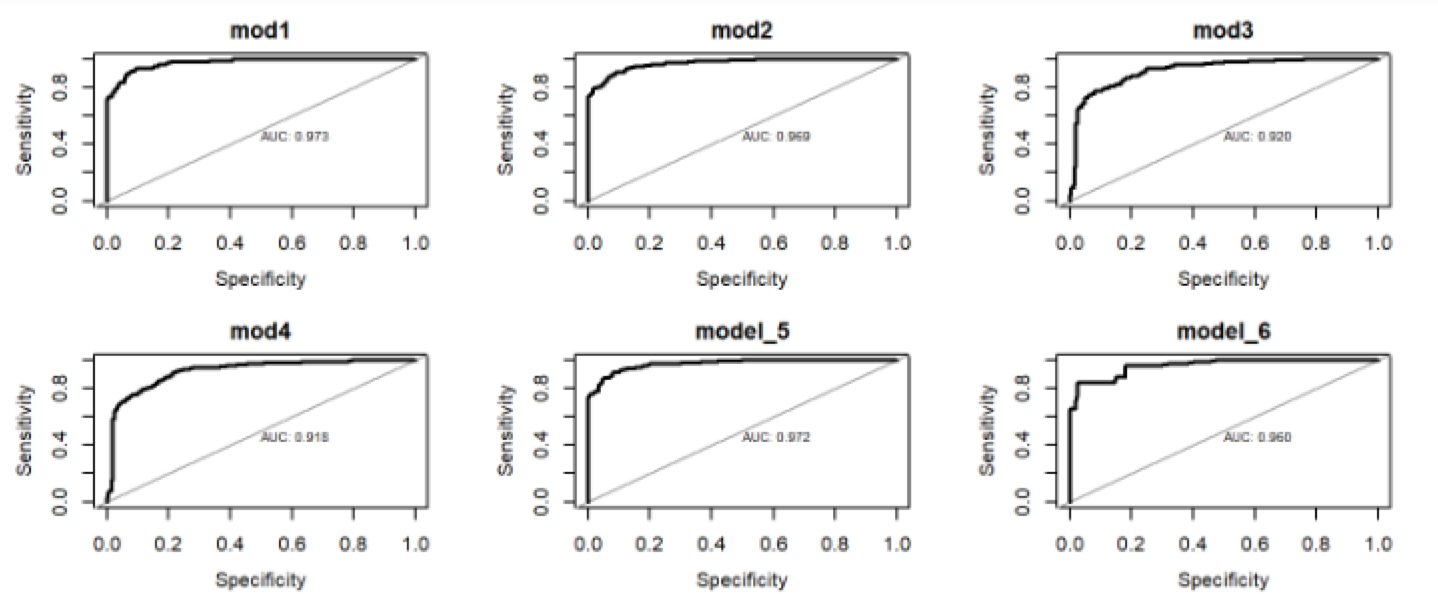
ROC / AUC

The larger the area under the curve, the better the model.

AUC: model 1 > model 5 > model 2 > model 6 > model 3 > model 4

Hide

```
getROC <- function(model) {
    name <- deparse(substitute(model))
    pred.prob1 <- predict(model, newdata = data_rescaled, type="prob")
    p1 <- data.frame(pred = data_rescaled$target, prob = pred.prob1[[1]])
    p1 <- p1[order(p1$prob),]
    rocobj <- pROC::roc(p1$pred, p1$prob)
    plot(rocobj, asp=NA, legacy.axes = TRUE, print.auc=TRUE,
         xlab="Specificity", main = name)
}
par(mfrow=c(3,3))
getROC(mod1)
getROC(mod2)
getROC(mod3)
getROC(mod4)
getROC(model_5)
getROC(model_6)
```



R^2, AIC, AICc & BIC

Although Model 1 has the largest R^2, Model 5 has the smallest AIC and AICc, and the second largest R^2.

Hide

```
null_model <- glm(target ~ 1, data = partial_train, family = 'binomial')
#refit models using glm() function
model_1 <- glm(target~., partial_train, family = 'binomial')
model_2 <- glm(target~zn + nox + age + dis + rad + ptratio + medv, partial_train, family = 'binomia
model_3 <- glm(target~indus + rm + age + dis + tax + ptratio + lstat + medv, partial_train, family 
model_4 <- glm(target~age + dis + tax + medv, partial_train, family = 'binomial')
model_5 <- glm(target~nox + rad + tax + ptratio + age + medv + dis + zn + age, partial_train, famil
model_6 <- glm(target~nox + rad + tax, partial_train, family = 'binomial')
models <- list(model_1, model_2, model_3, model_4, model_5, model_6)

Predictor <- models %>%
  lapply(function(x) str_c(unlist(row.names(summary(x)$coefficients)), collapse = ',')) %>%
  unlist() %>% str_remove('\\(Intercept\\)\\,')

McFaddens_R2 <- list(1-logLik(model_1)/logLik(null_model),
                     1-logLik(model_2)/logLik(null_model),
                     1-logLik(model_3)/logLik(null_model),
                     1-logLik(model_4)/logLik(null_model),
                     1-logLik(model_5)/logLik(null_model),
                     1-logLik(model_6)/logLik(null_model)) %>%
  unlist()

AIC <- models %>%
  lapply(function(x) AIC(x)) %>%
  unlist()

AICc <- models %>%
  lapply(function(x) AICc(x)) %>%
  unlist()

BIC <- models %>%
  lapply(function(x) BIC(x)) %>%
  unlist()

cbind(Predictor, McFaddens_R2, AIC, AICc, BIC) %>%
  as.data.frame(stringsAsFactors = FALSE) %>%
  mutate_at(c('McFaddens_R2','AIC','AICc','BIC'), as.numeric) %>%
  mutate(Model = c(str_c('Model ', c(1:6)))) %>%
```

```
  dplyr::select(Model, everything()) %>%
  add_row(Model = 'Model 7', Predictor = 'Same as Model 5') %>%
  add_row(Model = 'Model 8', Predictor = 'Same as Model 6') %>%
  mutate_if(is.numeric, round,3) %>%
  mutate(McFaddens_R2 = cell_spec(McFaddens_R2,
                                  bold = if_else(McFaddens_R2 == max(McFaddens_R2, na.rm = TRUE), T
                                  font_size = if_else(McFaddens_R2 == max(McFaddens_R2, na.rm = TRU
         AIC = cell_spec(AIC,
                         bold = if_else(AIC == min(AIC, na.rm = TRUE), TRUE, FALSE),
                         font_size = if_else(AIC == min(AIC, na.rm = TRUE), 14, 12)),
         AICc = cell_spec(AICc,
                          bold = if_else(AICc == min(AICc, na.rm = TRUE), TRUE, FALSE),
                          font_size = if_else(AICc == min(AICc, na.rm = TRUE), 14, 12)),
         BIC = cell_spec(BIC,
                         bold = if_else(BIC == min(BIC, na.rm = TRUE), TRUE, FALSE),
                         font_size = if_else(BIC == min(BIC, na.rm = TRUE), 14, 12)),
         Model = cell_spec(Model,
                           bold = if_else(Model == 'Model 5', TRUE, FALSE),
                           font_size = if_else(Model == 'Model 5', 14, 12))) %>%
  kable('html', escape = F) %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"),full_width = F
```

Model	Predictor	McFaddens_R2	AIC	AICc	BIC
Model 1	zn,indus,chas1,nox,rm,age,dis,rad,tax,ptratio,lstat,medv	**0.68**	201.853	202.804	253.644
Model 2	zn,nox,age,dis,rad,ptratio,medv	0.653	207.183	207.555	239.055
Model 3	indus,rm,age,dis,tax,ptratio,lstat,medv	0.463	313.336	313.801	349.191
Model 4	age,dis,tax,medv	0.455	310.012	310.165	329.932
Model 5	nox,rad,tax,ptratio,age,medv,dis,zn	0.669	**199.883**	**200.348**	235.739
Model 6	nox,rad,tax	0.627	213.03	213.132	**228.966**
Model 7	Same as Model 5	NA	NA	NA	NA
Model 8	Same as Model 6	NA	NA	NA	NA

Model Selection

From the model selection process above, we know that Model 1 suffers from co-linearity issues, the rest of the models tried to eliminate these issues but also to achieve best prediction performance. Among them, Model 5 has 1) the highest Specificity, 2) second highest accuracy, precision, sensitivity, F1 Score, AUC and McFadden's R squared proceed by model1, 3) lowest AIC and AICc. Therefore Model 5 is selected to be the final model.

Hide

```
finalmod_FS <- train(target ~ nox + rad + tax + ptratio + age + medv + dis + zn + age,
            data = data_rescaled,
            method = "glm", family = "binomial",
            trControl = trainControl(
                  method = "cv", number = 10,
                  savePredictions = TRUE),
            tuneLength = 5,
            preProcess = c("center", "scale"))
summary(finalmod_FS)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8295  -0.1752  -0.0021   0.0032   3.4191
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.4406     0.6769   3.606 0.000311 ***
## nox           4.9942     0.7792   6.410 1.46e-10 ***
## rad           6.2982     1.3010   4.841 1.29e-06 ***
## tax          -1.3023     0.4454  -2.924 0.003459 **
## ptratio       0.7110     0.2447   2.905 0.003668 **
## age           0.9332     0.3101   3.009 0.002622 **
## medv          1.0207     0.3275   3.117 0.001829 **
## dis           1.3798     0.4510   3.060 0.002217 **
## zn           -1.6039     0.7481  -2.144 0.032033 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 197.32  on 457  degrees of freedom
## AIC: 215.32
##
## Number of Fisher Scoring iterations: 9
```

Odds Ratio

We will also create a table of the Odds Ratio for our final model beside the 95% confidence interval of those boundaries. Odd Ratio (OR) is a measure of association between exposure and an outcome. The OR represents the odds that an outcome will occur given a particular exposure, compared to the odds of the outcome occurring in the absence of that exposure.

Hide

```
odds <- round(exp(cbind(OddsRatio = coef(finalmod_FS$finalModel), confint(finalmod_FS$finalModel)))
knitr::kable(odds)
```

	OddsRatio	2.5 %	97.5 %
(Intercept)	11.479	3.161	45.685
nox	147.561	35.436	763.373
rad	543.616	49.797	8423.905
tax	0.272	0.104	0.615
ptratio	2.036	1.274	3.343
age	2.543	1.408	4.780
medv	2.775	1.503	5.441
dis	3.974	1.693	10.049
zn	0.201	0.040	0.747

So we can now say that with a one unit increase in the scaled age variable, the odds of the neighborhood being below the median crime rate increase by 2.543%.

All that is left is to use our final model to make predictions over the evaluation dataset.

Make Predictions

We make our final predictions, create a dataframe with the prediction and the predicted probabilities along with the `evaluation set`. The data set is rescaled in as well. The result shows that among the 40 observations, 23 are predicted to have crime rate below median (`0`), 17 are predicted to be above median (`1`).

Hide

```
data_e_rescaled <- data_e %>%
  dplyr::select(-chas) %>%
  scale() %>%
  cbind(data_e[3]) %>%
  as.data.frame() %>%
  mutate(chas = as.factor(chas))

finalpreds <- predict(finalmod_FS, data_e_rescaled)
finalpreds.probs <- predict(finalmod_FS, data_e, type="prob")
finaldf <- cbind(predicted_Response=finalpreds, Predicted_Prob=finalpreds.probs, data_e) %>%
  mutate(Predicted_Prob.0 = percent(Predicted_Prob.0),
         Predicted_Prob.1 = percent(Predicted_Prob.1))

DT::datatable(finaldf, caption = 'Predicted Result of Final Model: Model 5')
```

Show 10 entries

Search:

Predicted Result of Final Model: Model 5

	predicted_Response	Predicted_Prob.0	Predicted_Prob.1	zn	indus	chas	nox
1	0	100%	0%	0	7.07	0	0.469
2	1	100%	0%	0	8.14	0	0.538
3	1	100%	0%	0	8.14	0	0.538
4	0	100%	0%	0	8.14	0	0.538
5	0	100%	0%	0	5.96	0	0.499
6	0	100%	0%	25	5.13	0	0.453
7	0	100%	0%	25	5.13	0	0.453
8	0	100%	0%	0	4.49	0	0.449
9	0	100%	0%	0	4.49	0	0.449
10	0	100%	0%	0	2.89	0	0.445

Showing 1 to 10 of 40 entries

Previous 1 2 3 4 Next

rm	age	dis	rad	tax	ptratio	lstat	medv
7.185	61.1	4.9671	2	242	17.8	4.03	34.7
6.096	84.5	4.4619	4	307	21	10.26	18.2
6.495	94.4	4.4547	4	307	21	12.8	18.4
5.95	82	3.99	4	307	21	27.71	13.2
5.85	41.5	3.9342	5	279	19.2	8.77	21
5.741	66.2	7.2254	8	284	19.7	13.15	18.7
5.966	93.4	6.8185	8	284	19.7	14.44	16
6.63	56.1	4.4377	3	247	18.5	6.53	26.6
6.121	56.8	3.7476	3	247	18.5	8.44	22.2
6.163	69.6	3.4952	2	276	18	11.34	21.4

ext

Hide

```
finaldf %>%
  group_by(predicted_Response) %>%
  tally() %>%
  datatable(caption = 'Summary of Model 5 Predicted Result')
```

Show 10 entries

Search:

Summary of Model 5 Predicted Result

	predicted_Response	n
1	0	23
2	1	17

Showing 1 to 2 of 2 entries

Previous 1 Next

Appendix

https://github.com/Rajwantmishra/DATA621_CR4/blob/master/HW3/Homework3_Final.Rmd

Thank you