Homework #3: Crime Prediction

Data 621 Business Analytics and Data Mining

Aadi Kalloo, Nathan Lim, Asher Meyers, Daniel Smilowitz, Logan Thomson

Due July 3, 2016

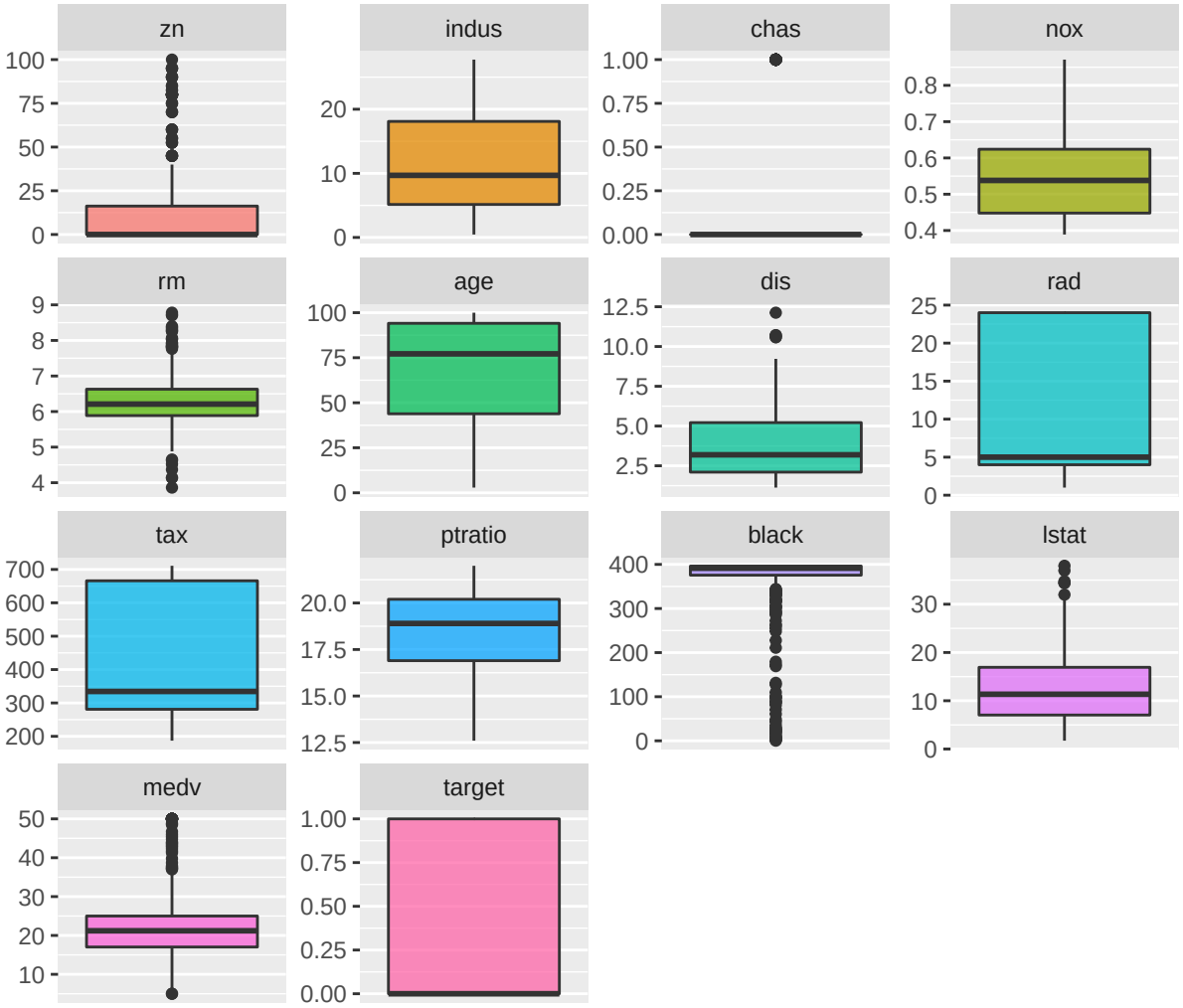
Contents

Data Exploration

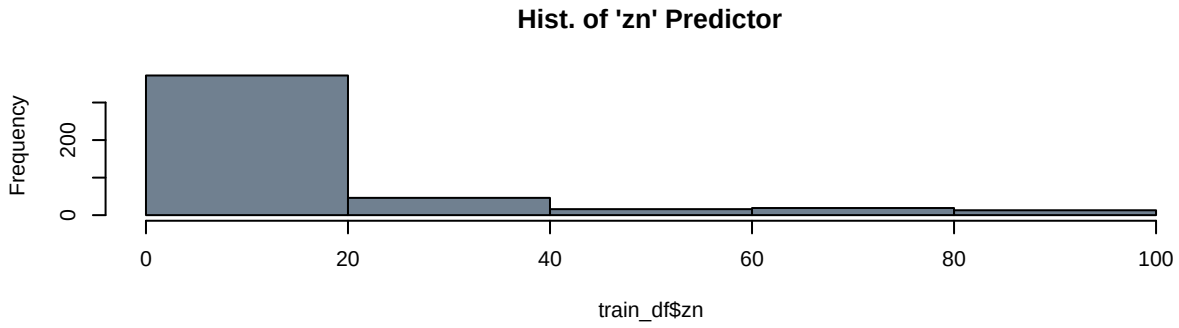
The supplied data set contains various pieces of information about neighborhoods in a major U.S. city. The data set is made up of 466 rows, with 14 variables; 13 predictors and 1 response variable (`target`), which is a binary categorical variable. A summary of each variable is below:

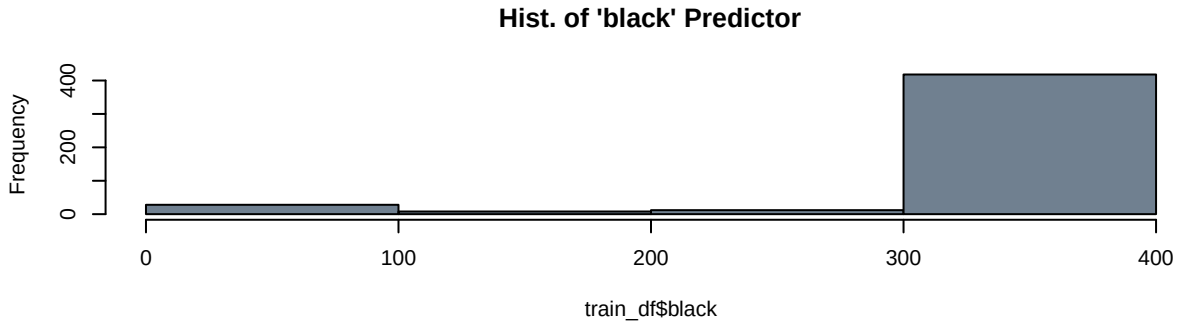
	MEAN	MEDIAN	IQR	SKEWNESS	CORRELATION TO TARGET
zn	11.58	0	16.25	2.18	-0.43
indus	11.11	9.69	12.96	0.29	0.6
chas	0.07	0	0	3.34	0.08
nox	0.55	0.54	0.18	0.75	0.73
rm	6.29	6.21	0.74	0.48	-0.15
age	68.37	77.15	50.22	-0.58	0.63
dis	3.8	3.19	3.11	1	-0.62
rad	9.53	5	20	1.01	0.63
tax	409.5	334.5	385	0.66	0.61
ptratio	18.4	18.9	3.3	-0.75	0.25
black	357.1	391.3	20.63	-2.92	-0.35
lstat	12.63	11.35	9.89	0.91	0.47
medv	22.59	21.2	7.98	1.08	-0.27
target	0.49	0	1	0.03	1

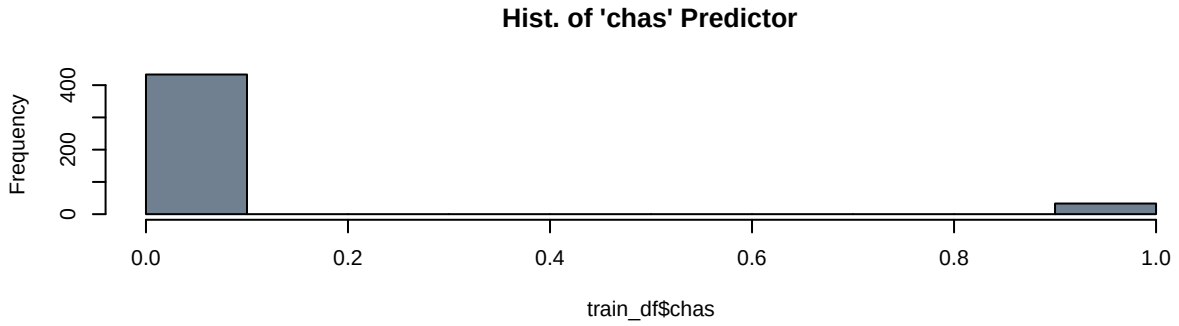
None of our predictors have missing values, so there will be no need to impute any variables. However, some of the predictors have a good amount of skew to the distribution of cases, and may need to be transformed to find a best fit model.



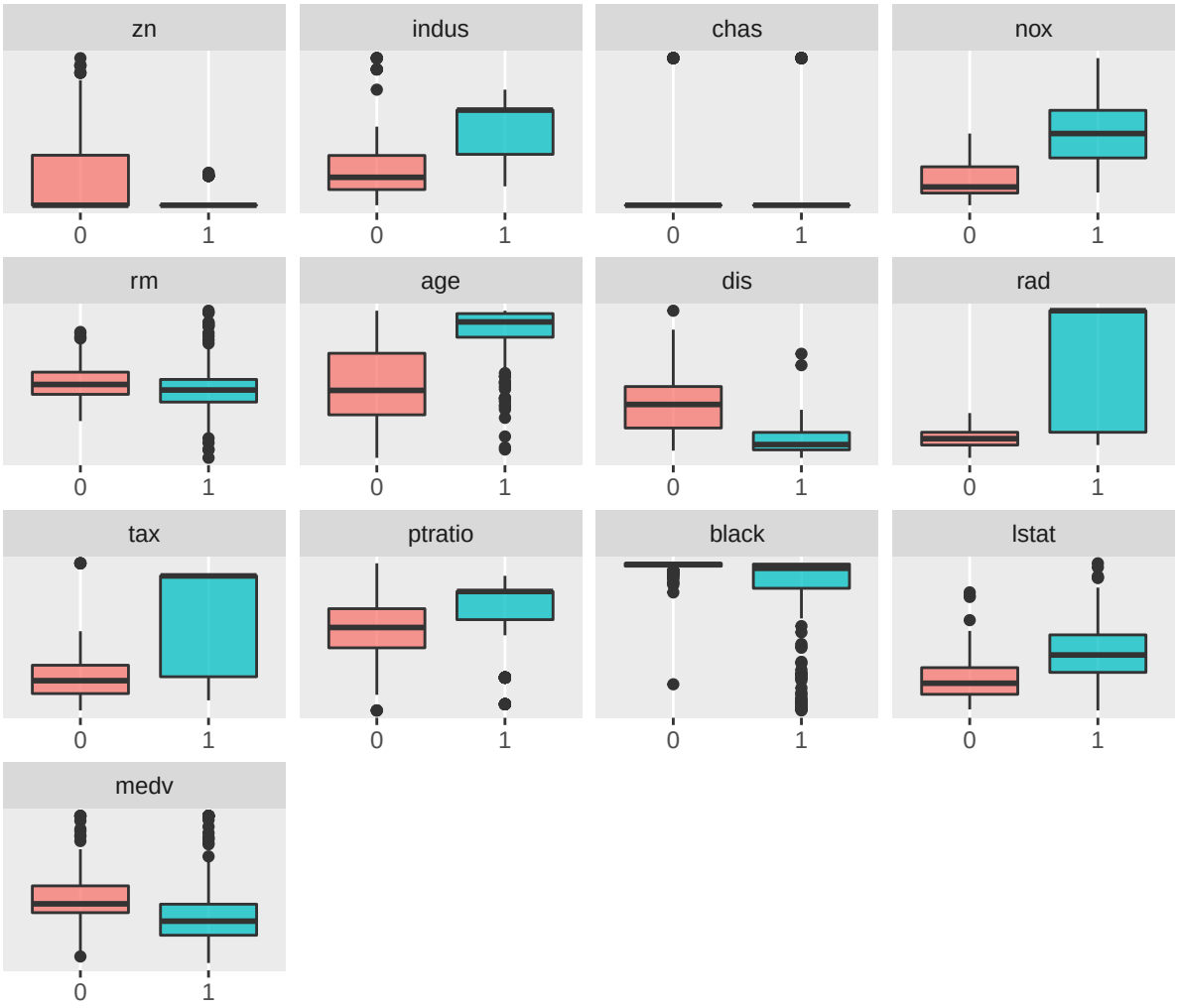
Looking at the box plots above, we can see that the `zn` and `black` predictors are strongly skewed. `chas`, indicating if a suburb borders the Charles River, is also strongly skewed, but like our response variable, it is a two-level categorical predictor and the skew shows because 92% of the values are 0.



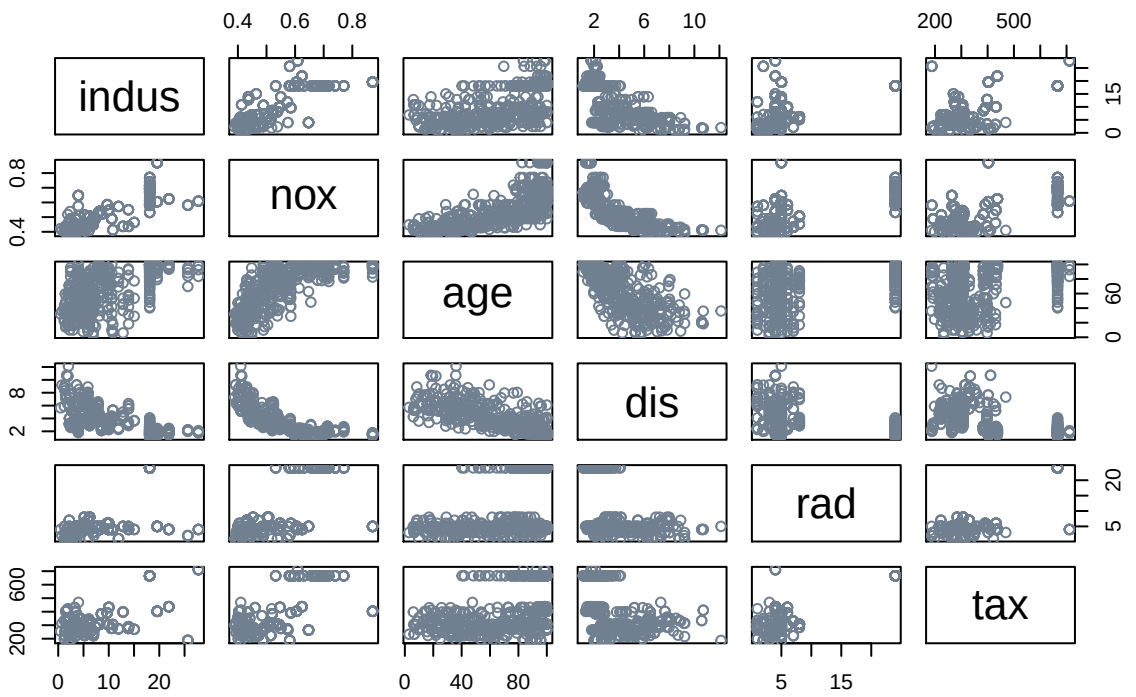




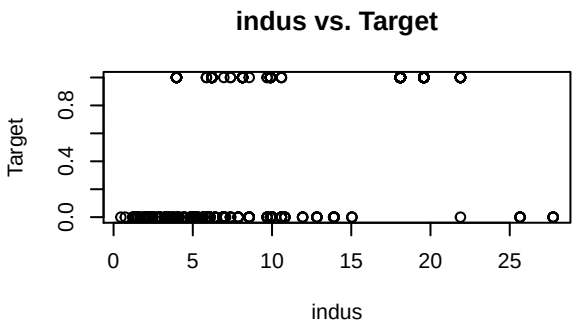
Since our response variable `target` is a two-level factor, we can take a look at a plot of each predictor, subset by `target` and see the relationship between the predictor and our response variable. Right away, the difference in means for some of the variables is quite apparent. These variables also have a higher corrleation to `target`, and may be some of the more significant predictors in our models.

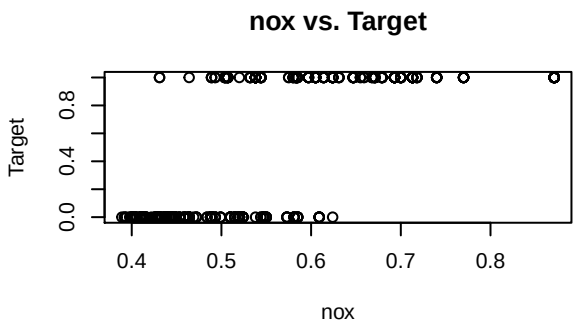


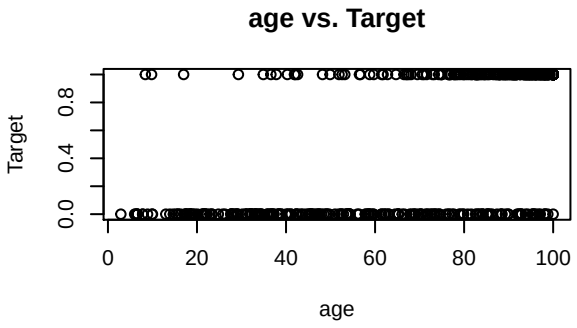
The predictors with the highest correlation to `target` are `indus`, `nox`, `age`, `dis`, `rad`, and `tax`, and all are related to each other in some form when we consider industrial zoning and the effects it has on a surrounding area. In fact, looking at these variables against each other, we see that they are highly correlated as well, evidence that multicollinearity exists.

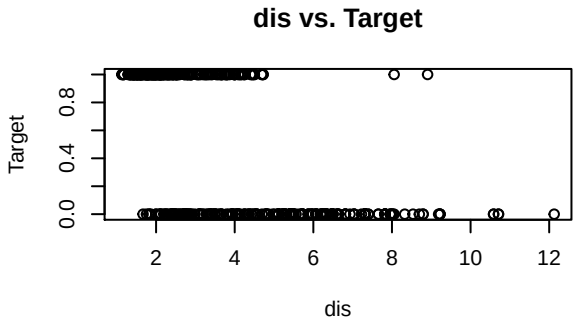


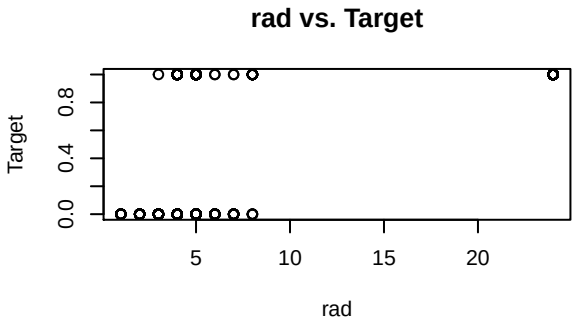
Scatterplots are shown above for those variables with higher correlations and larger differences in means when subset by the `target` response variable. Certain variables, such as `nox` seem to show great correlation with many other variables and it is hypothesized that `nox` will be a significant predictor in the final predictive model.

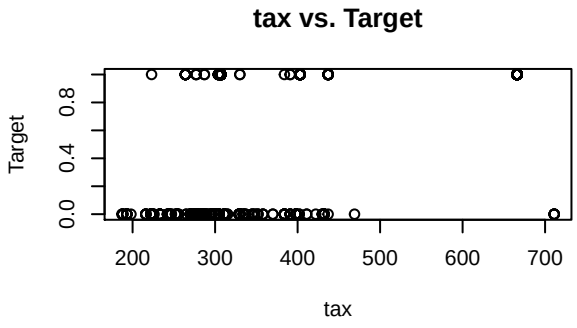

indus vs. Target
Target
0.0
0.4
0.8
0
5
10
15
20
25
indus


nox vs. Target
Target
0.0
0.4
0.8
0.4
0.5
0.6
0.7
0.8
nox


age vs. Target
Target
0.0
0.4
0.8
0
20
40
60
80
100
age


dis vs. Target
Target
0.0
0.4
0.8
2
4
6
8
10
12
dis


rad vs. Target
Target
0.0
0.4
0.8
5
10
15
20
rad


tax vs. Target
Target
0.0
0.4
0.8
200
300
400
500
600
700
tax

Data Preparation

As stated in Part 1, there are no missing values in the training data set, so we do not need to impute any of the variables, or exclude any cases in the data set. The predictor `black`, which measures the proportion of blacks by town appears to already have been transformed (adding a constant value, squaring, then multiplying by 1000). Undoing this transformation simply results in data that is even more skewed against the response variable, so the transformation was done to reduce this.

Combining other predictors or using ratios did not seem to create any new meaningful predictors that would improve our models, however due to the left- and right-skewness of some of our predictors, transformations like log, square root, or 1/X reduce the amount of skew and increase the correlation of our predictors to the response.

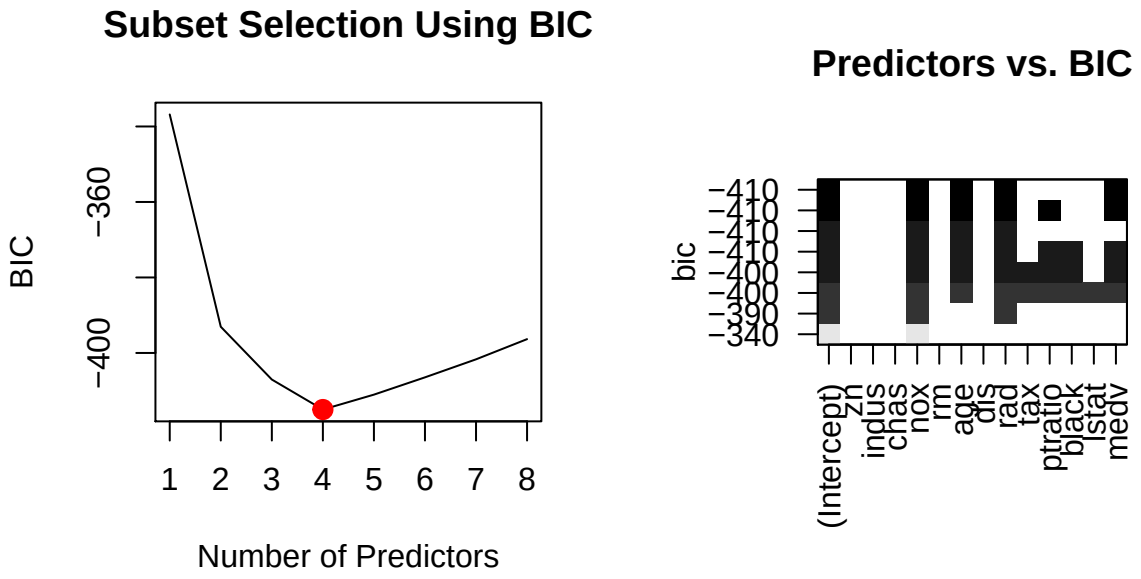
In the table below, we can see the effect of each of the transformations on the correlation to the `target` variable. Many of the transformations do not help in improving the correlation, but the log transformation of `nox` and `rad` may fit our models better, as well as the square root of `indus`.

	No Transform	LOG	SQ.ROOT	1/X
zn	-0.43	NA	-0.47	NA
indus	0.6	0.6	0.62	-0.44
chas	0.08	NA	0.08	NA
nox	0.73	0.75	0.74	-0.75
rm	-0.15	-0.17	-0.16	0.19
age	0.63	0.54	0.6	-0.33
dis	-0.62	-0.66	-0.64	0.64
rad	0.63	0.64	0.64	-0.49
tax	0.61	0.61	0.61	-0.6
ptratio	0.25	0.22	0.24	-0.19
black	-0.35	-0.27	-0.32	0.08
lstat	0.47	0.44	0.47	-0.32
medv	-0.27	-0.36	-0.32	0.38
target	1	NA	1	NA

Model Creation

Model 1: Bayesian Information Criterion

The first model created utilizes the Bayesian Information Criterion (BIC) to determine the number of predictors to use and which predictors should be used.



The left plot above shows that the BIC is minimized using 4 predictors. The plot on the right shows that the 4 predictors with the lowest BIC are `nox`, `age`, `rad`, and `medv`. As such, a model is created using these predictors; this model is presented below.

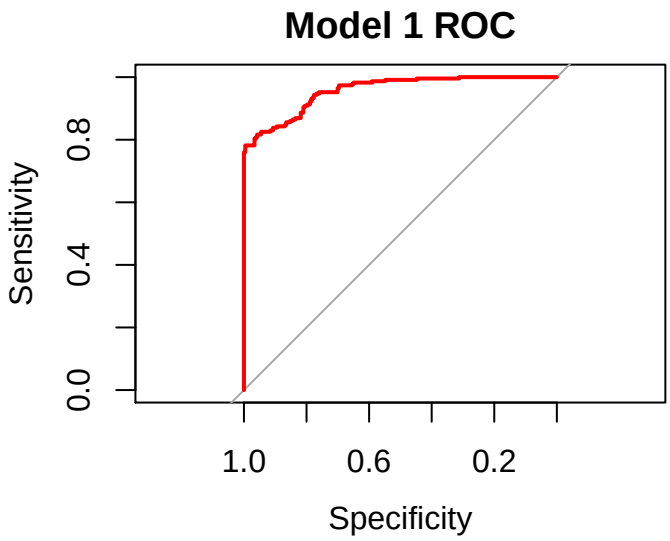
	Estimate	Std. Error	z value	Pr(>|z|)
nox	23.62	3.936	6.003	1.942e-09
age	0.01824	0.009172	1.989	0.04673
rad	0.4528	0.1093	4.144	3.413e-05
medv	0.04481	0.02319	1.932	0.05338
(Intercept)	-17.63	2.168	-8.131	4.246e-16

(Dispersion parameter for binomial family taken to be 1)

Null deviance:	645.9 on 465 degrees of freedom
Residual deviance:	232.8 on 461 degrees of freedom

The coefficients in this model indicate that nitrogen oxides concentration has the strongest, as well as the most statistically significant, effect on on the target variable. Age, highway access, and home value all have far weaker effects on the target. All of the estimated coefficients are statistically significant at the $\alpha = 0.5$ level except `medv` – the p-value for this coefficient is 0.0534.

The receiver operating characteristic (ROC) curve and confusion matrix returned by this model are shown below:

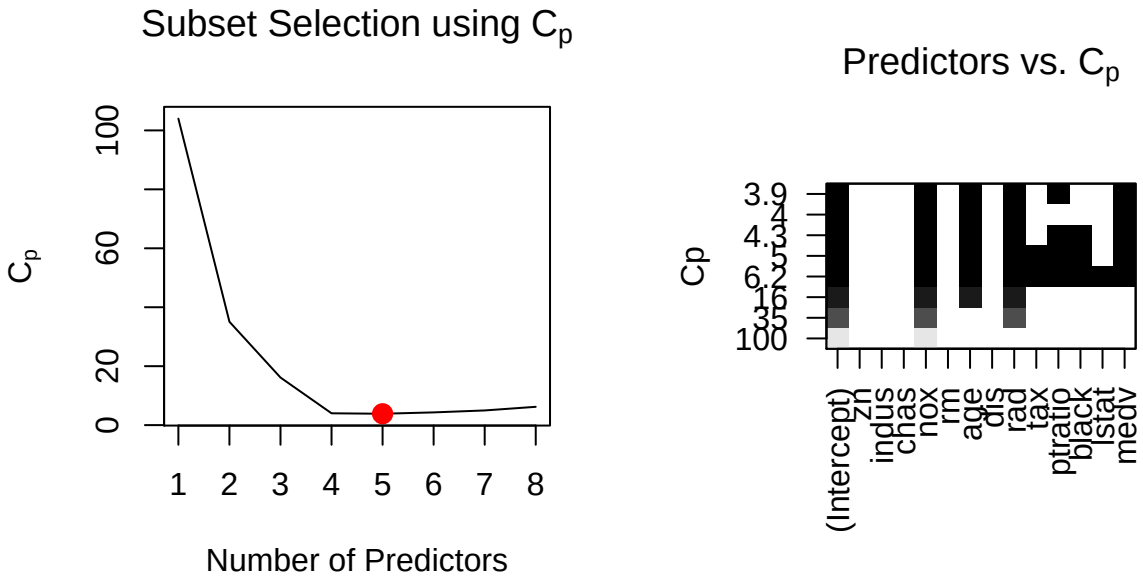


	0	1
0	214	23
1	37	192

This model provides an area under the curve of 0.9570 and an accuracy of 0.8691.

Model 2: Mallow's C_p

The second model created utilizes Mallow's C_p to determine the number of predictors to use and which predictors should be used.



The left plot above shows that C_p is minimized using 5 predictors. The plot on the right shows that the 5 predictors with the lowest C_p are `nox`, `age`, `rad`, `ptratio`, and `medv`. As such, a model is created using these predictors; this model is presented below.

	Estimate	Std. Error	z value	Pr(>|z|)
nox	25.33	4.084	6.203	5.53e-10
age	0.0194	0.009308	2.085	0.03711
rad	0.5126	0.1148	4.464	8.027e-06

	Estimate	Std. Error	z value	Pr(>|z|)
ptratio	0.2742	0.09874	2.777	0.005486
medv	0.08544	0.02798	3.054	0.002259
(Intercept)	-24.94	3.683	-6.77	1.289e-11

(Dispersion parameter for binomial family taken to be 1)

Null deviance:	645.9 on 465 degrees of freedom
Residual deviance:	224.7 on 460 degrees of freedom

As in model 1, the coefficients in this model indicate that nitrogen oxides concentration has the strongest, as well as the most statistically significant, effect on on the target variable. This may be due to the fact that a single part per 10 million in concentration merits a unit increase in this variable. Each of the estimated coefficients are statistically signficant.

The ROC curve and confusion matrix returned by this model are shown below:

This model provides an area under the curve of 0.9605 and an accuracy of 0.8691.

Model 3: Transformed BIC

Model 3 is created using the same BIC selection from Model 1, with modification from transformation. Based on the distributions of the `nox` and `rad` variables, log transformations of these variables are used. The model using these transformed variables, the model is presented below.

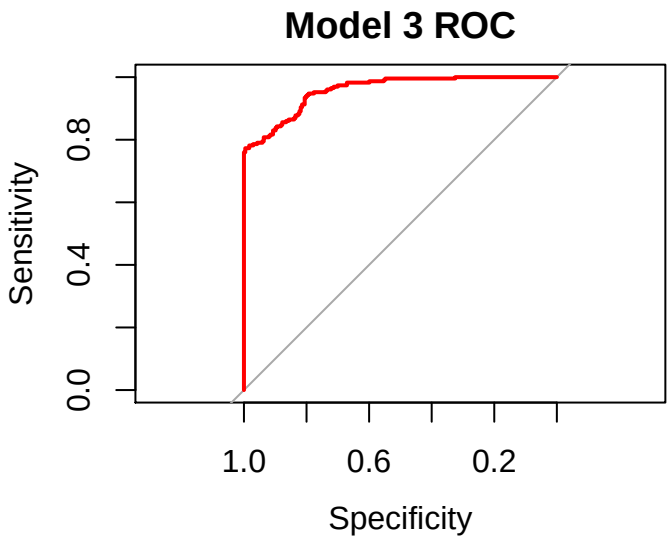
	Estimate	Std. Error	z value	Pr(>|z|)
log(nox)	13	2.097	6.199	5.681e-10
age	0.01696	0.009147	1.855	0.06366
log(rad)	2.269	0.4494	5.05	4.428e-07
medv	0.0489	0.02354	2.078	0.03774
(Intercept)	1.96	1.925	1.018	0.3086

(Dispersion parameter for binomial family taken to be 1)

Null deviance:	645.9 on 465 degrees of freedom
Residual deviance:	231.7 on 461 degrees of freedom

In model 3, the coefficient associated with nitrogen oxide concentration is once again the largest in magnitude, although it has decreased in magnitude. The estimated coefficient for home value is now statistically signficant; however, the coefficient for age no longer is.

The ROC curve and confusion matrix returned by this model are shown below:



	0	1
0	213	24
1	37	192

This model has an area under the curve of 0.9584 and an accuracy of 0.8691.

Model 4: Significant BIC

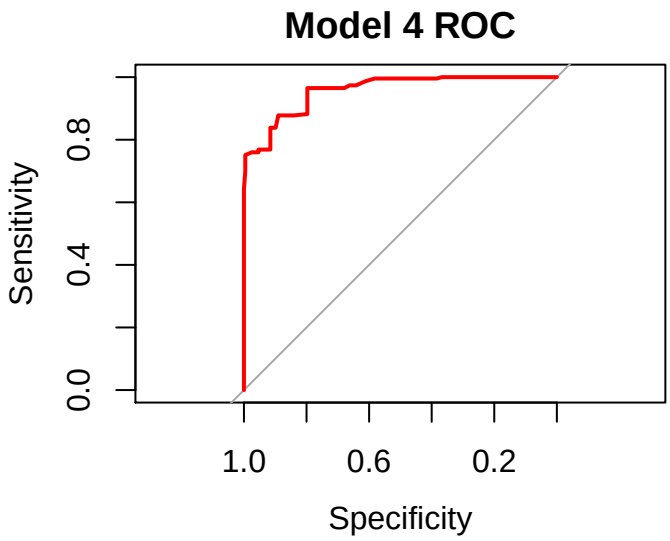
Like Model 1, this model is formed using best subsets regression with BIC as a criterion for choosing the number of predictors; however, with this model, predictors are removed sequentially until all predictors are statistically significant ($p < 0.05$). This leads to the removal of age and median home values as predictors.

	Estimate	Std. Error	z value	Pr(>|z|)
nox	27.2	3.232	8.415	3.915e-17
rad	0.5139	0.1082	4.75	2.036e-06
(Intercept)	-17.45	1.949	-8.956	3.376e-19

(Dispersion parameter for binomial family taken to be 1)

Null deviance:	645.9 on 465 degrees of freedom
Residual deviance:	239.5 on 463 degrees of freedom

The coefficients for this model are similar to the coefficients associated with the two variables in the larger BIC model (Model 1), but show far greater statistical significance with the additional variables removed.



	0	1
0	213	24
1	37	192

This model has an area under the curve of 0.9575 and an accuracy of 0.8691.

Model 5: Best GLM

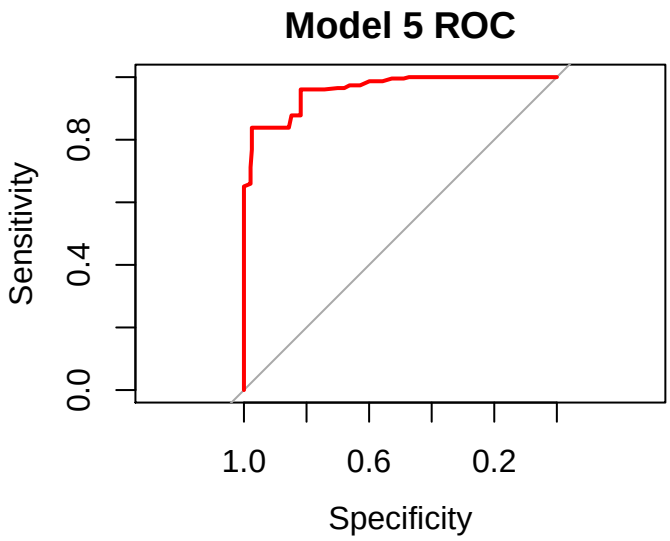
The `bestglm` package is implemented to "[select] the best subset of inputs for the GLM family." The BIC is still used as the information criterion. The model generated by the package is presented below.

	Estimate	Std. Error	z value	Pr(>|z|)
nox	35.63	4.524	7.877	3.35e-15
rad	0.6376	0.1194	5.338	9.377e-08
tax	-0.008146	0.002332	-3.493	0.0004776
(Intercept)	-19.87	2.368	-8.389	4.911e-17

(Dispersion parameter for binomial family taken to be 1)

Null deviance:	645.9 on 465 degrees of freedom
Residual deviance:	224.5 on 462 degrees of freedom

The coefficients for this model again indicate the strong influence of a unit increase in nitrogen oxide concentration. The inclusion of property tax rate in the model lead to the first negative coefficient seen. Each of the coefficient estimates is of very high statistical significance.



	0	1
0	222	15
1	37	192

This model has an area under the curve of 0.8876 and an accuracy of 0.8691.

Model 6: Transformed Best GLM

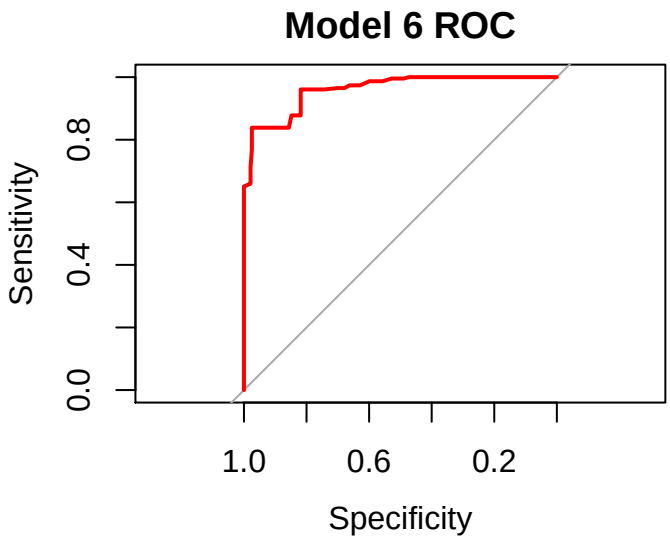
Using the three variables identified by the `bestglm` function for Model 5, the `nox` and `rad` predictors are transformed using logarithms, as in Model 3. The model using these transformations is presented below.

	Estimate	Std. Error	z value	Pr(>|z|)
log(nox)	19.35	2.519	7.682	1.564e-14
log(rad)	3.356	0.5447	6.162	7.196e-10
tax	-0.008214	0.002335	-3.518	0.0004345
(Intercept)	9.385	1.829	5.132	2.866e-07

(Dispersion parameter for binomial family taken to be 1)

Null deviance:	645.9 on 465 degrees of freedom
Residual deviance:	223.5 on 462 degrees of freedom

The coefficient for nitrogen oxide concentration decreased following the transformation, as did the standard error of the estimate of the coefficient. While the p-value related to the coefficients changed following the tranformation, all estimates remain statistically significant.



	0	1
0	222	15
1	37	192

This model has an area under the curve of 0.9594 and an accuracy of 0.8884.

Model 7: Full Model

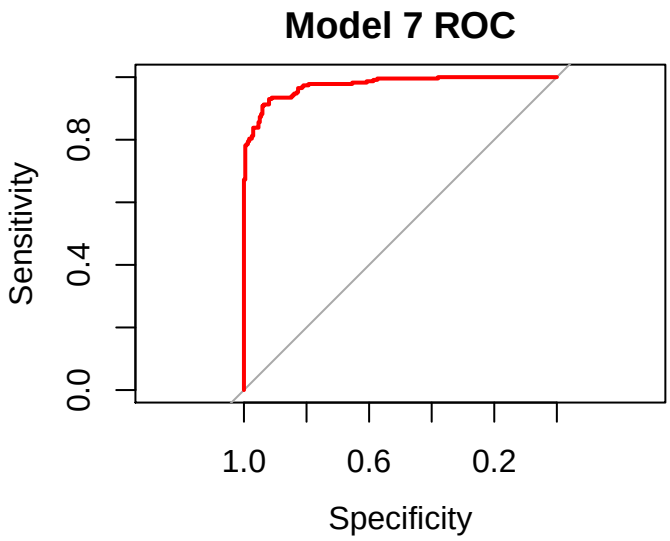
For completeness, a full generalized linear model is created using all explanatory variables.

	Estimate	Std. Error	z value	Pr(>|z|)
zn	-0.06172	0.03441	-1.794	0.07287
indus	-0.07258	0.04855	-1.495	0.1349
chas	1.032	0.7596	1.359	0.1741
nox	50.16	8.05	6.231	4.623e-10
rm	-0.6921	0.7414	-0.9335	0.3505
age	0.03452	0.01388	2.487	0.01289
dis	0.7658	0.2344	3.267	0.001087
rad	0.663	0.1651	4.015	5.945e-05
tax	-0.006593	0.003064	-2.152	0.03142
ptratio	0.4422	0.1322	3.344	0.0008252
black	-0.01309	0.00668	-1.96	0.04997
lstat	0.04757	0.05451	0.8727	0.3828
medv	0.1997	0.07102	2.812	0.004919
(Intercept)	-36.84	7.029	-5.241	1.595e-07

(Dispersion parameter for binomial family taken to be 1)

Null deviance:	645.9 on 465 degrees of freedom
Residual deviance:	186.1 on 452 degrees of freedom

The 13 coefficients in the full model vary very widely in magnitude, and five of the coefficients are negative. There is also a wide range of signifcances – 5 of the coefficients are not significant under any reasonable α, and one is very nearly exactly 0.05. It is interesting to note that the coefficient for `indus` is negative, while the correlation of this variable with `target` is relatively strongly positive – this may be due to collinearity between `indus` and `nox` – highly industrialized neighborhoods will likely have higher air pollution.



	0	1
0	222	15
1	20	209

This model has an area under the curve of 0.9753 and an accuracy of 0.9249.

Model Selection and Prediction

Model Comparison

The characteristics and performance of the seven models from the previous section are compared below:

Model #	# of Predictors	AUC	Accuracy
1	4	0.9570	0.8712
2	5	0.9605	0.8691
3	4	0.9584	0.8691
4	2	0.9575	0.8691
5	3	0.8876	0.8691
6	3	0.9594	0.8884
7	13	0.9753	0.9249

The model using only the significant predictors as determined by BIC (Model 3) was selected as the best model for prediction of `TARGET` in the crime data set. While the AUC value of this model was not the highest of the models tested, its mean cross-validation error indicates that it is has the best predictive value for unseen data. Additionally, it is a parsimonious model, and the simplicity lends itself to easier understanding of the model by other users.

10-fold Cross Validation

Mean CV Error

Model1	36.6
Model2	46.69
Model3	15.48
Model4	22.14
Model5	55.49
Model6	21.8
Model7	93.87

The linear model is applied to a test dataset containing response variables for 40 cases. A table of the predicted team wins is presented below.

Predicted Targets for Test Data

0	1
24	16

Targets for Training Data

0	1
237	229

Similar to the training dataset, the predictions for the test data set predictions are weighted more toward crime being below the median. It is important to note that Model 3 did not achieve the highest AUC or accuracy on the training data but, given its cross-validation performance, we believe that Model 3 will produce the greatest generalizability of the models explored here.

A comparison of the full sets of predictions for the evaluation dataset is available in Appendix A.

Appendix A – Index-wise Results from Predictive Model

Index	Predicted Value	Index	Predicted Value
1	0	21	0
2	0	22	0
3	0	23	0
4	0	24	0
5	0	25	0
6	0	26	0
7	0	27	0
8	0	28	1
9	0	29	1
10	0	30	1
11	0	31	1
12	0	32	1
13	1	33	1
14	1	34	1
15	1	35	1
16	0	36	1
17	0	37	1
18	1	38	1
19	0	39	1
20	0	40	0

Appendix B – R Code

```
library(knitr)
opts_chunk$set(echo = FALSE, warning = FALSE, message = FALSE,
    comment = NA, fig.align = "center")
knitr::opts_chunk$set(error = TRUE)

library(stringr)
library(pander)
library(ggplot2)
library(gridExtra)
library(reshape2)
library(MASS)
library(leaps)
library(pROC)
library(caret)
library(bestglm)
library(ggplot2)
library(reshape2)
library(dplyr)
library(e1071)

train_df <- read.csv(url("https://raw.githubusercontent.com/dsmilo/DATA621/master/HW3/Data/crime-training-data

test_df <- read.csv(url("https://raw.githubusercontent.com/dsmilo/DATA621/master/HW3/Data/crime-evaluation-dat


mai = c(1, 0.1, 0.1, 0.1)
# Stuff for Part 1 table/figure preparation Summary Table

means <- sapply(train_df, mean)
medians <- sapply(train_df, median)
IQRs <- sapply(train_df, IQR)
skews <- sapply(train_df, skewness)
cors <- as.vector(cor(train_df$target, train_df[, 1:ncol(train_df)]))

header <- (c("MEAN", "MEDIAN", "IQR", "SKEWNESS", "CORRELATION TO TARGET"))

datasummary <- as.data.frame(cbind(means, medians, IQRs, skews,
    cors))
datasummary <- round(datasummary, 2)
colnames(datasummary) <- header
pander(datasummary)

# Boxpots with ggplot2

m <- melt(train_df, variable.name = "Predictor")
ggplot(m, aes(Predictor, value)) + geom_boxplot(aes(fill = Predictor),
    alpha = 0.75, show.legend = FALSE) + facet_wrap(~Predictor,
    scale = "free") + scale_y_continuous("") + scale_x_discrete("",
    breaks = NULL) + ggtitle("Distribution of Predictor and Target Variables\n")

par(mfrow = c(3, 1))
hist(train_df$zn, breaks = 5, col = "slategrey", main = "Hist. of 'zn' Predictor")
hist(train_df$black, breaks = 5, col = "slategrey", main = "Hist. of 'black' Predictor")
hist(train_df$chas, col = "slategrey", main = "Hist. of 'chas' Predictor")
par(mfrow = c(1, 1))

# Boxplot of predictors split by Target value (0 or 1)
```

```r
n <- melt(train_df, id.vars = "target", variable.name = "Predictor")
n$target <- factor(n$target)
ggplot(n, aes(target, value)) + geom_boxplot(aes(fill = target),
    alpha = 0.75) + facet_wrap(~Predictor, scale = "free") +
    scale_fill_discrete(guide = FALSE) + scale_y_continuous("",
    labels = NULL, breaks = NULL) + scale_x_discrete("") + ggtitle("Distribution of Predictors by Target\n")
pairs(~indus + nox + age + dis + rad + tax, data = train_df,
    main = "Highly Correlated Predictors", col = "slategrey")

# scatterplot of highly correlated predictors against Target

par(mfrow = c(3, 2))
for (i in c(2, 4, 6:9)) {
    plot(train_df[, i], train_df$target, main = paste(names(train_df[i]),
        "vs. Target"), xlab = names(train_df)[i], ylab = "Target")
}
par(mfrow = c(1, 1))

# Some of the values in each predictor are leverage points,
# and due to the outliers in each variable, may be bad
# leverage points (standard residuals with abs. value > 2)
# depending on what predictors are used in our models.
# Transformations on the predictors with higher skewness may
# elminate these leverage points as well.  Scatterplots of
# predictors against response for appendix

par(mfrow = c(4, 3), mar = c(1, 1, 1, 1))
for (i in c(1, 2, 4:13)) {
    plot(train_df[, i], train_df$target, main = paste(names(train_df[i]),
        "vs. Target"), xlab = names(train_df)[i], ylab = "Target")
}

# `chas` plotted separately since it is a two-level factor

par(mfrow = c(1, 1))
plot(jitter(train_df$chas), jitter(train_df$target, amount = 0.005),
    main = "Charles River vs.Target", xlab = "chas", ylab = "target")  # jitter applied to prevent points in c

# Transformation Table

cors2 <- as.vector(cor(train_df$target, train_df[, 1:ncol(train_df)]))
log_cors <- as.vector(cor(train_df$target, log(train_df[, 1:ncol(train_df)])))
sqrt_cors <- as.vector(cor(train_df$target, sqrt(train_df[, 1:ncol(train_df)])))
recip_cors <- as.vector(cor(train_df$target, (1/train_df[, 1:ncol(train_df)])))

header2 <- (c("No Transform", "LOG", "SQ.ROOT", "1/X"))

transforms <- as.data.frame(cbind(cors2, log_cors, sqrt_cors,
    recip_cors))
transforms <- round(transforms, 2)
rownames(transforms) <- colnames(train_df)
colnames(transforms) <- header2
pander(transforms)

# Only for comparison

summary(glm(target ~ nox, data = train_df, family = binomial))
summary(glm(target ~ log(nox), data = train_df, family = binomial))  # results in lower AIC
```

```r
summary(glm(target ~ indus, data = train_df, family = binomial))
summary(glm(target ~ I(indus^0.5), data = train_df, family = binomial))

train_df$target <- factor(train_df$target)  # to factor

regfit.full <- regsubsets(factor(target) ~ ., data = train_df)

par(mfrow = c(1, 2))

reg.summary <- summary(regfit.full)
plot(reg.summary$bic, xlab = "Number of Predictors", ylab = "BIC",
    type = "l", main = "Subset Selection Using BIC")
BIC_num <- which.min(reg.summary$bic)
points(BIC_num, reg.summary$bic[BIC_num], col = "red", cex = 2,
    pch = 20)

plot(regfit.full, scale = "bic", main = "Predictors vs. BIC")
par(mfrow = c(1, 1))


model1 <- glm(target ~ nox + age + rad + medv, family = binomial,
    data = train_df)

train_df$predicted_model1 <- predict(model1, train_df, type = "response")
train_df$target_model1 <- ifelse(train_df$predicted_model1 >
    0.5, 1, 0)

pander(summary(model1))

roc_model1 <- roc(target ~ predicted_model1, data = train_df)

plot_roc <- plot(roc_model1, col = "red", main = "Model 1 ROC")

pander(confusionMatrix(train_df$target, train_df$target_model1,
    positive = "1")$table)
# AUC is 0.957 Accuracy : 0.8712

par(mfrow = c(1, 2))

plot(reg.summary$cp, xlab = "Number of Predictors", ylab = expression("C"[p]),
    type = "l", main = expression("Subset Selection using C"[p]))
Cp_num <- which.min(reg.summary$cp)
points(Cp_num, reg.summary$cp[Cp_num], col = "red", cex = 2,
    pch = 20)

plot(regfit.full, scale = "Cp", main = expression("Predictors vs. C"[p]))
par(mfrow = c(1, 1))

model2 <- glm(target ~ nox + age + rad + ptratio + medv, family = binomial,
    data = train_df)

train_df$predicted_model2 <- predict(model2, train_df, type = "response")
train_df$target_model2 <- ifelse(train_df$predicted_model2 >
    0.5, 1, 0)

pander(summary(model2))

roc_model2 <- roc(factor(target) ~ predicted_model2, data = train_df)
```

```
plot_roc <- plot(roc_model2, col = "red", main = "Model 2 ROC")

pander(confusionMatrix(train_df$target, train_df$target_model2,
    positive = "1")$table)
# AUC is 0.9605 Accuracy : 0.8691

model3 <- glm(target ~ log(nox) + age + log(rad) + medv, family = binomial,
    data = train_df)

train_df$predicted_model3 <- predict(model3, train_df, type = "response")
train_df$target_model3 <- ifelse(train_df$predicted_model3 >
    0.5, 1, 0)

pander(summary(model3))

roc_model3 <- roc(factor(target) ~ predicted_model3, data = train_df)

plot_roc <- plot(roc_model3, col = "red", main = "Model 3 ROC")
pander(confusionMatrix(train_df$target, train_df$target_model3,
    positive = "1")$table)

# AUC is 0.9584, Accuracy : 0.8691

model4 <- glm(target ~ nox + rad, data = train_df, family = binomial(link = "logit"))

train_df$predicted_model4 <- predict(model4, train_df, type = "response")
train_df$target_model4 <- ifelse(train_df$predicted_model4 >
    0.5, 1, 0)

pander(summary(model4))

roc_model4 <- roc(factor(target) ~ predicted_model4, data = train_df)

plot_roc <- plot(roc_model4, col = "red", main = "Model 4 ROC")
pander(confusionMatrix(train_df$target, train_df$target_model4,
    positive = "1")$table)

# AUC is 0.9575, Accuracy: 0.8691

# bestglm_bic <- bestglm(train_df, IC= 'BIC', family =
# binomial) #very slow - do not evaluate

model5 <- glm(target ~ nox + rad + tax, family = binomial, data = train_df)

train_df$predicted_model5 <- predict(model5, train_df, type = "response")
train_df$target_model5 <- ifelse(train_df$predicted_model5 >
    0.5, 1, 0)

pander(summary(model5))

roc_model5 <- pROC::roc(factor(target) ~ predict(model5, train_df,
    type = "response"), data = train_df)

plot_roc <- plot(roc_model5, col = "red", main = "Model 5 ROC")
pander(confusionMatrix(train_df$target, train_df$target_model5,
    positive = "1")$table)

# AUC is 0.8876, Accuracy : 0.8691
```

```r
model6 <- glm(target ~ log(nox) + log(rad) + tax, family = binomial,
    data = train_df)

train_df$predicted_model6 <- predict(model5, train_df, type = "response")
train_df$target_model6 <- ifelse(train_df$predicted_model6 >
    0.5, 1, 0)

pander(summary(model6))

roc_model6 <- roc(factor(target) ~ predicted_model6, data = train_df)

plot_roc <- plot(roc_model6, col = "red", main = "Model 6 ROC")
pander(confusionMatrix(train_df$target, train_df$target_model6,
    positive = "1")$table)

# AUC is 0.9594, Accuracy : 0.8884

model7 <- glm(target ~ zn + indus + chas + nox + rm + age + dis +
    rad + tax + ptratio + black + lstat + medv, data = train_df,
    family = binomial(link = "logit"))

train_df$predicted_model7 <- predict(model7, train_df, type = "response")
train_df$target_model7 <- ifelse(train_df$predicted_model7 >
    0.5, 1, 0)

pander(summary(model7))

roc_model7 <- roc(factor(target) ~ predicted_model7, data = train_df)

plot_roc <- plot(roc_model7, col = "red", main = "Model 7 ROC")
pander(confusionMatrix(train_df$target, train_df$target_model7,
    positive = "1")$table)

# AUC 0.9753; Accuracy 0.9249

k = 10
set.seed(1306)
folds = sample(1:k, nrow(train_df), replace = TRUE)
cv.errors1 = matrix(NA, k, 10, dimnames = list(NULL, paste(1:10)))
cv.errors2 = matrix(NA, k, 10, dimnames = list(NULL, paste(1:10)))
cv.errors3 = matrix(NA, k, 10, dimnames = list(NULL, paste(1:10)))
cv.errors4 = matrix(NA, k, 10, dimnames = list(NULL, paste(1:10)))
cv.errors5 = matrix(NA, k, 10, dimnames = list(NULL, paste(1:10)))
cv.errors6 = matrix(NA, k, 10, dimnames = list(NULL, paste(1:10)))
cv.errors7 = matrix(NA, k, 10, dimnames = list(NULL, paste(1:10)))

train_df$target = as.numeric(as.character(train_df$target))

for (j in 1:k) {

    model1 <- glm(target ~ nox + age + rad + medv, family = binomial,
        data = train_df[folds != j, ])
    model2 <- glm(target ~ nox + age + rad + ptratio + medv,
        family = binomial, data = train_df[folds != j, ])
    model3 <- glm(target ~ log(nox) + age + log(rad) + medv,
        family = binomial, data = train_df[folds != j, ])
    model4 <- glm(target ~ log(nox) + log(rad) + tax, family = binomial,
        data = train_df[folds != j, ])
    model5 <- glm(target ~ nox + rad + tax, family = binomial,
```

```
        data = train_df)
    model6 <- glm(target ~ log(nox) + log(rad) + tax, family = binomial,
        data = train_df)
    model7 <- glm(target ~ zn + indus + chas + nox + rm + age +
        dis + rad + tax + ptratio + black + lstat + medv, data = train_df,
        family = binomial(link = "logit"))


    # best.fit = regsubsets(y ~ ., data = train_df[folds != j, ],
    # nvmax = 10)
    for (i in 1:10) {
        pred1 = predict(model1, train_df[folds == j, ], id = i)
        cv.errors1[j, i] = mean((train_df$target[folds == j] -
            pred1)^2)

        pred2 = predict(model2, train_df[folds == j, ], id = i)
        cv.errors2[j, i] = mean((train_df$target[folds == j] -
            pred2)^2)

        pred3 = predict(model3, train_df[folds == j, ], id = i)
        cv.errors3[j, i] = mean((train_df$target[folds == j] -
            pred3)^2)

        pred4 = predict(model4, train_df[folds == j, ], id = i)
        cv.errors4[j, i] = mean((train_df$target[folds == j] -
            pred4)^2)

        pred5 = predict(model5, train_df[folds == j, ], id = i)
        cv.errors5[j, i] = mean((train_df$target[folds == j] -
            pred5)^2)

        pred6 = predict(model6, train_df[folds == j, ], id = i)
        cv.errors6[j, i] = mean((train_df$target[folds == j] -
            pred6)^2)

        pred7 = predict(model7, train_df[folds == j, ], id = i)
        cv.errors7[j, i] = mean((train_df$target[folds == j] -
            pred7)^2)
    }

}


# cv.errors1
mean.cv.errors1 <- apply(cv.errors1, 2, mean)
# mean.cv.errors1 = apply(cv.errors1, 2, mean)

# which.min(mean.cv.errors1) mean.cv.errors1[6] cv.errors2
mean.cv.errors2 <- apply(cv.errors2, 2, mean)
# mean.cv.errors2 which.min(mean.cv.errors2)
# mean.cv.errors2[6]


# cv.errors3
mean.cv.errors3 <- apply(cv.errors3, 2, mean)
# mean.cv.errors3 which.min(mean.cv.errors3)
# mean.cv.errors3[6]
```

```r
# cv.errors4
mean.cv.errors4 <- apply(cv.errors4, 2, mean)
# mean.cv.errors4 which.min(mean.cv.errors4)
# mean.cv.errors4[6]

mean.cv.errors5 <- apply(cv.errors5, 2, mean)
mean.cv.errors6 <- apply(cv.errors6, 2, mean)
mean.cv.errors7 <- apply(cv.errors7, 2, mean)

all.cv.error = data.frame(mean(mean.cv.errors1), mean(mean.cv.errors2),
    mean(mean.cv.errors3), mean(mean.cv.errors4), mean(mean.cv.errors5),
    mean(mean.cv.errors6), mean(mean.cv.errors7))
names(all.cv.error) = c("Model1", "Model2", "Model3", "Model4",
    "Model5", "Model6", "Model7")
# all.cv.error
all.cv.error = t(all.cv.error)
names(all.cv.error) = c("Model", "Mean CV Error")

pander(all.cv.error)

# evaluation_data <-
# read.csv('https://raw.githubusercontent.com/dsmilo/DATA621/master/HW1/data/moneyball-evaluation-data.csv')
par(mfrow = c(1, 2), pin = c(2, 2))

predicted_wins <- predict(model3, test_df, type = "response")
predicted_wins_bin = ifelse(predicted_wins > 0.5, 1, 0)


pander(table(predicted_wins_bin))

training_wins = train_df$target
pander(table(training_wins))
```