Crime - Logistic Regression

John Cruz, Noori Selina, Shaya Engelman, Daniel Craig, Gavriel Stweinmetz-Silber

2024-03-31

Required Libraries

```r
library(ggplot2)
library(tidyverse)
library(knitr)
library(ggcorrplot)
library(caret)
library(ROCR)
library(MASS)
library(summarytools)
library(latex2exp)
library(janitor)
library(kableExtra)
```

Introduction

Our objective is to explore and build a binary logistic regression model on the training data set to predict whether the neighborhood will be at risk for high crime levels.

An online version is pulished on RPubs

Data Exploration

The training dataset has 466 records (rows) with thirteen (13) variables. All the variables are numeric, except for `chas` being a binary dummy variable.

Predictor Variables

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

Response Variable

- `target`: whether the crime rate is above the median crime rate (1) or not (0)

Import 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

Dimensions:

466 x 13

Missing Values

We have no missing values in our dataset

zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	target
0	0	0	0	0	0	0	0	0	0	0	0	0

Summary Statistics

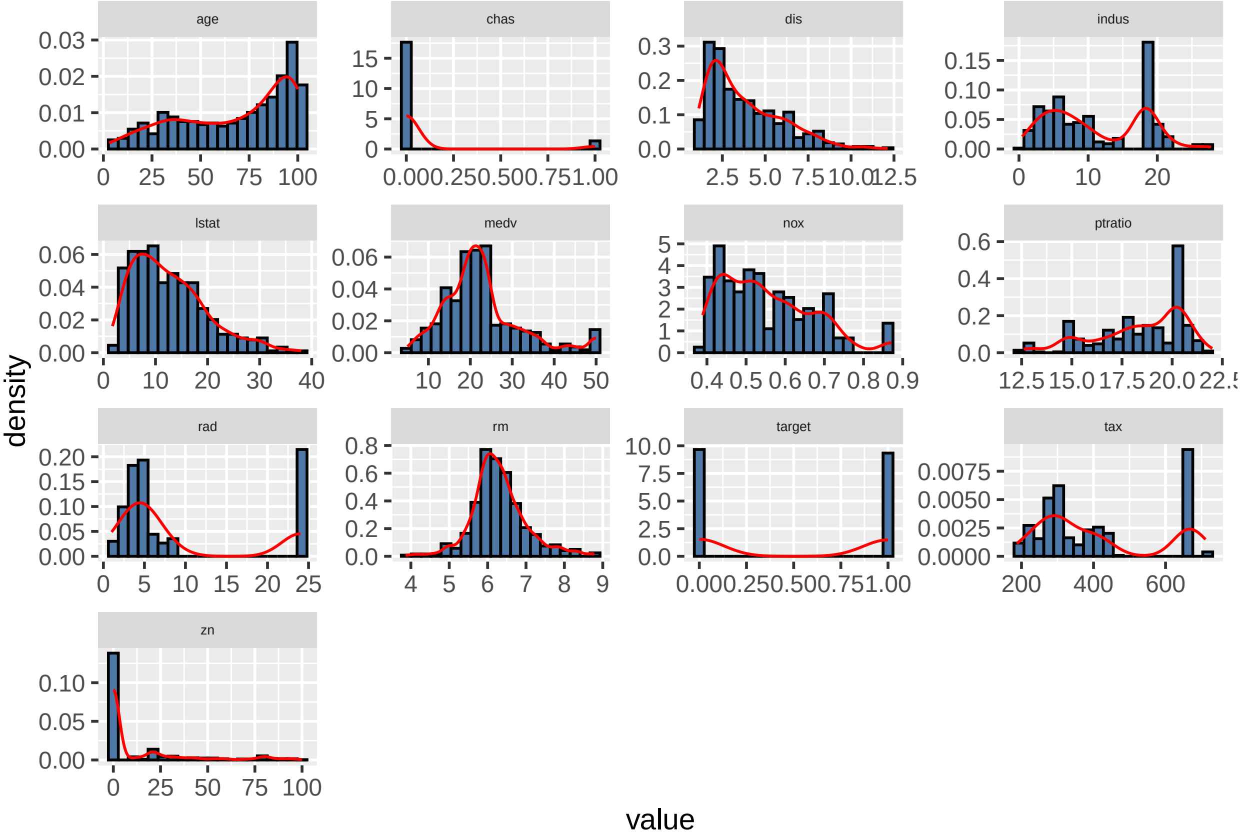
Our table gives us a summary of all our variables. At a quick glance, `age` and `rm` doesn't appear to have any odd value that would be concerning. We also see some significant skewness in some of the variables and they would probably need some type of transformation.

	age	chas	dis	indus	lstat	medv	nox	ptratio	rad	rm	target	tax	zn
Mean	68.37	0.07	3.80	11.11	12.63	22.59	0.55	18.40	9.53	6.29	0.49	409.50	11.58
Std.Dev	28.32	0.26	2.11	6.85	7.10	9.24	0.12	2.20	8.69	0.70	0.50	167.90	23.36
Min	2.90	0.00	1.13	0.46	1.73	5.00	0.39	12.60	1.00	3.86	0.00	187.00	0.00
Q1	43.70	0.00	2.10	5.13	7.01	17.00	0.45	16.90	4.00	5.89	0.00	281.00	0.00
Median	77.15	0.00	3.19	9.69	11.35	21.20	0.54	18.90	5.00	6.21	0.00	334.50	0.00
Q3	94.10	0.00	5.21	18.10	16.94	25.00	0.62	20.20	24.00	6.63	1.00	666.00	17.50
Max	100.00	1.00	12.13	27.74	37.97	50.00	0.87	22.00	24.00	8.78	1.00	711.00	100.00
MAD	30.02	0.00	1.91	9.34	7.07	6.00	0.13	1.93	1.48	0.52	0.00	104.52	0.00
IQR	50.22	0.00	3.11	12.96	9.89	7.98	0.18	3.30	20.00	0.74	1.00	385.00	16.25
CV	0.41	3.63	0.56	0.62	0.56	0.41	0.21	0.12	0.91	0.11	1.02	0.41	2.02
Skewness	-0.58	3.34	1.00	0.29	0.91	1.08	0.75	-0.75	1.01	0.48	0.03	0.66	2.18
SE.Skewness	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11
Kurtosis	-1.01	9.15	0.47	-1.24	0.50	1.37	-0.04	-0.40	-0.86	1.54	-2.00	-1.15	3.81
N.Valid	466.00	466.00	466.00	466.00	466.00	466.00	466.00	466.00	466.00	466.00	466.00	466.00	466.00
Pct.Valid	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

Visualizations

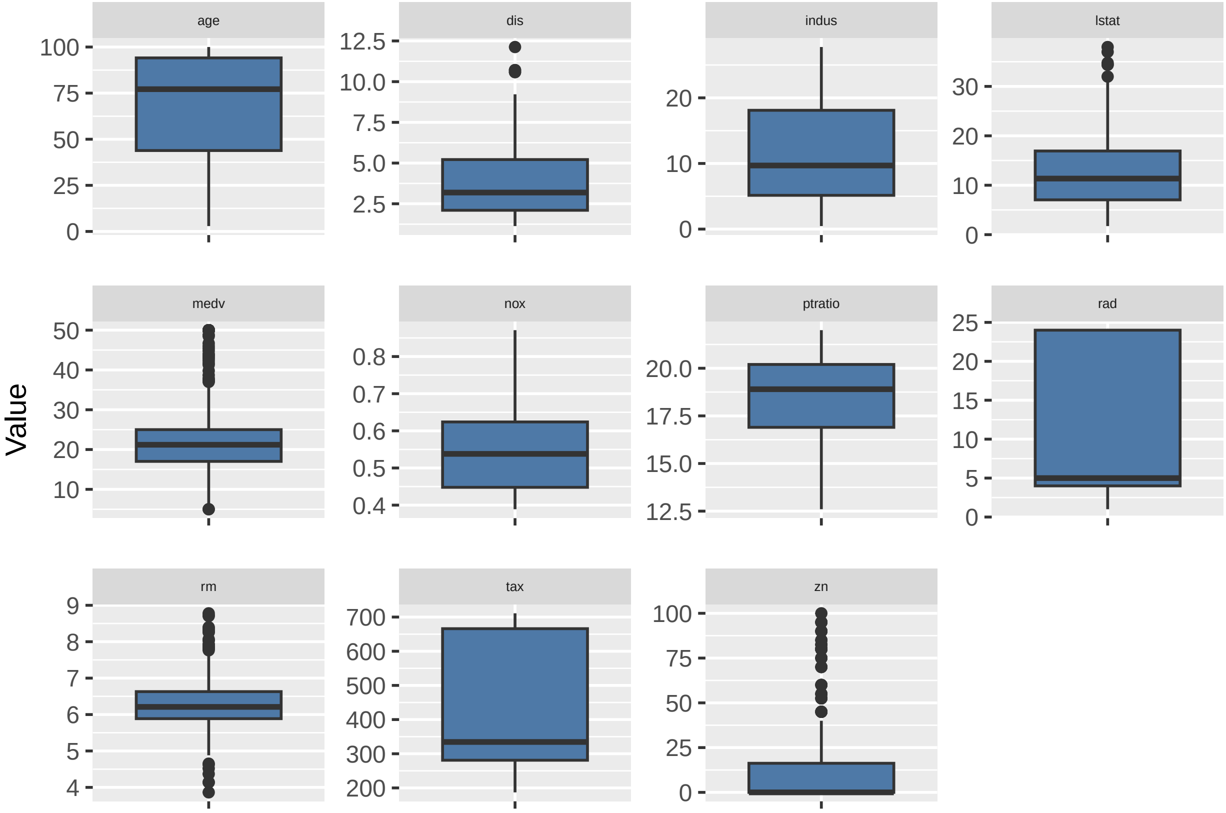
Density

We can get a better idea of the distributions and skewness by plotting our variables. The plots show significant right skew, kurtosis, in `dis`, and `lstat` while we have a left skew in `age` and `pratio`. These skewed variables might be candidates for transformation. The plot also shows `chas` is binary and can only have a value of 0 or 1. Another interesting observation is that variables `rad`, `tax` and possibly `indus` appear to be bimodal. Bimodal data is when we have two or more different classes in a dataset that act as groups.

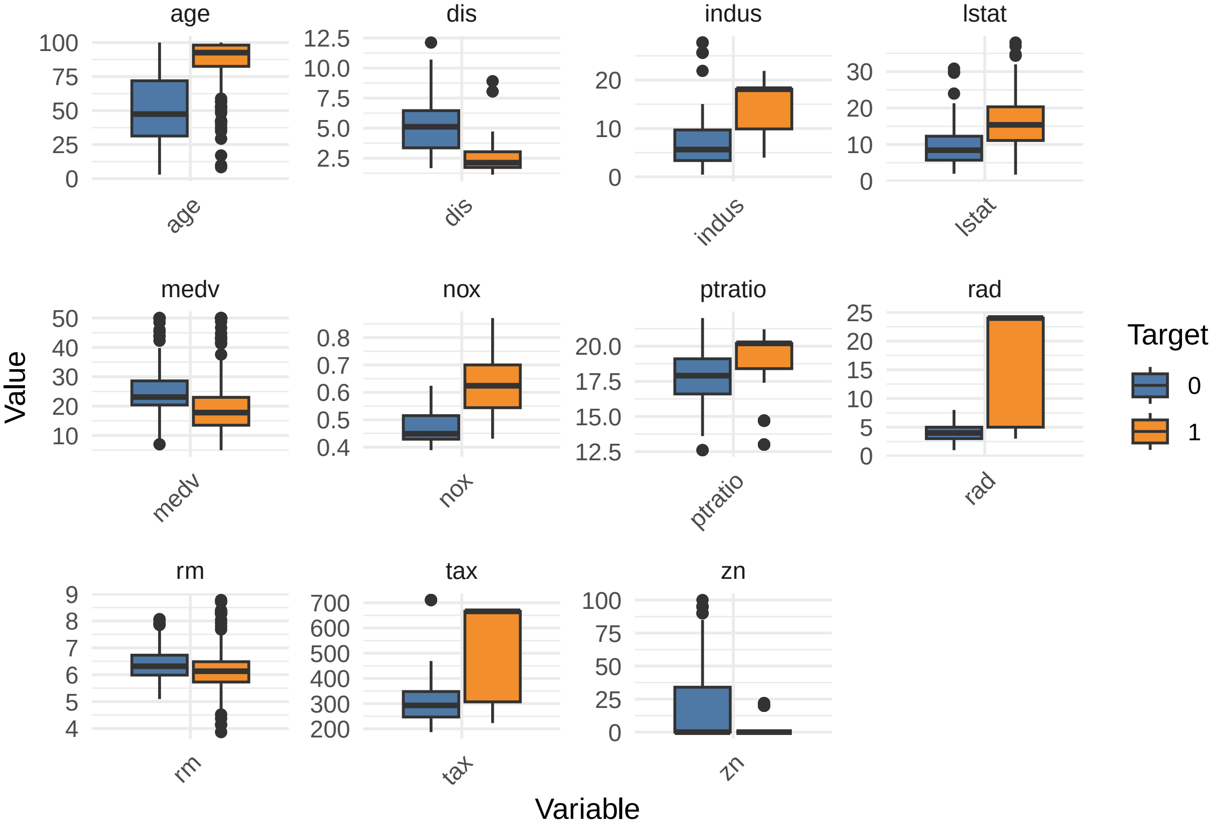


Boxplot

In our density plot some of the variables have wide distributions and many points above the density lines. These boxplots further confirm the skewness mentioned earlier. They also reveal that variables `medv`, `rm` and `zn` all have a large amount of outliers.



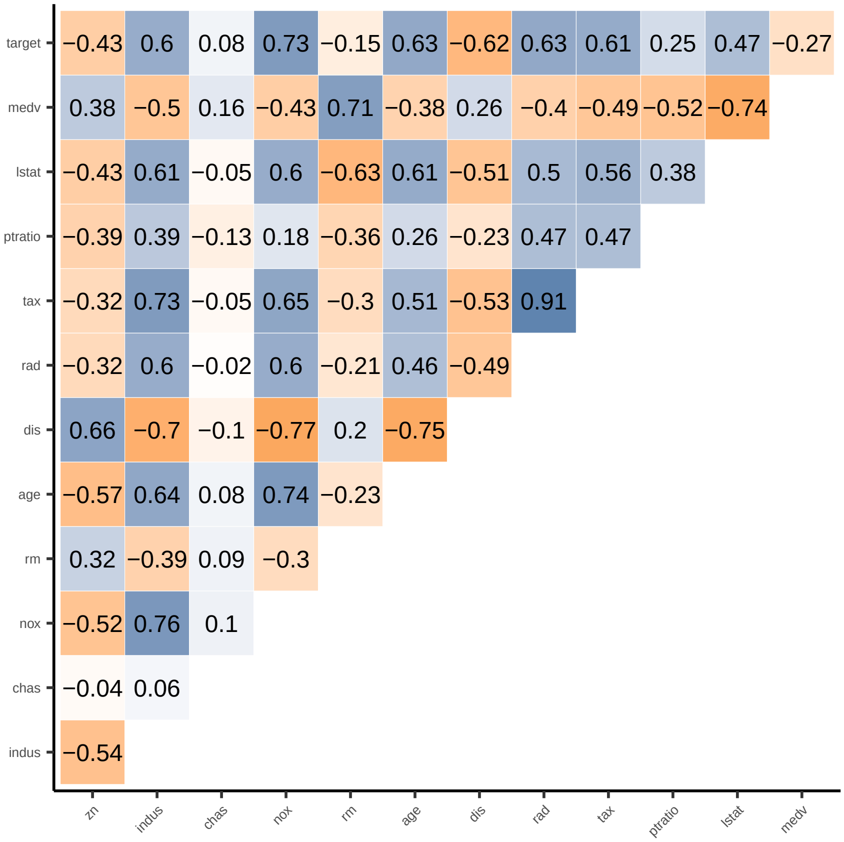
Grouping our predictor variables by the `target` response variable we can see some of the variables have very large differences in their distributions based on the `target` variable. These are variables that strongly seem to be correlated with the target variable and could be included in our model.



Correlation Matrix Our next step is to check the correlation between all our variables. We can check which seem to be correlated with our target variable for inclusion in our models and to check for multicollinearity between two of our predictor variables.

- **Negative Correlations with Crime Rate:** Predictors `indus`, `nox`, `age`, `dis`, `rad`, `tax`, `ptratio`, `lstat`, and `medv` exhibit negative correlations with the response variable `target`, indicating that as these variables increase, the likelihood of the crime rate being above the median decreases. This may suggest that areas with higher industrial presence, pollution levels, older housing stock, longer distances to employment centers, poorer accessibility to highways, higher tax rates, higher pupil-teacher ratios, lower socio-economic status, and lower median home values tend to have lower crime rates.
- **Positive Correlations with Crime Rate:** Conversely, predictors such as `zn` and `chas` exhibit positive correlations with the response variable `target`, implying that as these variables increase, the likelihood of the crime rate being above the median also increases. This may suggest that areas with larger residential lots and those bordering the Charles River may experience higher crime rates.

The correlation matrix also illustrates some strong relationships between some of the predictor variables. For example, `tax` and `rad` have a very strong correlation of 0.91. While none of the rest of the predictor variables have anything that high there are still a few with significant correlations greater than 0.7.



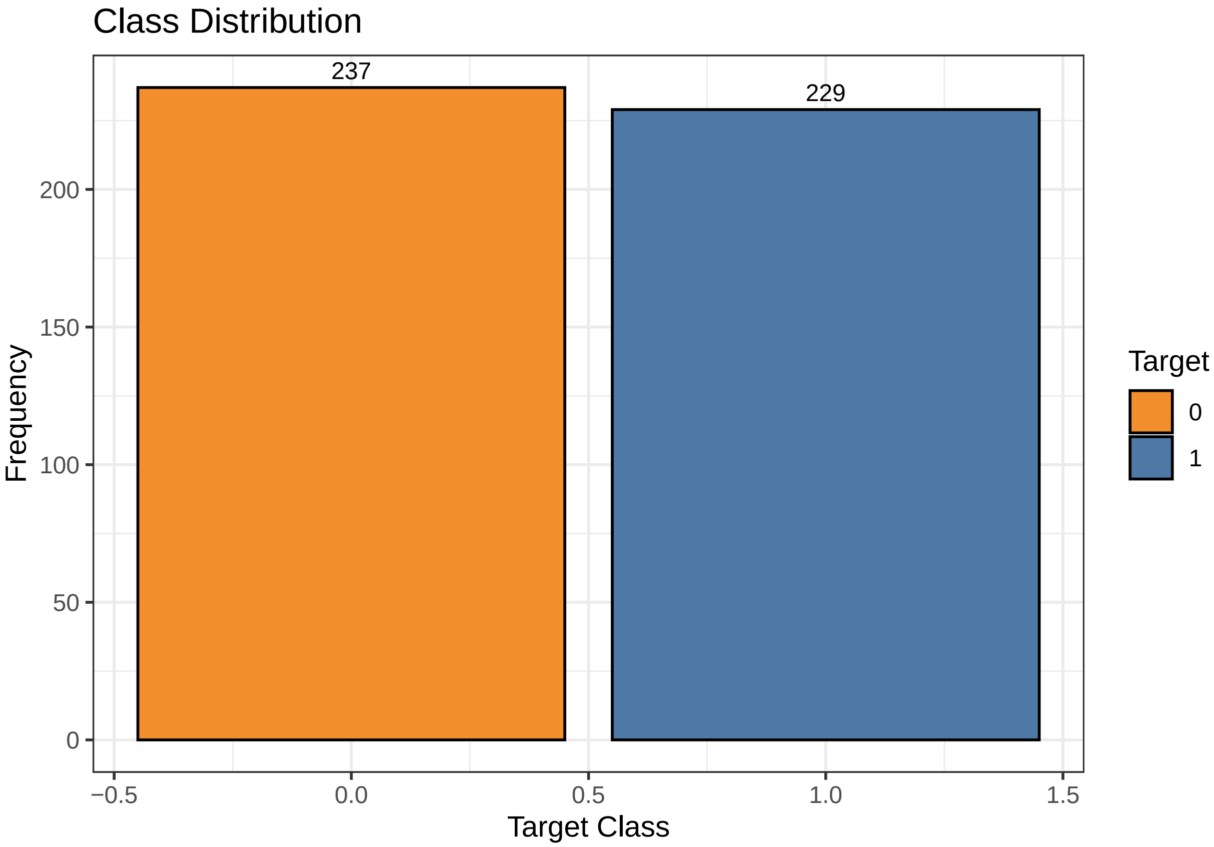
	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv
target	−0.43	0.6	0.08	0.73	−0.15	0.63	−0.62	0.63	0.61	0.25	0.47	−0.27
medv	0.38	−0.5	0.16	−0.43	0.71	−0.38	0.26	−0.4	−0.49	−0.52	−0.74	
lstat	−0.43	0.61	−0.05	0.6	−0.63	0.61	−0.51	0.5	0.56	0.38		
ptratio	−0.39	0.39	−0.13	0.18	−0.36	0.26	−0.23	0.47	0.47			
tax	−0.32	0.73	−0.05	0.65	−0.3	0.51	−0.53	0.91				
rad	−0.32	0.6	−0.02	0.6	−0.21	0.46	−0.49					
dis	0.66	−0.7	−0.1	−0.77	0.2	−0.75						
age	−0.57	0.64	0.08	0.74	−0.23							
rm	0.32	−0.39	0.09	−0.3								
nox	−0.52	0.76	0.1									
chas	−0.04	0.06										
indus	−0.54											

The following table extracts all the pairs of predictors with a correlation above 0.70, assuming this general threshold is high. These can cause issues with collinearity and should be treated as such for our models.

variable_1	variable_2	correlation
rad	tax	0.90646
nox	dis	-0.76888
indus	nox	0.75963
age	dis	-0.75090
lstat	medv	-0.73580
nox	age	0.73513
indus	tax	0.73223
nox	target	0.72611
rm	medv	0.70534
indus	dis	-0.70362

Class Imbalance

Lastly, we will check whether the classes of the `target` variable is balanced to avoid misleading models. For example, if the data has an imbalance of 95% to 5% success/fail rate, then predicting 100% percent of the time will be a success will result in a model successful 95% of the time but of zero actual value to us. Since we are dealing with above or below the mean crime rate, we confirm the data is balanced with 237 below mean crime rate and 229 above in our dataset.



Data Preparation

After our initial data exploration, we can now move on to data preparation. This involves handling missing values, outliers, and performing necessary transformations to address skewness in the data.

Fix Missing Values

As noted in the exploratory section, there no missing values within the data set, so we did not need to perform any imputation or handling of missing data.

Transformations

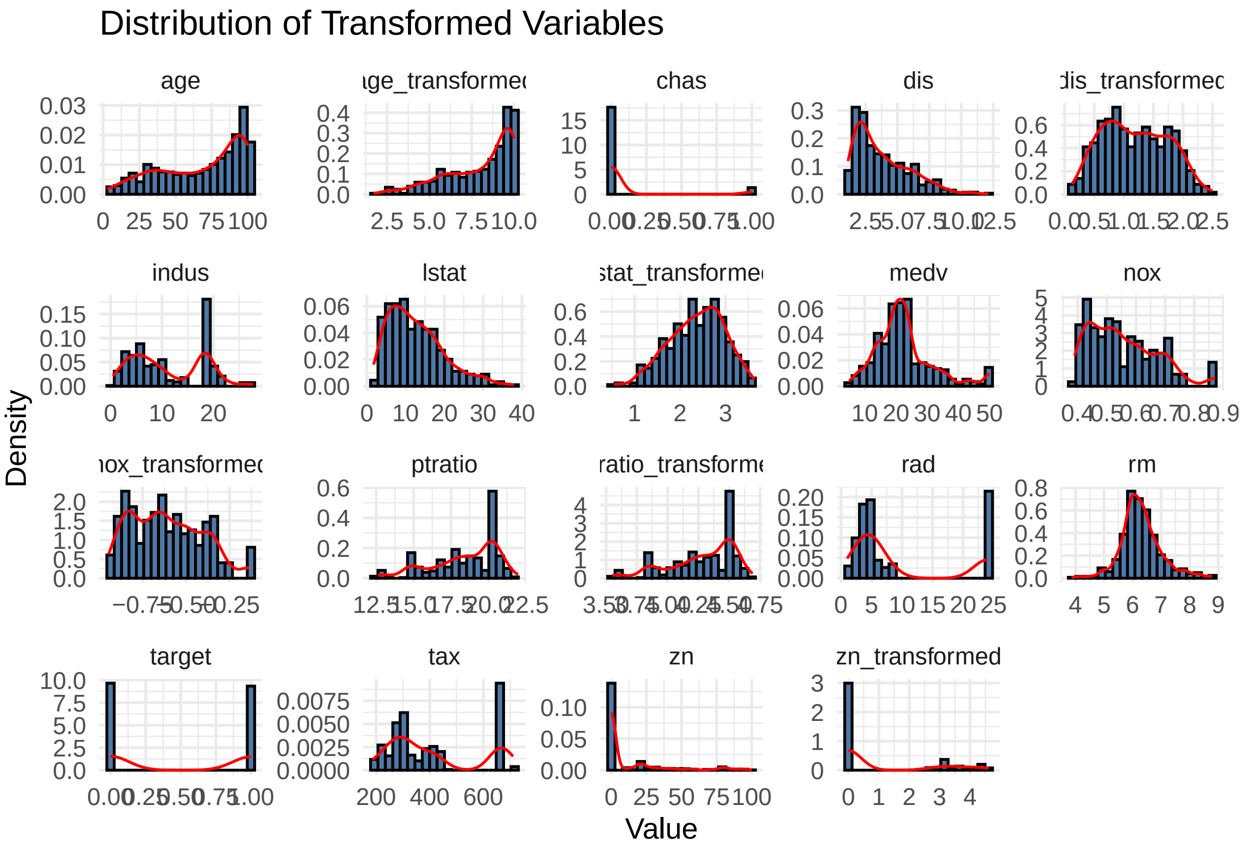
During our exploratory analysis, we noticed that some variables had skewed distributions, which could affect the accuracy of our models. To address this issue, we applied specific transformations to make the data more suitable for modeling:

- **Logarithmic Transformation**: Used for variables `dis`, `lstat`, `zn`, and `nox`. This transformation helps to reduce the impact of extreme values and make the distribution more balanced by compressing the range of values.
- **Square Root Transformation**: Applied to `age` and `ptratio`. By taking the square root, we make the distribution less skewed, which can improve model performance, especially for variables with a left-skewed pattern.

The rest of the variables were kept unchanged because they either didn't exhibit significant skewness in their distributions or because alternative transformations were not deemed necessary based on our exploratory analysis. By retaining these variables in their original form, we ensure that the original information is preserved while still addressing skewness in the variables where it was observed.

These transformations simplify the data distribution, making it easier for models to interpret and generate more reliable predictions. The same was done to our test set.

Visualizations of the cleaned dataset featuring the transformed variables are presented below through histograms. These visual representations aid in illustrating the distributions of the transformed variables compared to their original form.



Handling Outliers

After reviewing boxplots of our variables from the data exploration, we noticed that several variables, including `rm`, `medv`, `zn`, and others, contained a significant number of outliers. Despite their presence, we decided to retain these outliers in our dataset. This decision was made to keep the original data intact and ensure that we have a complete view of the variable distributions. Excluding outliers could lead to losing important information. Therefore, we decided to include the outliers in our dataset to ensure reliable modeling results.

Model Preparation

Correlation

For modeling, we start with using all available variables and evaluate their significance by the amount of variation they explain using ANOVA and their F-Statistic. We have expectations that variables with high correlation to `target` will be highly significant.

variable	correlation
indus	0.6049
nox_transformed	0.7457
dis_transformed	-0.6551
rad	0.6281
tax	0.6111

Training and Test Split

Here we will re-factor our `target` variable and then split our whole training data into a training and test set.

Table 1: Training Set

	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	target	dis_transformed
2	0	19.58	1	0.871	5.403	100.0	1.3216	5	403	14.7	26.82	13.4	Yes	0.2788431
4	30	4.93	0	0.428	6.393	7.8	7.0355	6	300	16.6	5.19	23.7	No	1.9509688
7	0	18.10	0	0.693	5.453	100.0	1.4896	24	666	20.2	30.59	5.0	Yes	0.3985076
9	0	5.19	0	0.515	6.316	38.1	6.4584	5	224	20.2	5.68	22.2	No	1.8653816
10	80	3.64	0	0.392	5.876	19.1	9.2203	1	315	16.4	9.25	20.9	No	2.2214076
11	22	5.86	0	0.431	6.438	8.9	7.3967	7	330	19.1	3.59	24.8	No	2.0010340

Table 2: Test Set

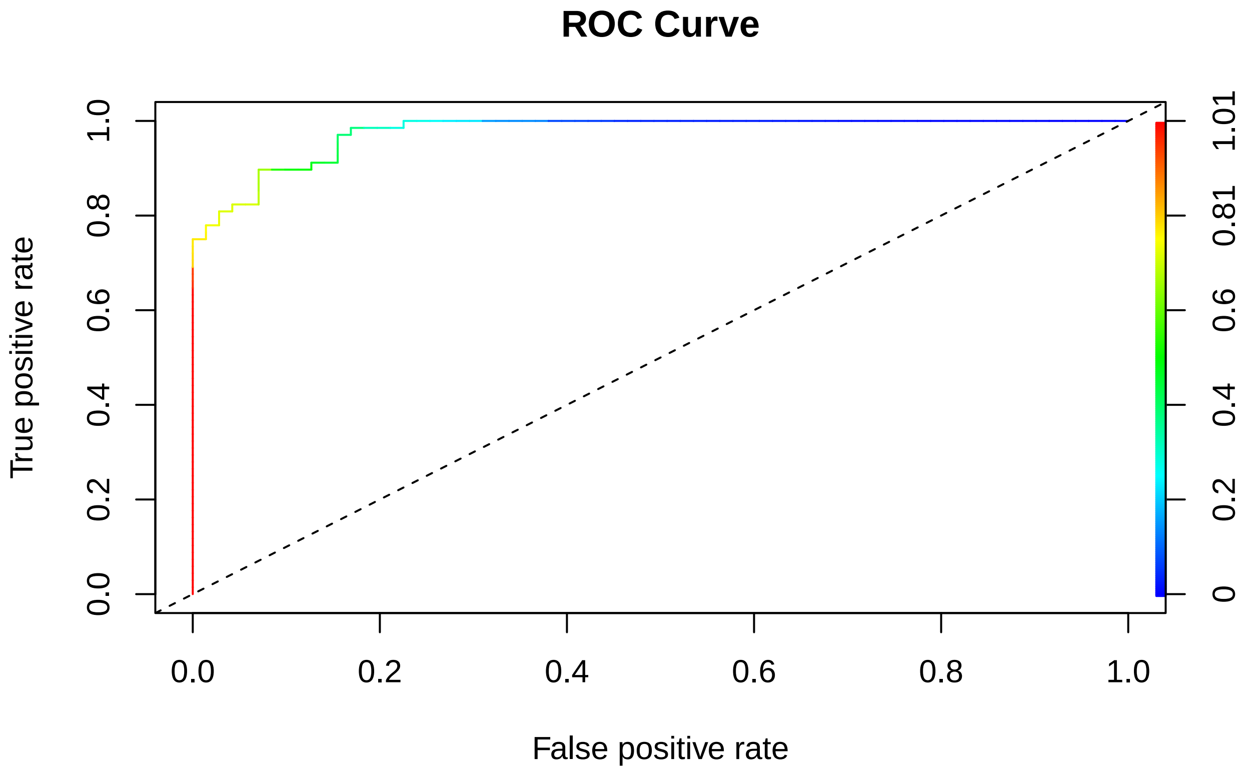
	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	target	dis_transformed
1	0	19.58	0	0.605	7.929	96.2	2.0459	5	403	14.7	3.70	50.0	Yes	0.7158378
3	0	18.10	0	0.740	6.485	100.0	1.9784	24	666	20.2	18.85	15.4	Yes	0.6822884
5	0	2.46	0	0.488	7.155	92.2	2.7006	3	193	17.8	4.82	37.9	No	0.9934740
6	0	8.56	0	0.520	6.781	71.3	2.8561	5	384	20.9	7.67	26.5	No	1.0494571
8	0	18.10	0	0.693	4.519	100.0	1.6582	24	666	20.2	36.98	7.0	Yes	0.5057327
13	0	18.10	0	0.532	7.061	77.0	3.4106	24	666	20.2	7.01	25.0	Yes	1.2268882

Model Building

Model 1: PCA

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.1175 1.2270 0.94271 0.86996 0.64916 0.62626 0.51027
## Proportion of Variance 0.4982 0.1673 0.09875 0.08409 0.04682 0.04358 0.02893
## Cumulative Proportion  0.4982 0.6655 0.76425 0.84835 0.89517 0.93875 0.96768
##                            PC8     PC9
## Standard deviation     0.40491 0.35632
## Proportion of Variance 0.01822 0.01411
## Cumulative Proportion  0.98589 1.00000
```

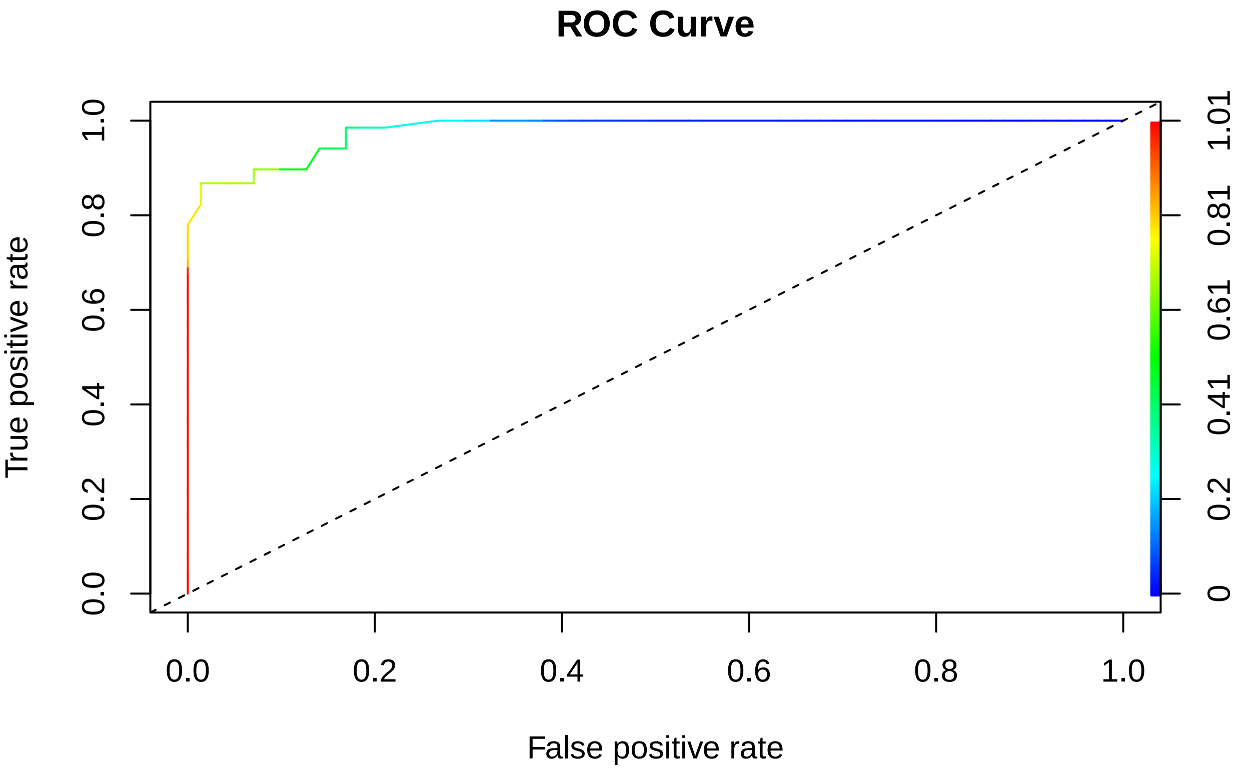
```
##
## Call:
## NULL
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      9.856611   2.152548   4.579 4.67e-06 ***
## nox_transformed 16.088941   2.769700   5.809 6.29e-09 ***
## rad              0.621594   0.152996   4.063 4.85e-05 ***
## tax             -0.009385   0.002897  -3.240   0.0012 **
## pc1              0.176959   0.180487   0.980   0.3269
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 453.24  on 326  degrees of freedom
## Residual deviance: 172.49  on 322  degrees of freedom
## AIC: 182.49
##
## Number of Fisher Scoring iterations: 8
```


ROC Curve
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
0
0.2
0.4
0.6
0.81
1.01

```
## AUC: 0.9762
```

Model 2: Simple Logistic Regression

```
##
## Call:
## NULL
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      10.488191   2.085141   5.030 4.91e-07 ***
## nox_transformed 17.119727   2.621852   6.530 6.59e-11 ***
## rad               0.557378   0.131846   4.227 2.36e-05 ***
## tax              -0.008322   0.002622  -3.173  0.00151 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 453.24  on 326  degrees of freedom
## Residual deviance: 173.45  on 323  degrees of freedom
## AIC: 181.45
##
## Number of Fisher Scoring iterations: 8
```



```
## AUC: 0.9801
```

Model 3: Interaction Terms

This time, we would like to use interaction terms, but we're not sure whether to use interaction terms of the transformeed or non-transformed variables. We also would prefer to not use interaction terms between one transformed and another non-transformed variable as this makes interpretability quite challenging. Finally, if a transformed variable and a non-transformed variable are similarly significant, we'll default to using the non-transformed version for sake of simplicity:

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##
## Call:
## glm(formula = target ~ ., family = binomial, data = trainData)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -5.201e+01  1.600e+02  -0.325  0.74518
## zn                  -2.129e-02  1.042e-01  -0.204  0.83805
## indus                7.792e-02  8.240e-02   0.946  0.34436
## chas                 1.175e+00  9.792e-01   1.200  0.23017
## nox                  2.432e+02  1.231e+02   1.975  0.04826 *
## rm                  -1.655e+00  1.078e+00  -1.534  0.12499
## age                  2.527e-01  8.554e-02   2.955  0.00313 **
## dis                 -3.045e+00  1.222e+00  -2.492  0.01271 *
## rad                  1.063e+00  2.575e-01   4.129 3.65e-05 ***
## tax                 -1.122e-02  4.194e-03  -2.677  0.00744 **
## ptratio              8.855e+00  5.011e+00   1.767  0.07721 .
## lstat                1.852e-01  1.527e-01   1.213  0.22505
## medv                 1.651e-01  9.795e-02   1.686  0.09187 .
## dis_transformed      1.513e+01  4.952e+00   3.054  0.00226 **
## lstat_transformed   -3.032e+00  2.057e+00  -1.474  0.14053
## zn_transformed      -1.925e-01  9.472e-01  -0.203  0.83899
## nox_transformed     -1.055e+02  6.238e+01  -1.690  0.09093 .
## age_transformed     -2.954e+00  1.138e+00  -2.596  0.00943 **
## ptratio_transformed -6.997e+01  4.258e+01  -1.643  0.10031
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 453.24  on 326  degrees of freedom
## Residual deviance: 115.80  on 308  degrees of freedom
## AIC: 153.8
##
## Number of Fisher Scoring iterations: 10
```

Given this summary, and given our intuitions about the relationships between variables, we'll explore the following interactions:

1. `zn` * `indus`: neither appear significant, but perhaps crime pops up in different ways in areas more highly zoned for residential use when there's a lot of non-retail businesses.
2. `chas` * `nox_transformed`: particularly because of the binary nature of chas, but maybe the *combination* of environmental quality and proximity to the Charles River influences crime (we're thinking about property values).
3. `chas` * `dis_transformed`: similarly, the combination of distance to Charles River and distance to employment centers might impact crime in a special way.

4. `nox_transformed` * `dis_transformed`: bad air quality might impact crime more, for example, if especially far from employment centers.
5. `rm` * `lstat`: this speaks to the combination of housing situations and socioeconomic conditions impact crime.
6. `age_transformed` * `rad`: older neighborhoods with better/worse access to highways might well experience crime differently.
7. `tax` * `lstat_transformed`: for example, if those in low-income areas also have to pay high taxes, we might expect more crime.
8. `ptratio_transformed` * `medv`: this speaks to the combination of educational quality (or at least resources) combines with home values to influence crime.

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##
## Call:
## glm(formula = target ~ zn * indus + chas * nox_transformed +
##     chas * dis_transformed + nox_transformed * dis_transformed +
##     rm * lstat + age_transformed * rad + tax * lstat_transformed +
##     ptratio_transformed * medv, family = binomial(link = "logit"),
##     data = trainData)
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -77.73663   28.22543  -2.754 0.005885 **
## zn                               -0.16473    0.09680  -1.702 0.088797 .
## indus                            -0.01582    0.07779  -0.203 0.838860
## chas                             -3.62504    7.74296  -0.468 0.639661
## nox_transformed                  45.74066   13.34680   3.427 0.000610 ***
## dis_transformed                  -5.69992    5.30621  -1.074 0.282734
## rm                                1.36678    2.29623   0.595 0.551692
## lstat                             1.23178    0.78782   1.564 0.117926
## age_transformed                  -0.51874    0.66946  -0.775 0.438415
## rad                              -0.54854    1.02467  -0.535 0.592419
## tax                               0.08816    0.03152   2.797 0.005156 **
## lstat_transformed                 9.36565    4.37572   2.140 0.032325 *
## ptratio_transformed              16.32664    4.90775   3.327 0.000879 ***
## medv                              2.42745    0.90778   2.674 0.007494 **
## zn:indus                          0.01636    0.01350   1.212 0.225608
## chas:nox_transformed            -30.72636   24.63311  -1.247 0.212265
## chas:dis_transformed            -12.09517   12.41856  -0.974 0.330077
## nox_transformed:dis_transformed -14.07220    8.34862  -1.686 0.091878 .
## rm:lstat                         -0.18414    0.16317  -1.129 0.259096
## age_transformed:rad               0.15178    0.12574   1.207 0.227402
## tax:lstat_transformed            -0.03721    0.01239  -3.003 0.002677 **
## ptratio_transformed:medv         -0.55671    0.21404  -2.601 0.009296 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 453.24  on 326  degrees of freedom
## Residual deviance: 117.95  on 305  degrees of freedom
## AIC: 161.95
##
## Number of Fisher Scoring iterations: 10
```

Only some variables are statistically significant; we'd much prefer a simpler model, so we add backward elimination.

```
## Start:  AIC=161.95
## target ~ zn * indus + chas * nox_transformed + chas * dis_transformed +
##     nox_transformed * dis_transformed + rm * lstat + age_transformed *
##     rad + tax * lstat_transformed + ptratio_transformed * medv
##
##                                   Df Deviance    AIC
## - zn:indus                         1   118.89 160.89
## - chas:dis_transformed             1   118.95 160.95
## - chas:nox_transformed             1   119.07 161.07
## - rm:lstat                         1   119.28 161.28
## - age_transformed:rad              1   119.45 161.45
## <none>                                 117.95 161.95
## - nox_transformed:dis_transformed  1   121.04 163.04
## - ptratio_transformed:medv         1   126.47 168.47
## - tax:lstat_transformed            1   135.37 177.37
##
## Step:  AIC=160.89
## target ~ zn + indus + chas + nox_transformed + dis_transformed +
##     rm + lstat + age_transformed + rad + tax + lstat_transformed +
##     ptratio_transformed + medv + chas:nox_transformed + chas:dis_transformed +
##     nox_transformed:dis_transformed + rm:lstat + age_transformed:rad +
##     tax:lstat_transformed + ptratio_transformed:medv
##
##                                   Df Deviance    AIC
## - indus                            1   118.89 158.89
## - chas:dis_transformed             1   119.91 159.91
## - age_transformed:rad              1   119.91 159.91
## - chas:nox_transformed             1   120.02 160.02
## - rm:lstat                         1   120.14 160.15
## <none>                                 118.89 160.89
## - nox_transformed:dis_transformed  1   121.62 161.62
## - zn                               1   122.87 162.87
## - ptratio_transformed:medv         1   126.49 166.49
## - tax:lstat_transformed            1   135.68 175.68
##
## Step:  AIC=158.89
## target ~ zn + chas + nox_transformed + dis_transformed + rm +
##     lstat + age_transformed + rad + tax + lstat_transformed +
##     ptratio_transformed + medv + chas:nox_transformed + chas:dis_transformed +
##     nox_transformed:dis_transformed + rm:lstat + age_transformed:rad +
##     tax:lstat_transformed + ptratio_transformed:medv
##
##                                   Df Deviance    AIC
## - chas:dis_transformed             1   119.91 157.91
## - age_transformed:rad              1   119.94 157.94
## - chas:nox_transformed             1   120.02 158.02
## - rm:lstat                         1   120.25 158.25
## <none>                                 118.89 158.89
## - nox_transformed:dis_transformed  1   122.16 160.16
## - zn                               1   123.09 161.09
## - ptratio_transformed:medv         1   127.10 165.10
## - tax:lstat_transformed            1   135.69 173.69
##
## Step:  AIC=157.91
## target ~ zn + chas + nox_transformed + dis_transformed + rm +
##     lstat + age_transformed + rad + tax + lstat_transformed +
##     ptratio_transformed + medv + chas:nox_transformed + nox_transformed:dis_transformed +
##     rm:lstat + age_transformed:rad + tax:lstat_transformed +
##     ptratio_transformed:medv
##
##                                   Df Deviance    AIC
## - chas:nox_transformed             1   120.06 156.06
## - age_transformed:rad              1   120.82 156.82
## - rm:lstat                         1   121.38 157.38
## <none>                                 119.91 157.91
## - nox_transformed:dis_transformed  1   122.78 158.78
## - zn                               1   123.76 159.76
## - ptratio_transformed:medv         1   127.79 163.79
## - tax:lstat_transformed            1   137.75 173.75
##
## Step:  AIC=156.05
## target ~ zn + chas + nox_transformed + dis_transformed + rm +
##     lstat + age_transformed + rad + tax + lstat_transformed +
##     ptratio_transformed + medv + nox_transformed:dis_transformed +
##     rm:lstat + age_transformed:rad + tax:lstat_transformed +
##     ptratio_transformed:medv
##
##                                   Df Deviance    AIC
## - age_transformed:rad              1   120.96 154.96
## - rm:lstat                         1   121.60 155.60
## - chas                             1   121.88 155.88
## <none>                                 120.06 156.06
## - nox_transformed:dis_transformed  1   122.89 156.89
## - zn                               1   123.86 157.86
## - ptratio_transformed:medv         1   128.26 162.26
## - tax:lstat_transformed            1   137.90 171.90
##
## Step:  AIC=154.96
## target ~ zn + chas + nox_transformed + dis_transformed + rm +
##     lstat + age_transformed + rad + tax + lstat_transformed +
##     ptratio_transformed + medv + nox_transformed:dis_transformed +
##     rm:lstat + tax:lstat_transformed + ptratio_transformed:medv
##
##                                   Df Deviance    AIC
## - rm:lstat                         1   122.31 154.31
## - age_transformed                  1   122.37 154.37
## - chas                             1   122.86 154.86
## <none>                                 120.96 154.96
## - nox_transformed:dis_transformed  1   123.01 155.01
## - zn                               1   124.17 156.17
## - ptratio_transformed:medv         1   129.44 161.44
## - tax:lstat_transformed            1   137.91 169.91
## - rad                              1   165.60 197.60
##
## Step:  AIC=154.31
## target ~ zn + chas + nox_transformed + dis_transformed + rm +
##     lstat + age_transformed + rad + tax + lstat_transformed +
##     ptratio_transformed + medv + nox_transformed:dis_transformed +
##     tax:lstat_transformed + ptratio_transformed:medv
##
##                                   Df Deviance    AIC
## - rm                               1   123.32 153.32
## <none>                                 122.31 154.31
## - chas                             1   124.38 154.38
## - age_transformed                  1   124.40 154.40
## - nox_transformed:dis_transformed  1   124.62 154.62
## - zn                               1   126.26 156.26
## - lstat                            1   128.39 158.39
## - ptratio_transformed:medv         1   130.71 160.71
## - tax:lstat_transformed            1   138.69 168.69
## - rad                              1   165.60 195.60
##
## Step:  AIC=153.32
## target ~ zn + chas + nox_transformed + dis_transformed + lstat +
##     age_transformed + rad + tax + lstat_transformed + ptratio_transformed +
##     medv + nox_transformed:dis_transformed + tax:lstat_transformed +
##     ptratio_transformed:medv
##
##                                   Df Deviance    AIC
## - age_transformed                  1   124.46 152.46
## - nox_transformed:dis_transformed  1   125.23 153.23
## <none>                                 123.32 153.32
## - chas                             1   125.66 153.66
## - zn                               1   127.34 155.34
## - lstat                            1   128.93 156.93
## - ptratio_transformed:medv         1   131.84 159.84
## - tax:lstat_transformed            1   139.11 167.11
## - rad                              1   165.68 193.68
##
## Step:  AIC=152.46
## target ~ zn + chas + nox_transformed + dis_transformed + lstat +
##     rad + tax + lstat_transformed + ptratio_transformed + medv +
##     nox_transformed:dis_transformed + tax:lstat_transformed +
##     ptratio_transformed:medv
##
##                                   Df Deviance    AIC
## - nox_transformed:dis_transformed  1   126.06 152.06
## <none>                                 124.46 152.46
## - chas                             1   127.13 153.13
## - zn                               1   128.78 154.78
## - lstat                            1   129.85 155.85
## - ptratio_transformed:medv         1   135.79 161.79
## - tax:lstat_transformed            1   141.50 167.50
## - rad                              1   166.73 192.73
##
## Step:  AIC=152.06
## target ~ zn + chas + nox_transformed + dis_transformed + lstat +
##     rad + tax + lstat_transformed + ptratio_transformed + medv +
##     tax:lstat_transformed + ptratio_transformed:medv
```

```
##
##                            Df Deviance    AIC
## <none>                          126.06 152.06
## - chas                      1   128.73 152.73
## - zn                        1   128.78 152.78
## - dis_transformed           1   131.52 155.52
## - lstat                     1   133.16 157.16
## - ptratio_transformed:medv  1   136.32 160.32
## - tax:lstat_transformed     1   142.78 166.78
## - rad                       1   168.48 192.48
## - nox_transformed           1   180.40 204.40
```

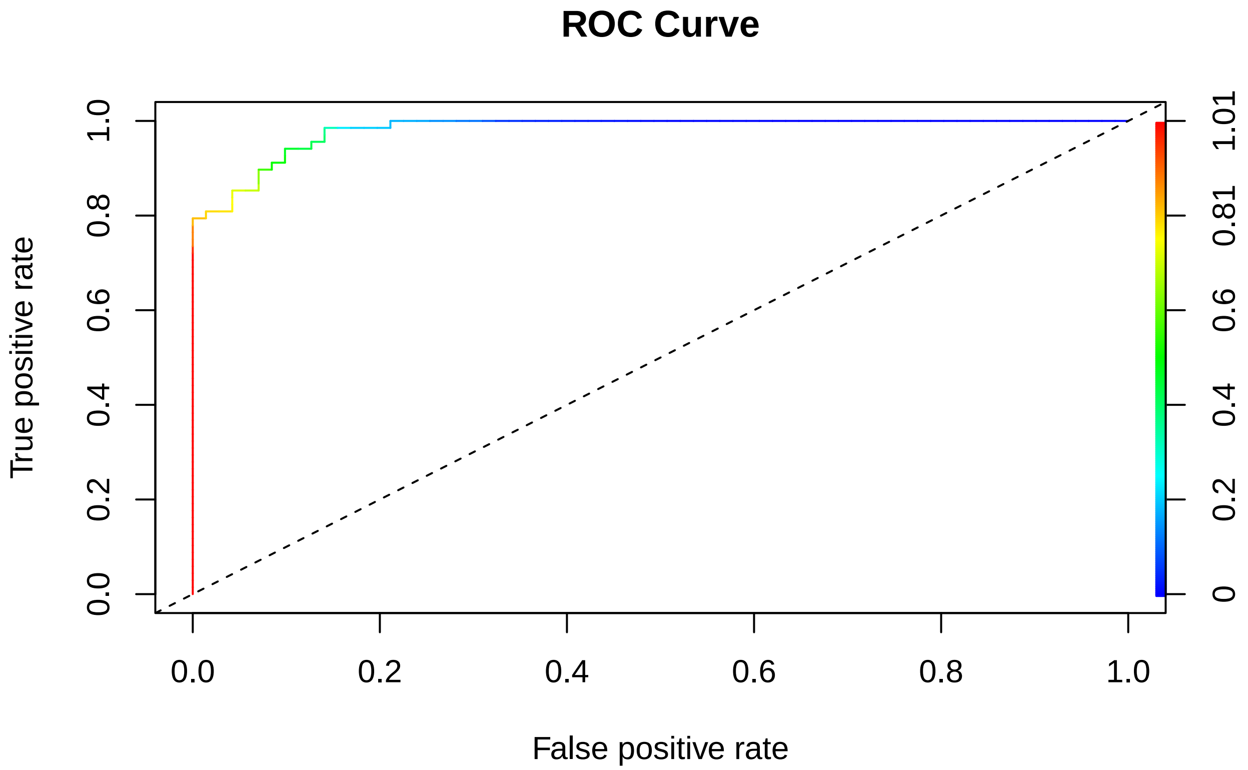
The final model has only 7 variables, 6 of which are interaction terms. This model was created with glm, which means that we need to do type = "response" to get the predicted probabilities for the evaluation set. We would then use some threshold to classify predictions. The default threshold is 0.5, but if we're anyways setting a threshold, we may as well optimize that threshold:

```
## [1] 0.32
```

Now we use that threshold to make predictions and construct a confusion matrix:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##        No  61   1
##        Yes 10  67
##
##                Accuracy : 0.9209
##                  95% CI : (0.8628, 0.9598)
##     No Information Rate : 0.5108
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.8421
##
##  Mcnemar's Test P-Value : 0.01586
##
##             Sensitivity : 0.8592
##             Specificity : 0.9853
##          Pos Pred Value : 0.9839
##          Neg Pred Value : 0.8701
##              Prevalence : 0.5108
##          Detection Rate : 0.4388
##    Detection Prevalence : 0.4460
##       Balanced Accuracy : 0.9222
##
##        'Positive' Class : No
##
```

And we again plot the ROC curve:


ROC Curve
True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
0
0.2
0.4
0.6
0.81
1.01

```
## AUC: 0.9816
```

Table 3: Simple Regression Metrics

	Class_Error_Rate	Precision	Sensitivity	Specificity	F1	AUC
	0.1	0.9	0.9	0.9	0.9	0.98

Model Selection

Baseline models showed that `nox` and `rad` were both highly significant and served to explain the majority of variance. Through backward elimination two models were selected. `nox_transformed`, `rad`, and `tax` were used as core variables in both models. The second model included attempting a Principal Components Analysis transformation to transform the weak variables and use the single most useful principal component. This second model did not show this component as significant. Overall, the models used and their accuracies can be seen as follows, assuming coefficients are placed through the logit-odds formula:

Simple Model

$$\begin{aligned}\log\left(\frac{p}{1-p}\right) = &+ 10.49\\ &+ 17.12\,(nox\_trans)\\ &+ 0.56\,(rad)\\ &- 0.01\,(tax)\end{aligned}$$

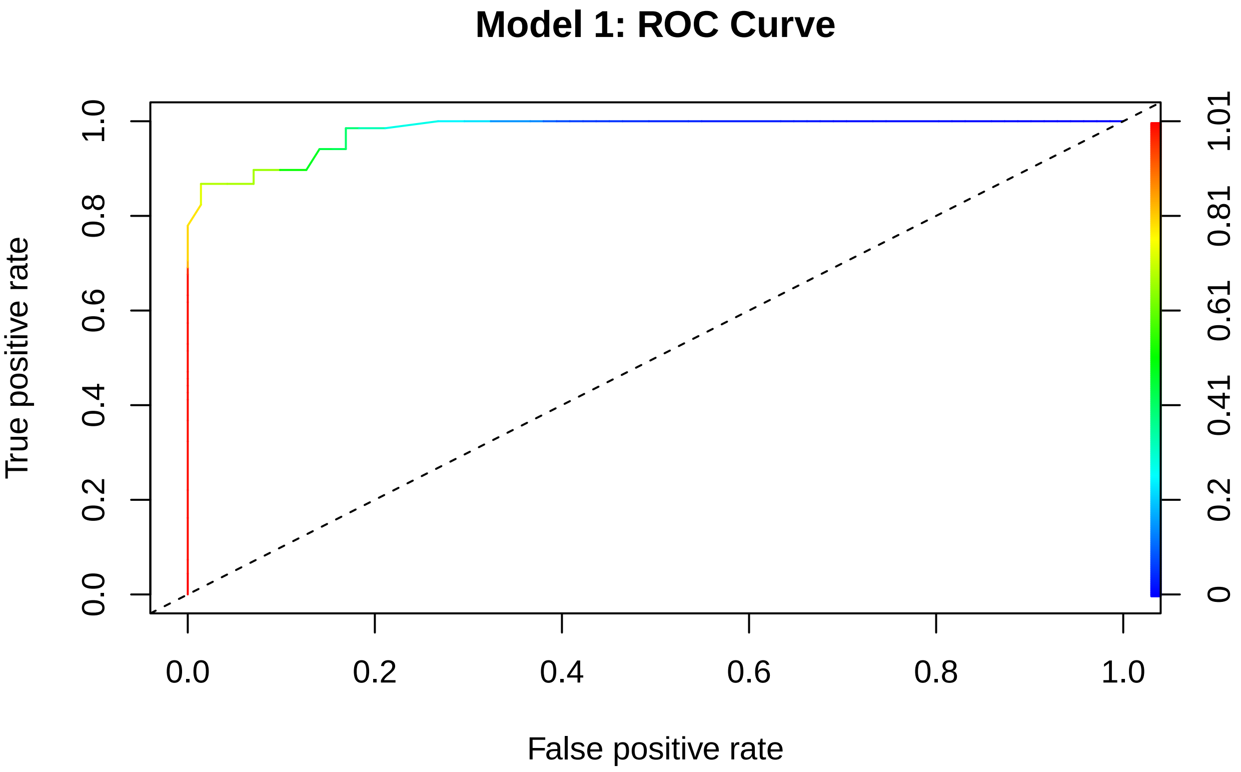
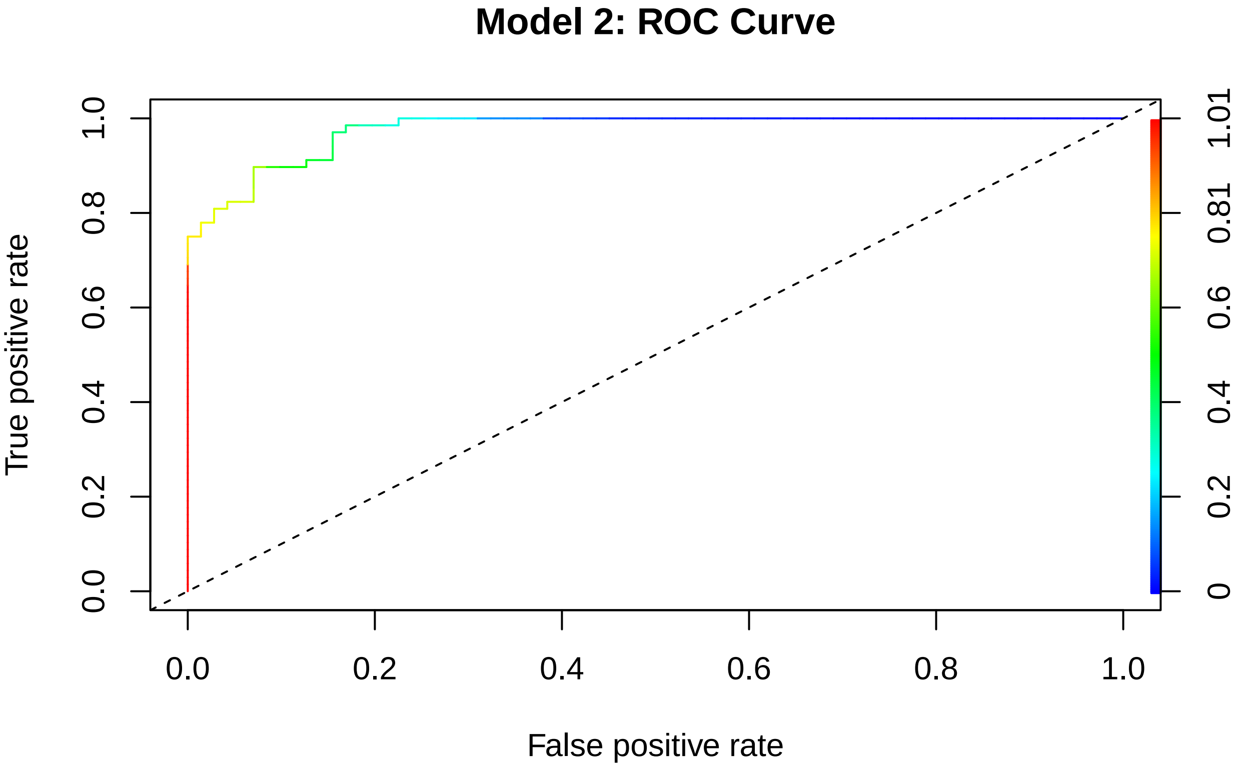


Table 4: PCA Metrics

	Class_Error_Rate	Precision	Sensitivity	Specificity	F1	AUC
	0.11	0.9	0.89	0.9	0.89	0.98

PCA Model

$$\begin{aligned}\log\left(\frac{p}{1-p}\right) = &+10.49\\ &+18.81\,(nox\_trans)\\ &+.70\,(rad)\\ &-.01\,(tax)\\ &+.10\,(pc1)\end{aligned}$$



As for the third model, the Interaction Model, its equation is:

Table 5: Interaction Metrics

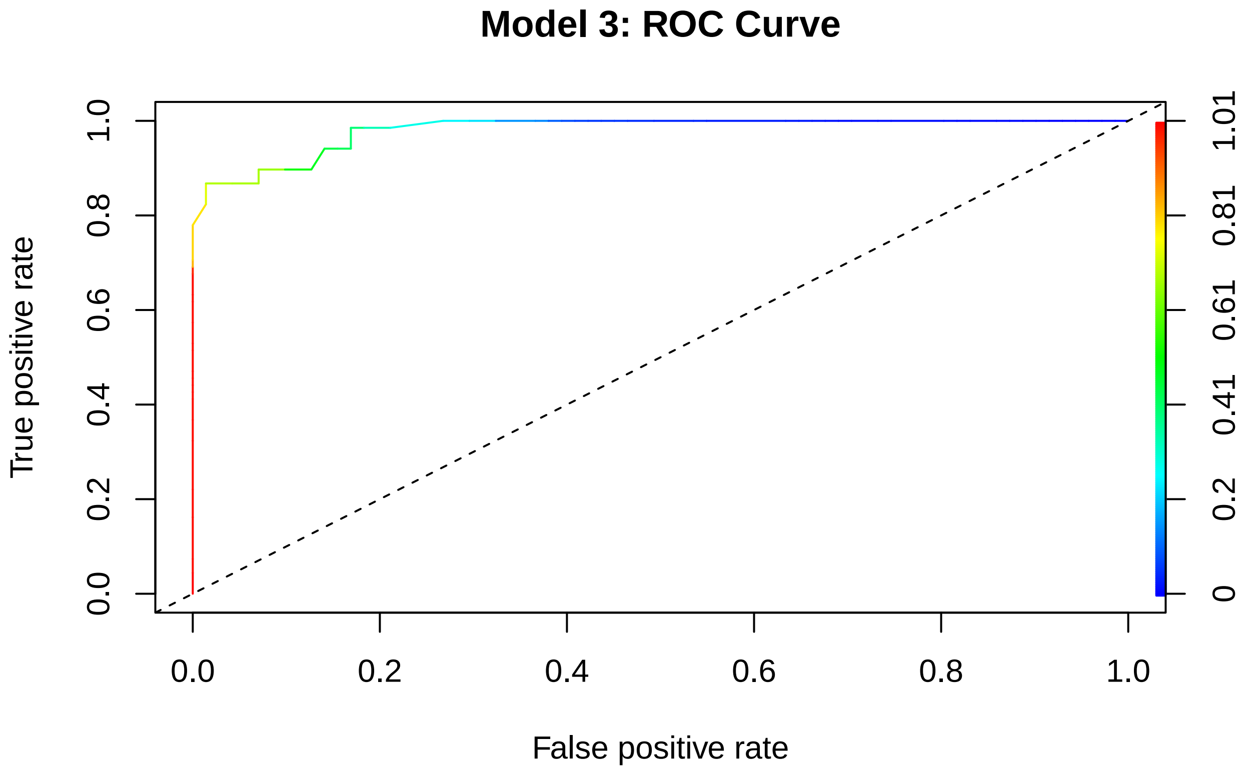
	Class_Error_Rate	Precision	Sensitivity	Specificity	F1	AUC
	0.08	0.98	0.86	0.99	0.92	0.98

Interaction Model

$$\begin{aligned}
\log\left(\frac{p}{1-p}\right) = &- 73.89614 \\
&- 0.10415\,(zn) \\
&- 9.36481\,(chas) \\
&+ 44.09606\,(nox\_transformed) \\
&- 2.84960\,(dis\_transformed) \\
&+ 2.04746\,(rm) \\
&+ 1.32175\,(lstat) \\
&- 0.55321\,(age\_transformed) \\
&- 0.60241\,(rad) \\
&+ 0.08510\,(tax) \\
&+ 10.68699\,(lstat\_transformed) \\
&+ 13.29307\,(ptratio\_transformed) \\
&+ 1.92650\,(medv) \\
&- 15.01157\,(chas : nox\_transformed) \\
&- 10.81333\,(nox\_transformed : dis\_transformed) \\
&- 0.21539\,(rm : lstat) \\
&+ 0.17194\,(age\_transformed : rad) \\
&- 0.03567\,(tax : lstat\_transformed) \\
&- 0.43344\,(ptratio\_transformed : medv)
\end{aligned}$$

That equation is difficult to look at, and that’s a major problem; despite the promising performance of this model, its lack of interpretability will ultimately disqualify it as a viable model. Still, here are those promising statistics:

Finally, we see the ROC curve:



Final Model Choice

In our first two models, both models showed high results, but the Simple Model had better percentages across the board.

Lets define a few terms first.

- *Classification Error Rate* measures how often the model predicted incorrectly, whether it be a false positive or a false negative.
- *Precision* measures how often the model correctly predicts the positives in the positive class.
- *Sensitivity* measures how well a model correctly predicts positives in all observations.
- *Specificity* measures how well a model correctly predicts the negatives in the negatives class.

Metrics of Interest

- The F1 score is an average of precision and sensitivity, and is typically more useful than precision to measure a classification model, particularly if one class is more prevalent than another.
- The AUC score measures the rate at which a random positive example is would be more likely to be classified as positive than a negative example.
- The confusion matrix shows the exact breakdown of how many observations were classified as positive or negative and how many of them were actually positive or negative.
- The ROC Curve shows the changing rates of True Positives and False Positives as different thresholds of rounding are used to classify as a positive or negative.

Depending on the goals for this assignment, these metrics can be used to pick different models. If the goal were to be highly sensitive to high-crime areas to identify areas a patrol should be sent to deter crime, valuing the Precision metric over others would be useful. This is assuming that sending a patrol is not a high cost endeavor.

Since the Simple Model was more parsimonious, and easier to understand, with little loss in accuracy compared to the Interaction Model, it was used to generate predictions. The PCA model complicated the model more with less accuracy in most measurements and was not used to generate predictions.

Generate Predictions

```
##  [1] 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1
## [39] 1 0
```