DATA621 HW3

Joseph Elikishvili

April 16, 2018

Overview

In this project, we will explore, analyze and model a data set 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). Our objective is to build a binary logistic regression model on the training data set to predict whether the neighborhood will be at risk for high crime levels. We will provide classifications and probabilities for the evaluation data set using your binary logistic regression model.

1. Data Exploration

We will get started by loading the data and exploring the dimensions of the dataset and getting to know the variables

```
## [1] 466  13
```

It appears we are dealing with a total of 13 variables and 466 records.

Next we will preview the data in raw format

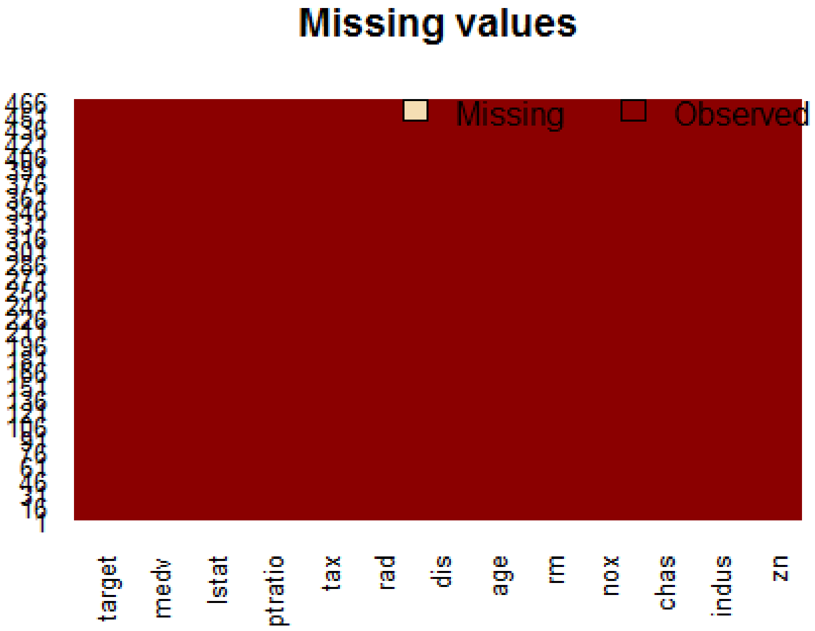
```
##   zn indus chas   nox    rm   age    dis rad tax ptratio lstat medv target
## 1  0 19.58    0 0.605 7.929  96.2 2.0459   5 403    14.7  3.70 50.0      1
## 2  0 19.58    1 0.871 5.403 100.0 1.3216   5 403    14.7 26.82 13.4      1
## 3  0 18.10    0 0.740 6.485 100.0 1.9784  24 666    20.2 18.85 15.4      1
## 4 30  4.93    0 0.428 6.393   7.8 7.0355   6 300    16.6  5.19 23.7      0
## 5  0  2.46    0 0.488 7.155  92.2 2.7006   3 193    17.8  4.82 37.9      0
## 6  0  8.56    0 0.520 6.781  71.3 2.8561   5 384    20.9  7.67 26.5      0
```

We can see that we have a target variable which is in a binary format and we have 12 predictor variables some being categorical. This should not be an issue for a logistic binary regression as it can handle both numerical and categorical values.

Next we will check data for any missing values and develope a strategy to deal with those if we find any

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

We can also visually check to see if there are any missing values, in this case it might not help us much but normally it is always helpful to visualize data and see how much data is missing by various predictors relative to others, we will use Amelia library to do that.



It appears we have a pretty clean data set and we do not have any missing values, so we can proceed to the next step

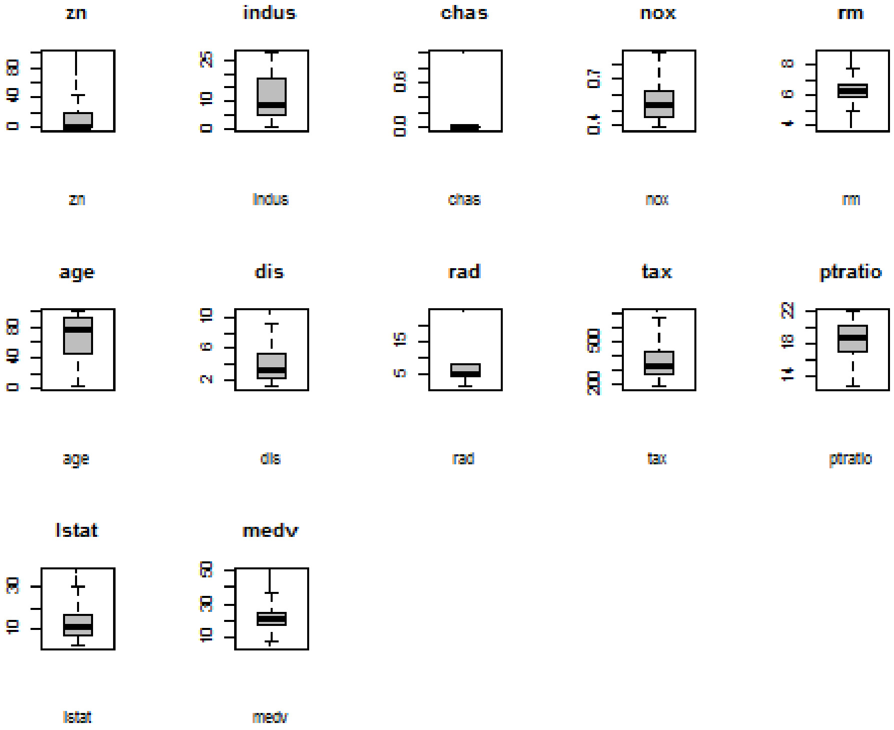
We will set 80 records aside for validation and testing and train our models on the remaining 386 records.

Next we will review the predictors given to us to better understand the data we are dealing with. The table gives us a basic overview of the data

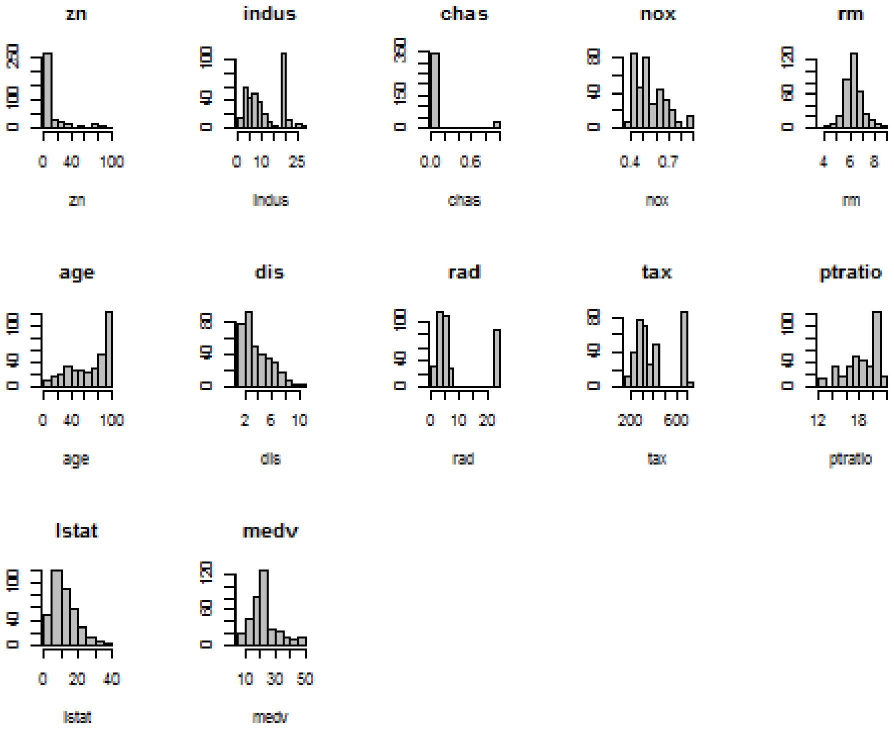
	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
zn	1	365	11.57	23.13	0.00	5.44	0.00	0.00	100.00	100.00	2.21	4.01	1.21
indus	2	365	10.91	6.91	8.56	10.62	7.75	0.46	27.74	27.28	0.38	-1.18	0.36
chas	3	365	0.07	0.26	0.00	0.00	0.00	0.00	1.00	1.00	3.32	9.05	0.01
nox	4	365	0.55	0.12	0.53	0.54	0.13	0.39	0.87	0.48	0.79	0.04	0.01
rm	5	365	6.30	0.73	6.23	6.27	0.53	3.86	8.78	4.92	0.47	1.51	0.04
age	6	365	68.47	28.39	78.30	71.08	29.06	2.90	100.00	97.10	-0.58	-1.01	1.49
dis	7	365	3.80	2.06	3.26	3.56	1.98	1.14	10.71	9.57	0.90	0.09	0.11
rad	8	365	9.03	8.44	5.00	8.09	1.48	1.00	24.00	23.00	1.15	-0.55	0.44
tax	9	365	399.39	165.17	330.00	389.16	108.23	188.00	711.00	523.00	0.76	-0.97	8.65
ptratio	10	365	18.35	2.21	18.70	18.54	2.22	12.60	22.00	9.40	-0.72	-0.44	0.12

lstat	11	365	12.44	7.12	10.74	11.65	6.49	1.92	37.97	36.05	0.97	0.63	0.37
medv	12	365	22.76	9.23	21.50	21.83	5.93	5.00	50.00	45.00	1.04	1.31	0.48

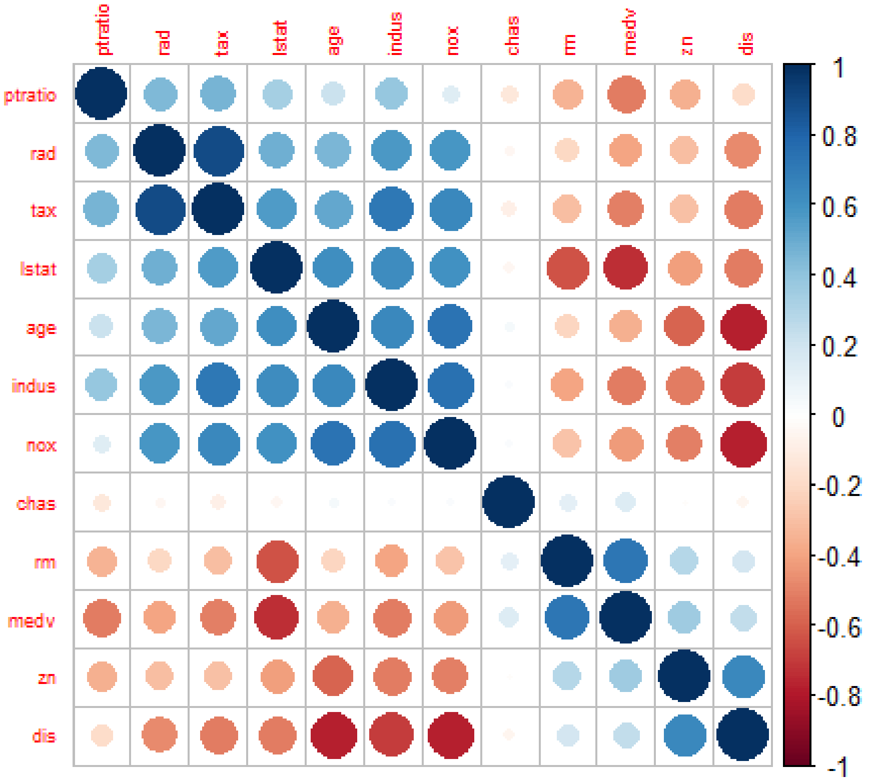
Next we will create a boxplot of each of the predictors to visualize the variability of the data, showing us the variability of the data, the range and various other metrics that allow us to get a better sense of the data we are dealing with.



And finally we will review the histograms of the predictors to see the distribution and the skew.



Next we will create the correlation plot to visually identify correlated predictors.

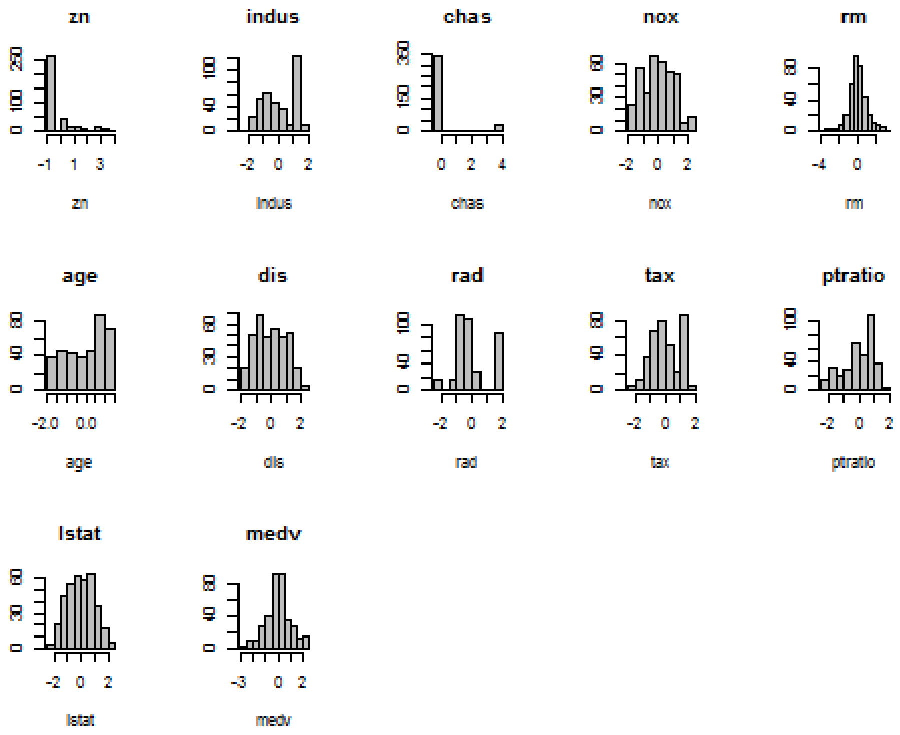


We can see that There are some highly correlated variables. We see that tax variable is correlated to rad, we also see some negative correlations, for example dis is negatively correlated to dis is highly correlated to target and to age, indus and nox, also medv is highly correlated to lstat. We can use this information later in model selection to fine tune our model, but for now we will simply make a note of the existing correlations.

2. DATA PREPARATION

Next we will use the Boxcox transformation to normalize the data. Once normalized, this will allow us to better work with the data. We will use Boxcox transformation to automatically select the appropriate transformation algorithm for our data

Next we will visualize data to see how well it was normalized after boxcox transformation and compare the results to the histograms before we applied the transformation.



Since we are dealing with the dataset that contains a some categorical variables and some variables that have been created as a result of binning of certain categories, it seems that boxcox transformation has not produced any improvements, so we will use the raw data for this study.

3. BUILD MODELS

We will use logit model in order to construct pur models since it seems that logit is the preferred option over probit model.

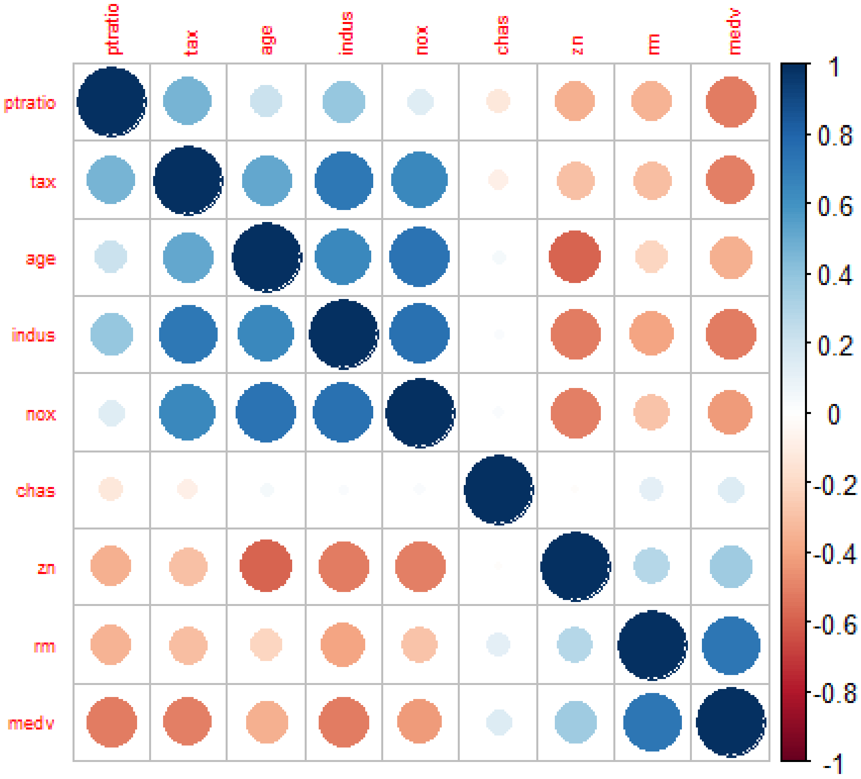
Model1

For model1 we will use all predictor variables, since we do not have an excessive number of variables it is reasonable to expect that each variable will have some contributing factor to the model. We will use this model and compare all future results against this one.

```
##
## Call:  glm(formula = target ~ ., family = binomial(link = "logit"),
##     data = train)
##
## Coefficients:
## (Intercept)           zn        indus         chas          nox
##  -42.378015    -0.043537    -0.090642     0.973488    51.794781
##          rm          age          dis          rad          tax
##   -0.725980     0.050714     0.810891     0.639372    -0.004419
##     ptratio        lstat         medv
##    0.410362    -0.014980     0.174812
##
## Degrees of Freedom: 364 Total (i.e. Null);  352 Residual
## Null Deviance:       505.7
## Residual Deviance: 155.5     AIC: 181.5
```

Model2

For the model2 we will try to remove some of the high correlated variables and decrease overal correlation within the dataset. We will remove rad variable since it is correlated with tax and we will remove dis since it is corelated with age, indus and nox. We will also remove lstat as it is correlated with medv variable. We will check the correlation plot to make sure high correlations are removed



We can see that the highest correlations have been removed from the dataset. Next we will build a model and check the results.

```
##
## Call:  glm(formula = target ~ ., family = binomial(link = "logit"), 
##     data = train1)
##
## Coefficients:
## (Intercept)           zn        indus         chas          nox
##  -27.988672    -0.020649    -0.145060     1.368582    33.999080
##          rm          age          tax      ptratio         medv
##    0.253990     0.018732     0.005024     0.245338     0.073823
##
## Degrees of Freedom: 364 Total (i.e. Null);  355 Residual
## Null Deviance:       505.7
## Residual Deviance: 201.2     AIC: 221.2
```

We can see that model 2 is not as accurate as model 1, so removing highly correlated predictors did not have a positive impact on the mmodel accuracy.

Model3

Since we determined that removing highly correlated variables did not improve our model, we will use model1 and try to improve it by using stepwise procedure and see the variable selection it produces. We will run 3 test variations using backwards, forward and full and see what the experiment shows us, also instead of using logit, we will try and use probit and see what results we get comparable to model1

```
##
## Call:  glm(formula = target ~ ., family = binomial(link = "probit"),
##     data = train)
##
## Coefficients:
## (Intercept)           zn        indus         chas          nox
##  -21.834403    -0.014084    -0.044570     0.501834    27.013556
##          rm          age          dis          rad          tax
##   -0.344626     0.023296     0.373559     0.349823    -0.002787
##     ptratio        lstat         medv
##    0.216747    -0.002256     0.084920
##
## Degrees of Freedom: 364 Total (i.e. Null);  352 Residual
## Null Deviance:       505.7
## Residual Deviance: 159.7     AIC: 185.7
```

Full stepwise

```
##
## Call:
## glm(formula = target ~ nox + age + dis + rad + tax + ptratio +
##     medv, family = binomial(link = "probit"), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9436  -0.2014  -0.0012   0.0000   3.3267
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -20.235245   3.358202  -6.026 1.68e-09 ***
## nox          22.941630   3.836298   5.980 2.23e-09 ***
## age           0.020409   0.006839   2.984  0.00284 **
## dis           0.260967   0.113984   2.290  0.02205 *
## rad           0.397703   0.087646   4.538 5.69e-06 ***
## tax          -0.004197   0.001553  -2.703  0.00687 **
## ptratio       0.191621   0.065811   2.912  0.00360 **
## medv          0.052035   0.019787   2.630  0.00855 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 505.67  on 364  degrees of freedom
## Residual deviance: 164.59  on 357  degrees of freedom
## AIC: 180.59
##
## Number of Fisher Scoring iterations: 10
```

Forward stepwise

```
##
## Call:
## glm(formula = target ~ zn + indus + chas + nox + rm + age + dis +
##     rad + tax + ptratio + lstat + medv, family = binomial(link =
"probit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7999  -0.1586  -0.0001   0.0000   3.6831
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -21.834403   3.899155  -5.600 2.15e-08 ***
## zn           -0.014084   0.016379  -0.860 0.389865
## indus        -0.044570   0.030260  -1.473 0.140775
## chas          0.501834   0.455543   1.102 0.270628
## nox          27.013556   4.824643   5.599 2.15e-08 ***
## rm           -0.344626   0.430799  -0.800 0.423729
## age           0.023296   0.008790   2.650 0.008043 **
## dis           0.373559   0.136942   2.728 0.006375 **
## rad           0.349823   0.094345   3.708 0.000209 ***
## tax          -0.002787   0.001763  -1.581 0.113895
## ptratio       0.216747   0.075536   2.869 0.004112 **
## lstat        -0.002256   0.034571  -0.065 0.947963
## medv          0.084920   0.039294   2.161 0.030683 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 505.67  on 364  degrees of freedom
## Residual deviance: 159.67  on 352  degrees of freedom
## AIC: 185.67
##
## Number of Fisher Scoring iterations: 10
```

Backward Stepwise

```
##
## Call:
## glm(formula = target ~ nox + age + dis + rad + tax + ptratio +
##     medv, family = binomial(link = "probit"), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9436  -0.2014  -0.0012   0.0000   3.3267
##
## Coefficients:
```

```
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept) -20.235245   3.358202  -6.026 1.68e-09 ***
## nox          22.941630   3.836298   5.980 2.23e-09 ***
## age           0.020409   0.006839   2.984  0.00284 **
## dis           0.260967   0.113984   2.290  0.02205 *
## rad           0.397703   0.087646   4.538 5.69e-06 ***
## tax          -0.004197   0.001553  -2.703  0.00687 **
## ptratio       0.191621   0.065811   2.912  0.00360 **
## medv          0.052035   0.019787   2.630  0.00855 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 505.67  on 364  degrees of freedom
## Residual deviance: 164.59  on 357  degrees of freedom
## AIC: 180.59
##
## Number of Fisher Scoring iterations: 10
```

It appears that forward stepwise method is producing the most accurate results with the lowest deviance. So we came to the same conclusion that using all predictor variables produces the most accurate model. We will select stepwise forward as model3 as it had the lowest deviance score.

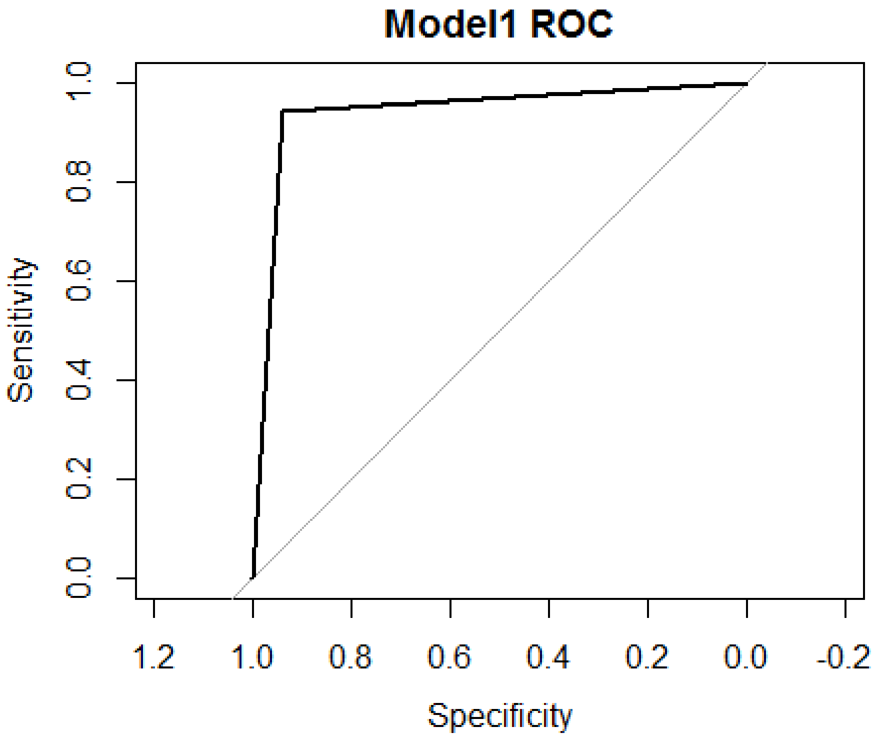
4. SELECT MODELS

We will do some further analyses and will determine how well each of the models actually predicts and compare various metrics such as sensitivity, specificity and others.

Model1

```
##           Reference
## Prediction  0  1
##          0 46  3
##          1  3 49

##          Sensitivity          Specificity       Pos Pred Value
##            0.9423077            0.9387755            0.9423077
##       Neg Pred Value            Precision               Recall
##            0.9387755            0.9423077            0.9423077
##                   F1           Prevalence       Detection Rate
##            0.9423077            0.5148515            0.4851485
## Detection Prevalence    Balanced Accuracy
##            0.5148515            0.9405416
```

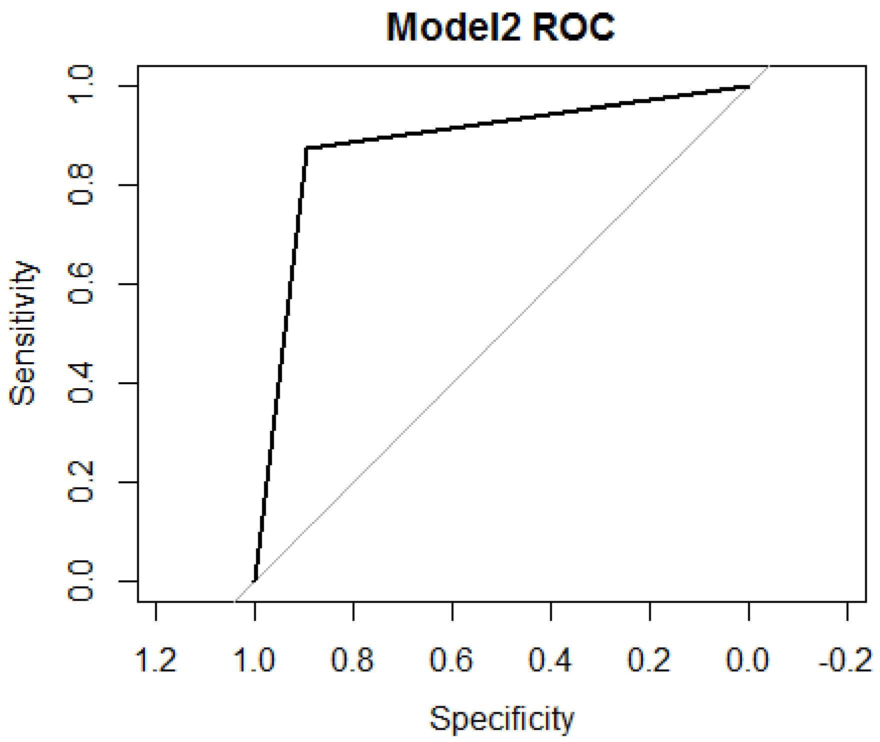


```
## Area under the curve: 0.9405
```

Model2

```
##           Reference
## Prediction  0  1
##          0 42  5
##          1  7 47

##          Sensitivity          Specificity       Pos Pred Value
##            0.9038462            0.8571429            0.8703704
##       Neg Pred Value            Precision               Recall
##            0.8936170            0.8703704            0.9038462
##                   F1           Prevalence       Detection Rate
##            0.8867925            0.5148515            0.4653465
## Detection Prevalence    Balanced Accuracy
##            0.5346535            0.8804945
```

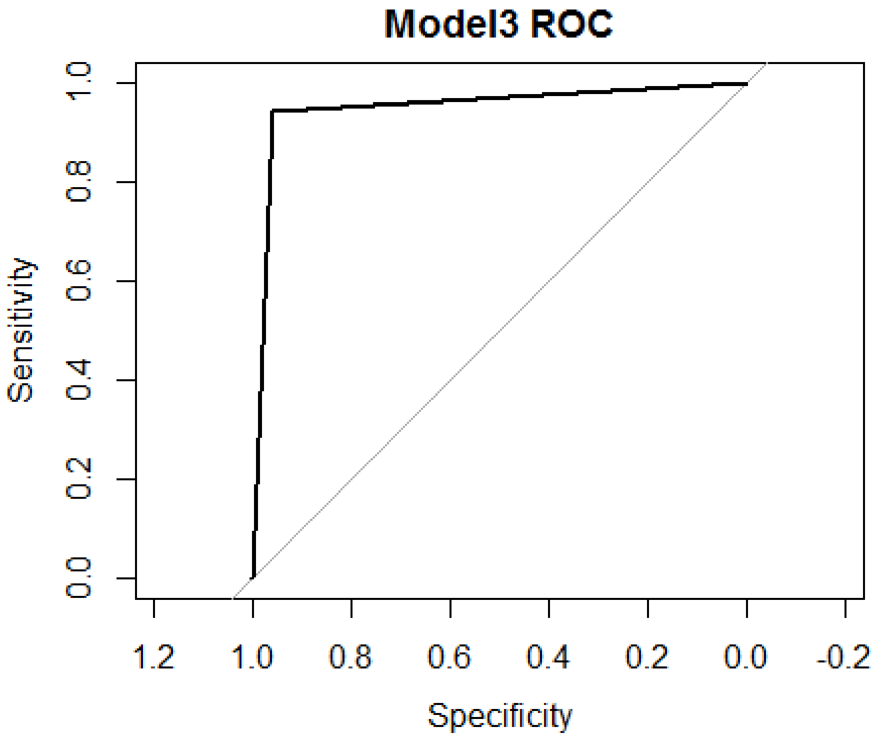


```
## Area under the curve: 0.882
```

Model3

```
##           Reference
## Prediction  0  1
##          0 46  2
##          1  3 50

##          Sensitivity          Specificity       Pos Pred Value
##            0.9615385            0.9387755            0.9433962
##       Neg Pred Value            Precision               Recall
##            0.9583333            0.9433962            0.9615385
##                   F1           Prevalence       Detection Rate
##            0.9523810            0.5148515            0.4950495
## Detection Prevalence    Balanced Accuracy
##            0.5247525            0.9501570
```



```
## Area under the curve: 0.9509
```

Final Model

After running the tests we can see that Model2 has significantly underperformed Model1 and Model3 we can see that across all the metrics as it shows higher deviance, lower sensitivity, specificity and AUC score.

Choosing between model1 and model3 is more difficult since the results are pretty close, but Model3 does outperform Model1, it has better AUC numbers as well as higher sensitivity and specificity rates, Since results are pretty close, I wanted to make sure that its not random and ran model1 and model3 on slightly higher number of test records, as I increased the number of test records, model 3 kept outperforming model1, it seems that at 80 test records they are have the same metrics, but as the number increases model3 is more accurate, so out of the 3 models we will choose Model3 as it is has the best predictive performance and good sensitivity and AUC rates.

Predicting on testdata

We will use model3 to make the predictions on our test dataset. The following are the predictions, you can see the predictions in target column:

zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	target
0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	4.03	34.7	0
0	8.14	0	0.538	6.096	84.5	4.4619	4	307	21.0	10.26	18.2	1
0	8.14	0	0.538	6.495	94.4	4.4547	4	307	21.0	12.80	18.4	1
0	8.14	0	0.538	5.950	82.0	3.9900	4	307	21.0	27.71	13.2	0
0	5.96	0	0.499	5.850	41.5	3.9342	5	279	19.2	8.77	21.0	0
25	5.13	0	0.453	5.741	66.2	7.2254	8	284	19.7	13.15	18.7	0
25	5.13	0	0.453	5.966	93.4	6.8185	8	284	19.7	14.44	16.0	1
0	4.49	0	0.449	6.630	56.1	4.4377	3	247	18.5	6.53	26.6	0
0	4.49	0	0.449	6.121	56.8	3.7476	3	247	18.5	8.44	22.2	0
0	2.89	0	0.445	6.163	69.6	3.4952	2	276	18.0	11.34	21.4	0
0	25.65	0	0.581	5.856	97.0	1.9444	2	188	19.1	25.41	17.3	0
0	25.65	0	0.581	5.613	95.6	1.7572	2	188	19.1	27.26	15.7	0
0	21.89	0	0.624	5.637	94.7	1.9799	4	437	21.2	18.34	14.3	1
0	19.58	0	0.605	6.101	93.0	2.2834	5	403	14.7	9.81	25.0	1
0	19.58	0	0.605	5.880	97.3	2.3887	5	403	14.7	12.03	19.1	1
0	10.59	1	0.489	5.960	92.1	3.8771	4	277	18.6	17.27	21.7	0
0	6.20	0	0.504	6.552	21.4	3.3751	8	307	17.4	3.76	31.5	0
0	6.20	0	0.507	8.247	70.4	3.6519	8	307	17.4	3.95	48.3	1
22	5.86	0	0.431	6.957	6.8	8.9067	7	330	19.1	3.53	29.6	0
90	2.97	0	0.400	7.088	20.8	7.3073	1	285	15.3	7.85	32.2	0
80	1.76	0	0.385	6.230	31.5	9.0892	1	241	18.2	12.93	20.1	0
33	2.18	0	0.472	6.616	58.1	3.3700	7	222	18.4	8.93	28.4	0
0	9.90	0	0.544	6.122	52.8	2.6403	4	304	18.4	5.98	22.1	0
0	7.38	0	0.493	6.415	40.1	4.7211	5	287	19.6	6.12	25.0	0
0	7.38	0	0.493	6.312	28.9	5.4159	5	287	19.6	6.15	23.0	0
0	5.19	0	0.515	5.895	59.6	5.6150	5	224	20.2	10.56	18.5	1
80	2.01	0	0.435	6.635	29.7	8.3440	4	280	17.0	5.99	24.5	0
0	18.10	0	0.718	3.561	87.9	1.6132	24	666	20.2	7.12	27.5	1
0	18.10	1	0.631	7.016	97.5	1.2024	24	666	20.2	2.96	50.0	1
0	18.10	0	0.584	6.348	86.1	2.0527	24	666	20.2	17.64	14.5	1
0	18.10	0	0.740	5.935	87.9	1.8206	24	666	20.2	34.02	8.4	1
0	18.10	0	0.740	5.627	93.9	1.8172	24	666	20.2	22.88	12.8	1
0	18.10	0	0.740	5.818	92.4	1.8662	24	666	20.2	22.11	10.5	1
0	18.10	0	0.740	6.219	100.0	2.0048	24	666	20.2	16.59	18.4	1
0	18.10	0	0.740	5.854	96.6	1.8956	24	666	20.2	23.79	10.8	1

0	18.10	0	0.713	6.525	86.5	2.4358	24	666	20.2	18.13	14.1	1
0	18.10	0	0.713	6.376	88.4	2.5671	24	666	20.2	14.65	17.7	1
0	18.10	0	0.655	6.209	65.4	2.9634	24	666	20.2	13.22	21.4	1
0	9.69	0	0.585	5.794	70.6	2.8927	6	391	19.2	14.10	18.3	1
0	11.93	0	0.573	6.976	91.0	2.1675	1	273	21.0	5.64	23.9	0

Appendix A

R markdown file with code available at:
https://github.com/jelikish/Cuny1/tree/master/Spring2018/621/hw3