Classification

Name: Gabriel Bentley

Date: 9/13/22

Dataset: Housing

https://www.kaggle.com/code/ryanholbrook/binary-classification/data?select=housing.csv

How does linear Classification work?

Logistic regression involves picking a qualitiative target variable and creating a model that will predict the class of the target variable. The linear model for classification will create a decision boundary and use it to seperate the different classes. All observations that land on one side of the boundary will be predicted as one class and the observations that land on the other side will be predicted as the other class.

Import the data field and seperate it into train and test sets

We will read in the housing data field, modify the House age column to be usable for classificiation by giving houses that are 30 years or older the value of 1 and houses that are less then 30 years old the value of 0. Next we factor the target column HouseAge and divide it into train and test sets.

```r
df <- read.csv("housing.csv")

X <- vector(mode="integer", length=nrow(df))
count <- 1

for (i in df$HouseAge) {

  if(i >= 30){
    X[count] <- 1
  }
  else{
    X[count] <- 0
  }
  count <- count + 1

}

df$HouseAge <- as.factor(X)


set.seed(4689)
i <- sample(1:nrow(df), nrow(df)*0.80, replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

Explore the training data

Here we will explore the data to get a better idea of what we are working with, we will str the data, get a summary of the columns, show the first few observations of the data field, output the names of the columns, and show how many observations have N/A values.

```
str(train)
```

```
## 'data.frame':    16512 obs. of  10 variables:
##  $ X          : int  18030 16350 6288 10951 37 10939 19752 20501 4029 9996 ...
##  $ MedInc     : num  4.73 3.43 7.63 1.51 1.41 ...
##  $ HouseAge   : Factor w/ 2 levels "0","1": 2 1 2 1 2 2 1 1 1 1 ...
##  $ AveRooms   : num  6.26 5.57 8.39 2.48 3.75 ...
##  $ AveBedrms  : num  1.126 1.002 1.013 0.907 0.968 ...
##  $ Population : num  1114 2183 253 3356 901 ...
##  $ AveOccup   : num  2.99 3.4 3.37 5.09 2.24 ...
##  $ Latitude   : num  37.2 38 34 33.8 37.8 ...
##  $ Longitude  : num  -122 -121 -118 -118 -122 ...
##  $ MedHouseVal: num  2.72 1.41 3.3 1.38 1.04 ...
```

```
summary(train)
```

```
##        X             MedInc        HouseAge    AveRooms
##  Min.   :    1   Min.   : 0.4999   0:8515   Min.   :  0.8461
##  1st Qu.: 5162   1st Qu.: 2.5596   1:7997   1st Qu.:  4.4362
##  Median :10320   Median : 3.5241            Median :  5.2248
##  Mean   :10331   Mean   : 3.8666            Mean   :  5.4192
##  3rd Qu.:15530   3rd Qu.: 4.7426            3rd Qu.:  6.0295
##  Max.   :20639   Max.   :15.0001            Max.   :141.9091
##    AveBedrms         Population       AveOccup           Latitude
##  Min.   : 0.3333   Min.   :    5   Min.   :   1.066   Min.   :32.54
##  1st Qu.: 1.0064   1st Qu.:  787   1st Qu.:   2.429   1st Qu.:33.93
##  Median : 1.0491   Median : 1165   Median :   2.817   Median :34.26
##  Mean   : 1.0953   Mean   : 1421   Mean   :   3.101   Mean   :35.65
##  3rd Qu.: 1.1000   3rd Qu.: 1722   3rd Qu.:   3.283   3rd Qu.:37.73
##  Max.   :34.0667   Max.   :35682   Max.   :1243.333   Max.   :41.95
##    Longitude       MedHouseVal
##  Min.   :-124.3   Min.   :0.150
##  1st Qu.:-121.8   1st Qu.:1.193
##  Median :-118.5   Median :1.794
##  Mean   :-119.6   Mean   :2.070
##  3rd Qu.:-118.0   3rd Qu.:2.659
##  Max.   :-114.3   Max.   :5.000
```

```
head(train)
```

```
##           X MedInc HouseAge AveRooms AveBedrms Population AveOccup Latitude
## 18031 18030 4.7279        1 6.255376 1.1263441       1114 2.994624    37.24
## 16351 16350 3.4280        0 5.572317 1.0015552       2183 3.395023    38.02
## 6289   6288 7.6286        1 8.386667 1.0133333        253 3.373333    34.05
## 10952 10951 1.5054        0 2.476480 0.9074355       3356 5.092564    33.75
## 38       37 1.4103        1 3.749380 0.9677419        901 2.235732    37.83
## 10940 10939 4.0769        1 5.392593 0.9925926        837 6.200000    33.73
##       Longitude MedHouseVal
## 18031   -121.91       2.720
## 16351   -121.35       1.407
## 6289    -117.90       3.304
## 10952   -117.86       1.375
## 38      -122.28       1.039
## 10940   -117.89       1.639
```

```
names(train)
```

```
##  [1] "X"            "MedInc"       "HouseAge"     "AveRooms"     "AveBedrms"
##  [6] "Population"   "AveOccup"     "Latitude"     "Longitude"    "MedHouseVal"
```

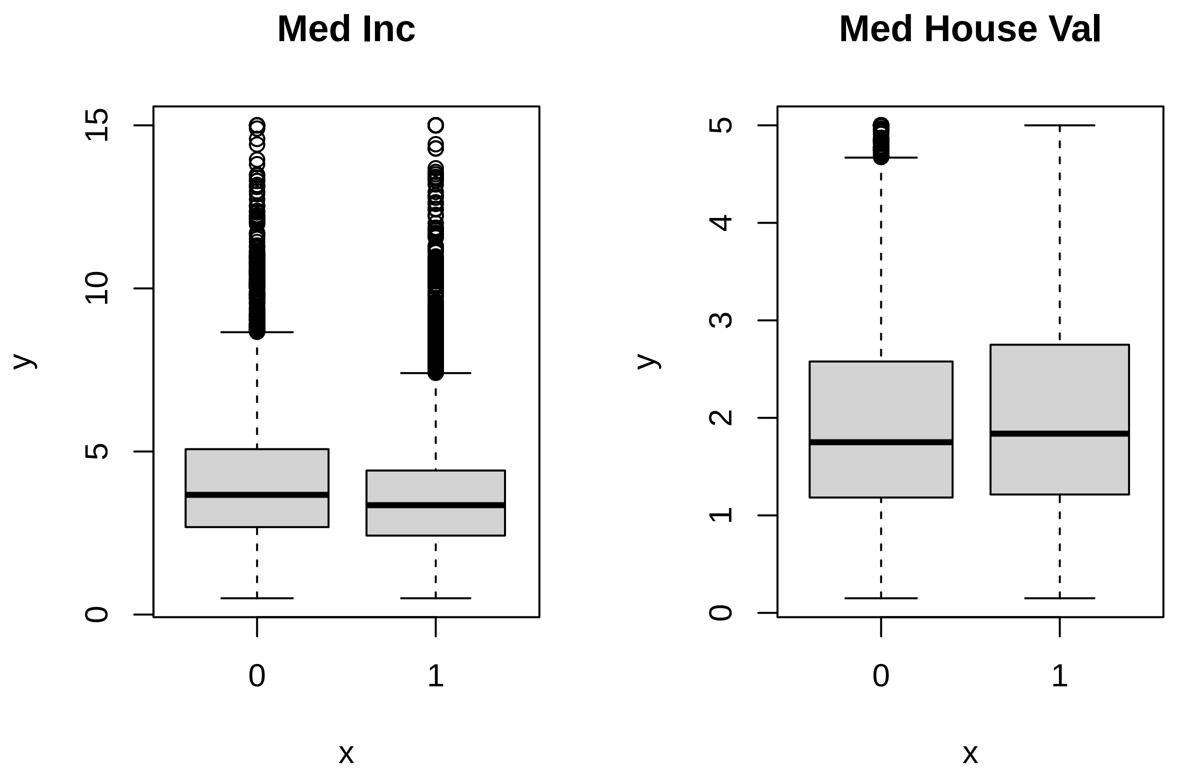
```
colSums(is.na(train))
```

```
##          X      MedInc    HouseAge    AveRooms   AveBedrms  Population
##          0           0           0           0           0           0
##    AveOccup    Latitude   Longitude MedHouseVal
##          0           0           0           0
```

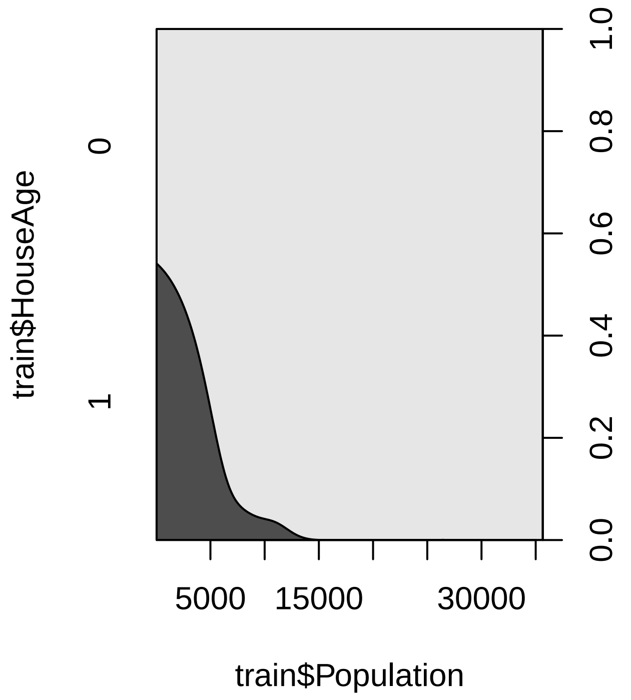
Create graphs

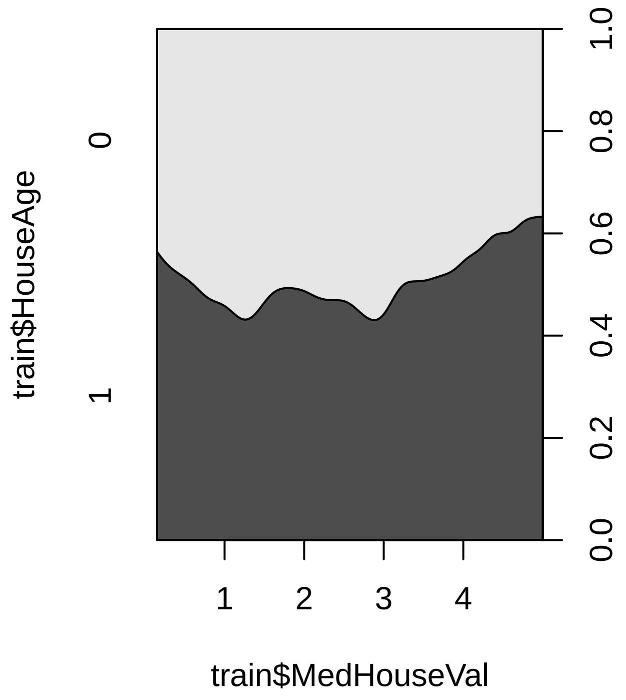
Here we create two sets of graphs to further observe the housing data set. The first two graphs plot House age against average income, and House Age against the average house value. The second set of graphs are cdplots and they compare House age against population, and average house value respectivly. When preforming a cdplot against population the resulting graph was unreadable, so I increased the bandwith until an acceptable result was shown.

```
par(mfrow=c(1,2))
plot(train$HouseAge, train$MedInc, data=train, main="Med Inc",
    varwidth=TRUE)
plot(train$HouseAge, train$MedHouseVal, data=train, main="Med House Val", varwidth=TRUE)
```



```
par(mfrow=c(1,2))
cdplot(train$HouseAge~train$Population, bw = 1400)
cdplot(train$HouseAge~train$MedHouseVal)
```





Build a logistic regression model

Here we create the logistic regression model for predicting if the house age is greater than or equal to 30, or less than 30 year old.

```
glm1 <- glm(train$HouseAge~., data = train, family = binomial)
```

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

```
summary(glm1)
```

```
##
## Call:
## glm(formula = train$HouseAge ~ ., family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -6.0046  -0.9855  -0.2196   0.9723   8.4904
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.729e+01  2.604e+00 -18.165  < 2e-16 ***
## X           -7.176e-05  3.021e-06 -23.753  < 2e-16 ***
## MedInc      -2.784e-01  1.946e-02 -14.306  < 2e-16 ***
## AveRooms    -7.732e-02  2.391e-02  -3.233  0.00122 **
## AveBedrms   -6.173e-01  1.499e-01  -4.118 3.82e-05 ***
## Population  -8.768e-04  2.557e-05 -34.287  < 2e-16 ***
## AveOccup     2.992e-01  2.432e-02  12.302  < 2e-16 ***
## Latitude    -5.374e-01  2.907e-02 -18.488  < 2e-16 ***
## Longitude   -5.759e-01  2.993e-02 -19.241  < 2e-16 ***
## MedHouseVal  3.372e-01  2.566e-02  13.140  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22874  on 16511  degrees of freedom
## Residual deviance: 19293  on 16502  degrees of freedom
## AIC: 19313
##
## Number of Fisher Scoring iterations: 8
```

```
glm2 <- glm(train$HouseAge~train$Population, data = train, family = binomial)
summary(glm2)
```

```
##
## Call:
## glm(formula = train$HouseAge ~ train$Population, family = binomial,
##     data = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.543  -1.154  -0.432   1.106   3.380
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       8.322e-01  3.198e-02   26.03   <2e-16 ***
## train$Population -6.619e-04  2.151e-05  -30.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22874  on 16511  degrees of freedom
## Residual deviance: 21606  on 16510  degrees of freedom
## AIC: 21610
##
## Number of Fisher Scoring iterations: 4
```

```
glm3 <-glm(train$HouseAge~train$MedHouseVal + train$MedInc, data = train, family = binomial)
summary(glm3)
```

```
##
## Call:
## glm(formula = train$HouseAge ~ train$MedHouseVal + train$MedInc,
##     family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1885  -1.1083  -0.8557   1.1767   2.7621
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        0.16582    0.03778   4.389 1.14e-05 ***
## train$MedHouseVal  0.45705    0.02027  22.547  < 2e-16 ***
## train$MedInc      -0.30387    0.01263 -24.052  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22874  on 16511  degrees of freedom
## Residual deviance: 22164  on 16509  degrees of freedom
## AIC: 22170
##
## Number of Fisher Scoring iterations: 4
```

What the summary tells us is The first general linear model is the best one of the three. Based off of the p value for each of the columns they seem to be good predictors for the house age. The Residual deviance is lower than the the Null deviance by around 3500 values, which is is a much better value than the other models. Finally the AIC for the first model which uses all columns is the lowest of the three models making the first glm the best logistical regression model.

Build a naive Bayes model

Here we build a naive Bayes model for classification of HouseAge

```
library(e1071)
nb1 <- naiveBayes(HouseAge~., data=train)
nb1
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##         0         1
## 0.5156856 0.4843144
##
## Conditional probabilities:
##    X
## Y         [,1]     [,2]
##   0 11376.765 5925.407
##   1  9217.322 5801.559
##
##    MedInc
## Y       [,1]     [,2]
##   0 4.046126 1.905467
##   1 3.675532 1.883033
##
##    AveRooms
## Y       [,1]     [,2]
##   0 5.723942 2.579229
##   1 5.094652 2.439129
##
##    AveBedrms
## Y       [,1]      [,2]
##   0 1.125350 0.4809289
##   1 1.063233 0.4861150
##
```

```
##    Population
## Y       [,1]      [,2]
##   0 1685.184 1360.9010
##   1 1139.652  662.7457
##
##    AveOccup
## Y       [,1]     [,2]
##   0 2.912748  1.46299
##   1 3.301807 16.60142
##
##    Latitude
## Y       [,1]     [,2]
##   0 35.72894 2.244312
##   1 35.56175 2.018758
##
##    Longitude
## Y        [,1]     [,2]
##   0 -119.5156 2.033621
##   1 -119.6665 1.975743
##
##    MedHouseVal
## Y       [,1]     [,2]
##   0 1.998985 1.082068
##   1 2.144895 1.226635
```

Here we see the naive Bayes model for HouseAge with all the other columns of the data fields as parameters. In A-priori probabilties we see that the model has the value of .515 for the house age to be less than 30 years old and the likelyhood of the house being 30 years or older is .485. So the likelyhood of either option is roughly random. Since all the predictors are quantitiative the model gives us the standard deviation and mean for each class and the predictor.

Using these two classification models models, predict and evaluate on the test data using all of the classification metrics discussed in class. Compare the results and indicate why you think these results happened.

Here we predict the models and find the accuracy of both models

```r
probs1 <- predict(glm1, newdata = test, type = "response")
pred1 <- ifelse(probs1>0.5, 1, 0)
acc1 <- mean(pred1==test$HouseAge)
print(paste("Logistic Regression Accuracy = ", acc1))
```

```
## [1] "Logistic Regression Accuracy =  0.717054263565892"
```

```r
table(pred1, test$HouseAge)
```

```
##
## pred1    0    1
##     0 1521  535
##     1  633 1439
```

```r
pred2 <- predict(nb1, newdata = test, type = "class")
acc2 <- mean(pred2==test$HouseAge)
print(paste("Naive Bayes Accuracy = ", acc2))
```

```
## [1] "Naive Bayes Accuracy =  0.521075581395349"
```

```r
table(pred2, test$HouseAge)
```

```
##
## pred2    0    1
##     0 2135 1958
##     1   19   16
```

This shows that the logistic regression model is much more accurate when compared to the naive Bayes model which is basically random when predicting the class in the test data field.

Find sensitivity and specificity, and Kappa for the two models using confusion matrix.

Here we construct a confusion matrix for both models and use those matrixes to output the sensitivity, specificity, and Kappa values for both models

```r
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```r
library(ggplot2)
cm1 <- confusionMatrix(test$HouseAge, as.factor(pred1))

cat("Logistic regression \nSensitivity value: ", cm1$byClass['Sensitivity'], "\nSpecificity: ", cm1$byCl
```

```
## Logistic regression
## Sensitivity value:  0.739786
## Specificity:  0.6944981
## Kappa:  0.4342042
```

```r
cm2 <- confusionMatrix(test$HouseAge, pred2)

cat("\n\nNaive Bayes \nSensitivity value: ", cm2$byClass['Sensitivity'], "\nSpecificity: ", cm2$byClass
```

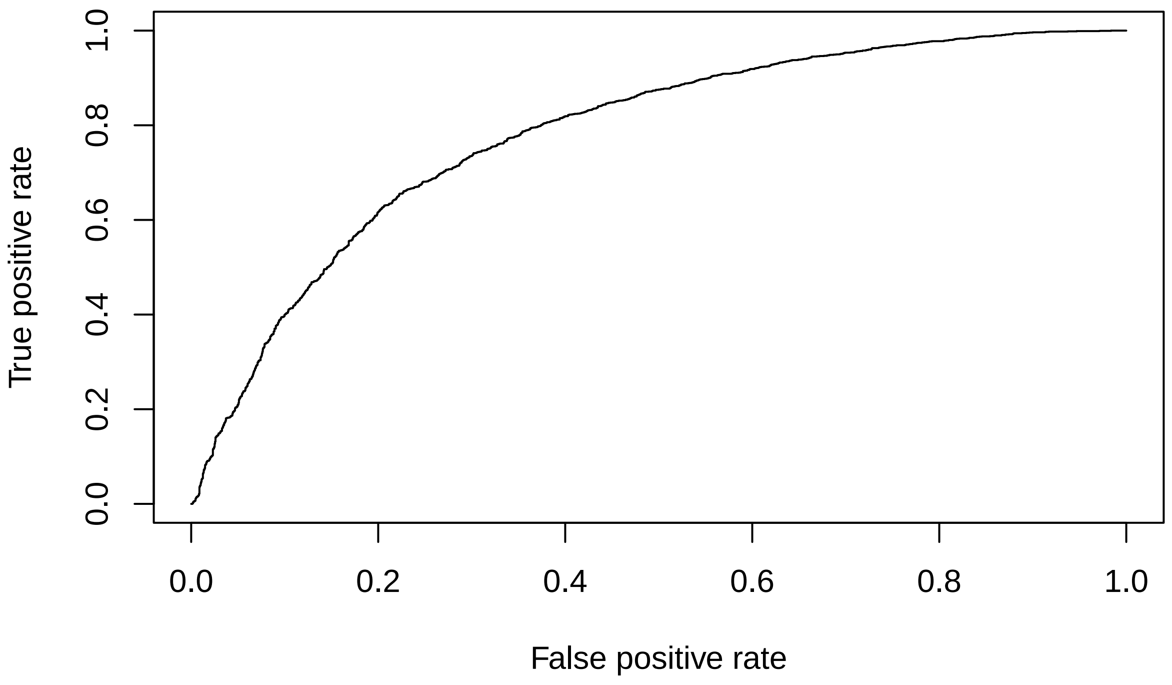
```
##
##
## Naive Bayes
## Sensitivity value:  0.5216223
## Specificity:  0.4571429
## Kappa:  -0.0007460479
```

For the Logistic regression model the true positive rate is about .74 which means that the model has a 74% chance of positivly identifying a house as being 30 years or older when it actually is, and the model has a true negative rate of about .69 which means that the model has a 69% chance of positivly identifying a house as being less than 30 years old when it is. The Kappa value for the logistic regression is about 0.43 which means that there is a moderate agreement between the data being modeled. For the naive Bayes model the true positive rate is about .52 and the true negative rate is about .45 which means that the model is basicly guessing randomly. The kappa value is almost 0 meaning that for the model there is no agreement between the data.

Show the ROC and find the AUC under the curve of ROC for both the logistic regression model and the naive Bayes model

Here we show the ROC for both models and output the area under the curve for each ROC.

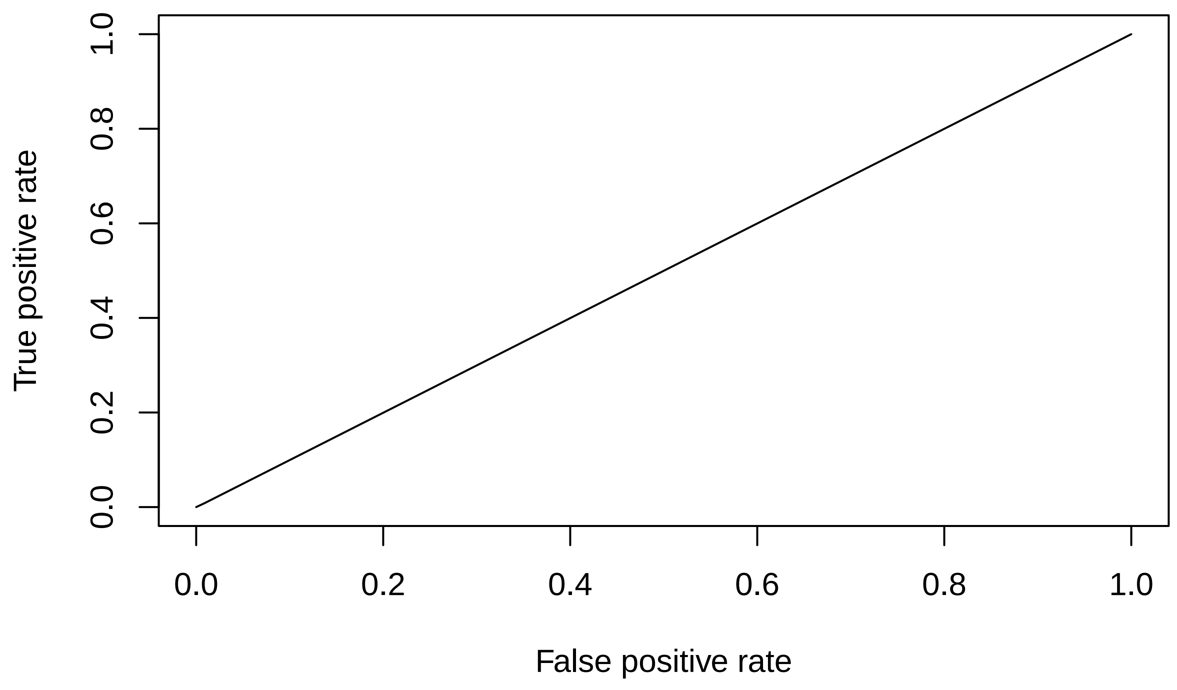
```r
library(ROCR)
p1 <- predict(glm1, newdata = test, type = "response")
pr1 <- prediction(p1, test$HouseAge)

prf1 <- performance(pr1, measure = "tpr", x.measure = "fpr")
plot(prf1)
```



```r
auc1 <- performance(pr1, measure = "auc")
auc1 <- auc1@y.values[[1]]

cat("Area under logistic regression ROC is : ", auc1)
```

```
## Area under logistic regression ROC is :  0.7814269
```

```r
p2 <- predict(nb1, newdata = test, type = "class")

pr2 <- prediction(as.numeric(p2), as.numeric(test$HouseAge))

prf2 <- performance(pr2, measure = "tpr", x.measure = "fpr")
plot(prf2)
```



```
auc2 <- performance(pr2, measure = "auc")
auc2 <- auc2@y.values[[1]]

cat("\nArea under naive Bayes ROC is : ", auc2)
```

```
##
## Area under naive Bayes ROC is :  0.4996423
```

What this means 1. the Logistic Regression is doing pretty good 2. the naive Bayes is preforming randomly

Find the MCC for both models

Here we will find the Mathew correlation coefficient

```
library(mccr)
cat("The Logistic regression MCC is", mccr(test$HouseAge, pred1), "\nThe naive Bayes MCC is", mccr(test$
```

```
## The Logistic regression MCC is 0.4346943
## The naive Bayes MCC is -0.003897695
```

The reason for the random output of the naive Bayes model on the given data field is most likely do to some of the disadvantages of the naive Bayes model. Since the naive Bayes model gives equal importance to all predictors it could be that most of the predictors in the data field have little correlation with the target column. Additionally naive Bayes works better with smaller data sets it could be that a dataset of over 10,000 values was large enough to decrease its effectiveness. And finally when faced with values in the test set that were not in the training set the naive Bayes model guesses randomly.

Compare the the strengths and weaknesses of naive Bayes and logistic regression.

The strengths of the Logistic regression method are that it can separate different classes well if they can be seperated linearly, it is relatively inexpensive to perform a logistic regression computation, and a logistic regression gives a nice probabilistic output. The strengths of the naive Bayes method are its ability to work well with small sets of data, the ease at which it can be implemented, the easy interpretation of its output, and its ability to handle high dimensions well.

The weakness of the logistic regression model is that it is prone to under fitting the data making the model inflexible. The weakness of the naive Bayes model is its underpreformance with large data sets and it's naive assumption that all the predictors in the data set are independent from each other. Additionally when the naive Bayes model encounters values in the test set that were not in the data set it guesses randomly.

List the benefits and drawbacks of each classification metric, and describe what each metric tells us.

Accuracy: Tells us the rate of correct predictions over number of test observations, it has the advantage of being simple and easy to calculate and is the most common classification metric used. The weakness of accuracy is that it is not a good measurement on imbalanced data sets.

Sensitivity and Specificity: Tells us the true positive rate and the true negative rate of the model which means the amount the model guesses a class and it was acctually that class and the amount it does not guess that class and it was not that class. The benefit of this classification method is that it shows if a class could have been misclassified, and the weakness is that there is a tradeoff between sensitivty and specificity where when one increases the other decreases.

Kappa: Tells us the amount of agreement between the predictors two annotators in a data set and adjusts the accuracy of the dataset to account for prediction by random chance. It is found by subtracting the probability of the expected agreement from the probability of the actual agreement and dividing that by 1 - the expected agreement probability. The benefit of Kappa is that it can provide a more comprehensive and objective description of the models performance in addition it can handle imbalanced data sets and multi-class problems. The detriment of the Kappa model is that it makes assumptions about the data that could be wrong in reality leading to an incorrect evaluation of the model.

ROC and AUC: ROC tells us the tradeoff between predicing true valeu rates and false value rates in the form of a graph with a curve on it. AUC tells us teh area under a ROC curve and gives us an indication on how good the model is as a classifier with 0.5 being random, and 1 being perfect. The benefit of ROC and AUC is that it can show people the cutoff values for the data by looking at where the curve bends. the detrement of ROC and AUC is that if the dataset is unbalanced a small number of correct or incorrect guesses by the model can dramatically change the shape of the curve giving the user a false assumption about the data.

MCC: Tells is another form of accuracy but unlike accuracy it takes into account the differences in class distribution. The disadvantage of MCC as a classification metric is that it can only be used for binary classification.