Classification

Ruben Mathew

2023-02-18

What is Logistic Regression?

Logistic Regression is a technique used to define a relationship to classify different values. It really should be called Classification rather than regression. It allows us to approximate a “decision boundary” that can show us visually whether something belongs to one class or another (based on qualitative data rather than quantitative). Just like Linear Regression, Logistic Regression is highly biased as it usually tries to create this decision boundary based on a straight line.

Set up

Here we reset the environment so that we have a clean slate to work with. We load in the income.csv file. The data was found here (https://www.kaggle.com/datasets/wenruliu/adult-income-dataset) on Kaggle.

```
rm(list = ls()) # Reset Environment
df <- read.csv("income.csv")
df[df == "?"] <- NA
df <- df[complete.cases(df), ]
df$income <- factor(df$income)
```

Separate out Training and Test Data

Here we partition the data into training and test data. We do this to more accurately assess the model and how well it fits to data it hasn’t seen before.

```
set.seed(4829)
i <- sample(1:nrow(df), .8*nrow(df), replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

Data Exploration on Training Data

Here we use a few of R's built-in functions to "explore" the data. The head and tail functions show us a few rows from the front and back of the training data (just to see what a few rows look like). The dim and str functions give us some more information on the structure of the data (dimensions and column names/types). Finally the summary function gives us a summary of the data by column (shows min/max, mean, median, etc.). We also use a couple of functions to graph out the relationships between a few of the different columns.

```
# 5 Functions
head(train)
```

	... <int>	workclass <chr>	fnlwgt <int>	education <chr>	educational.num <int>	marital.status <chr>	occupation <chr>
47077	34	Private	111985	HS-grad	9	Married-civ-spouse	Craft-repair
26885	20	Private	279763	11th	7	Never-married	Craft-repair
47210	37	Private	150057	HS-grad	9	Married-civ-spouse	Craft-repair
33689	62	Private	123582	10th	6	Divorced	Other-service
25665	28	Private	133696	Bachelors	13	Never-married	Sales
13049	20	Private	163665	Assoc-acdm	12	Never-married	Adm-clerical

6 rows | 1-8 of 16 columns

```
tail(train)
```

	... <int>	workclass <chr>	fnlwgt <int>	education <chr>	educational.num <int>	marital.status <chr>
40201	46	Private	202560	Some-college	10	Married-civ-spouse
30683	37	Self-emp-not-inc	352882	HS-grad	9	Divorced
37286	47	Private	193285	HS-grad	9	Married-civ-spouse
44543	45	Self-emp-not-inc	315984	HS-grad	9	Married-civ-spouse

35613	23	Local-gov	210029	Some-college	10	Never-married
37237	40	Private	286750	11th	7	Separated

6 rows | 1-7 of 16 columns

```
dim(train)
```

```
## [1] 36177    15
```

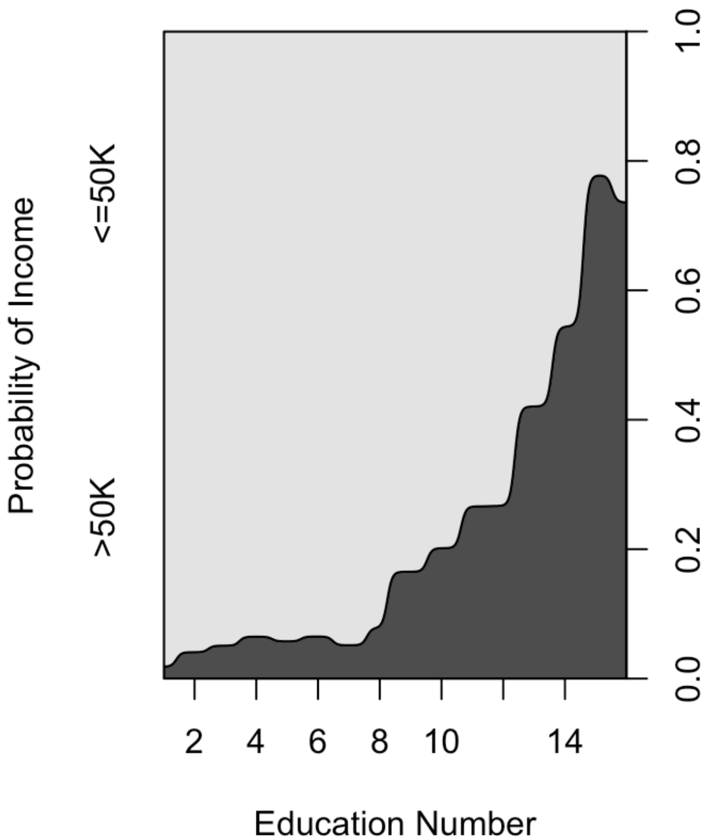
```
str(train)
```

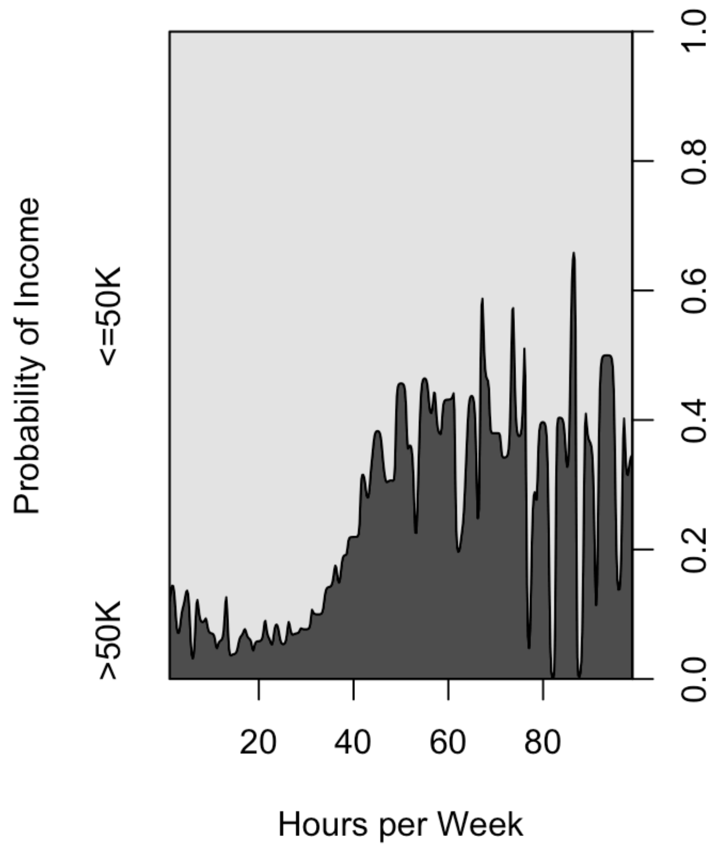
```
## 'data.frame':    36177 obs. of  15 variables:
##  $ age            : int  34 20 37 62 28 20 50 30 35 37 ...
##  $ workclass      : chr  "Private" "Private" "Private" "Private" ...
##  $ fnlwgt         : int  111985 279763 150057 123582 133696 163665 173754 169583 1
42282 189382 ...
##  $ education      : chr  "HS-grad" "11th" "HS-grad" "10th" ...
##  $ educational.num: int  9 7 9 6 13 12 9 13 10 11 ...
##  $ marital.status : chr  "Married-civ-spouse" "Never-married" "Married-civ-spouse"
"Divorced" ...
##  $ occupation     : chr  "Craft-repair" "Craft-repair" "Craft-repair" "Other-servi
ce" ...
##  $ relationship   : chr  "Husband" "Not-in-family" "Husband" "Unmarried" ...
##  $ race           : chr  "White" "Black" "White" "White" ...
##  $ gender         : chr  "Male" "Male" "Male" "Female" ...
##  $ capital.gain   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ capital.loss   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours.per.week : int  45 25 52 40 65 15 40 40 25 38 ...
##  $ native.country : chr  "United-States" "United-States" "United-States" "United-S
tates" ...
##  $ income         : Factor w/ 2 levels "<=50K",">50K": 1 1 2 1 1 1 1 2 1 1 ...
```

```
summary(train)
```

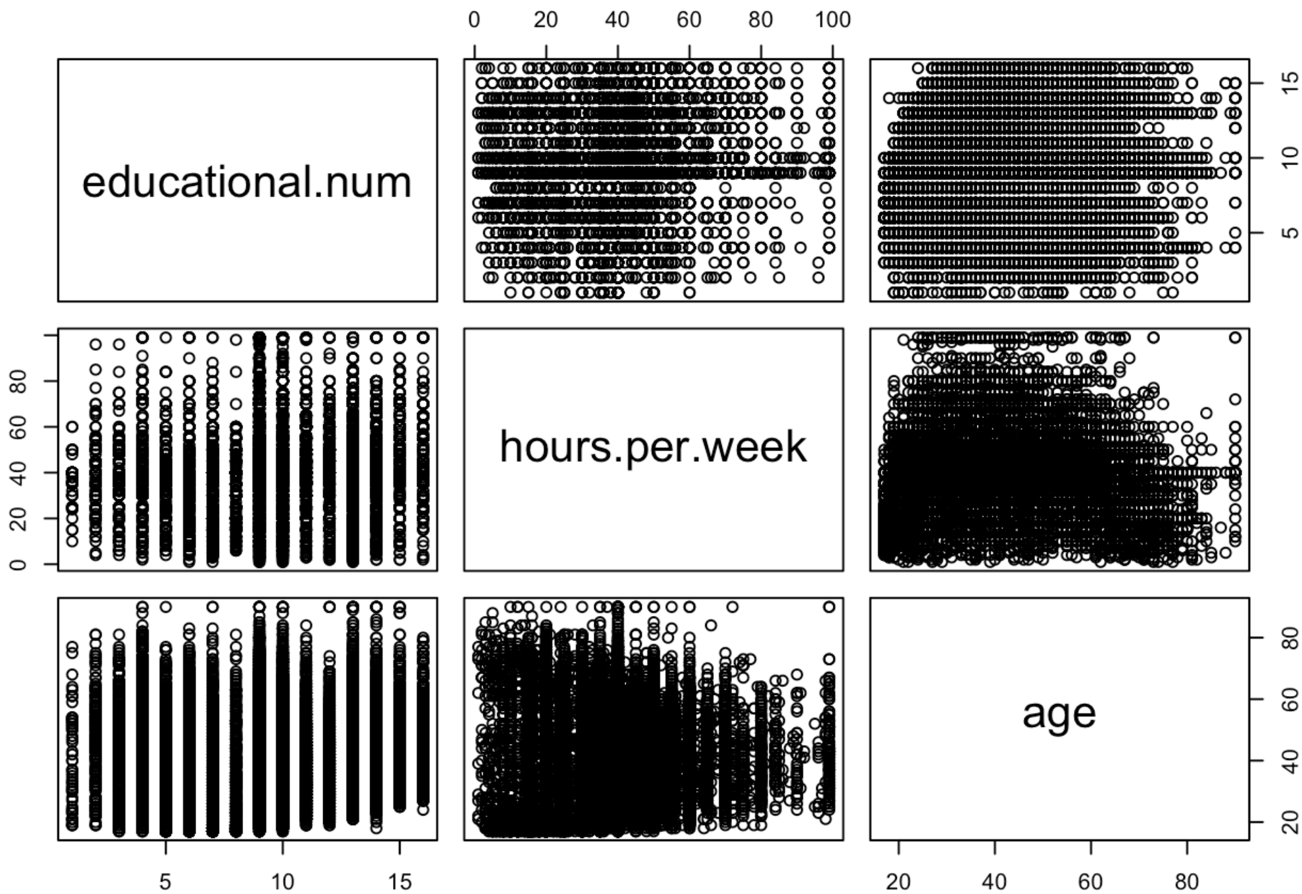
```
##       age         workclass            fnlwgt         education
##  Min.   :17.00   Length:36177       Min.   :  13769   Length:36177
##  1st Qu.:28.00   Class :character   1st Qu.: 117700   Class :character
##  Median :37.00   Mode  :character   Median : 178615   Mode  :character
##  Mean   :38.54                      Mean   : 190372
##  3rd Qu.:47.00                      3rd Qu.: 238917
##  Max.   :90.00                      Max.   :1490400
##  educational.num marital.status      occupation         relationship
##  Min.   : 1.00   Length:36177       Length:36177       Length:36177
##  1st Qu.: 9.00   Class :character   Class :character   Class :character
##  Median :10.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :10.13
##  3rd Qu.:13.00
##  Max.   :16.00
##      race              gender           capital.gain    capital.loss
##  Length:36177       Length:36177       Min.   :    0   Min.   :   0.0
##  Class :character   Class :character   1st Qu.:    0   1st Qu.:   0.0
##  Mode  :character   Mode  :character   Median :    0   Median :   0.0
##                                        Mean   : 1123   Mean   :  89.2
##                                        3rd Qu.:    0   3rd Qu.:   0.0
##                                        Max.   :99999   Max.   :4356.0
##  hours.per.week  native.country       income
##  Min.   : 1.00   Length:36177       <=50K:27157
##  1st Qu.:40.00   Class :character   >50K : 9020
##  Median :40.00   Mode  :character
##  Mean   :40.96
##  3rd Qu.:45.00
##  Max.   :99.00
```

```
# 3 Graph functions
par(mfrow=c(1,2))
cdplot(train$income~train$educational.num, xlab="Education Number", ylab="Probability
of Income")
cdplot(train$income~train$hours.per.week, xlab="Hours per Week", ylab="Probability of
Income")
```






```
pairs(train[,c("educational.num", "hours.per.week", "age")])
```



Train Model and Predict

Here we train the model using the training data. We can make a summary of this model and see then use the built-in predict function to calculate other statistics such as accuracy, and even generate a confusion matrix for the model

```
glm1 <- glm(income~educational.num, data=train, family=binomial)

summary(glm1)
```

```
##
## Call:
## glm(formula = income ~ educational.num, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5388  -0.6846  -0.5820  -0.1445   3.0222
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -4.914812   0.066363  -74.06   <2e-16 ***
## educational.num  0.358334   0.005893   60.81   <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: 40634  on 36176  degrees of freedom
## Residual deviance: 36219  on 36175  degrees of freedom
## AIC: 36223
##
## Number of Fisher Scoring iterations: 4
```

```
pred1 <- predict(glm1, newdata=test, type="response")
probs <- ifelse(pred1>0.5, 2, 1)
acc1 <- mean(probs==as.integer(test$income))
print(paste("glm1 accuracy = ", acc1))
```

```
## [1] "glm1 accuracy =  0.77910447761194"
```

```
table(probs, as.integer(test$income))
```

```
##
## probs    1    2
##     1 6571 1712
##     2  286  476
```

This summary gives us good information on the effect education has on income. In the case of Logistic Regression the estimated slope coefficient found, shows difference of log odds of the target in reference to the independent. In our case, this shows a positive change in log odds for an increase in education. We can also see our Residual Deviance is lower than the Null Deviance which shows a better fit. The accuracy is approximated to 0.779 which is fairly good. But as the confusion matrix below the summary shows us, the amount of observations in the >50K class, is more likely to be incorrectly classified compared to the <=50K class. This makes sense since there is less data on the >50K class and the dataset is imbalanced.

Naive Bayes Model

Naive Bayes Model is another type of Classification model. It allows you to see a few different probabilities based on the concept of likelihood. Here we can train the model and get some idea of how some factors affect the likelihood of an observation being in one class or the other.

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

nb1
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##     <=50K      >50K
## 0.7506703 0.2493297
##
## Conditional probabilities:
##        age
## Y           [,1]     [,2]
##   <=50K 36.72287 13.56695
##   >50K  44.02827 10.41852
##
##        workclass
## Y         Federal-gov    Local-gov      Private Self-emp-inc Self-emp-not-inc
##   <=50K 0.0252236992 0.0653238576 0.7651802482 0.0216518761     0.0801266708
##   >50K  0.0471175166 0.0821507761 0.6496674058 0.0808203991     0.0947893570
##        workclass
## Y           State-gov  Without-pay
##   <=50K 0.0419413043 0.0005523438
##   >50K  0.0452328160 0.0002217295
##
##        fnlwgt
## Y           [,1]     [,2]
##   <=50K 190887.0 107646.5
##   >50K  188821.6 102384.7
```

```
##
##        education
## Y               10th         11th         12th      1st-4th      5th-6th
##   <=50K 0.0343557830 0.0445925544 0.0156129175 0.0060389586 0.0123356777
##   >50K  0.0072062084 0.0073170732 0.0039911308 0.0007760532 0.0019955654
##        education
## Y            7th-8th          9th   Assoc-acdm    Assoc-voc    Bachelors
##   <=50K 0.0228302095 0.0180432301 0.0331774496 0.0417940126 0.1294693817
##   >50K  0.0047671840 0.0033259424 0.0363636364 0.0456762749 0.2830376940
##        education
## Y          Doctorate      HS-grad      Masters    Preschool  Prof-school
##   <=50K 0.0043082815 0.3635158523 0.0337666163 0.0019516147 0.0051183857
##   >50K  0.0361419069 0.2166297118 0.1213968958 0.0001108647 0.0539911308
##        education
## Y       Some-college
##   <=50K 0.2330890746
##   >50K  0.1772727273
##
##        educational.num
## Y            [,1]     [,2]
##   <=50K  9.637699 2.414744
##   >50K  11.598115 2.367129
##
##        marital.status
## Y           Divorced Married-AF-spouse Married-civ-spouse Married-spouse-absent
##   <=50K 0.1644511544      0.0005891667       0.3370769967          0.0141400007
##   >50K  0.0595343681      0.0011086475       0.8533259424          0.0049889135
##        marital.status
## Y       Never-married    Separated      Widowed
##   <=50K  0.4112383548 0.0391795854 0.0333247413
##   >50K   0.0620842572 0.0078713969 0.0110864745
##
##        occupation
## Y       Adm-clerical Armed-Forces Craft-repair Exec-managerial Farming-fishing
##   <=50K 0.1408108407 0.0002945833 0.1368707884    0.0916522444    0.0389954708
##   >50K  0.0681818182 0.0004434590 0.1222838137    0.2527716186    0.0152993348
##        occupation
## Y       Handlers-cleaners Machine-op-inspct Other-service Priv-house-serv
##   <=50K      0.0560444821      0.0775858895  0.1348087049    0.0064440108
##   >50K       0.0131929047      0.0328159645  0.0172949002    0.0002217295
##        occupation
## Y       Prof-specialty Protective-serv        Sales Tech-support
##   <=50K   0.0985013072    0.0197002614 0.1151820893 0.0294951578
##   >50K    0.2409090909    0.0269401330 0.1303769401 0.0376940133
##        occupation
## Y       Transport-moving
##   <=50K     0.0536141695
```

```
##   >50K      0.0415742794
##
##        relationship
## Y           Husband Not-in-family Other-relative   Own-child   Unmarried
##   <=50K 0.298412932   0.307066318    0.037817137 0.195161468 0.129948080
##   >50K  0.759090909   0.109756098    0.004767184 0.009977827 0.026607539
##        relationship
## Y              Wife
##   <=50K 0.031594064
##   >50K  0.089800443
##
##        race
## Y       Amer-Indian-Eskimo Asian-Pac-Islander       Black       Other
##   <=50K        0.011010053        0.025812866 0.109511360 0.009463490
##   >50K         0.005099778        0.033481153 0.046230599 0.004212860
##        race
## Y             White
##   <=50K 0.844202231
##   >50K  0.910975610
##
##        gender
## Y          Female      Male
##   <=50K 0.3830320 0.6169680
##   >50K  0.1490022 0.8509978
##
##        capital.gain
## Y             [,1]       [,2]
##   <=50K  148.9613   912.5785
##   >50K  4057.2542 14849.8262
##
##        capital.loss
## Y            [,1]    [,2]
##   <=50K  54.9008 315.085
##   >50K  192.4748 590.192
##
##        hours.per.week
## Y            [,1]     [,2]
##   <=50K 39.37136 11.98835
##   >50K  45.75477 10.83199
##
##        native.country
## Y           Cambodia       Canada        China     Columbia         Cuba
##   <=50K 4.050521e-04 3.129948e-03 2.172552e-03 2.393490e-03 2.909011e-03
##   >50K  7.760532e-04 5.321508e-03 3.104213e-03 3.325942e-04 2.882483e-03
##        native.country
## Y       Dominican-Republic      Ecuador  El-Salvador      England       France
##   <=50K       2.872188e-03 1.104688e-03 3.976875e-03 2.025261e-03 5.891667e-04
```

```
##   >50K        4.434590e-04 5.543237e-04 7.760532e-04 4.323725e-03 1.330377e-03
##        native.country
## Y            Germany       Greece    Guatemala        Haiti Holand-Netherlands
##   <=50K 4.087344e-03 9.205730e-04 2.467136e-03 1.657031e-03       3.682292e-05
##   >50K  5.432373e-03 1.552106e-03 2.217295e-04 7.760532e-04       0.000000e+00
##        native.country
## Y           Honduras         Hong      Hungary        India         Iran
##   <=50K 5.891667e-04 4.786979e-04 3.682292e-04 2.356667e-03 9.573959e-04
##   >50K  2.217295e-04 6.651885e-04 5.543237e-04 5.654102e-03 2.106430e-03
##        native.country
## Y            Ireland        Italy      Jamaica        Japan         Laos
##   <=50K 8.469271e-04 1.877969e-03 2.467136e-03 1.509740e-03 5.891667e-04
##   >50K  7.760532e-04 3.325942e-03 1.330377e-03 2.993348e-03 2.217295e-04
##        native.country
## Y             Mexico    Nicaragua Outlying-US(Guam-USVI-etc)         Peru
##   <=50K 2.515005e-02 1.362448e-03               6.628125e-04 1.104688e-03
##   >50K  3.991131e-03 3.325942e-04               1.108647e-04 4.434590e-04
##        native.country
## Y        Philippines       Poland     Portugal  Puerto-Rico     Scotland
##   <=50K 5.449792e-03 1.804323e-03 1.583385e-03 4.381927e-03 5.891667e-04
##   >50K  7.427938e-03 1.330377e-03 8.869180e-04 1.773836e-03 1.108647e-04
##        native.country
## Y              South       Taiwan     Thailand Trinadad&Tobago United-States
##   <=50K 2.393490e-03 9.205730e-04 6.259896e-04    7.732813e-04  9.080163e-01
##   >50K  1.884701e-03 2.217295e-03 5.543237e-04    2.217295e-04  9.318182e-01
##        native.country
## Y            Vietnam   Yugoslavia
##   <=50K 2.062083e-03 3.314063e-04
##   >50K  4.434590e-04 7.760532e-04
```

```
pred2 <- predict(nb1, newdata=test, type="class")
table(pred2, test$income)
```

```
##
## pred2   <=50K >50K
##   <=50K  6394 1078
##   >50K    463 1110
```

```
mean(pred2==test$income)
```

```
## [1] 0.8296296
```

From this output we can see that the accuracy is estimated to be 0.829 which is pretty good. We can also see via the confusion matrix, that it is a bit better at accurately classifying >50K observations than the logistic regression model from earlier. But whats nice about the Naive Bayes model is we can also look at the probability tables and see some interesting results. Like if you make >50K, you are much more likely to be married, than someone <=50K. Another thing the model found was if you make >50K, there is about a 28.3% chance your highest level of education is a bachelors, but if you make <=50K that chance drops down to 12.9%. These percentages won't be completely accurate to the population (that's just the nature of samples) but with a large enough amount of data, we can make some good predictions.

Compare and Contrast Metrics

Accuracy

In the Logistic Regression (LR) model, we see that the estimated accuracy is 0.779, compared to the Naive Bayes (NB) model which was estimated at 0.829. This is an indicator that the features of the data might be more independent from each other, since NB assumes this and has a higher bias towards it.

Sensitivity and Specificity

LR Sensitivity - 0.958 LR Specificity - 0.217 NB Sensitivity - 0.932 NB Specificity - 0.507

This shows that while the LR and NB models are fairly accurate, and similar in how they classify <=50K observations, they are both fairly inaccurate in classifying >50K, with NB being the much better option. This inaccuracy can be the result of having much less data on >50k observations than <=50K ones. With an imbalanced dataset like this, NB's assumption of independence can result in a higher specificity.

Kappa

```r
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```r
confusionMatrix(pred2, test$income)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <=50K >50K
##      <=50K  6394 1078
##      >50K    463 1110
##
##                Accuracy : 0.8296
##                  95% CI : (0.8217, 0.8373)
##     No Information Rate : 0.7581
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4863
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9325
##             Specificity : 0.5073
##          Pos Pred Value : 0.8557
##          Neg Pred Value : 0.7057
##              Prevalence : 0.7581
##          Detection Rate : 0.7069
##    Detection Prevalence : 0.8261
##       Balanced Accuracy : 0.7199
##
##        'Positive' Class : <=50K
##
```

```
#confusionMatrix(as.factor(probs), test$income) #Couldn't figure out how to make it w
ork
```

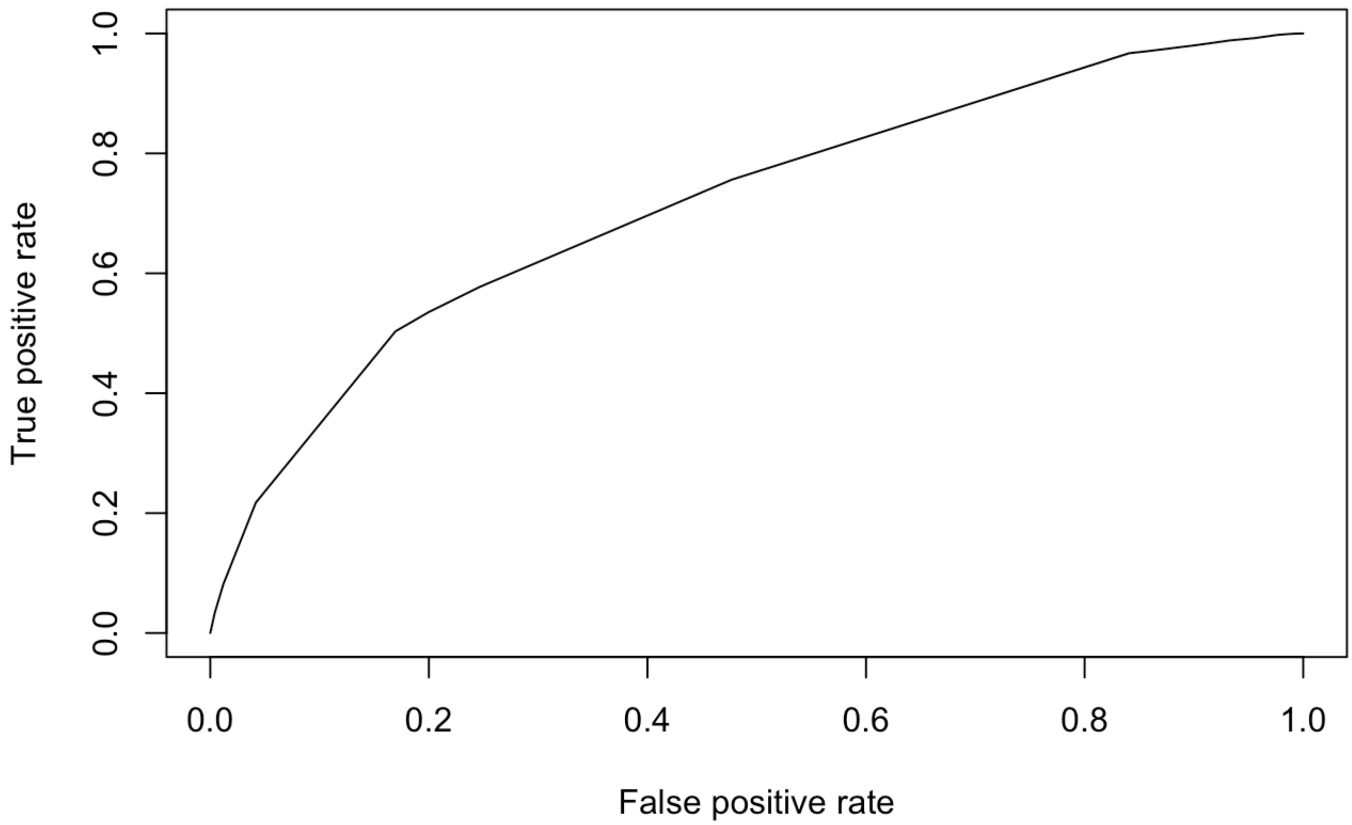
Kappa value for LR - Kappa value for NB - 0.486

Since the kappa for NB model is between 0.4 and 0.6, we can assume a moderate agreement. I was unable to figure out how to convert the predict vector back to a factor to do this for the LR model.

ROC and AUC

```
library(ROCR)
# TPR = sensitivity, FPR=specificity

lnPR <- prediction(pred1, test$income)
prf <- performance(lnPR, measure = "tpr", x.measure = "fpr")
plot(prf)
```




```
# compute AUC
auc <- performance(lnPR, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.7166435
```

AUC LN- 0.716

Since the AUC is 0.716, it is pretty good at classifying the positive observations over the negative ones. With the ROC graph, you notice the gentle incline vs the immediate shoot upwards that we would want to see. This shows a somewhat lack of predictive power.

Comparison of Models

LR and NB are both good models for classification. LR is pretty good because it doesn't make assumptions on the data like NB (which assumes that the features are independent). However it tends to overfit data. NB on the other hand, is simple and requires less data than LR. However NB has that biased independance assumption which can lead it astray.

Comparison of Metrics

Accuracy and Kappa values are both good indicators of how good the model is at classifying the data. Kappa tries to improve on accuracy by trying to negate the chance that the model could have just randomly gotten the value correct.

Sensitivity and Specificity are good at showing whether the model may be better at classifying one type than the other, or might be biased towards one of the classes.