Introduction to Logistic Regression and Simple Machine Learning Classification Techniques

John Zhang

8/12/2020

Machine Learning Classification Models:

Classificaiton models are mathematical models that attempt to draw some conclusion from observed values. Therefore, given one or more inputs, a classification model will predict the outcomes. In this article we will talk about the two main types of machine learning – Supervised and Unsupervised – and how they are used.

Supervised Models:

In Supervised Models, training datasets are fed into a classification algorithm. This allows the model to separate the data into two parts, the "training" data, and the "test" data. The model will create a model based off the training data and validate its predictive capabilities with the test data. Such a learning technique falls under the category of" Classification".

Unsupervised Models:

In Unsupervised models, datasets are inputted into a model without any labels and the algorithm will look for clusters of data points. This is typically used for detecting paterns or outliers in a dataset. An application of this technique could be to group similar images together; such a technique falls under the category of "Clustering".

Applications of Classification Models

Classification models are used widely throughout many different industries including business, healthcare, and sports. A good example of this is how our email accounts can detect spam emails. Given a training set of emails that is comprehensive of the user domain, a classification algorithm will analyze the predictors and map the values of the training set. From this analysis the classificaiton model will encode a binary variable: 1 for the email being spam or 0 for the email not being spam. Then it will validate its performance on the test dataset.

There are many different types of classification models, including logistic regression, decision tree, random forest, gradient-boosted tree, multilayer perceptron, one-vs-rest, and Naive Bayes. However, for our analysis we will focus on logistic regression.

Introduction to Logistic Regression:

Definition: Logistic Regression modeling is a statistical procedure to determine how one or more predictors/labels affect the probability that the response variable takes on one of two outcomes.

A general example of such a concept would be the probability that a student passes a future test. We are able to represent the outcome of the test as 1 for passing and 0 for failing. We then are able to analyze the student's predictors like hours spent studying, hours of sleep before exam day, and volume of water drank immediately before the exam as labels that affect the probability of the student passing the test. This is the motivation of utilizng logistic regression.

General Terms / Vocabuary

$$\pi: \textit{ Probability of Response Variable equaling 1 (Success)}$$

$$Odds : (\frac{\pi}{1-\pi})$$

$$Log\ Odds : log(\frac{\pi}{1-\pi}) = \beta 0 + \beta 1x1 + \beta 2x2 + \beta ...x...$$

Discussion between probabilities and odds:

While probabilities and odds ratios both measure how likelihood an outcome is to happen, probability is a much more intuitive concept. Probability is defined as the likelihood of one outcome or a set of outcomes divided by the total sample space. Odds ratios on the other hand, in the context of logistic regression, represent the constant effect of a predictor "X" on the likelihood one particular outcome will occur. This subtle difference is the reason why we utilize both probbailities and odds ratios in logistic analysis.

Using log odds we are able to represent our logistic regerssion as a linear combination of predicotrs, as shown by the above formula.

Computation of Odds and Probabilities:

$$\textit{Algebrically Equivalent:}$$

$$ODDS = (\frac{\pi}{1-\pi}) = e^{\beta 0 + \beta x1 + \beta x2 + \beta xk}$$

$$Probability = \frac{ODDS}{1 + ODDS}$$

Assumptions of Logistic Regression:

For Logistic Regression to be a valid approach, the data must satisfy two conditions.
1.Observations are independent
2. The number of observations with the value 1, should follow a binomial distribution with parameters n,p. Binom(n,p)

Side Note: For ordinary regression models, the coefficients are estimated by least square criterion. For logistic regression, coefficients are estimated by maximizing likelihood through iterative numerical algorithm.

Example Data Introduction:

In our previous section we explored both classification and logistic regression. In this section we will use an NBA dataset to explore regression coefficients and interpretation of the coefficients in the context of logistic regression as well as perform classification on the dataset.

Exploration of Data

Our Dataset is called NBA_LOGREG.CSV. It is a dataset that keeps the statistics of every NBA player during their rookie year. In addition to their normal statistics, the csv has one additional column: “TARGET_5YRS”. TARGET_5YRS is a binary response variable, this means it takes on the value either 1 or 0. If TARGET_5YRS is equal to 1, it implies that player has made it to 5 or more years in the NBA. Otherwise, TARGET_5YRS is equal to 0, which implies the player either is too young to have played 5 years or did not make it to 5 years in the NBA. Our goal from this dataset is to use logistic regression and classification to help us predict the probabilities that each individual player will last at least 5 years in the NBA based on their rookie statistics.

Column Key:

GP: Games Played
MIN: Minutes
PTS: Points
FGM: Field Goals Made
FGA: Field Goals Attempted
FTA: Free Throws Attempted
OREB: Offensive Rebounds
DREB: Defensive Rebounds
REB: Total Rebounds
AST: Assists
STL: Steals
BLK: Blocks

```
#’We saved our dataset as players. players <- read_csv(’nba_logreg.csv’)’

# Convert Categorical Variable into factor
players$TARGET_5Yrs <- as.factor(players$TARGET_5Yrs)

# Random Split, this is a 75, 25 % split. 75% of the data will be used to train and 25%
#will be used to test.
set.seed(623)
test_rows <- sample(nrow(players), nrow(players) * 0.25)

# Create train and test datasets. The training and test datasets should exlclude each other.
test <- players[test_rows,]
train <- players[-test_rows,]
```

Fitting the appropriate model

Full Model

The first logistic model we look at is the full model with all the predictors in the regression. Use the glm function (general linear model). Then we look at the P values for the coefficients; if it is significantly above our cutoff p value of 0.05 or (alpha), we may consider dropping the predictors from our model altogether. This will produce our reduced model. In our full logistic regression we see that all the predictors except for Games Played, Points, and Free Throw Attempts are insignificant based on their p values and could be dropped.

```
full_model <- glm(TARGET_5Yrs ~GP + MIN + PTS  +  FGM + FGA + FTA + OREB + DREB + REB + AST + STL + BLK
summary(full_model)
```

```
##
## Call:
## glm(formula = TARGET_5Yrs ~ GP + MIN + PTS + FGM + FGA + FTA +
##     OREB + DREB + REB + AST + STL + BLK, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9691  -1.0184   0.5144   0.9141   1.8909
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.014393   0.282306  -7.135 9.64e-13 ***
## GP           0.030901   0.005463   5.657 1.54e-08 ***
## MIN         -0.063951   0.038721  -1.652  0.09862 .
## PTS          0.684468   0.259845   2.634  0.00844 **
## FGM         -0.667756   0.551107  -1.212  0.22564
## FGA         -0.228711   0.146985  -1.556  0.11970
## FTA         -0.452553   0.210212  -2.153  0.03133 *
## OREB         0.098832   1.447476   0.068  0.94556
## DREB        -1.010778   1.452926  -0.696  0.48663
## REB          0.847152   1.441851   0.588  0.55684
## AST          0.299711   0.106330   2.819  0.00482 **
## STL         -0.231979   0.356587  -0.651  0.51533
## BLK          0.679810   0.300765   2.260  0.02380 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1332.9  on 1004  degrees of freedom
## Residual deviance: 1133.1  on  992  degrees of freedom
## AIC: 1159.1
##
## Number of Fisher Scoring iterations: 5
```

Reduced Model:

In our reduced model, we dropped all the predictors in the full model except for GP, PTS, and FTA because their p values were significant in the full analysis. An interpretation for this is in the presence of the three predictors GP, PTS, and FTA, all the other predictors were not significant enough to be included in the overall logistic regression. To check the validity of our reduced model we have to perform the likelihood ratio test, which tests whether the full or reduced model is valid for the regression.

```
reduced_model <- glm(TARGET_5Yrs ~ GP  + PTS +FTA, family = binomial, data = train)
summary(reduced_model)
```

```
##
## Call:
## glm(formula = TARGET_5Yrs ~ GP + PTS + FTA, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5046  -1.0558   0.5935   0.9048   1.8519
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.203102   0.260953  -8.443  < 2e-16 ***
## GP           0.033205   0.005017   6.619 3.63e-11 ***
## PTS          0.069510   0.039463   1.761   0.0782 .
## FTA          0.180095   0.127758   1.410   0.1586
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1332.9  on 1004  degrees of freedom
## Residual deviance: 1165.2  on 1001  degrees of freedom
## AIC: 1173.2
##
## Number of Fisher Scoring iterations: 4
```

Likelihood Ratio Test:

Ho: (Null Hypothesis): Full Model is True
Ha: (Alt Hypothesis): Reduced Model is True

Process: Fit both models and compare their residual deviances. We compare this difference to the Chi squared distribution with the parameter degrees of freedom equal to the number of predictors dropped. We reject the reduced model for large values of the difference between residual deviances.

```
#Comparing the difference in residual Deviances
residual_deviance_difference <- (1124.6 - 1095.5)

# Using the Chi Squared Distribution test, with 9 dropped predictors.
1- pchisq(residual_deviance_difference,9)
```

```
## [1] 0.000623345
```

```
#the computed value was much smaller than alpha at 0.05 so it implies we can use the reduced model
```

Validation Of Our Model / Classification

ROC - AUC CURVE

The ROC curve is a performance measurement for classification problems at various threshold settings. ROC is a probability curve and the AUC (Area Under Curve) Represents the measure of separability. Basically, it tells how much a model is capable of distinguishing between classes. The higher the AUC, the better the model is at classification between classes. For our ROC-AUC Curve we have an AUC of approximately 72%. This informs us that the model we have is decent for classification; it's better than randomly guessing, but could be improved upon greatly in future work. In addition to the ROC - AUC Curve, we also use a confusion matrix to further assess the predictive power of our model.

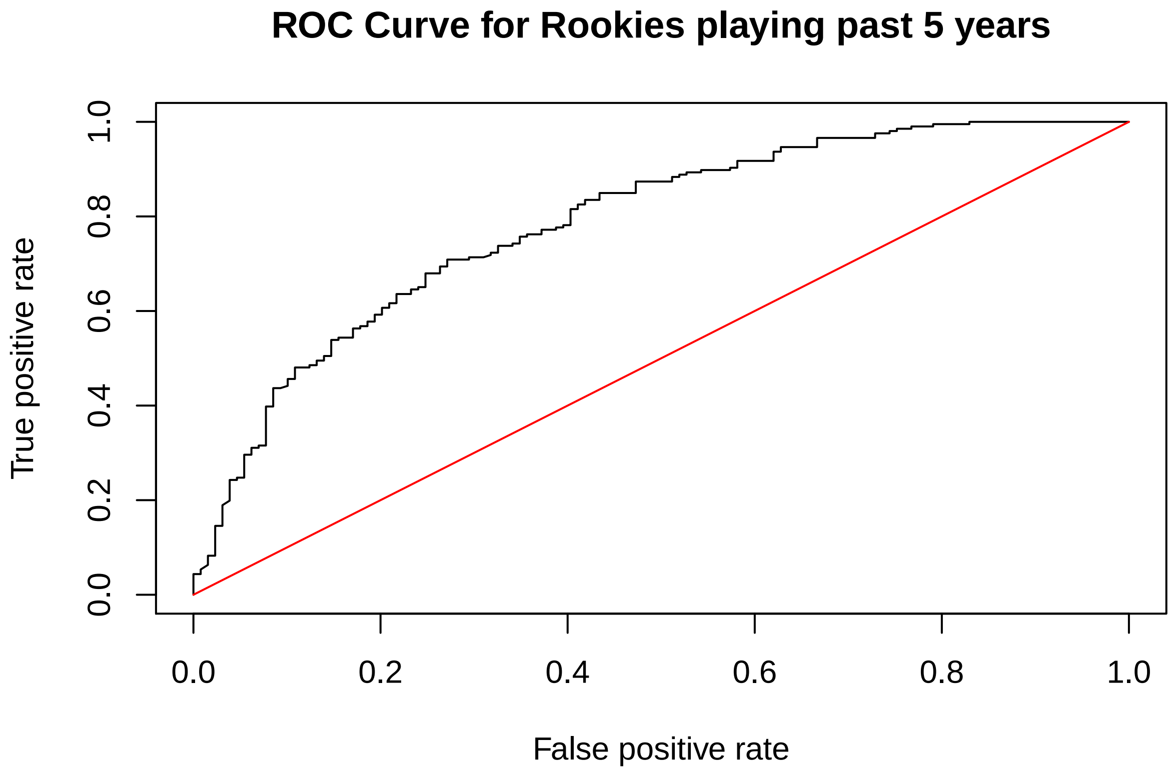
Confusion Matrix

Confusion Matrices are summaries of prediction results. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This allows easy identification of confusion between classes, e.g., one class is commonly mislabeled as the other. Most performance measures are computed from the confusion matrix.

```
rates<- prediction(predict(reduced_model, newdata = test, type = "response"), test$TARGET_5Yrs)
roc_result <- performance(rates, measure = "tpr", x.measure = "fpr")

preds <- predict(reduced_model, newdata = test, type = "response")

plot(roc_result, main = "ROC Curve for Rookies playing past 5 years")
lines(x = c(0, 1),
      y = c(0, 1),
      col = "red")
```



```
auc <- performance(rates, measure = "auc")@y.values[[1]]
print(paste("The AUC for our reduced model is", round(auc, 4)))
```

```
## [1] "The AUC for our reduced model is 0.7852"
```

```
kable(table(test$TARGET_5Yrs, rates@predictions[[1]] > 0.5), caption = "Confusion Matrix")
```

Table 1: Confusion Matrix

	FALSE	TRUE
0	73	56
1	33	173

Confusion Matrix Interpretation:

The Row that starts with 0 implies that the Player did not make it 5 years.
The Row that starts with 1 implies that the player did make it 5 years.
Therefore the interpretation is given that the player never made it 5 years in the NBA. 73 of them were classified as not having made it to 5 years, but 56 of them were classified as making it to 5 years.
And given that the player did make it 5 years, 33 of them were classified as not making it to 5 years while 173 were classified as making it to 5 years.
Using a threshold of 0.5, we can see that:

- Overall error rate is: $(56 + 33)/(73 + 56 + 33 + 173) = 26.56\%$
- False Positive rate is: $33/(33 + 173) = 16.01\%$
- False Negative rate is: $56/(73 + 56) = 43.41\%$
- Sensitivity rate is: $1 - FNR = 56.59\%$
- Specificity rate is: $1 - FPR = 83.99\%$

Predicting Probability that a player will last 5 years or longer in the NBA:

```
probability_table <- cbind(players,preds)
```

```
## Warning in data.frame(..., check.names = FALSE): row names were found from a
## short variable and have been discarded
```

```
new_df <- probability_table[,c("Name","preds")]
print(new_df[1:50,])
```

```
##                  Name     preds
## 1      Brandon Ingram 0.9465730
## 2     Andrew Harrison 0.5202431
## 3      JaKarr Sampson 0.7883192
## 4         Malik Sealy 0.4067638
## 5         Matt Geiger 0.5937282
## 6        Tony Bennett 0.7748671
## 7         Don MacLean 0.1814767
## 8        Tracy Murray 0.8657747
## 9        Duane Cooper 0.9443404
## 10       Dave Johnson 0.8590509
## 11     Corey Williams 0.7874945
## 12           Sam Mack 0.3564075
## 13  Lorenzo Williams 0.3694375
## 14       P.J. Hairston 0.1706581
## 15     Elmore Spencer 0.5512962
## 16        John Crotty 0.3958764
## 17     Stephen Howard 0.3531529
## 18        Randy Woods 0.6557260
## 19      Larry Johnson 0.7516481
## 20      Larry Johnson 0.5181924
## 21        Billy Owens 0.7368804
## 22      Stacey Augmon 0.8545862
## 23         Mark Macon 0.6957209
## 24       Steven Smith 0.4835866
## 25       Mitch McGary 0.3367685
## 26      Larry Stewart 0.3318966
## 27     Mike Iuzzolino 0.6893683
## 28         Doug Smith 0.8206779
## 29        Paul Graham 0.7105482
## 30       Donald Hodge 0.8845469
## 31        Dale Davis 0.8836141
## 32   Stanley Roberts 0.8019779
## 33   Terrell Brandon 0.3153308
## 34        Bison Dele 0.8371562
## 35 Spencer Dinwiddie 0.8845730
## 36       Tracy Moore 0.6546078
## 37      Greg Anthony 0.5144689
## 38    Kenny Anderson 0.9070405
## 39  Victor Alexander 0.3500555
## 40      David Benoit 0.7336354
## 41       Kevin Lynch 0.9396443
## 42  LaBradford Smith 0.7299886
## 43   Chris Corchiani 0.3893108
## 44       Travis Wear 0.7562199
## 45       Robert Pack 0.5319948
## 46     Pete Chilcutt 0.5416013
## 47     Chris Gatling 0.6028908
## 48      Tharon Mayes 0.7793364
## 49      Eric Murdock 0.4255842
## 50       Randy Brown 0.7883870
```

/

Conclusion:

Using the reduced model with predictors GP, PTS, and FTA we found that our logistic regression and classification model was able to distinguish between classes with accuracy 73.44%. The implications of this can be used for future predictions on rookie careers in the NBA. This can be useful for general NBA scouts and evaluators when they are assessing new players. Though more generally, this process details how we can use predictors in our data to predict a binary response. This in turn can be utilized in many other different appliactions than just sports.

/

References:

1.https://data.world/exercises/logistic-regression-exercise-1 (Dataset found at)
2.https://machinelearningmastery.com/types-of-classification-in-machine-learning/
3.https://www.edureka.co/blog/classification-in-machine-learning/
4. Chao Du's Notes. Stat 5120 UVA