Section 9.2 | Classification - Logistic Regression

Mohammad Saqib Ansari

2023-12-19

Logistic Regression

Theory

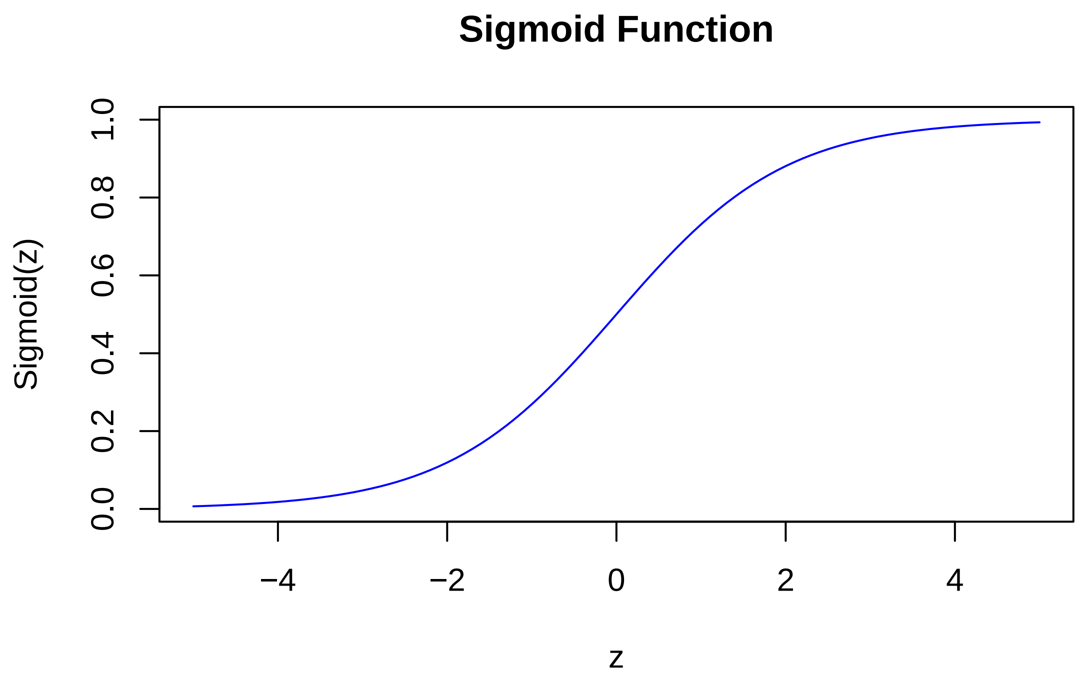
Logistic regression is a statistical method used for modeling the relationship between a categorical dependent variable and one or more independent variables. It's primarily employed in binary classification problems where the outcome variable has two categories (e.g., 0 or 1, Yes or No).

Logistic (Sigmoid) Function

The logistic regression model uses the logistic function (also called the sigmoid function) to transform the output into a probability value between 0 and 1:

$$P(Y = 1|X) = \frac{1}{1 + e^{-z}} \quad \text{where} \quad z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n$$

- $P(Y = 1|X)$ is the probability of the dependent variable being 1 given the values of independent variables X.
- e is the base of natural logarithms.
- $\beta_0, \beta_1, \ldots, \beta_n$ are the coefficients estimated by the model.
- $X_1, X_2, \ldots, X_n$ are the independent variables.



Assumptions

1. **Binary Outcome**: The dependent variable must be binary (having only two categories).
2. **Independence of Observations**: Observations should be independent of each other.
3. **Linear Relationship**: The log odds of the outcome should be linearly related to the independent variables.
4. **No Multicollinearity**: Independent variables should not be highly correlated.
5. **Large Sample Size**: Logistic regression often performs well with a sufficient sample size for accurate estimation.

Model Interpretation

The coefficients (β) in logistic regression represent the change in the log-odds of the dependent variable for a one-unit change in the corresponding independent variable, holding other variables constant.

Model Fitting

The logistic regression model estimates the coefficients by maximizing the likelihood function, aiming to find the set of coefficients that best fit the observed data.

Evaluation

Model evaluation in logistic regression involves various metrics such as confusion matrices, ROC curves, AUC-ROC scores, and precision-recall curves to assess the model's performance and predictive accuracy.

Conclusion

Logistic regression is a fundamental technique used for binary classification tasks. It models the probability of an event occurring based on given inputs and is widely employed in various fields including healthcare, finance, and social sciences. Understanding its assumptions and interpretation of coefficients is crucial for meaningful analysis and interpretation of results.

Logistic Regression (LR) and Linear Discriminant Analysis (LDA) are both techniques used for classification. However, they have distinct differences in their underlying assumptions and methodologies:

Logistic Regression (LR):

1. **Nature of Output:** LR is a regression-based technique used for binary or multi-class classification problems. It models the probability of the outcome using the logistic (sigmoid) function.
2. **Assumptions:** LR assumes no specific distribution for the predictors and the errors. It doesn't make assumptions about the predictors' distribution and allows for non-linear relationships between predictors and the target variable.
3. **Decision Boundary:** LR uses a logistic function to model the relationship between the predictors and the binary outcome. It generates a linear decision boundary in the input space that separates the classes.
4. **Output Interpretation:** LR provides probabilities for the outcome, allowing for interpretation of the likelihood of a given observation belonging to a specific class. It provides coefficients that represent the change in log-odds of the outcome for unit changes in predictors.
5. **Robustness:** LR is relatively robust against outliers.

Linear Discriminant Analysis (LDA):

1. **Nature of Output:** LDA is a technique used for multi-class classification. It models the distribution of predictors separately for each class and calculates posterior probabilities based on Bayes' theorem.
2. **Assumptions:** LDA assumes that predictors are normally distributed within each class and that the classes have a common covariance matrix (homoscedasticity). It also assumes independence of predictors.
3. **Decision Boundary:** LDA finds the linear combination of predictors that best separates different classes by maximizing the between-class variance and minimizing the within-class variance.
4. **Output Interpretation:** LDA provides class-specific probabilities and directly assigns observations to the class with the highest posterior probability based on Bayes' theorem.
5. **Robustness:** LDA assumes normality and equality of covariances among classes, making it sensitive to violations of these assumptions, especially in high-dimensional spaces or when the assumptions are not met.

Differences Summary:

- LR is more flexible in terms of distributional assumptions and suitable for both binary and multi-class classification problems.
- LDA assumes normality, equal covariance matrices, and independence among predictors within classes, and it's primarily used for multi-class problems.
- LR provides probabilities and coefficients for interpreting the influence of predictors, while LDA focuses on maximizing class separation based on Bayes' theorem.

Both LR and LDA have their strengths and weaknesses, and the choice between them often depends on the nature of the problem, assumptions' fulfillment, and the specific characteristics of the dataset.

Step 1: Read the Data

```
# Read the Titanic dataset
datt <- read.csv("titanic.csv")

# Display the first few rows of the dataset
head(datt)
```

```
##   PassengerId Survived Pclass    Sex Age SibSp Parch    Fare Embarked
## 1           1        0      3   male  22     1     0  7.2500        S
## 2           2        1      1 female  38     1     0 71.2833        C
## 3           3        1      3 female  26     0     0  7.9250        S
## 4           4        1      1 female  35     1     0 53.1000        S
## 5           5        0      3   male  35     0     0  8.0500        S
## 6           7        0      1   male  54     0     0 51.8625        S
```

```
# Summary statistics of the dataset
summary(datt)
```

```
##   PassengerId       Survived          Pclass          Sex           
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.00   Length:712        
##  1st Qu.:222.8   1st Qu.:0.0000   1st Qu.:1.00   Class :character  
##  Median :445.0   Median :0.0000   Median :2.00   Mode  :character  
##  Mean   :448.6   Mean   :0.4045   Mean   :2.24                     
##  3rd Qu.:677.2   3rd Qu.:1.0000   3rd Qu.:3.00                     
##  Max.   :891.0   Max.   :1.0000   Max.   :3.00                     
##       Age            SibSp           Parch             Fare       
##  Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Min.   :  0.00  
##  1st Qu.:20.00   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:  8.05  
##  Median :28.00   Median :0.000   Median :0.0000   Median : 15.65  
##  Mean   :29.64   Mean   :0.514   Mean   :0.4326   Mean   : 34.57  
##  3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.: 33.00  
##  Max.   :80.00   Max.   :5.000   Max.   :6.0000   Max.   :512.33  
##    Embarked        
##  Length:712        
##  Class :character  
##  Mode  :character  
##                    
##                    
## 
```

```
# Dimensions of the dataset
dim(datt)
```

```
## [1] 712   9
```

Step 2: Data Preprocessing

```r
# Convert categorical variables 'Embarked' and 'Sex' to factors
datt$Embarked <- factor(datt$Embarked)
datt$Sex <- factor(datt$Sex)

# Check unique values in 'Embarked' and 'Sex' columns
unique(datt$Embarked)
```

```
## [1] S C Q
## Levels: C Q S
```

```r
unique(datt$Sex)
```

```
## [1] male   female
## Levels: female male
```

```r
# Map 'Embarked' categories to numeric values (0, 1, 2)
datt$Embarked <- as.numeric(datt$Embarked) - 1
datt$Sex <- as.numeric(datt$Sex) - 1

# Display the updated dataset
head(datt)
```

```
##   PassengerId Survived Pclass Sex Age SibSp Parch    Fare Embarked
## 1           1        0      3   1  22     1     0  7.2500        2
## 2           2        1      1   0  38     1     0 71.2833        0
## 3           3        1      3   0  26     0     0  7.9250        2
## 4           4        1      1   0  35     1     0 53.1000        2
## 5           5        0      3   1  35     0     0  8.0500        2
## 6           7        0      1   1  54     0     0 51.8625        2
```

```r
# Check for missing values
sum(is.na(datt))   # Check for NA/null values
```

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

Step 3: Constructing and Choosing the Model

```r
# Construct the initial logistic regression model with all variables
model_log <- glm(Survived ~ Pclass + Sex + Age + SibSp + Fare + Embarked,
                 family = binomial, data = datt)

# Summary of the initial model
summary(model_log)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Fare +
##     Embarked, family = binomial, data = datt)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.603711   0.630230   8.892  < 2e-16 ***
## Pclass      -1.231424   0.161903  -7.606 2.83e-14 ***
## Sex         -2.594133   0.215120 -12.059  < 2e-16 ***
## Age         -0.043422   0.008205  -5.292 1.21e-07 ***
## SibSp       -0.375046   0.123295  -3.042  0.00235 **
## Fare         0.001210   0.002422   0.499  0.61744
## Embarked    -0.174678   0.132629  -1.317  0.18782
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 960.90  on 711  degrees of freedom
## Residual deviance: 633.81  on 705  degrees of freedom
## AIC: 647.81
##
## Number of Fisher Scoring iterations: 5
```

```r
# Reconstructing the model without 'Fare' and 'Embarked' due to higher p-values
model_final <- glm(Survived ~ Pclass + Sex + Age + SibSp,
                   family = binomial, data = datt)

# Summary of the final model
summary(model_final)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp, family = binomial,
##     data = datt)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.59083    0.54342  10.288  < 2e-16 ***
## Pclass      -1.31392    0.14091  -9.324  < 2e-16 ***
## Sex         -2.61477    0.21473 -12.177  < 2e-16 ***
## Age         -0.04459    0.00817  -5.457 4.83e-08 ***
## SibSp       -0.37465    0.12093  -3.098  0.00195 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 960.90  on 711  degrees of freedom
## Residual deviance: 636.18  on 707  degrees of freedom
## AIC: 646.18
##
## Number of Fisher Scoring iterations: 5
```

```r
# Building a model considering only 'Age'
model_Age <- glm(Survived ~ Age, family = binomial, data = datt)

# Summary of the 'Age' model
summary(model_Age)
```

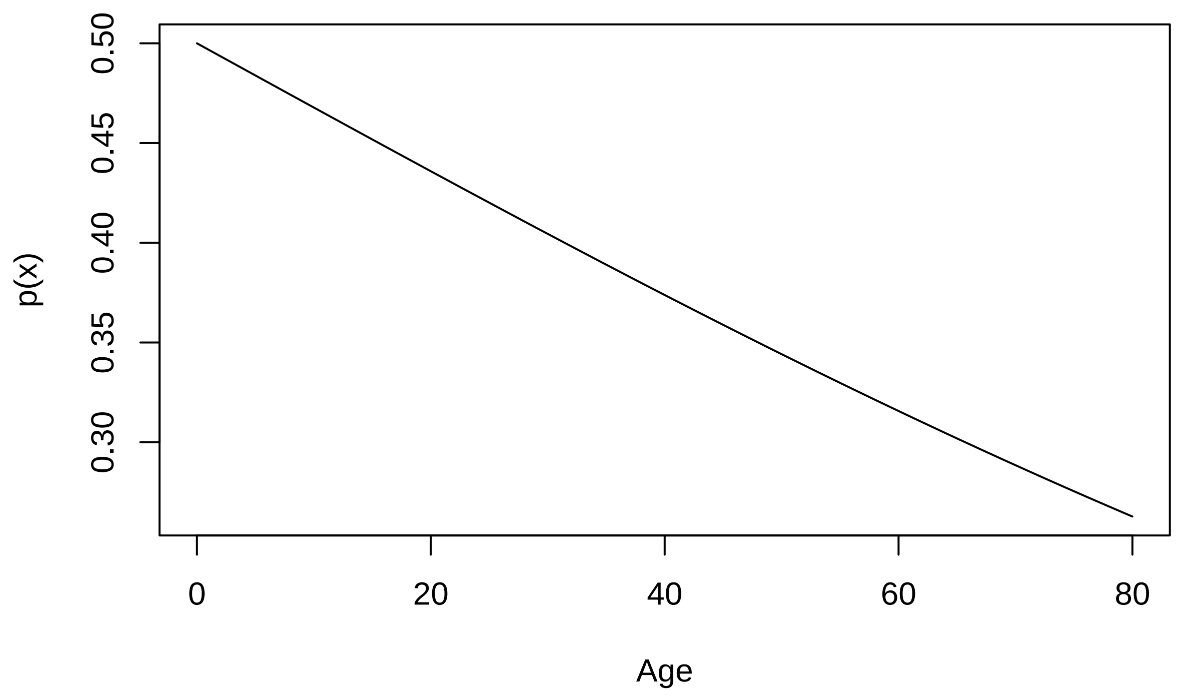
```
##
## Call:
## glm(formula = Survived ~ Age, family = binomial, data = datt)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.041169   0.174070  -0.237   0.8130
## Age         -0.011757   0.005363  -2.192   0.0284 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 960.90  on 711  degrees of freedom
## Residual deviance: 956.03  on 710  degrees of freedom
## AIC: 960.03
##
## Number of Fisher Scoring iterations: 4
```

```r
# Removing intercept as it's not significant
model_Age_wi <- glm(Survived ~ -1 + Age, family = binomial, data = datt)

# Summary of the 'Age' model without intercept
summary(model_Age_wi)
```

```
##
## Call:
## glm(formula = Survived ~ -1 + Age, family = binomial, data = datt)
##
## Coefficients:
##      Estimate Std. Error z value Pr(>|z|)
## Age -0.012897   0.002364  -5.455 4.89e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 987.04  on 712  degrees of freedom
## Residual deviance: 956.08  on 711  degrees of freedom
## AIC: 958.08
##
## Number of Fisher Scoring iterations: 4
```

```r
# Plotting the 'Age' model without intercept
beta <- coef(model_Age_wi)
x <- seq(0, 80, by = 1)
```

```r
y <- exp(beta * x) / (1 + exp(beta * x))

plot(x, y, type = "l", xlab = "Age", ylab = "p(x)")
```



Step 3: Choosing the Model

```r
# Conduct ANOVA tests between models to compare
# anova(model_Age, model_Age_wi)
# anova(model_Age, model_final)

# Stepwise backward selection for variable elimination
model_step <- step(model_final, direction = "backward")
```

```
## Start:  AIC=646.18
## Survived ~ Pclass + Sex + Age + SibSp
##
##          Df Deviance    AIC
## <none>        636.18 646.18
## - SibSp   1   646.70 654.70
## - Age     1   669.11 677.11
## - Pclass  1   741.12 749.12
## - Sex     1   821.40 829.40
```

```
model_step
```

```
##
## Call:  glm(formula = Survived ~ Pclass + Sex + Age + SibSp, family = binomial,
##     data = datt)
##
## Coefficients:
## (Intercept)       Pclass          Sex          Age        SibSp
##     5.59083     -1.31392     -2.61477     -0.04459     -0.37465
##
## Degrees of Freedom: 711 Total (i.e. Null);  707 Residual
## Null Deviance:       960.9
## Residual Deviance: 636.2     AIC: 646.2
```

Step 4: Making Predictions

```
# Creating new data for predictions
new_datt <- as.data.frame(matrix(c(2, 1, 25, 3, 1, 0, 25, 2, 3, 0, 26, 1),
                                  byrow = TRUE, ncol = 4, nrow = 3))
names(new_datt) <- names(datt)[c(3, 4, 5, 6)]

# Predicting using the selected model
predict(model_step, newdata = new_datt, type = "response")
```

```
##         1         2         3
## 0.1312016 0.9178104 0.5287675
```

```
# Calculating missclassification probabilities
prediction_prob <- predict(model_step, newdata = datt)
prediction_label <- as.numeric(prediction_prob > 0.5)

# Creating a table of observed vs. predicted labels
table(datt$Survived, prediction_label)
```

```
##    prediction_label
##       0   1
##   0 396  28
##   1 107 181
```

```
# Calculating the model's missclassification rate
error <- sum(prediction_label != datt$Survived) / nrow(datt)
error
```

```
## [1] 0.1896067
```

```
# Assessing the model's performance using cross-validation
library(boot)
```

```r
# Defining a cost function for cross-validation
cost <- function(Y_obs, prediction_prob) {
  return(mean(abs(Y_obs - prediction_prob) > 0.5))
}

# Cross-validation using bootstrapping
cv.glm(datt, model_step, cost)$delta[1]
```

```
## [1] 0.1938202
```

The provided R Markdown code continues from the previous steps and includes:

- Conducting ANOVA tests to compare different models.
- Implementing stepwise backward selection (`step()`) to refine the model.
- Making predictions on new data using the selected model.
- Calculating missclassification probabilities and generating a table of observed vs. predicted labels.
- Assessing the model's missclassification rate.
- Performing cross-validation using bootstrapping to evaluate the model's performance with a defined cost function.

These steps are crucial in refining the logistic regression model, making predictions, and assessing its performance, providing a comprehensive understanding of the model's predictive abilities and accuracy. Adjustments and interpretations can be made based on the obtained results.