Classification

Kirk Bansak

February 21, 2023

Using Multiple Linear Regression for Continuous Outcomes

Employing linear function to relate outcome y to predictors x_1, x_2, ..., x_p:

$$\mathrm{y}_i \quad = \quad \beta_0 + \beta_1 \mathrm{x}_{1i} + \beta_2 \mathrm{x}_{2i} + ... + \beta_p \mathrm{x}_{pi} + \epsilon_i$$

After Estimation:

$$\mathrm{y}_i \quad = \quad \underbrace{\hat{\beta}_0 + \hat{\beta}_1 \mathrm{x}_{1i} + \hat{\beta}_2 \mathrm{x}_{2i} + ... + \hat{\beta}_p \mathrm{x}_{pi}}_{\widehat{\mathrm{y}_i}} + \hat{\epsilon}_i$$

With $\hat{\beta}_0$, ..., $\hat{\beta}_p$ chosen via:

$$\underset{\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_p}{\arg\min} \quad \sum_{i=1}^{N} \left(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{1i} - \hat{\beta}_2 x_{2i} - ... - \hat{\beta}_p x_{pi} \right)^2$$

Using Multiple Linear Regression for Continuous Outcomes

Employing linear function to relate outcome y to predictors x_1, x_2, ..., x_p:

$$\mathrm{y}_i \quad = \quad \boldsymbol{\beta} \cdot \boldsymbol{x}_i + \epsilon_i$$

After Estimation:

$$\mathrm{y}_i \quad = \quad \underbrace{\hat{\boldsymbol{\beta}} \cdot \boldsymbol{x}_i}_{\widehat{\mathrm{y}}_i} + \hat{\epsilon}_i$$

With $\hat{\boldsymbol{\beta}}$ chosen via:

$$\underset{\hat{\boldsymbol{\beta}}}{\arg\min} \quad \sum_{i=1}^{N} \left(y_i - \hat{\boldsymbol{\beta}} \cdot \boldsymbol{x}_i \right)^2$$

Classification

Intro to Classification

Classification refers to the process of predicting response variables that are qualitative (also often called categorical or discrete).

We will study approaches for classification in the case of binary response variables (response variables that have two possible values).

Two Estimation Goals

Imagine we are trying to predict how Senators will vote on particular issues.

Let Yes_i denote the *ith* Senator's vote, where:
$\text{Yes}_i = 1$ if Senator i votes Yes
$\text{Yes}_i = 0$ if Senator i votes No (or Abstains)

Let $\boldsymbol{x}_i$ denote a vector of predictor values for Senator i.

Two Estimation Goals

Imagine we are trying to predict how Senators will vote on particular issues.

Let Yes_i denote the *ith* Senator's vote, where:
$\text{Yes}_i = 1$ if Senator i votes Yes
$\text{Yes}_i = 0$ if Senator i votes No (or Abstains)

Let $\boldsymbol{x}_i$ denote a vector of predictor values for Senator i.

Two quantities to predict/estimate:

Two Estimation Goals

Imagine we are trying to predict how Senators will vote on particular issues.

Let Yes_i denote the *ith* Senator's vote, where:
$\text{Yes}_i = 1$ if Senator i votes Yes
$\text{Yes}_i = 0$ if Senator i votes No (or Abstains)

Let $\boldsymbol{x}_i$ denote a vector of predictor values for Senator i.

Two quantities to predict/estimate:

- Probability of voting yes: $\widehat{\Pr}(\text{Yes}_i = 1|\boldsymbol{x}_i)$
  - $\rightarrow$ Takes values between 0 and 1.

Two Estimation Goals

Imagine we are trying to predict how Senators will vote on particular issues.

Let Yes_i denote the *ith* Senator's vote, where:
$\text{Yes}_i = 1$ if Senator i votes Yes
$\text{Yes}_i = 0$ if Senator i votes No (or Abstains)

Let $\boldsymbol{x}_i$ denote a vector of predictor values for Senator i.

Two quantities to predict/estimate:

- Probability of voting yes: $\widehat{\Pr}(\text{Yes}_i = 1|\boldsymbol{x}_i)$
  - $\rightarrow$ Takes values between 0 and 1.
- Classification of vote: $\widehat{\text{Yes}_i} = \mathrm{I}\left(\widehat{\Pr}(\text{Yes}_i = 1|\boldsymbol{x}_i) > t\right)$, where t is a threshold
  - $\rightarrow$ If $\widehat{\Pr}(\text{Yes}_i = 1|\boldsymbol{x}_i) > t$, then $\mathrm{I}\left(\widehat{\Pr}(\text{Yes}_i = 1|\boldsymbol{x}_i) > t\right) = 1$, otherwise 0.
  - $\rightarrow$ Takes only the values 0 and 1.

Linear Probability Model

Let y be a binary outcome variable. That is, $y_i \in \{0, 1\}$ for all i.

$$y_i = \boldsymbol{\beta} \cdot \boldsymbol{x}_i + \epsilon_i$$

Linear Probability Model

Let y be a binary outcome variable. That is, $y_i \in \{0, 1\}$ for all i.

$$y_i = \boldsymbol{\beta} \cdot \boldsymbol{x}_i + \epsilon_i$$

Linear probability model employs the usual linear regression process. The β's can be estimated using the exact same process as before (OLS Regression), ignoring the fact that the outcome is binary.

Linear Probability Model

Let y be a binary outcome variable. That is, $y_i \in \{0, 1\}$ for all i.

$$y_i = \boldsymbol{\beta} \cdot \boldsymbol{x}_i + \epsilon_i$$

Linear probability model employs the usual linear regression process. The β's can be estimated using the exact same process as before (OLS Regression), ignoring the fact that the outcome is binary. But now, the predicted/fitted values can be interpreted as predicted probabilities:

$$\widehat{\Pr}(y_i = 1 | \boldsymbol{x}_i) = \widehat{\boldsymbol{\beta}} \cdot \boldsymbol{x}_i$$

Linear Probability Model

Let y be a binary outcome variable. That is, $y_i \in \{0, 1\}$ for all i.

$$y_i = \boldsymbol{\beta} \cdot \boldsymbol{x}_i + \epsilon_i$$

Linear probability model employs the usual linear regression process. The β's can be estimated using the exact same process as before (OLS Regression), ignoring the fact that the outcome is binary. But now, the predicted/fitted values can be interpreted as predicted probabilities:

$$\widehat{\Pr}(y_i = 1 | \boldsymbol{x}_i) = \widehat{\boldsymbol{\beta}} \cdot \boldsymbol{x}_i$$

And classifications can be made as follows:

$$\widehat{y}_i = 1 \text{ if } \widehat{\boldsymbol{\beta}} \cdot \boldsymbol{x}_i > t$$
$$\widehat{y}_i = 0 \text{ if } \widehat{\boldsymbol{\beta}} \cdot \boldsymbol{x}_i \leq t$$

Potential Problems with Linear Probability Model

- Probabilities greater than 1, less than 0
- Potentially implausible relationship between covariates and response

To R...

A Brief Reminder About (Natural) Logarithms

Logarithm (log) is a class of functions.

- $\log_b(z) = x$, where x is the number that solves $b^x = z$.
  - e.g. $\log_{10}(1000) = 3$ (because $10^3 = 1000$)

A Brief Reminder About (Natural) Logarithms

Logarithm (log) is a class of functions.

- $\log_b(z) = x$, where x is the number that solves $b^x = z$.
  e.g. $\log_{10}(1000) = 3$ (because $10^3 = 1000$)
- We call $\log_e$ the natural logarithm (often written as ln), where e is Euler's number (≈ 2.71828).
- And we'll assume $\log_e = \log$
- $\log(e) = 1$ (because $e^1 = e$)

A Brief Reminder About (Natural) Logarithms

Logarithm (log) is a class of functions.

- $\log_b(z) = x$, where x is the number that solves $b^x = z$.
  e.g. $\log_{10}(1000) = 3$ (because $10^3 = 1000$)
- We call $\log_e$ the natural logarithm (often written as ln), where e is Euler's number (≈ 2.71828).
- And we'll assume $\log_e = \log$
- $\log(e) = 1$ (because $e^1 = e$)

Some rules of logarithms

- $\log(a \times b) = \log(a) + \log(b)$
- $\log(\frac{a}{b}) = \log(a) - \log(b)$
- $\log(a^b) = b \log(a)$
- $\log(1) = 0$
- $\log(e) = 1$
- $\log(a)$ does not have a real solution for all $a \leq 0$

Logistic Regression

$y_i \sim \text{Bernoulli}(p_i)$
Let $p_i = \Pr(y_i = 1 | \boldsymbol{x}_i)$

Logistic Regression

$y_i \sim \text{Bernoulli}(p_i)$
Let $p_i = \Pr(y_i = 1 | \boldsymbol{x}_i)$

The logit model imposes linearity in the log-odds:

$$\log\left(\frac{p_i}{1-p_i}\right) = \boldsymbol{\beta} \cdot \boldsymbol{x}_i$$

Logistic Regression

$y_i \sim \text{Bernoulli}(p_i)$
Let $p_i = \Pr(y_i = 1|\boldsymbol{x}_i)$

The logit model imposes linearity in the log-odds:

$$\log\left(\frac{p_i}{1-p_i}\right) \quad = \quad \boldsymbol{\beta} \cdot \boldsymbol{x}_i$$

What values can p_i take?

Logistic Regression

$y_i \sim \text{Bernoulli}(p_i)$
Let $p_i = \Pr(y_i = 1|\boldsymbol{x}_i)$

The logit model imposes linearity in the log-odds:

$$\log\left(\frac{p_i}{1-p_i}\right) = \boldsymbol{\beta} \cdot \boldsymbol{x}_i$$

What values can p_i take?

$$p_i = \frac{\exp(\boldsymbol{\beta} \cdot \boldsymbol{x}_i)}{1 + \exp(\boldsymbol{\beta} \cdot \boldsymbol{x}_i)}$$

Logistic Regression

$y_i \sim \text{Bernoulli}(p_i)$
Let $p_i = \Pr(y_i = 1|\boldsymbol{x}_i)$

The logit model imposes linearity in the log-odds:

$$\log\left(\frac{p_i}{1-p_i}\right) = \boldsymbol{\beta}\cdot\boldsymbol{x}_i$$

What values can p_i take?

$$\begin{aligned} p_i &= \frac{\exp(\boldsymbol{\beta}\cdot\boldsymbol{x}_i)}{1+\exp(\boldsymbol{\beta}\cdot\boldsymbol{x}_i)} \\ &= \frac{1}{1+\exp(-\boldsymbol{\beta}\cdot\boldsymbol{x}_i)} \end{aligned}$$

Logistic Regression

$y_i \sim \text{Bernoulli}(p_i)$
Let $p_i = \Pr(y_i = 1 | \boldsymbol{x}_i)$

The logit model imposes linearity in the log-odds:

$$\log\left(\frac{p_i}{1-p_i}\right) = \boldsymbol{\beta} \cdot \boldsymbol{x}_i$$

What values can p_i take?

$$\begin{aligned} p_i &= \frac{\exp(\boldsymbol{\beta} \cdot \boldsymbol{x}_i)}{1 + \exp(\boldsymbol{\beta} \cdot \boldsymbol{x}_i)} \\ &= \frac{1}{1 + \exp(-\boldsymbol{\beta} \cdot \boldsymbol{x}_i)} \end{aligned}$$

What values will this function produce?

Logistic Regression

$y_i \sim \text{Bernoulli}(p_i)$
Let $p_i = \Pr(y_i = 1 | \boldsymbol{x}_i)$

The logit model imposes linearity in the log-odds:

$$\log\left(\frac{p_i}{1-p_i}\right) = \boldsymbol{\beta} \cdot \boldsymbol{x}_i$$

What values can p_i take?

$$p_i = \frac{\exp(\boldsymbol{\beta} \cdot \boldsymbol{x}_i)}{1 + \exp(\boldsymbol{\beta} \cdot \boldsymbol{x}_i)} = \frac{1}{1 + \exp(-\boldsymbol{\beta} \cdot \boldsymbol{x}_i)}$$

What values will this function produce?

Important functions:

Logistic Regression

$y_i \sim \text{Bernoulli}(p_i)$
Let $p_i = \Pr(y_i = 1 | \boldsymbol{x}_i)$

The logit model imposes linearity in the log-odds:

$$\log\left(\frac{p_i}{1-p_i}\right) = \boldsymbol{\beta} \cdot \boldsymbol{x}_i$$

What values can p_i take?

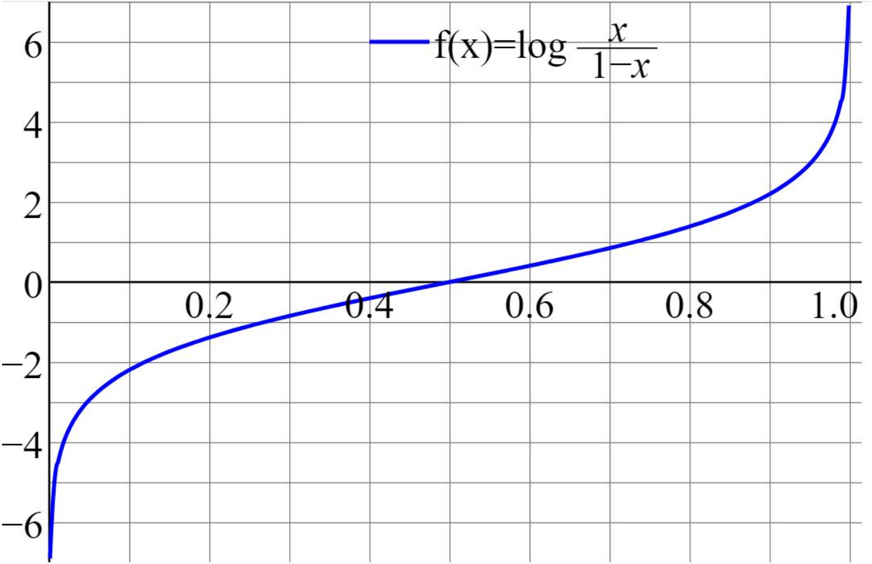
$$p_i = \frac{\exp(\boldsymbol{\beta} \cdot \boldsymbol{x}_i)}{1+\exp(\boldsymbol{\beta} \cdot \boldsymbol{x}_i)} = \frac{1}{1+\exp(-\boldsymbol{\beta} \cdot \boldsymbol{x}_i)}$$

What values will this function produce?

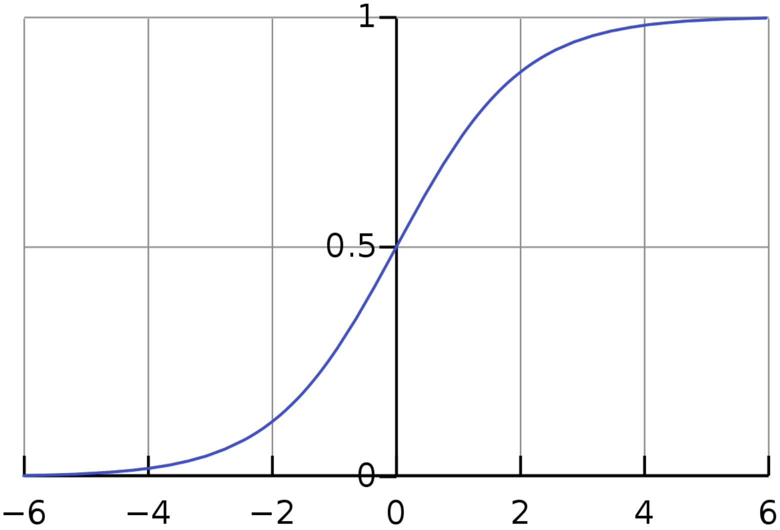
Important functions:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) \qquad \text{logit}^{-1}(a) = \frac{1}{1+\exp(-a)}$$

Logit Function



Logistic ($Logit^{-1}$) Function



Bonus Slide: How Logistic Regression Models are Estimated

Maximum Likelihood Estimation (MLE):

- Assumption about data generating process $\rightsquigarrow$ likelihood function (an objective function that measures goodness of fit of model to data)

$$\prod_{i=1}^{N} p_i^{y_i}(1-p_i)^{1-y_i}$$

$$L(\beta) = \prod_{i=1}^{N} \left(\frac{1}{1+\exp(-\boldsymbol{\beta} \cdot \boldsymbol{x}_i)} \right)^{y_i} \left(1 - \left(\frac{1}{1+\exp(-\boldsymbol{\beta} \cdot \boldsymbol{x}_i)} \right) \right)^{1-y_i}$$

- Coefficient values $\boldsymbol{\beta}$ chosen to maximize the likelihood $L(\boldsymbol{\beta})$
- Computational optimization methods used for MLE

Fitting a Logistic Regression in R

- We use the `glm` function to fit the model
- We must be very careful interpreting the coefficients and extracting predictions when using `glm`!!!

Predicting with a Logistic Regression

Predicting with a Logistic Regression

Predicting Probabilities:

$$\widehat{p}_i \quad = \quad \frac{1}{1 + \exp(-\widehat{\boldsymbol{\beta}} \cdot \boldsymbol{x}_i)}$$

Predicting with a Logistic Regression

Predicting Probabilities:

$$\widehat{p}_i = \frac{1}{1 + \exp(-\widehat{\boldsymbol{\beta}} \cdot \boldsymbol{x}_i)}$$

Classification:

$$\begin{aligned} \widehat{y}_i &= 1 \text{ if } \widehat{p}_i > t \\ \widehat{y}_i &= 0 \text{ if } \widehat{p}_i \leq t \end{aligned}$$

for some threshold t between 0 and 1.

To R...

Some Classification Performance Metrics

		TRUE CONDITION	
		Positive	Negative
PREDICTED CONDITION	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

Some Classification Performance Metrics

		TRUE CONDITION	
		Positive	Negative
PREDICTED CONDITION	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Some Classification Performance Metrics

		TRUE CONDITION	
		Positive	Negative
PREDICTED CONDITION	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

Some Classification Performance Metrics

		TRUE CONDITION	
		Positive	Negative
PREDICTED CONDITION	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Some Classification Performance Metrics

		TRUE CONDITION	
		Positive	Negative
PREDICTED CONDITION	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Some Classification Performance Metrics

		TRUE CONDITION	
		Positive	Negative
PREDICTED CONDITION	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Some Classification Performance Metrics

		TRUE CONDITION	
		Positive	Negative
PREDICTED CONDITION	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$False\ Positive\ Rate\ (FPR) = \frac{FP}{FP + TN}$$

Some Classification Performance Metrics

		TRUE CONDITION	
		Positive	Negative
PREDICTED CONDITION	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$False\ Positive\ Rate\ (FPR) = \frac{FP}{FP + TN}$$

$$False\ Negative\ Rate\ (FNR) = \frac{FN}{FN + TP}$$

Some Classification Performance Metrics

		TRUE CONDITION	
		Positive	Negative
PREDICTED CONDITION	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$False\ Positive\ Rate\ (FPR) = \frac{FP}{FP + TN}$$

$$False\ Negative\ Rate\ (FNR) = \frac{FN}{FN + TP}$$

1 - FNR = Recall = Sensitivity

1 - FPR = Specificity