

# Logistic Regression

## Outline

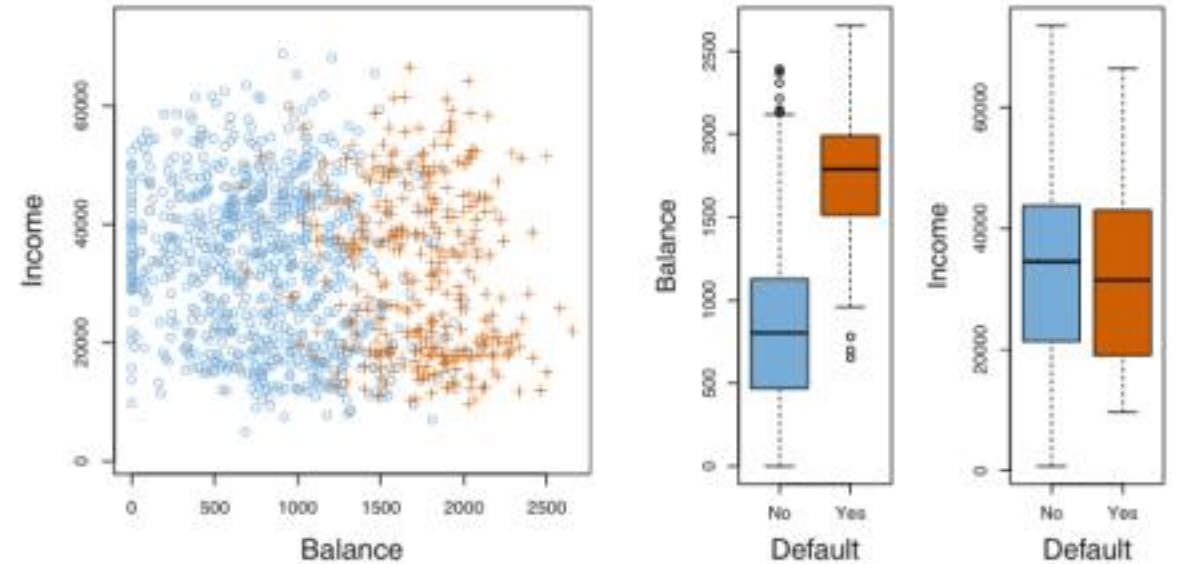
- Introduction
- Overview of Classification
- Logistic Regression
- Multiple Logistic Regression
- Advantages and Disadvantages

# Introduction

- The linear regression model discussed in Lesson 2 assumes that the response variable  $Y$  is **quantitative**. But in many situations, the response variable is instead **qualitative**.
- Often qualitative variables are referred to as categorical. In this lesson, we study approaches for predicting qualitative responses, a process that is known as **classification**.
- Often the methods used for classification **first predict the probability of each of the categories** of a qualitative variable, **as the basis** for making the classification.
- In this lesson we discuss one of the most widely-used classifiers: **logistic regression**

# Overview of Classification

- Classification problems occur often, perhaps even more so than regression problems.
- In the classification setting we have a set of training observations  $(x_1, y_1), \dots, (x_n, y_n)$  that we can use to build a classifier.
- For example in **Default data set** we are interested in predicting whether an individual will default on his or her credit card payment, on the basis of annual income and monthly credit card balance.
- How to build a model to predict default (Y) for any given value of balance ( $X_1$ ) and

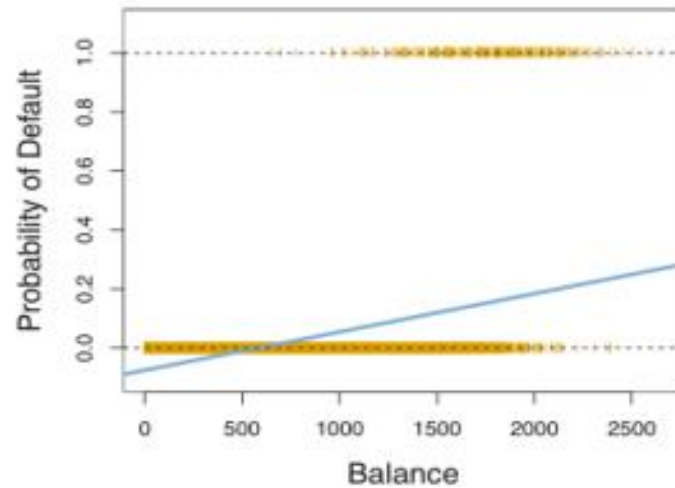


income ( $X_2$ ). Since Y is not quantitative, the simple linear regression model of Lesson 2 is not appropriate.

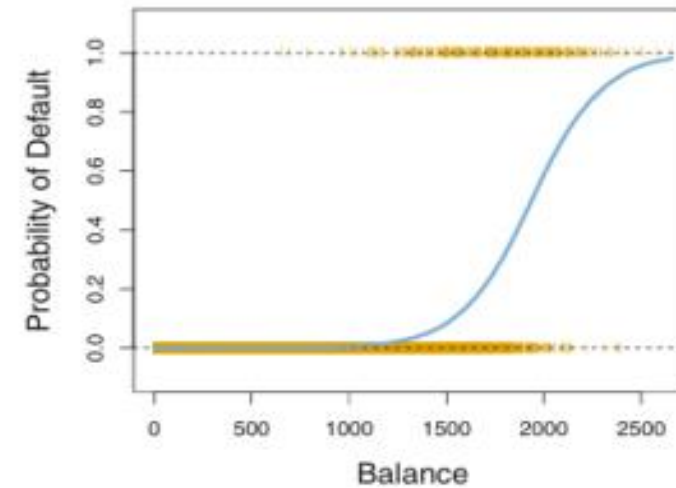
*The individuals who defaulted on their credit card payments are shown in orange*

# Logistic Regression

- Consider again the Default data set, where the response default falls into one of two categories, Yes or No. Rather than modeling this response Y directly, logistic regression models the probability that Y belongs to a particular category.



*Estimated probability of default using linear regression. Some estimated probabilities are negative! The orange ticks indicate the 0/1 values coded for default (No or Yes)*



*Predicted probabilities of default using logistic regression. All probabilities lie between 0 and 1.*

For the Default data, logistic regression models the probability of default. For example, the probability of default given balance can be written as  $\Pr(\text{default}=\text{Yes} \mid \text{balance})$

# The Logistic Model

□ How should we model the relationship between  $p(X)=\Pr(Y=1 \mid X)$  and  $X$ ?

If using a linear regression model to represent these probabilities:

$$p(X) = \beta_0 + \beta_1 X.$$

The outcome that we are expecting is either 1 or 0, and the issue with linear regression is that it can give values large than 1 or less than 0.

## The Logistic Model (cont.)

- To avoid this problem, we must model  $p(X)$  using a function that gives outputs between 0 and 1 for all values of  $X$ . Many functions meet this description. In logistic regression, we use the logistic function,

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}.$$

- Logistic regression can be explained better in odds ratio. The odds of an event occurring are defined as the probability of an event occurring divided by the probability of that event not occurring.

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0 + \beta_1 X}.$$

- By taking the logarithm of both sides of the above, we arrive at

$$\log \left( \frac{p(X)}{1 - p(X)} \right) = \beta_0 + \beta_1 X.$$

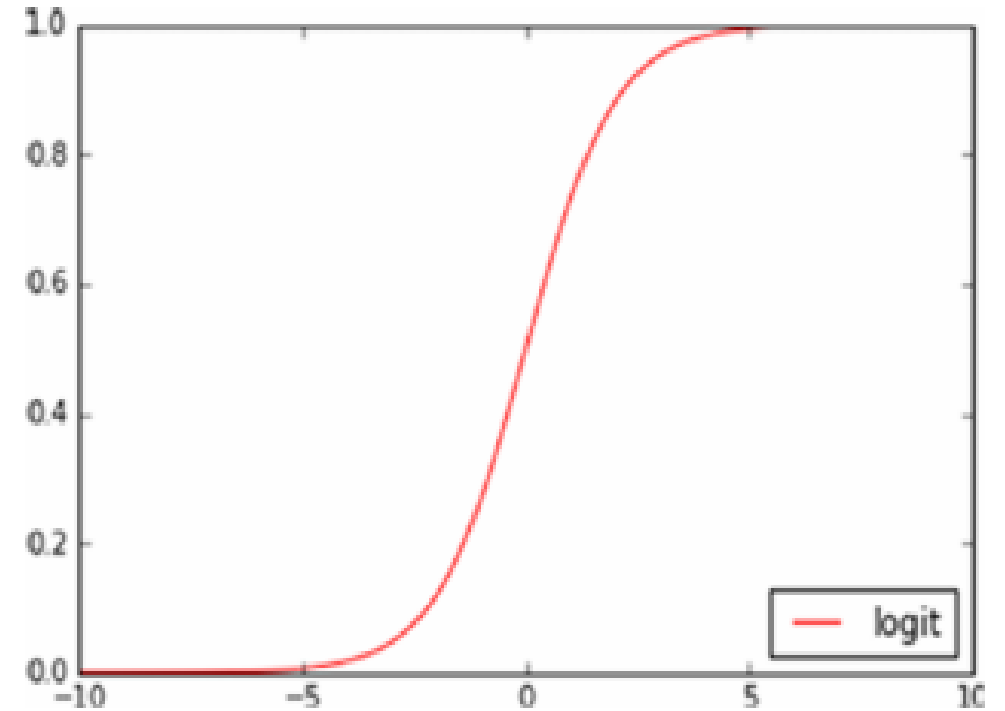
**Logit Function**

# Logistic regression equation probability

Logistic regression equation  $p(X) = \Pr(Y=1 | X) = 1 / (1 + e^{-(\beta_0 + \beta_1 x)})$ . Plot sigmoid function



```
Plot sigmoid function
x = np.linspace(-10, 10, 100)
y = 1.0 / (1.0 + np.exp(-x))
plt.plot(x, y, 'r-', label='logit')
plt.legend(loc='lower right')
# --- output ---
```



# Estimating the Regression Coefficients

- The coefficients  $\beta_0$  and  $\beta_1$  are unknown, and must be estimated based on the available training data.
- Although we could use (non-linear) least squares to fit the model, the more general method of **maximum likelihood** is preferred, since it has better statistical properties.
- Using a mathematical equation called a likelihood function:

$$\ell(\beta_0, \beta_1) = \prod_{i: y_i=1} p(x_i) \prod_{i': y_{i'}=0} (1 - p(x_{i'})).$$

- The estimates  $\hat{\beta}_0$  and  $\hat{\beta}_1$  are chosen to maximize this likelihood function.



# LR: For Example Making Predictions

- The table shows the coefficient estimates and related information that result from fitting a logistic regression model on the Default data in order to predict the probability of default=Yes using balance

	Coefficient	Std. error
<b>Intercept</b>	-10.6513	0.3612
<b>balance</b>	0.0055	0.0002

- Predict that the default probability for an individual with a balance of \$1,000

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X}} = \frac{e^{-10.6513 + 0.0055 \times 1,000}}{1 + e^{-10.6513 + 0.0055 \times 1,000}} = 0.00576,$$

# Model building in Scikit-learn

➤ Let's build the diabetes prediction model using a logistic regression classifier.  
Dataset:

<https://www.kaggle.com/uciml/pima-indians-diabetes-database>

**Loading data :**

```
#import pandas
import pandas as pd
col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree', 'age', 'label'] # load dataset
pima = pd.read_csv("pima-indians-diabetes.csv", header=None, names=col_names)
```

## Display the first few rows of a Pandas DataFrame called pima

```
pima.head()
```

	pregnant ∨	glucose ∨	bp ∨	skin ∨	insulin ∨	bmi ∨	pedigree ∨	age ∨	label ∨
0	1	85	66	29	0	26.6	0.351	31	0
1	8	183	64	0	0	23.3	0.672	32	1
2	1	89	66	23	94	28.1	0.167	21	0
3	0	137	40	35	168	43.1	2.288	33	1
4	5	116	74	0	0	25.6	0.201	30	0

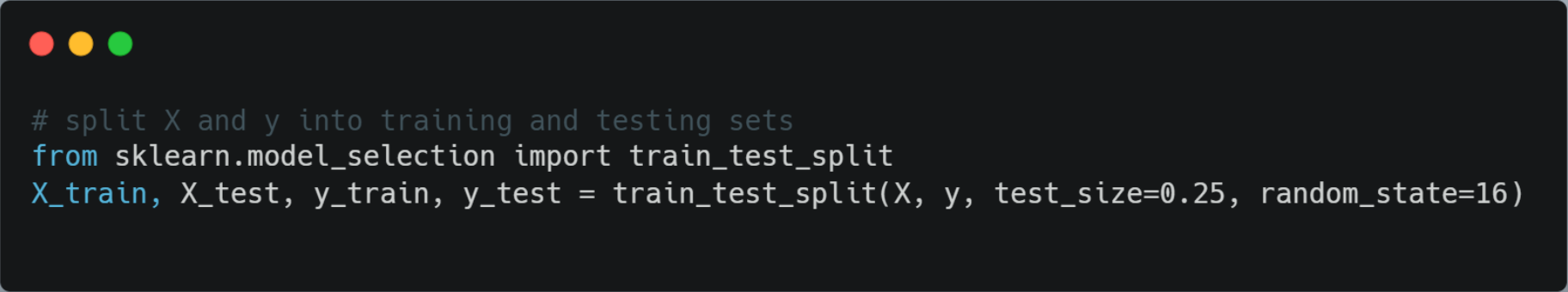
# Selecting features

- Here, need to divide the given columns into two types of variables dependent (or target variable) and independent variable (or feature variables).

```
#split dataset in features and target variable
feature_cols = ['pregnant', 'insulin', 'bmi', 'age', 'glucose', 'bp', 'pedigree']
X = pima[feature_cols] # Features
y = pima.label # Target variable
```

# Splitting data

- To understand model performance, dividing the dataset into a training set and a test set is a good strategy.
- Let's split the dataset by using the function `train_test_split()`. You need to pass 3 parameters: features, target, and test\_set size. Additionally, you can use `random_state` to select records randomly.



```
# split X and y into training and testing sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=16)
```

# Model development and prediction

- First, import the LogisticRegression module and create a logistic regression classifier object using the LogisticRegression() function with random\_state for reproducibility.
- Then, fit your model on the train set using fit() and perform prediction on the test set using predict().

```
# import the class
from sklearn.linear_model import LogisticRegression
# instantiate the model (using the default parameters)
logreg = LogisticRegression(random_state=16)
# fit the model with data
logreg.fit(X_train, y_train)
y_pred = logreg.predict(X_test)
```

# Multiple Logistic Regression

- We consider the problem of predicting a binary response using multiple predictors. By analogy with the extension from simple to multiple linear regression in Lesson 2

$$\log \left( \frac{p(X)}{1 - p(X)} \right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p,$$

where  $X = (X_1, \dots, X_p)$  are  $p$  predictors

$$p(X) = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}.$$

- Just as in above, we use the maximum likelihood method to estimate  $\beta_0, \beta_1, \dots, \beta_p$ .

# MLR: For Example Making Predictions

- For the Default data, estimated coefficients of the logistic regression model that predicts the probability of default using **balance**, **income**, and **student status**. Student status is encoded as a dummy variable **student[Yes]**, with a value of 1 for a student and a value of 0 for a non-student. In fitting this model, income was measured in thousands of dollars.

	Coefficient	Std. error
<b>Intercept</b>	-10.8690	0.4923
<b>balance</b>	0.0057	0.0002
<b>income</b>	0.0030	0.0082
<b>student [Yes]</b>	-0.6468	0.2362

For example, a student with a credit card balance of \$1,500 and an income of \$40,000

$$\hat{p}(X) = \frac{e^{-10.869 + 0.00574 \times 1,500 + 0.003 \times 40 - 0.6468 \times 1}}{1 + e^{-10.869 + 0.00574 \times 1,500 + 0.003 \times 40 - 0.6468 \times 1}} = 0.058.$$

Student:

$$\hat{p}(X) = \frac{e^{-10.869 + 0.00574 \times 1,500 + 0.003 \times 40 - 0.6468 \times 0}}{1 + e^{-10.869 + 0.00574 \times 1,500 + 0.003 \times 40 - 0.6468 \times 0}} = 0.105.$$

Non-student:



# Advantages

- Because of its efficient and straightforward nature, it doesn't require high computation power, is easy to implement, easily interpretable, and used widely by data analysts and scientists.
- Also, it doesn't require scaling of features. Logistic regression provides a probability score for observations.

# Disadvantages

- Logistic regression is not able to handle a large number of categorical features/variables. It is vulnerable to overfitting.
- Also, it can't solve the non-linear problems, which is why it requires a transformation of non-linear features.
- Logistic regression will not perform well with independent variables that are not correlated to the target variable and are very similar or correlated to each other.

# Homework

Suppose we collect data for a group of students in a statistics class with variables  $X_1$  = hours studied,  $X_2$  = undergrad GPA, and  $Y$  = receive an A. We fit a logistic regression and produce estimated coefficient,  $\hat{\beta}_0 = -6$ ,  $\hat{\beta}_1 = 0.05$ ,  $\hat{\beta}_2 = 1$ .

- (a) Estimate the probability that a student who studies for 40 h and has an undergrad GPA of 3.5 gets an A in the class.
- (b) How many hours would the student in part (a) need to study to have a 50 % chance of getting an A in the class?

End of Lesson

# My answer for this homework

- a. Estimate the probability that a student who studies for 40 hours and has an undergrad GPA of 3.5 gets an A in the class.

$$P(\text{receive} = A | X_1 = 40, X_2 = 3.5) = \frac{1}{1 + e^{-(-6 + 40 * 0.05 + 3.5)}} = 0.3775407$$

- b. How many hours would the student in part (a) need to study to have a 50% chance of getting an A in the class?

We can get

$$-6 + 0.05 \text{hours} + 3.5 = 0$$

$$\Rightarrow \text{hours} = 50$$