# 09 - Classification

Logistic Regression, Discriminant Analysis, and Naive Bayes

*SYS 4582/6018 | Spring 2019*

*09-classification.pdf*

## Contents

# 1 Classification Intro

## 1.1 Required R Packages

We will be using the R packages of:

- `glmnet` for ridge, lasso, and elasticnet regression
- `tidyverse` for data manipulation and visualization

```
library(glmnet)
library(tidyverse)
```

## 1.2 Credit Card Default data (`Default`)

The textbook An Introduction to Statistical Learning (ISL) has a description of a simulated credit card default dataset. The interest is on predicting whether an individual will default on their credit card payment.

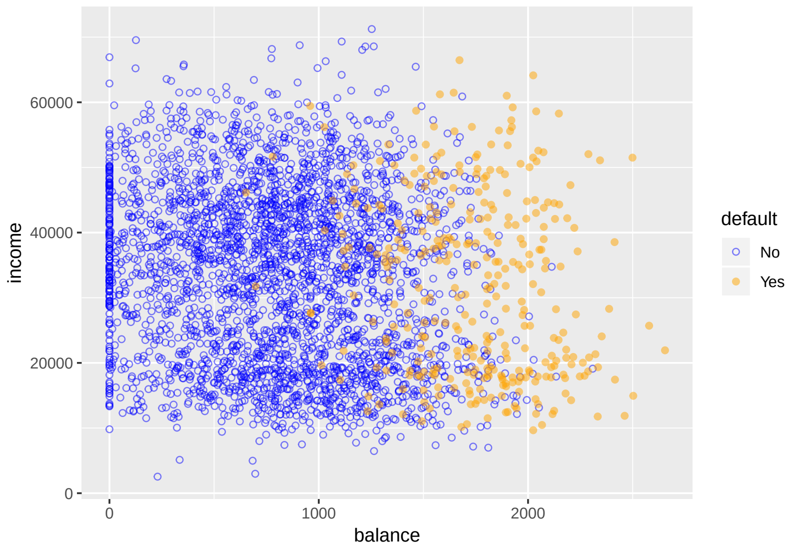
```
data(Default, package="ISLR")
```

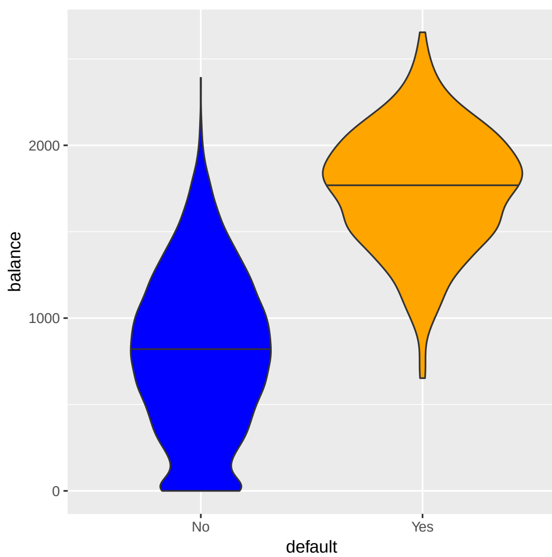
The variables are:

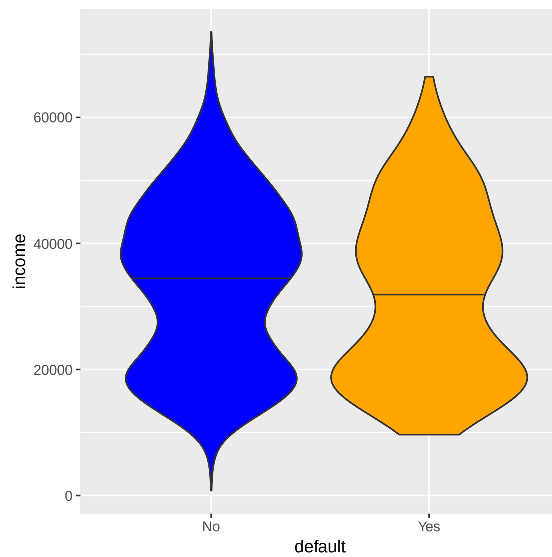
- *response variable* is categorical (factor) `Yes` and `No`, (`default`)
- the categorical (factor) variable (`student`) is either `Yes` or `No`
- the average balance a customer has after making their monthly payment (`balance`)
- the customer's income (`income`)

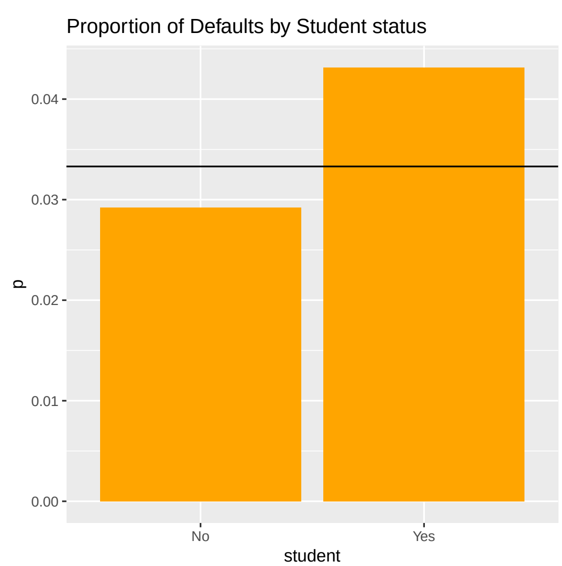
| default | student | balance | income |
|---|---|---|---|
| No | No | 729.5 | 44362 |
| No | Yes | 817.2 | 12106 |
| No | No | 1073.5 | 31767 |
| No | No | 529.3 | 35704 |
| No | No | 785.7 | 38463 |
| No | Yes | 919.6 | 7492 |

```
summary(Default)
#>  default     student        balance          income     
#>  No :9667   No :7056   Min.   :   0   Min.   :  772  
#>  Yes: 333   Yes:2944   1st Qu.: 482   1st Qu.:21340  
#>                        Median : 824   Median :34553  
#>                        Mean   : 835   Mean   :33517  
#>                        3rd Qu.:1166   3rd Qu.:43808  
#>                        Max.   :2654   Max.   :73554  
```


income
60000
40000
20000
0
0
1000
2000
balance
default
No
Yes


balance
2000
1000
0
No
Yes
default


income
60000
40000
20000
0
No
Yes
default


Proportion of Defaults by Student status
p
0.04
0.03
0.02
0.01
0.00
No
Yes
student

**Your Turn #1 : Credit Card Default Modeling**

How would you construct a model to predict defaults?

## 2 Classification and Pattern Recognition

- The response variable is categorical and denoted $G \in \mathcal{G}$
  - Default Credit Card Example: $\mathcal{G} = \{$"Yes", "No"$\}$
  - Medical Diagnosis Example: $\mathcal{G} = \{$"stroke", "heart attack", "drug overdose", "vertigo"$\}$
- The training data is $D = \{(X_1, G_1), (X_2, G_2), \ldots, (X_n, G_n)\}$
- Let $f_g(\mathbf{x}) = \Pr(\mathbf{X} = \mathbf{x} \mid G = g)$ be the *class conditional density* function

**Your Turn #2**

Think of two *models* (i.e., PDFs) to estimate $f_{\text{Yes}}(\text{balance} = 1500)$ in the credit default example.

- The optimal decision/classification is based on the posterior probability $\Pr(G = g \mid \mathbf{X} = \mathbf{x})$
  - We will discuss some optimal decisions in the binary case

**Your Turn #3**

Write out the equation for $\Pr(G = g \mid \mathbf{X} = \mathbf{x})$ using $f_g(\mathbf{x})$ and $\pi_g = \Pr(G = g)$, $g \in \mathcal{G}$.

- In real life, all of this is complicated because we have to estimate everything.
  - This is not easy in general, and is made more difficult in high dimensions (e.g., when $\mathbf{x}$ is a long vector).

## 2.1 Binary Classification

- Classification is simplified when there are only 2 classes.
- It is often convienient to *code* the response variable to a binary $\{0, 1\}$ variable:

$$Y_i = \begin{cases} 1 & G_i = \mathcal{G}_1 \quad \text{(outcome of interest)} \\ 0 & G_i = \mathcal{G}_2 \end{cases}$$

- In the `Default` data, it would be natural to set `default=Yes` to 1 and `default=No` to 0.
- Now we can use the more general descriptions:
  - $f_1(\mathbf{x}) = \Pr(X = x \mid Y = 1)$, $f_0(\mathbf{x}) = \Pr(X = x \mid Y = 0)$
  - $p(x) = \Pr(Y = 1 \mid \mathbf{X} = \mathbf{x})$
  - $\pi = \Pr(Y = 1)$

### 2.1.1 Linear Regression

- In this set-up we can run linear regression

$$\hat{y}(\mathbf{x}) = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_j$$

```r
#-- Create binary column (y)
Default = Default %>% mutate(y = ifelse(default == "Yes", 1L, 0L))

#-- Fit Linear Rergression Model
fit.lm = lm(y~student + balance + income, data=Default)
```

| term | estimate | std.error | statistic | p.value |
|---|---|---|---|---|
| (Intercept) | -0.0812 | 0.0084 | -9.685 | 0.0000 |
| studentYes | -0.0103 | 0.0057 | -1.824 | 0.0682 |
| balance | 0.0001 | 0.0000 | 37.412 | 0.0000 |
| income | 0.0000 | 0.0000 | 1.039 | 0.2990 |

**Your Turn #4 : OLS for Binary Responses**

1. For the binary $Y$, what is linear regression estimating?
2. What is the *loss function* that linear regression is using?
3. How could you create a *hard classification* from the linear model?
4. Does is make sense to use linear regression for binary classification?

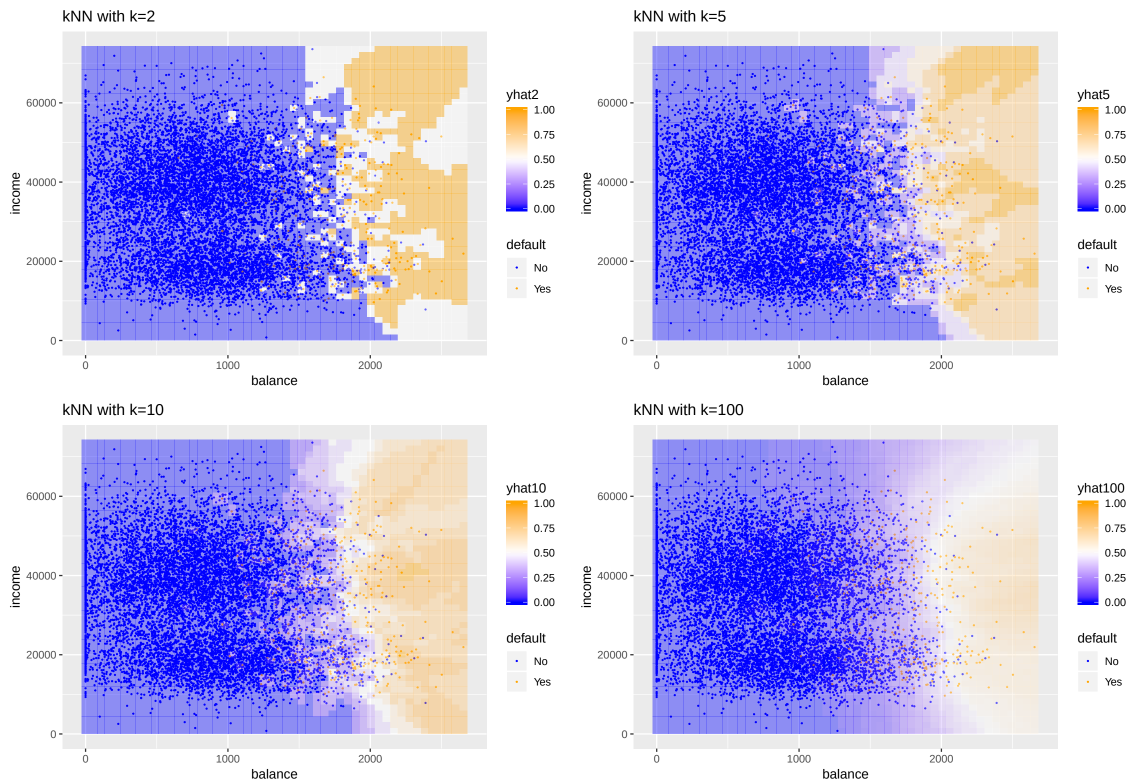
### 2.1.2 $k$-nearest neighbor (kNN)

- The $k$-NN method is a non-parametric *local* method, meaning that to make a prediction $\hat{y}|x$, it only uses the training data in the *vicinity* of $x$.
  - contrast with OLS linear regression, which uses all $X$'s to get prediction.
- The model is simple to describe

$$
\begin{aligned}
f_{\text{knn}}(x;k) &= \frac{1}{k} \sum_{i: x_i \in N_k(x)} y_i \\
&= \text{Avg}(y_i \mid x_i \in N_k(x))
\end{aligned}
$$

  - $N_k(x)$ are the set of $k$ nearest neighbors
  - only the $k$ closest $y$'s are used to generate a prediction
  - it is a *simple mean* of the $k$ nearest observations
- When $y$ is binary (i.e., $y \in \{0, 1\}$), the kNN model estimates

$$f_{\text{knn}}(x;k) \approx p(x) = \Pr(Y = 1|X = x)$$




**Your Turn #5 : Thoughts about kNN**

The above plots show a kNN model using the *continuous* predictors of `balance` and `income`.

- How can you use kNN with the categorical `student` predictor?

# 3 Logistic Regression

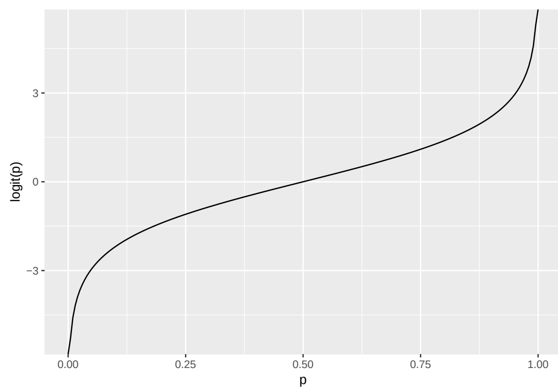
## 3.1 Basics

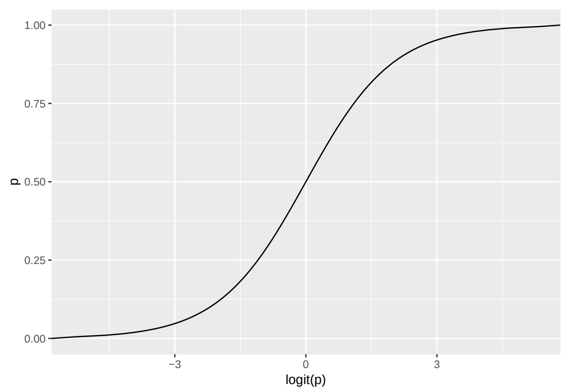
- Let $0 \leq p \leq 1$ be a probability.
- The log-odds of $p$ is called the *logit*

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right)$$

- The inverse logit is the *logistic function*. Let $f = \text{logit}(\text{p})$, then

$$\begin{aligned} p &= \frac{e^f}{1+e^f} \\ &= \frac{1}{1+e^{-f}} \end{aligned}$$



**Logistic Regression Details**

## 3.2 Estimation

### 3.2.1 Bernoulli Likelihood Function

**Bernoulli Likelihood**

#### 3.2.2 MLE for Logistic Regression

**MLE for Logistic Regression**

### 3.3 Logistic Regression in Action

- In R , logistic regression can be implemented with the `glm()` since it is a type of *Generalized Linear Model*.
- Because logistic regression is a special case of *Binomial* regression, use the `family=binomial()` argument

```
#-- Fit logistic regression model
fit.lr = glm(y~student + balance + income, data=Default,
             family="binomial")
```

| term | estimate | std.error | statistic | p.value |
|---|---|---|---|---|
| (Intercept) | -10.8690 | 0.4923 | -22.0801 | 0.0000 |
| studentYes | -0.6468 | 0.2363 | -2.7376 | 0.0062 |
| balance | 0.0057 | 0.0002 | 24.7376 | 0.0000 |
| income | 0.0000 | 0.0000 | 0.3698 | 0.7115 |

**Your Turn #6 : Interpreting Logistic Regression**

1. What is the estimated probability of default for a Student with a balance of \$1000?
2. What is the estimated probability of default for a *Non-Student* with a balance of \$1000?
3. Why does student=Yes appear to lower risk of default, when plot of student status

vs. default appears to increase risk?

# 4 Linear Discriminant Analysis (LDA)

# 5 Evaluation of Binary Classification Models

# 6 Naive Bayes