JOHNS HOPKINS CAREY BUSINESS SCHOOL

# Topic # 04: Data Analytics

## Logistic Regression, Classification

Instructor: Prof. Arnab Bisi, Ph.D.

Johns Hopkins Carey Business School

# Model Selection and Extensions

**Session 4:**

**Agenda**

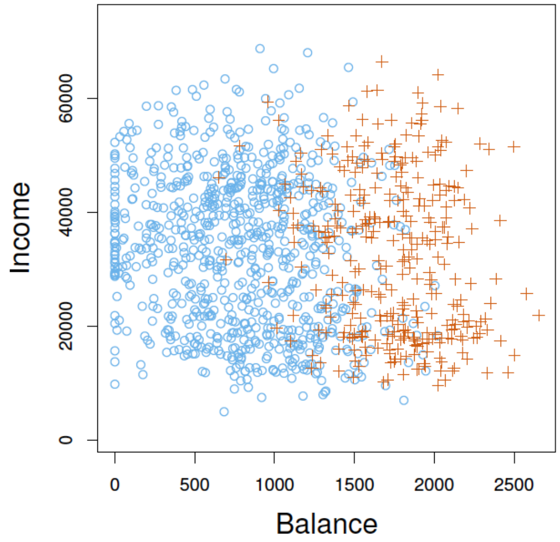
- Part 1: Logistic Regression
- Break
- Part 2: Working with code

# Regression is not Universal

- Response variables may not be quantitative
- In many instances the response is qualitative
  - Origin: US, China, Japan, etc.
  - Brand name: Chevrolet, BMW, etc.
  - Course grades: A, A-, B+, B, B-…
- Special case – qualitative and binary ($y$ = 0 or 1)
  - Head or tail (flip of a coin)
  - Profit or loss
  - Being able to pay back a loan or not
  - Buy or not Buy
  - Sick or Healthy
  - Republican or Democrat

# Example: Credit Card Default

- Default data, annual incomes and average monthly credit card balances

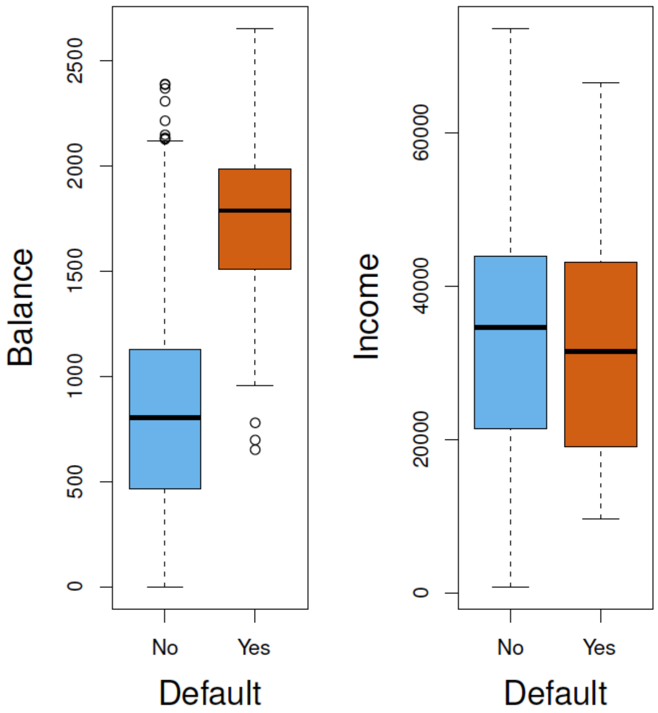



- The default data set contains a number of individuals
- Those who defaulted are shown in orange
- Those who did not are shown in blue

JOHNS HOPKINS
CAREY BUSINESS SCHOOL

# Example: Credit Card Default

- Boxplot of credit card balance as a function of default
- Boxplot of annual income as a function of default
- Those who defaulted are shown in orange
- Those who did not are shown in blue



# What Is Linear Regression?

- Regression modeling is one of the most useful statistical techniques

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + E$$

- $E$ follows a normal distribution with mean 0 and variance $\sigma^2$
- In expectation, it can be expressed as,

$$Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$$

JOHNS HOPKINS
CAREY BUSINESS SCHOOL

# Why Not Linear Regression?

- For a Binary response, the conditional mean in the regression model becomes

$$E[Y \mid X] = 1 * \Pr(y = 1 \mid X) + 0 * \Pr(y = 0 \mid X)$$
$$= \Pr(y = 1 \mid X)$$

- The expectation is now a probability between 0 and 1 and we cannot use just any linear regression function
- In logistic "regression" we use

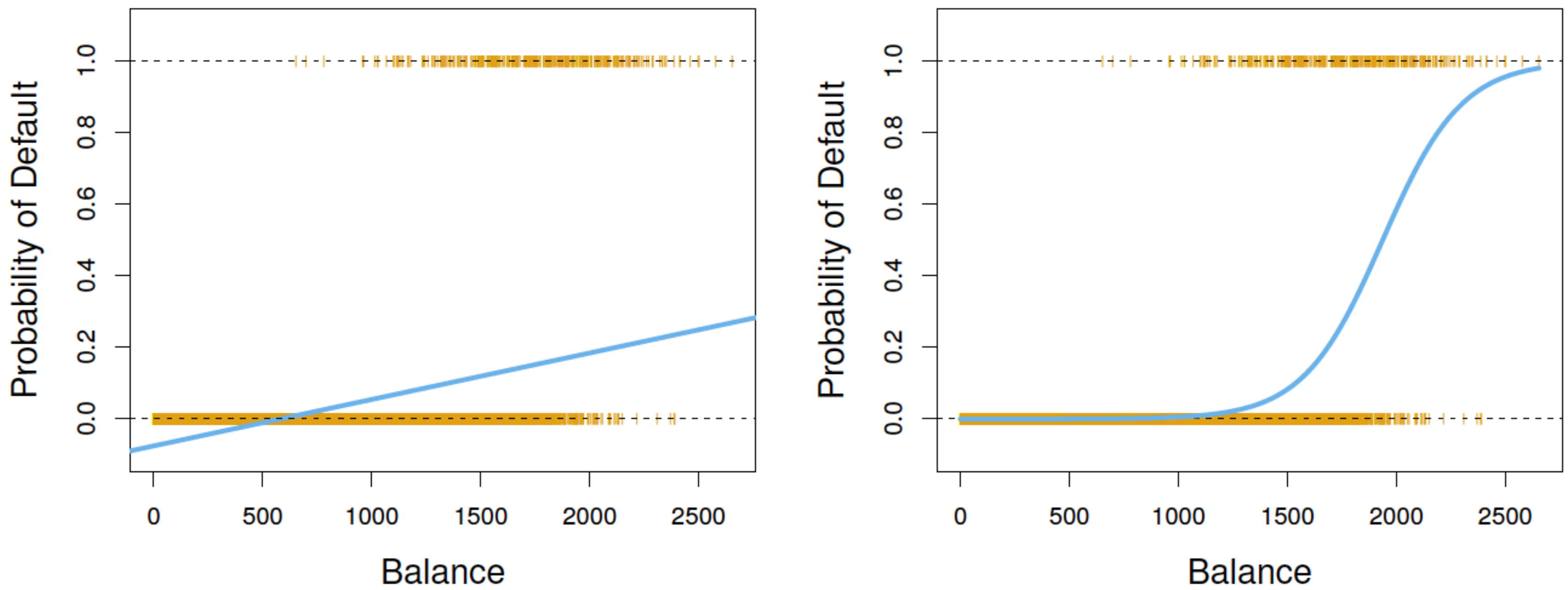
$$q = \Pr(y = 1|\mathbf{X}) = \frac{\exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p)}{1 + \exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p)}$$

  - Note that $q$ is always between 0 and 1

JOHNS HOPKINS
CAREY BUSINESS SCHOOL

# Comparison between Linear and Logistic

Figure: Classification using the `Default` data




- Estimated probability of default using linear regression
  - Some estimated probabilities are negative
  - Some can be > 1
- Estimated probability of default using logistic regression
  - All probabilities are between 0 and 1

# Odds of Success

- The Logistic model is to assume that:

$$\log \frac{q}{1-q} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = \hat{y}$$

- The quantity $q/(1\text{-}q)$ relates the probability of success to the probability of failure
- We refer to $q/(1\text{-}q)$ as the *odds of success*, which takes on any value between 0 and infinity
- The quantity $\log[q/(1\text{-}q)]$ is referred to as the *logit* or *log-odds*
- Thus we are using a regression as a linear model for *log-odds*

# Interpretation of Regression Coefficients

- The logit of $q$ is defined by

$$\log \frac{q}{1-q} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$$

- All other variables remaining fixed, a change of one unit in $x_i$ changes the log of the odds of success by $\beta_i$ units
- The odds of success is changed by the multiplicative factor $\exp(\beta_i)$
- For $b_i = 0$, $\exp(b_i) = 1$ which implies that a change in the explanatory variable has no effect on the odds
- For $b_i = 1$, $\exp(b_i)$ » $2.72$ which implies that a unit change in $x_i$ changes the odds by the multiplicative factor 2.72, or an increase by 172%
- For $b_i = 2$, $\exp(b_i) = 7.39$ which implies that a unit change in $x_i$ changes the odds by the multiplicative factor 7.39, or an increase by 639%

# In Logistic Regression

- There is no error term
- Model of the probability of an event using a linear model for the logit
  - Parameters can be estimated by the method of maximum likelihood estimation (MLE)
  - In other words we maximize a likelihood function which is the probability that the entire observed data set results from a given set of parameters
- Given *n* observations containing covariates $x_{i1}, x_{i2}, \ldots, x_{ip}$ and success indicator $y_i = 0/1$ the likelihood function is,

$$\begin{aligned}
\prod_{i=1}^{n} q(y_i|x_i) &= \prod_{i=1}^{n} (q_i)^{y_i} (1-q_i)^{1-y_i} \\
&= \prod_{i=1}^{n} \left[ \frac{\exp(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ip})}{1 + \exp(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ip})} \right]^{y_i} \\
&\quad \cdot \left[ \frac{1}{1 + \exp(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})} \right]^{1-y_i}
\end{aligned}$$

JOHNS HOPKINS
CAREY BUSINESS SCHOOL

# Maximum Likelihood Estimation

- Maximizing the likelihood function is equivalent to minimizing the deviance
  - In this setting this is the negative logarithm of the likelihood

$$D = -\left[\sum_{i=1}^{n} y_i \log(q_i) + \sum_{i=1}^{n} (1 - y_i) \log(1 - q_i)\right],$$

where $q_i$ is defined as follows:

$$q_i = \frac{\exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p)}{(1 + \exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p))}.$$

JOHNS HOPKINS
CAREY BUSINESS SCHOOL

# Statistical Inference

- Parameters can be estimated through maximum likelihood estimation
  - Likelihood is a fitted probability of your data
  - You want this to be as large as possible
  - Deviance refers to the distance between data and fit
  - You always want Deviance to be as small as possible
- Many statistical packages (such as **R**) include routines for the estimation of logistic regression models
- Output typically includes estimates as well as standard errors of those estimates

JOHNS HOPKINS
CAREY BUSINESS SCHOOL

# Logistic Regression in R

- The glm() command fits generalized linear models
  - This class includes linear models and,
  - This class includes logistic regressions
    - glm(formula, family = binomial, data)
  - Also includes probit models
- summary() function shows results

# Example: The Stock Market Data

- The Smarket data set is in the ISLR library
- Smarket consists of daily percentage returns for the S&P 500 stock index from 2001 to 2005
- For each trading day it records the percentage returns for each of the previous 5 trading days as Lag1 – Lag5
- It also includes Volume (traded on previous day in billions) Today (percentage return), and Direction (Up or Down)
- Objective: predict direction of market tomorrow based on performance over the 5 days (including today) leading up to it

# Model Assessment: Estimation Accuracy

- Train observations $\{(x_1, y_1), \ldots, (x_n, y_n)\}$
- Estimation accuracy: *error rate* (training data)

$$\frac{1}{n}\sum_{i=1}^{n} I\left(y_i \neq \hat{y}_i\right), \quad where$$

$$I\left(y_i \neq \hat{y}_i\right) = \begin{cases} 1, & y_i \neq \hat{y}_i \\ 0, & otherwise \end{cases}$$

- Primary focus is the *error rate* (test data)
  - Use model fit to *training* data to make predictions about the *test* data
  - A better model is one with a smaller error rate when applied to the *test* data

# Consider Stock Market Data

- Fit a logistic regression in order to predict Direction using Lag1 – Lag5 and Volume
  - > glm.fit=glm(Direction ~ Lag1 + Lag2 + Lag3+ Lag4 + + Lag5 + Volume, family = binomial, data=Smarket)
- Use the model to make predictions
  - > glm.probs = predict(glm.fit, type = “response”)
- Convert these probabilities into predictions of direction
  - > glm.pred = rep(“Down”, 1250)
  - > glm.pred[glm.probs > 0.5] = “Up”
- Compare the predictions to the actual outcomes
  - > table(glm.pred, Direction)
  - > mean(glm.pred == Direction)

JOHNS HOPKINS
CAREY BUSINESS SCHOOL

# Consider Output from Logistic Regression

```
Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = binomial, data = Smarket, subset = train)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
 -1.30   -1.19    1.08    1.16    1.35

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.19121    0.33369    0.57     0.57
Lag1        -0.05418    0.05179   -1.05     0.30
Lag2        -0.04581    0.05180   -0.88     0.38
Lag3         0.00720    0.05164    0.14     0.89
Lag4         0.00644    0.05171    0.12     0.90
Lag5        -0.00422    0.05114   -0.08     0.93
Volume      -0.11626    0.23962   -0.49     0.63

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1383.3  on 997  degrees of freedom
Residual deviance: 1381.1  on 991  degrees of freedom
AIC: 1395

Number of Fisher Scoring iterations: 3
```

# Understanding the Regression Summary

- Null Deviance: deviance of a model that contains only the intercept
- Residual Deviance: deviance of the fitted model
- Is the model we made better than the null model?
  - Is the reduction in deviance significant?

```
    Null deviance: 1383.3  on 997  degrees of freedom
Residual deviance: 1381.1  on 991  degrees of freedom
```

  - < 1-pchisq(Null Dev. – Res. Dev, Null DOF – Res. DOF)
  - < 1 – pchisq(1383.3 – 1381.3, 997-991) = 0.9
  - This is the probability of seeing this reduction purely by chance: i.e. the inclusion of the variables added no new information
- For more detailed discussion see:
- https://www.youtube.com/watch?v=xl5dZo_BSJk

JOHNS HOPKINS
CAREY BUSINESS SCHOOL

# Consider Training Data and Test Data

- Divide data into two sets: train and test
- > train=(Year < 2005)
- > Smarket.2005 = Smarket[!train,]
- > Direction.2005 = Direction[!train]
- Regression using train data, prediction using test data
  - > glm.fit=glm(Direction ~ Lag1 + Lag2 + Lag3+ Lag4 + Lag5 + Volume, family = binomial, data=Smarket, subset=train)
  - > glm.probs = predict(glm.fit, Smarket.2005, type = "response")
- Convert these probabilities into predictions of direction
  - > glm.pred = rep("Down", 252)
  - > glm.pred[glm.probs > 0.5] = "Up"
- Compare the predictions to the actual outcomes
  - > table(glm.pred, Direction.2005)
  - > mean(glm.pred == Direction.2005)

JOHNS HOPKINS
CAREY BUSINESS SCHOOL

# Exercise

- Use Smarket data set
- Create logistic regression model using only Lag1 and Lag2
  - Fit model to Smarket dataset
  - Make predictions
  - Report test error rate

- 15 mins

# Non-Parametric Methods

- Parametric methods estimate the value of specific "parameters"
- Many advantages
  - Easy to fit
  - Estimate a small number of values
  - Simple interpretation
- Some disadvantages
  - Strong assumptions are made about the world
  - True relationship may be far from linear or logistic
  - Poor data fit, wrong conclusion

- Non-parametric methods
  - Do not explicitly assume a parametric model
  - Provide more flexible approaches

# K-Nearest Neighbors (KNN)

- Given a positive integer K and a test observation $x_0$
- KNN first identifies the K points in the training data that are closest to $x_0$
  - Call this set, $Set_0$
  - Estimate the conditional probability by

$$\Pr\left(Y = j \mid X = x_0\right) = \frac{1}{K} \sum_{i \in Set_0} I\left(y_i = j\right),$$

  - Where $I()$ is an indicator function
  - The estimated value of $f(x_0)$ is

$$\hat{f}\left(x_0\right) = \frac{1}{K} \sum_{x_i \in Set_0} y_i$$

  - KNN applies Bayes rule and classifies $x_0$ in the class with the largest probability

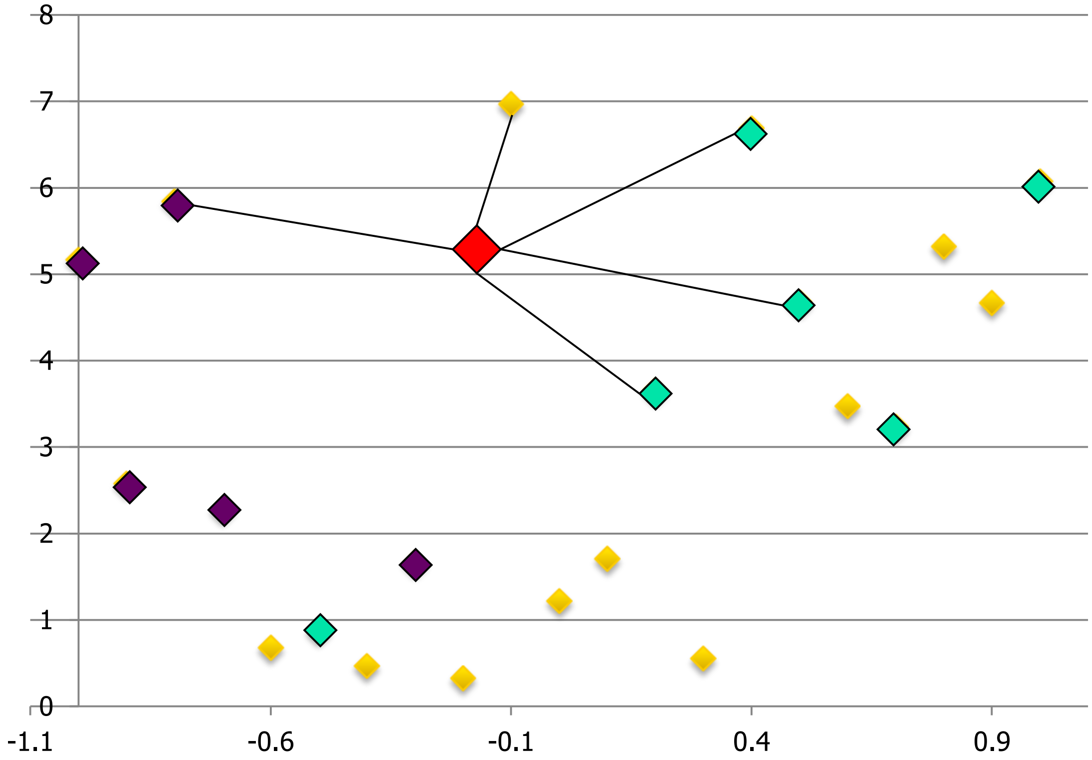
JOHNS HOPKINS
CAREY BUSINESS SCHOOL

# Steps in KNN

1. Identify class for each point in a set
2. Find distance between these points and new point $x_0$

$$D_{1,2} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

3. Identify KNN
4. Count number of neighbors in each class
5. Assign new point to class containing highest number of neighbors



# KNN Example

- We use the knn() function which is part of the class library
  - First we fit a model using the training data
    - This identifies the class of each of your neighbors
  - Then we use the model to make predictions
    - Assign new points to a class
- KNN requires four inputs
  - Data Frame containing the predictors associated with the training data
    - Coordinates of points already assigned to a class
  - A Data Frame containing the predictors associated with the test data for which we wish to make predictions
    - Coordinates of points yet to be assigned to a class
  - A vector (list) containing the class labels for the training observations
    - Which class each assigned point is already in
  - A value for K, which is the number of nearest neighbors to be used by the classifier

JOHNS HOPKINS
CAREY BUSINESS SCHOOL

# KNN Example: Smarket Data

- Sample code
  - ```
    > require(ISLR)
    ```
  - ```
    > Smarket = Smarket
    ```
  - ```
    >  require(class)
    ```
  - ```
    > knn1.fit=knn(train.data[,c(2:7)], test.data[,c(2:7)], train$Direction, 1)
    ```
  - ```
    > head(knn1.fit)
    ```
  - ```
    > mean(knn1.fit == test.data$Direction)
    ```

# KNN Example: Smarket Data

- Sample code
  - > train=(Year < 2005)
  - > Smarket.2005 = Smarket[!train,]
  - > library(class)
  - > train.X = cbind(Lag1, Lag2)[train,]
  - > test.X = cbind(Lag1, Lat2)[!train,]
  - > train.Direction=Direction[train]
  - > Direction.2005=Direction[!train]
  - > knn.pred=knn(train.X, test.X, train.Direction, k=1)
  - > table(knn.pred, Direction.2005)
  - > mean(knn.pred == Direction.2005)

- When k = 1 the prediction is correct 50% of the time

JOHNS HOPKINS
CAREY BUSINESS SCHOOL

# Exercise

- Use Smarket data set
- Perform KNN on the training data with several values of K (1, 3, 5, 7, 9, etc)
  - Plot the error rate as a function of K
  - Which value of K works best?
- 15 mins

# Summary of Classification Models

- Logistic regression
  - Best when output is one of a small (2 is small) number of possibilities
  - Using an added function to transform a continuous output into one constrained to be between 0 and 1
- KNN method
  - Assumes that if a point is similar to its neighbors according to X it will also be similar according to Y
  - Allows neighbors to “vote” on admission to the group

JOHNS HOPKINS
CAREY BUSINESS SCHOOL

Questions, Comments?

Let's move to the Code.

JOHNS HOPKINS
CAREY BUSINESS SCHOOL