Linear Statistical Modeling Methods with SAS

Classification - Part I

Xuemao Zhang
East Stroudsburg University

April 10, 2024

Outline

- Introduction
- K-Nearest Neighbors
- Logistic Regression

Introduction

- Classification involves predicting a **categorical/qualitative response**:
  - Cancer versus Normal
  - Tumor Type 1 versus Tumor Type 2 versus Tumor Type 3
- Classification problems tend to occur even more frequently than regression problems in biomedical applications.
- Categorical/qualitative variables take values in an unordered set: e.g.
  - eye color $\in$ {brown, blue, green}
  - email $\in$ {spam; not spam}.
- We want to build a function that $C(X)$ takes as input the feature vector $X = (X_1, \ldots, X_p)$ and predicts the value for Y, i.e. $C(X)$ is in which category.
- Often we are more interested in estimating the probability that X belongs to a given category.
  - For example: we might want to know the probability that someone will develop diabetes, rather than to predict whether or not they will develop diabetes.

Introduction

- We used training data to conduct classifications.
- What we really care about is how well the method works on new data. We call this new data "**Test Data**".
- Let n be the total number observations. For example, the data $(x1, y1), \ldots, (x_n, y_n)$. And let $\hat{y}$ be our estimate. Then the training **error rate**, the proportion of mistakes that are made

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i).$$

- We are most interested in the error rates that result from applying our classifier to test observations that were not used in training. The **test error rate** associated with a set of test observations of the form test error (x_0, y_0) is given by

$$Ave(I(y_0 \neq \hat{y}_0)).$$

K-Nearest Neighbors

- Can we take a totally non-parametric (model-free) approach to classification?
- K-nearest neighbors (KNN):
  - 1. For any given X_0, identify the k observations whose X values are *closest to the observation X_0 at which we want to make a prediction.*
  - 2. Classify the observation of interest X_0 to the most frequent class label of those K nearest neighbors: If the majority of the Y's are orange we predict orange otherwise guess blue.
- The smaller that k is the more flexible the method will be.

K-Nearest Neighbors

- Example: K-nearest neighbors in two dimensions

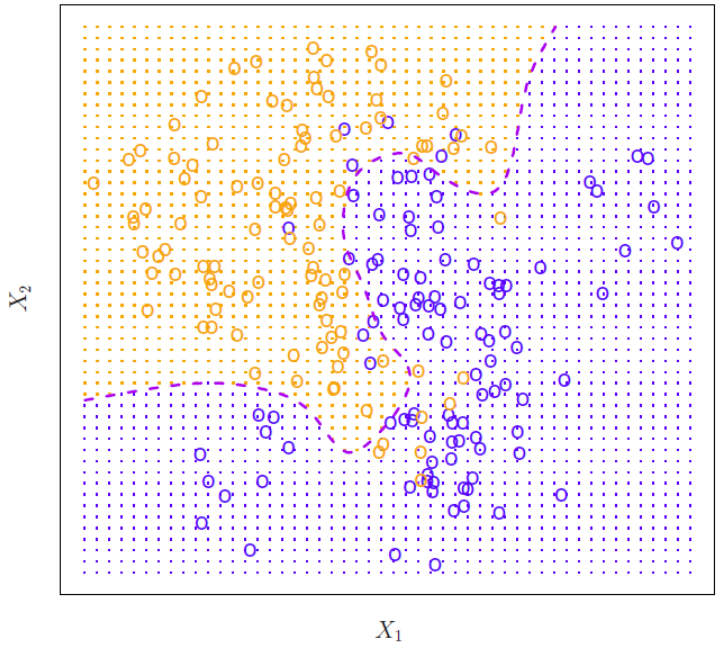
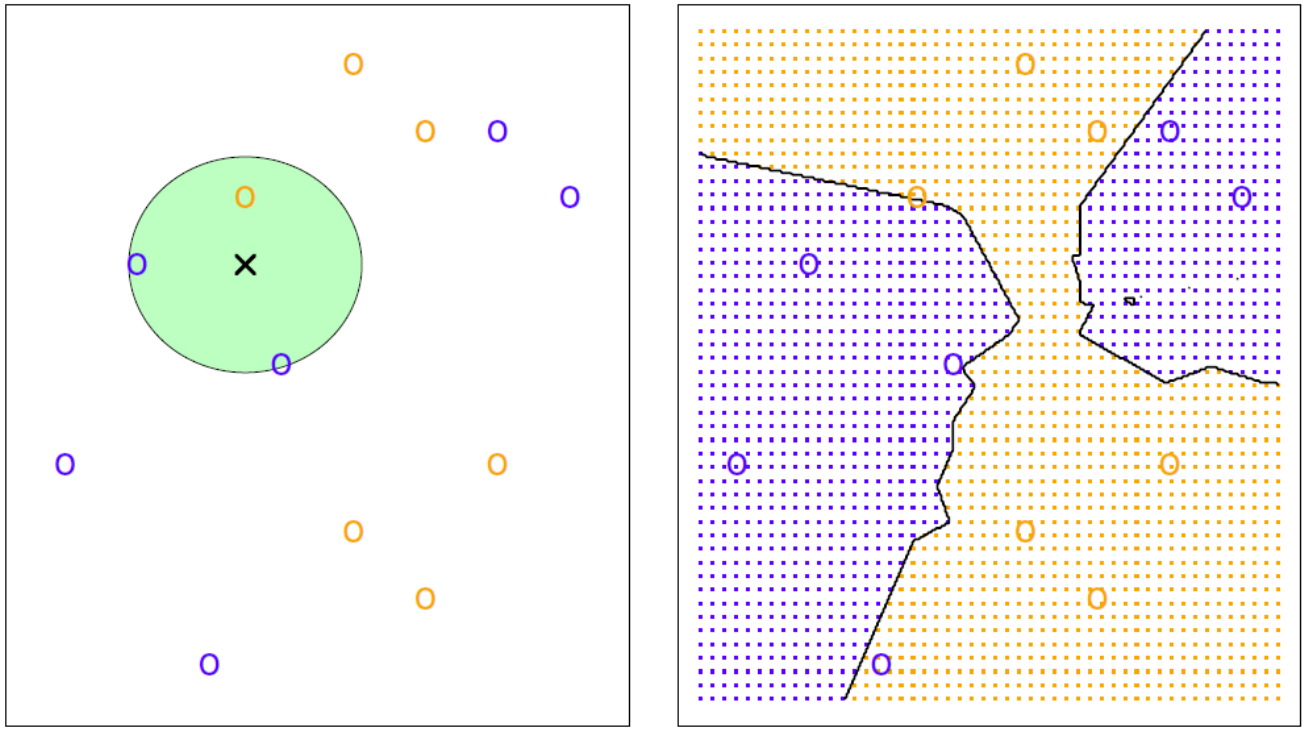


Figure 1: A simulated data set consisting of 100 observations in each of two groups, indicated in blue and in orange. The purple dashed line represents the Bayes decision boundary. The orange background grid indicates the region in which a test observation will be assigned to the orange class, and the blue background grid indicates the region in which a test observation will be assigned to the blue class.

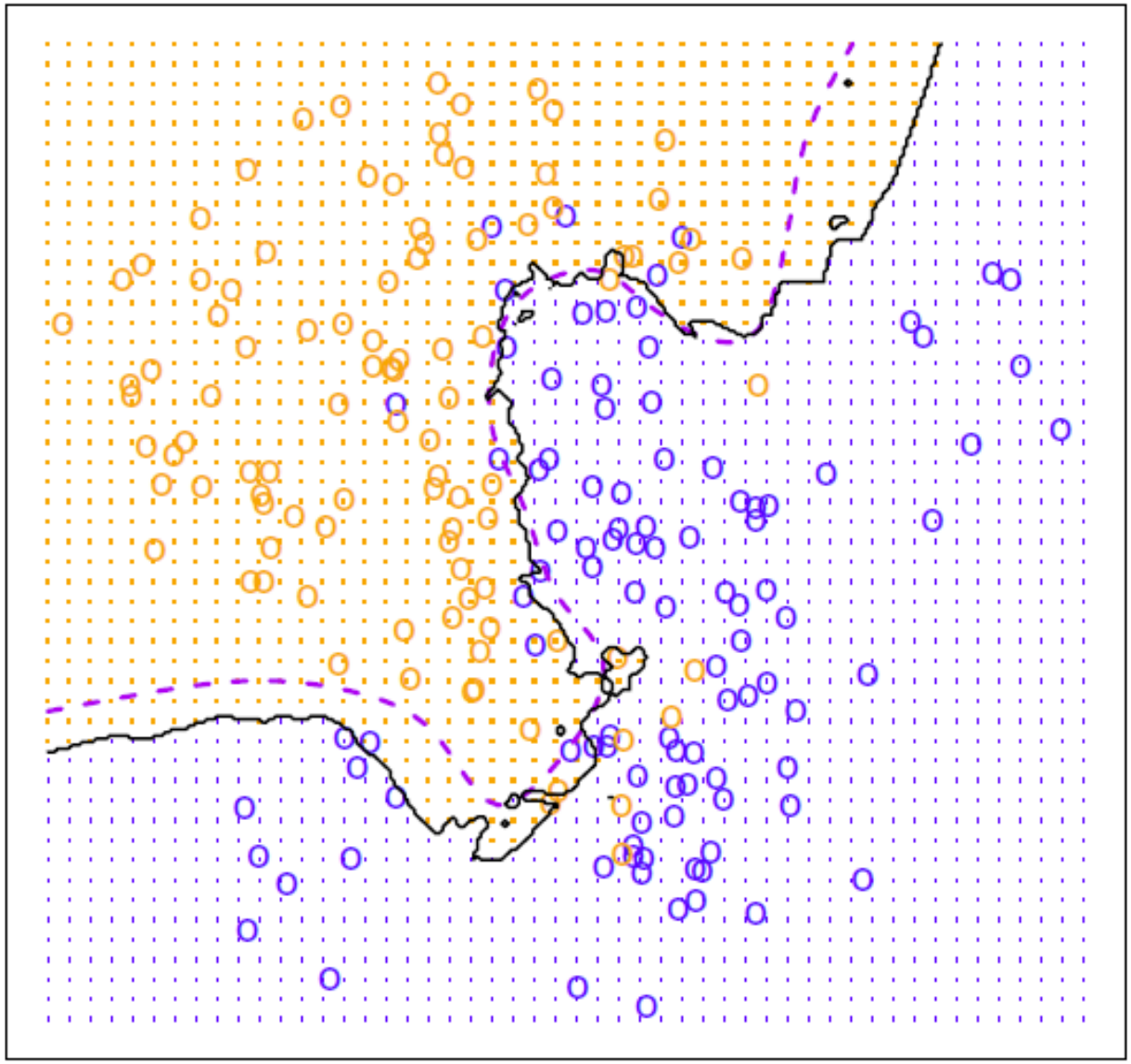
K-Nearest Neighbors

- Example: KNN Example with $k = 3$



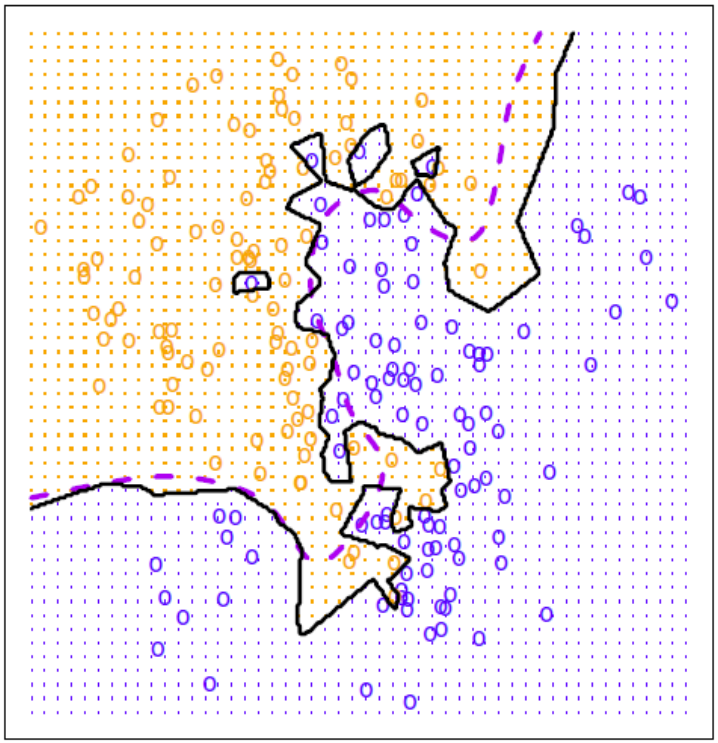
K-Nearest Neighbors

- Example: KNN Example with $k = 10$

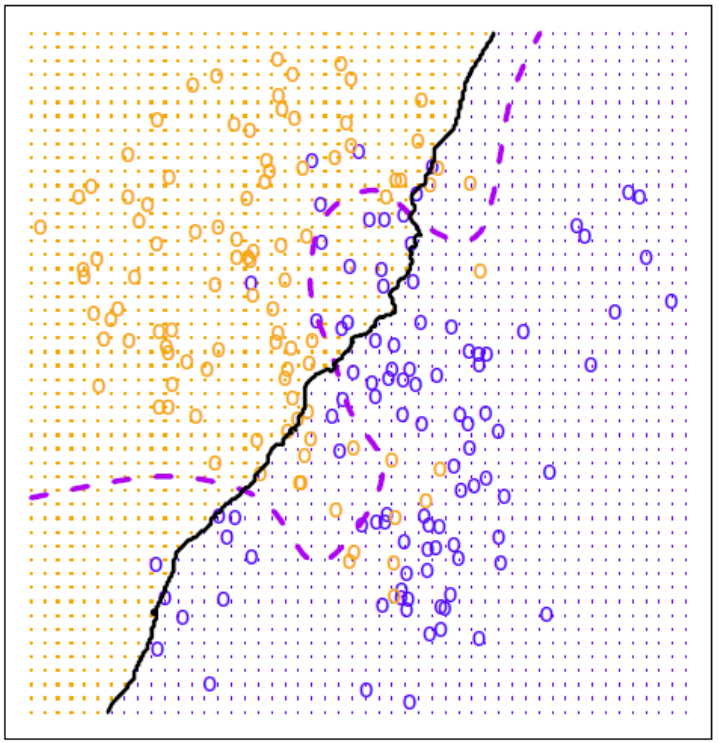


K-Nearest Neighbors

KNN: K=1



KNN: K=100



K-Nearest Neighbors

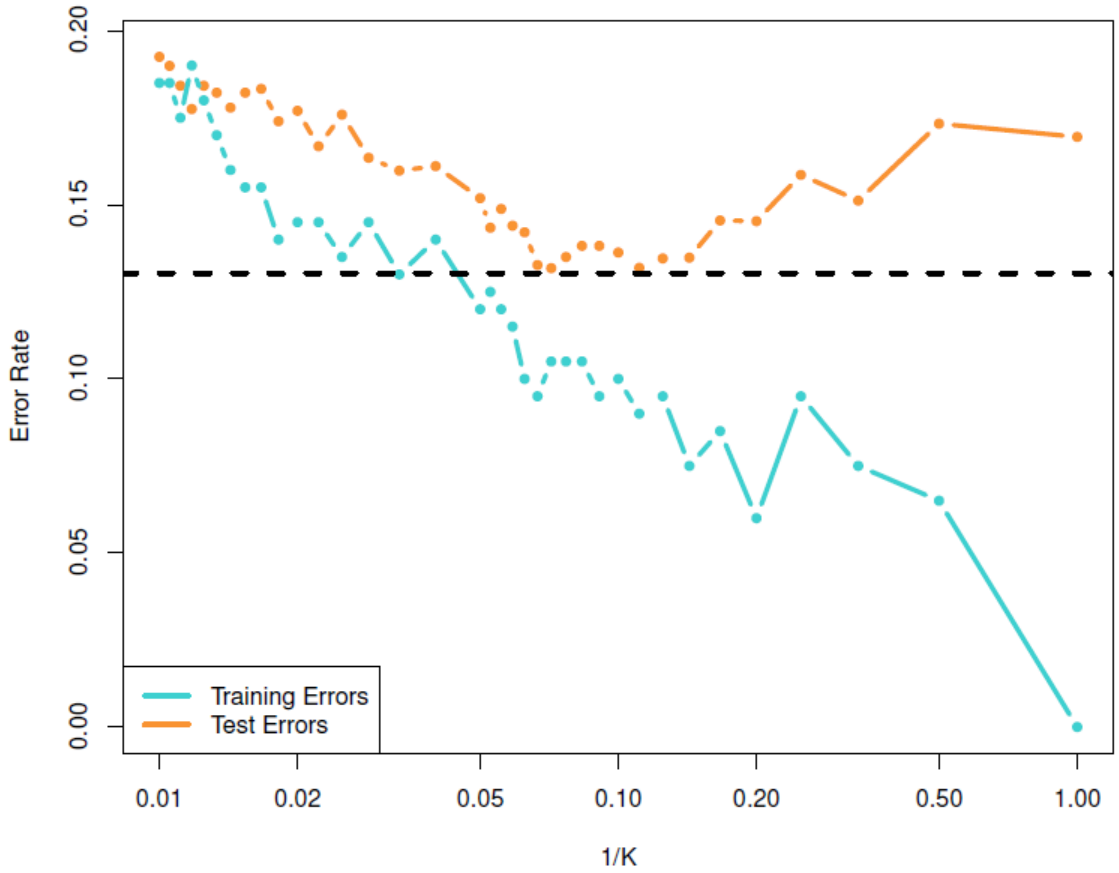
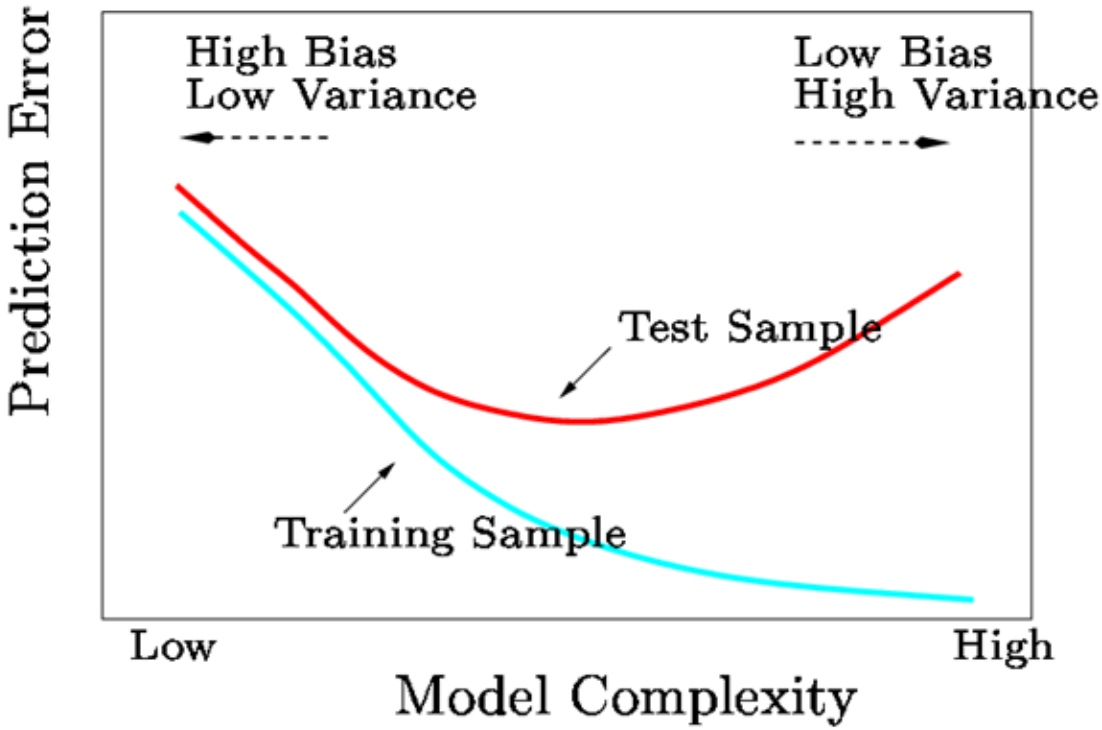


Figure 2: The KNN training error rate (blue, 200 observations) on the data from Figure 1 and test error rate (orange, 5,000 observations), as the level of flexibility (assessed using $1/K$) increases, or equivalently as the number of neighbors K decreases. The black dashed line indicates the Bayes error rate. The jumpiness of the curves is due to the small size of the training data set.

A Fundamental Picture

- In general training errors will always decline.
- However, test errors will decline at first (as reductions in bias dominate) but will then start to increase again (as increases in variance dominate).
- We must always keep the picture mentioned already in mind when choosing a learning method. More flexible/complicated is not always better!



K-Nearest Neighbors

- Simple, intuitive, model-free.
- Good option when p is very small.
- Curse of dimensionality: when p is large, no neighbours are "near". All observations are close to the boundary.

K-Nearest Neighbors

- Consider the Stock Market Data

```
PROC IMPORT
DATAFILE='/home/u5235839/my_shared_file_links/u5235839/Smarket.csv'
    DBMS=CSV
    OUT=Smarket;
    GETNAMES=YES;
RUN;

proc contents data=Smarket;
run;

proc freq data=Smarket;
tables Direction;
run;
```

K-Nearest Neighbors

```
data Smarket1;
set Smarket;
/* Create binary indicators */
Up = (Direction = 'Up');
drop Direction;
run;
```

K-Nearest Neighbors

- `Lag1` through `Lag5`: the percentage returns for each of the five previous trading days
- `Volume`: the number of shares traded on the previous day, in billions
- `Today`: the percentage return on the date in question
- `Direction`: whether the market was Up or Down on the date `Today`

- The goal is to predict whether the index will increase or decrease on a given day using the past 5 days' percentage changes in the index.

K-Nearest Neighbors

- We split the data as training data and test data in the following way because the data is time-series data
  - Suppose the test data are future data

```
data train test;
    set Smarket1;
    if Year < 2005 then output train;
    else output test;
run;

proc print data=train;
run;

proc print data=test;
run;
```

K-Nearest Neighbors

- KNN Classification can be conducted using the DISCRIM procedure
  - `METHOD=NPAR k=1`: kNN Classification method using 1 nearest neighbors.
  - `TESTLIST`: Displays the classification results of `TEST=`
  - `TESTLISTERR`: Displays the misclassified observations of TEST=

```
proc discrim data=train testdata=test METHOD=NPAR k=1
testout=tout TESTLIST TESTLISTERR;
   class Up;
   var Lag1 Lag2;
   run;
```

K-Nearest Neighbors

- Let's repeat the analysis using $k = 3$.

```
proc discrim data=train testdata=test METHOD=NPAR k=3
testout=tout2 TESTLIST TESTLISTERR;
   class Up;
   var Lag1 Lag2;
   run;
```

- The results have improved slightly. But increasing k further turns out to provide no further improvements

K-Nearest Neighbors

- kNN algorithms have two basic tuning parameters: the number of nearest neighbors k and the number of predictors p.
- Selection of k and p can be based on a cross-validation approach. We skip this discussion.

Logistic Regression

- Let $\mathcal{C}$ be the set of collection of responses of Y. For example, email is one of $\mathcal{C}$ = (spam, not spam), digit class is one of $\mathcal{C}$= $\{0, 1, \ldots, 9\}$.
- Is there an ideal $C(X)$? Suppose the K elements in $\mathcal{C}$ are numbered $1, 2, \ldots, K$. For any x, let

$$p_k(x) = P(Y = k|X = x), k = 1, \ldots, K.$$

  These are the **conditional class probabilities** at $X = x$.
- Then the **Bayes classifier** at x is

$$C(x) = j, \text{ if } p_j(x) = \max\{p_1(x), \ldots, p_K(x)\}$$

  For responses with two categories, we just check which probability is greater than 50%.

Logistic Regression

- The Bayes classifier produces the lowest possible test error rate, called the Bayes error rate.
- The error rate of the Bayes classifier at $X = x_0$ will be

$$1 - \max_j Pr(Y = j|X = x_0).$$

- In general, the overall Bayes error rate is given by

$$1 - E\left(\max_j Pr(Y = j|X)\right)$$

- We can build parametric models for representing the conditional class probabilities $p_k(x)$ to construct a Bayes classifier.
- Logistic regression is such a method.

Logistic Regression

- Example: Credit Card Default

```
PROC IMPORT
DATAFILE='/home/u5235839/my_shared_file_links/u5235839/Default.csv'
DBMS=CSV
OUT=Default;
GETNAMES=YES;
RUN;

proc contents data=Default;
run;

proc freq data=Default;
tables default student;
run;

data Default1;
set Default;
/* Create binary indicators */
student_yes = (student = 'Yes');
default_yes = (default = 'Ye');
run;
```

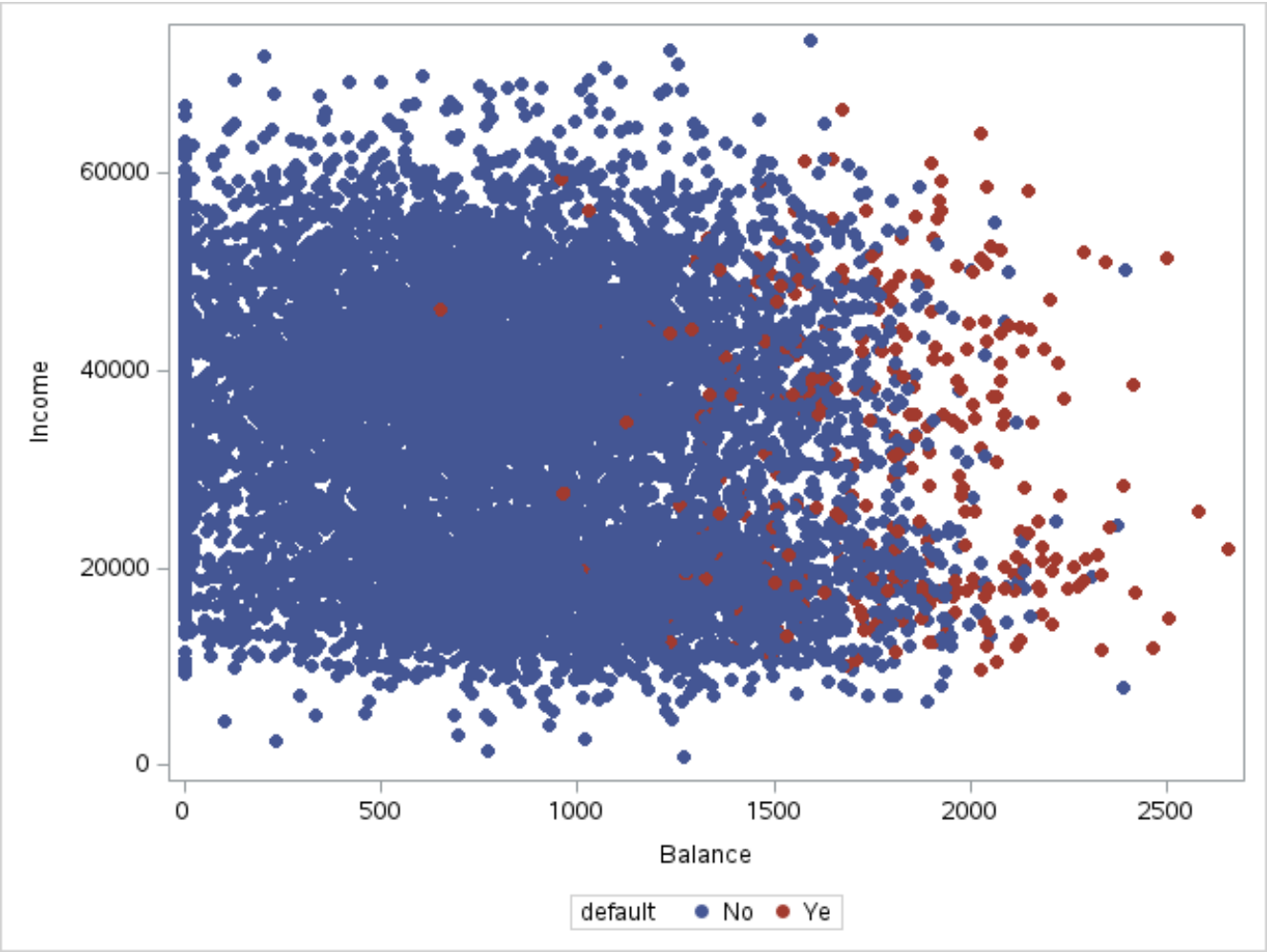
Logistic Regression

- Scatter plot

```
proc sgplot data=Default;
   scatter x=balance y=income / group=default
markerattrs=(symbol=CircleFilled);
   xaxis label='Balance';
   yaxis label='Income';
run;
```

Logistic Regression

- Scatter plot



Logistic Regression

- Create a boxplot for balance

```
proc sgplot data=Default;
   vbox balance / group=default;
   xaxis label='Default';
   yaxis label='Balance';
run;
```

- Create a boxplot for income

```
proc sgplot data=Default;
   vbox income /  group=default;
   xaxis label='Default';
   yaxis label='Income';
run;
```

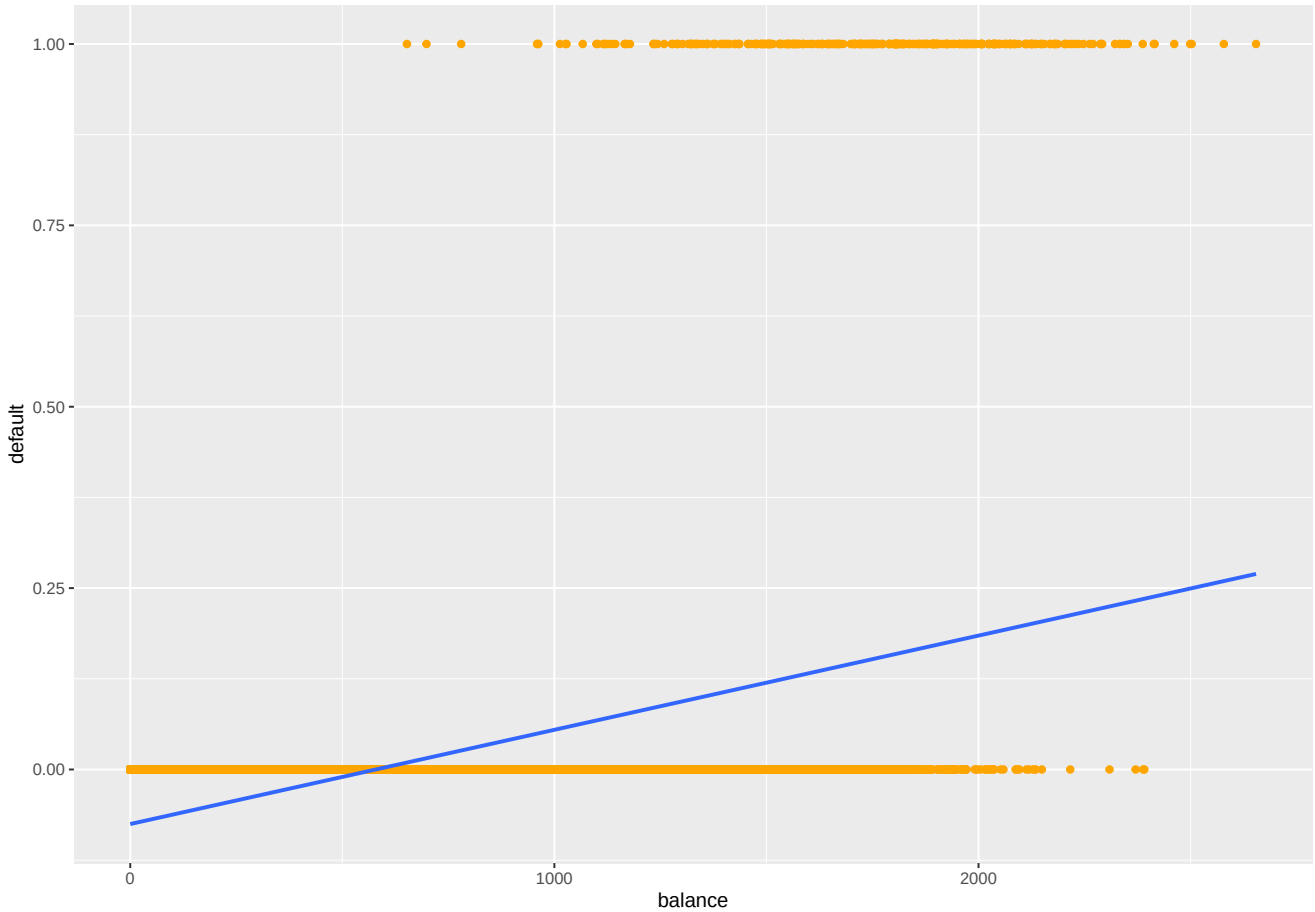
Logistic Regression

- Can we use Linear Regression?

Suppose $Y = \begin{cases} 0 & \text{if No} \\ 1 & \text{if Yes} \end{cases}$, Can we simply perform a linear regression of Y on X and classify as Yes if $\hat{Y} > 0.5$?

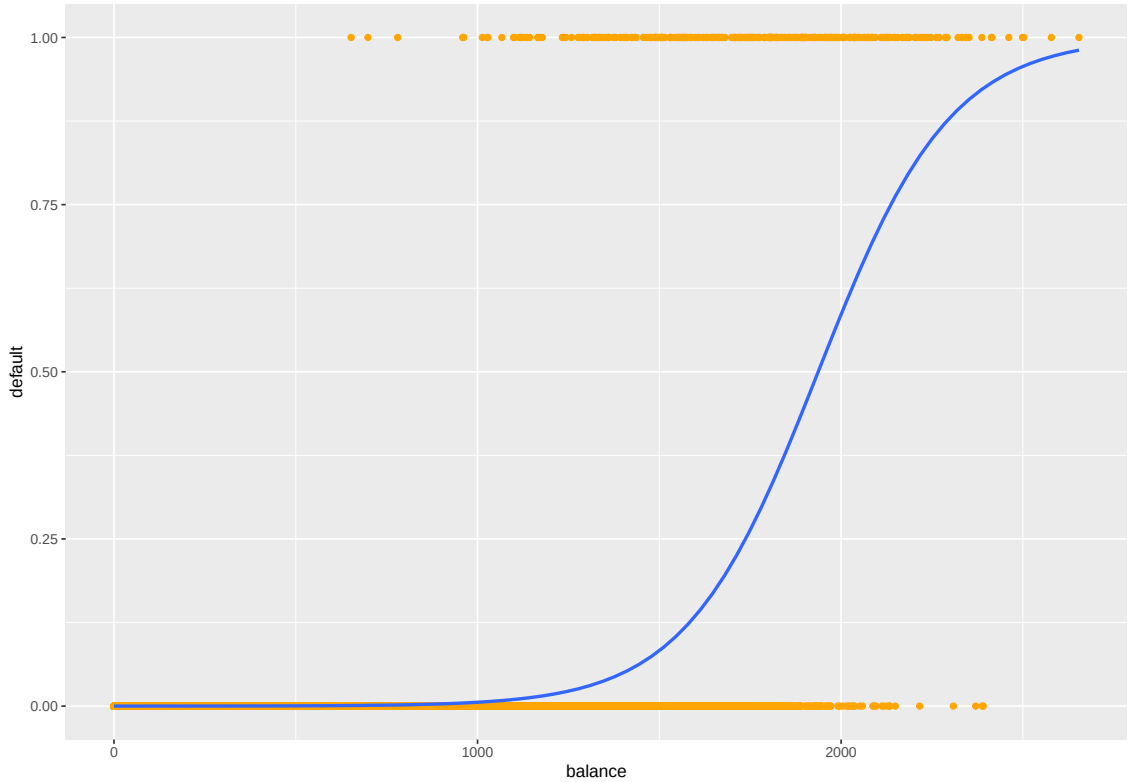
```
proc sgplot data=Default1;
scatter x=balance y=default_yes;
    reg x=balance y=default_yes / lineattrs=(color=orange);
  run;
```

Logistic Regression



Logistic Regression

- Linear regression might produce probabilities less than zero or bigger than one. So it can not give a good estimate of $E(Y|X = x) = Pr(Y = 1|X = x)$. Logistic regression is more appropriate.



Logistic Regression

- Recall that Logistic regression is the straightforward extension of linear regression to the classification setting.
- We first consider the case that $y \in \{0, 1\}$: a two-class classification problem.
- Let $p(X) = Pr(Y = 1|X)$
  - For example, we want to use biomarker level to predict probability of cancer.
- Logistic regression uses the form

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

  - $p(X)$ will lie between 0 and 1.
- Furthermore,

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

  - This function of $p(X)$ is called the logit or log odds (by log we mean natural log : ln).

Logistic Regression

- We use maximum likelihood to estimate the parameters
- We used `proc genmod` fit logistic regression model before. Now we use proc logistic because it supports variable selection.
  - https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.4/statug/statug_logistic_toc.htm

```
/*event='1' specifies that we model Pr(default=yes)*/
proc logistic data=Default1 outest=betas;
   model default_yes(event='1') = balance;
   output out=pred p=phat lower=lcl upper=ucl
          predprob=(individual);
run;

proc print data=betas;
   title2 'Parameter Estimates';
run;

proc print data=pred;
   title2 'Predicted Probabilities and 95% Confidence Limits';
run;
```

Logistic Regression

- What is our estimated probability of default for someone with a balance of $1000?
- With a balance of $2000?
- What value of Balance will give a predicted Default rate of 50%?

Logistic Regression

- Lets do it again, using student as the predictor

```
proc logistic data=Default1 outest=betas;
   model default_yes(event='1') = student_yes;
run;

proc print data=betas;
   title2 'Parameter Estimates';
run;
```

Logistic Regression with Several Variables

- Suppose that there are p features: $X_1, \ldots, X_p$.
- Just like before

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

- And just like before

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

Logistic Regression with Several Variables

```
proc logistic data=Default1;
   model default_yes(event='1') = balance income student_yes;
   output out=LogisticOutput predicted=fitted;
run;

proc print data=LogisticOutput;
   run;
```

Logistic Regression with Several Variables

- Why is coefficient for student negative, while it was positive before?

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-10.8690	0.4923	487.4978	<.0001
balance	1	0.00574	0.000232	611.8980	<.0001
income	1	3.033E-6	8.203E-6	0.1368	0.7115
student_yes	1	-0.6468	0.2363	7.4943	0.0062

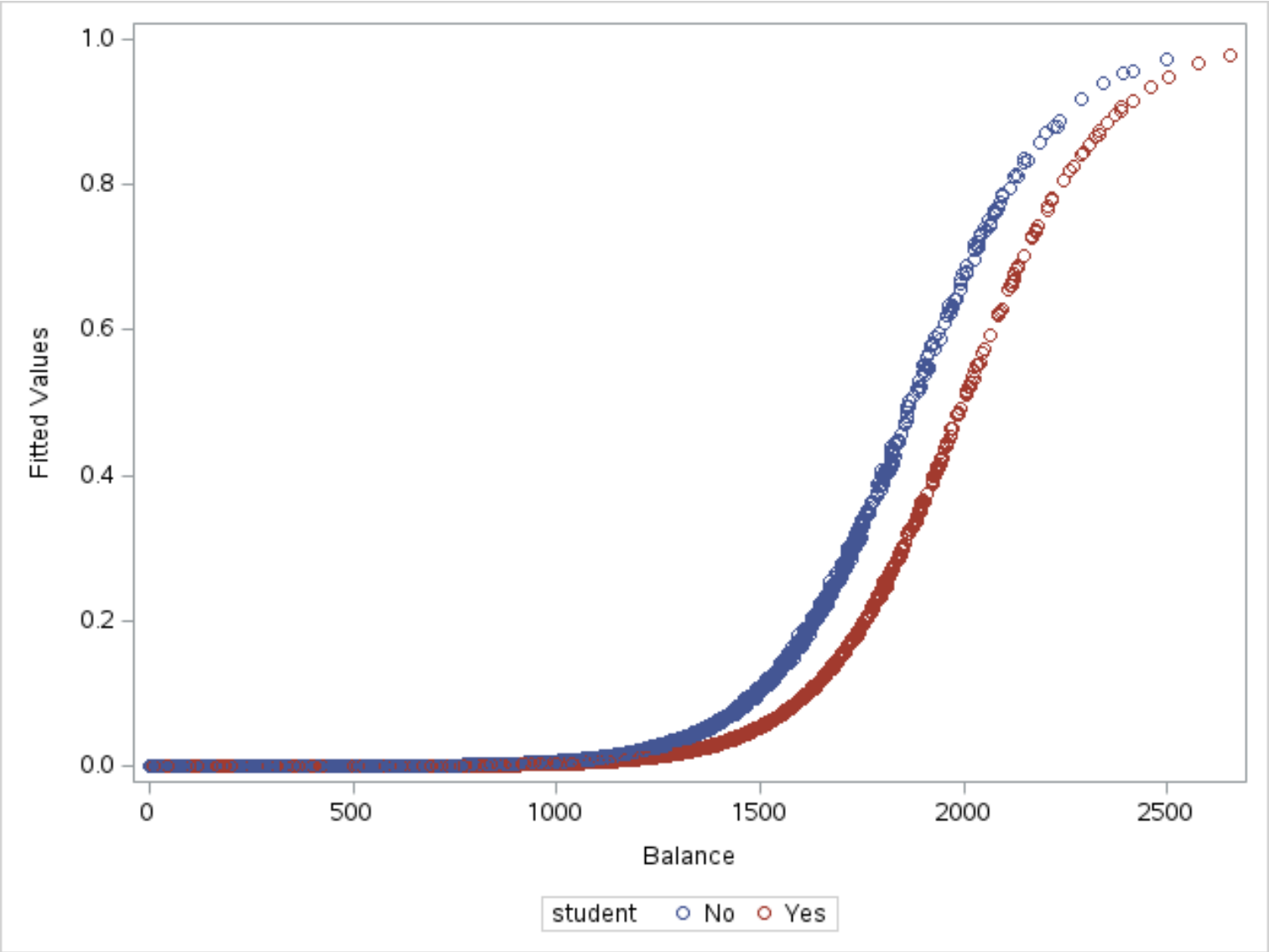
- Students tend to have higher balances than non-students, so their marginal default rate is higher than for non-students.
- But for each level of balance, students default less than non-students.
- Multiple logistic regression here can tease this out.

Logistic Regression with Several Variables

- Scatter plot of balance vs. fitted values, colored by student

```
proc sgplot data=LogisticOutput;
   scatter x=balance y=fitted / group=student;
   xaxis label='Balance';
   yaxis label='Fitted Values';
run;
```

Logistic Regression with Several Variables



Logistic Regression with Several Variables

- **Note** `proc logistic` can conduct variable selection.
  - But it does not provide built-in support for cross-validation during variable selection.

```
proc logistic data=Default1;
   model default_yes(event='1') = balance income
student_yes/selection=stepwise;
  run;
```

- It can be seen that `income` is removed from the model

License