JOHNS HOPKINS
CAREY BUSINESS SCHOOL

Data Science and Business Intelligence

BU.330.780

Session 4

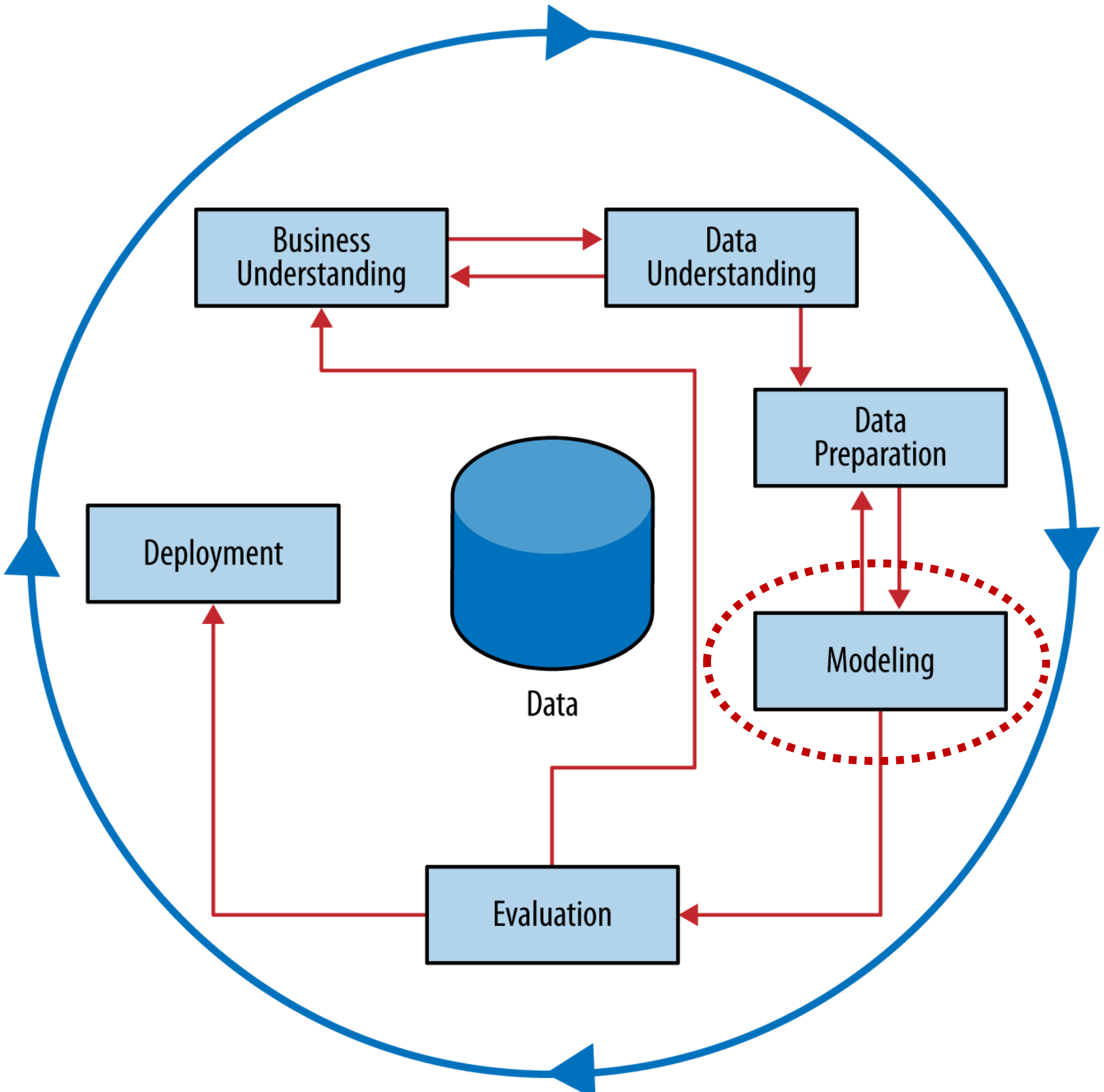
Instructor: Changmi Jung, Ph.D.

Announcement

- Assignment #3 due before week 6 class
  - Two files you need to download and work on: R Markdown and an Excel file (data) on Canvas > Week 6
  - Follow the directions in the markdown file to complete the assignment
  - The style is similar to the previous assignments (answer the test style questions and attach the knitted file – render it to a Word docx or HTML)

- Project status report (1 page suggested) is due Week 6

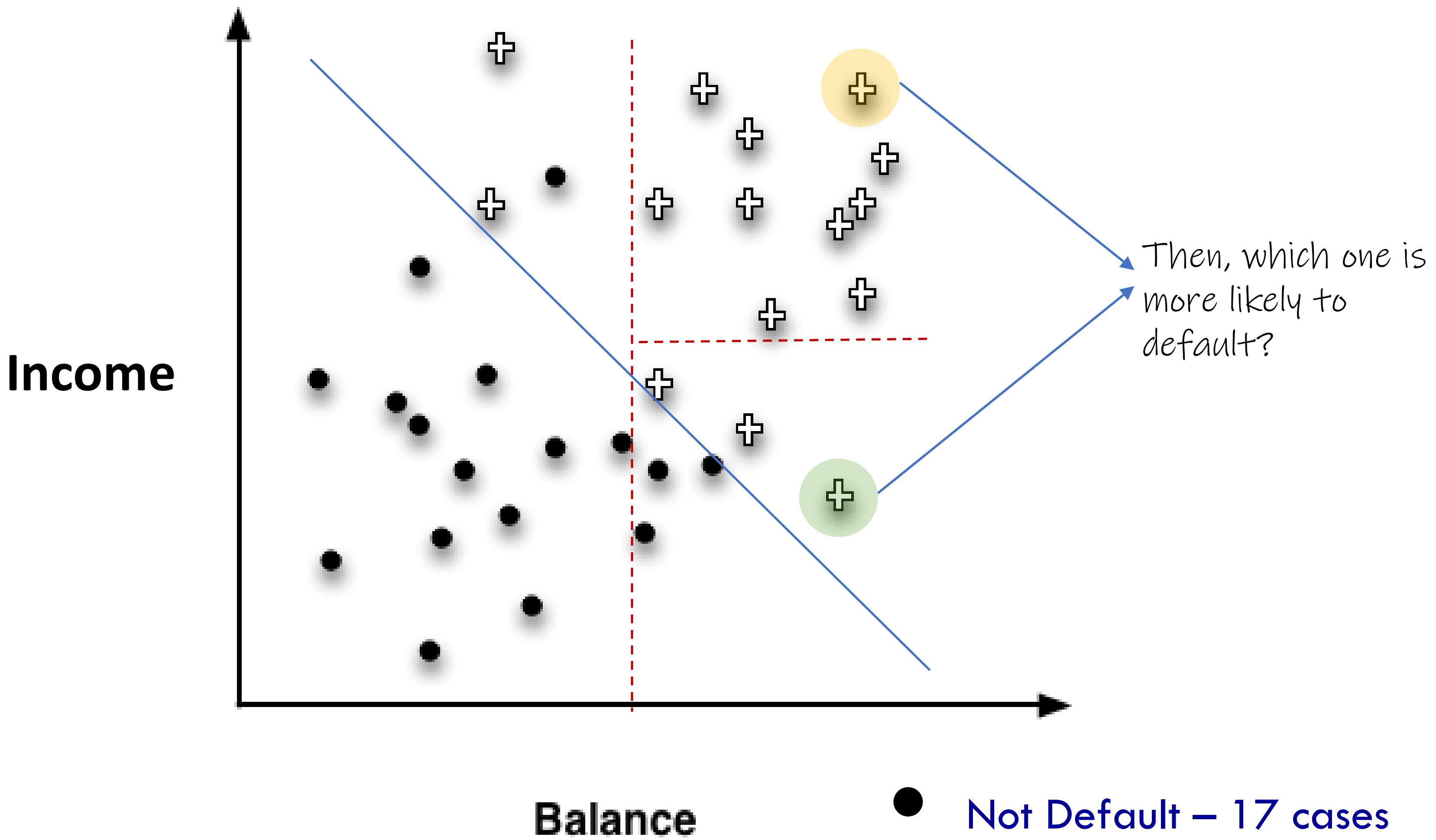
Data science as a process



Supervised Segmentation:

Linear Classifier – SVM and Logit Regression

What alternatives are there to partitioning?



● Not Default – 17 cases
✚ Default – 15 cases

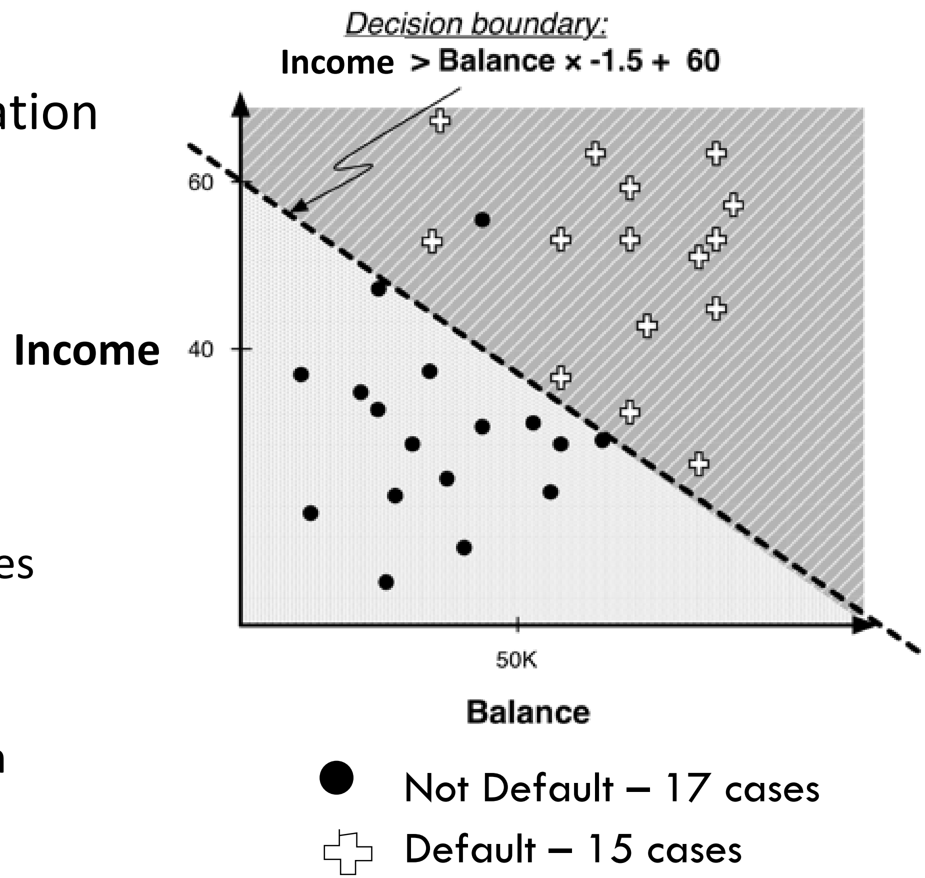
Linear classifiers

$$class(x) = \begin{cases} + \text{ if } 1.0 \times \text{Income } - 1.5 \times Balance + 60 > 0 \\ \bullet \text{ if } 1.0 \times \text{Income } - 1.5 \times Balance + 60 \leq 0 \end{cases}$$

- A linear classifier is a numeric classification model, which can be written as a linear function.

  $f(x) = w_1 x_1 + w_2 x_2 + w_3 x_3 + \cdots c$

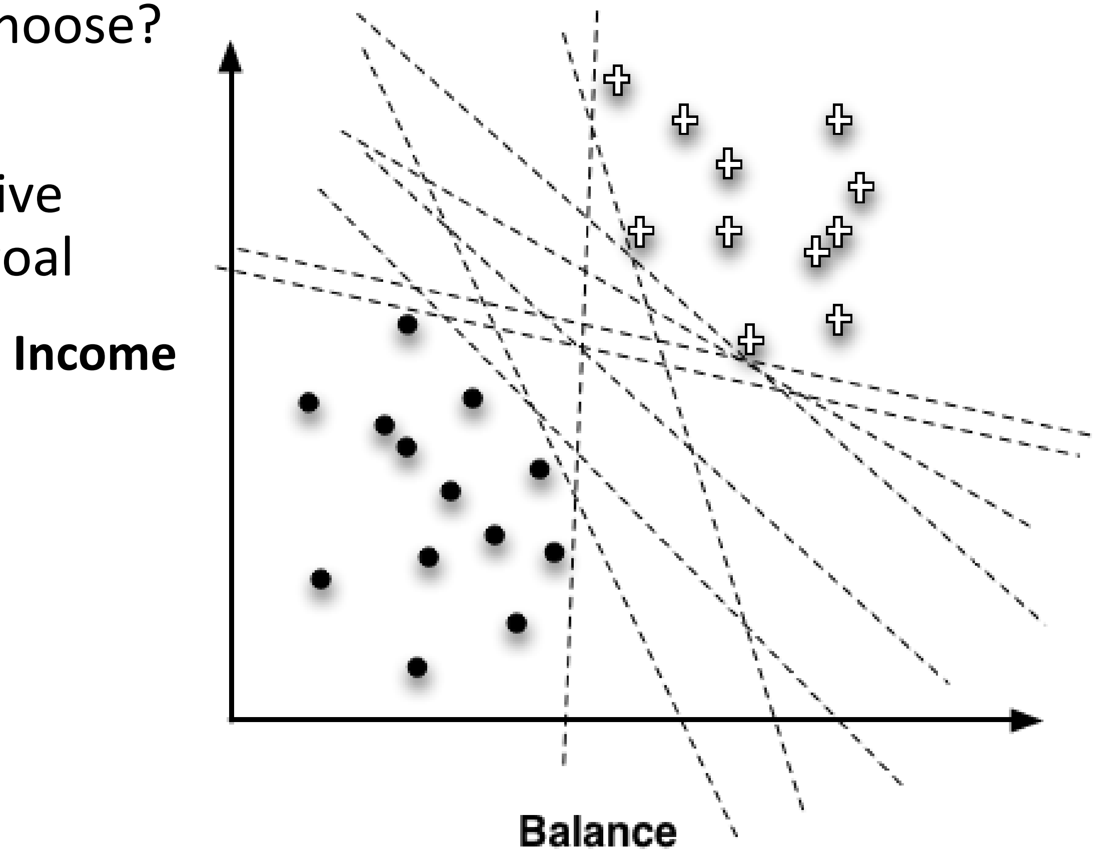
- Fit parameters to a particular dataset
  - Find a good set of weights w.r.t. the features
  - For normalized variables, weights may be interpreted as the importance indicators
  - Considers all attributes at once rather than select one at a time



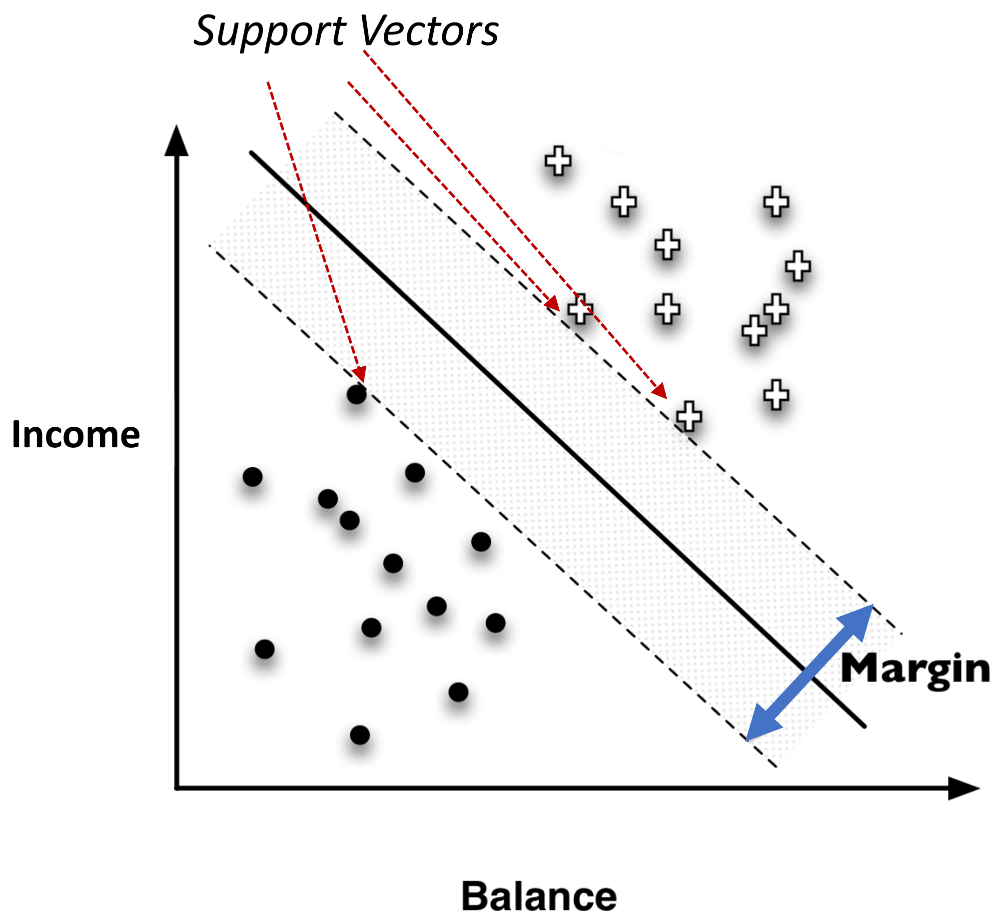
Which linear classifier should we choose?

- What should be our objective in choosing the parameters?
  = What weights should we choose?

- We need to define an objective function that represents our goal sufficiently



An intuitive approach to Support Vector Machines (SVM)



- SVM classifies instances based on a linear function of the features
- Objective function is based on a simple idea: **maximize the margin**
  - Fit the widest bar between the classes
  - Once the widest bar is found, the linear discriminant will be the center line through the bar
- We will focus on linear kernels, but SVM can use different types of kernels.

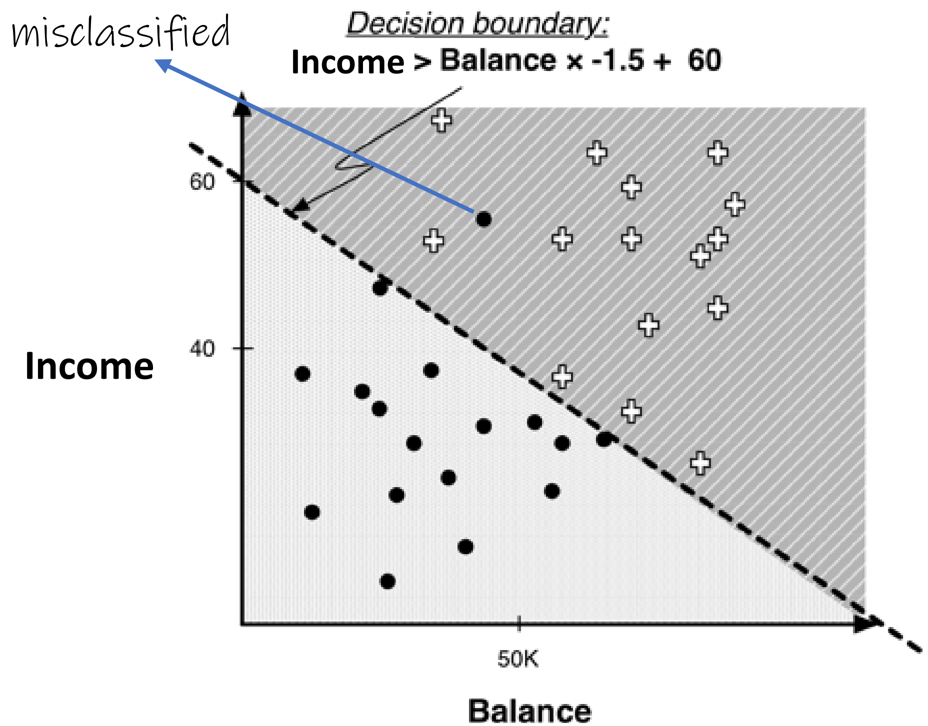
What if we cannot find a margin that perfectly separate two classes?

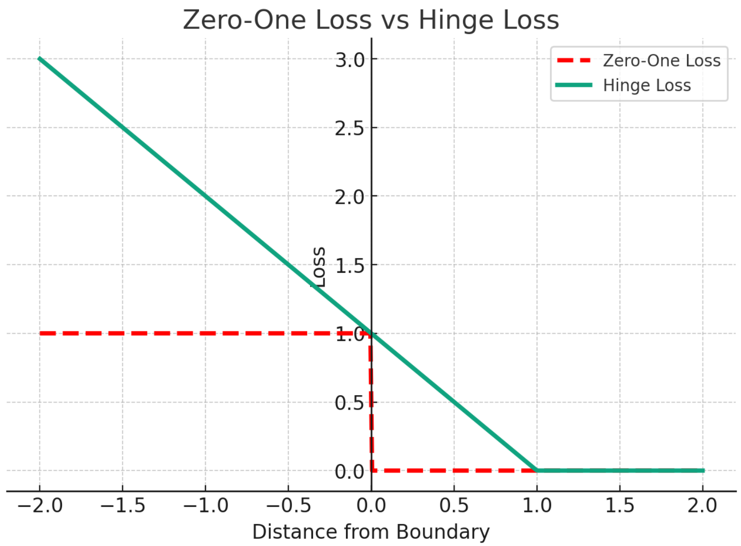
What if there is no perfect separating line?

- Intuitively, solve the following optimization problem:

  Maximize (Margin – loss)

- A loss function determines how much penalty should be assigned to an instance based on the error in the model's predicted value.





zero-one loss: 0 for a correct prediction and 1 for an incorrect prediction. This one is hard to minimize.

SVM Algorithm

- Soft Margin SVM

$$Minimize\ \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \xi_i$$

slack variable.

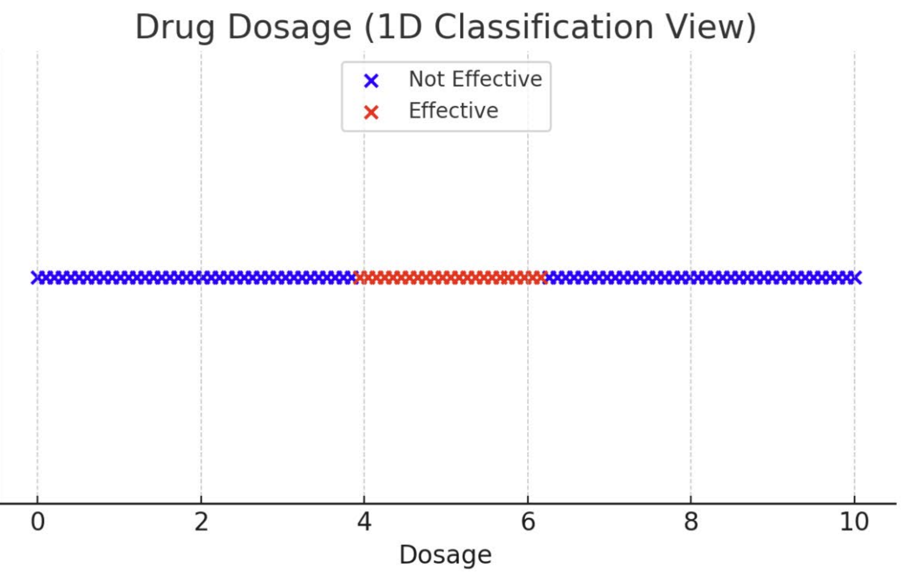
$$Subject\ to$$

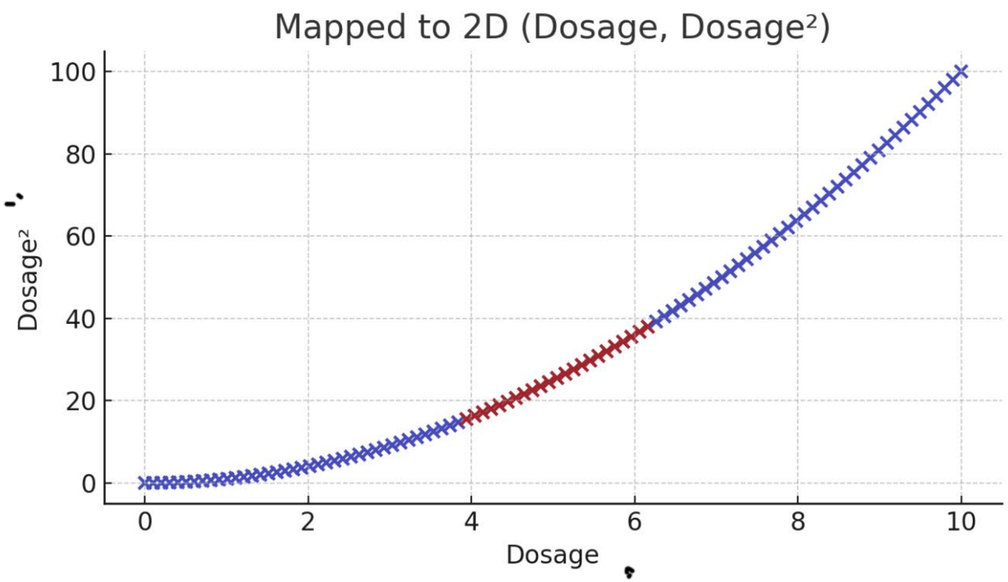
$$y_i(w^T x_i + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0,\ \ \forall i$$

- w: weight vector which defines the hyperplane
- b: bias term
- ξ_i: slack variables = penalties for misclassification or margin violation
- C: regularization parameter – controls the trade-off between margin width and classification error

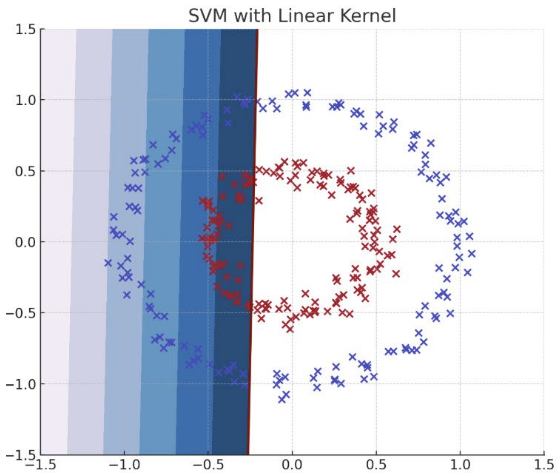
The Role of Kernel Functions

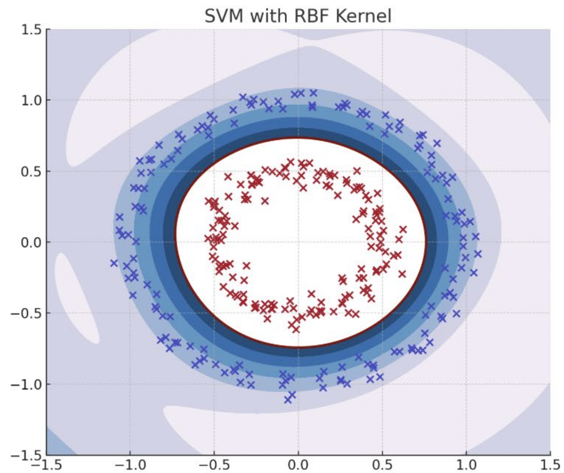
- A kernel function implicitly maps the input data into a higher-dimensional feature space where it becomes easier to separate the data with a linear hyperplane.

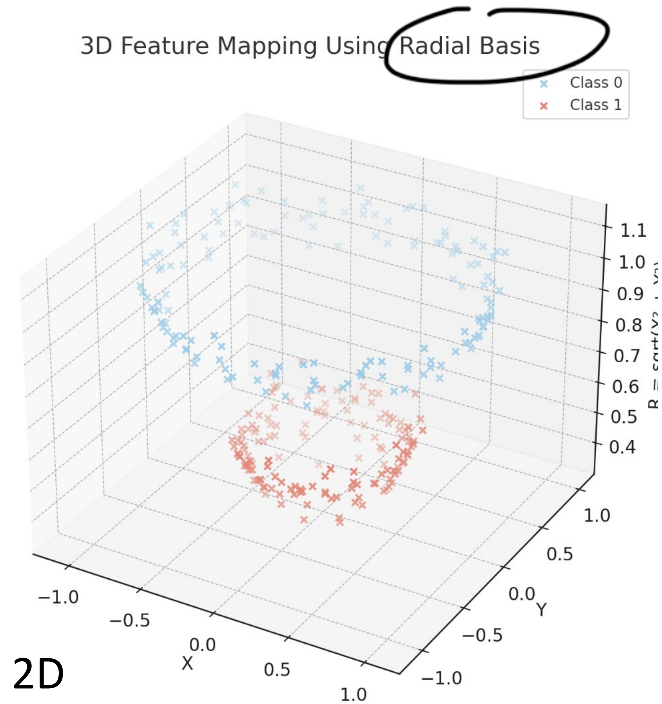




The Role of Kernel Functions







Radial transformation is applied to lift 2D into 3D. $r = \sqrt{(x^2 + y^2)}$

Strength and Weakness of SVM

Feature	Advantage
High-Dimensional Data	Very effective
Kernel Trick	Powerful for non-linear problems
Small training data	Efficient with small datasets
Outliers & Noise	Margin helps some generalization
Multi-class Problems	Can handle with extensions

Feature	Disadvantage
Scalability	Not efficient with a large dataset
Kernel Functions	Performance varies based on Kernel types, kernel-specific parameters, and C (regularization parameter)
Probability scores	Does not produce probability output
Interpretability	Hard to explain

What if we want estimates of class membership probability?

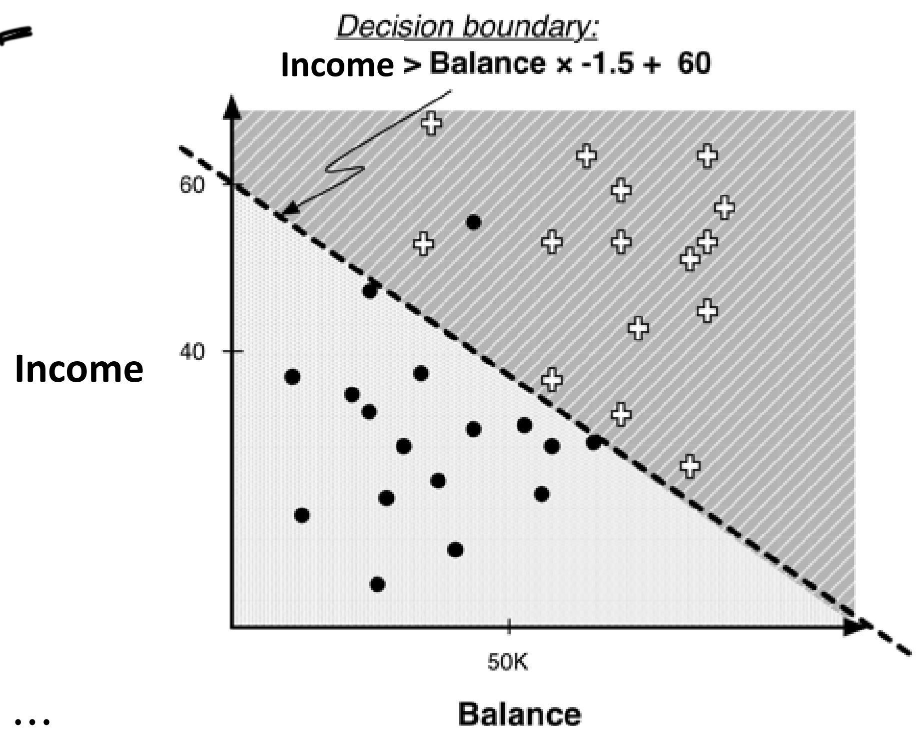
- SVM produces decision values (scores) not probability. Probability should range from zero to one.

- What if we want numeric function based model and estimates of class membership probability?

- **Logistic Regression**

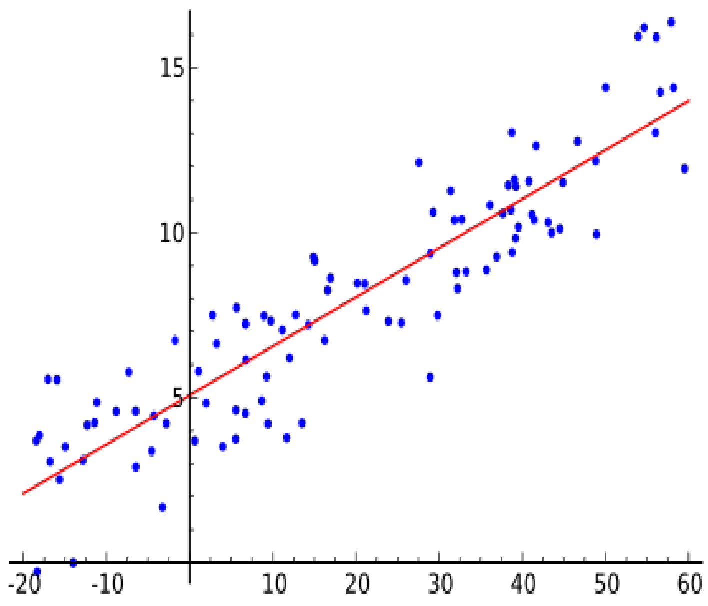
$$\log\left(\frac{p}{1-p}\right) = f(x) = w_0 + w_1 x_1 + w_2 x_2 + \cdots$$

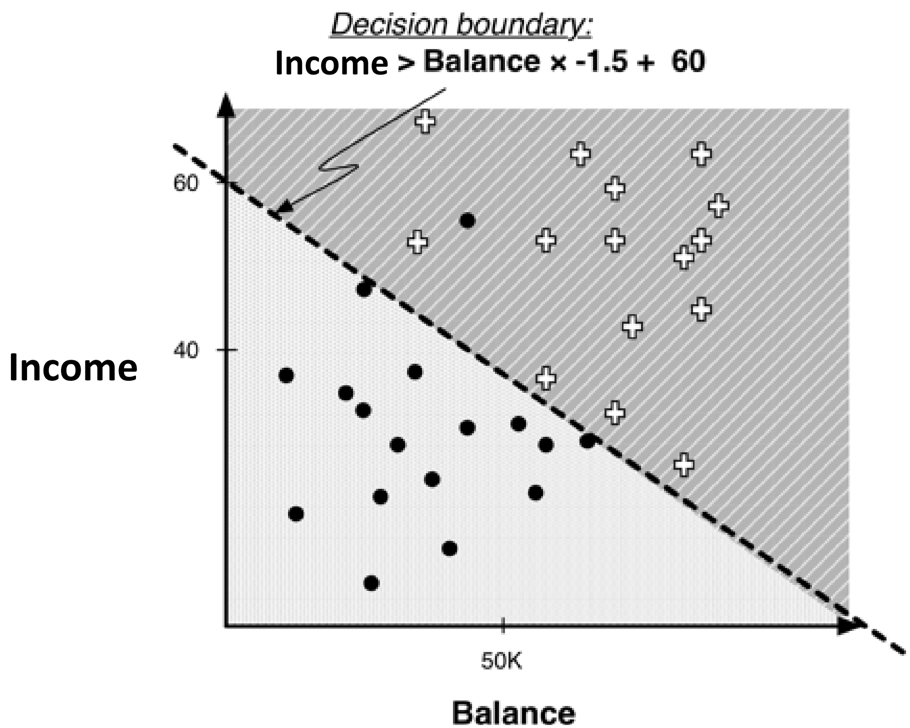
$$\hat{p}_+(x) = \frac{1}{1+e^{-f(x)}}$$



Linear Regression vs. Logistic Regression (1/2)

$$\hat{y} = a + bx$$

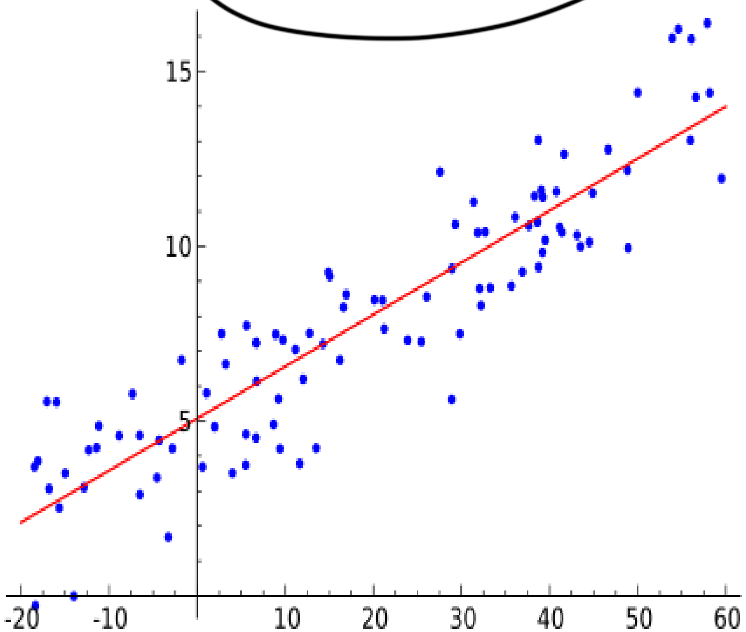




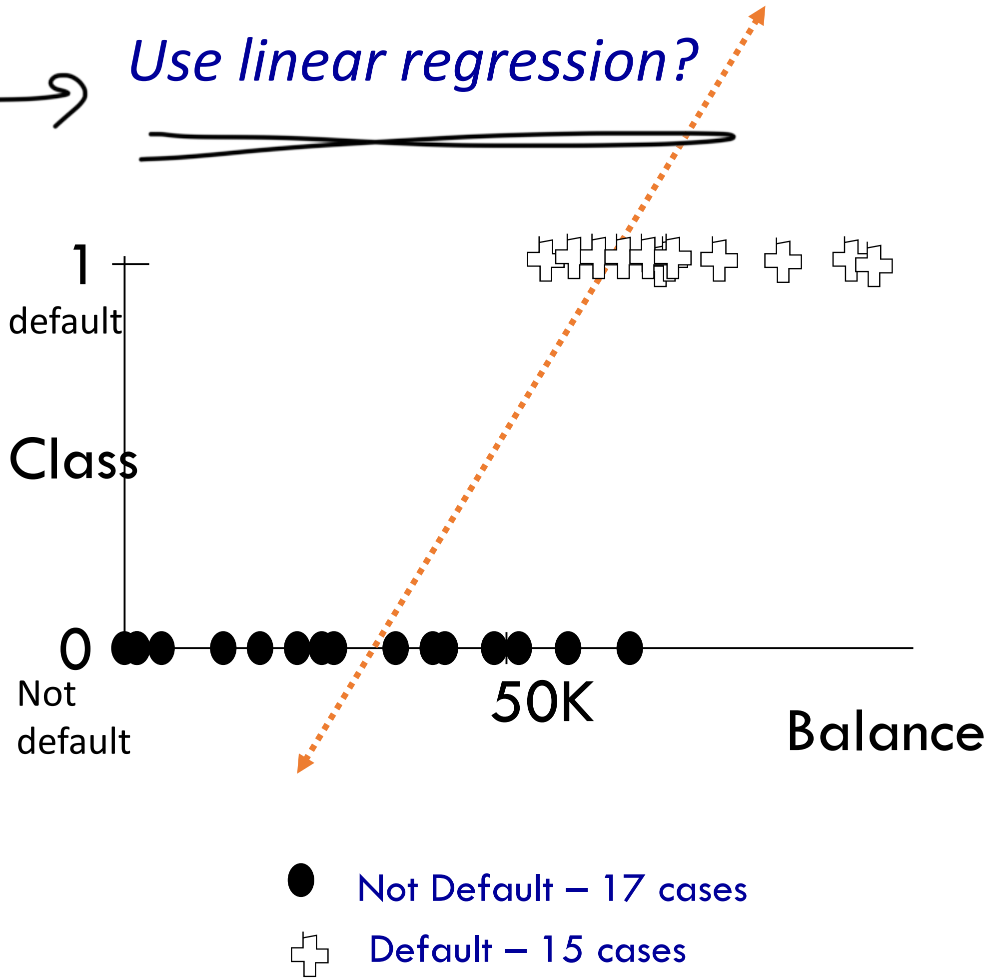
- Purpose?
- Good regression line vs. Good decision boundary?
- Well… it's not an apple-to-apple comparison (one vs. two variables)

Linear Regression vs. Logistic Regression (2/2)

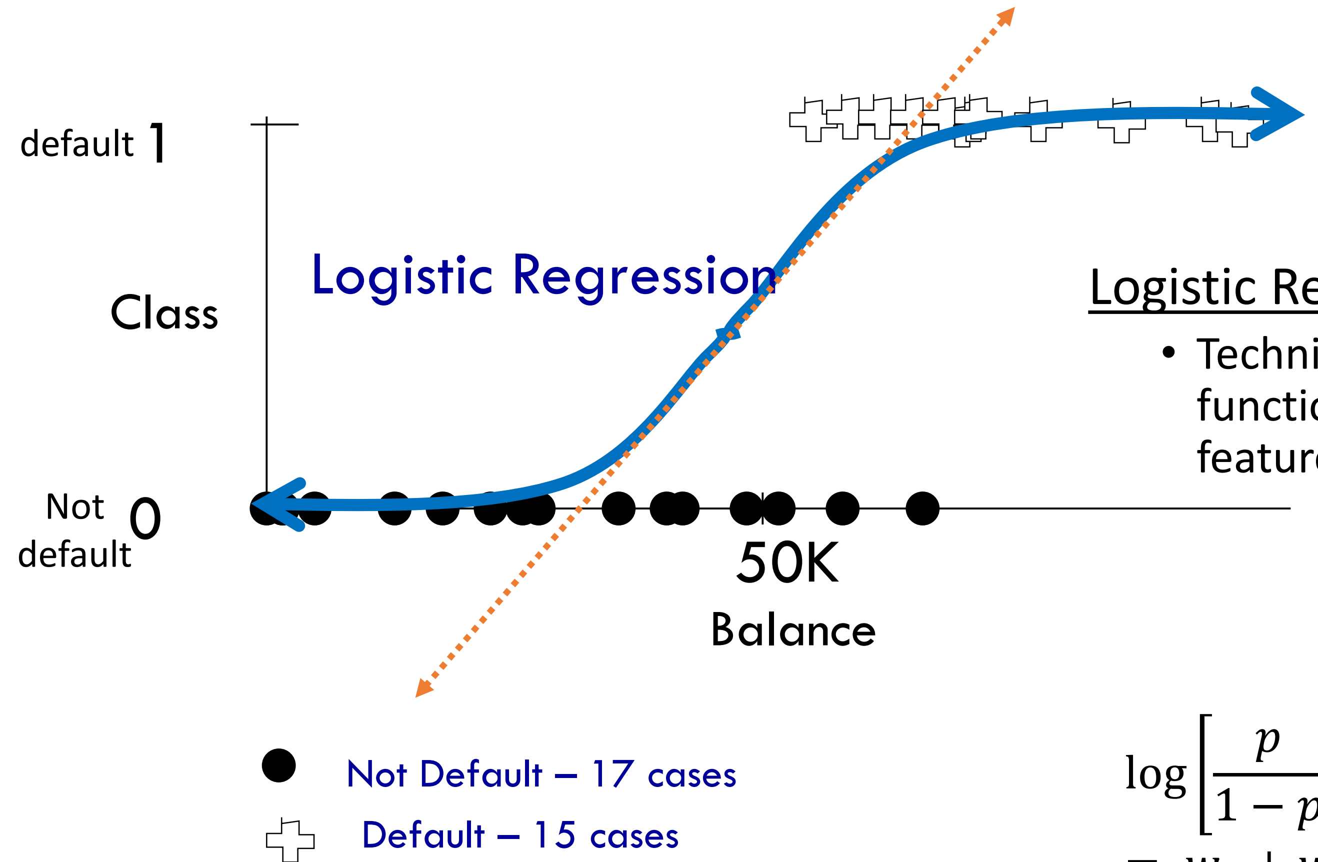
$$\hat{y} = a + bx$$



~~*Use linear regression?*~~



A simpler case (one independent variable) - Estimate the probability of membership in class 1



$$\hat{p}_{+}(x) = \frac{1}{1 + e^{-f(x)}}$$

Logistic Regression

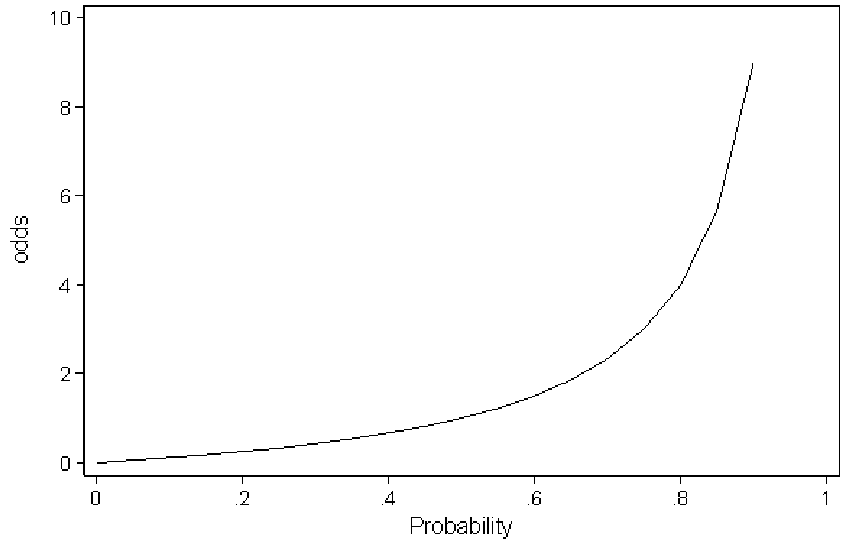
- Technically: "log odds," or logit function, is a linear function of the features:

$$\log\left[\frac{p}{1-p}\right] = f(x)$$

$$= w_0 + w_1 x_1 + w_2 x_2 + \ \ldots$$

From Probability to Odds to Log Odds

p	p/(1-p)	Log(p/1-p)
0.5	1	0
0.9	9	2.19
0.999	999	6.9
0.01	0.010101	-4.6
0.001	0.001001	-6.9



- The odd of event is the ratio of the probability of the event occurring to the probability of the event not occurring.

Technically: "log odds" is a linear function of the features and ranges between -∞ to ∞.

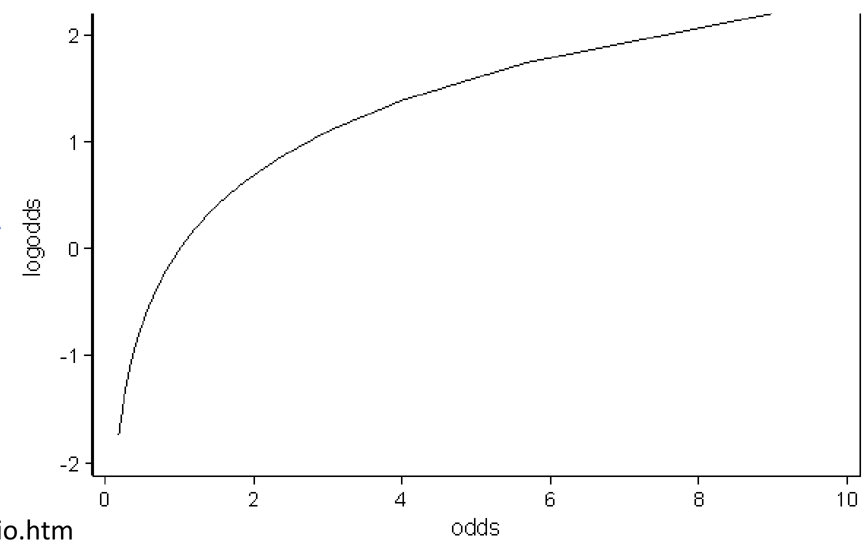


Image source: http://www.ats.ucla.edu/stat/mult_pkg/faq/general/odds_ratio.htm

SVM and Logit Regression with R

Logistic Regression with R

- The R command `glm()` fits generalized linear models, a class of models that includes logistic regression.
- For logistic regression, the family of error distribution is the *binomial*,
  `>glm (formula, family = “binomial”, data)`

- Model:
- Our goal is to find $w_0, w_1, w_2,$ and w_3 that fit the data best.

$$f(x) = Default = w_0 + w_1 \times Student + w_2 \times Balance + w_3 \times Income$$

```
> ols_model<-lm(default~student+balance+income,data=Default)
> logit_model<-glm(default~student+balance+income,
                             family="binomial",
                             data=Default)
> summary(logit_model)  # the logistic regression results.
```

Interpret the Results (1/2)

```
> summary(logit_model)
Call:
glm(formula = default ~ student +                c
"binomial", data = Default)

Devia
    M                                3Q        Max
-2.46                                .0203   3.7383

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
studentYes  -6.468e-01  2.363e-01  -2.738  0.00619 **
balance      5.737e-03  2.319e-04  24.738  < 2e-16 ***
income       3.033e-06  8.203e-06   0.370  0.71152
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The changes of the **log odds** of the target when the variable changes by one unit

The z-value is obtained by calculating “estimate”/”std.error”.

The larger z-value is, the smaller Pr(>|z|) is. A small p-value (usually, less than 0.05) indicates the higher chance that the variable is associated with the target.

Interpret the Results (2/2)

- **Log(p/1-p) = -10.87 - 0.64 Student + 0.0057 Balance** $= z = f(x).$

$$\frac{1}{1+e^{-z}}$$

- The estimated coefficient is the expected change in the log odds of being a defaulter for a unit increase in the predictor variable, holding the other predictors equal.

- ***Student***: The log odd of a student being a defaulter is 0.64 *less* than a customer who is not a student (holding balance constant).
- ***Balance***: The log odds of being a defaulter increase by 0.0057 with the increase in Balance by one dollar.

```
                Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
studentYes  -6.468e-01  2.363e-01  -2.738  0.00619 **
balance      5.737e-03  2.319e-04  24.738  < 2e-16 ***
income       3.033e-06  8.203e-06   0.370  0.71152
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Use of Model to Predict

$$f(x) = Default = -10.1 + (-0.64) \times Student + (0.005) \times Balance + (0.0000003) \times Income$$

```
> Default$log_odd<-predict(logit_model)                    #get log-odd (default)
> Default$prob<-predict(logit_model,type="response")   #get probability
> head(Default)
  default student   balance    income log_odd         prob
1      No      No  729.5265 44361.625 -6.549544 0.0014287239
2      No     Yes  817.1804 12106.135 -6.791338 0.0011222039
3      No      No 1073.5492 31767.139 -4.614261 0.0098122716
4      No      No  529.2506 35704.494 -7.724689 0.0004415893
5      No      No  785.6559 38463.496 -6.245449 0.0019355062
6      No     Yes  919.5885  7491.559 -6.217871 0.0019895182
```

prob <- 1 / (1 + exp(-logit))

$$\hat{p}_+(x) = \frac{1}{1 + e^{-f(x)}}$$

Supervised Segmentation - Summary

Important differences between classification tree and linear models

» A classification tree

- uses decision boundaries that are perpendicular to the instance-space axes
- is a "piecewise" classifier that segments the instance space recursively → cut into arbitrarily small regions, possible

» A linear classifier

- use decision boundaries of any direction or orientation
- places a single decision surface through the entire space

Which of these characteristics are a better match to a given data set?

Classification Tree vs. Linear model: Factors to consider

- What is more comprehensible to the stakeholders?
  - rules?
  - a numeric function?

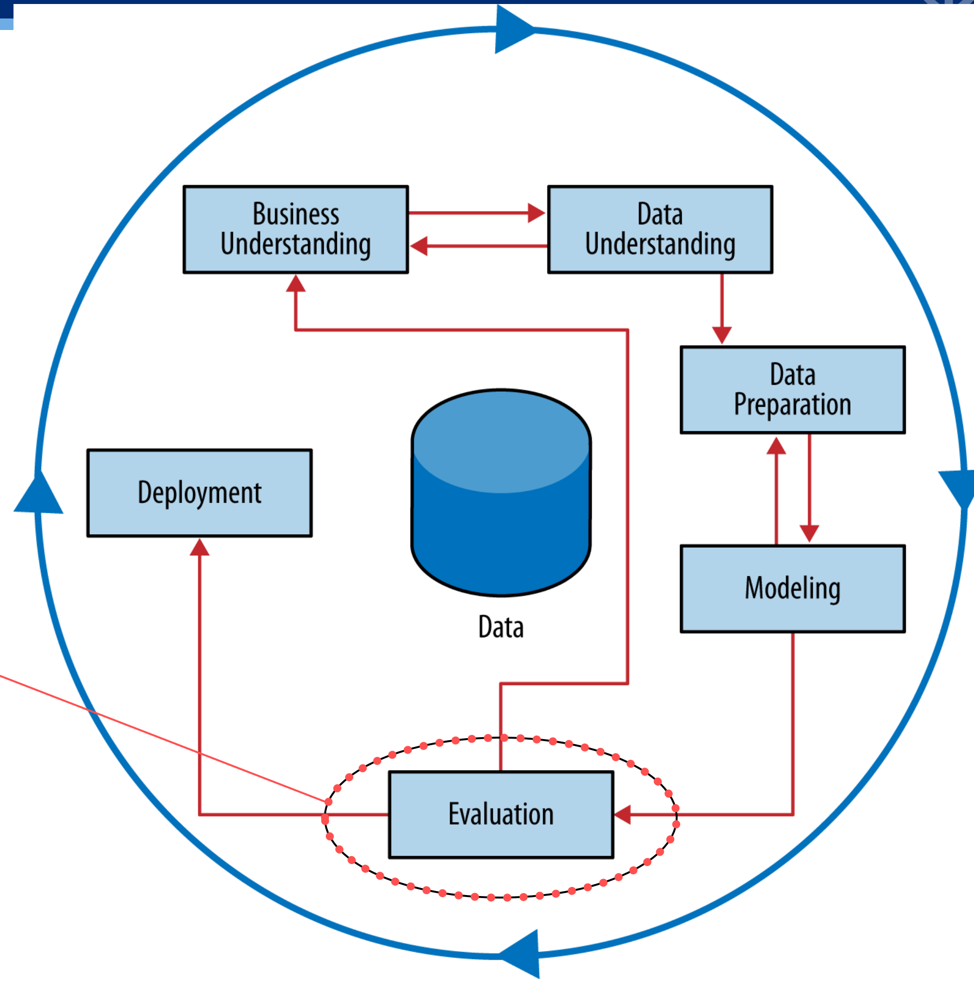
- How much data do you have?!
  - There is a key tradeoff between the complexity that can be modeled and the amount of training data available

- What are the characteristics of the data: missing values, types of variables (numeric, categorical), relationships between them, how many are irrelevant, etc.
  - Trees are fairly robust to these complications

Choice of algorithm is not trivial!

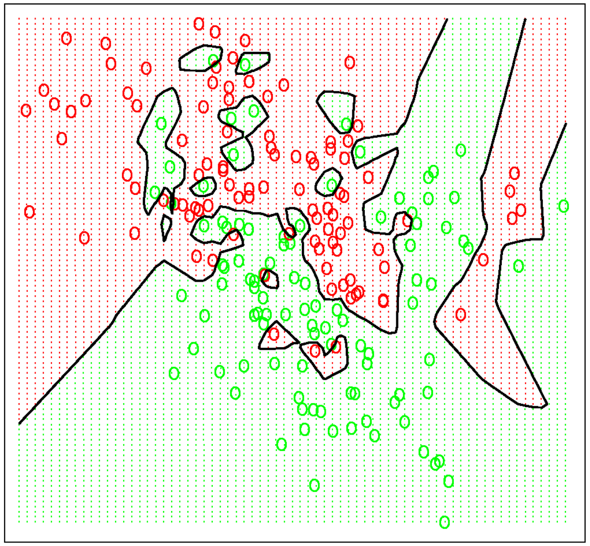
Or, why not try both!

- Integrated DM/ML packages now allow us to try multiple models easily…
- … and sort them out in Evaluation



Preview of next week...

- What is overfitting and can we avoid it?
- Hold-out (& cross) validation
- Confusion matrix
- ROC analysis



Actual classes

Predicted classes

	P	N
PP	True positives	False positives
PN	False negatives	True negatives