

# Features: Representing your data

COMS21202, Part III

- Song Liu ([song.liu@bristol.ac.uk](mailto:song.liu@bristol.ac.uk)), Lecturer in Data Science and A.I.

# Introduction

- Song Liu ([song.liu@bristol.ac.uk](mailto:song.liu@bristol.ac.uk)), Lecturer in Data Science and A.I.

# Machine Learning Pipeline

Part I

Data Acquisition

sampling/quantization



We are here, Part III

Feature Engineering

representing your data



Part I, II

Training algorithms

classification/regres  
sion/clustering...



Prediction/Inference

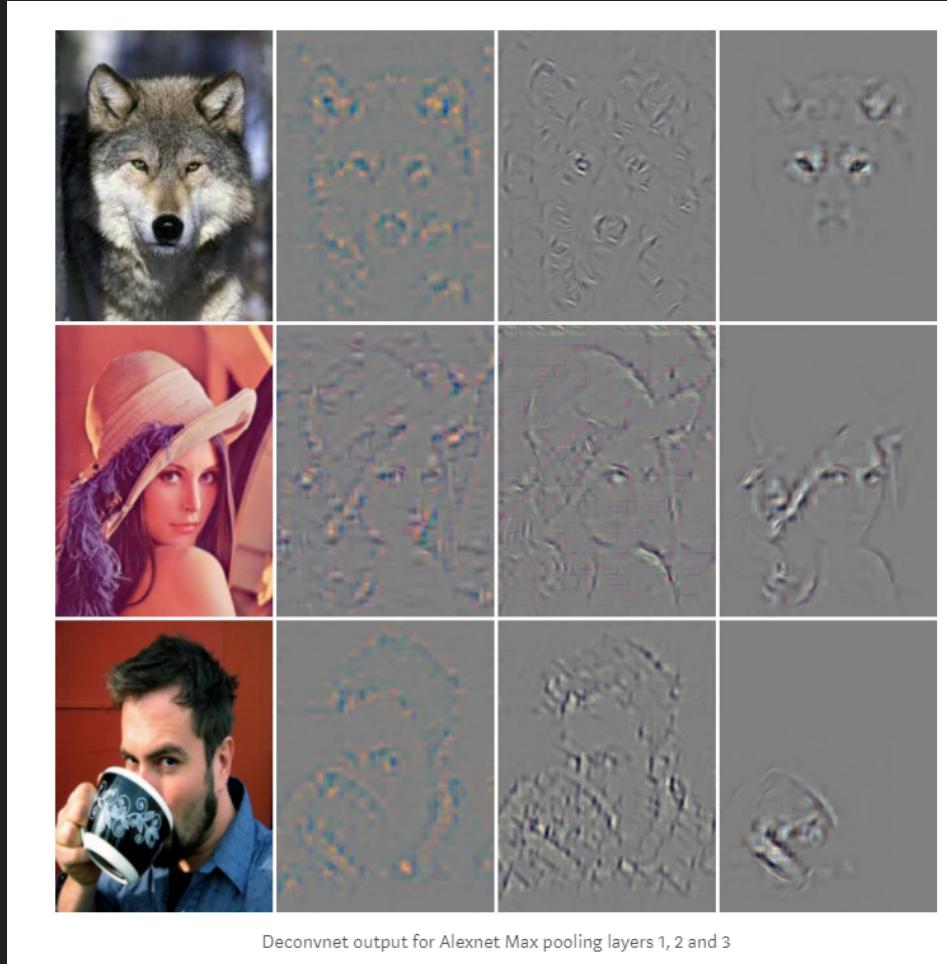
Making decisions,  
cats or dogs?

- Song Liu ([song.liu@bristol.ac.uk](mailto:song.liu@bristol.ac.uk)), Lecturer in Data Science and A.I.

# How does machine see the world?

- Machine does not see the world in the same way we do.
- It does not need to.
- It only needs the representation of info to perform its task.

# How does machine learning algorithm see the world?



○ Visualization of layers in Alexnet.  
○ Zeiler and Fergus, ECCV 2014

- Song Liu ([song.liu@bristol.ac.uk](mailto:song.liu@bristol.ac.uk)), Lecturer in Data Science and A.I.

# Turning Data into Features

- Modern machine learning **rarely** uses **raw data** input to perform learning tasks.
- Raw input is usually transformed into a more powerful representation: **features**.
- This procedure of representing data using features is usually referred as **feature engineering** in literatures.

# Feature Engineering

- Task: finding a feature **transform function**  $f(x)$ , which takes a  **$d$ -dimensional raw input  $x$**  and outputs a  **$m$ -dimensional feature vector.**
- Feature function  $f$  is the medium through which your learning algorithm interacts with your data.
- Let us put feature engineering in the context of **Least squares**.

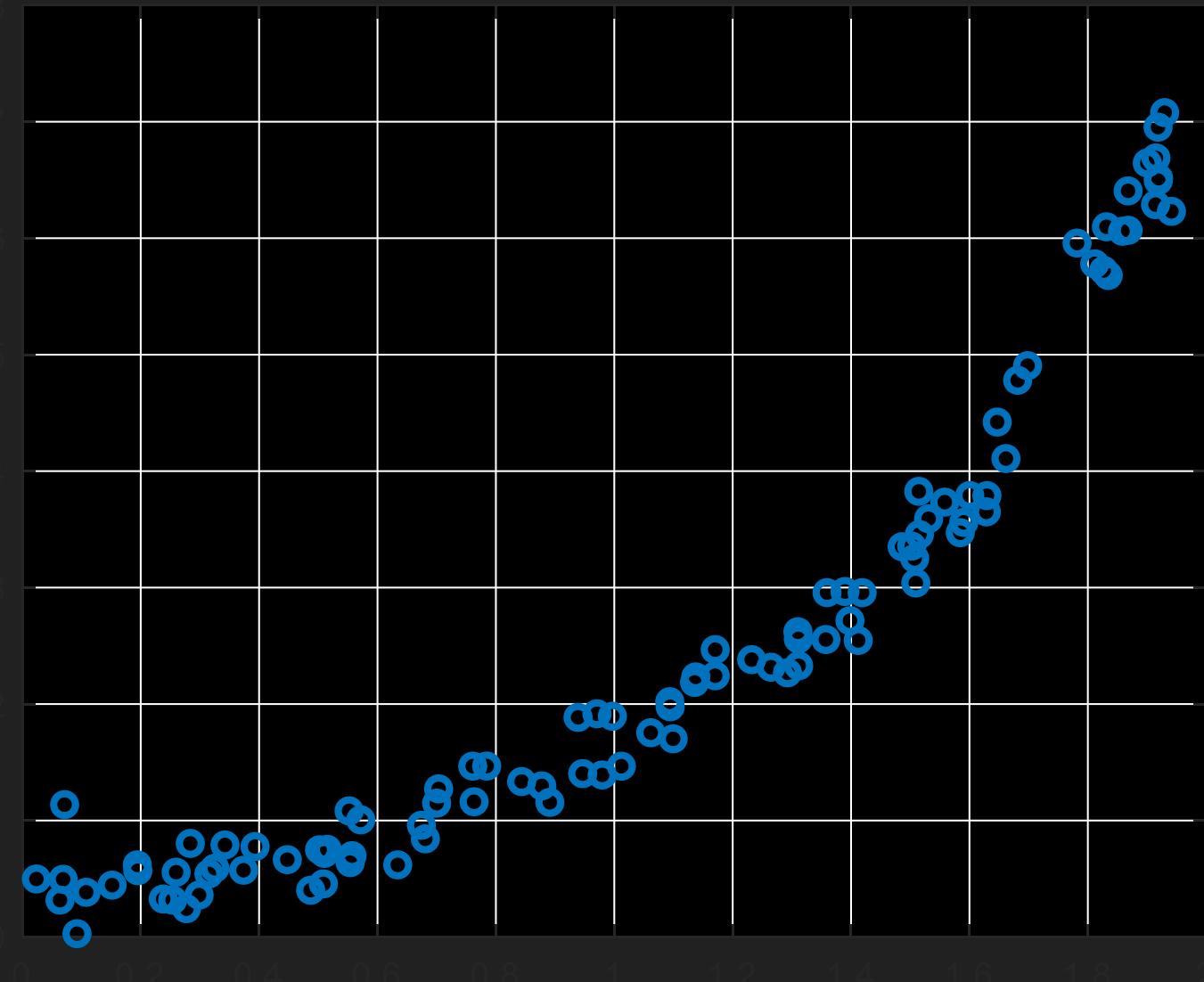
# An Appetizer

- Song Liu ([song.liu@bristol.ac.uk](mailto:song.liu@bristol.ac.uk)), Lecturer in Data Science and A.I.

# Least Squares (LS) + Feature Transform $f$

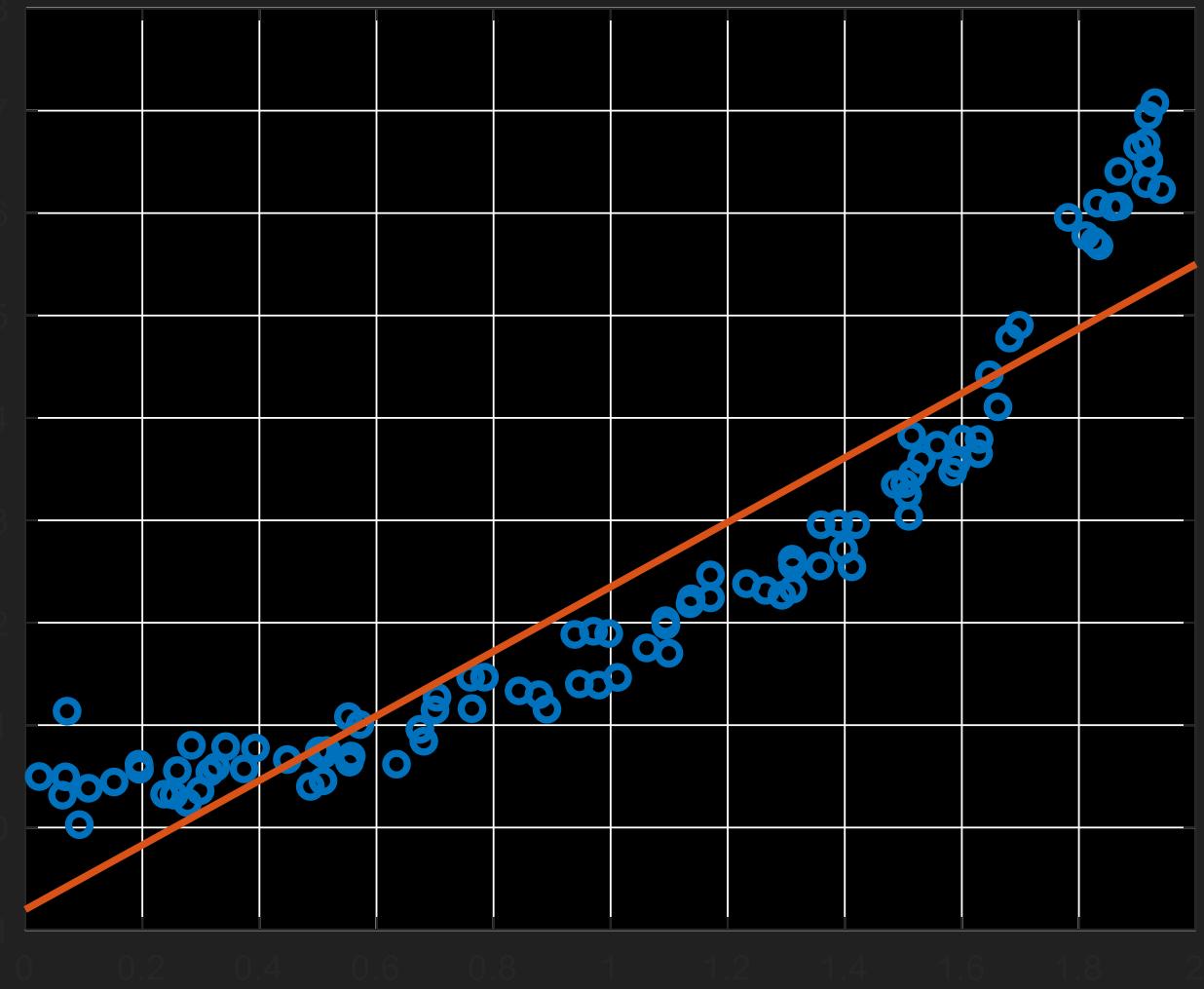
- Recall, given  $D = \{(y_i, x_i)\}_i, y_i \in R,$
- LS solves the following minimization:
  - $\hat{\beta} := \arg \min_{\beta} \sum_{i=1}^n (y_i - \beta x_i)^2 \quad (1)$
- Replace  $x$  with  $f(x)$ , a feature transform
  - $\hat{\beta} := \arg \min_{\beta} \sum_{i=1}^n (y_i - \beta f(x_i))^2 \quad (2)$
- (1) and (2) are identical if  $f(x) = x.$

# LS Regression on Nonlinear Dataset



See Lab 3, Q7  
for a similar  
example

# LS Regression on Nonlinear Dataset

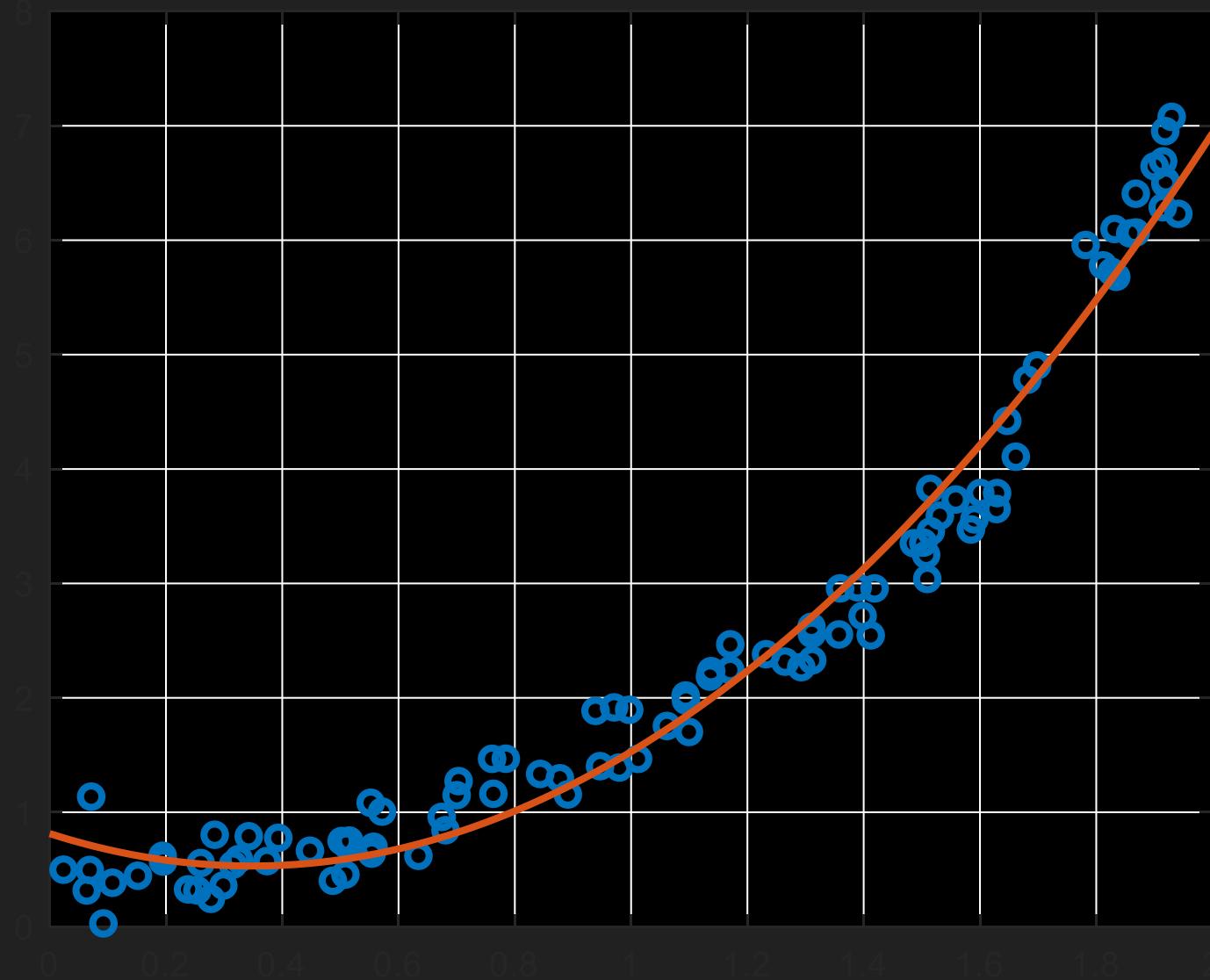


$$f(x) = x$$

Square error:  
 $\sum_{i=1}^n (y_i - \hat{\beta}f(x_i))^2,$

Square error: 57

# LS Regression on Nonlinear Dataset



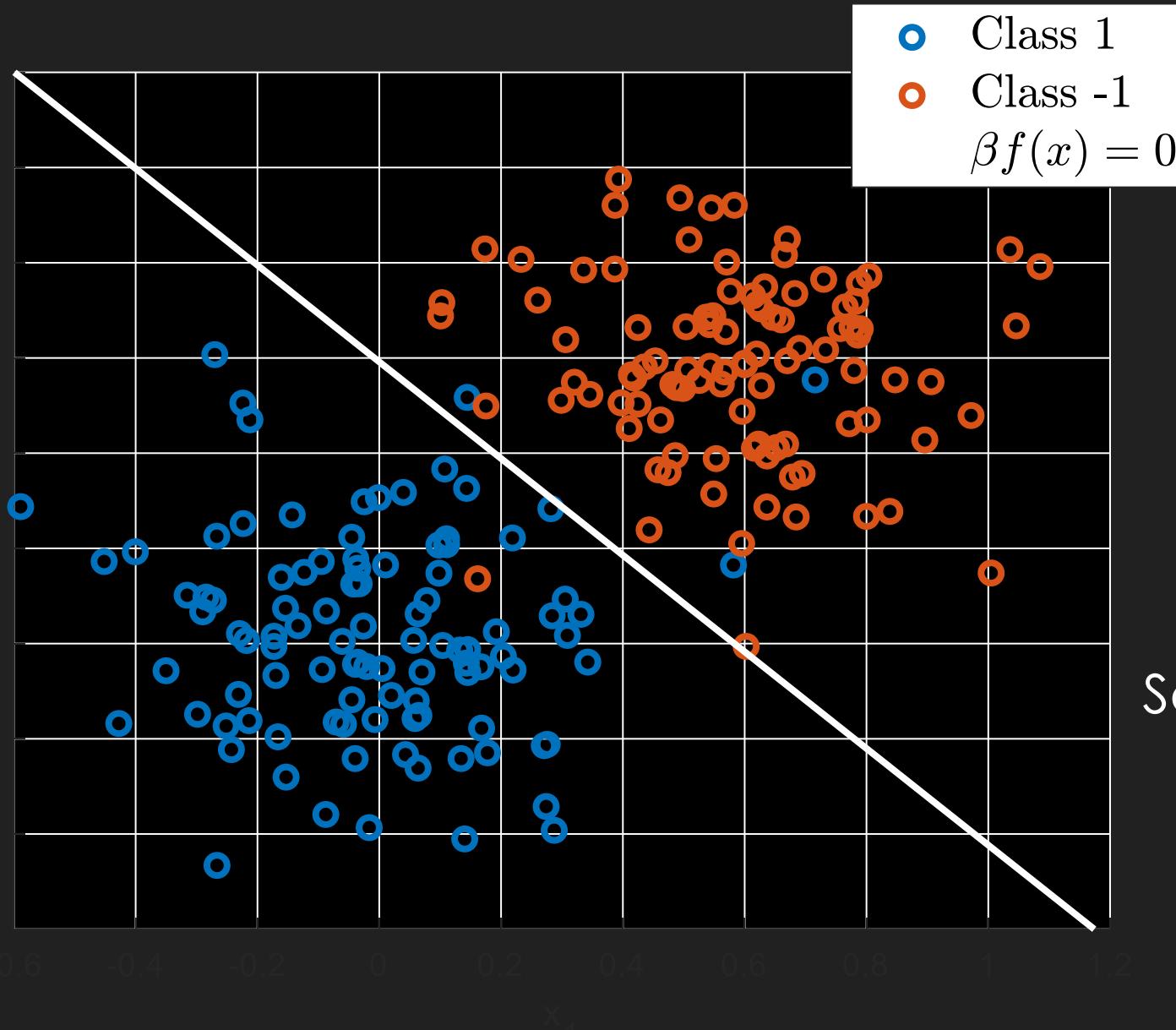
$$f(x) = x^2$$

Square error: 10

# LS Classification + Feature Transform

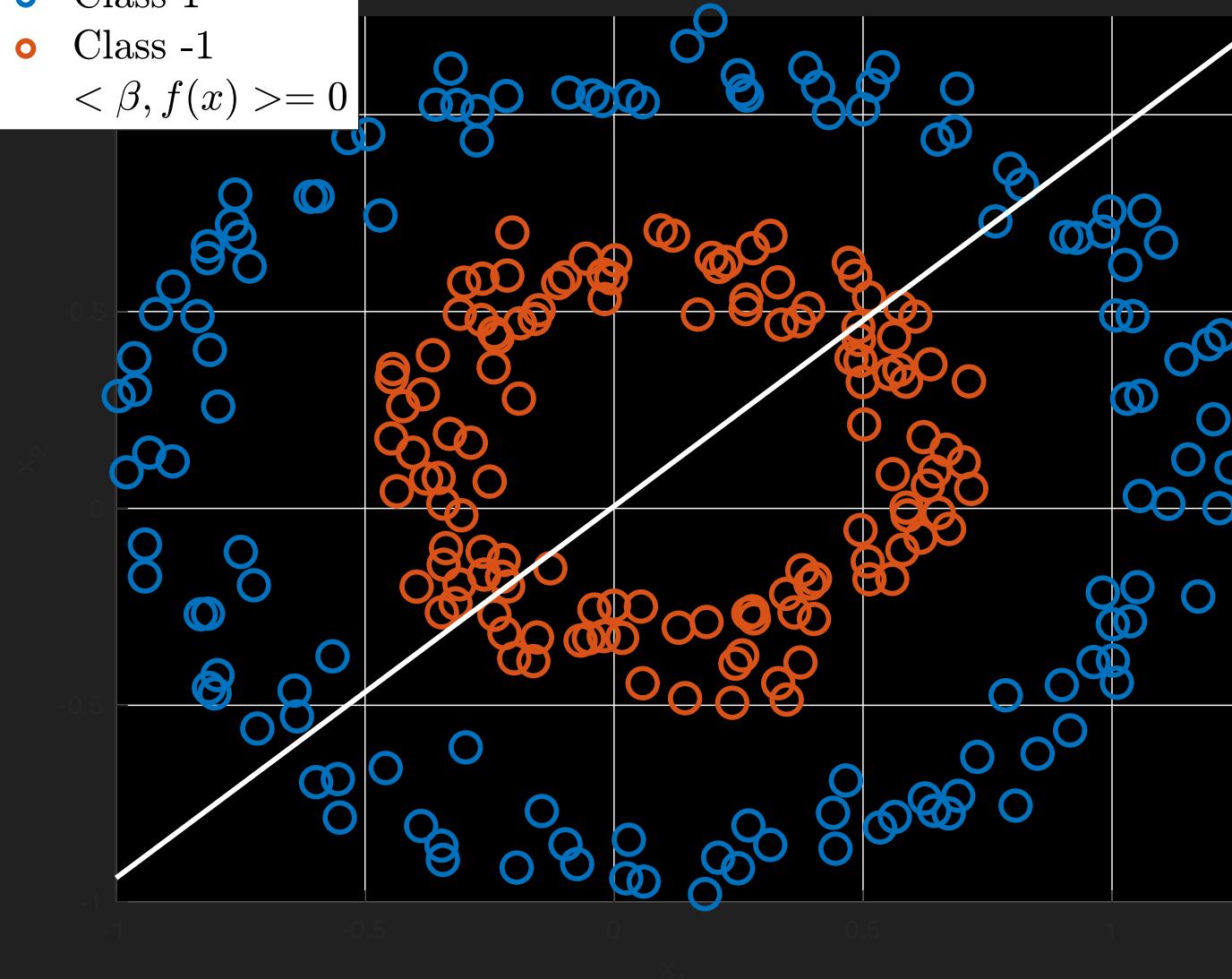
- Classification dataset:  $D = \{(y_i, x_i)\}_{i=1}^n, y \in \{-1, 1\}$ .
- Now  $y$  only takes two discrete values -1 or 1 as **class labels**.
  - If  $y_i = 1/-1$ ,  $x_i$  belongs to pos/neg class.
- Solving LS on  $D$  using feature transform  $f$ :
  - $\hat{\beta} = \arg \min_{\beta} \sum_{i=1}^n (y_i - \beta f(x_i))^2$
- $\hat{\beta}f(x) = 0$  indicates the **classification boundary**.
  - Why?
- Song Liu ([song.liu@bristol.ac.uk](mailto:song.liu@bristol.ac.uk)), Lecturer in Data Science and A.I.

# LS Classification



# LS Classification on Nonlinear Dataset

- Class 1
- Class -1
- $\langle \beta, f(x) \rangle \geq 0$



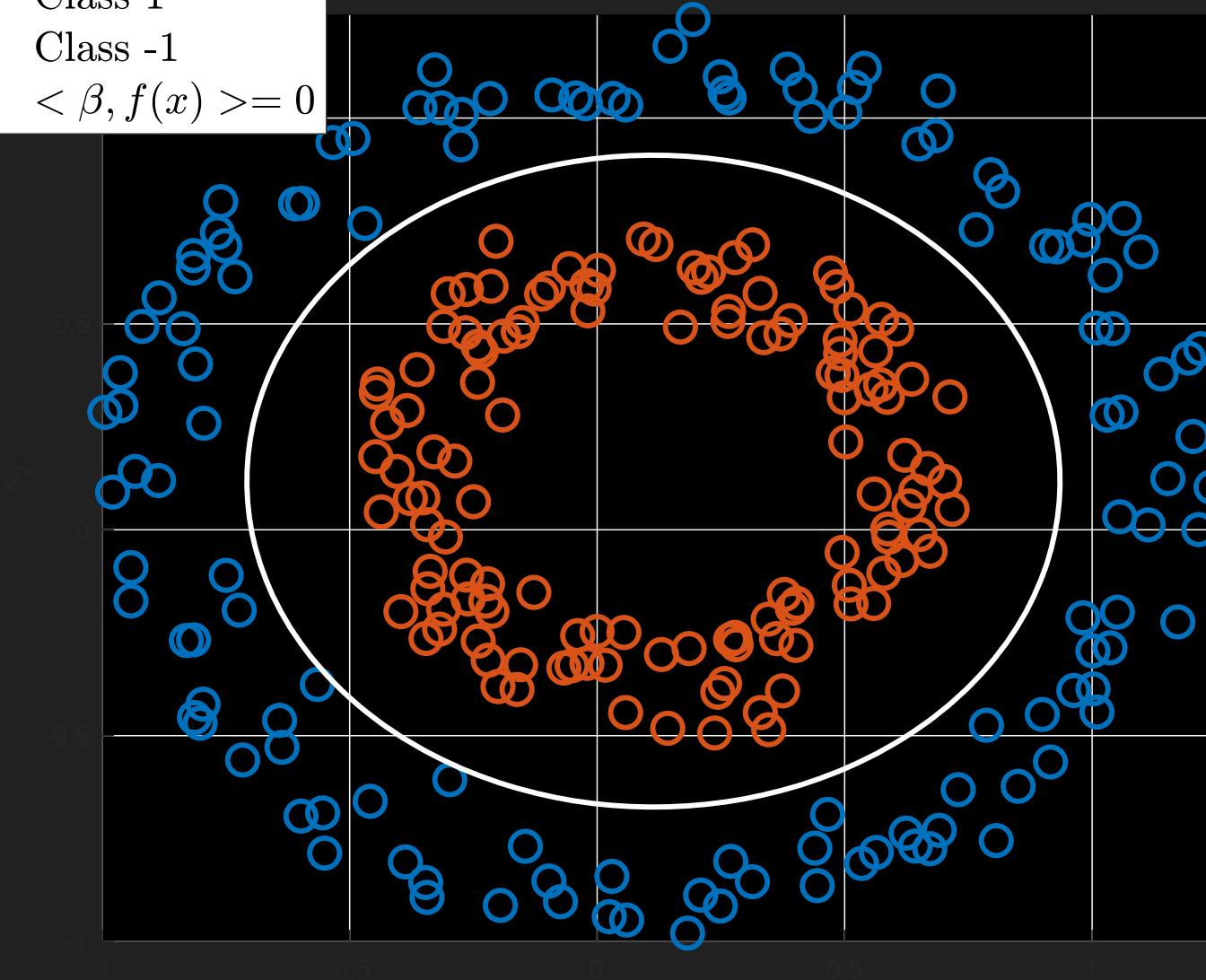
$$f(x) = x$$

Square error:

252

# LS Classification on Nonlinear Dataset

- Class 1
- Class -1
- $\langle \beta, f(x) \rangle \geq 0$



$$f(x) = x^2$$

Square error:

40



# How to construct $f$ in a more principled way?

- Song Liu ([song.liu@bristol.ac.uk](mailto:song.liu@bristol.ac.uk)), Lecturer in Data Science and A.I.

# Two schools of thoughts:

- Choosing  $f$  manually (Week 20,22)
  - **Pros:**
    - Efficient, require little computational effort.
    - Works well if you have domain knowledge.
  - **Cons:** Less flexible, requires tuning on different datasets.
- Choosing  $f$  automatically (Week 21)
  - **Pros:** Adaptive, automatically done on different datasets
  - **Cons:**
    - Extra computational burden.
    - Hard to integrate your domain knowledge.
- **Real-world problem solving involves a bit of both!!**

# A Note on Math

- Song Liu ([song.liu@bristol.ac.uk](mailto:song.liu@bristol.ac.uk)), Lecturer in Data Science and A.I.

# Math required in this part

## ○ Multivariate Linear Algebra

- COMS10003,
- Mathematical Methods for Computer Scientists

## ○ Probability and Statistics

- COMS10011
- Probability and Statistics

## ○ Refer to these units for detailed math explanation.

- Song Liu ([song.liu@bristol.ac.uk](mailto:song.liu@bristol.ac.uk)), Lecturer in Data Science and A.I.

# Formal Notations

○  $x, y, z$ , scalars,  $\mathbf{x}, \mathbf{y}, \mathbf{z}$ , vectors.

○  $\mathbf{x} \in R^d$ , vector in  $d$  dimensional real-space.

○  $x^{(i)}$ , the  $i$ -th dimension of  $\mathbf{x}$ .

○  $\mathbf{x}_i$ , the  $i$ -th data point in our dataset.

○  $f(\mathbf{x}) \in R^m$ , function takes input vector  $\mathbf{x}$  and maps it into  $m$  dimensional real space.

○  $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \in R^{b \times d}$ , **matrices**, with  $b$  rows and  $d$  columns.

○ “=” is equality, “:=” is definition.

# Polynomial Transform

- Song Liu ([song.liu@bristol.ac.uk](mailto:song.liu@bristol.ac.uk)), Lecturer in Data Science and A.I.

# A Generic Model

- We introduce a generic model.
- $\hat{y} := \langle \beta, f(\mathbf{x}) \rangle = \sum_i \beta^{(i)} f^{(i)}(\mathbf{x})$ .
- Inner product between  $\beta$  and  $f$ .
- $\hat{y}$  is linear w.r.t. parameter  $\beta$ .
- Special case:
- when  $f(x), \beta \in R, \hat{y} = \beta f(x)$ .

# Polynomial Transform

- Let  $f(x)$  be polynomial functions:
- When  $x \in R$ ,  $f(x) := [x^0, x^1, x^2, \dots, x^b]$ .
  - $b$  is called the degree of  $f$ .
  - $f(x) = [0, x, x^2]$  is called a degree 2 polynomial trans. on  $x$ .

# Polynomial Transform

- When  $\mathbf{x} \in R^d$ ,
- $f(\mathbf{x}) := [\mathbf{h}(x^{(1)}), \mathbf{h}(x^{(2)}), \dots, \mathbf{h}(x^{(d)})]$ .
- $\mathbf{h}(t) := [t^0, t^1, t^2, \dots, t^b] \in R^{b+1}$ .
- $f(\mathbf{x}) \in R^{d(b+1)}$ , which means  $\beta \in R^{d(b+1)}$ .
- PC: Write down  $f^{(i)}(\mathbf{x})$  given  $i, b$  and  $d$ .

# Polynomial Transform on Data Matrix

○  $X \in R^{n \times d}$  is data matrix with  $n$  observations and  $d$  dimensions.

○  $f(X) := \begin{bmatrix} f(\mathbf{x}_1) \\ f(\mathbf{x}_2) \\ \dots \\ f(\mathbf{x}_n) \end{bmatrix} \in R^{n \times d(b+1)}.$

○ We expanded our data matrix.  
○ from  $d$  to  $d(b + 1)$

# Pairwise Polynomial Transform

○ So far, the polynomial transform is applied on each dimension:

○ i.e.,  $f(\mathbf{x}) = [\mathbf{h}(x^{(1)}), \mathbf{h}(x^{(2)}), \dots, \mathbf{h}(x^{(d)})]$ .

○ It does **not** consider the dependencies between features.

○ Can be solved by appending cross terms i.e.,  $f(\mathbf{x}) := [\mathbf{h}(x^{(1)}), \dots, \mathbf{h}(x^{(d)}), \forall_{u < v} x^{(u)} x^{(v)} ]$

# LS Solution

$$\textcircled{O} \hat{\beta} = \arg \min \sum_{i=1}^n (y_i - \langle \beta, f(x_i) \rangle)^2$$

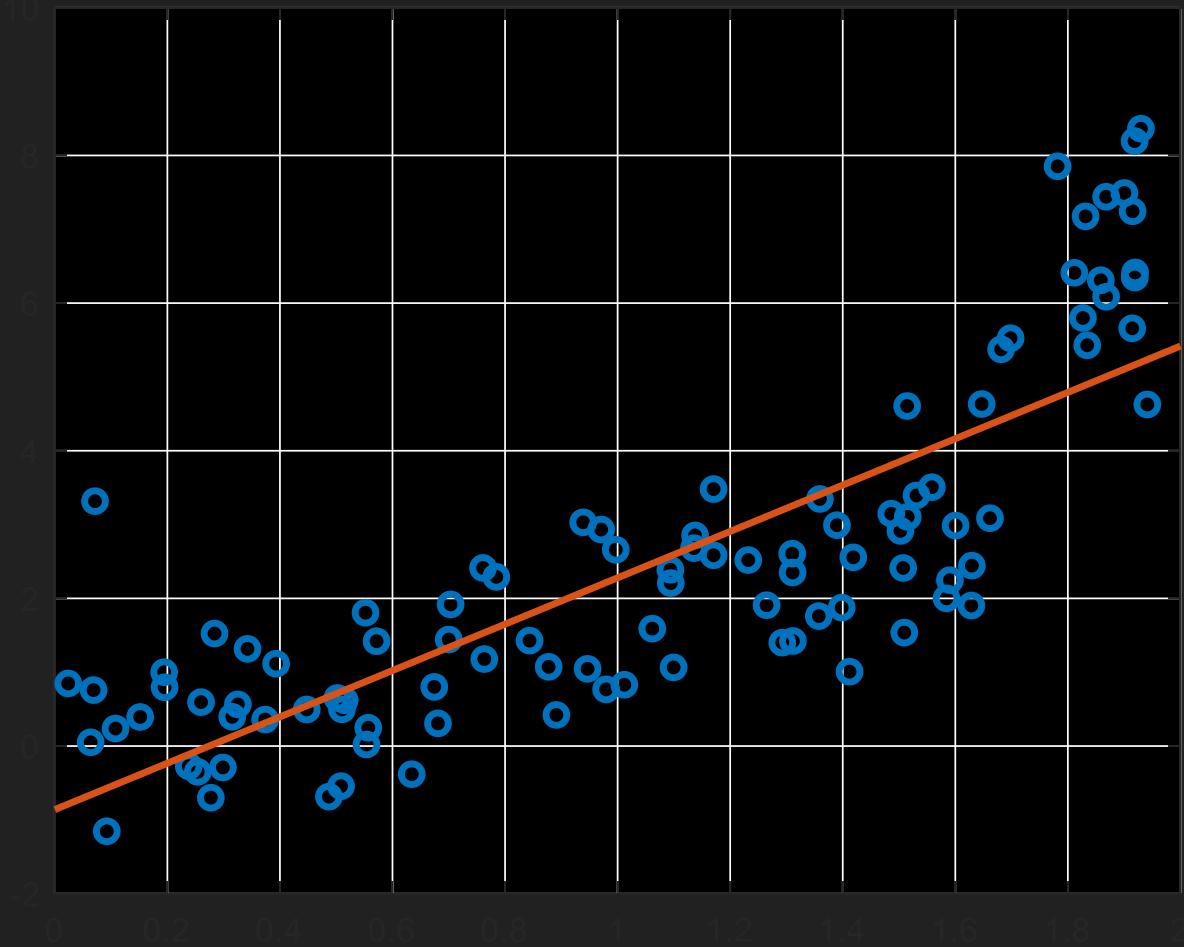
$$\textcircled{O} \hat{\beta} = (f(X)^\top f(X))^{-1} f(X)^\top y$$

- Song Liu ([song.liu@bristol.ac.uk](mailto:song.liu@bristol.ac.uk)), Lecturer in Data Science and A.I.

# Questions

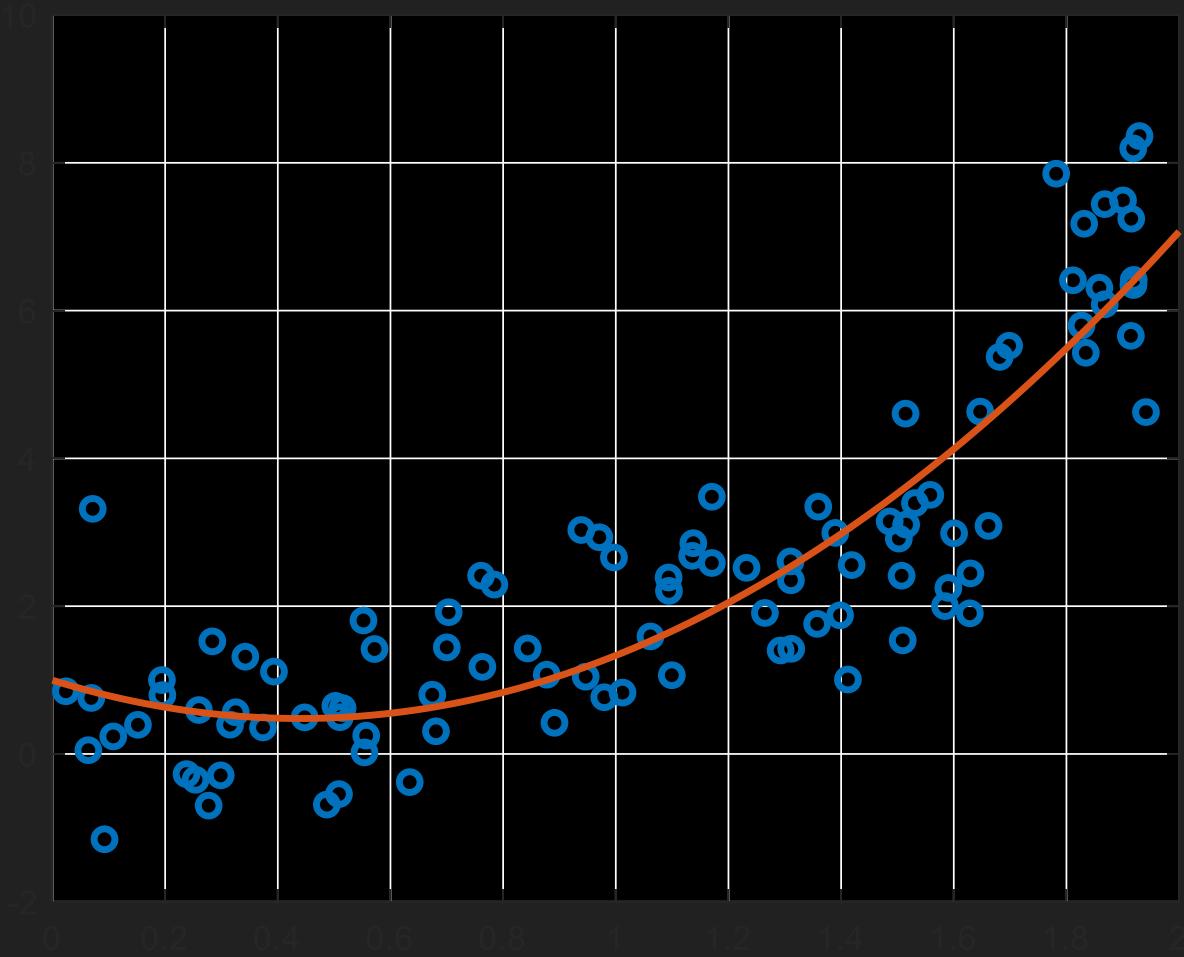
- At least, how many observations are needed to compute  $\hat{\beta}$  with  $f \in R^{d(b+1)}$  using the formula above?
- <https://pollev.com/songliu644>
- OPC: what is the computational complexity?

**Example:**  $y = \exp(1.5x - 1) + \epsilon$ ,  
 $\epsilon \sim N(0,1)$



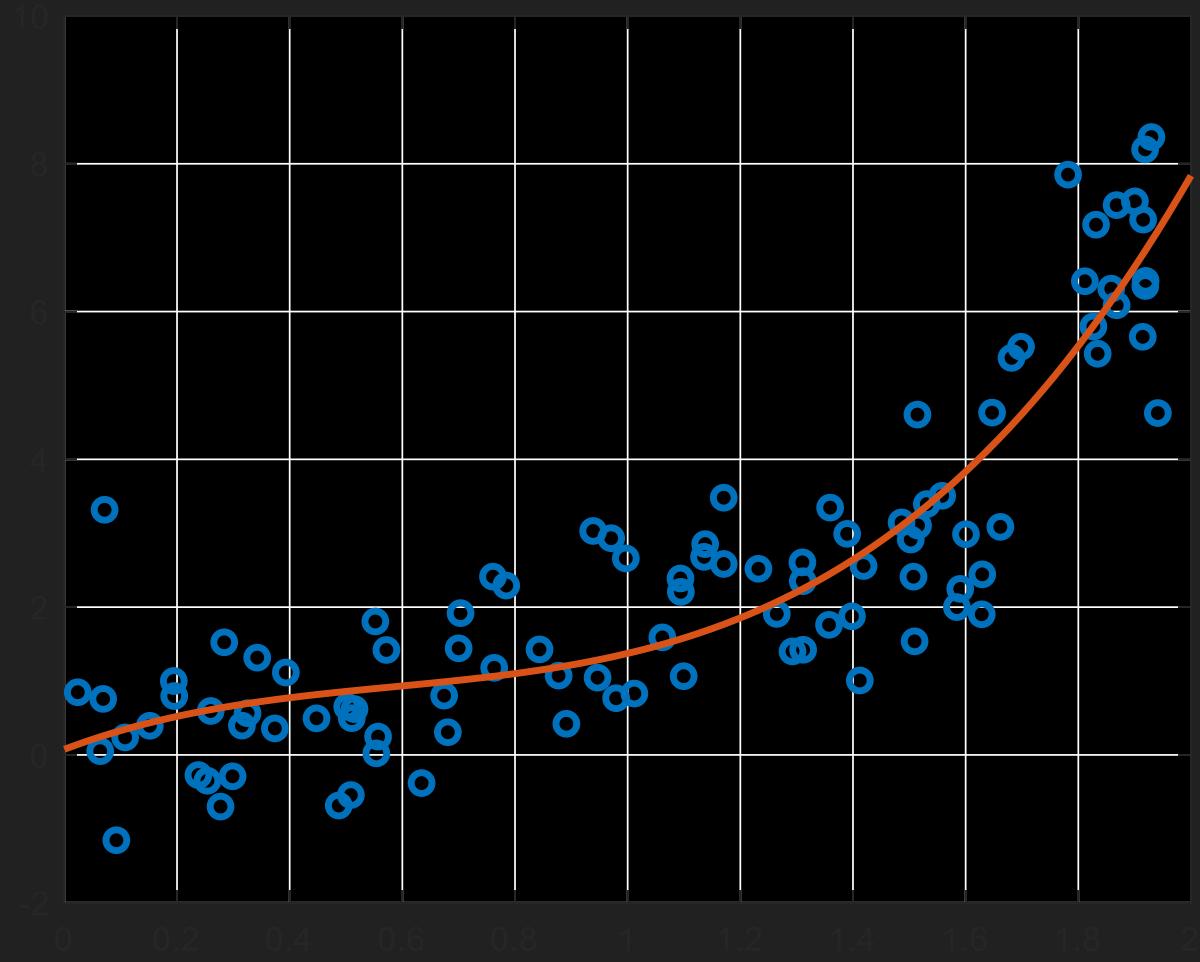
- Polynomial transform with  $b = 1$ .
- Square error: 171.0

**Example:**  $y = \exp(1.5x - 1) + \epsilon$ ,  
 $\epsilon \sim N(0,1)$



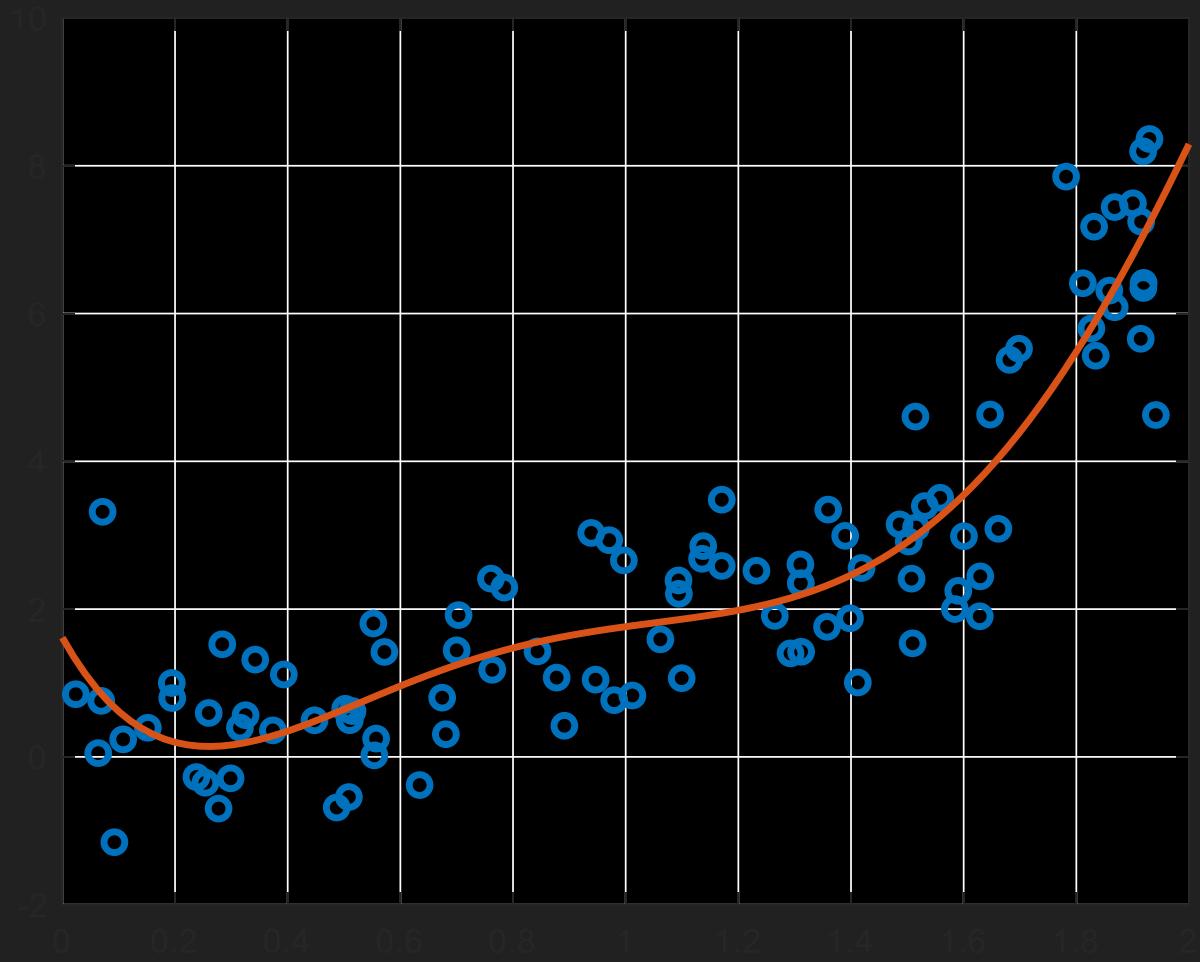
- Polynomial transform with  $b = 2$ .
- Square error: 108.97

**Example:**  $y = \exp(1.5x - 1) + \epsilon$ ,  
 $\epsilon \sim N(0,1)$



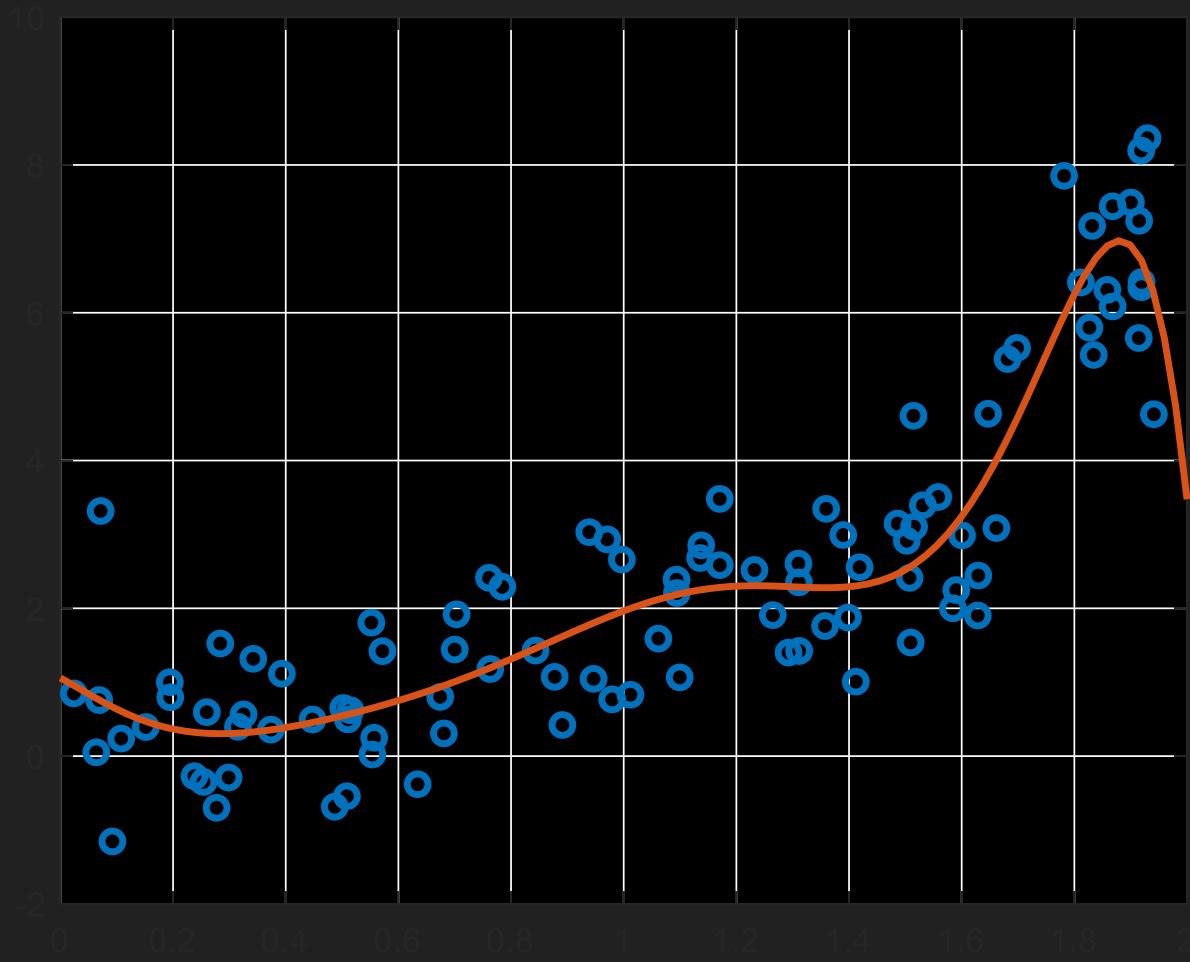
- Polynomial transform with  $b = 3$ .
- Square error: 99.618

**Example:**  $y = \exp(1.5x - 1) + \epsilon$ ,  
 $\epsilon \sim N(0,1)$



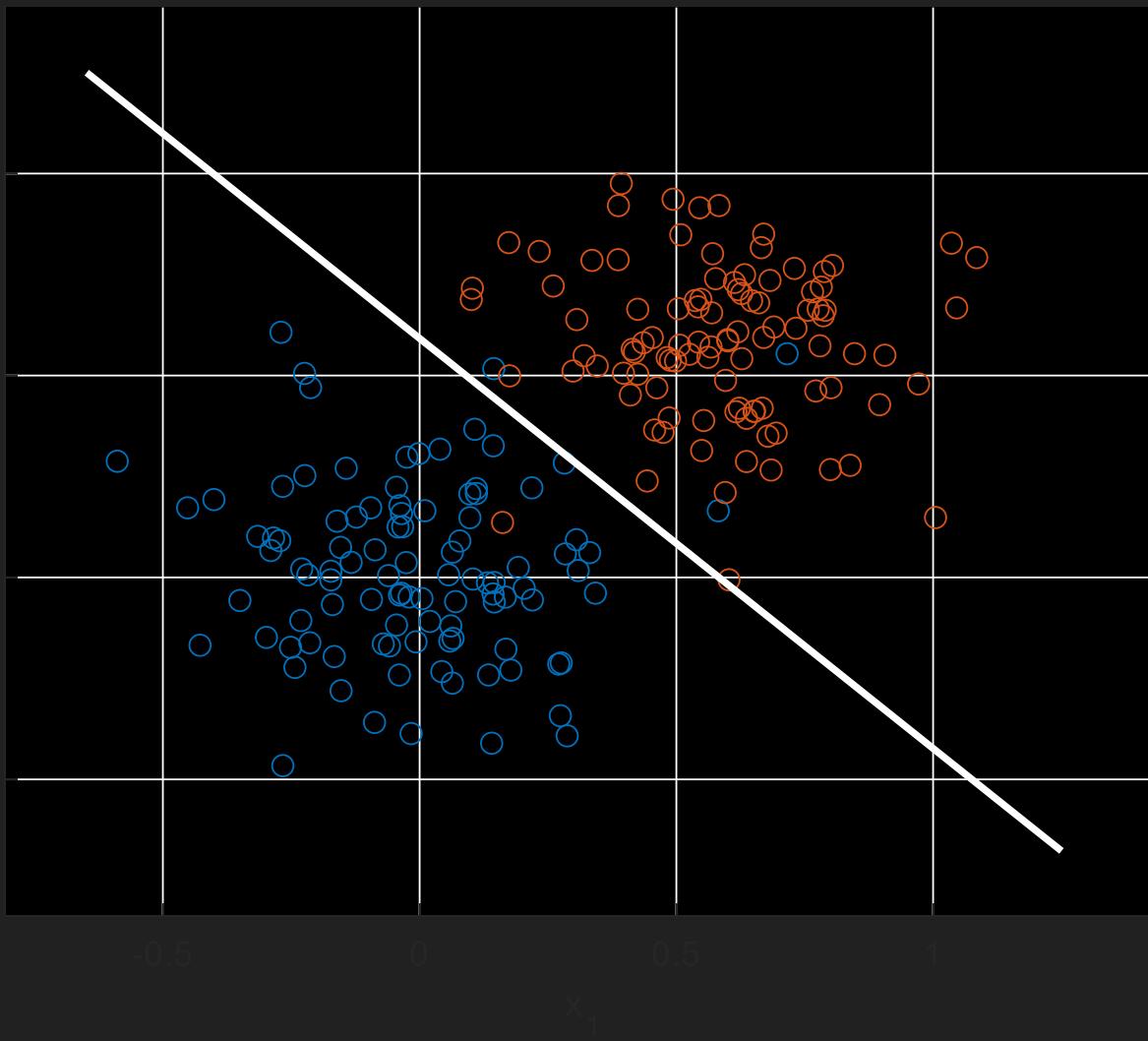
○ Polynomial transform with  $b = 5$ .  
○ Square error: 89.378

**Example:**  $y = \exp(1.5x - 1) + \epsilon$ ,  
 $\epsilon \sim N(0,1)$



○ Polynomial transform with  $b = 8$ .  
○ Square error: 78.87

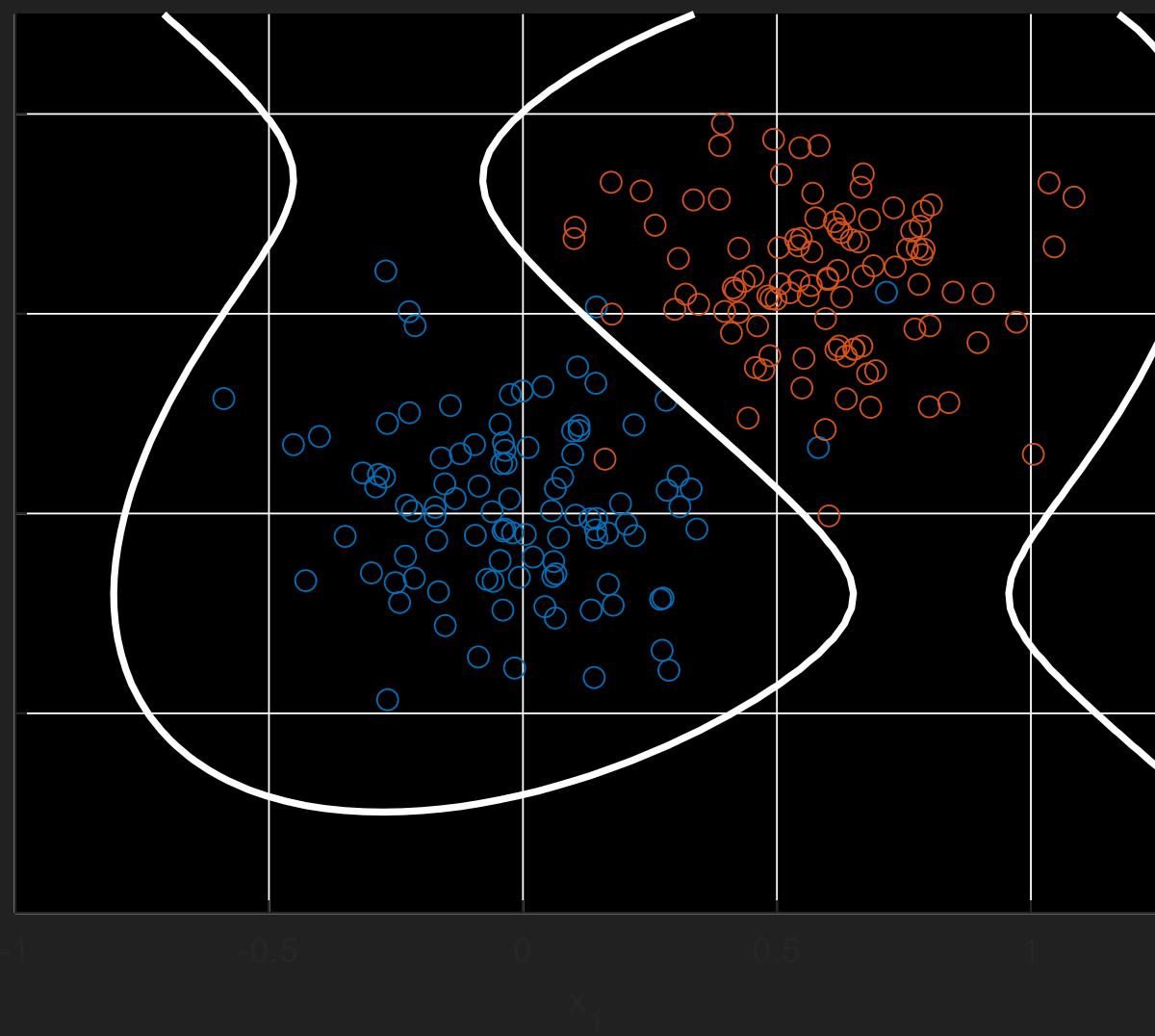
# Example: Binary Classification



○ Polynomial transform with  $b = 1$ .

○ Square Error:  
39.0547

# Example: Binary Classification



○ Polynomial  
transform  
with  $b = 3$ .

○ Square  
Error:  
32.0632

# Observations:

- Pay attention on
- how square error keeps **dropping** when **increasing** degree  $b$ .
- how  $\hat{y}$  becomes more **flexible** when **increasing**  $b$ .
- We will revisit this point in the next lecture.

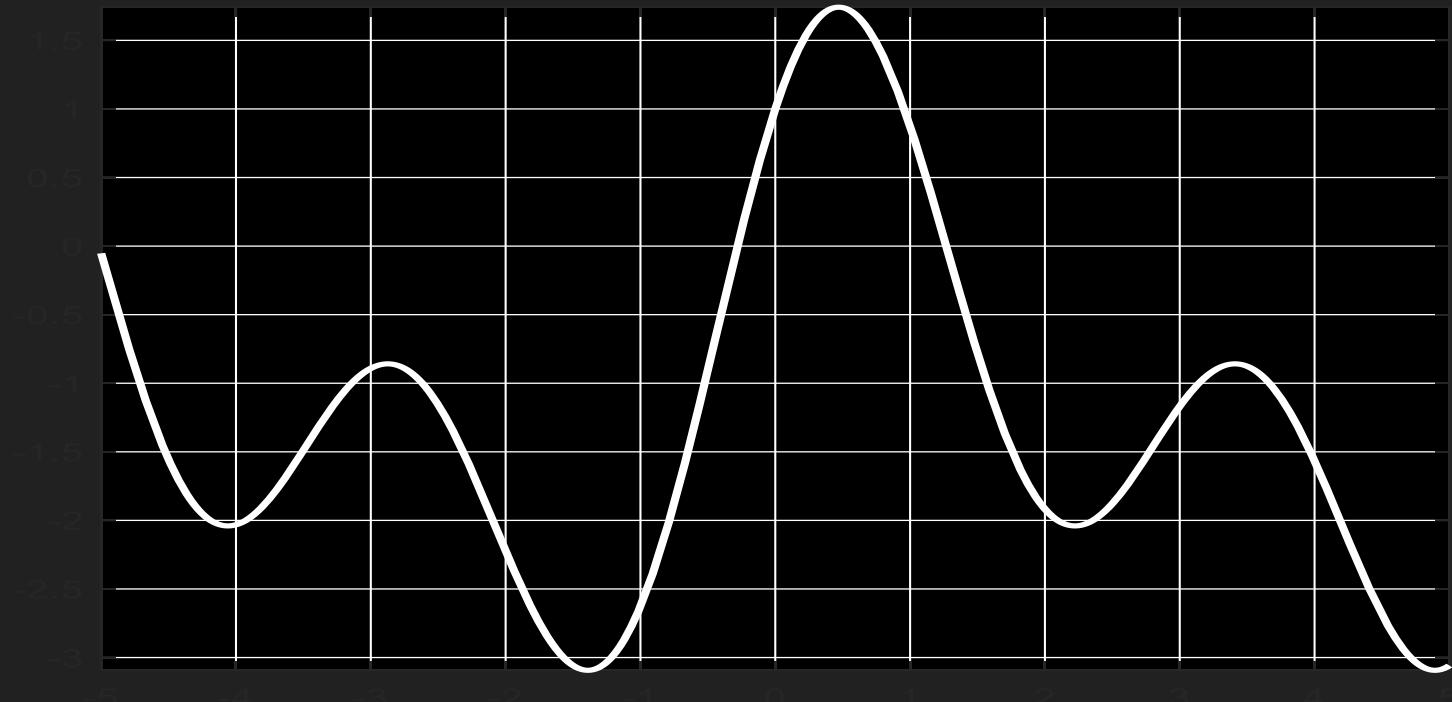
# Why it works?

- 1-dimensional intuition: Taylor Series.
- Taylor Series of  $g(x)$  at 0:
  - $$g(x) = g(0)(x - 0)^0 + g'(0)(x - 0)^1 + \frac{g''(0)}{2!}(x - 0)^2 + \frac{g'''(0)}{3!}(x - 0)^3 + \dots$$
  - You can approximate a **smooth** function using polynomial terms (at some cost).

# Fourier Series

- What are **other ways** of decomposing a function?
- Suppose we have a periodic signal  $g(x)$  over the time domain.
  - e.g. a sound wave or a stock price
  - $$g(x) = a_0 + \sum_{i=1}^{\infty} [a_i \sin(ix) + b_i \cos(ix)]$$
  - This decomposition is called Fourier Series.

# Fourier Series



O  $g(x) = \sin(x) + \cos(x) + \sin(2x) + \cos(2x)$

- Song Liu ([song.liu@bristol.ac.uk](mailto:song.liu@bristol.ac.uk)), Lecturer in Data Science and A.I.

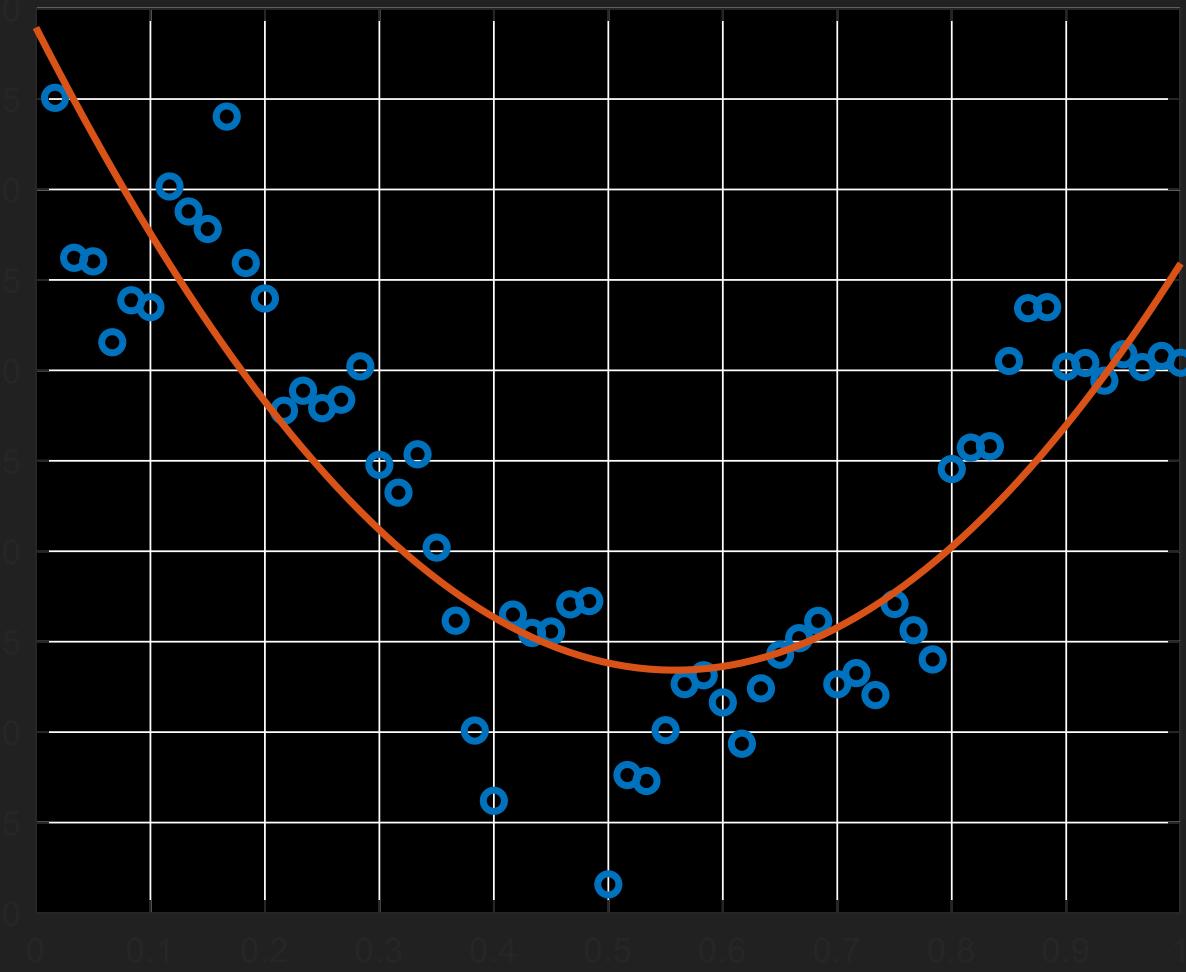
# Trigonometric Transform

○ Trigonometric Transform are used to approximate function over **time domain**.

○  $f(x) := [1, \sin(x), \cos(x), \sin(2x), \cos(2x) \dots \sin(bx), \cos(bx)]$

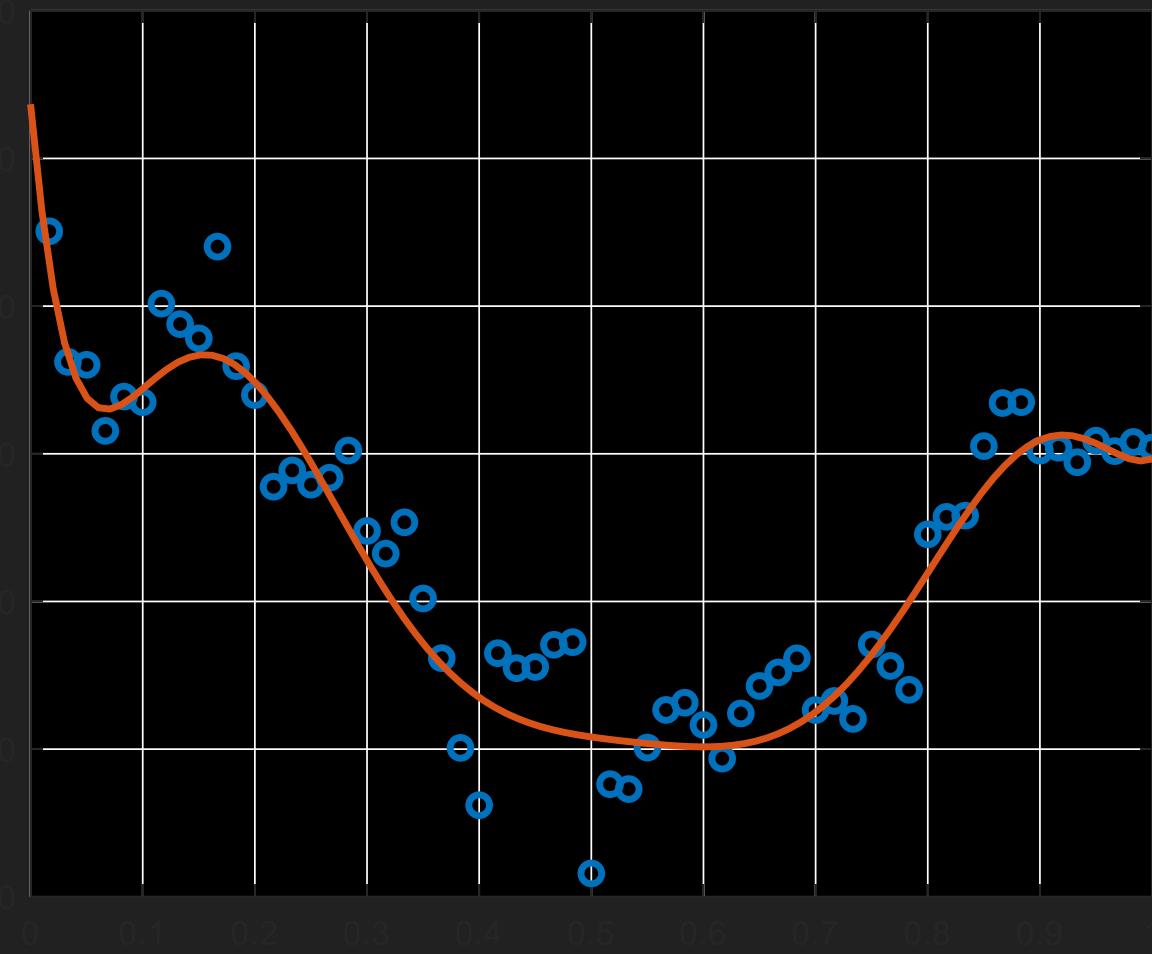
○  $f(x) \in R^{2b+1}$

# Example: Apple Stock Price, Feb 2019



○ Trigonometric transform with  $b = 1$ .  
○ Squared error:  
 $1.5681 \times 10^3$

# Example: Apple Stock Price, Feb 2019



- Trigonometric transform with  $b = 4$ .
- Squared error: 699.9117

# Linear Expansion of Basis Functions

○ Polynomial and Trigonometric transforms based on the idea a function can be approximated by:

$$y \approx \hat{y} = \sum_{i=1}^m \beta^{(i)} f^{(i)}(x)$$

○ called a linear basis expansion of  $y$

○  $f^{(i)}$  are called **basis function**

○ Polynomial basis, Trigonometric basis...

# Radial Basis Function (RBF)

○ RBF is another widely used basis function for function approximation.

○  $f^{(i)}(x) := \exp\left(-\frac{\|x-x_i\|^2}{\sigma^2}\right)$

○  $\sigma > 0$  is called width

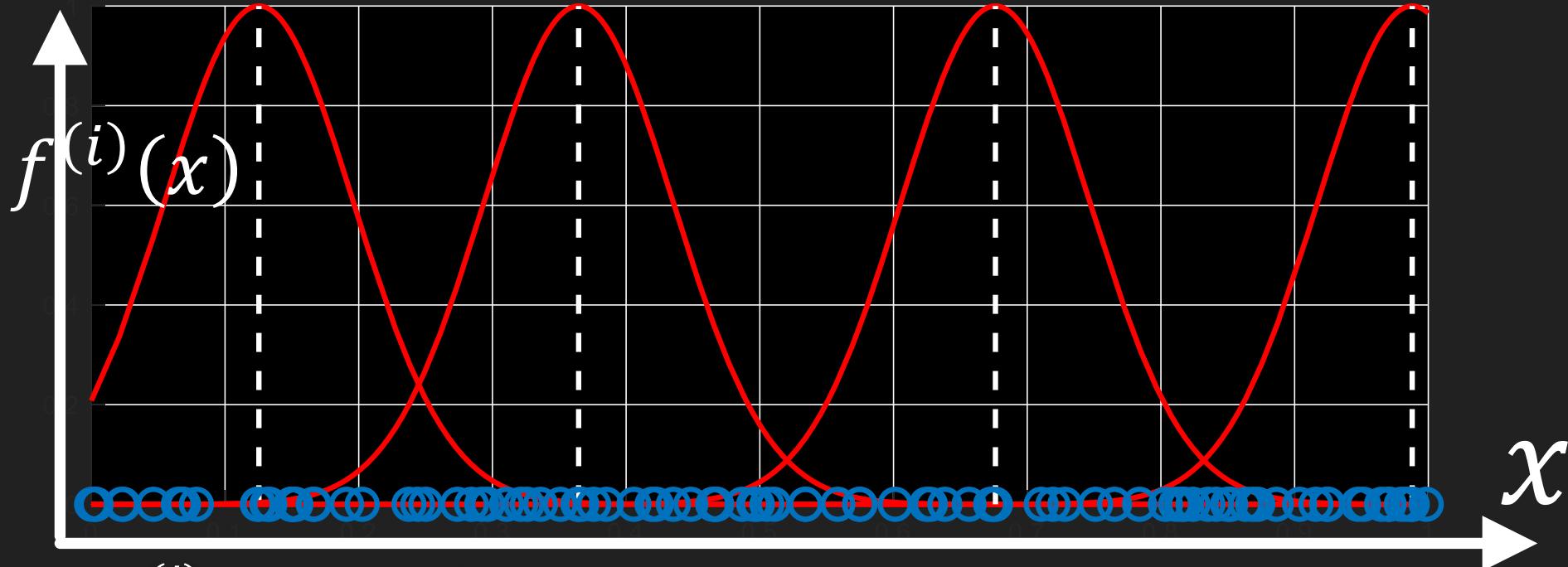
○  $\sigma$  is determined **before** fitting

○ A practice is setting  $\sigma$  as the median of all pairwise distances of  $x$  in your data.

# Radial Basis Function (RBF)

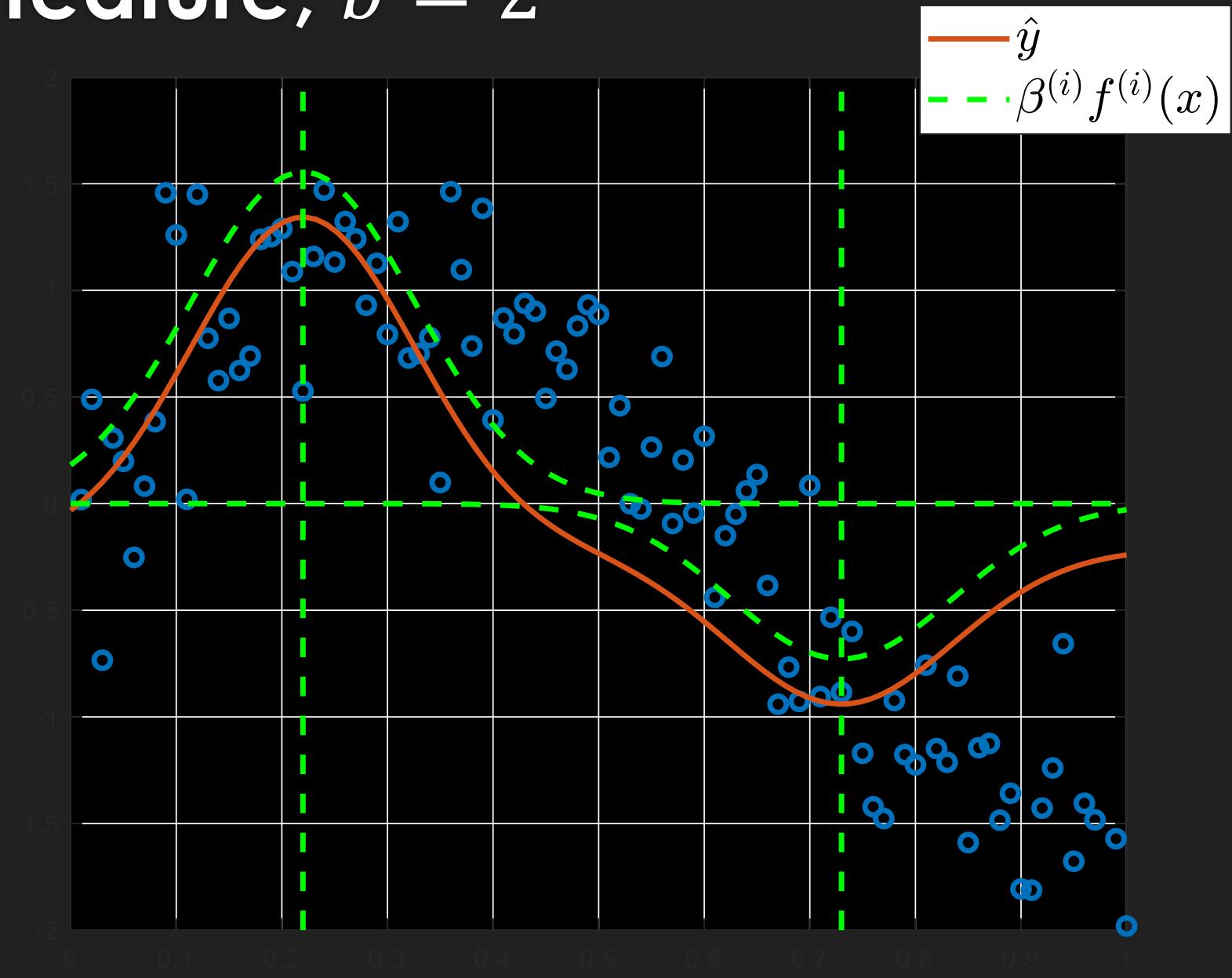
- $x_i$  are called **RBF centroids**.
- $x_i$  can be **randomly chosen** from the  $x$  in your dataset
- $f(x) := [1, f^{(1)}(x), f^{(2)}(x), \dots, f^{(b)}(x)]$
- Do not forget 1!

# Radial Basis Function (RBF)

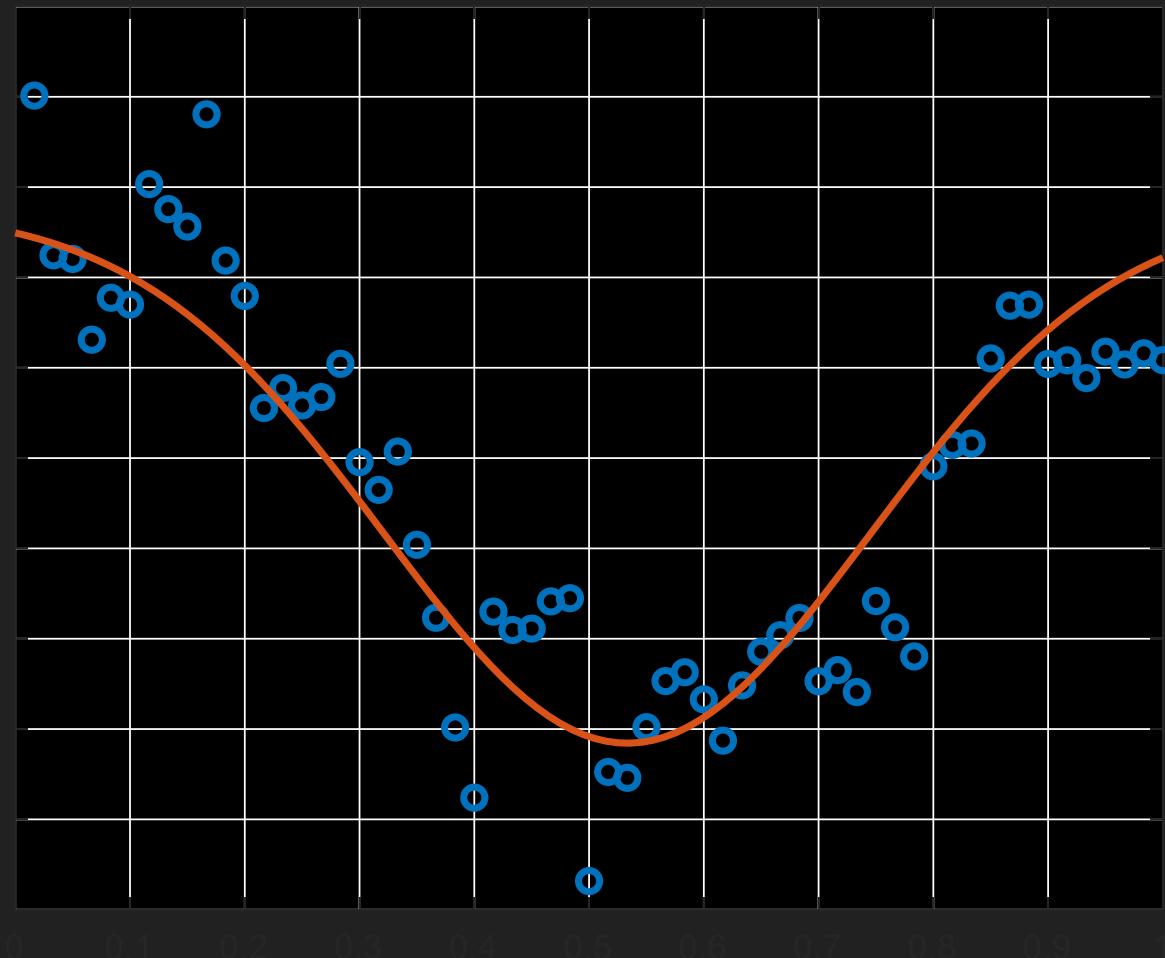


- $f^{(i)}(x)$  are visualized in red at random 4 centroids among 100 uniformly drawn  $x$ .
- At each “bump”,
  - If  $\beta^{(i)} > 0$ , basis at  $x^{(i)}$  gives  $\hat{y}$  a “lift”.
  - If  $\beta^{(i)} < 0$ , basis at  $x^{(i)}$  gives  $\hat{y}$  a “push”.

# RBF feature, $b = 2$



# Example: Apple Stock Price, Feb 2019



ORBF

$\sigma = 0.2121$

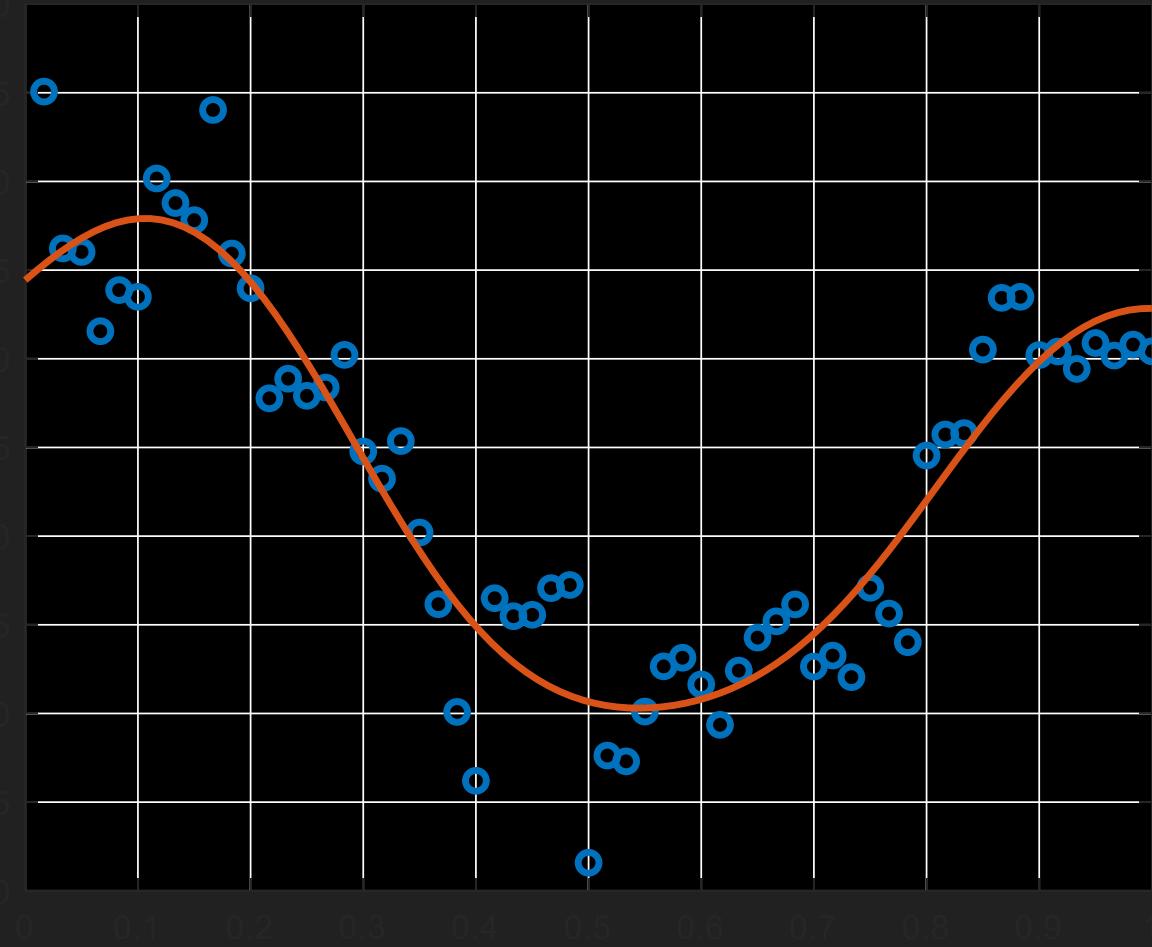
$b = 1.$

Squared

error:

$1.1908e+03$

# Example: Apple Stock Price, Feb 2019



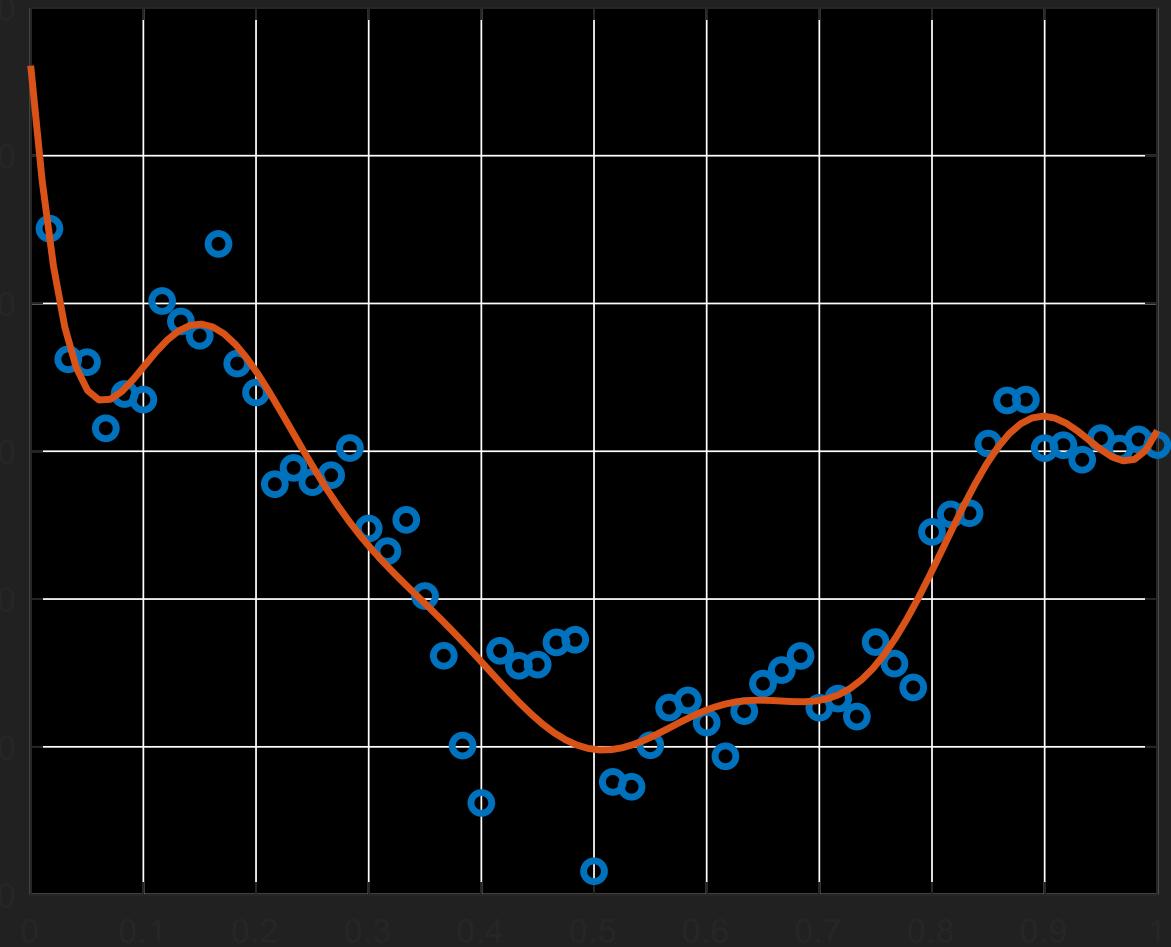
ORBF

$\sigma = 0.2121$

$b = 5.$

Squared  
error:  
842.7575

# Example: Apple Stock Price, Feb 2019



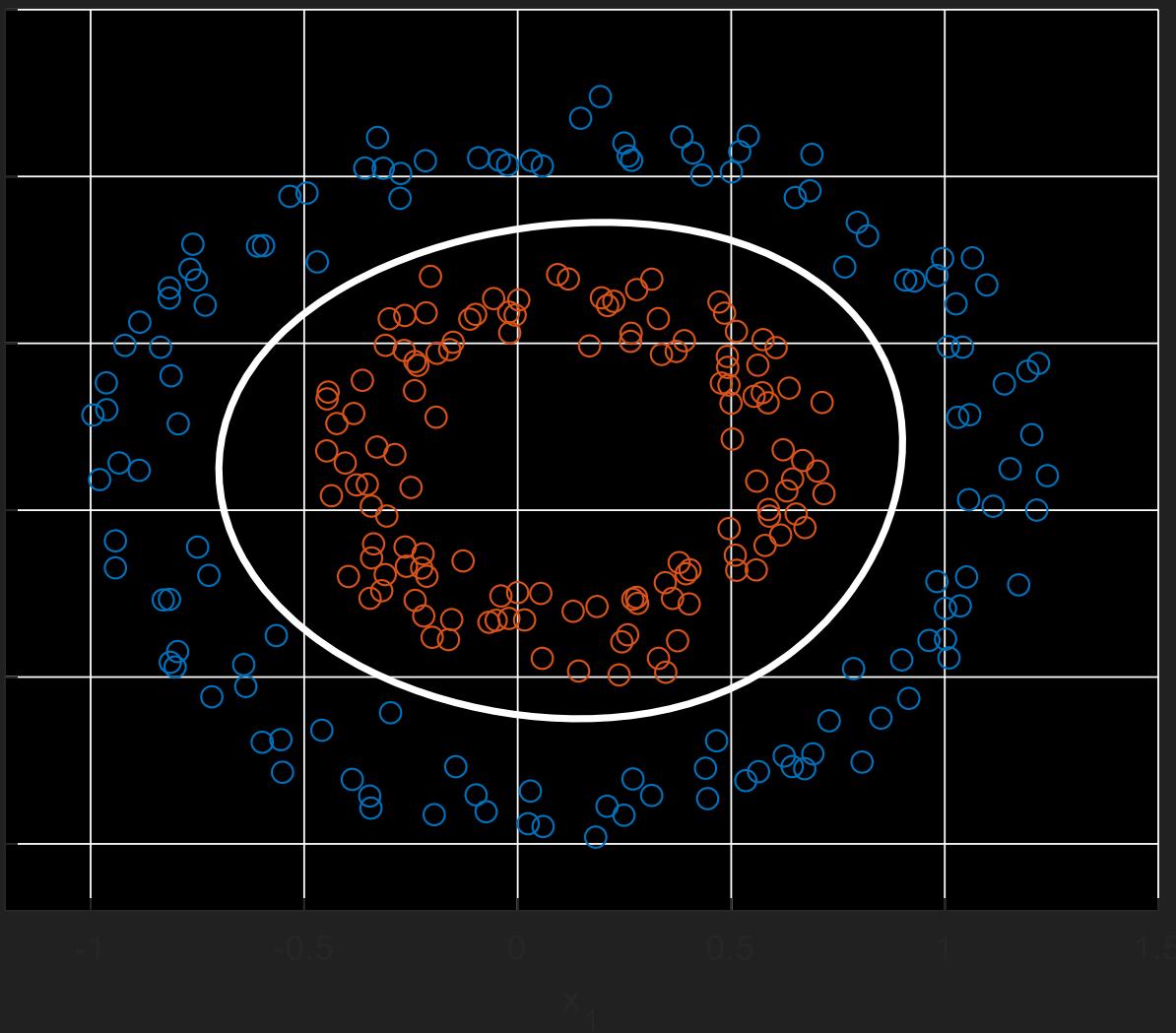
ORBF

$$\sigma = 0.2121$$

$$b = 10.$$

Squared  
error:  
593.8104

# Example: Double Ring Classification



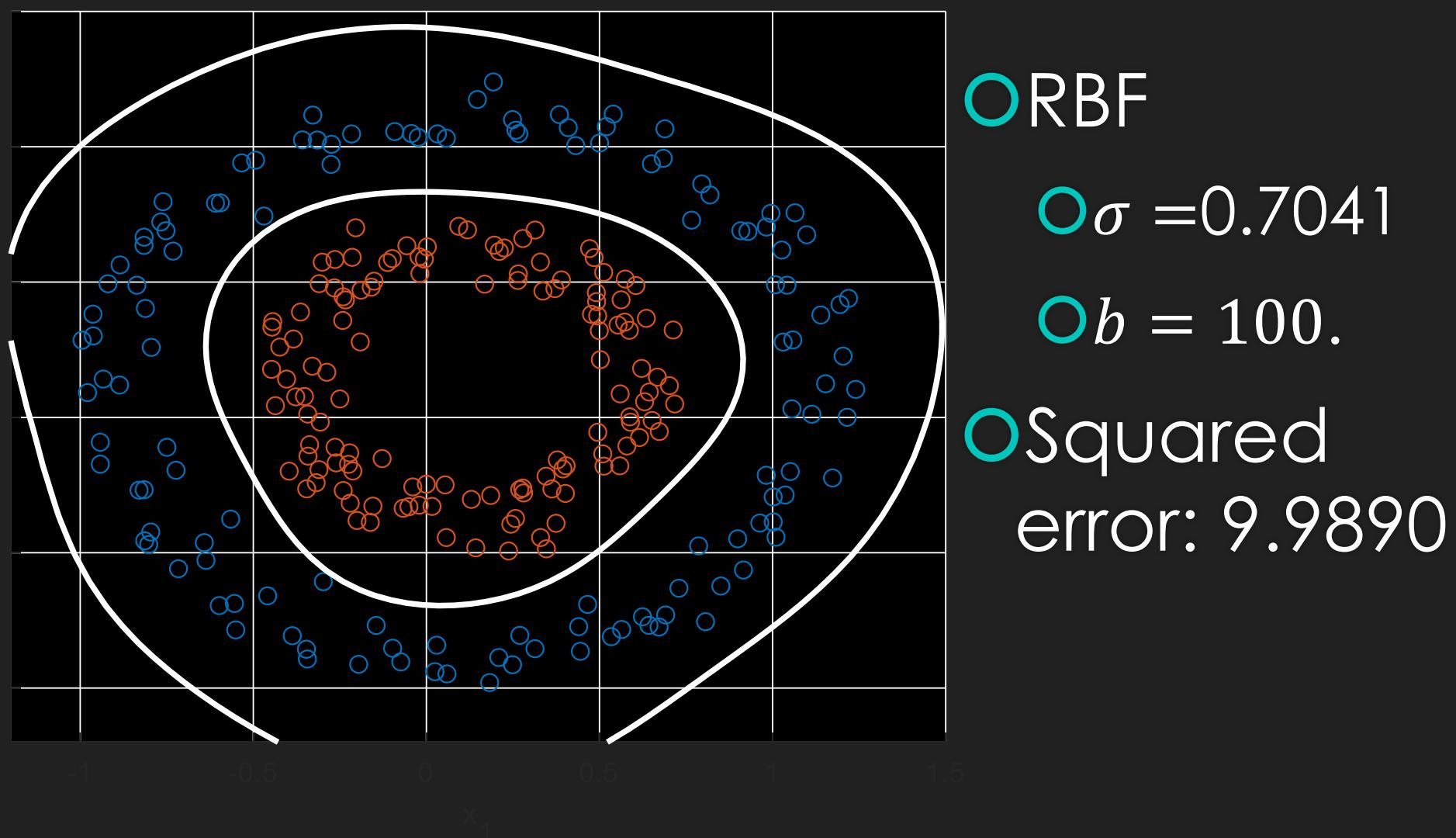
ORBF

$\sigma = 0.7041$

$b = 5.$

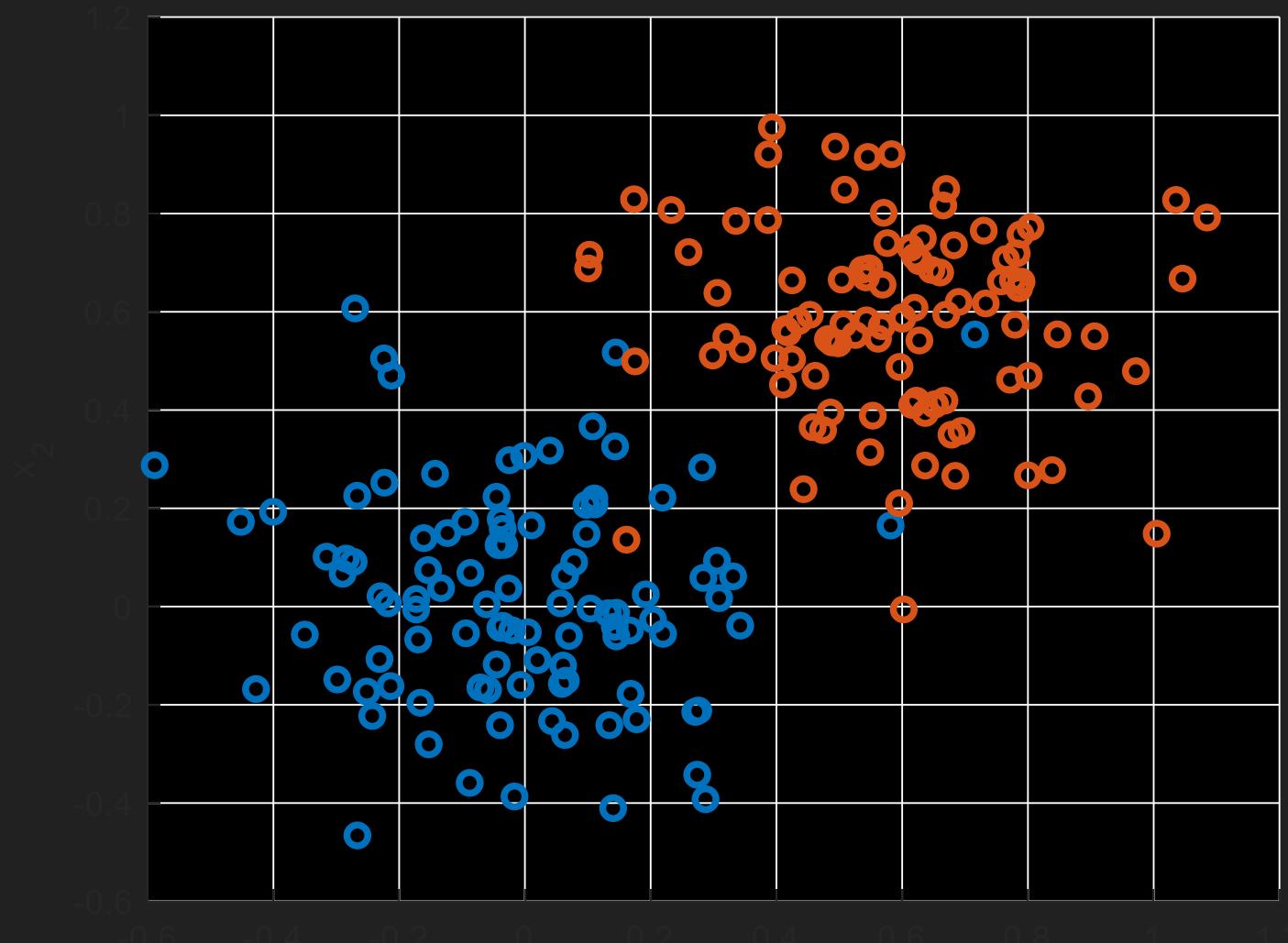
Squared  
error:  
16.3351

# Example: Double Ring Classification



- Song Liu ([song.liu@bristol.ac.uk](mailto:song.liu@bristol.ac.uk)), Lecturer in Data Science and A.I.

# Selecting Features using Prior Knowledge



- Song Liu ([song.liu@bristol.ac.uk](mailto:song.liu@bristol.ac.uk)), Lecturer in Data Science and A.I.

# Question

- Seeing your dataset above, what  $f$  should you use for classification? Hint: consider computational cost and overfitting
  - Polynomial,  $b = 1$
  - Polynomial,  $b = 2$
  - Polynomial,  $b = 3$
  - ORBF,  $b = 100$
- <https://bit.ly/2FnjryC>

# Conclusion

- Feature transform can be crucial to regression and classification tasks.
- Three useful feature transform:
  - Polynomial
  - Trigonometric (on time series)
  - RBF
- As  $b$  increases,  $\hat{y}$  become more flexible, squared error is lowered.

# Unanswered Questions

- Increasing  $b$  drops squared error.
- How do you select number of basis  $b$ ?
- Knowing an  $f$  with a larger  $b$  makes  $\hat{y}$  more flexible, can we make  $b = \infty$ ?
- Next two lectures, The selection of number of basis  $b$  and Kernel methods.

# To know more...

- The Elements of Statistical Learning:  
Data Mining, Inference, and  
Prediction, Hastie et al., 2009
- 2.3.1 Linear Models and Least Squares
- 2.6.3 Function Approximation
- 2.8.3 Basis Functions

- Song Liu ([song.liu@bristol.ac.uk](mailto:song.liu@bristol.ac.uk)), Lecturer in Data Science and A.I.