04

A Basic Framework for Supervised Learning: Regression Problem

THE HITSZ
SCHOOL OF ECONOMICS
AND MANAGEMENT

What do we have?

Continuous variable

(Training) Data:

	第 1 列	第 2 列	第 3 列	… …	第 k 列	第 k+1 列
	市值规模	毛利率	账面市值比	… …	财务杠杆率	股票收益率
第 1 行	5.312	0.201	0.149	… …	0.498	0.127
第 2 行	7.801	0.173	0.203	… …	0.203	0.042
第 3 行	4.112	0.598	0.081	… …	0.637	0.214
第 4 行	4.981	0.301	0.111	… …	0.301	0.147
⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮
第 N 行	6.892	0.151	0.398	… …	0.502	-0.018

- N data points. k features.
- For data point $i \in \{1, 2, \dots, N\}$
  - Feature vector: $\boldsymbol{x_i} = (x_{i,1}, x_{i,2}, \dots, x_{i,k}) \in R^k$
  - Label: y_i
  - Data point i: $(\boldsymbol{x_i}, y_i)$
- Training dataset: $D_N = \{(\boldsymbol{x_1}, y_1), (\boldsymbol{x_2}, y_2), \dots, (\boldsymbol{x_N}, y_N)\} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^{N}$

What do we want?

市值规模	毛利率	账面市值比	… …	财务杠杆率	股票收益率
4.115	0.341	0.219	… …	0.128	?

How to label?

$$\boldsymbol{x} \longrightarrow \boxed{f} \longrightarrow \hat{y}$$

Example: f = 市值规模 + 0.1×毛利率 + 2×账面市值比 − 财务杠杆率

What do we have in this example?

Variables   Parameters   Function form

Machine learning model is essentially a function that links the input (feature vector) $\boldsymbol{x}$ with predicted output (label) $\hat{y}$.

- Input (feature vector): $\boldsymbol{x} = \left(x^{(1)},\ x^{(2)},\ \ldots\ ,\ x^{(d)}\right)^{\mathrm{T}} \in \boldsymbol{R}^{d}$
- Output (predicted label): $\hat{y} = f(\boldsymbol{x}; \boldsymbol{\beta})$

Different Types of Machine Learning Models

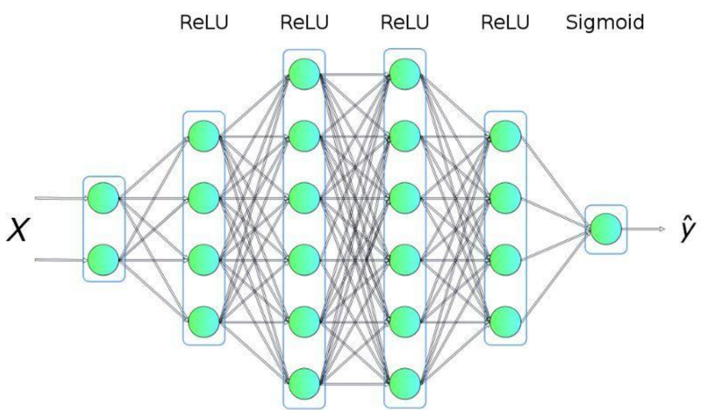
- **Linear Model** $f(\boldsymbol{x};\boldsymbol{\beta}) = \beta_1 x^{(1)} + \beta_2 x^{(2)} + \cdots + \beta_d x^{(d)} = \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}$
- **Polynomial Model**

$$f(\boldsymbol{x};\boldsymbol{\beta}) = \mu_1 x^{(1)} + \mu_2 x^{(2)} + \cdots + \mu_d x^{(d)} +$$
$$a_{11}x^{(1)}x^{(1)} + a_{12}x^{(1)}x^{(2)} + \cdots + a_{1d}x^{(1)}x^{(d)} +$$
$$a_{12}x^{(2)}x^{(1)} + a_{22}x^{(2)}x^{(2)} + \cdots + a_{2d}x^{(2)}x^{(d)} +$$
$$\cdots$$
$$a_{d1}x^{(d)}x^{(1)} + a_{d2}x^{(d)}x^{(2)} + \cdots + a_{dd}x^{(d)}x^{(d)}$$
$$= \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}$$

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_d \end{pmatrix}; \boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1d} \\ a_{12} & a_{22} & \cdots & a_{2d} \\ \vdots & & \ddots & \vdots \\ a_{d1} & a_{d2} & \cdots & a_{dd} \end{pmatrix}$$

$$\boldsymbol{\beta} = (\boldsymbol{\mu}, \boldsymbol{A}), \boldsymbol{\mu} \in \boldsymbol{R}^d, \boldsymbol{A} \in \boldsymbol{R}^d \times \boldsymbol{R}^d$$

- **Neural Networks**

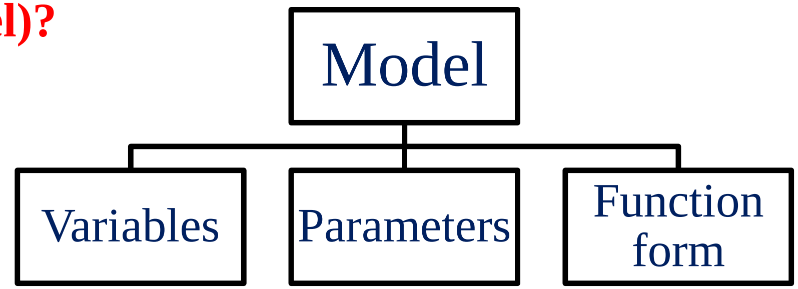


What is a good model?

市值规模	毛利率	账面市值比	… …	财务杠杆率	股票收益率
4.115	0.341	0.219	… …	0.128	?

Essentially, we want a good way (model) to predict the label of new data points.

Good Way (Model)?



Given variables (inputs) and function form, how could we derive a good model ($\boldsymbol{f}$)?

$D_N \longrightarrow$ ***Algorithm*** $\longrightarrow$ Parameters of $\boldsymbol{f}$

↓

What is the objective of this algorithm?

↓

Minimize the distance between **the predicted value of label** and **the actual value of label**.

Define Distance: Loss Function

Loss fuction defines the distance between **the predicted value of label** and **the actual value of label**.

Residual or Error

$$L_1(\boldsymbol{x}, y) = |y - \hat{y}| = |y - f(\boldsymbol{x}; \boldsymbol{\beta})|$$

$$L_2(\boldsymbol{x}, y) = [y - \hat{y}]^2 = [y - f(\boldsymbol{x}; \boldsymbol{\beta})]^2$$

- For data point $i \in \{1, 2, \ldots, N\}$
  - Loss: $L(\boldsymbol{x_i}, y_i) = [y_i - f(\boldsymbol{x_i}; \boldsymbol{\beta})]^2$
- The sum of losses for all the data points in D_N (Risk Fuction): $\sum_{i=1}^{N}[y_i - f(\boldsymbol{x_i}; \boldsymbol{\beta})]^2$

$$\mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^{N}[y_i - f(\boldsymbol{x_i}; \boldsymbol{\beta})]^2$$

Model Training Summary

We want a good way (model) to predict the label.

$\Downarrow$

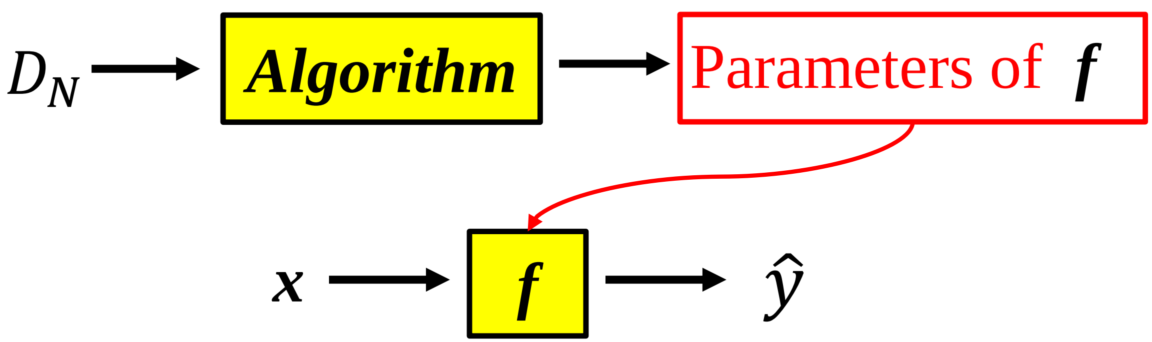
Minimize the distince between **the predicted value of label** and **the actual value of label**.

$\Downarrow$

Minimize the sum of losses for all the data points in D_N:

$\Downarrow$

$$\boxed{\hat{\boldsymbol{\beta}} = \operatorname{argmin} \mathcal{L}(\boldsymbol{\beta}) = \operatorname{argmin} \sum_{i=1}^{N} [y_i - f(\boldsymbol{x}_i; \boldsymbol{\beta})]^2} \quad ?$$

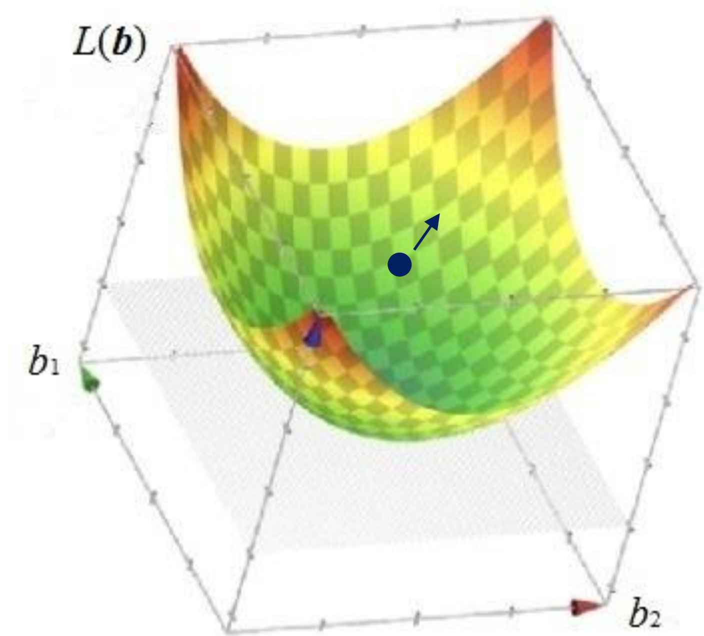


Optimization

$$\hat{\boldsymbol{\beta}} = \operatorname{argmin} \mathcal{L}(\boldsymbol{\beta}) = \operatorname{argmin} \sum_{i=1}^{N} [y_i - f(\boldsymbol{x}_i;\boldsymbol{\beta})]^2$$

Gradient

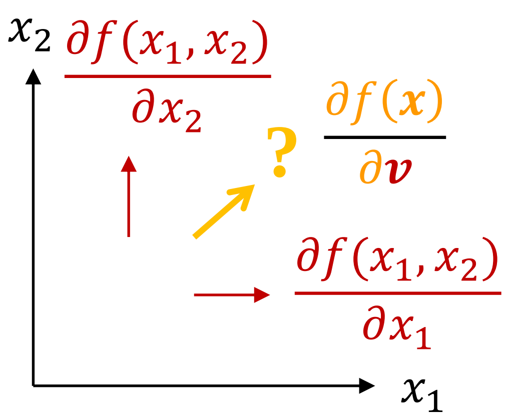
$$\boxed{\nabla\mathcal{L}(\boldsymbol{\beta})} = \frac{\partial\mathcal{L}(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}} = \begin{pmatrix} \dfrac{\partial\mathcal{L}(\boldsymbol{\beta})}{\partial\beta_1} \\ \dfrac{\partial\mathcal{L}(\boldsymbol{\beta})}{\partial\beta_2} \\ \vdots \\ \dfrac{\partial\mathcal{L}(\boldsymbol{\beta})}{\partial\beta_n} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \mathbf{0}$$



Gradient

Why does the gradient vector point in the fastest growing direction of the function?

- **Directional Derivative**



- For any direction $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$ with $\|\boldsymbol{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2} = 1$
- A line pass through point $\boldsymbol{x}^*$ with direction $\boldsymbol{v}$ can be written as:

$$\boldsymbol{x} = \boldsymbol{x}^* + t\boldsymbol{v}$$

- The function value along this line:

$$g(t) = f(\boldsymbol{x}^* + t\boldsymbol{v}) = f(x_1^* + tv_1, x_2^* + tv_2, \ldots, x_n^* + tv_n)$$

- Directional Derivative:

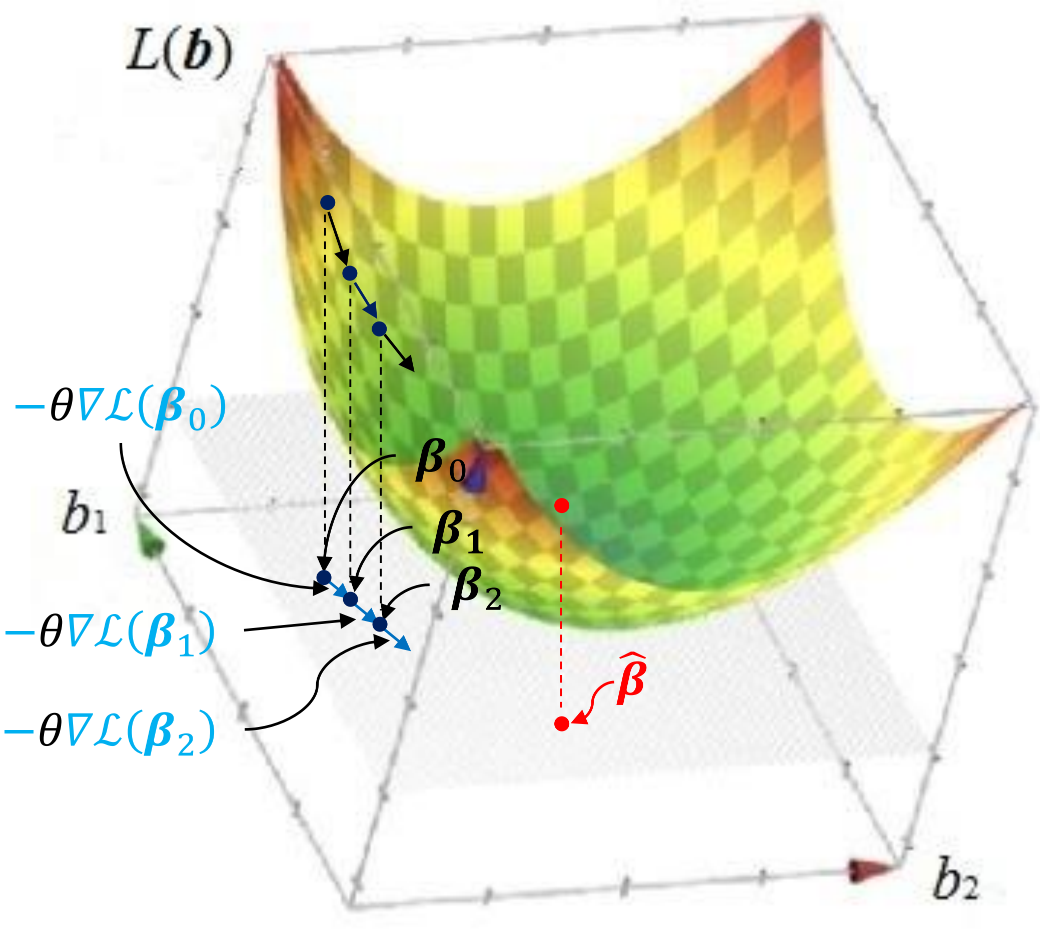
$$\frac{\partial f(\boldsymbol{x}^*)}{\partial \boldsymbol{v}} = \frac{dg(t)}{dt}\Big|_{t=0} = \frac{\partial f(\boldsymbol{x}^*)}{\partial x_1} v_1 + \frac{\partial f(\boldsymbol{x}^*)}{\partial x_2} v_2 + \ldots + \frac{\partial f(\boldsymbol{x}^*)}{\partial x_n} v_n = \nabla f(\boldsymbol{x}^*)^{\mathrm{T}} \boldsymbol{v}$$

Assuming the angle between $\nabla f(\boldsymbol{x}^*)$ and $\boldsymbol{v}$ is θ, then:

$$\cos\theta = \frac{\nabla f(\boldsymbol{x}^*)^{\mathrm{T}} \boldsymbol{v}}{\|\nabla f(\boldsymbol{x}^*)\| \|\boldsymbol{v}\|} \qquad \frac{\partial f(\boldsymbol{x}^*)}{\partial \boldsymbol{v}} = \nabla f(\boldsymbol{x}^*)^{\mathrm{T}} \boldsymbol{v} = \|\nabla f(\boldsymbol{x}^*)\| \cos\theta$$

The directional derivative reaches its maximum when $\cos\theta = 1$ $(\theta = 0)$

Gradient Descent



For step t ($t = 0, 1, 2, \ldots$):

$$\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \underbrace{\theta}_{\text{Step size}}\nabla\mathcal{L}(\boldsymbol{\beta}_t)$$

$$\nabla\mathcal{L}(\boldsymbol{\beta}_{t+k}) = \begin{pmatrix} \dfrac{\partial\mathcal{L}(\boldsymbol{\beta}_{t+k})}{\partial\beta_{t+k,1}} \\ \dfrac{\partial\mathcal{L}(\boldsymbol{\beta}_{t+k})}{\partial\beta_{t+k,2}} \\ \vdots \\ \dfrac{\partial\mathcal{L}(\boldsymbol{\beta}_{t+k})}{\partial\beta_{t+k,3}} \end{pmatrix} \rightarrow \boxed{\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \mathbf{0}}$$

$$\widehat{\boldsymbol{\beta}} = \boldsymbol{\beta}_{t+k}$$

Goodness of Fit

$D_N \longrightarrow$ **Algorithm** $\longrightarrow \hat{\boldsymbol{\beta}} \quad \Rightarrow \quad f(\boldsymbol{x}; \hat{\boldsymbol{\beta}})$

How good is our model ?

Label value variations explained by the model

Residual variations

$$R^2 = \frac{\sum_{i=1}^{N}\left[f(\boldsymbol{x}_i; \hat{\boldsymbol{\beta}}) - \bar{y}\right]^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} = \frac{\sum_{i=1}^{N}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} = \boxed{1 - \frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

Total label value variations

% of label value variations explained by the machine learning model

In-sample and out-of-sample R^2

In-sample R^2:

R^2 is calculated using the training dataset D_N.

Out-of-sample R^2 (OOS R^2):

R^2 is calculated using the new dataset other than the training dataset D_N (testing dataset) .

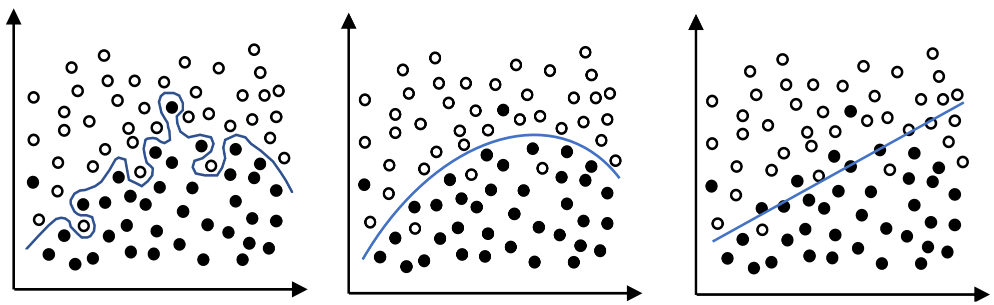
Can we say that a model with high in-sample R^2 is a good model?

Overfitting and Underfitting

When the model trains for too long on sample data or when the model is too complex, it can start to learn the “noise,” or irrelevant information, within the dataset.

When the model memorizes the noise and fits too closely to the training set, the model becomes “overfitted,” and it is unable to generalize well to new data.

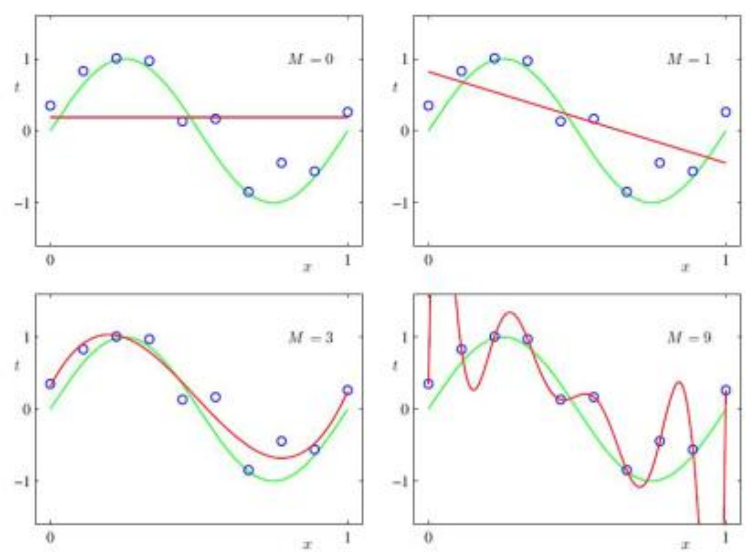
The opposite problem is underfitting. Underfitting occurs when the model has not trained for enough time or the input variables are not significant enough to determine a meaningful relationship between the input and output variables.



Overfitting

Underfitting

Underfitting Underfitting



Overfitting

Cross-validation

训练+测试					验证
折 1	折 2	折 3	折 4	折 5	
折 1	**折 2**	折 3	折 4	折 5	
折 1	折 2	**折 3**	折 4	折 5	
折 1	折 2	折 3	**折 4**	折 5	
折 1	折 2	折 3	折 4	**折 5**	

Summary

Machine learning

Supervised learning

- Regression Problem
- Classification Problem

Unsupervised learning

Reinforcement learning

Framework for regression problem

$$\boldsymbol{x} \longrightarrow \boxed{f} \longrightarrow \hat{y}$$

$$\boxed{f}$$

Variable $\boldsymbol{x}$

$\boxed{\text{Parameters } \boldsymbol{\beta}}$

Function form f

$$D_N \longrightarrow \boxed{\textbf{\textit{Algorithm}}} \longrightarrow \hat{\boldsymbol{\beta}}$$

$$\operatorname{argmin} \mathcal{L}(\boldsymbol{\beta}) = \operatorname{argmin} \sum_{i=1}^{N} [y_i - f(\boldsymbol{x}_i; \boldsymbol{\beta})]^2$$

$$\frac{\partial f(\boldsymbol{x}^*)}{\partial \boldsymbol{v}} = \nabla f(\boldsymbol{x}^*)^{\mathrm{T}} \boldsymbol{v} = \|\nabla f(\boldsymbol{x}^*)\| \cos\theta$$

$$\nabla \mathcal{L}(\boldsymbol{\beta}) = \frac{\partial \mathcal{L}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \begin{pmatrix} \frac{\partial \mathcal{L}(\boldsymbol{\beta})}{\partial \beta_1} \\ \frac{\partial \mathcal{L}(\boldsymbol{\beta})}{\partial \beta_2} \\ \vdots \\ \frac{\partial \mathcal{L}(\boldsymbol{\beta})}{\partial \beta_n} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \mathbf{0}$$

Gradient Descent $\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \theta \nabla \mathcal{L}(\boldsymbol{\beta}_t)$

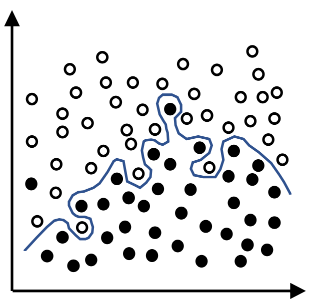
Summary

$$R^2 = \frac{\sum_{i=1}^{N}\left[f\left(\boldsymbol{x}_i;\hat{\boldsymbol{\beta}}\right)-\bar{y}\right]^2}{\sum_{i=1}^{N}(y_i-\bar{y})^2} = \frac{\sum_{i=1}^{N}(\hat{y}_i-\bar{y})^2}{\sum_{i=1}^{N}(y_i-\bar{y})^2} = 1-\frac{\sum_{i=1}^{N}(\hat{y}_i-y_i)^2}{\sum_{i=1}^{N}(y_i-\bar{y})^2}$$

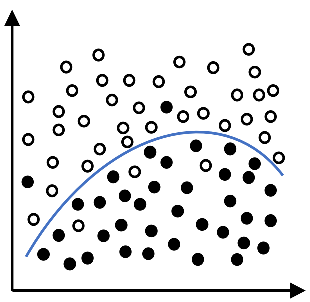
% of label value variations explained by the machine learning model

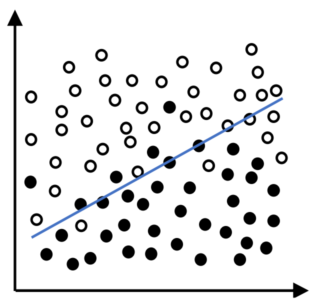
In-sample R^2 Out-of-sample R^2

Overfitting/Underfitting



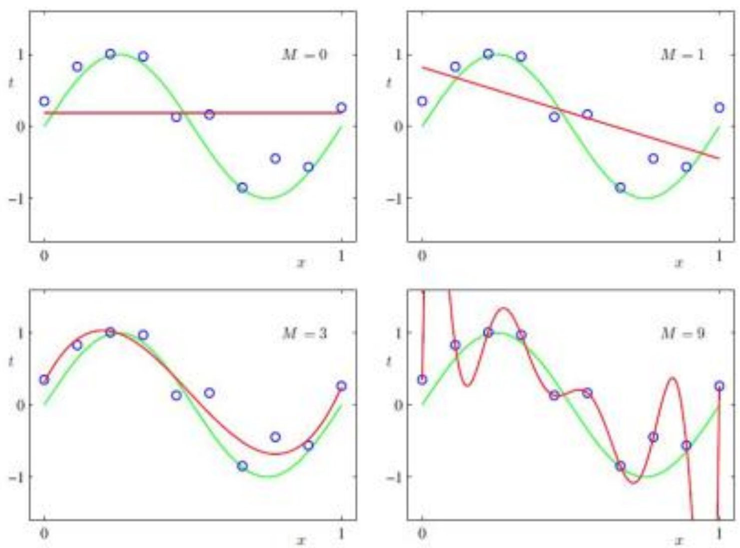
Overfitting





Underfitting

Underfitting Underfitting



Overfitting

05

A Basic Framework for Supervised Learning: Classification Problem

THE HITSZ
SCHOOL OF ECONOMICS
AND MANAGEMENT

Categorical Label

1	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud
2	1	PAYMENT	9839.64	C1231006815	170136	160296.36	M1979787155	0	0	0
3	1	PAYMENT	1864.28	C1666544295	21249	19384.72	M2044282225	0	0	0
4	1	TRANSFER	181	C1305486145	181	0	C553264065	0	0	1
5	1	CASH_OUT	181	C840083671	181	0	C38997010	21182	0	1
6	1	PAYMENT	11668.14	C2048537720	41554	29885.86	M1230701703	0	0	0
7	1	PAYMENT	7817.71	C90045638	53860	46042.29	M573487274	0	0	0
8	1	PAYMENT	7107.77	C154988899	183195	176087.23	M408069119	0	0	0
9	1	PAYMENT	7861.64	C1912850431	176087.23	168225.59	M633326333	0	0	0
10	1	PAYMENT	4024.36	C1265012928	2671	0	M1176932104	0	0	0
11	1	DEBIT	5337.77	C712410124	41720	36382.23	C195600860	41898	40348.79	0
12	1	DEBIT	9644.94	C1900366749	4465	0	C997608398	10845	157982.12	0
13	1	PAYMENT	3099.97	C249177573	20771	17671.03	M2096539129	0	0	0
14	1	PAYMENT	2560.74	C1648232591	5070	2509.26	M972865270	0	0	0
15	1	PAYMENT	11633.76	C1716932897	10127	0	M801569151	0	0	0

(Training) Data: $D_N = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ Label $y_i = \{0,1,2,\ldots,l\}$, $l \geq 1$

Examples:

The firm has financial fraud → Label = 1;

The firm does not have financial fraud → Label = 0;

Blood Type A → Label = 0;

Blood Type B → Label = 1;

Blood Type AB → Label = 2;

Blood Type O → Label = 3

Machine Learning Models

One common way to deal with classification problem is that we predict the probability of the label taking a particular value rather than the label value itself.

$$f(\boldsymbol{x};\boldsymbol{\beta}) = P(y = l|\boldsymbol{x})$$

- **Logit Model**

$\exp\left(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}\right) = e^{\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}}$

$$f(\boldsymbol{x};\boldsymbol{\beta}) = \frac{\exp(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x})}{1+\exp(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x})} = \frac{1}{1+\exp(-\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x})}$$

Input variables: $\boldsymbol{x} = \left(x^{(1)},\ x^{(2)},\ \dots\ ,\ x^{(d)}\right)^{\mathrm{T}} \in \boldsymbol{R}^d$

Parameters: $\boldsymbol{\beta} = (\beta_1,\ \beta_2,\ \dots\ ,\ \beta_d)^{\mathrm{T}} \in \boldsymbol{R}^d$

- **Probit Model**

Cumulative Distribution Function (CDF) for standard normal distribution

$$f(\boldsymbol{x};\boldsymbol{\beta}) = \Phi\left(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}\right) = \int_{-\infty}^{\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}} \phi(t)dt$$

Input variables: $\boldsymbol{x} = \left(x^{(1)},\ x^{(2)},\ \dots\ ,\ x^{(d)}\right)^{\mathrm{T}} \in \boldsymbol{R}^d$

Parameters: $\boldsymbol{\beta} = (\beta_1,\ \beta_2,\ \dots\ ,\ \beta_d)^{\mathrm{T}} \in \boldsymbol{R}^d$

Probability Density Function (PDF) for standard normal distribution

What is a good model?

$D_N \longrightarrow$ ***Algorithm*** $\longrightarrow$ Parameters of $\boldsymbol{f}$

What is the objective of this algorithm?

For regression problem:
Minimize the distince between **the predicted value of label** and **the actual value of label**.

$$\widehat{\boldsymbol{\beta}} = \operatorname{argmin} \sum_{i=1}^{N} [y_i - f(\boldsymbol{x}_i; \boldsymbol{\beta})]^2$$

For classification problem:

?

(we predict the probability of the label taking a particular value)

Maximize the probability of actual label value distribution in D_N.

Likelihood Function (Binary Classification Problems)

Binary classification problems:

Label: 0 or 1 (e.g., whether a firm has financial constraints, whether a firm engage in fraud)

We use model $f(\boldsymbol{x};\boldsymbol{\beta})$ to predict the probability that the label value equals 1:

$$P(y=1|\boldsymbol{x}) = f(\boldsymbol{x};\boldsymbol{\beta})$$

Therefore

$$P(y=0|\boldsymbol{x}) = 1 - f(\boldsymbol{x};\boldsymbol{\beta})$$

- For data point $i \in \{1, 2, \dots, N\}$

$$P(y=y_i|\boldsymbol{x_i}) = \begin{cases} f(\boldsymbol{x_i};\boldsymbol{\beta}) & if\ \ y_i = 1 \\ 1 - f(\boldsymbol{x_i};\boldsymbol{\beta}) & if\ \ y_i = 0 \end{cases}$$

$$\Updownarrow$$

$$P(y=y_i|\boldsymbol{x_i}) = [f(\boldsymbol{x_i};\boldsymbol{\beta})]^{y_i}[1 - f(\boldsymbol{x_i};\boldsymbol{\beta})]^{1-y_i}$$

- The probability of actual label value distribution in D_N (the product of probabilities for all the data points in D_N):

Likelihood Fuction

$$L(\boldsymbol{\beta}) = P = \prod_{i=1}^{N} [f(\boldsymbol{x_i};\boldsymbol{\beta})]^{y_i}[1 - f(\boldsymbol{x_i};\boldsymbol{\beta})]^{1-y_i}$$

Model Training Summary

We want a good way (model) to predict the probability of the label taking a particular value.

$\Downarrow$

Maximize the probability of actual label value distribution in D_N.

$\Downarrow$

$$\boxed{\hat{\boldsymbol{\beta}} = \operatorname{argmax} L(\boldsymbol{\beta}) = \operatorname{argmax} \prod_{i=1}^{N} [f(\boldsymbol{x}_i;\boldsymbol{\beta})]^{y_i}[1 - f(\boldsymbol{x}_i;\boldsymbol{\beta})]^{1-y_i}}$$?

$D_N \longrightarrow \boxed{\textbf{\textit{Algorithm}}} \longrightarrow \boxed{\text{Parameters of } \boldsymbol{f}}$

$\boldsymbol{x} \longrightarrow \boxed{\boldsymbol{f}} \longrightarrow P(y = 1|\boldsymbol{x})$

$\xrightarrow{\text{criteria}} P(y = 1|\boldsymbol{x}) > \boxed{P_{th}} \longrightarrow \hat{y}$

Optimization

$L(\boldsymbol{\beta}) > 0$ $\quad\therefore\quad$ $\text{argmax}\, L(\boldsymbol{\beta}) \Longleftrightarrow \text{argmax}\, ln[L(\boldsymbol{\beta})]$

$$\boxed{ln[L(\boldsymbol{\beta})]} = \sum_{i=1}^{N} y_i \, ln[f(\boldsymbol{x_i};\boldsymbol{\beta})] + \sum_{i=1}^{N} (1-y_i) ln[1-f(\boldsymbol{x_i};\boldsymbol{\beta})]$$

Log likelihood function

$$\nabla lnL(\boldsymbol{\beta}) = \frac{\partial lnL(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \begin{pmatrix} \frac{\partial lnL(\boldsymbol{\beta})}{\partial \beta_1} \\ \frac{\partial lnL(\boldsymbol{\beta})}{\partial \beta_2} \\ \vdots \\ \frac{\partial lnL(\boldsymbol{\beta})}{\partial \beta_n} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \mathbf{0}$$

Gradient ascent

$$\boldsymbol{\beta_{t+1}} = \boldsymbol{\beta_t} + \theta \nabla lnL(\boldsymbol{\beta_t})$$

$$\nabla \mathcal{L}(\boldsymbol{\beta_{t+k}}) = \mathbf{0}$$

$$\widehat{\boldsymbol{\beta}} = \boldsymbol{\beta_{t+k}}$$

Goodness of Fit

Log likelihood of the model with constant term only

$$\boldsymbol{Pseudo\ R^2} = \frac{lnL_0 - lnL_1}{lnL_0}$$

→ Log likelihood of the full model (with all variables)

$$lnL_0 \leq lnL_1 \leq 0$$

$$0 \leq Pseudo\ R^2 \leq 1$$

Higher $Pseudo\ R^2$ → Greater model fit

$$\boldsymbol{A} = \frac{\sum_{i=1}^{N} \boxed{\mathbf{1}(\hat{y}_i = y_i)}}{N}$$

→ Returns 1 if $\hat{y}_i = y_i$, and 0 otherwise.

Number of data points

Accuracy (准确率)

% of correctly classified data points.

Goodness of Fit

$$\boldsymbol{M} = \frac{\sum_{i=1}^{N} \boxed{\mathbf{1}(\hat{y}_i \neq y_i)}}{N}$$

Returns 1 if $\hat{y}_i \neq y_i$, and 0 otherwise.

Number of data points

% of misclassified data points.

Misclassification Rate
(错误率/错分率)

THE HITSZ SCHOOL OF ECONOMICS AND MANAGEMENT

Confusion Matrix

		True Values	
		True	False
Predicted Values	Positive	True Positive (TP, 真阳性) $(\hat{y}=1, y=1)$	False Positive (FP, 假阳性) $(\hat{y}=1, y=0)$
	Negative	False Negative (FN, 假阴性) $(\hat{y}=0, y=1)$	True Negative (TN, 真阴性) $(\hat{y}=0, y=0)$

$$Sensitivity = \frac{TP}{TP+FN}$$

Sensitivity (灵敏度/真阳率)

$$Specificity = \frac{TN}{FP+TN}$$

Specificity (特异度/真阴率)

$1 - Specificity \longrightarrow$ 假阳率，反例预测错误的比例

Receiver Operating Characteristic (ROC) Curve

现实中假阴性和假阳性对应的成本可能并不对称：

- 有病的病人被诊断为没病 vs 没病的人被诊断为有病
- 劣质客户被判断为正常客户而放款 vs 正常客户被判断为劣质客户而被拒绝放款

$$P(y=1|\boldsymbol{x}) \longrightarrow P(y=1|\boldsymbol{x}) > P_{th} \longrightarrow \hat{y}$$

例如相比于错失正常客户，客户断供给银行造成的损失更大，因此银行可以将门槛值P_{th}设置为0.2，即模型预测出20%以上概率客户会断供就将客户分类为劣质客户并拒绝放款。

$P_{th} \downarrow$

Sensitivity (灵敏度/真阳率) $\uparrow$

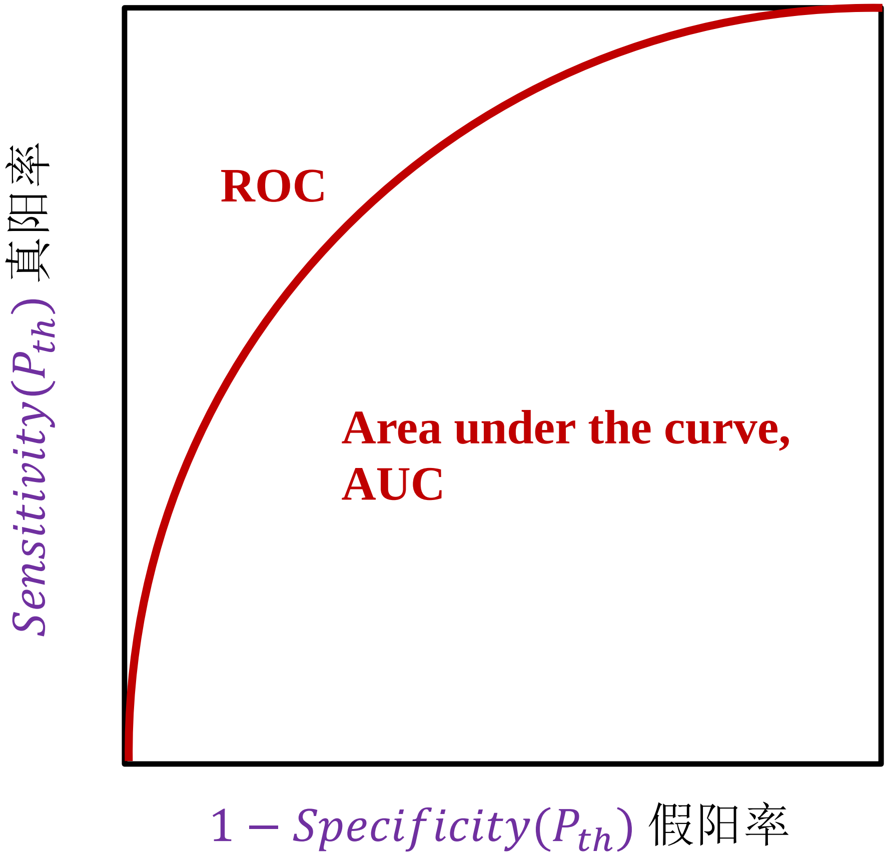
Specificity (特异度/真阴率) $\downarrow$

1-Specificity (假阳率，1-特异度) $\uparrow$

$Sensitivity(P_{th})$

$1-Specificity(P_{th})$

Receiver Operating Characteristic (ROC) Curve



Comparison between machine learning, statistics and econometrics

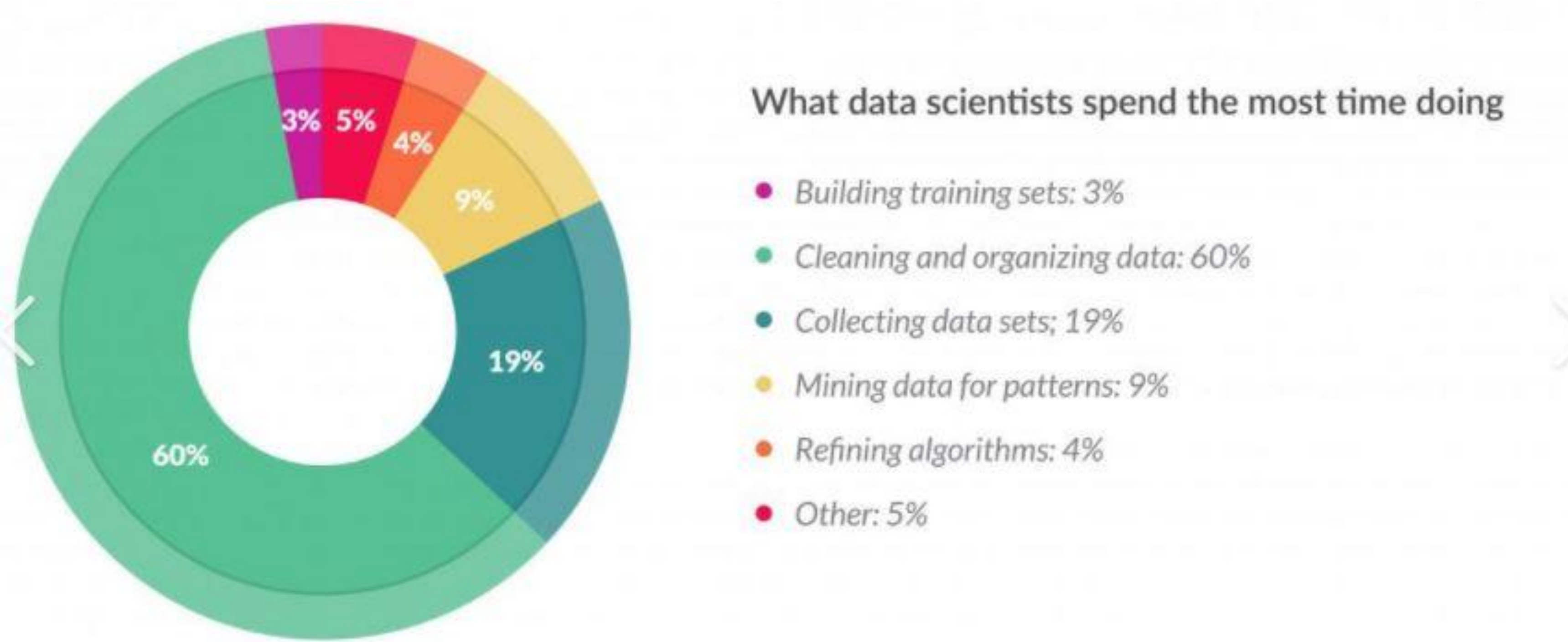
学科领域	数据分析目的	对参数估计的重视度	模型直观性
机器学习	预测为主	低	取决于模型复杂程度
统计学	统计推断为主	高	高
计量经济学	因果推断为主	高	高

06

Features

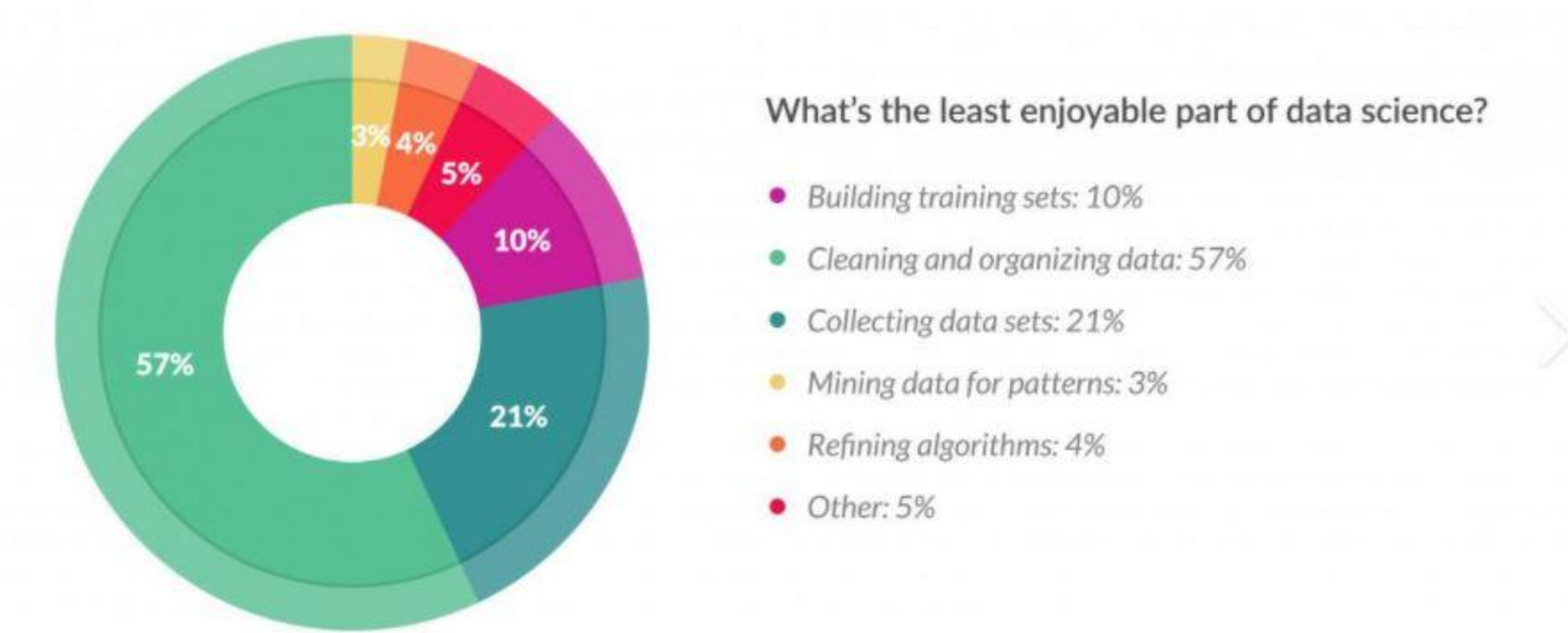
THE HITSZ
SCHOOL OF ECONOMICS
AND MANAGEMENT

What Does a Data Scientist Do?



Data scientists spend 60% of their time on cleaning and organizing data. Collecting data sets comes second at 19% of their time, meaning data scientists spend around 80% of their time on preparing and managing data for analysis.

What Does a Data Scientist Do?



57% of data scientists regard cleaning and organizing data as the least enjoyable part of their work and 19% say this about collecting data sets.

Feature Creation

	资产	负债	净利润	营收	行业	审计师是否为四大	S&P债券信用评级	高管期权激励	是否存在舞弊
1	101			100		是			否
2	40			20		否		CEO 权	否
3	61			50		否		CEO 均存	是
4	120			99		是			否
5	87			100		否		CEO 权	是

资产	负债
101	30
40	23
61	45
120	12
87	52

资产负债率
0.297
0.575
0.738
0.100
0.598

净利润	营收
6	100
-2	20
1	50
14	99
2	100

净利润率
0.060
-0.100
0.020
0.141
0.020

Data Encoding

	资产负债率	净利润率	行业	审计师是否为四大	S&P债券信用评级	高管期权激励	是否存在舞弊
1	0.297	0.060		是		无	否
2	0.575	-0.100		否		CEO存在期权激励	否
3	0.738	0.020		否		CEO和CFO均存在期权激励	是
4	0.100	0.141		是		无	否
5	0.598	0.020		否		CEO存在期权激励	是

$\{‘是’, ‘否’\} \leftrightarrow \{1, 0\}$

是否存在舞弊
否
否
是
否
是

是否存在舞弊
0
0
1
0
1

Data Encoding

One-hot encoding

1 m-categories variable → m binary variables

	资产负债率	净利润率	行业	审计师是否为四大	S&P债券信用评级	高管期权激励	是否存在舞弊
1	0.297	0	制造				0
2	0.575	00	采矿				0
3	0.738	0	农				1
4	0.100	1	制造				0
5	0.598	0	建筑				1

行业	是否制造业	是否采矿业	是否农业	是否建筑业
制造业	1	0	0	0
采矿业	0	1	0	0
农业	0	0	1	0
制造业	1	0	0	0
建筑业	0	0	0	1

Data Encoding

	资产负债率	净利润率	制造业	采矿业	农业	建筑业	审计师是否为四大	S&P债券信用评级	高管期权激励	是否存在舞弊
1	0.297	0.060	1	0	0	0	1	AA	无	0
2	0.575	-0.100	0	1	0	0	0	C	CEO存在期权激励	0
3	0.738	0.020	0	0	1	0	0	BBB	CEO和CFO均存在期权激励	1
4	0.100	0.141	1	0	0	0	1	AAA	无	0
5	0.598	0.020	0	0	0	1	0	B	CEO存在期权激励	1

Data Encoding

Any difference？

级别		评定
AAA	8	最高评级。偿还债务能力极强。
SD/D	0	动以致债务的偿付受阻时，标准普尔亦会给予'D'评级。当发债人有选择地对某些或某类债务违约时，标准普尔会给予"SD"评级（选择性违约

行业	S&P债券信用评级	S&P债券信用评级	AAA	AA	A	BBB	BB	B	CCC	CC	C	D
制造业	AA	7	0	1	0	0	0	0	0	0	0	0
采矿业	C	1	0	0	0	0	0	0	0	0	1	0
农业	BBB	5	0	0	0	1	0	0	0	0	0	0
制造业	AAA	8	0	0	0	0	0	0	0	0	0	0
建筑业	B	3	0	0	0	0	0	1	0	0	0	0

Data Encoding

	资产负债率	净利润率	制造业	采矿业	农业	建筑业	审计师是否为四大	S&P债券信用评级	高管期权激励	是否存在舞弊
1										0
2										0
3										1
4										0
5										1

Should we use one-hot encoding?

高管期权激励	仅CEO有	仅CFO有	均有	均无
无	0	1	0	1
CEO存在期权激励	1	0	0	0
CEO和CFO均存在期权激励	0	0	1	0
无	0	1	0	1
CFO存在期权激励	0	0	0	0

CEO有期权激励	CFO有期权激励
0	0
1	0
1	1
0	0
0	1

Data Encoding

	资产负债率	净利润率	制造业	采矿业	农业	建筑业	审计师是否为四大	S&P债券信用评级	CEO有期权激励	CFO有期权激励	是否存在舞弊
1	0.297	0.060	1	0	0	0	1	7	0	0	0
2	0.575	-0.100	0	1	0	0	0	1	1	0	0
3	0.738	0.020	0	0	1	0	0	5	1	1	1
4	0.100	0.141	1	0	0	0	1	8	0	0	0
5	0.598	0.020	0	0	0	1	0	3	0	1	1

THANK YOU

感谢您的观看！

经济管理学院

THE HITSZ SCHOOL OF ECONOMICS AND MANAGEMENT

THE HITSZ
SCHOOL OF ECONOMICS
AND MANAGEMENT