



2021 K-ICT 빅데이터센터

분석 인프라 활용 AI 교육

인공지능 빅데이터 전문가 심화과정

강 사 성 민 석

고려대학교 인공지능학과



2021 K-ICT 빅데이터센터

분석 인프라 활용 AI 교육

인공지능 빅데이터 전문가 심화과정

강사 소개 및 강의 안내

머신러닝 2일차



과학기술정보통신부

NIA 한국지능정보사회진흥원

강사 소개

<https://open.kakao.com/me/minsuksung>



성 민 석 (Minsuk Sung)

● 학력

- 홍익대학교 컴퓨터공학과 학사 (졸업)
- 고려대학교 인공지능학과 석박사통합과정 (현재)

● 교육 경력

- Intel Korea 이미지분류 및 객체인식 강의 자료 제작
- SK 인공지능 신입전사교육 - 데이터 분석 및 머신러닝
- NAVER BoostCamp AI Tech 1기/2기 디버깅 멘토

● 대회 수상 경력

- 2019년 제 7회 빅콘테스트 최우수상 (과학기술부 주최)
- 2019년 제 3회 빅데이터 페스티벌 최우수상 (미래에셋증권)
- 2020년 공공 데이터 활용 온도 추정 AI 경진대회 2등 (DACON 및 한국원자력연구원 등)
- 2020년 제 4회 빅데이터 페스티벌 2등 (미래에셋증권)
- 2021년 구강계질환 의료영상 AI 경진대회 3등 (삼성서울병원 등)

세부 교육 일정

● 1일차 이론

- 머신러닝 (Machine Learning) 개요
- 머신러닝 기초 이론
- 일반적인 머신러닝 학습과정

● 1일차 실습

- 파이썬 (Python) 기초 문법
- 데이터 분석 라이브러리 활용 - NumPy
- 데이터 분석 라이브러리 활용 - Pandas
- 데이터 분석 라이브러리 활용 - Scikit-Learn

● 2일차 이론

- 회귀 (Regression)
- 분류 (Classification)
- K-최근접 이웃 알고리즘 (KNN)
- 나이브 베이즈 (Naïve Bayes)
- 서포트 벡터 머신 (SVM)

● 2일차 실습

- Boston 데이터를 통해서 알아보는 머신러닝 예제 (1)
- Iris 데이터를 통해서 알아보는 머신러닝 예제 (2)
- 와인 품질 데이터를 통해서 알아보는 머신러닝 예제 (3)
- KOSPI 지수를 통해서 알아보는 머신러닝 예제 (4)

● 3일차 이론

- 의사결정 나무 (Decision Tree)
- 앙상블 (Ensemble)
- 인공 신경망 (Artificial Neural Network)

● 3일차 실습

- 당뇨병 데이터를 통해서 알아보는 머신러닝 예제 (5)
- 유방암 데이터를 통해서 알아보는 머신러닝 예제 (6)
- 타이타닉 생존 여부 데이터를 통해서 알아보는 머신러닝 예제 (7)
- MNIST 숫자 데이터를 통해서 알아보는 머신러닝 예제 (8)

세부 교육 일정

● 1일차 이론

- 머신러닝 (Machine Learning) 개요
- 머신러닝 기초 이론
- 일반적인 머신러닝 학습과정

● 1일차 실습

- 파이썬 (Python) 기초 문법
- 데이터 분석 라이브러리 활용 - NumPy
- 데이터 분석 라이브러리 활용 - Pandas
- 데이터 분석 라이브러리 활용 - Scikit-Learn

● 2일차 이론

- 회귀 (Regression)
- 분류 (Classification)
- K-최근접 이웃 알고리즘 (KNN)
- 나이브 베이즈 (Naïve Bayes)
- 서포트 벡터 머신 (SVM)

● 2일차 실습

- Boston 데이터를 통해서 알아보는 머신러닝 예제 (1)
- Iris 데이터를 통해서 알아보는 머신러닝 예제 (2)
- 와인 품질 데이터를 통해서 알아보는 머신러닝 예제 (3)
- KOSPI 지수를 통해서 알아보는 머신러닝 예제 (4)

● 3일차 이론

- 의사결정 나무 (Decision Tree)
- 앙상블 (Ensemble)
- 인공 신경망 (Artificial Neural Network)

● 3일차 실습

- 당뇨병 데이터를 통해서 알아보는 머신러닝 예제 (5)
- 유방암 데이터를 통해서 알아보는 머신러닝 예제 (6)
- 타이타닉 생존 여부 데이터를 통해서 알아보는 머신러닝 예제 (7)
- MNIST 숫자 데이터를 통해서 알아보는 머신러닝 예제 (8)

세부 교육 일정

※ 주의: 실제 머신러닝 파이프라인과 다소 상이할 수 있습니다

- 이번 머신러닝 교육의 핵심



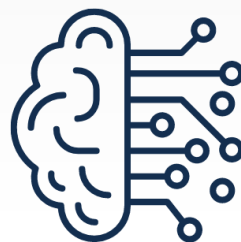
데이터 수집



데이터 전처리
(Data Preprocessing)



피처 엔지니어링
(Feature Engineering)



모델링
(Modeling)



하이퍼파라미터 튜닝
(Hyperparameter Tuning)



평가
(Testing)

세부 교육 일정

※ 주의: 실제 머신러닝 파이프라인과 다소 상이할 수 있습니다

- 이번 머신러닝 교육의 핵심



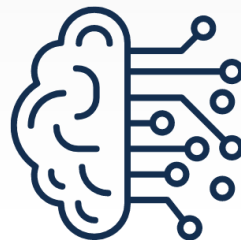
데이터 수집



데이터 전처리
(Data Preprocessing)



피처 엔지니어링
(Feature Engineering)



모델링
(Modeling)



하이퍼파라미터 튜닝
(Hyperparameter Tuning)



평가
(Testing)



모델 생성

`model = DecisionTree()`



모델 학습

`model.fit(X_train, y_train)`



모델 평가

`model.score(X_valid, y_valid)`



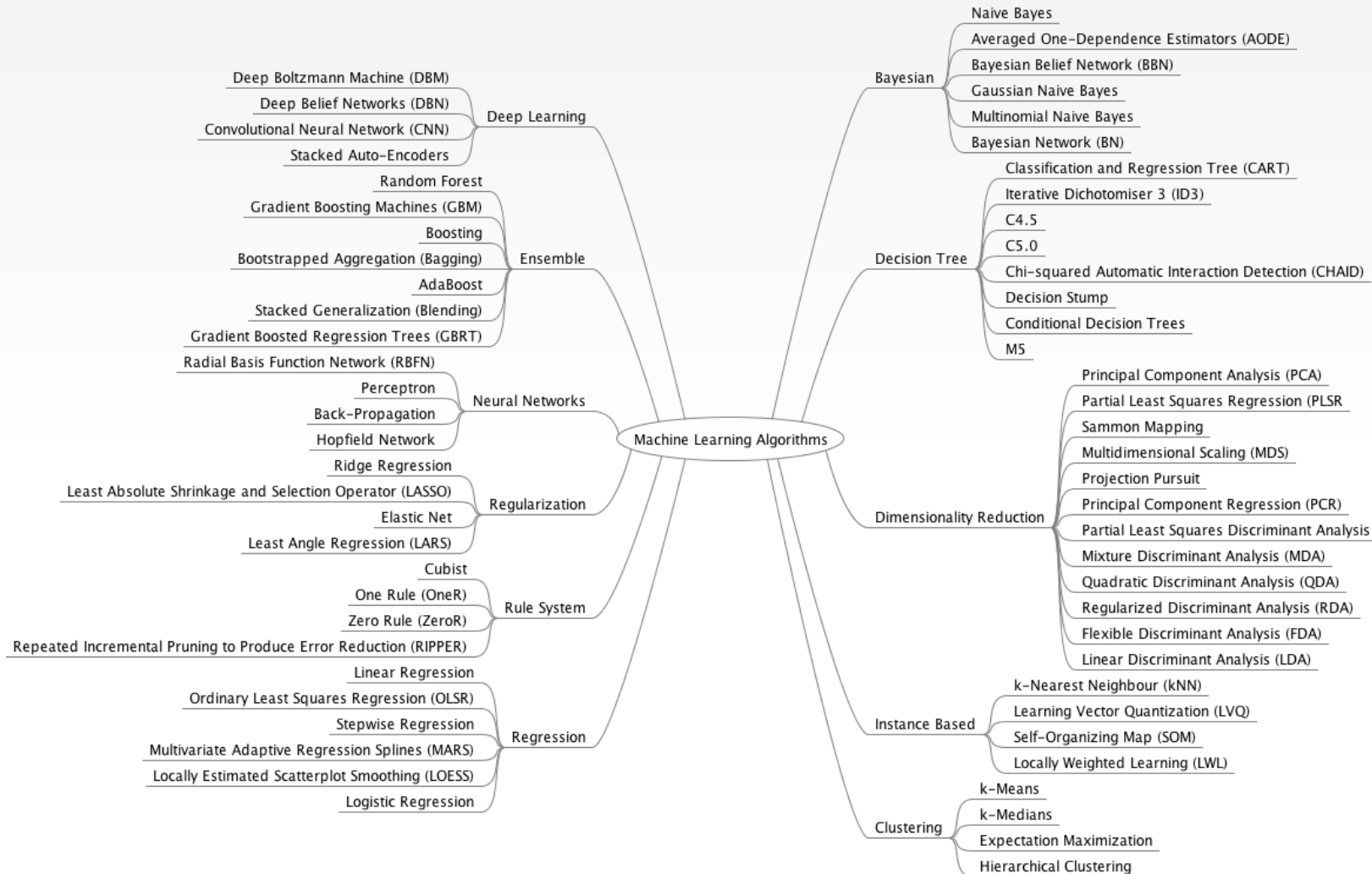
모델 예측

`model.predict(X_test)`

들어가기 앞서

※ 주의: 실제 머신러닝 파이프라인과 다소 상이할 수 있습니다

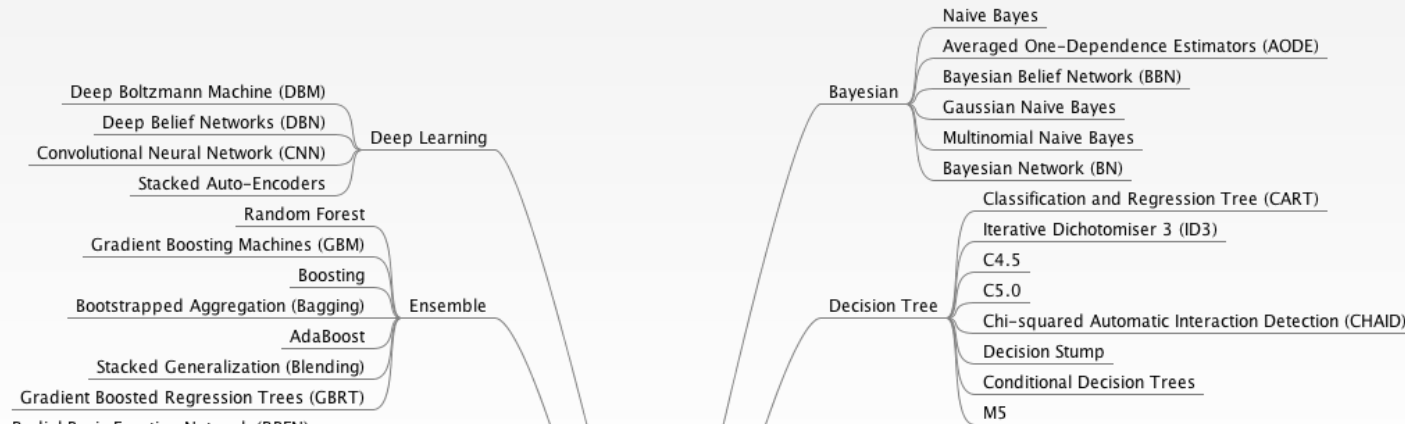
● 그때 그때 달라요



들어가기 앞서

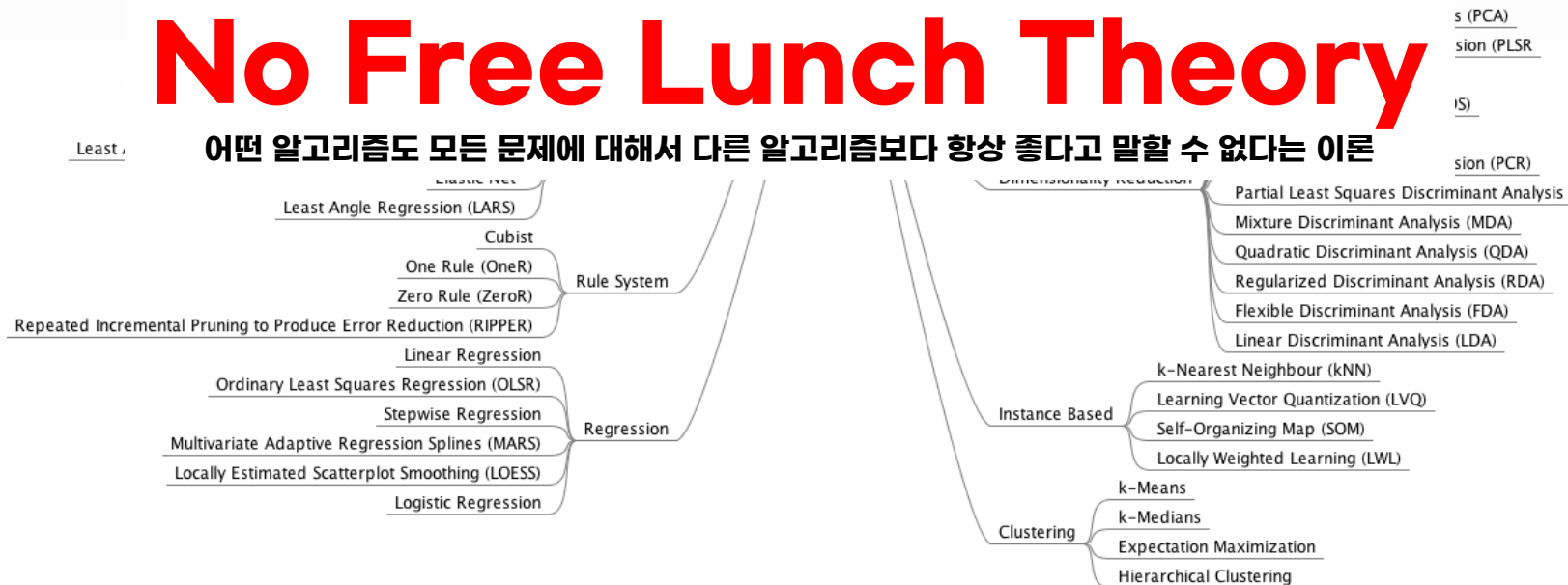
※ 주의: 실제 머신러닝 파이프라인과 다소 상이할 수 있습니다

- 그때 그때 달라요



No Free Lunch Theory

어떤 알고리즘도 모든 문제에 대해서 다른 알고리즘보다 항상 좋다고 말할 수 없다는 이론





인공지능 빅데이터 전문가 심화과정

머신러닝 2일차 이론

$$y = ax + b$$

회귀

$$y = ax + b$$

입력
↓

회귀

출력
↓

$$y = ax + b$$

$$y = ax + b$$

↑
기울기

$$y = ax + b$$

↑
편향

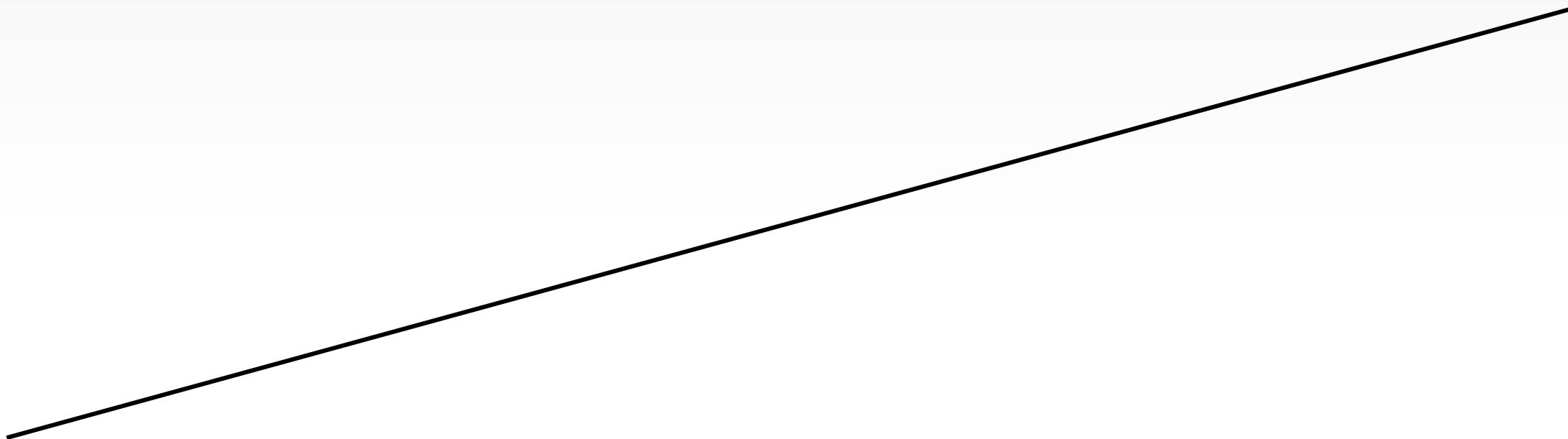
회귀

$$\begin{array}{c} \text{출력} \\ \downarrow \\ y \end{array} = \begin{array}{c} \text{입력} \\ \downarrow \\ ax \end{array} + \begin{array}{c} b \\ \uparrow \\ \text{편향} \end{array}$$

기울기

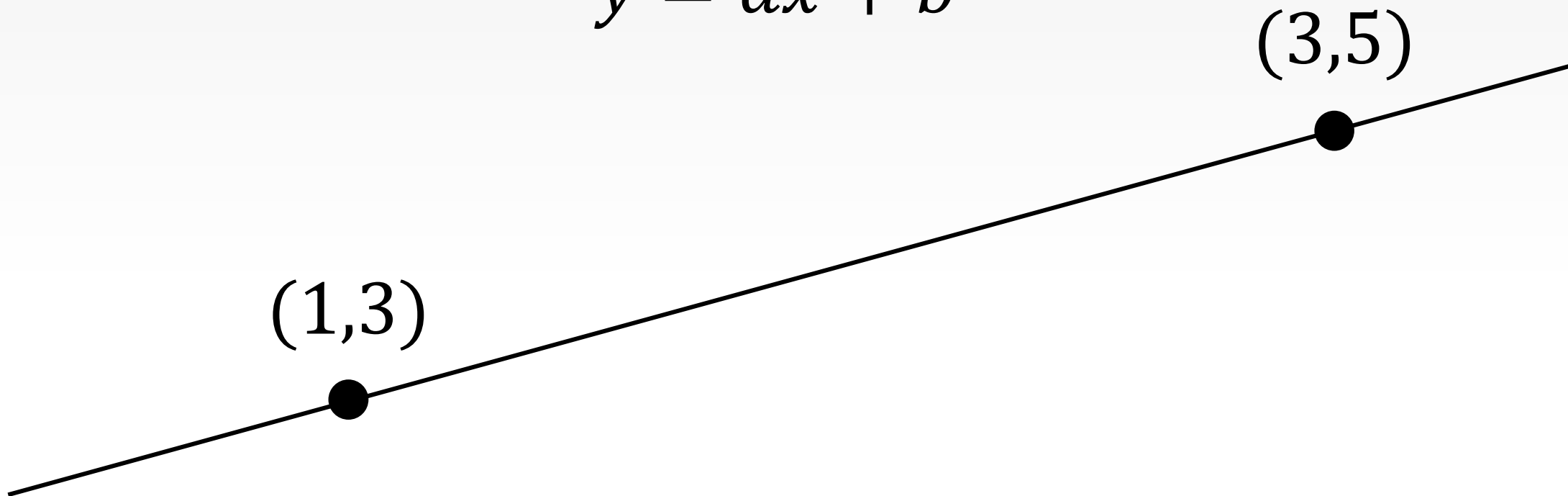
회귀

$$y = ax + b$$

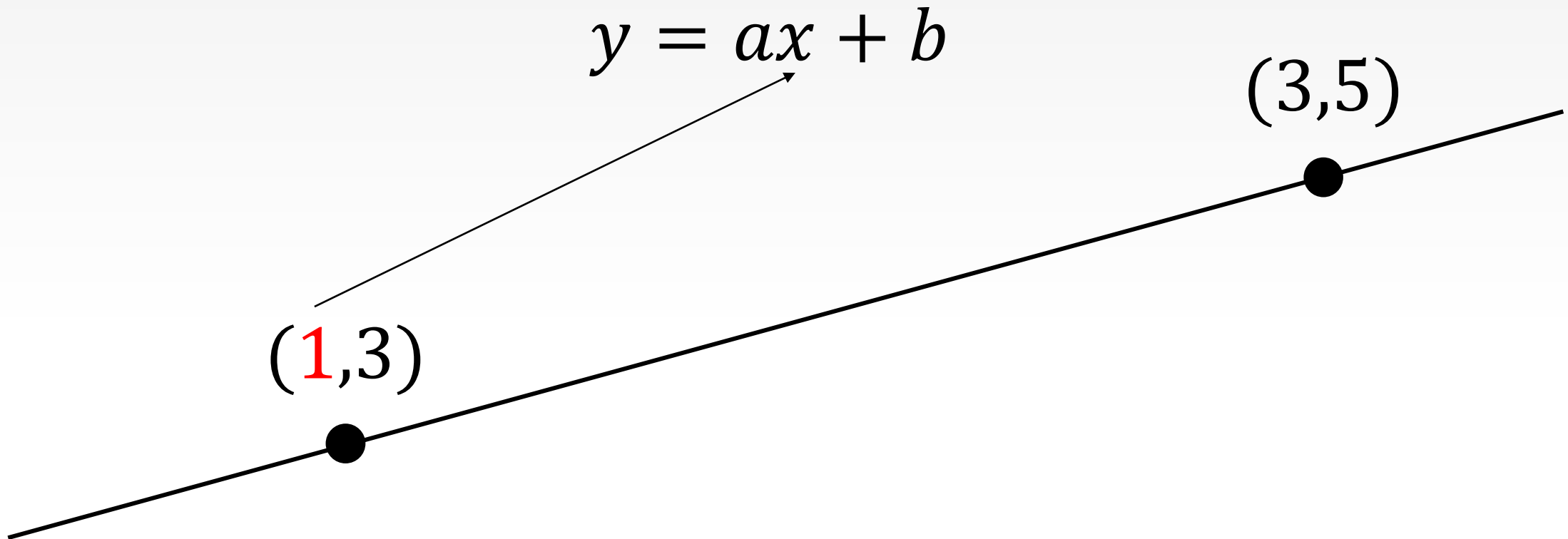


회귀

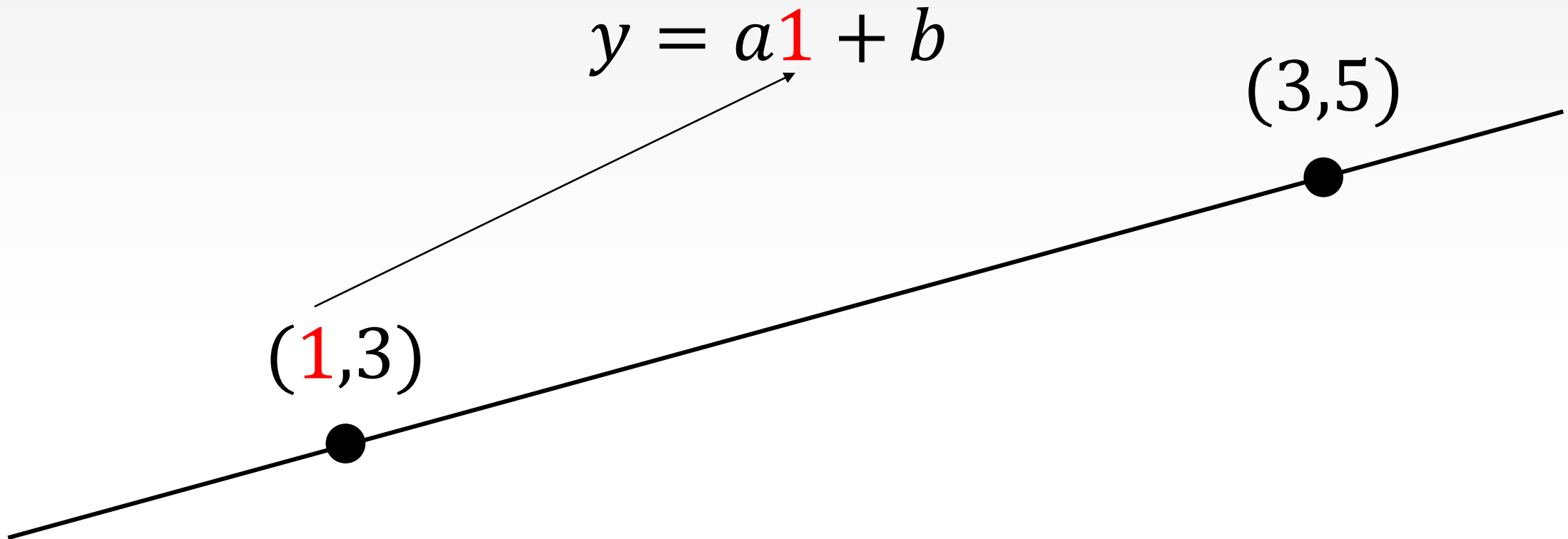
$$y = ax + b$$



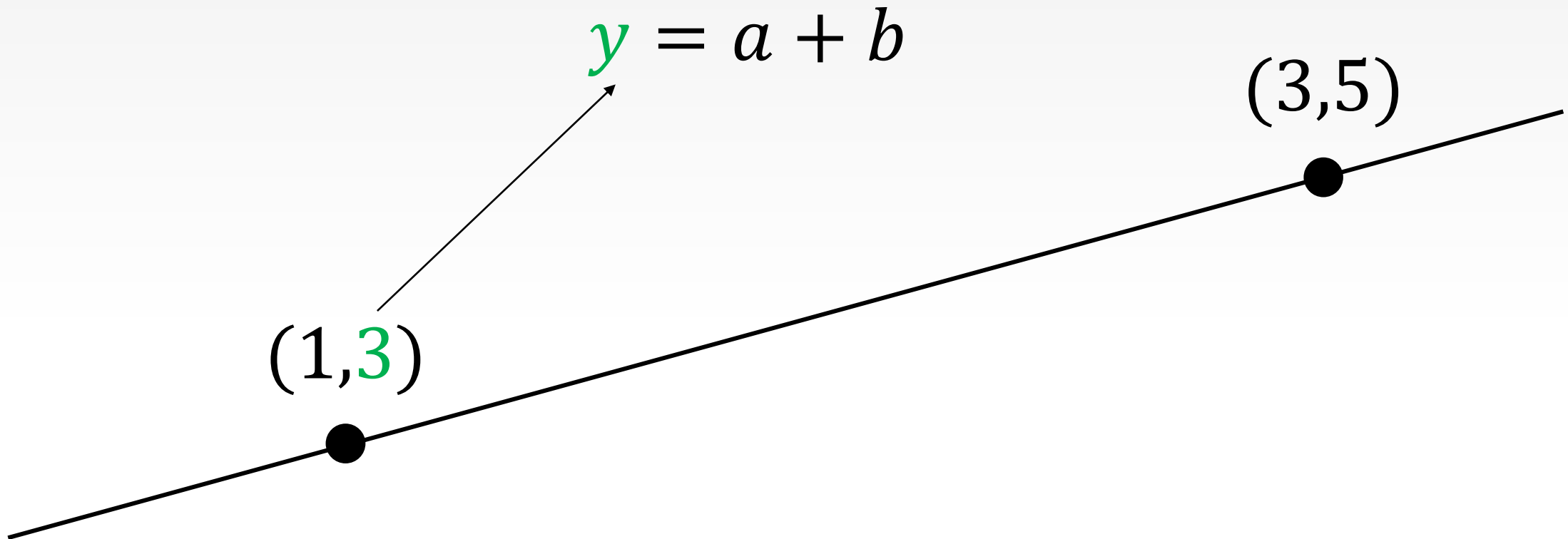
회귀



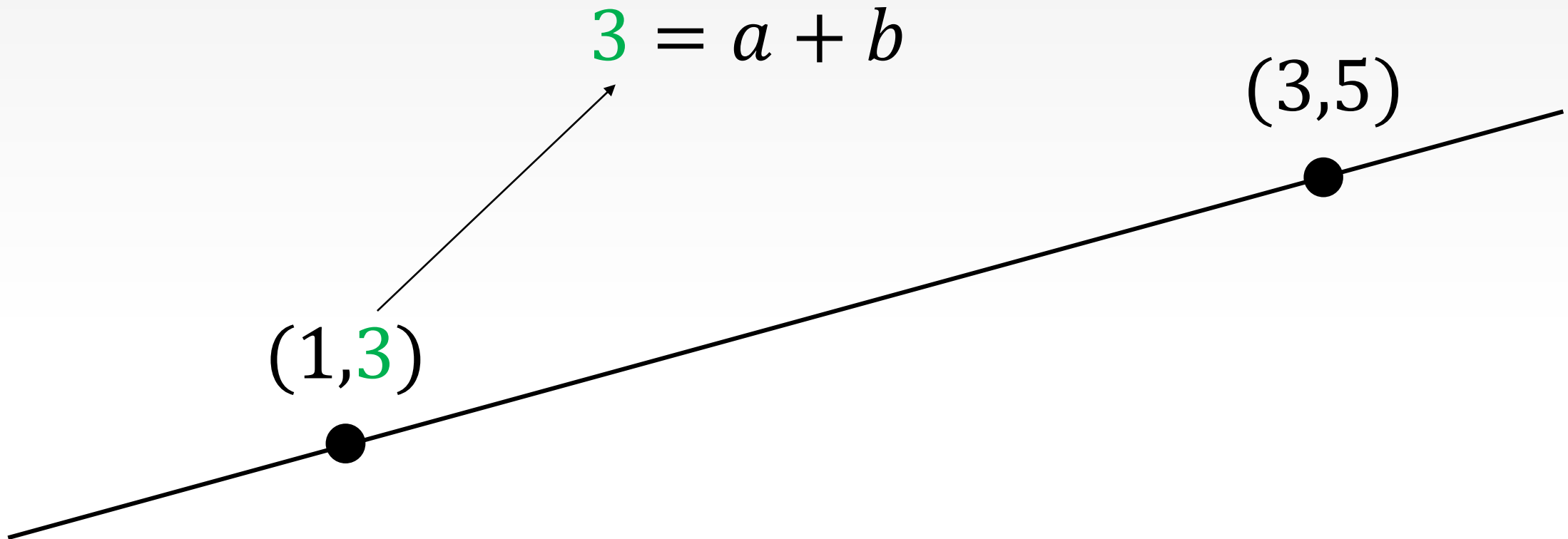
회귀



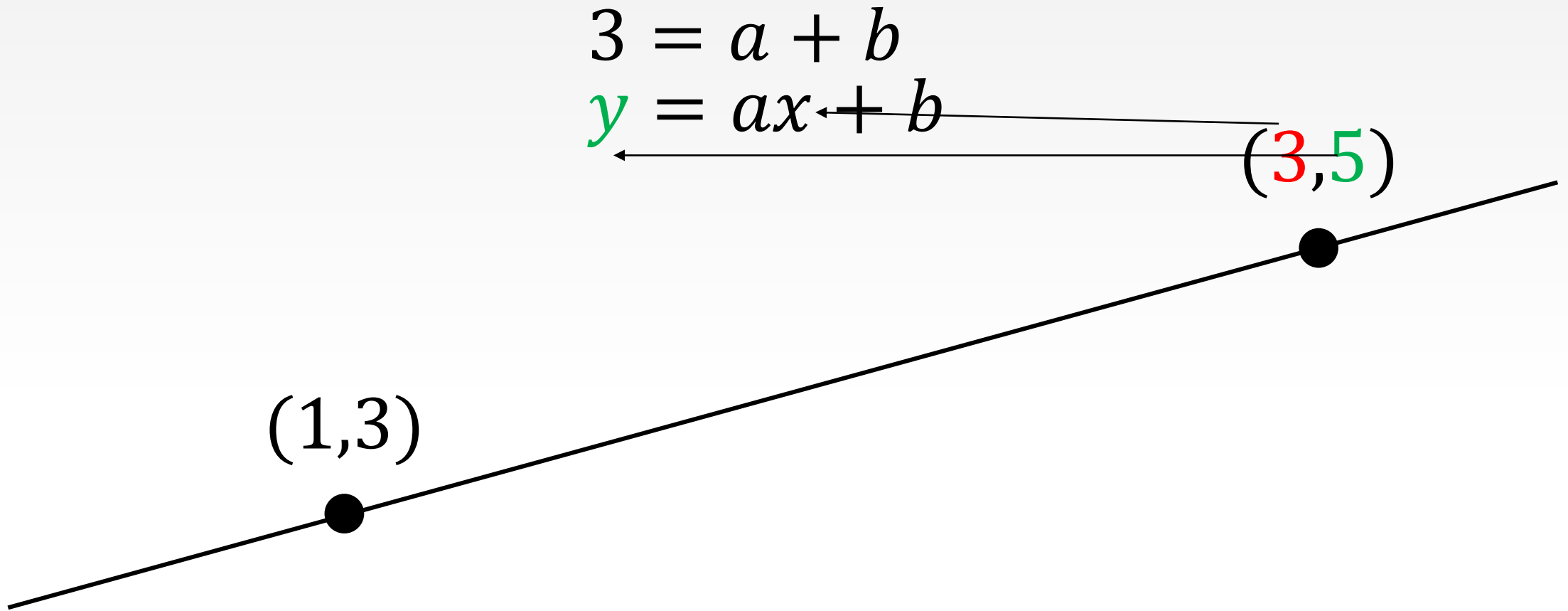
회귀



회귀

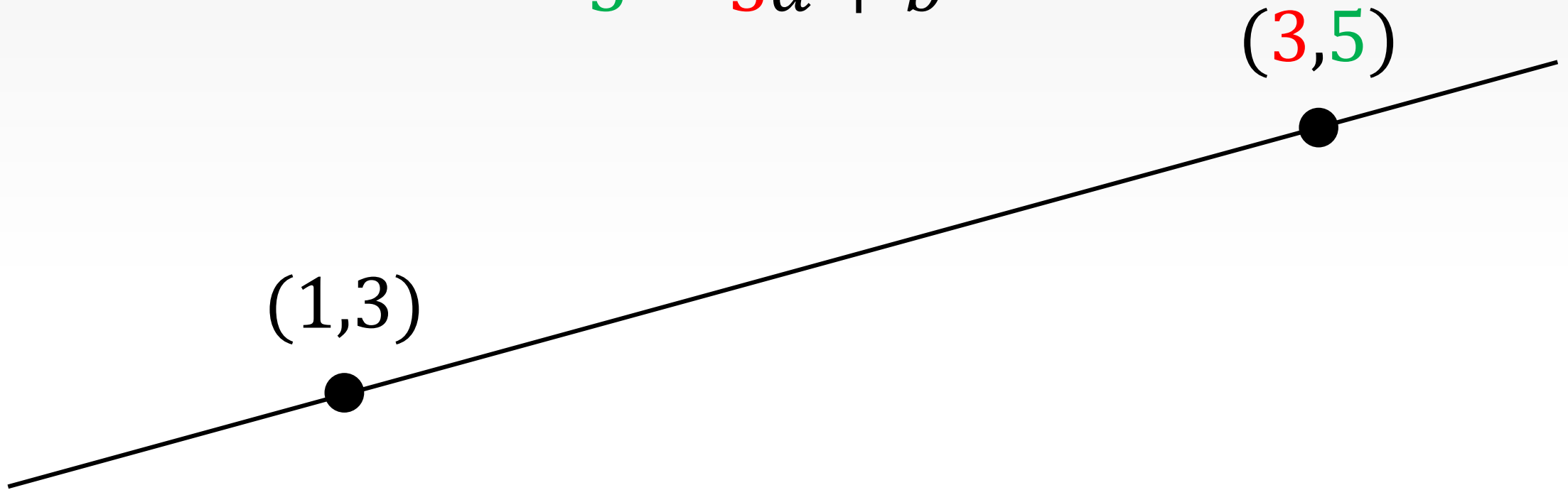


회귀



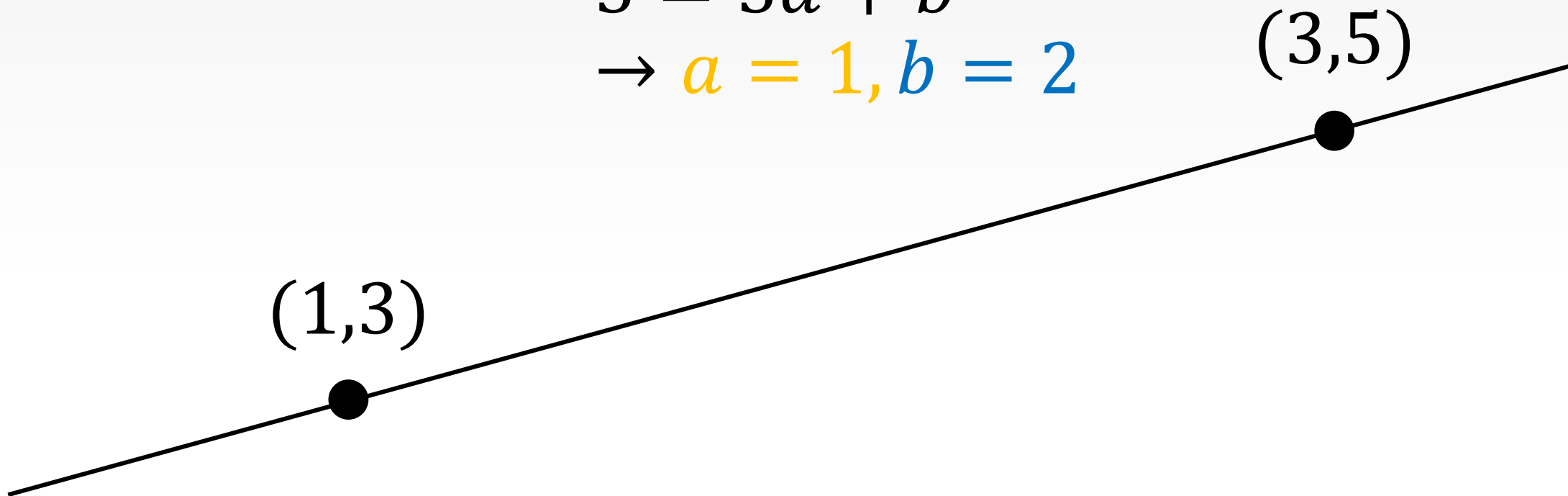
회귀

$$\begin{aligned} 3 &= a + b \\ 5 &= 3a + b \end{aligned}$$



회귀

$$\begin{aligned}3 &= a + b \\5 &= 3a + b \\ \rightarrow a &= 1, b = 2\end{aligned}$$



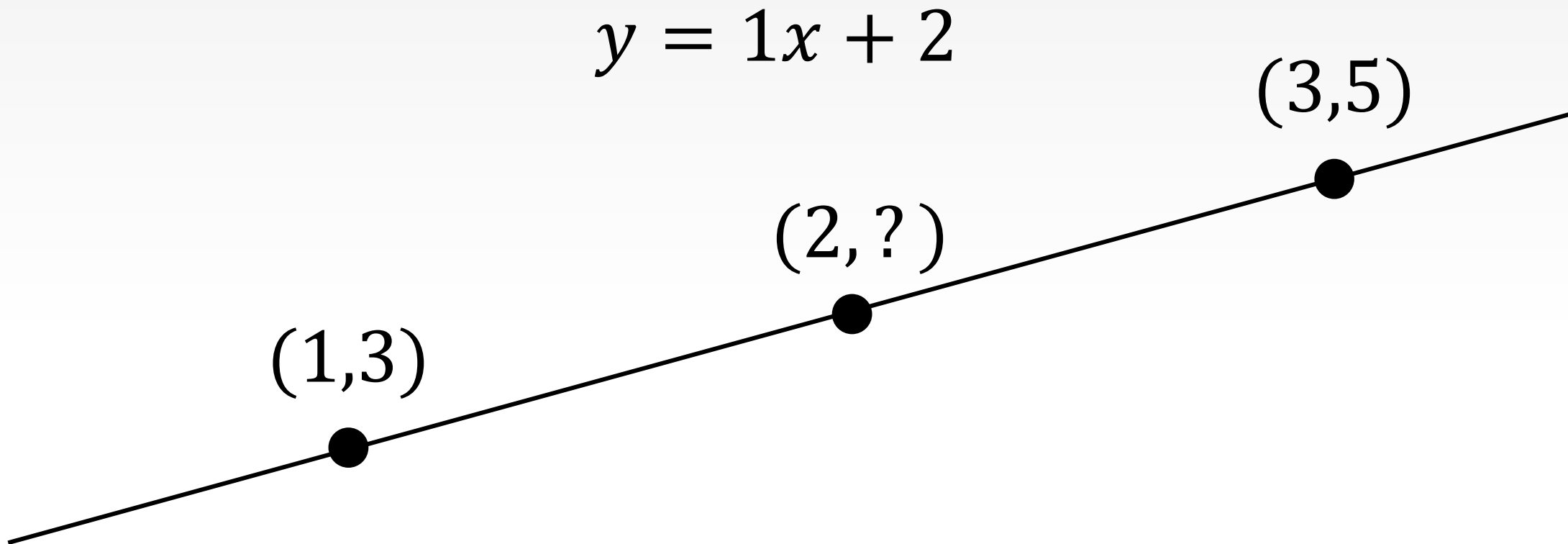
회귀

$$y = 1x + 2$$

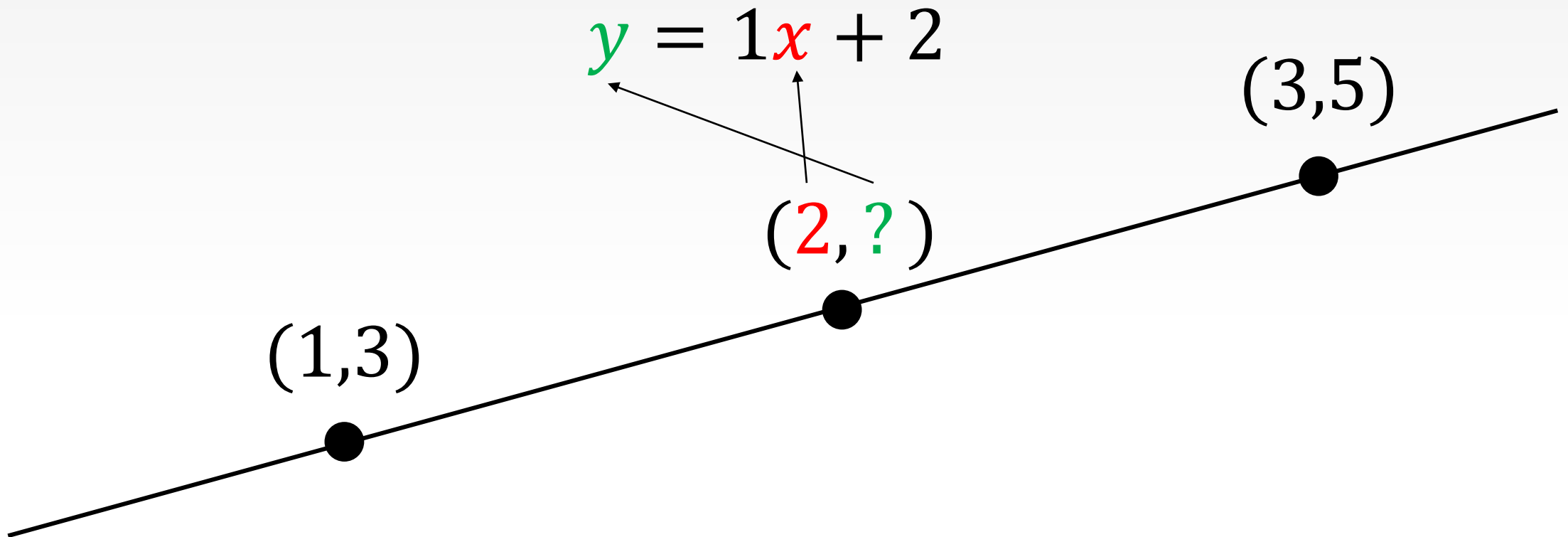
(1,3)

(3,5)

회귀

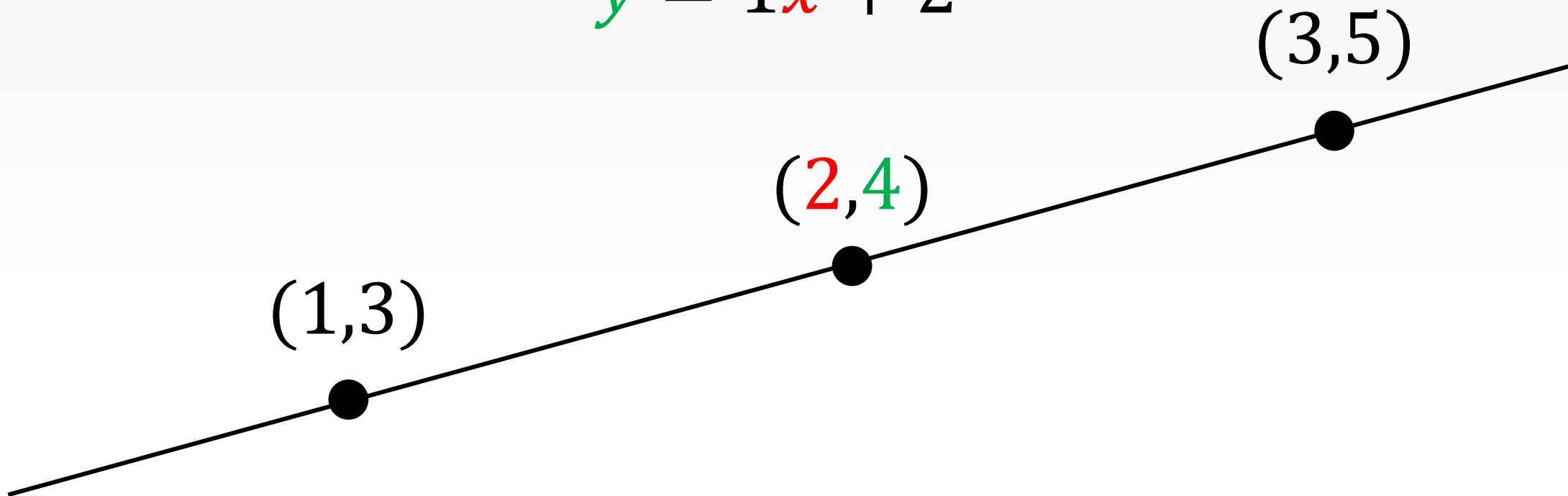


회귀



회귀

$$y = 1x + 2$$



회귀

$$y = 1x + 2$$

(3,5)

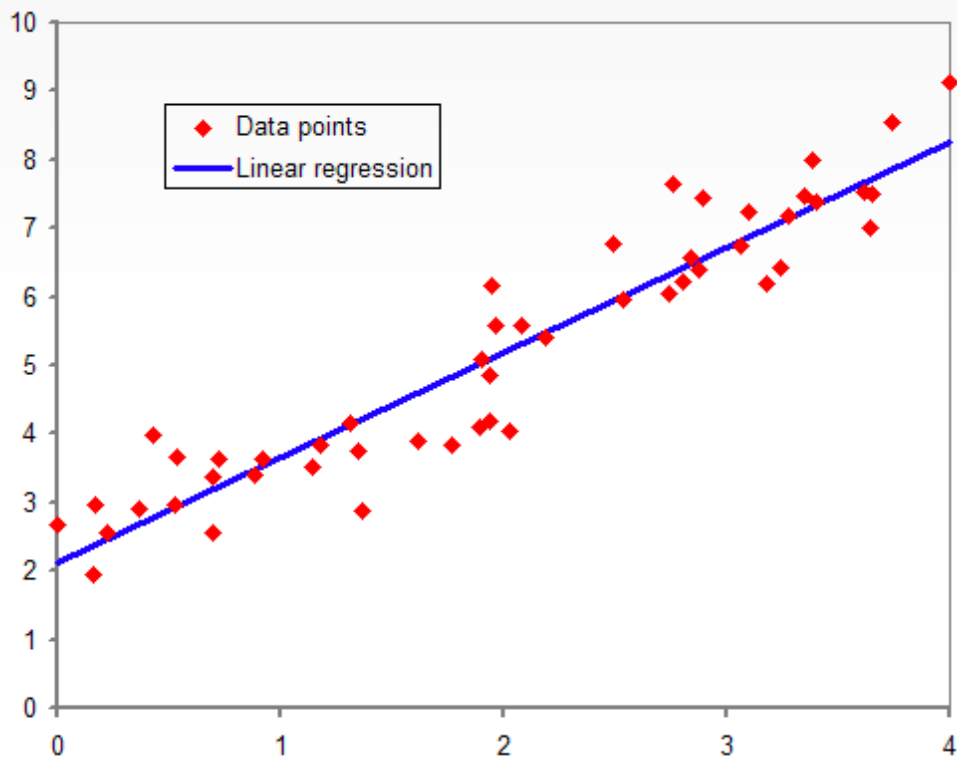
(1,3)

하지만!
실제로 수많은 데이터가
이렇게 **한 직선 상에** 있는 경우가
거의 없다

회귀

● 선형 회귀

- 종속 변수 y 와 한 개 이상의 독립 변수 x 와의 상관 관계를 모델링하는 분석 기법
- 변수들 간의 상관 관계를 파악하여, 어떤 특정 변수의 값을 다른 변수 값을 이용하여 설명하거나 예측하는 기법



$$\begin{array}{c} \text{출력} \\ \downarrow \\ y \end{array} = \begin{array}{c} \text{입력} \\ \downarrow \\ ax \end{array} + \begin{array}{c} \text{편향} \\ \uparrow \\ b \end{array}$$

a is labeled \uparrow 기울기

회귀

- 단순 선형 회귀

- 독립 변수 x 와 종속 변수 y 의 관계를 $y = w_0 + w_1 x$ 와 같은 형태의 1차 함수 식으로 표현 가능

- 회귀 계수 (Coefficient)

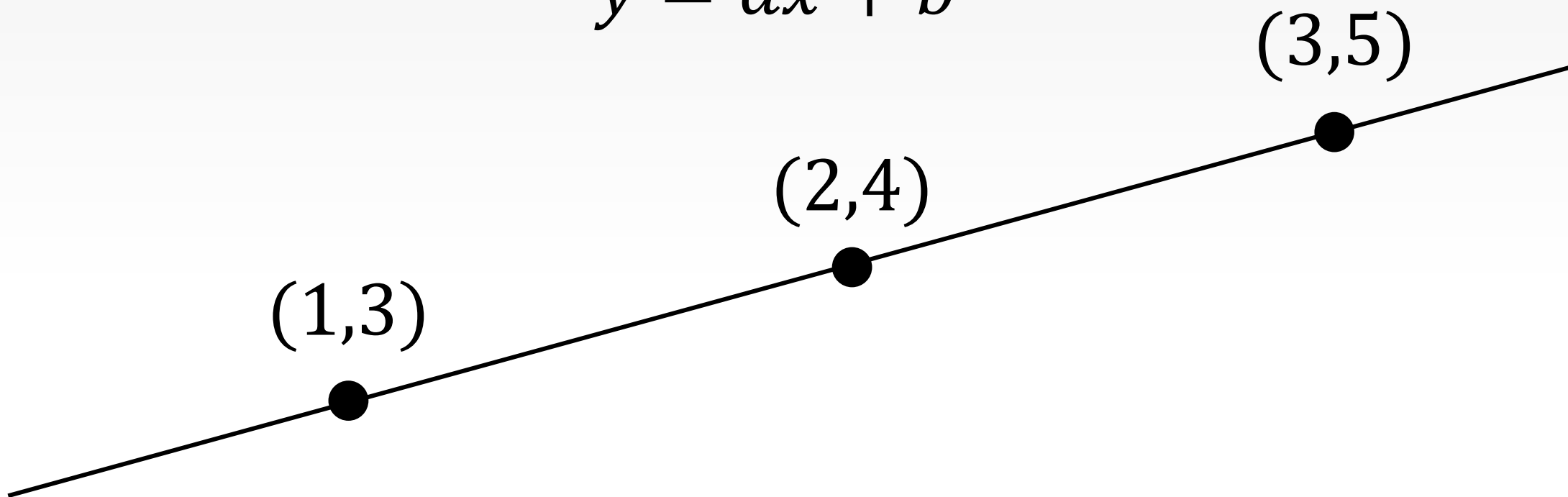
- 독립 변수가 종속 변수에 끼치는 영향력의 정도 → 기울기

- 절편 (Intercept)

- 독립변수가 0일 때의 상수

회귀

$$y = ax + b$$



회귀

어떻게 가중치와 편향을
변경해도 한 직선에
놓을 수 없네...

$$y = ax + b$$

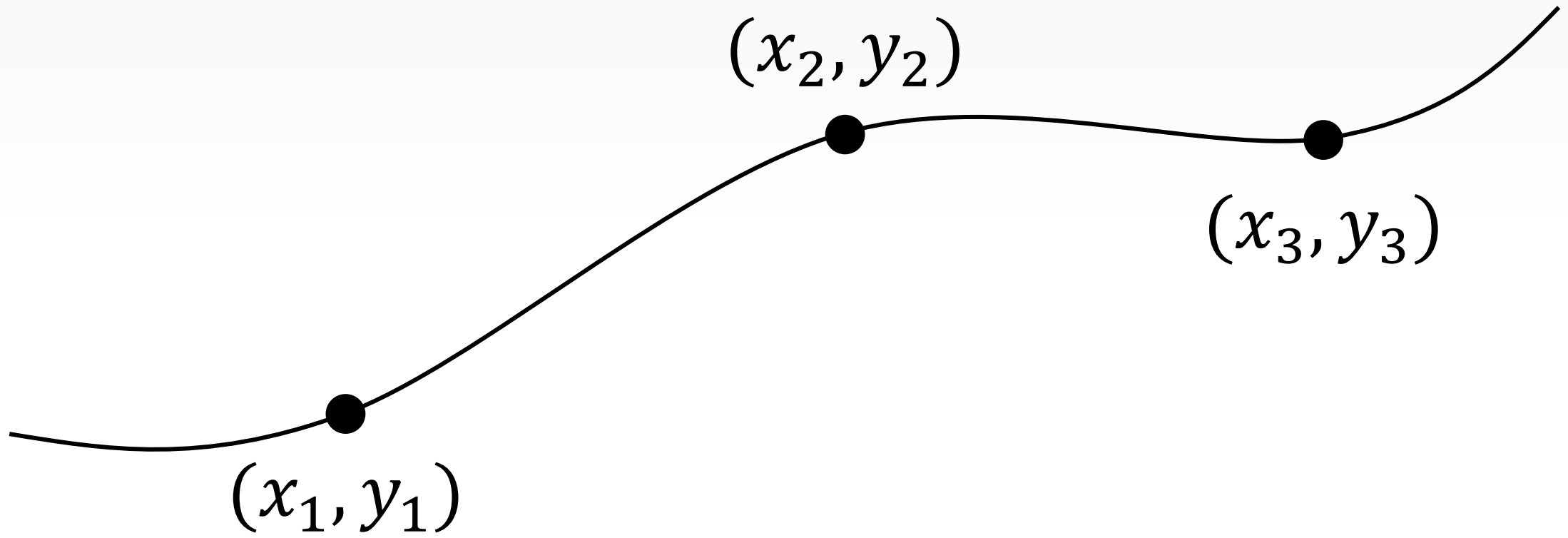
(x_2, y_2)

(x_3, y_3)

(x_1, y_1)

회귀

$$y = ax + b$$



회귀

최대한 한 직선에
놓을 수 있게끔 만들어보자

$$y = ax + b$$

(x_2, y_2)

(x_3, y_3)

(x_1, y_1)

회귀

$$y = ax + b$$

(x_2, y_2)

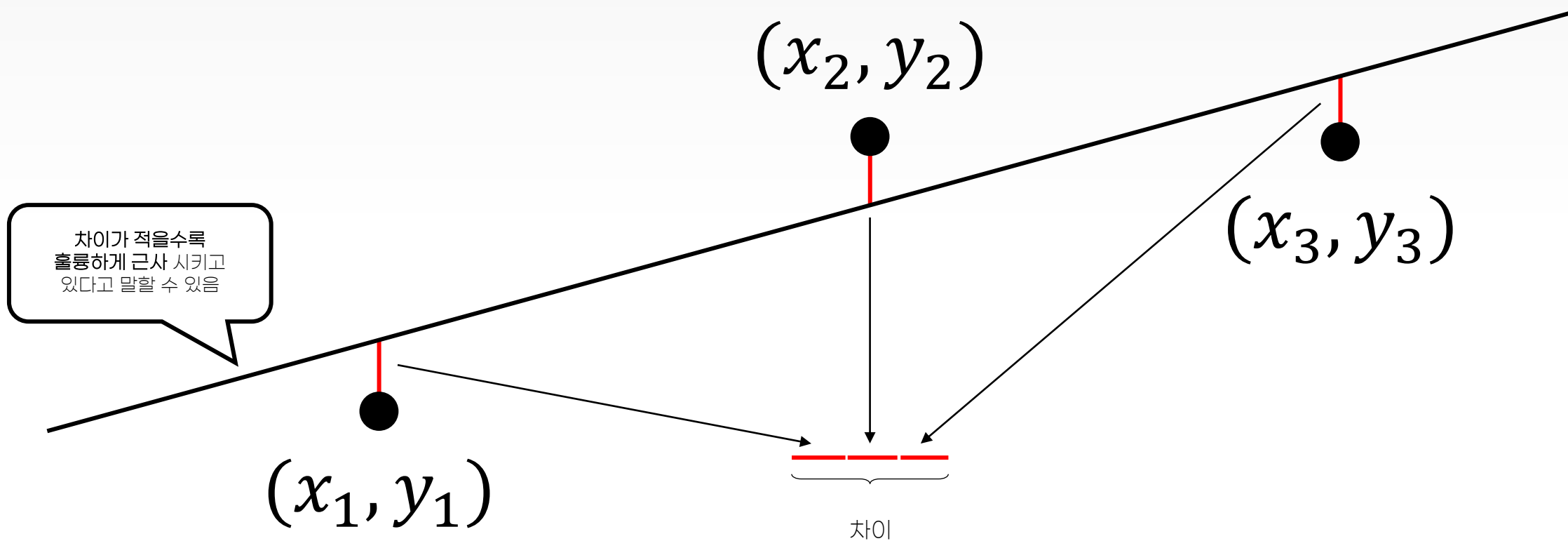
(x_3, y_3)

직선을 이리저리
움직여보면서 **차이**를
최소화해보자

(x_1, y_1)

회귀

$$y = ax + b$$



회귀

$$y = ax + b$$

(x_2, y_2)

(x_3, y_3)

기울기를 변경해보면서
차이를 줄이거나

(x_1, y_1)

회귀

$$y = ax + b$$

(x_2, y_2)

(x_3, y_3)

(x_1, y_1)

편향을 변경해보면서
차이를 줄이거나


회귀

- 주요 평가 지표

평가 지표	의미	수식
MAE (Mean Absoulte Error)	실제값과 예측값의 차이의 절대값의 합	$\sum y - \hat{y} $
MSE (Mean Squared Error)	실제값과 예측값의 차이의 제곱의 합	$\sum (y - \hat{y})^2$
RMSE (Root MSE)	MSE의 제곱근 값	\sqrt{MSE}
R^2	실제값의 분산 대비 예측값의 분산의 비율	$1 - \frac{\sum \text{오차}^2}{\sum \text{편차}^2}$

← L1

← L2


[Install](#)
[User Guide](#)
[API](#)
[Examples](#)
[More ▾](#)

[Prev](#)
[Up](#)
[Next](#)

scikit-learn 0.24.2
[Other versions](#)

Please [cite us](#) if you use the software.

[sklearn.linear_model.LinearRegression](#)
[Examples using sklearn.linear_model.LinearRegression](#)

[Toggle Menu](#)

sklearn.linear_model.LinearRegression

```
class sklearn.linear_model.LinearRegression(*, fit_intercept=True, normalize=False, copy_X=True, n_jobs=None, positive=False)
```

[\[source\]](#)

Ordinary least squares Linear Regression.

LinearRegression fits a linear model with coefficients $w = (w_1, \dots, w_p)$ to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.

Parameter	fit_intercept : bool, default=True
s:	Whether to calculate the intercept for this model. If set to False, no intercept will be used in calculations (i.e. data is expected to be centered).
	normalize : bool, default=False
	This parameter is ignored when <code>fit_intercept</code> is set to False. If True, the regressors X will be normalized before regression by subtracting the mean and dividing by the l2-norm. If you wish to standardize, please use <code>StandardScaler</code> before calling <code>fit</code> on an estimator with <code>normalize=False</code> .
	copy_X : bool, default=True
	If True, X will be copied; else, it may be overwritten.
	n_jobs : int, default=None
	The number of jobs to use for the computation. This will only provide speedup for <code>n_targets > 1</code> and sufficient large problems. <code>None</code> means 1 unless in a <code>joblib.parallel_backend</code> context. <code>-1</code> means using all processors. See Glossary for more details.

휴식 시간

10분간 쉬고 다시 시작합니다! ☺

<https://open.kakao.com/me/minsuksung>



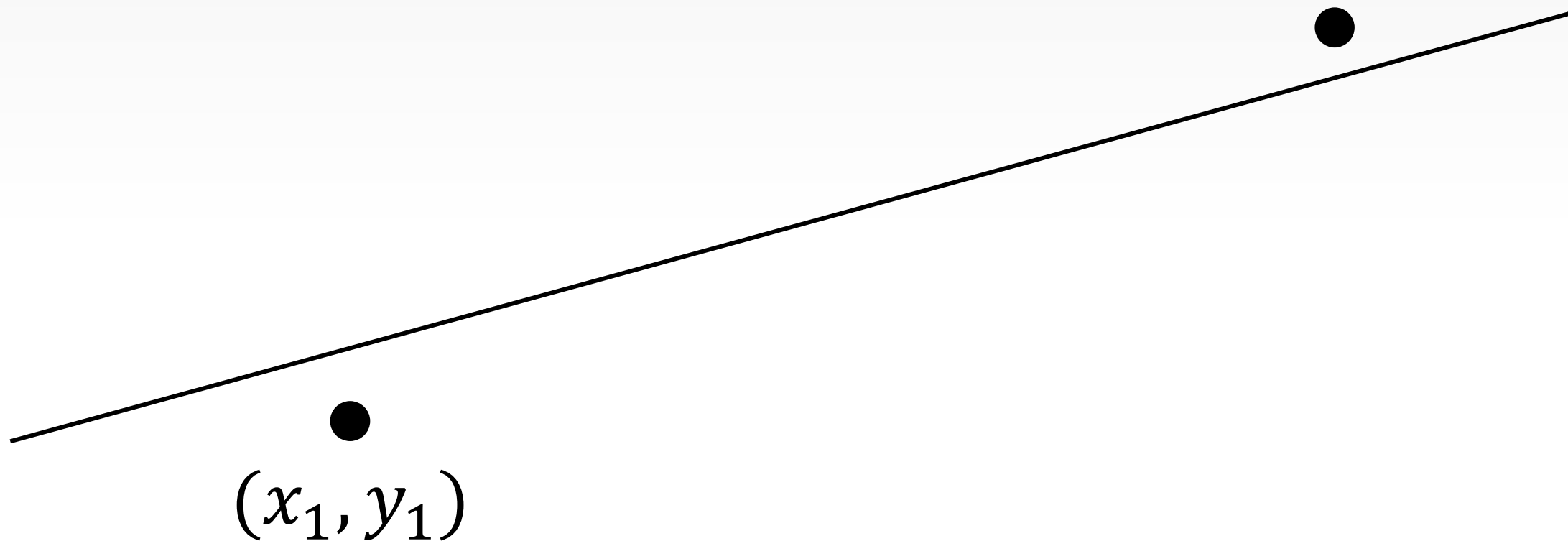


인공지능 빅데이터 전문가 심화과정

머신러닝 2일차 이론

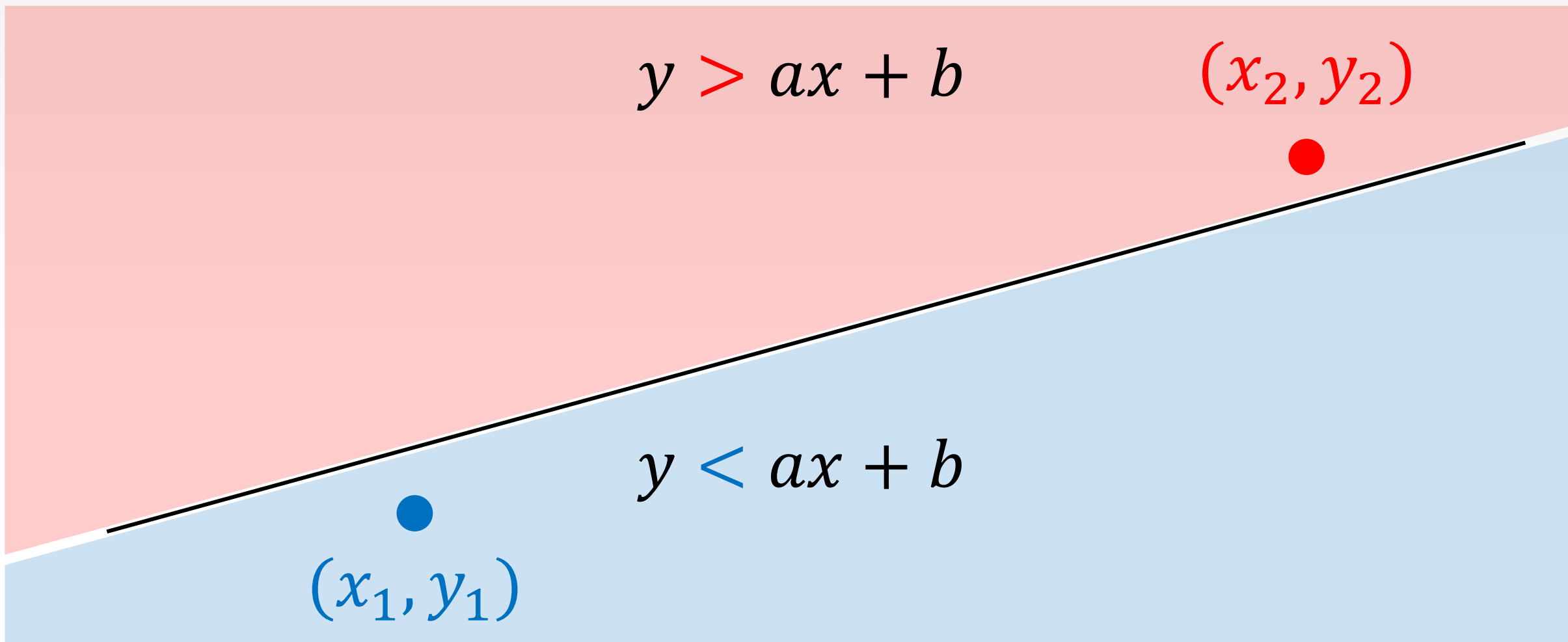
$$y = ax + b$$

(x_2, y_2)



분류

- 회귀인데 분류로 쓸 수 있다?

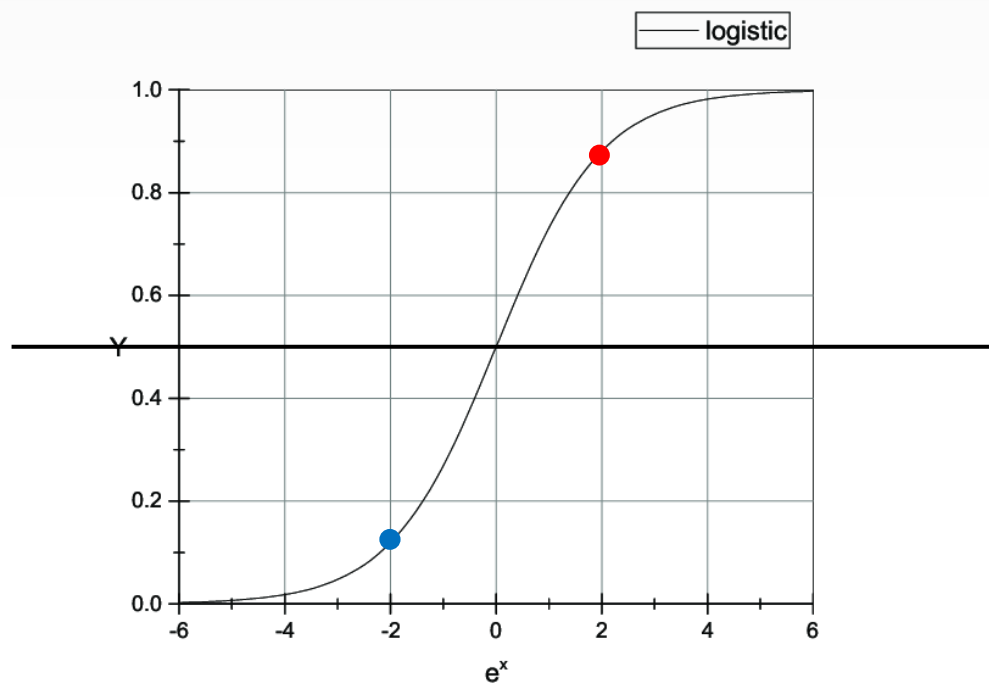



분류

- 로지스틱 회귀 (Logistic Regression)

- 로지스틱 함수

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$




scikit-learn

[Install](#)
[User Guide](#)
[API](#)
[Examples](#)
[More ▾](#)

Prev Up Next

scikit-learn 0.24.2
Other versions

Please **cite us** if you use
the software.

sklearn.linear_model.LogisticRegression

Examples using

sklearn.linear_model.LogisticRegression

sklearn.linear_model.LogisticRegression

```
class sklearn.linear_model.LogisticRegression(penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None) [source]
```

Logistic Regression (aka logit, MaxEnt) classifier.

In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the ‘multi_class’ option is set to ‘ovr’, and uses the cross-entropy loss if the ‘multi_class’ option is set to ‘multinomial’. (Currently the ‘multinomial’ option is supported only by the ‘lbfgs’, ‘sag’, ‘saga’ and ‘newton-cg’ solvers.)

This class implements regularized logistic regression using the ‘liblinear’ library, ‘newton-cg’, ‘sag’, ‘saga’ and ‘lbfgs’ solvers. **Note that regularization is applied by default.** It can handle both dense and sparse input. Use C-ordered arrays or CSR matrices containing 64-bit floats for optimal performance; any other input format will be converted (and copied).

The ‘newton-cg’, ‘sag’, and ‘lbfgs’ solvers support only L2 regularization with primal formulation, or no regularization. The ‘liblinear’ solver supports both L1 and L2 regularization, with a dual formulation only for the L2 penalty. The Elastic-Net regularization is only supported by the ‘saga’ solver.

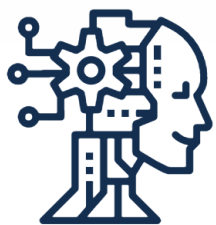
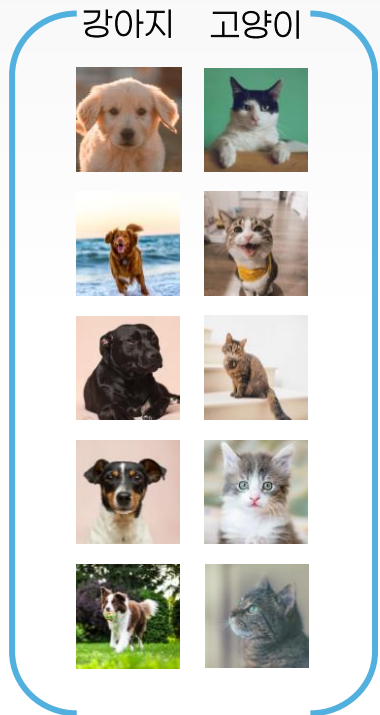
Read more in the [User Guide](#).

Parameter	penalty : {‘l1’, ‘l2’, ‘elasticnet’, ‘none’}, default=‘l2’
s:	Used to specify the norm used in the penalization. The ‘newton-cg’, ‘sag’ and ‘lbfgs’ solvers support only l2 penalties. ‘elasticnet’ is only supported by the ‘saga’ solver. If ‘none’ (not supported by the liblinear solver), no regularization is applied.

분류

- 간단한 예시

- 강아지와 고양이 이미지 분류하기



실제

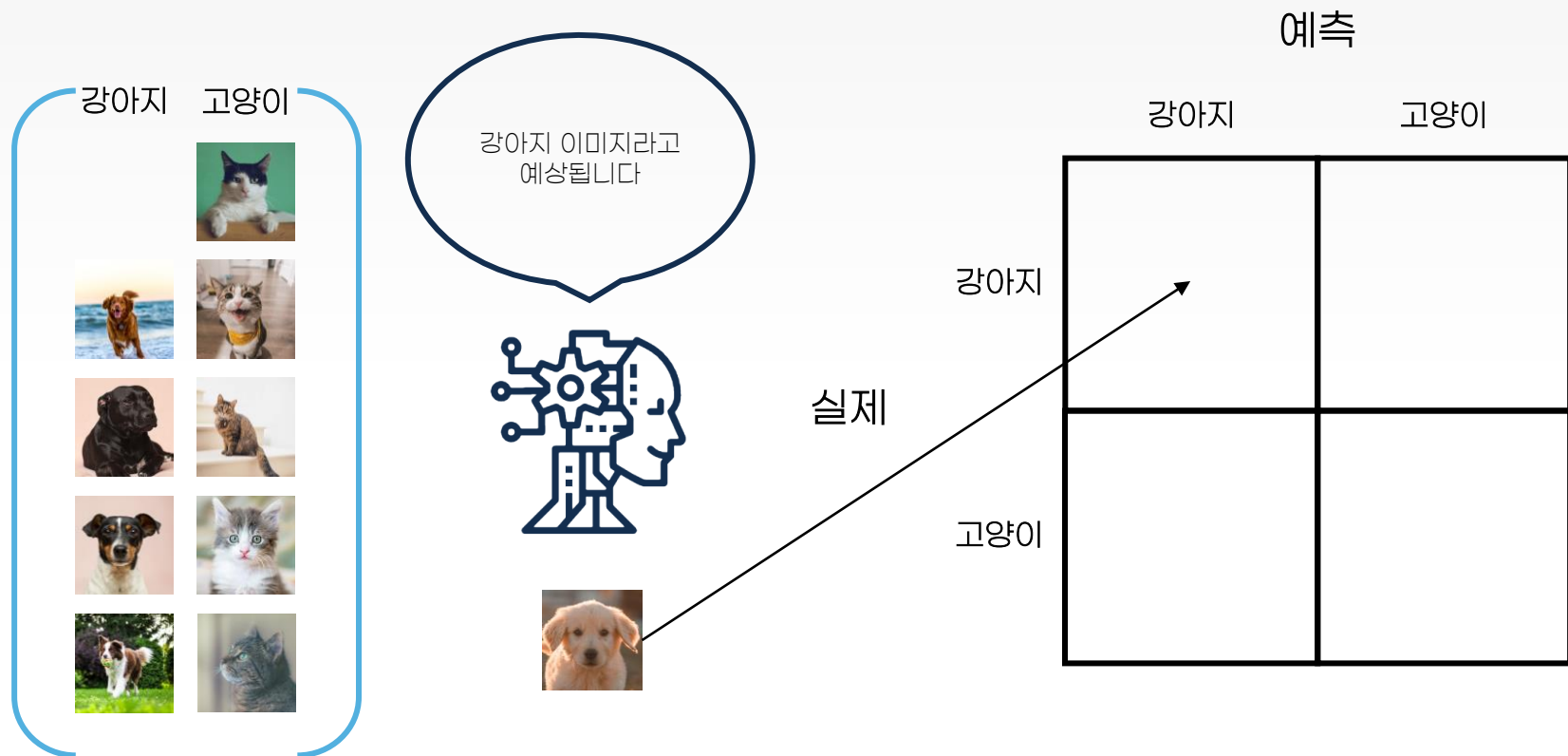
예측

	강아지	고양이
강아지		
고양이		

분류

- 간단한 예시

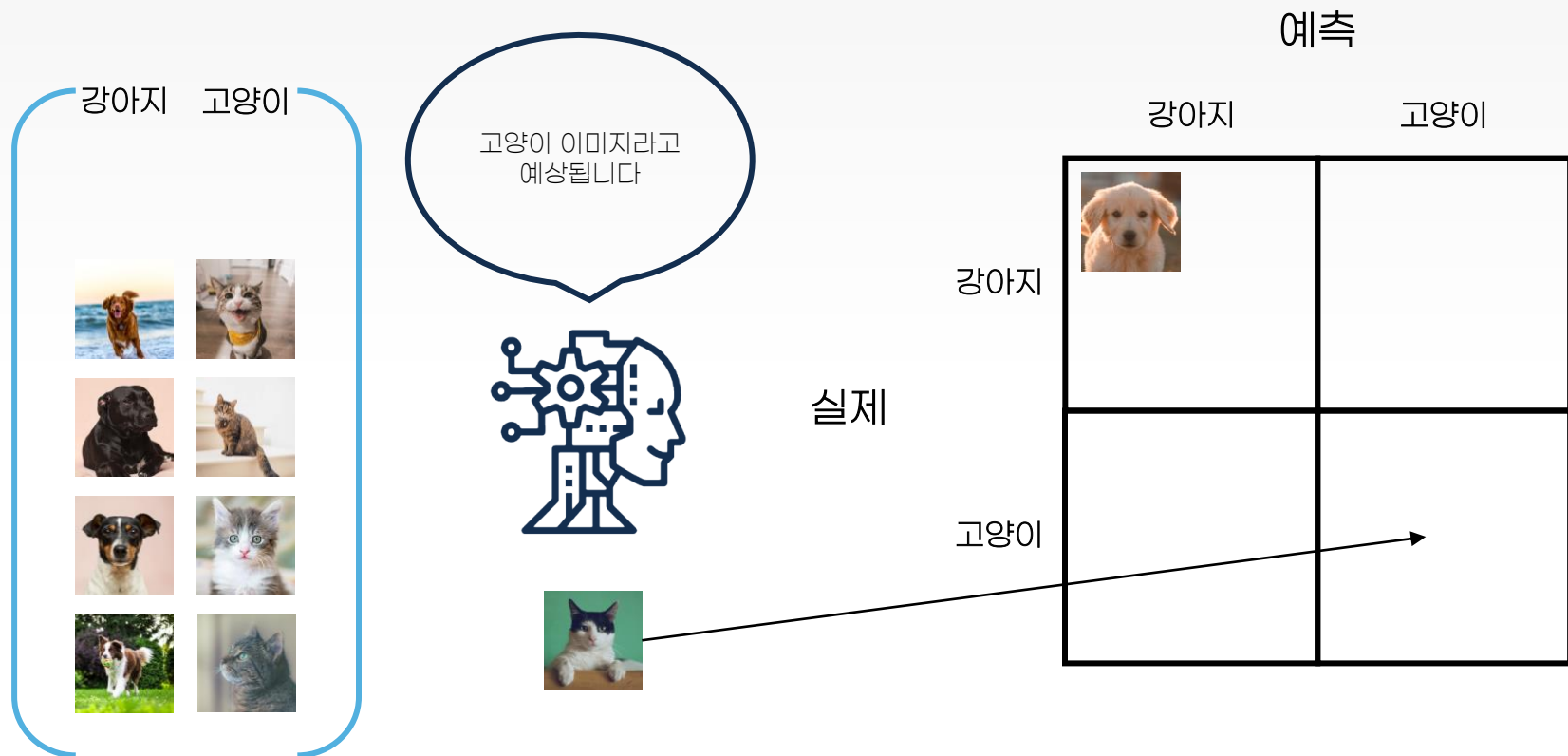
- 강아지와 고양이 이미지 분류하기



분류

● 간단한 예시

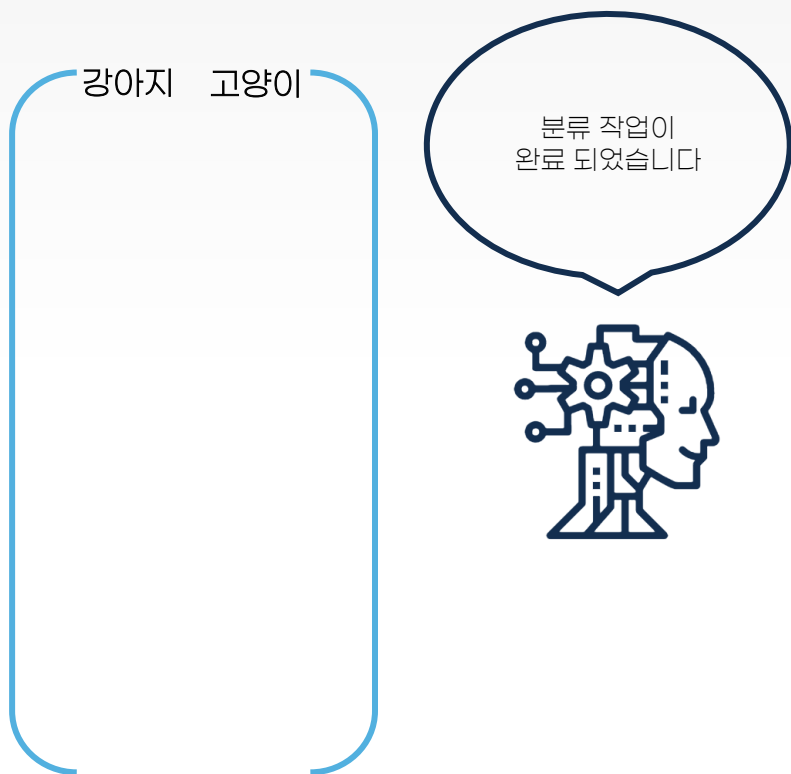
- 강아지와 고양이 이미지 분류하기
- 모든 데이터에 대해서 실제와 예측 비교



분류

- 간단한 예시

- 강아지와 고양이 이미지 분류하기



실제



예측

	강아지	고양이
강아지	   	
고양이	 	  

분류

- 간단한 예시

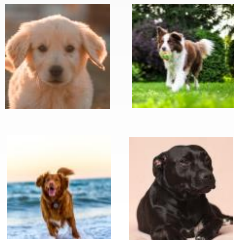
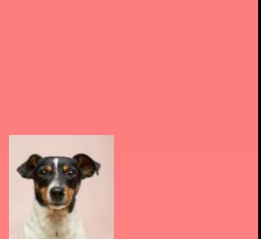
- 강아지와 고양이 이미지 분류하기
 - 이렇게 실제 강아지 이미지를 강아지라고 분류하고 실제 고양이 이미지를 고양이라고 올바르게 분류하는 경우

예측			
		강아지	고양이
실제	강아지		
	고양이	 	 

분류

- 간단한 예시

- 강아지와 고양이 이미지 분류하기
 - 실제 강아지 이미지를 고양이라고 잘 못 예측하거나, 혹은 실제 고양이 이미지를 강아지라고 잘 못 예측하는 경우

		예측	
		강아지	고양이
실제	강아지		
	고양이		

분류

- 오차 행렬 (Confusion matrix)

- 모델이나 알고리즘의 성능을 측정하기 위해서 예측 값과 실제 값을 비교하기 위한 표

		예측	
		○	X
실제	○	True Positive (TP)	False Negative (FN)
	X	False Positive (FP)	True Negative (TN)

분류

- 오차 행렬 (Confusion matrix)

- 모델이나 알고리즘의 성능을 측정하기 위해서 예측 값과 실제 값을 비교하기 위한 표
- True Positive (TP): 실제로 ○이고 예측도 ○라고 하는 경우

		예측	
		○	X
실제	○	True Positive (TP)	False Negative (FN)
	X	False Positive (FP)	True Negative (TN)

분류

- 오차 행렬 (Confusion matrix)

- 모델이나 알고리즘의 성능을 측정하기 위해서 예측 값과 실제 값을 비교하기 위한 표
- True Positive (TP) : 실제로 ○이고 예측도 ○라고 하는 경우
- False Negative (FN) : 실제로는 ○지만, 예측은 X라고 하는 경우

		예측	
		○	X
실제	○	True Positive (TP)	False Negative (FN)
	X	False Positive (FP)	True Negative (TN)

분류

- 오차 행렬 (Confusion matrix)

- 모델이나 알고리즘의 성능을 측정하기 위해서 예측 값과 실제 값을 비교하기 위한 표
- True Positive (TP) : 실제로 ○이고 예측도 ○라고 하는 경우
- False Negative (FN) : 실제로는 ○지만, 예측은 X라고 하는 경우
- **False Positive (FP)** : 실제로는 X지만, 예측은 ○라고 하는 경우

		예측	
		○	X
실제	○	True Positive (TP)	False Negative (FN)
	X	False Positive (FP)	True Negative (TN)

분류

● 오차 행렬 (Confusion matrix)

- 모델이나 알고리즘의 성능을 측정하기 위해서 예측 값과 실제 값을 비교하기 위한 표
- True Positive (TP) : 실제로 ○이고 예측도 ○라고 하는 경우
- False Negative (FN) : 실제로는 ○지만, 예측은 X라고 하는 경우
- False Positive (FP) : 실제로는 X지만, 예측은 ○라고 하는 경우
- True Negative (TN)** : 실제로 X이고 예측도 X라고 하는 경우

		예측	
		○	X
실제	○	True Positive (TP)	False Negative (FN)
	X	False Positive (FP)	True Negative (TN)

분류

- 정확도 (Accuracy)

- 모델이 올바르게 분류한 비율 (%)
- 즉 실제로 ○인걸 ○라고 예측하고, 실제로 X인걸 X라고 예측한 비율
- 계산 공식
 - $$Acc = \frac{TP+TN}{TP+FN+FP+TN}$$

		예측	
		○	X
실제	○	True Positive (TP)	False Negative (FN)
	X	False Positive (FP)	True Negative (TN)

분류

- 정밀도 (Precision)

- 모델이 ○라고 분류한 데이터 중에 실제로 ○였던 데이터의 비율

- 계산 공식

- $$Precision = \frac{TP}{TP+FP}$$

- 예시

- ○ ○ ○ ○ ○

		예측	
		○	X
실제	○	True Positive (TP)	False Negative (FN)
	X	False Positive (FP)	True Negative (TN)

분류

- 재현도 (Recall)

- 실제로 ○인 데이터 중에 모델이 ○라고 분류했던 데이터의 비율

- 계산 공식

- $Recall = \frac{TP}{TP+FN}$

- 예시

- ○ ○ ○ ○ ○ ○

		예측	
		○	X
실제	○	True Positive (TP)	False Negative (FN)
	X	False Positive (FP)	True Negative (TN)

분류

● 그 외 평가 지표

• F1 Score

- Precision과 Recall의 조화평균
- 일반적으로 Precision과 Recall 간의 Trade-off가 있기 때문에

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

• FPR (False Positive Rate)

- 실제로 X인 데이터 중에서 모델이 ○라고 예측한 비율

• ROC (Receiver Operating Characteristic)

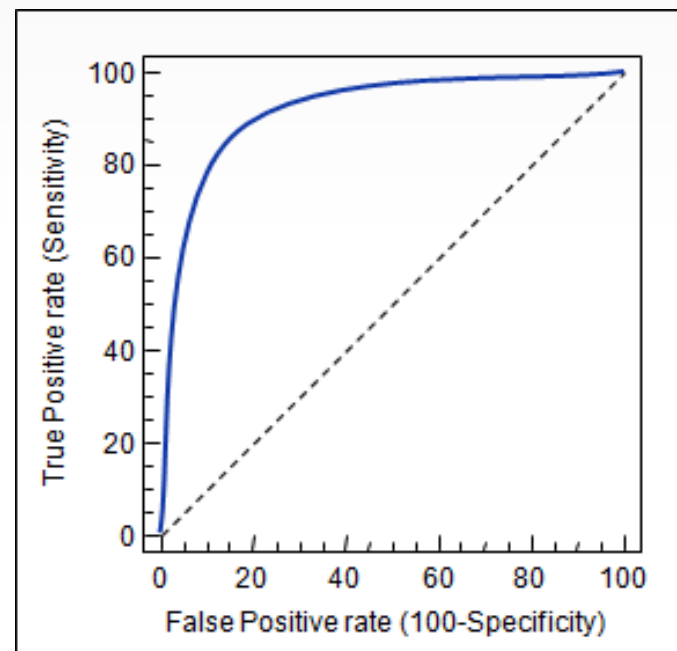
- Recall과 FPR 간의 변화를 시각화한 곡선

• AUC (Area Under Curve)

- ROC 곡선 아래의 면적

• ...

		예측	
		○	X
실제	○	True Positive (TP)	False Negative (FN)
	X	False Positive (FP)	True Negative (TN)
		FPR	



ROC와 AUC

휴식 시간

10분간 쉬고 다시 시작합니다! ☺

<https://open.kakao.com/me/minsuksung>





2021 K-ICT 빅데이터센터

분석 인프라 활용 AI 교육

인공지능 빅데이터 전문가 심화과정

K-근접 이웃 알고리즘 (KNN)

머신러닝 2일차 이론

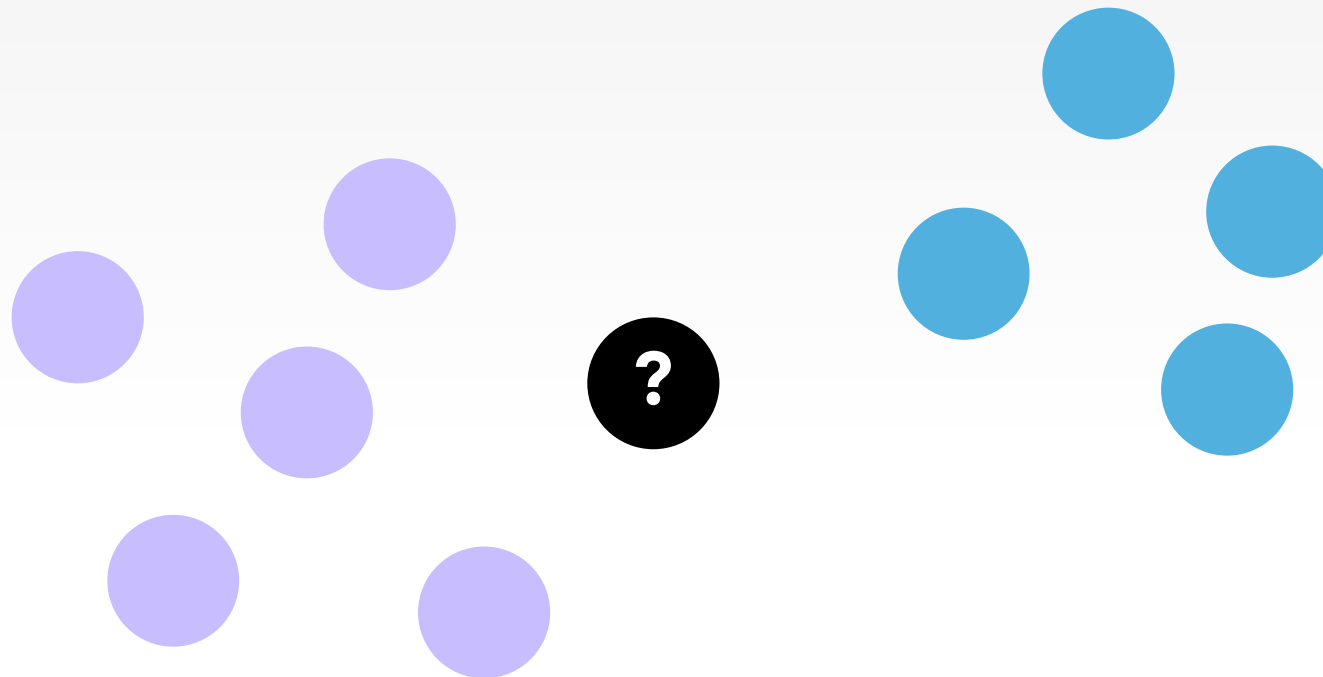


과학기술정보통신부

NIA 한국지능정보사회진흥원

KNN

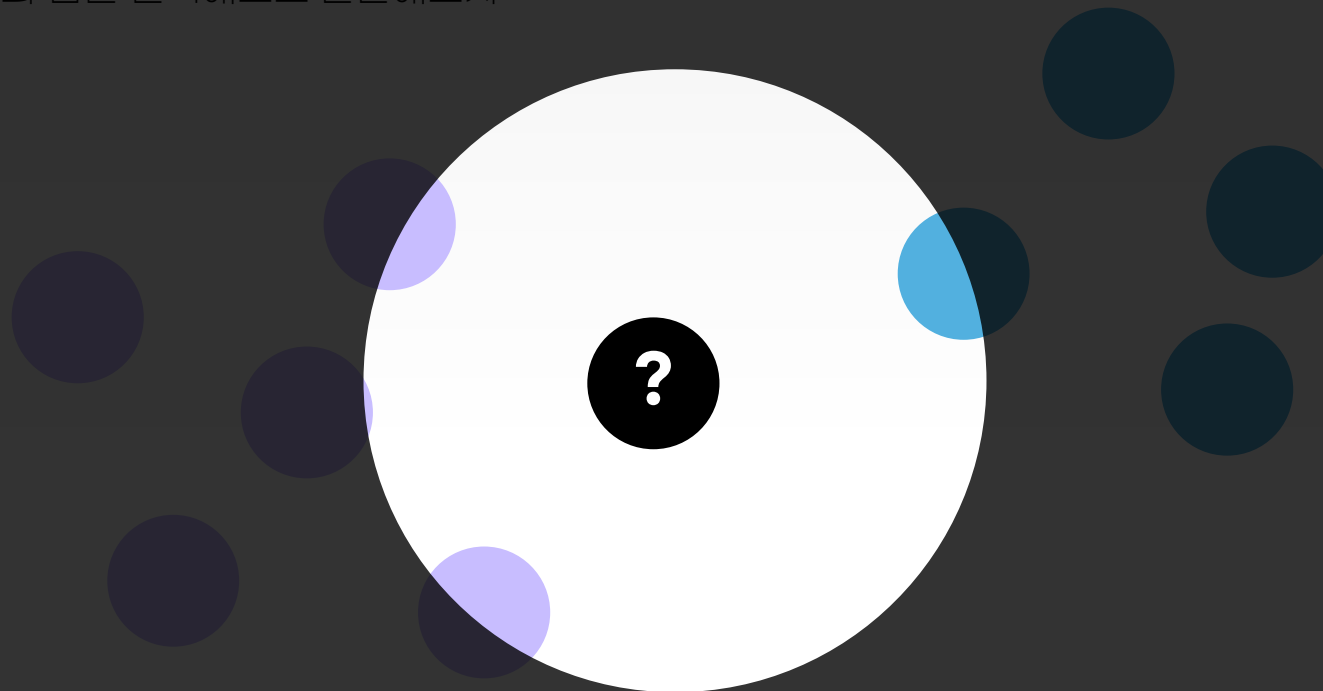
- 가장 직관적인 알고리즘
 - 가장 가까운 데이터를 찾아보자



KNN

- 가장 직관적인 알고리즘

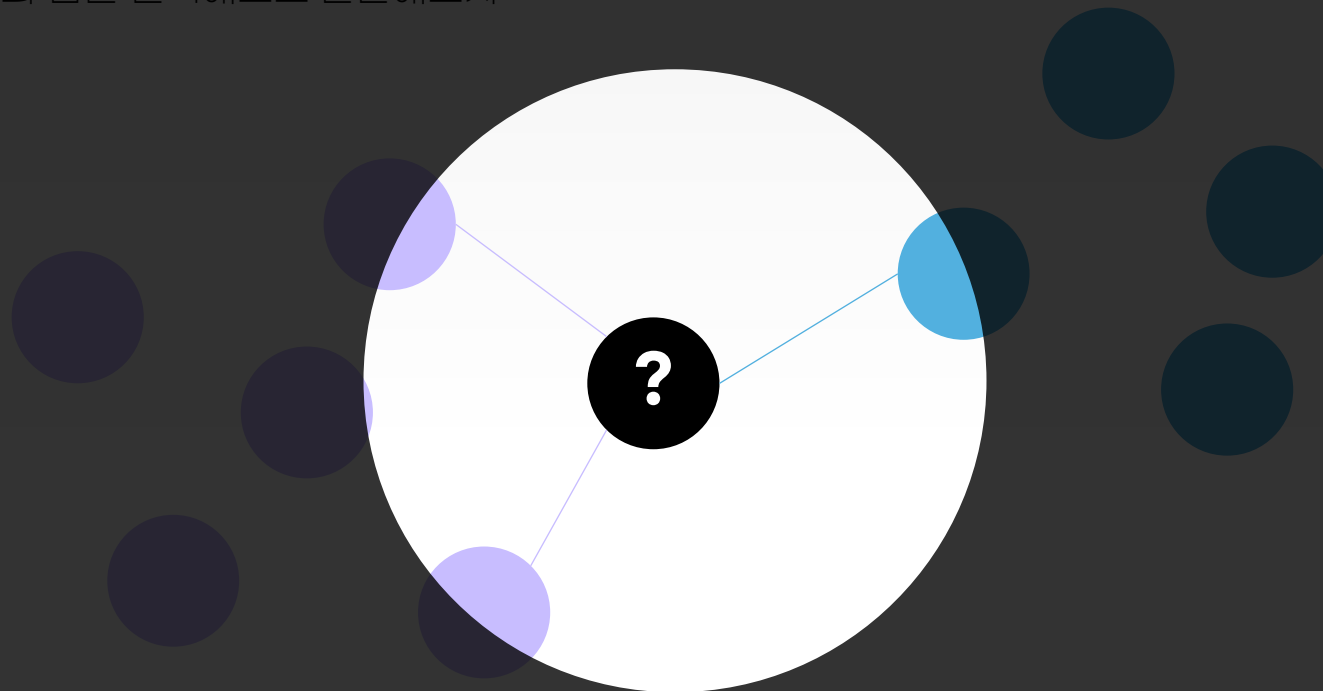
- 가장 가까운 데이터를 찾아보자
- 주변에서 가장 가까운 3개의 점을 선택해보고 판단해보자



KNN

- 가장 직관적인 알고리즘

- 가장 가까운 데이터를 찾아보자
- 주변에서 가장 가까운 3개의 점을 선택해보고 판단해보자



KNN

- 가장 직관적인 알고리즘

- 가장 가까운 데이터를 찾아보자
- 주변에서 가장 가까운 3개의 점($K = 3$)을 선택해보고 판단해보자
 - 일반적으로 K 값을 낮추면 모델의 복잡도가 낮아지고 높이면 모델의 복잡도가 높아짐





sklearn.neighbors.KNeighborsClassifier

```
class sklearn.neighbors.KNeighborsClassifier(n_neighbors=5, *, weights='uniform',
algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None,
**kwargs)
```

[\[source\]](#)

Classifier implementing the k-nearest neighbors vote.

Read more in the [User Guide](#).

Parameter `n_neighbors` : *int, default=5*

s:

Number of neighbors to use by default for `kneighbors` queries.

`weights` : *{'uniform', 'distance'} or callable, default='uniform'*

weight function used in prediction. Possible values:

- 'uniform' : uniform weights. All points in each neighborhood are weighted equally.
- 'distance' : weight points by the inverse of their distance. in this case, closer neighbors of a query point will have a greater influence than neighbors which are further away.
- [callable] : a user-defined function which accepts an array of distances, and returns an array of the same shape containing the weights.

`algorithm` : *{'auto', 'ball_tree', 'kd_tree', 'brute'}, default='auto'*

Algorithm used to compute the nearest neighbors:

- 'ball_tree' will use `BallTree`
- 'kd_tree' will use `KDTree`
- 'brute' will use a brute-force search.

휴식 시간

10분간 쉬고 다시 시작합니다! ☺

<https://open.kakao.com/me/minsuksung>





2021 K-ICT 빅데이터센터

분석 인프라 활용 AI 교육

인공지능 빅데이터 전문가 심화과정

나이브 베이즈 (Naïve Bayes)

머신러닝 2일차 이론



과학기술정보통신부

NIA 한국지능정보사회진흥원

나이브 베이즈

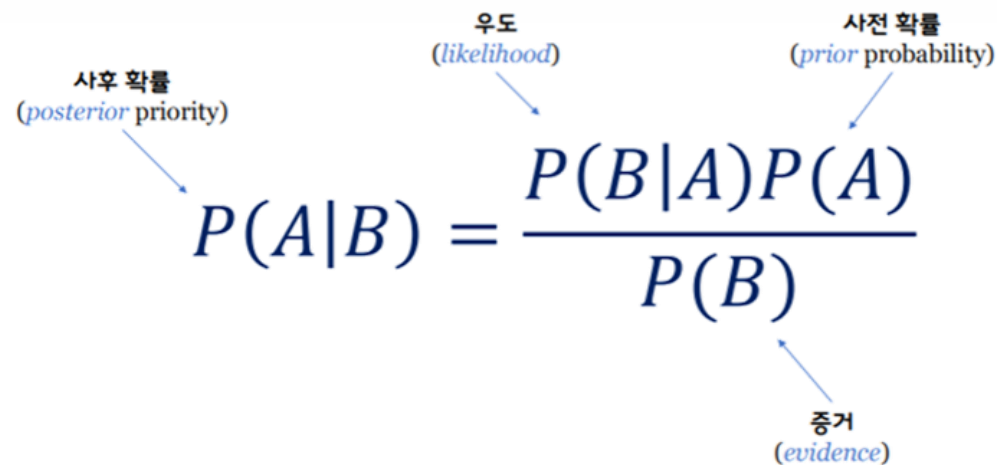
● 베이즈 정리

- $P(A)$: A 가 일어날 확률
- $P(B)$: B 가 일어날 확률
- $P(A|B)$: A 가 일어나고 B 가 일어날 확률
- $P(B|A)$: B 가 일어나고 A 가 일어날 확률

→ 쉽게 생각해서 사건 B 가 주어졌을 때 사건 A 가 일어날 확률인 $P(A|B)$, 조건부 확률과 베이즈 정리를 이용한 분류기

● 왜 나이브 (Naïve) 라고 했을까?

- 데이터의 모든 특징들이 동등하고 독립적이라고 가정


$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

나이브 베이즈

● 장점

- 간단하고 빠르며 효율적
- 노이즈와 누락된 데이터일지라도 성능이 나쁘지 않음
- 훈련을 할 때 데이터의 크기에 상관 없이 잘 동작
- 예측을 위한 추정 확률을 쉽게 얻을 수 있음

● 단점

- 모든 특징이 중요하다고 가정하기 때문에 성능이 안 나올 수 있다
- 수치 특징이 많은 데이터셋에는 이상적이지 않음
- 추정된 확률이 예측된 클래스보다 덜 신뢰

나이프 베이즈

[Install](#) [User Guide](#) [API](#) [Examples](#) [More ▾](#)

[Prev](#) [Up](#) [Next](#)

scikit-learn 0.24.2
[Other versions](#)

Please [cite us](#) if you use the software.

[sklearn.naive_bayes.GaussianNB](#)
[Examples using sklearn.naive_bayes.GaussianNB](#)

[Toggle Menu](#)

sklearn.naive_bayes.GaussianNB

```
class sklearn.naive_bayes.GaussianNB(*, priors=None, var_smoothing=1e-09) \[source\]
```

Gaussian Naive Bayes (GaussianNB)

Can perform online updates to model parameters via [partial_fit](#). For details on algorithm used to update feature means and variance online, see Stanford CS tech report STAN-CS-79-773 by Chan, Golub, and LeVeque:

<http://i.stanford.edu/pub/ctr/reports/cs/tr/79/773/CS-TR-79-773.pdf>

Read more in the [User Guide](#).

Parameters:	priors : array-like of shape (n_classes,) Prior probabilities of the classes. If specified the priors are not adjusted according to the data.
	var_smoothing : float, default=1e-9 Portion of the largest variance of all features that is added to variances for calculation stability. <i>New in version 0.20.</i>
Attributes:	class_count_ : ndarray of shape (n_classes,) number of training samples observed in each class.
	class_prior_ : ndarray of shape (n_classes,) probability of each class.
	classes_ : ndarray of shape (n_classes,)

휴식 시간

10분간 쉬고 다시 시작합니다! ☺

<https://open.kakao.com/me/minsuksung>





2021 K-ICT 빅데이터센터

분석 인프라 활용 AI 교육

인공지능 빅데이터 전문가 심화과정

SVM

머신러닝 2일차 이론



과학기술정보통신부

NIA 한국지능정보사회진흥원

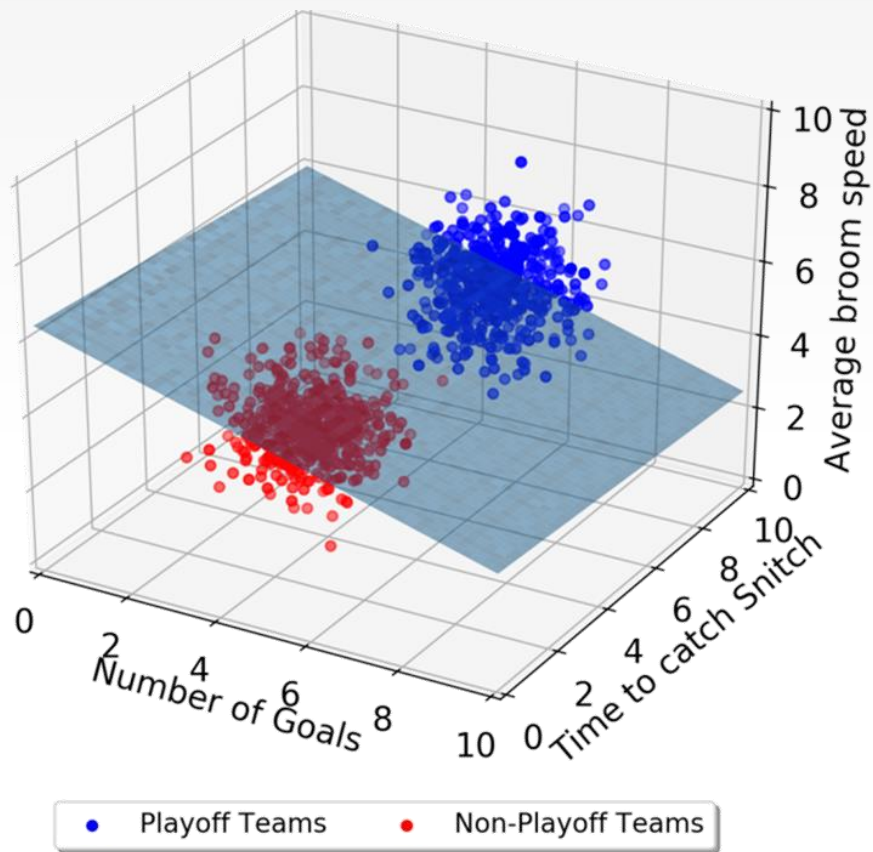
Support Vector Machine

Support Vector Machine

서포트 벡터가 도대체 뭡까요?

SVM

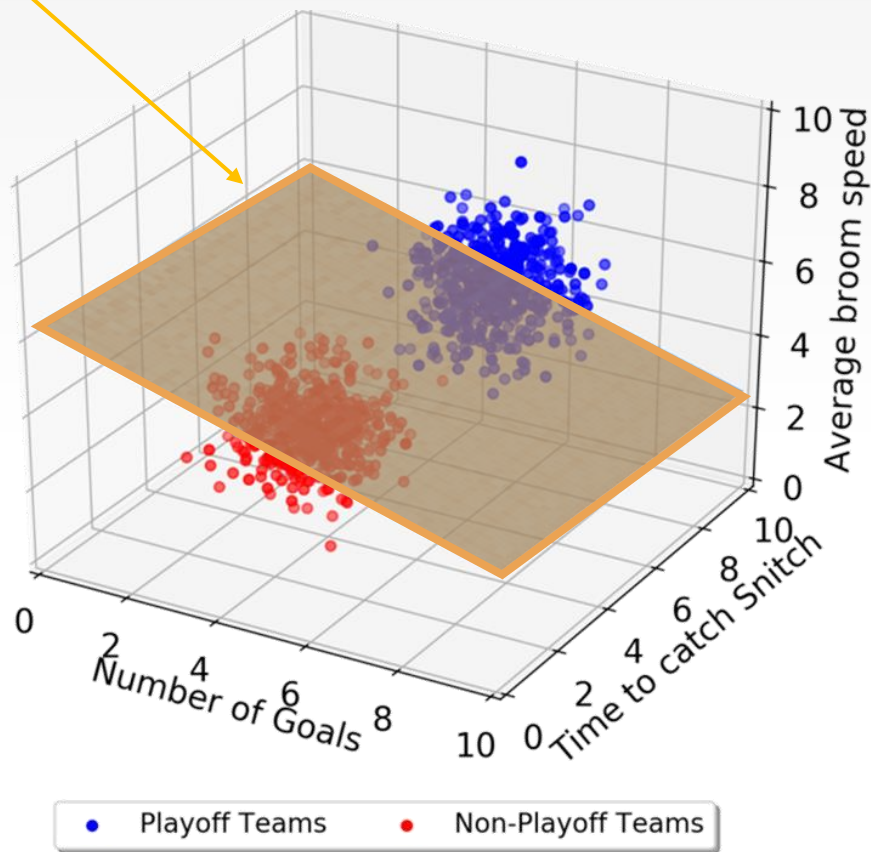
- 딥러닝 이전 최고의 알고리즘



SVM

- 기본 아이디어

- 분류를 위한 적절한 선이나 면을 찾아보자
- 그럼 저런 선이나 면을 어떻게 찾을 수 있지?
- 해답이 바로 Support Vector!!



SVM

- 서포트 벡터 (Support Vector)

- 결정 경계의 위치를 결정짓는 샘플들
- 수학적으로 어떻게 구하는지는 생략

- 마진 (Margin)

- 클래스들 사이의 간격, 즉 여백
- 각 클래스들의 말단에 위치한 데이터들 사이의 거리에 해당

- 라지 마진 (Large Margin) 분류

- 모든 샘플이 결정 경계 바깥쪽으로 올바르게 분류

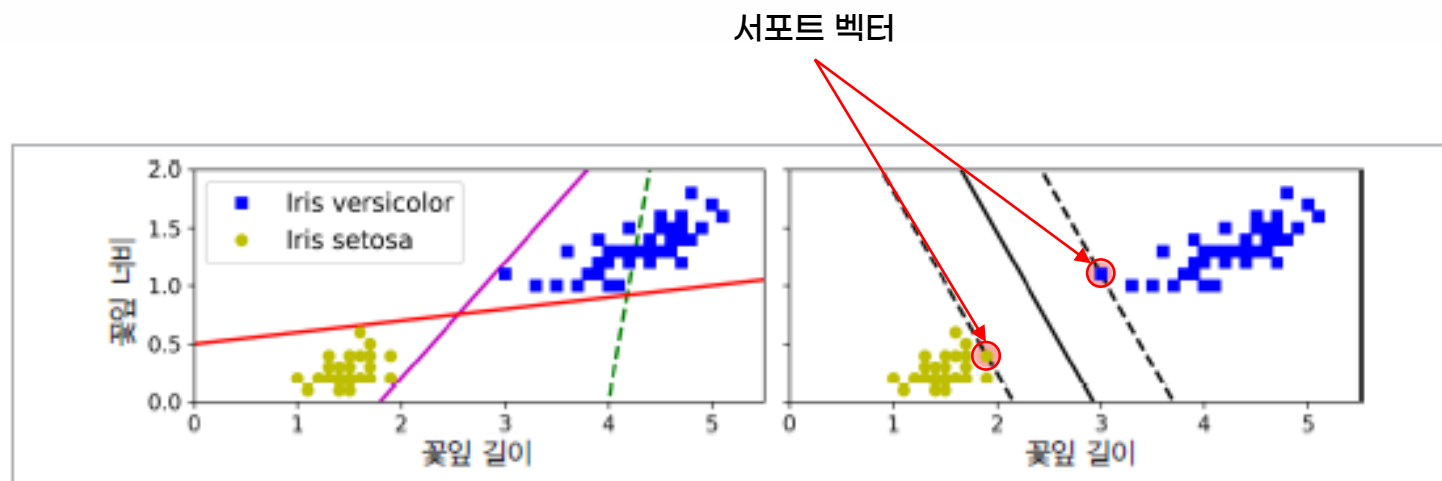
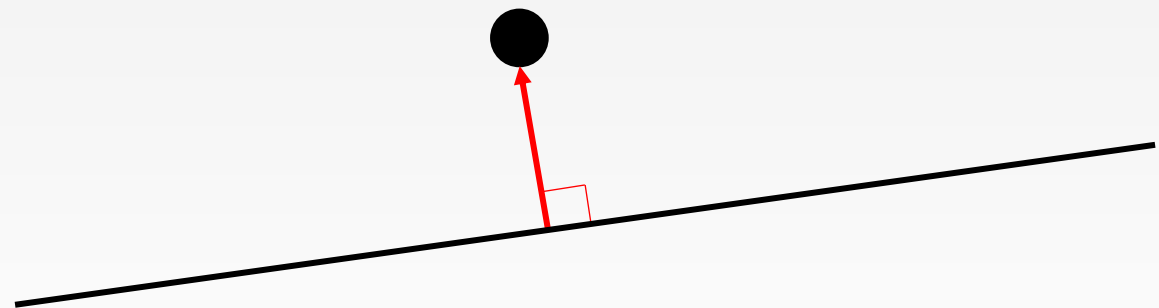


그림 5-1 라지 마진 분류

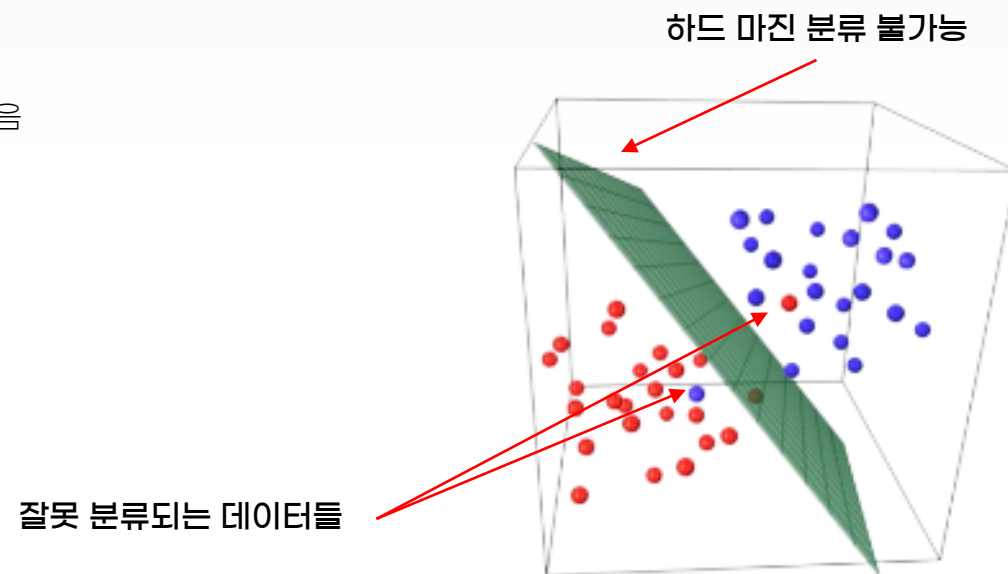
SVM

● 선형 SVM

- 데이터를 선형으로 구분하는 최적의 초평면을 찾는 기법
- 이번 교육에서는 비선형 SVM을 다루지 않겠음

● 선형 SVM의 종류

- 하드 마진(Hard margin) SVM
 - 두 개의 클래스에 대해 최대 마진이 되는 초평면을 찾음
 - 단 하나의 오분류 데이터도 허용하지 않기 때문에 초평면을 찾지 못할 수 있음
- 소프트 마진(Soft margin) SVM
 - 하드 마진 분류에서 초평면이 존재하지 않을 때, 오분류 데이터를 허용하여 찾음



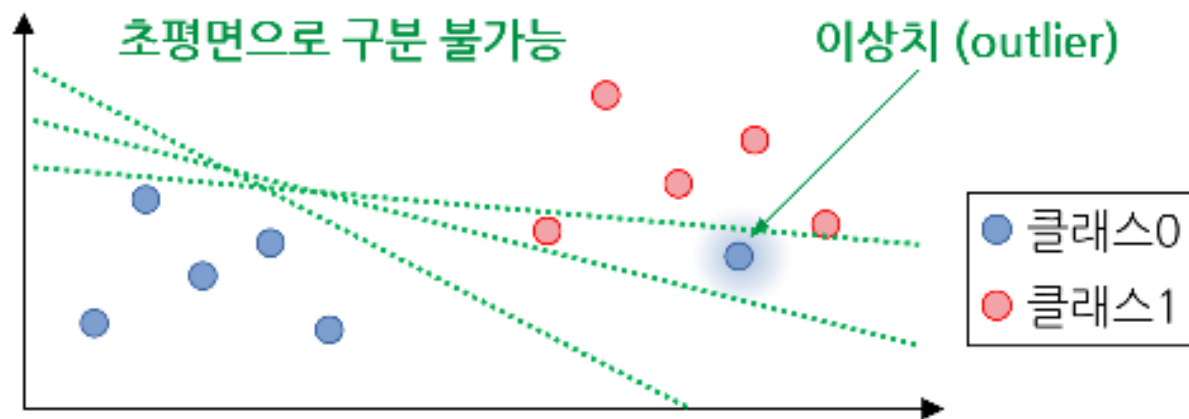
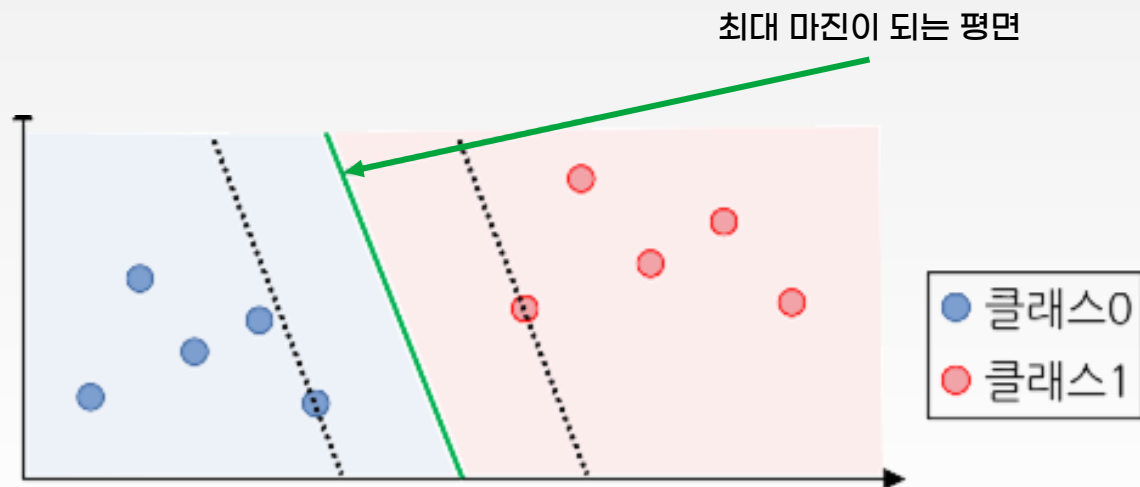
SVM

● 하드 마진 SVM

- 두 개의 클래스에 대해 최대 마진이 되는 초 평면을 찾음
- 모든 훈련 데이터들은 마진의 바깥쪽에 위치
- 데이터들이 정확하게 선형적으로 구분되는 경우에만 분류가 가능
- 이상치에 민감함

● 하드 마진 SVM의 한계

- 모든 경우에 반드시 초평면이 존재하는 것은 아님
 - 데이터가 정확하게 선형적으로 구분되지 않은 경우에는 결정 경계를 찾는 것이 불가능
- 분류 모형이 일반화되기 어려움
 - 이상치가 존재할 경우, 초평면이 없거나 잘 일반화되지 않음



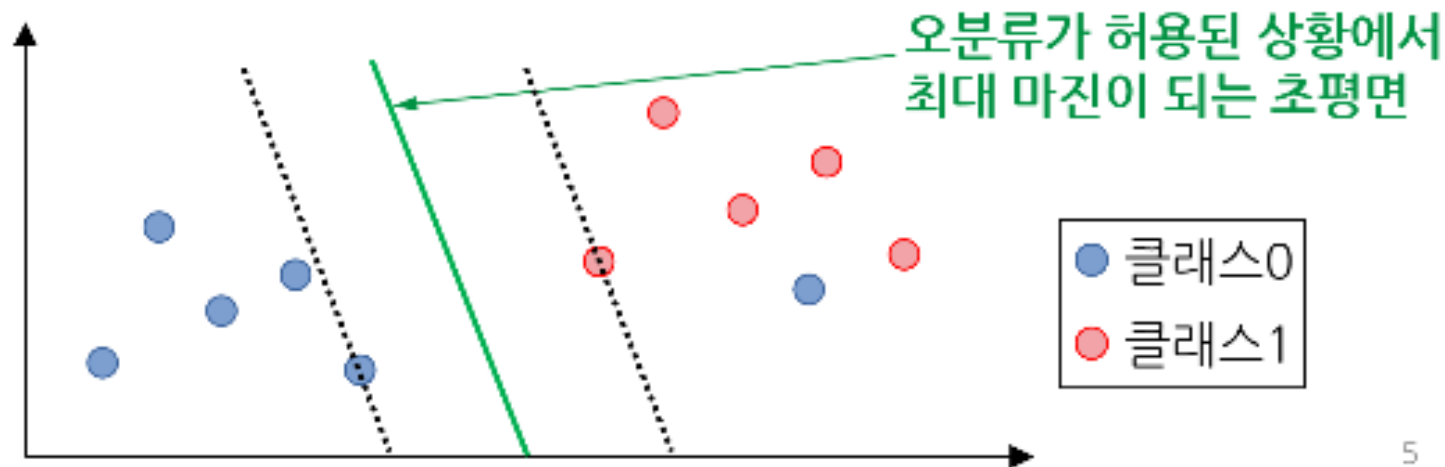
SVM

● 소프트 마진 SVM

- 어느 정도의 오류를 허용하면서 가급적 최대 마진이 되는 초평면을 찾음
- 잘못 분류되는 데이터가 있지만 초평면을 찾을 수 있음
- 과대적합을 방지하거나 줄일 수 있음

● 소프트 마진 SVM의 허용 수준 결정

- 하이퍼파라미터 C (Cost)를 이용하여 허용할 오류의 수준을 결정가능
- 원래 데이터와 다른 클래스로 분류되는 경우를 얼마나 많이 허용할 것인가를 조정하는 값
 - C 의 값이 큰 경우, 오류에 대해서 더 엄격하게 적용
 - 마진이 작아짐
 - 오분류율이 낮아짐
 - 과대적합이 될 수 있음
 - C 의 값이 작은 경우, 오류에 대해서 덜 엄격하게 적용
 - 마진이 커짐
 - 오분류율이 높아짐
 - 과소적합이 될 수 있음



5

SVM

● SVM을 이용한 회귀

- 기본 아이디어는 SVM 분류와 비슷하지만 정반대
- **마진(Margin)** 사이에 가능한 한 많은 샘플이 들어가도록 하자
- ϵ 으로 조절하여 마진 사이의 폭을 조절할 수 있음
 - 마진을 크게 할 경우: 마진 안에서는 훈련 샘플이 추가되어도 모델의 예측에는 영향이 없음
 - 마진을 작게 할 경우: 마진이 작을 경우 훈련 샘플이 추가될 때마다 예측이 달라질 수 있음

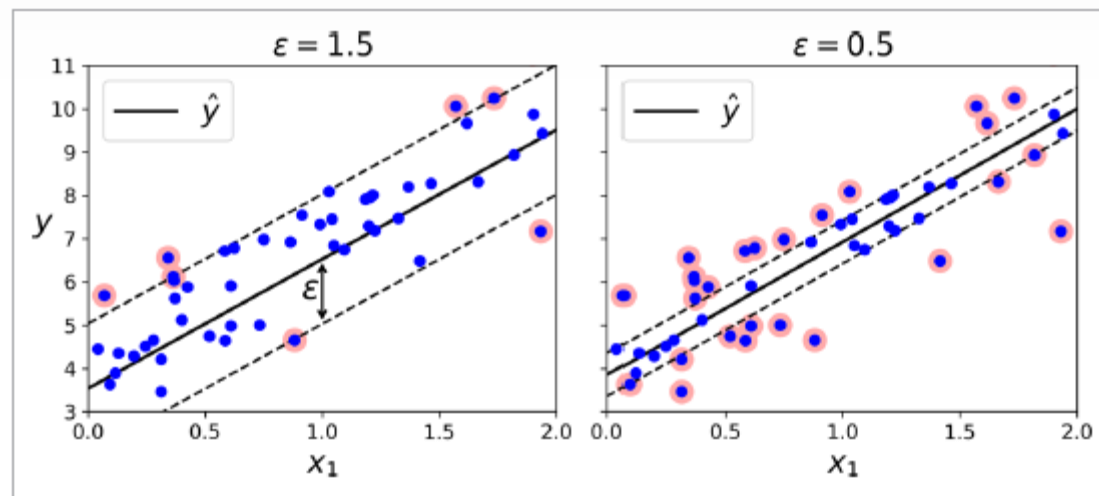


그림 5-10 SVM 회귀

SVM

scikit-learn

Install

User Guide

API

Examples

More ▾

Prev

Up

Next

scikit-learn 0.24.2

Other versions

Please cite us if you use the software.

sklearn.svm.SVC

Examples using

sklearn.svm.SVC

Toggle Menu

sklearn.svm.SVC

```
class sklearn.svm.SVC(*, C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, decision_function_shape='ovr', break_ties=False, random_state=None)
```

[source]

C-Support Vector Classification.

The implementation is based on libsvm. The fit time scales at least quadratically with the number of samples and may be impractical beyond tens of thousands of samples. For large datasets consider using [LinearSVC](#) or [SGDClassifier](#) instead, possibly after a [Nystroem](#) transformer.

The multiclass support is handled according to a one-vs-one scheme.

For details on the precise mathematical formulation of the provided kernel functions and how `gamma`, `coef0` and `degree` affect each other, see the corresponding section in the narrative documentation: [Kernel functions](#).

Read more in the [User Guide](#).

Parameter s:	C : float, default=1.0 Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive. The penalty is a squared l2 penalty.
	kernel : {'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'}, default='rbf' Specifies the kernel type to be used in the algorithm. It must be one of 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed' or a callable. If none is given, 'rbf' will be used. If a callable is given it is used to pre-compute the kernel matrix from data matrices; that matrix should be an array of shape (n_samples, n_samples).
	degree : int, default=3

이론 파트는 여기까지 입니다

휴식 시간

10분간 쉬고 다시 시작합니다! ☺

<https://open.kakao.com/me/minsuksung>





2021 K-ICT 빅데이터센터

분석 인프라 활용 AI 교육

인공지능 빅데이터 전문가 심화과정

머신러닝 예제

머신러닝 2일차 실습



과학기술정보통신부

NIA 한국지능정보사회진흥원

Boston 주택 가격 데이터를 활용한 예제

Browser address bar: 163.152.51.111:8888/notebooks/GitHub/2021-NIA-K-ICT-AI-Lecture/day2/(2일차)%202021년%20NIA%208월%20분석인프라교육%20머신러닝%20-%20실습자료%20(1)%20보스턴%20주택%20가격%20데이터

Jupyter (2일차) 2021년 NIA 8월 분석인프라교육 머신러닝 - 실습자료 (1) 보스턴 주택 가격 데이터 Last Checkpoint: 2분 전 (autosaved)

File Edit View Insert Cell Kernel Navigate Widgets Help Not Trusted Python 3

Contents

- 1 데이터 불러오기
- 2 데이터 살펴보기
- 3 데이터 전처리
- 4 피쳐 엔지니어링
- 5 모델링
- 6 모델 평가
- 7 결과 예측

K-ICT 빅데이터센터
BIG DATA CENTER

분석 인프라 활용 AI 교육
인공지능 빅데이터 전문가 심화과정

강사 [성민석](mailto:minsung@korea.ac.kr)
minsung@korea.ac.kr



실습 시간

휴식 시간

10분간 쉬고 다시 시작합니다! ☺

<https://open.kakao.com/me/minsuksung>





인공지능 빅데이터 전문가 심화과정

머신러닝 예제

머신러닝 2일차 실습

Iris 붓꽃 데이터를 통해서 알아보는 예제


주요 메뉴: 163.152.51.111:8888/notebooks/GitHub/2021-NIA-K-ICT-AI-Lecture/day2/(2일차)%202021년%20NIA%208월%20분석인프라교육%20머신러닝%20-%20실습자료%20(2)%20아이리스%20붓꽃%20데이터.ipynb

jupyter (2일차) 2021년 NIA 8월 분석인프라교육 머신러닝 - 실습자료 (2) 아이리스 붓꽃 데이터 Last Checkpoint: 2시간 전 (autosaved) Logout

File Edit View Insert Cell Kernel Navigate Widgets Help Not Trusted Python 3

Contents

- 1 데이터 불러오기
- 2 데이터 살펴보기
- 3 데이터 전처리
- 4 피쳐 엔지니어링
- 5 모델링
- 6 모델 평가
- 7 결과 예측




분석 인프라 활용 AI 교육

인공지능 빅데이터 전문가 심화과정


강사 성민석
minsung@korea.ac.kr

iris setosa




petal sepal

iris versicolor



petal sepal

iris virginica



petal sepal

Iris 데이터를 통해서 알아보는 머신러닝 예제 (2)

실습 시간

휴식 시간

10분간 쉬고 다시 시작합니다! ☺

<https://open.kakao.com/me/minsuksung>





2021 K-ICT 빅데이터센터

분석 인프라 활용 AI 교육

인공지능 빅데이터 전문가 심화과정

머신러닝 예제

머신러닝 2일차 실습



과학기술정보통신부

NIA 한국지능정보사회진흥원

와인 품질 데이터를 통해서 알아보는 예제


주요 탭: 163.152.51.111:8888/notebooks/GitHub/2021-NIA-K-ICT-AI-Lecture/day2/(2일차)%202021년%20NIA%208월%20분석인프라교육%20머신러닝%20-%20실습자료%20(3)%20와인%20품질%20데이터.ipynb

jupyter (2일차) 2021년 NIA 8월 분석인프라교육 머신러닝 - 실습자료 (3) 와인 품질 데이터 Last Checkpoint: 2시간 전 (autosaved) Logout

File Edit View Insert Cell Kernel Navigate Widgets Help Not Trusted Python 3


Contents

- 1 데이터 불러오기
- 2 데이터 살펴보기
- 3 데이터 전처리
- 4 피쳐 엔지니어링
- 5 모델링
- 6 모델 평가
- 7 결과 예측



분석 인프라 활용 AI 교육
인공지능 빅데이터 전문가 심화과정

강사 [성민석](mailto:minsung@korea.ac.kr)
minsung@korea.ac.kr



와인 품질 데이터를 통해서 알아보는 머신러닝 예제 (3)

Table of Contents

- [1 데이터 불러오기](#)
- [2 데이터 살펴보기](#)
- [3 데이터 전처리](#)
- [4 피쳐 엔지니어링](#)
- [5 모델링](#)
- [6 모델 평가](#)
- [7 결과 예측](#)

실습 시간

휴식 시간

10분간 쉬고 다시 시작합니다! ☺

<https://open.kakao.com/me/minsuksung>





2021 K-ICT 빅데이터센터

분석 인프라 활용 AI 교육

인공지능 빅데이터 전문가 심화과정

머신러닝 예제

머신러닝 2일차 실습



과학기술정보통신부

NIA 한국지능정보사회진흥원


한국 코스피 지수 데이터를 통해서 알아보는 예제

jupyter (2일차) 2021년 NIA 8월 분석인프라교육 머신러닝 - 실습자료 (4) 한국 코스피 지수 데이터 Last Checkpoint: 2시간 전 (autosaved) Logout

File Edit View Insert Cell Kernel Navigate Widgets Help Trusted Python 3


Contents

- 1 데이터 불러오기
- 2 데이터 살펴보기
- 3 데이터 전처리
- 4 피쳐 엔지니어링
- 5 모델링
- 6 모델 평가
- 7 결과 예측



분석 인프라 활용 AI 교육
인공지능 빅데이터 전문가 심화과정

강사 [성민석](mailto:minsung@korea.ac.kr)
minsung@korea.ac.kr



KOSPI 지수를 통해서 알아보는 머신러닝 예제 (4)

Table of Contents

- [1 데이터 불러오기](#)
- [2 데이터 살펴보기](#)

실습 시간

실습 파트는 여기까지 입니다

휴식 시간

10분간 쉬고 다시 시작합니다! ☺

<https://open.kakao.com/me/minsuksung>



일반적인 머신러닝 학습과정

※ 주의: 실제 머신러닝 파이프라인과 다소 상이할 수 있습니다

- 이번 교육 때 집중한 부분



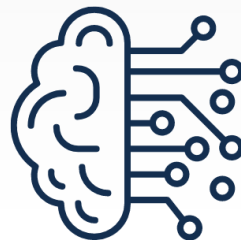
데이터 수집



데이터 전처리
(Data Preprocessing)



피처 엔지니어링
(Feature Engineering)



모델링
(Modeling)



하이퍼파라미터 튜닝
(Hyperparameter Tuning)



평가
(Testing)

오늘 배운 내용

● 이론

- 회귀
- 분류
- KNN
- 나이브 베이즈
- SVM

● 실습

- Boston 데이터를 통해서 알아보는 머신러닝 예제 (1)
- Iris 데이터를 통해서 알아보는 머신러닝 예제 (2)
- KOSPI 지수를 통해서 알아보는 머신러닝 예제 (3)
- 와인 데이터를 통해서 알아보는 머신러닝 예제 (4)

오늘 강의는 여기까지 입니다



QnA

<https://open.kakao.com/me/minsungsung>



감사합니다





오늘 진행한 모든 자료는 아래 링크에서 확인하실 수 있습니다

<https://github.com/minsuk-sung/2021-NIA-K-ICT-AI-Lecture>

참고 도서

※ 주의: 개인적인 추천이며 출판사로부터 받은 광고가 아닙니다

