LOGO

BigData Engineering

12주차: Model Evaluation

강의 : 신경섭

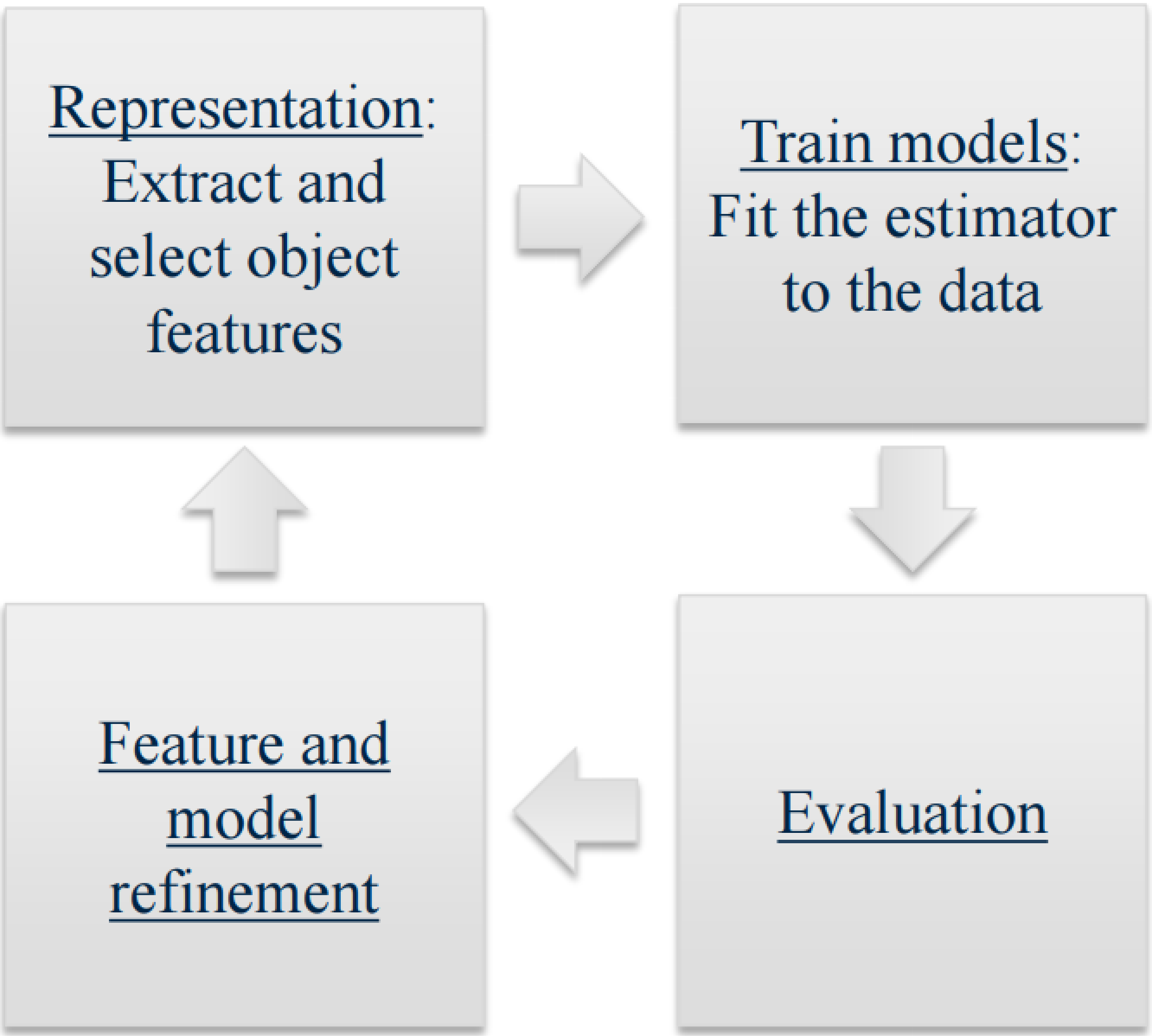
We covered…

- Data management/Visualization by python
  - Numpy, pandas, data acquisition
- Machine learning workflow with data
- EDA (Exploratory Data Analysis)
- Supervised learning
  - k-NN classifier
  - logistic regression based binary classification
  - Support vector machine
  - Decision tree
  - Random Forest

Today's Subjects

- Model Evaluation
  - Accuracy
  - Precision
  - Recall

Represent / Train / Evaluate / Refine Cycle



Evaluation

- Different applications have very different goals
- Accuracy is widely used, but many others are possible, e.g.:
  - User satisfaction (Web search)
  - Amount of revenue (e-commerce)
  - Increase in patient survival rates (medical)

Evaluation

- It's very important to choose evaluation methods that match the goal of your application.
- Compute your selected evaluation metric for multiple different models.
- Then select the model with 'best' value of evaluation metric.

Accuracy with imbalanced classes

- Suppose you have two classes:
  - Relevant (R): the positive class
  - Not_Relevant (N): the negative class
- Out of 1000 randomly selected items, on average
  - One item is relevant and has an R label
  - The rest of the items (999 of them) are not relevant and labelled N.
- Recall that:

$$\text{Accuracy} = \frac{\#\text{correct predictions}}{\#\text{total instances}}$$

Accuracy with imbalanced classes

- You build a classifier to predict relevant items, and see that its accuracy on a test set is 99.9%.
  - Wow! Amazingly good, right?
- For comparison, suppose we had a "dummy" classifier that didn't look at the features at all, and always just blindly predicted the most frequent class (i.e. the negative N class).
- Assuming a test set of 1000 instances, what would this dummy classifier's accuracy be?

Dummy classifiers

- Some commonly-used settings for the strategy parameter for DummyClassifier in scikit-learn:
  - most_frequent : predicts the most frequent label in the training set.
  - stratified : random predictions based on training set class distribution.
  - uniform : generates predictions uniformly at random.
  - constant : always predicts a constant label provided by the user.

Binary prediction outcomes

	Predicted negative	**Predicted** positive
True negative	TN	FP
True positive	FN	TP

Label 1 = positive class
(class of interest)

Label 0 = negative class
(everything else)

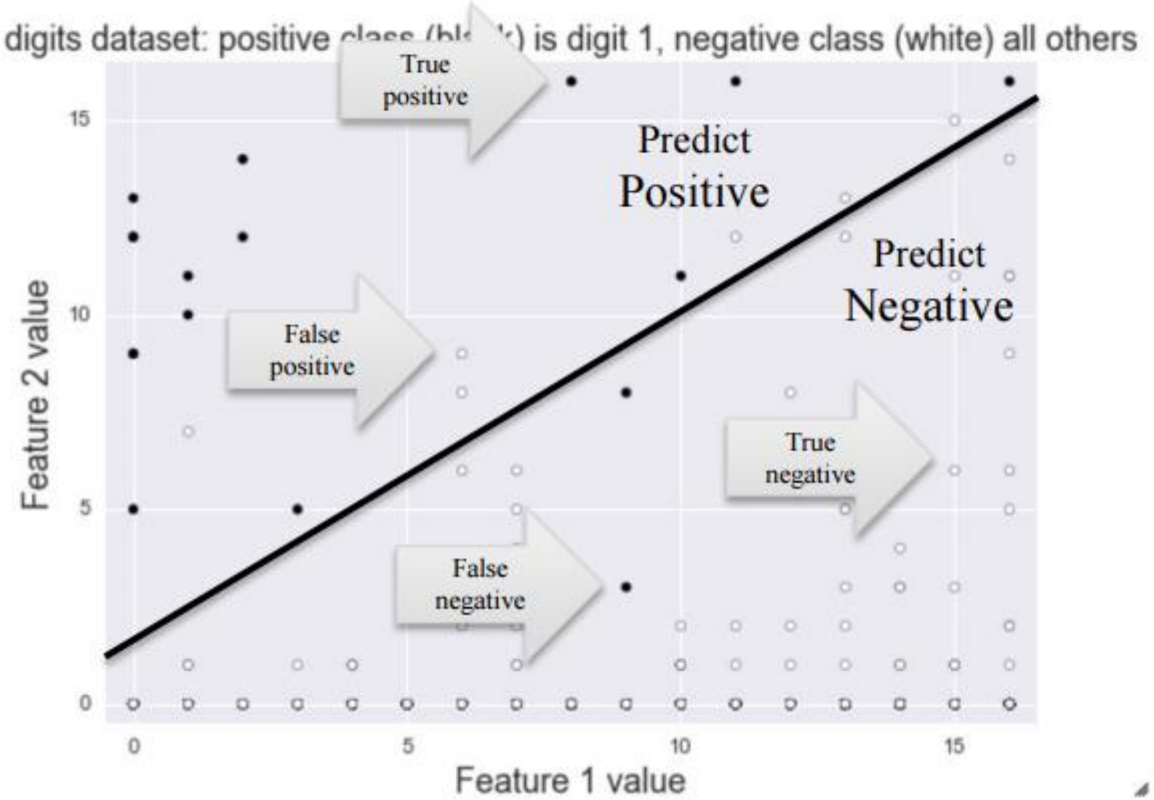
TP = true positive
FP = false positive (Type I error)
TN = true negative
FN = false negative (Type II error)

Confusion Matrix for Binary Prediction Task

True negative	TN = 400	FP = 7	
True positive	FN = 17	TP = 26	
	Predicted negative	Predicted positive	$N = 450$

Always look at the confusion matrix for your classifier.

Visualization of Different Error Types



TN = 429	FP = 6
FN = 2	TP = 13

Model Evaluation Metric: Accuracy

True negative	TN = 400	FP = 7	
True positive	FN = 17	TP = 26	
	Predicted negative	Predicted positive	$N = 450$

$$\text{Accuracy} = \frac{TN+TP}{TN+TP+FN+FP}$$

$$= \frac{400+26}{400+26+17+7}$$

$$= 0.95$$

Model Evaluation Metric: Classification Error

True negative	TN = 400	FP = 7	
True positive	FN = 17	TP = 26	
	Predicted negative	Predicted positive	$N = 450$

$$\text{ClassificationError} = \frac{FP + FN}{TN + TP + FN + FP}$$

$$= \frac{7+17}{400+26+17+7}$$

$$= 0.060$$

Model Evaluation Metric: Precision

True negative	TN = 400	FP = 7	
True positive	FN = 17	TP = 26	
	Predicted negative	Predicted positive	$N = 450$

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

$$= \frac{26}{26+7}$$

$$= 0.79$$

Model Evaluation Metric: Recall

True negative	$TN = 400$	$FP = 7$	
True positive	$FN = 17$	$TP = 26$	
	Predicted negative	Predicted positive	$N = 450$

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

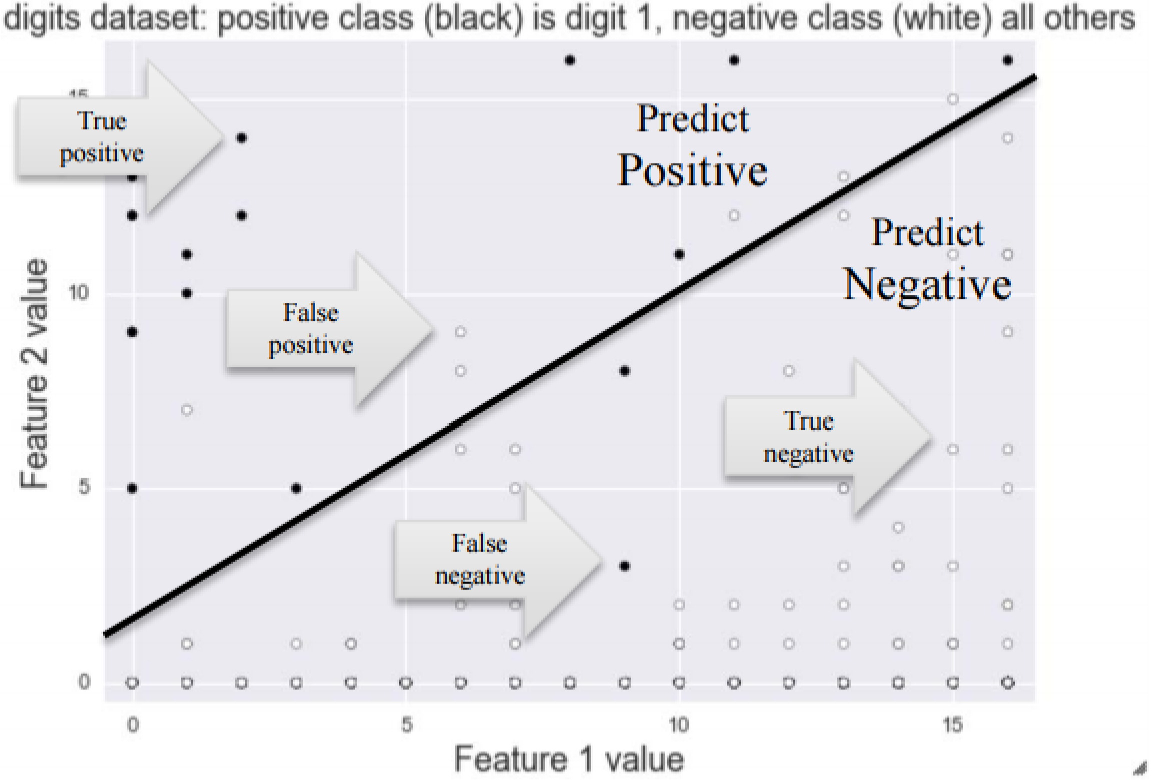
$$= \frac{26}{26+17}$$

$$= 0.60$$

Recall is also known as:

- True Positive Rate (TPR)
- Sensitivity
- Probability of detection

The Precision-Recall tradeoff

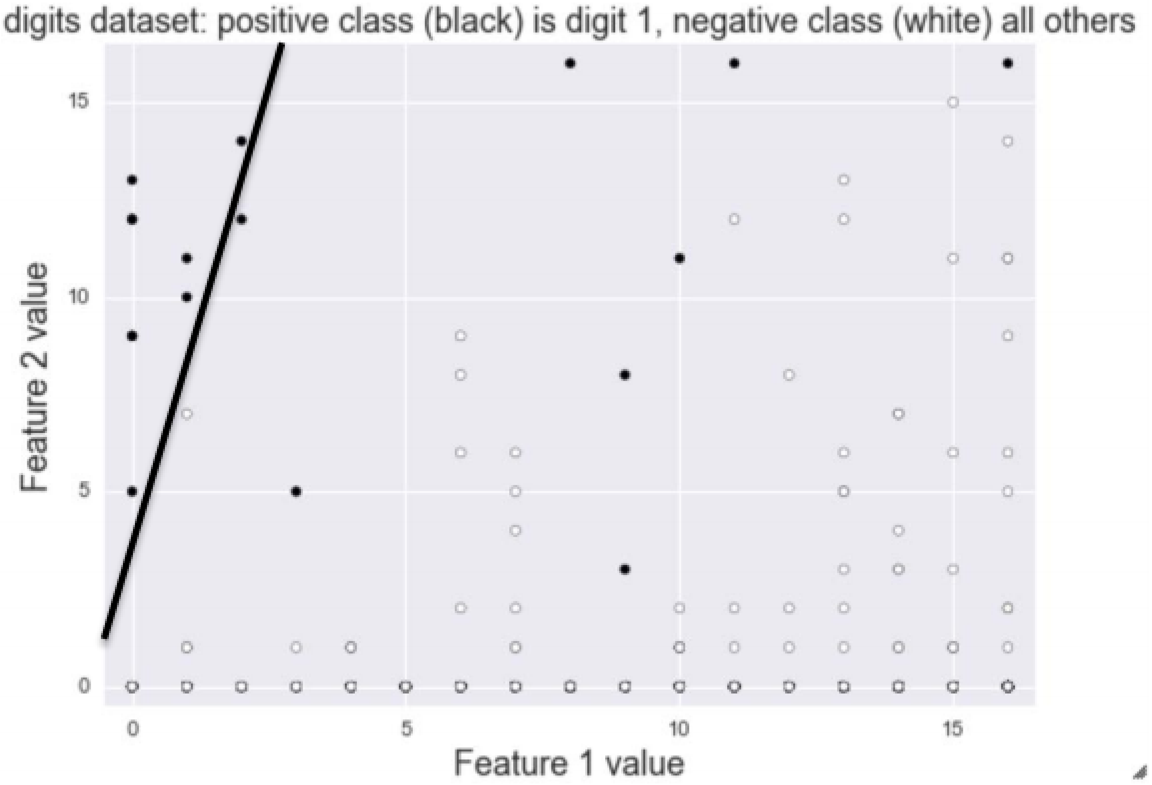


TN = 429	FP = 6
FN = 2	TP = 13

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{13}{19} = 0.68$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{13}{15} = 0.87$$

The Precision-Recall tradeoff

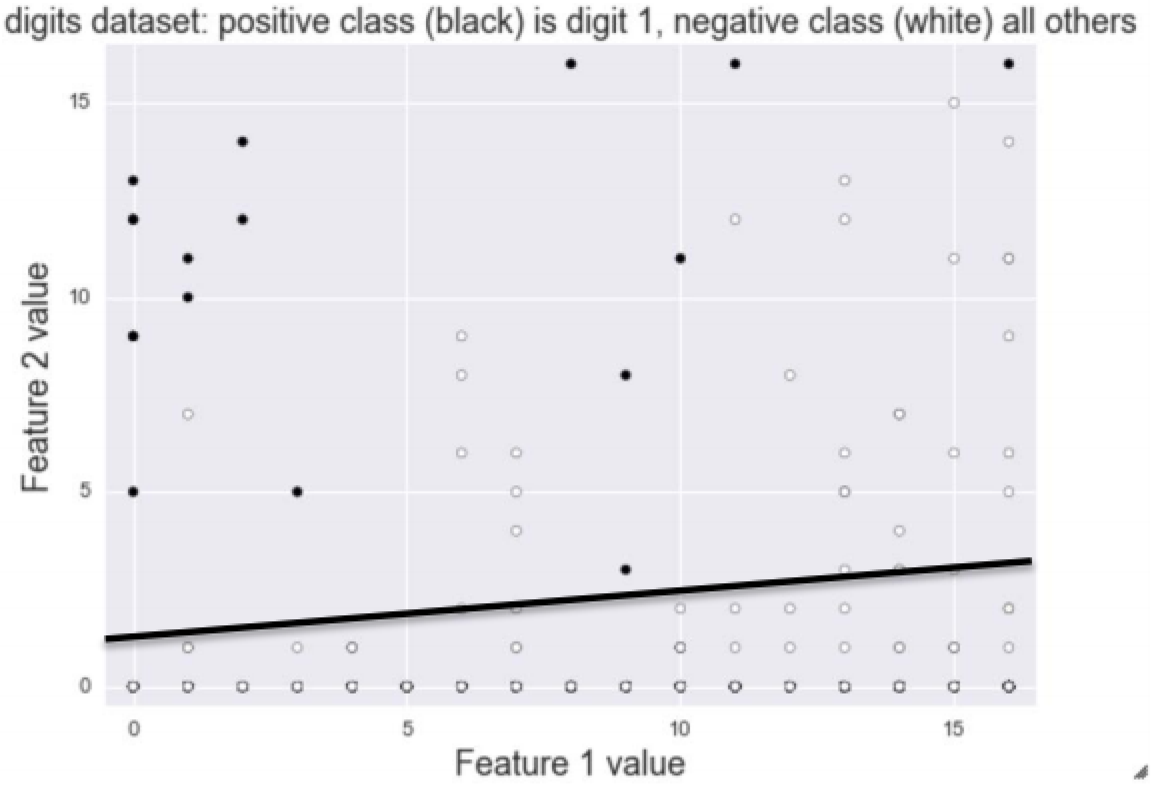


TN = 435	FP = 0
FN = 8	TP = 7

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{7}{7} = 1.00$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{7}{15} = 0.47$$

The Precision-Recall tradeoff



TN = 408	FP = 27
FN = 0	TP = 15

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{15}{42} = 0.36$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{15}{15} = 1.00$$

The Precision-Recall tradeoff

- Recall-oriented machine learning tasks:
  - Search and information extraction in legal discovery
  - Tumor detection
  - Often paired with a human expert to filter out false positives

- Precision-oriented machine learning tasks:
  - Search engine ranking, query suggestion
  - Document classification
  - Many customer-facing tasks (users remember failures!)