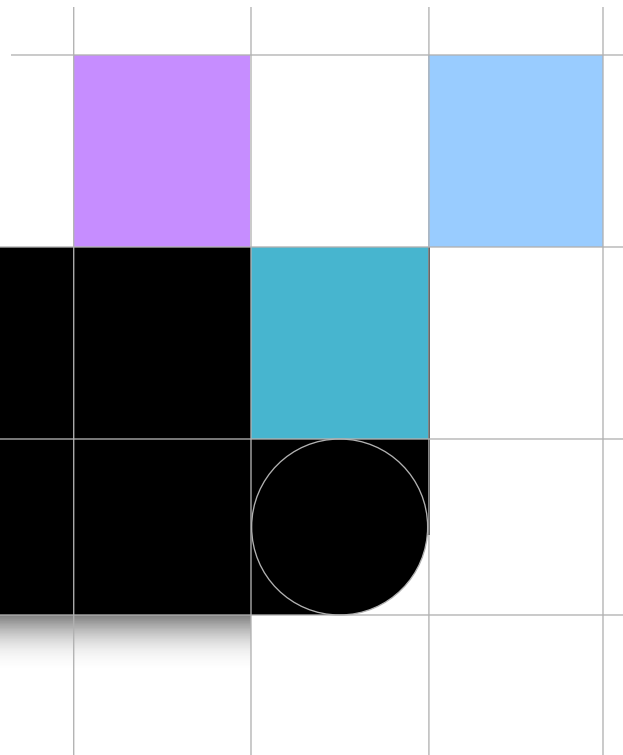


2021 머신러닝 여름 스터디

모델의 성능평가



Sejong Oh
Dankook University

- 예측 모델은 어떤 기준으로 성능을 평가하는지에 따라 성능이 달리 측정될 수 있다
- 예측 모델을 어떤 분야에 적용할 것인가에 따라 모델의 성능에 기대하는 부분이 달라질 수 있다
 - 예) 질병 진단 모델의 경우는 평균 예측 정확도가 높은 모델 보다는 환자를 정상으로 오분류하는 비율이 낮은 모델을 원함
- 따라서 다양한 성능평가 기준 및 측정 방법에 대해 알고 있어야 한다.
- Regression, classification 에 따라 여러 측정 기준이 존재

1. Regression 모델 평가

- Mean Squared Error (MSE)

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

손실 (loss)
실제 *예측값*

⇒ 제곱하면 낮지 커져서
이해 힘들

- Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n}}$$

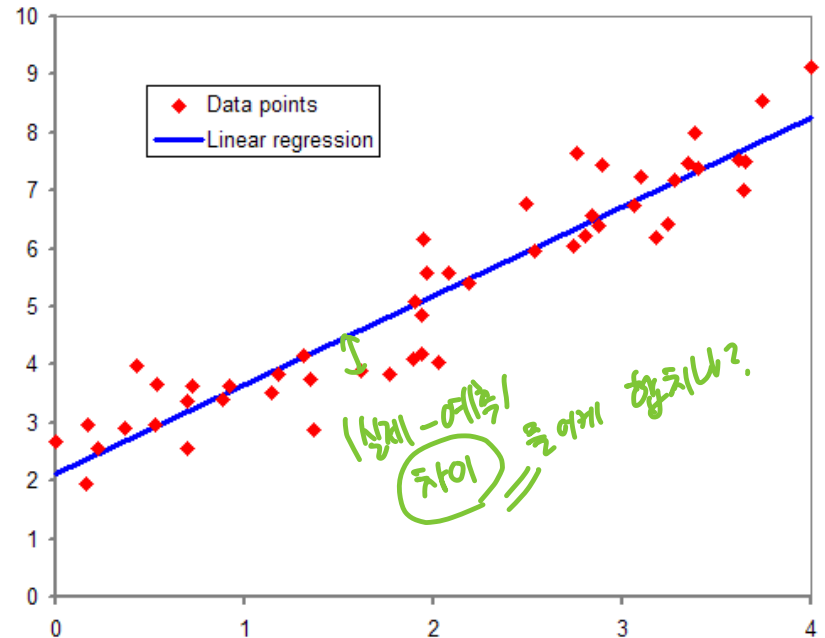
*예측값 ↑
평균*

*root
되려!!*

- Mean Absolute Error (MAE)

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i|$$

같은 양도



2. classification 모델 평가

- Binary classification

accuracy (ACC)

$$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN}$$

sensitivity, recall, hit rate, or true positive rate (TPR)

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN} = 1 - FNR$$

specificity, selectivity or true negative rate (TNR)

$$TNR = \frac{TN}{N} = \frac{TN}{TN + FP} = 1 - FPR$$

F1 score

is the harmonic mean of precision and sensitivity:

$$F_1 = 2 \times \frac{PPV \times TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN}$$

		Actual	
		+	-
		(진짜환자)	(진짜정상인)
Predict	+	True Positive (a)	False Positive (c)
	-	False Negative (b)	True Negative (d)

민감도

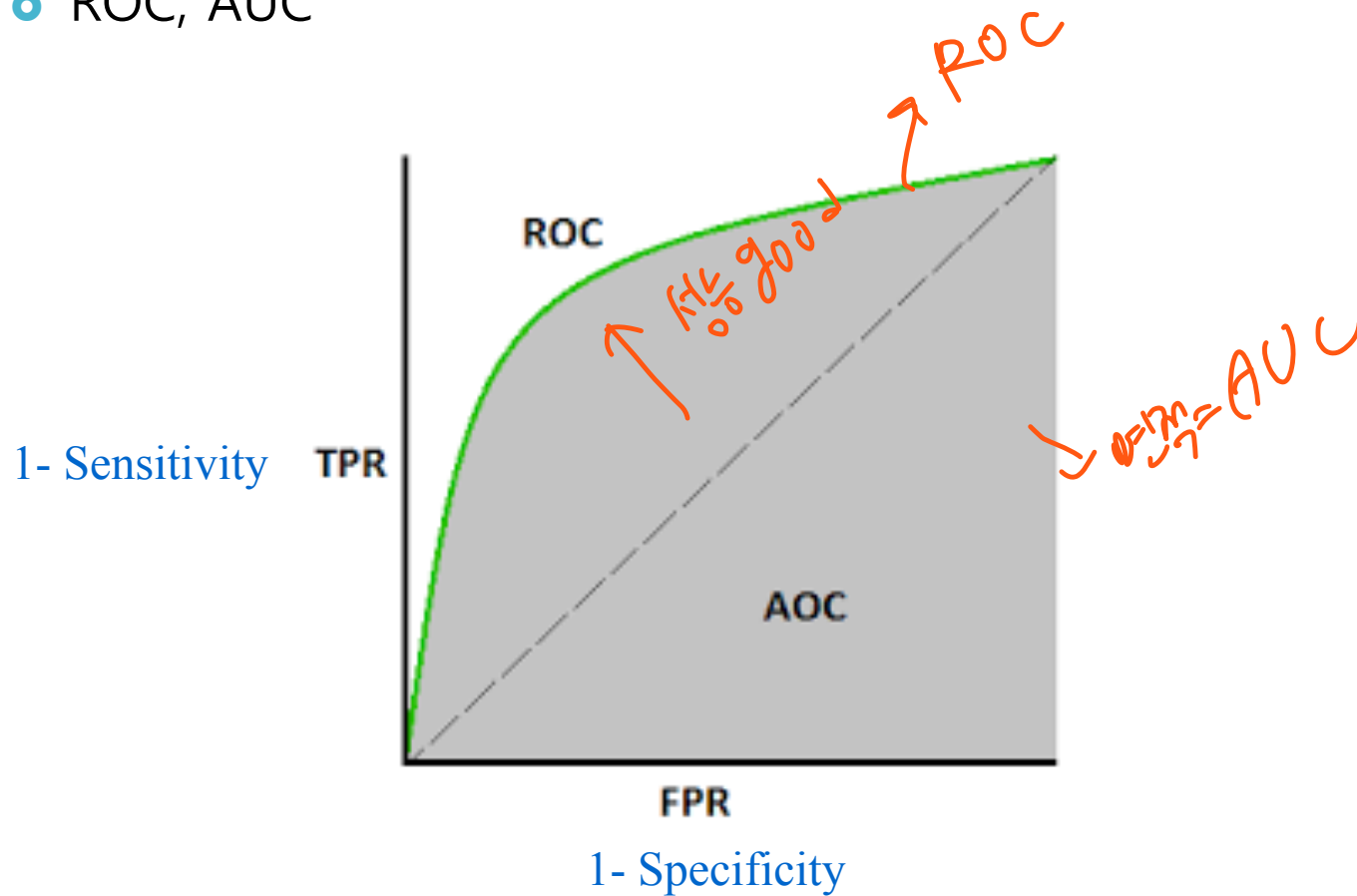
특이도

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

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

2. classification 모델 평가

- Binary classification
 - ROC, AUC



2. classification 모델 평가

- Multi-class classification

① Accuracy is possible

② Transform multi-class to binary class

평가
ACC 가능...

2 변환 해야지...

```
> levels(iris$Species)
[1] "setosa" "versicolor" "virginica"
```

```
> levels(iris$Species)
[1] "setosa" "versicolor" "virginica"
```

```
> levels(iris$Species)
[1] "setosa" "versicolor" "virginica"
```

} => 2개씩...

3. Metrics package

<u>accuracy</u>	Accuracy
<u>ae</u>	Absolute Error
<u>ape</u>	Absolute Percent Error
<u>apk</u>	Average Precision at k
<u>auc</u>	Area under the ROC curve (AUC)
<u>bias</u>	Bias
<u>ce</u>	Classification Error
<u>f1</u>	F1 Score
<u>fbeta_score</u>	F-beta Score
<u>ll</u>	Log Loss
<u>logLoss</u>	Mean Log Loss
<u>mae</u>	Mean Absolute Error
<u>mape</u>	Mean Absolute Percent Error
<u>mapk</u>	Mean Average Precision at k
<u>mase</u>	Mean Absolute Scaled Error
<u>mdae</u>	Median Absolute Error
<u>MeanQuadraticWeightedKappa</u>	Mean Quadratic Weighted Kappa
<u>mse</u>	Mean Squared Error
<u>msle</u>	Mean Squared Log Error
<u>params_binary</u>	Inherit Documentation for Binary Classification Metrics
<u>params_classification</u>	Inherit Documentation for Classification Metrics
<u>params_regression</u>	Inherit Documentation for Regression Metrics
<u>percent_bias</u>	Percent Bias
<u>precision</u>	Precision
<u>rae</u>	Relative Absolute Error
<u>recall</u>	Recall

3. Metrics package

<u>rmse</u>	Root Mean Squared Error
<u>rmsle</u>	Root Mean Squared Log Error
<u>rrse</u>	Root Relative Squared Error
<u>rse</u>	Relative Squared Error
<u>ScoreQuadraticWeightedKappa</u>	Quadratic Weighted Kappa
<u>se</u>	Squared Error
<u>sle</u>	Squared Log Error
<u>smape</u>	Symmetric Mean Absolute Percentage Error
<u>sse</u>	Sum of Squared Errors

3. Metrics package

```
library(Metrics)

# regression
library(mlbench)
Boston <- na.omit(BostonHousing)
head(Boston)

model.lm <- lm(medv~., data=Boston)
pred <- predict(model.lm, Boston)

mae(Boston$medv, pred)
mse(Boston$medv, pred)
rmse(Boston$medv, pred)
```

```
> mae(Boston$medv, pred)
[1] 3.270863
> mae(Boston$medv, pred)
[1] 3.270863
> mse(Boston$medv, pred)
[1] 21.89483
> rmse(Boston$medv, pred)
[1] 4.679191
```

3. Metrics package

```
# classification
data("PimaIndiansDiabetes2", package = "mlbench")
Pima <- na.omit(PimaIndiansDiabetes2)
head(Pima)

library(e1071)
my.iris <- subset(iris, Species != 'setosa')
model.svm <- svm(Species~., data=my.iris)
pred <- predict(model.svm, my.iris)

accuracy(my.iris$Species, pred)
recall(as.integer(my.iris$Species)-2, as.integer(pred)-2)
precision(as.integer(my.iris$Species)-2, as.integer(pred)-2)
f1(as.integer(my.iris$Species)-2, as.integer(pred)-2)
```

→ 7.5% 정확도 80% 정확도

```
> accuracy(my.iris$Species, pred)
[1] 0.97
> recall(as.integer(my.iris$Species)-2, as.integer(pred)-2)
[1] 0.98
> precision(as.integer(my.iris$Species)-2, as.integer(pred)-2)
[1] 0.9607843
> f1(as.integer(my.iris$Species)-2, as.integer(pred)-2)
[1] 1
```