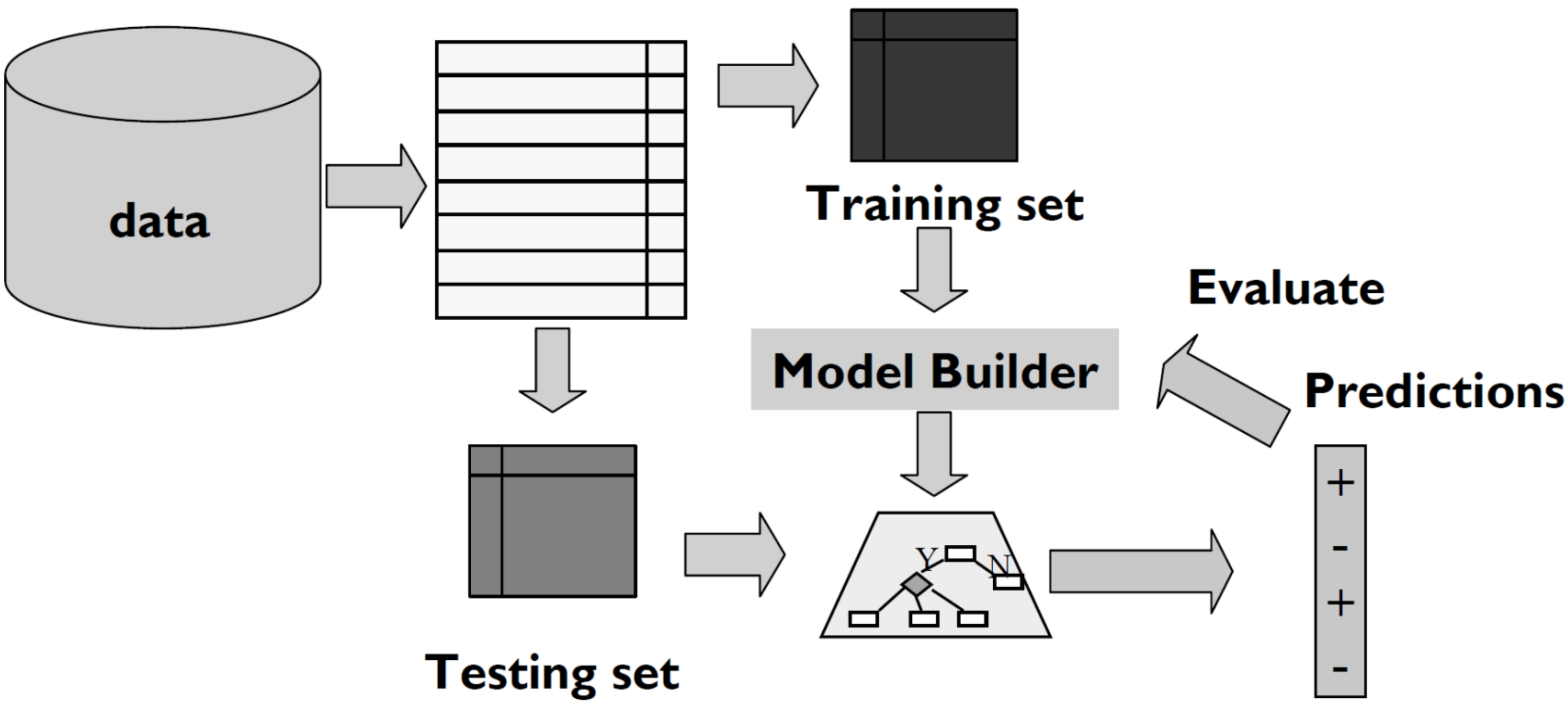
Evaluation

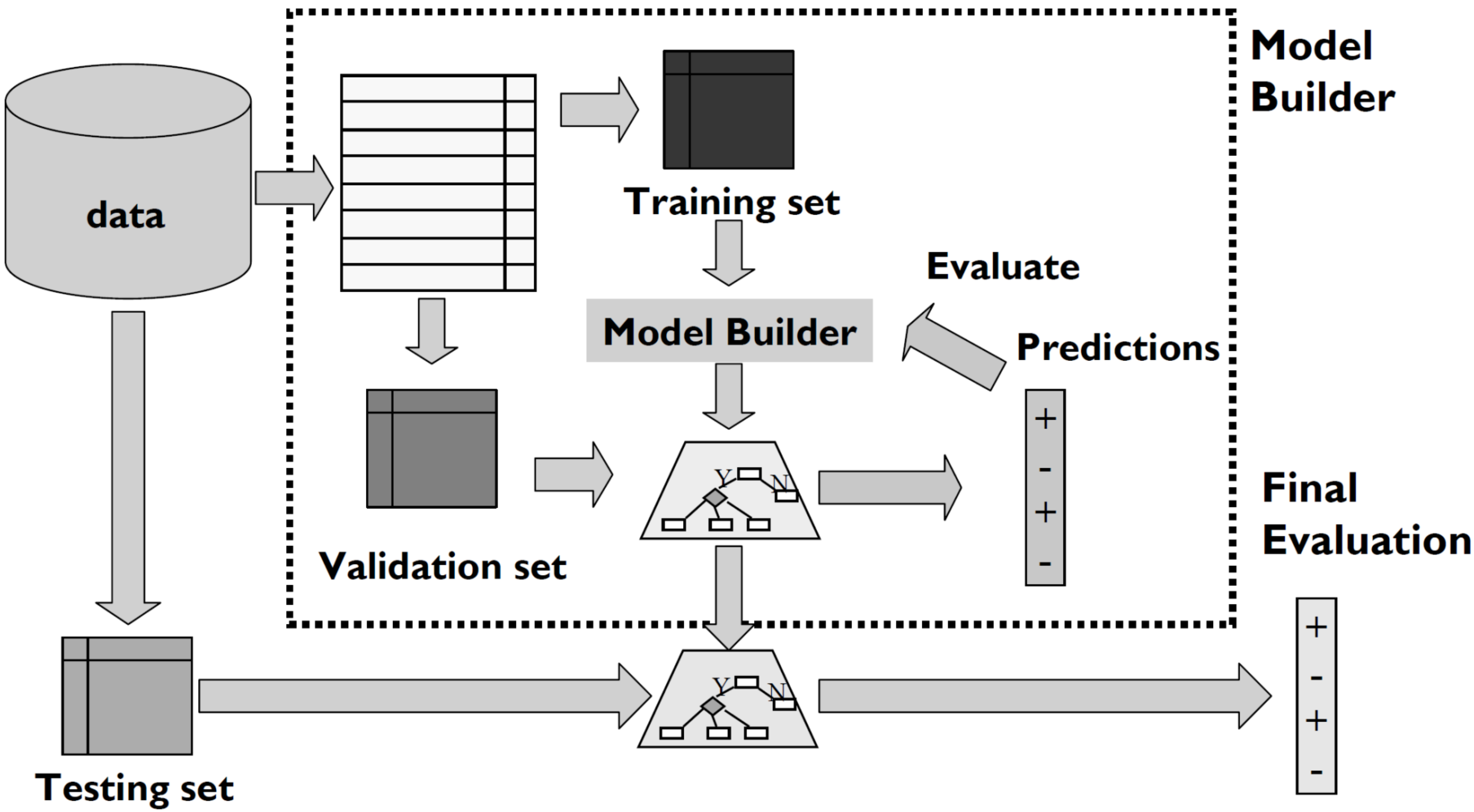
Instructor: Junghye Lee

Department of Industrial Engineering
junghyelee@unist.ac.kr

Evaluation of test set



Generalized Evaluation



Evaluation on “Small” Data

- The holdout method reserves a certain amount for testing and uses the remainder for training
- Usually, one third for testing, the rest for training
- For small or “unbalanced” datasets, samples might not be representative
- For instance, few or none instances of some classes
- Stratified sample
  - Advanced version of balancing the data
  - Make sure that each class is represented with approximately equal proportions in both subsets

Repeated Holdout Method

- Holdout estimate can be made more reliable by repeating the process with different subsamples
  - In each iteration, a certain proportion is randomly selected for training (possibly with stratification)
  - The error rates on the different iterations are averaged to yield an overall error rate
- This is called the repeated holdout method
- Still not optimum: the different test sets overlap

Cross-Validation

- Avoids overlapping test sets
  - First step: data is split into *k* subsets of equal size
  - Second step: each subset in turn is used for testing and the remainder for training
- This is called *k*-fold cross-validation.
- Often the subsets are stratified before the cross-validation is performed.
- The error estimates are averaged to yield an overall error estimate.

Cross-Validation

Iteration 1	Test	Train	Train	Train	Train
Iteration 2	Train	Test	Train	Train	Train
Iteration 3	Train	Train	Test	Train	Train
Iteration 4	Train	Train	Train	Test	Train
Iteration 5	Train	Train	Train	Train	Test

More on Cross-Validation

- Standard method for evaluation
  - Stratified ten-fold cross-validation
- Why ten? Extensive experiments have shown that this is the best choice to get an accurate estimate.
- Stratification reduces the estimate’s variance.
- Even better: repeated stratified cross-validation
  - e.g. ten-fold cross-validation is repeated ten times and results are averaged (reduces the variance).

Leave-One-Out Cross-Validation

- It is a particular form of cross-validation
  - Set number of folds to number of training instances
  - i.e., for *n* training instances, build classifier *n* times
- Makes best use of the data
- Involves no random subsampling
- Very computationally expensive

Leave-One-Out Cross-Validation

This is just one time evaluation.

Can we get the standard deviation of accuracy?

Evaluation criteria

- **Predictive accuracy:** this refers to the ability of the model to correctly predict the target of new or previously unseen data:
- **Time & Memory:** this refers to the computation costs involved in generating and using the model
- **Robustness:** this is the ability of the model to make correct predictions given noisy data or data with missing values
- **Scalability:** this refers to the ability to construct the model efficiently given large amount of data

Evaluation criteria

- **Interpretability:** this refers to the level of understanding and insight that is provided by the model
- **Simplicity:**
  - decision tree size
  - rule compactness
- Domain-dependent quality indicators

Prediction Model

- **Regression**

- Classification

Prediction Output

Compare these two!

AGE	gender	WAIST	BP_HIGH	BP_LWST	BLDS	TOT_CHOLE true	TOT_CHOLE estimated
44	1	86	120	80	75	184.2	183.4
56	0	84	110	70	151	221.1	222.4
38	1	78	103	61	82	170.3	171.6
60	1	88	130	77	153	172.3	170.2
28	1	92	128	77	101	201.5	199.7
42	1	94	134	85	91	194.2	193.1
36	1	83	122	88	95	173.8	175.1
46	1	79	120	80	91	181.1	180.2
56	0	77	99	72	81	178.6	176.8
58	0	80	128	76	126	172.7	170.7
56	0	89	130	80	98	239.0	238.2
46	1	79	118	69	98	148.3	147.1
45	1	93	120	80	80	208.3	207. 2
39	1	88	109	68	82	214.2	211.1

Evaluation of Prediction Model

- **BIAS - The arithmetic mean of the errors**

$$BIAS = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{n} = \frac{\sum_{i=1}^{n} error}{n}$$

  - n is the number of test samples.

- **Mean Absolute Deviation - MAD**

$$MAD = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} = \frac{\sum_{i=1}^{n}|error|}{n}$$

Evaluation of Prediction Model

- **Mean Square Error – MSE (most popular)**

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

  - Standard error is square root of MSE or (RMSE)

- **Mean Absolute Percentage Error – MAPE**

$$MAPE = \frac{\sum_{i=1}^{n}\left(\frac{|y_i - \hat{y}_i|}{y_i}\right) * 100\%}{n}$$

Evaluation of Prediction Model

- **Root relative squared error - RRSE**

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

- In general, the lower the error measure (BIAS, MAD, MSE, MAPE, RRSE) or the higher the R^2, r the better the forecasting model

Which measure?

- Best to look at all of them
- Often it doesn't matter
- Example:

	A	B	C	D
Root mean-squared error	67.8	91.7	63.3	57.4
Mean absolute error	41.3	38.5	33.4	29.2
Root rel squared error	42.2%	57.2%	39.4%	35.8%
Relative absolute error	43.1%	40.1%	34.8%	30.4%
Correlation coefficient	0.88	0.88	0.89	0.91

Prediction Model

- Regression
- **Classification**

Prediction Output

Compare these two!

AGE	gender	WAIST	BP_HIGH	BP_LWST	BLDS	Diabetes true	Diabetes estimated
44	1	86	120	80	75	0	0
56	0	84	110	70	151	0	0
38	1	78	103	61	82	1	0
60	1	88	130	77	153	1	1
28	1	92	128	77	101	0	1
42	1	94	134	85	91	0	0
36	1	83	122	88	95	0	0
46	1	79	120	80	91	1	1
56	0	77	99	72	81	0	0
58	0	80	128	76	126	0	1
56	0	89	130	80	98	0	0
46	1	79	118	69	98	0	0
45	1	93	120	80	80	1	1
39	1	88	109	68	82	0	0

Evaluation of Prediction Model

- Two-class case (yes, no)
- Four different outcomes
  - true positive, true negative, false positive, false negative
- We display these outcomes in the following **confusion matrix**

		Predicted class	
		Yes	No
Actual class	Yes	TP: True positive	FN: False negative
	No	FP: False positive	TN: True negative

Prediction Output

AGE	gender	WAIST	BP_HIGH	BP_LWST	BLDS	Diabetes true	Diabetes estimated
44	1	86	120	80	75	0	0
56	0	84	110	70	151	0	0
38	1	78	103	61	82	**1**	**0**
60	1	88	130	77	153	**1**	**1**
28	1	92	128	77	101	**0**	**1**
42	1	94	134	85	91	**0**	**0**
36	1	83	122	88	95	0	0
46	1	79	120	80	91	1	1
56	0	77	99	72	81	0	0
58	0	80	128	76	126	0	1
56	0	89	130	80	98	0	0
46	1	79	118	69	98	0	0
45	1	93	120	80	80	1	1
39	1	88	109	68	82	0	0

Positive: the event you are interested in, Negative: otherwise

→ false negative, true positive, false positive, true negative

Evaluation of Prediction Model

- **Accuracy**

$$\frac{TP+TN}{TP+FP+TN+FN}$$

- **Sensitivity = Recall**

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

- **Specificity**

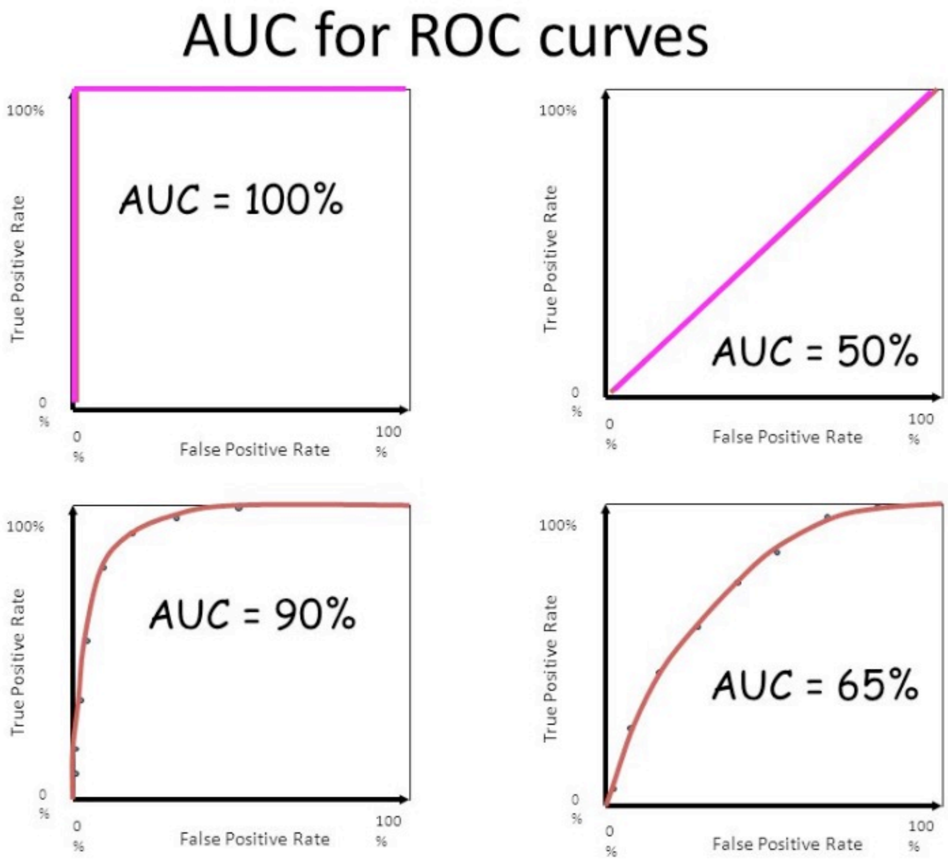
$$\frac{TN}{TN+FP}$$

- **Precision**

$$\frac{TP}{TP+FP}$$

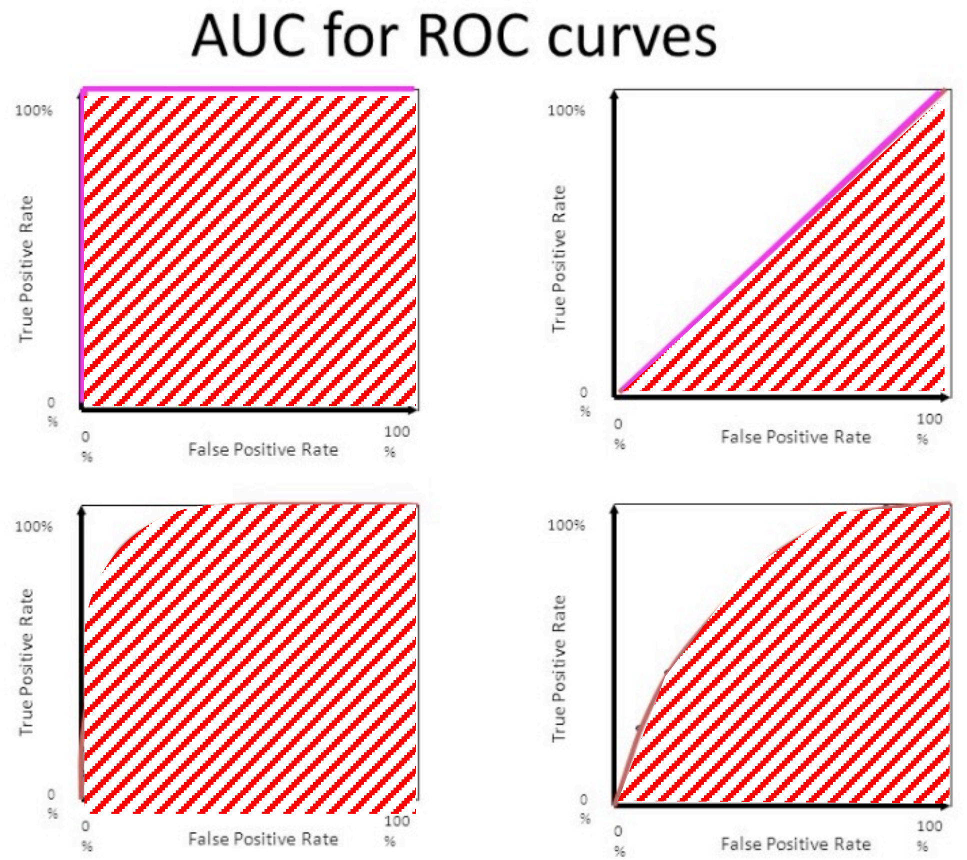
AUC

- Stands for "**Area under Receiver Operating characteristic**"
- It can show tradeoff between **true positives** and **false positives** over the threshold (probability cutoff).
  - (1-specificity) vs. sensitivity



AUC

- Stands for "**Area under Receiver Operating characteristic**"
- It can show tradeoff between true positives and false positives over noisy channel
  - (1-specificity) vs. sensitivity



F-Measure

- It can show tradeoff between precision and recall over noisy channel

$$F_\beta = \frac{1}{\frac{\beta^2}{\beta^2+1} \cdot \frac{1}{Recall} + \frac{1}{\beta^2+1} \cdot \frac{1}{Precision}} = \frac{(\beta^2+1) P \cdot R}{\beta^2 P + R}$$

- The most popular measure is

$$F_1 = \frac{2PR}{P+R}$$ (Harmonic average)

Cross-validation and AUC, F1

- Collect probabilities for instances in test folds
- Sort instances according to probabilities

- Generate an AUC or a F1 for each fold
- Average them

- Generate an AUC or a F1 for each repetition
- Average them

Questions?