# Model Assessment & Selection

Data Mining

Prof. Sujee Lee

Department of Systems Management Engineering

Sungkyunkwan University

# Performance Evaluation (revisited)

- **Evaluation Metrics for Regression**
  - Given a data set $\boldsymbol{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$
  - **Mean Squared Error (RMSE):** $\mathrm{MSE} = \frac{1}{n}\sum_i (y_i - \widehat{y_i})^2$
  - **Root Mean Squared Error (RMSE):** $\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_i (y_i - \widehat{y_i})^2}$
  - **Mean Absolute Error (MAE):** $\mathrm{MAE} = \frac{1}{n}\sum_i |y_i - \widehat{y_i}|$
  - **Coefficient of Determination ($\boldsymbol{R}^2$):** $R^2 = 1 - \frac{\sum_i (y_i - \widehat{y_i})^2}{\sum_i (y_i - \bar{y})^2}$ , $\bar{y} = \frac{1}{n}\sum_i y_i$

# Performance Evaluation (revisited)

## Evaluation Metrics for Classification

- Given a test set $\boldsymbol{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$
- **Accuracy**: the fraction of correctly classified data points

$$\text{Accuracy} = \frac{1}{n}\sum_{i} \mathbb{I}(y_i = \hat{y}_i) \times 100\%$$

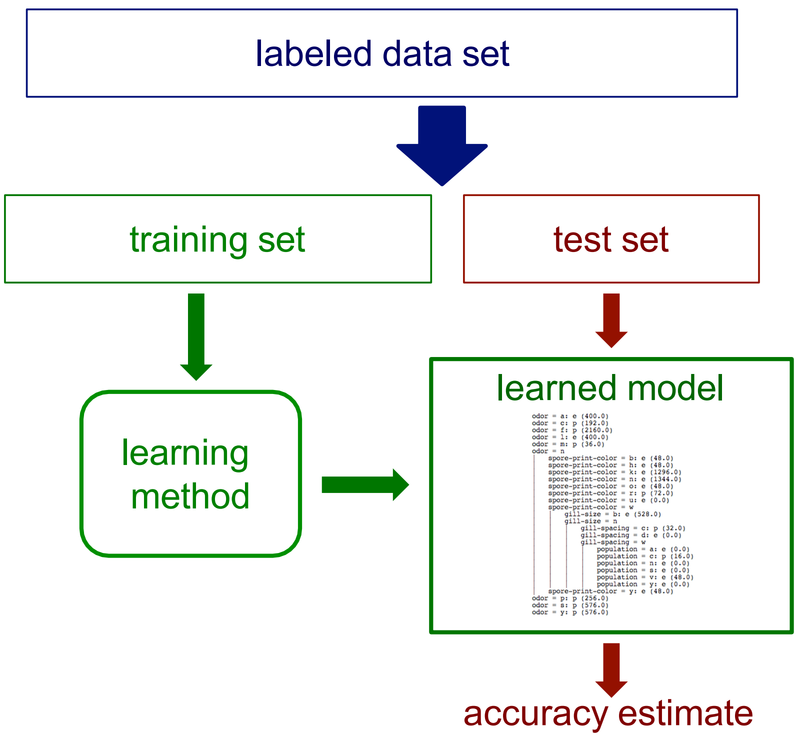
- **Confusion Matrix**
  - **Precision**: The proportion of true positives (TP) among the predicted positives (FP + TP).
    - assess the performance of positive predictions.
  - **Recall**: The proportion of true positives (TP) among the actual positives (FN + TP).
    - evaluate how well the model predicts actual positive cases (also called sensitivity or True Positive Rate, TPR).
  - **F1-score**: A metric that combines precision and recall.

| | | Predicted Class | |
|---|---|---|---|
| | | positive | negative |
| Actual Class | positive | true positives (TP) | false negative (FN) |
| | negative | false positive (FP) | true negatives (TN) |

# Data Splitting

- **Test Sets**
  - How can we get an unbiased estimate of the accuracy of a learned model?
    - The goal is to find a predictive model that not only fits well to our past data, but more importantly, one that predicts a future outcome accurately.
    - In the absence of new data, we can assess the performance of models by dividing our data into two sets

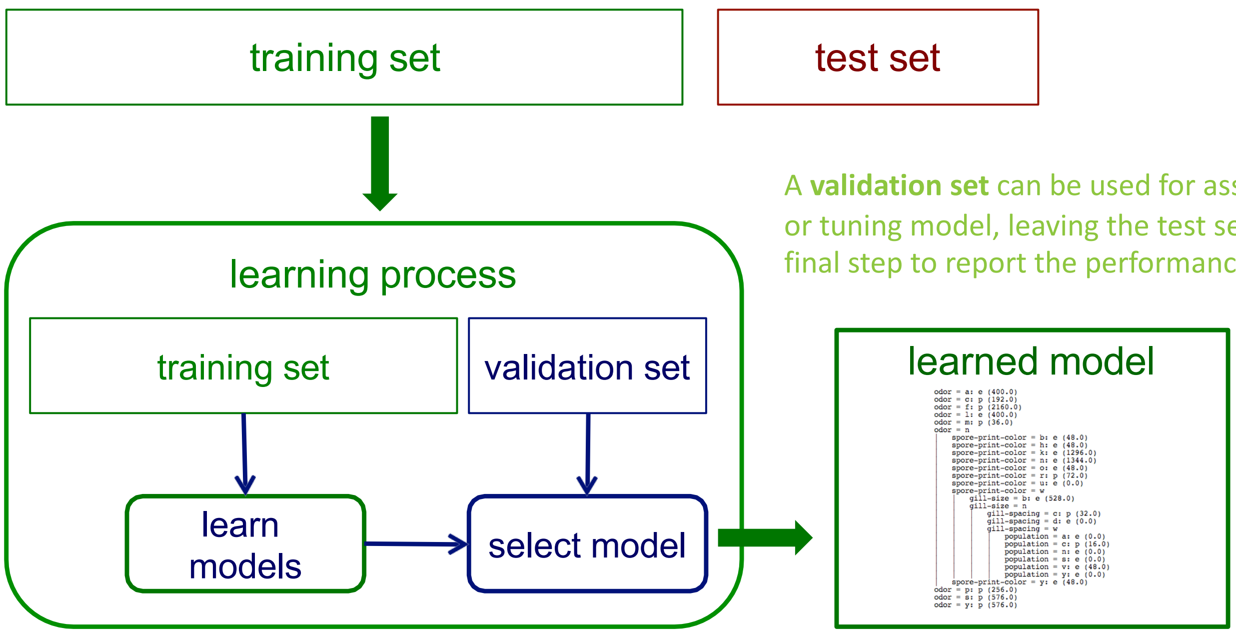
**Training set**: used to develop feature sets, train our models, tune hyperparameters, compare models, etc.




**Test set:** used to estimate an assessment of the final model's performance.

# Data Splitting

- **Validation Sets**
  - Suppose we want unbiased estimates of accuracy during the learning process (e.g. to choose the best models among all)?
    - If a test set is used to assess model performance in the training phase, then the model that best fits the test set is selected, which violates the principle of finding a model that fits unknown future data well.

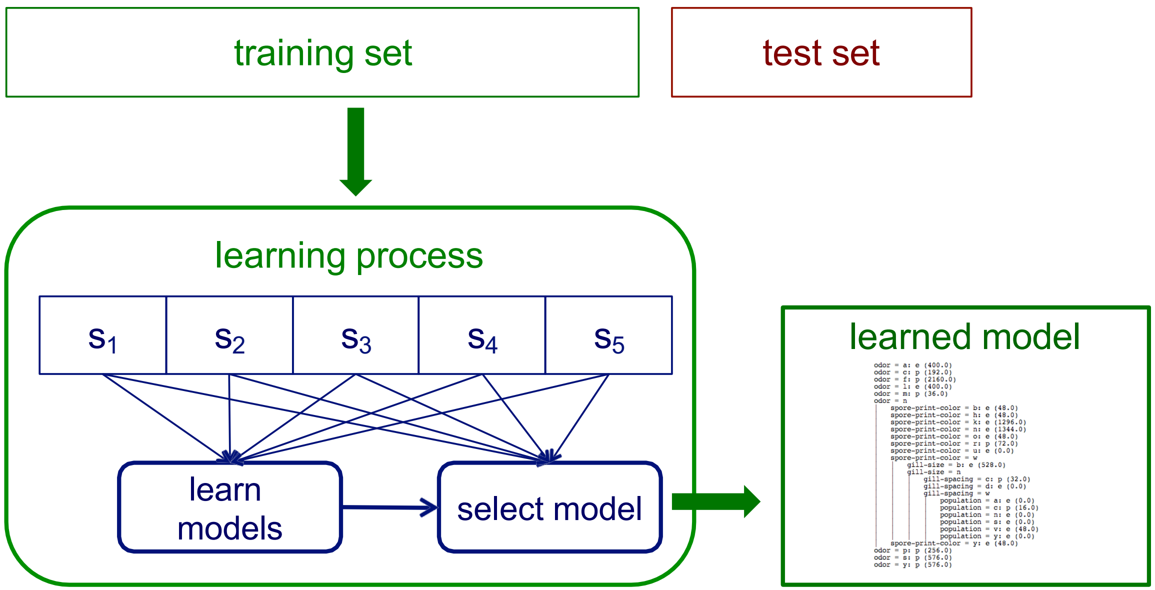



A **validation set** can be used for assessing model performance or tuning model, leaving the test set to be used only once as a final step to report the performance of the final model.

# Data Splitting

- **Limitations of using a single training/validation/test partition**
  - We may not have enough data to make sufficiently large training and test sets
    - a larger test set gives us more reliable estimate of accuracy (i.e. a lower variance estimate)
    - a larger training set will be more representative of how much data we actually have for learning process
  - a single training set doesn't tell us how sensitive accuracy is to a particular training sample

- **Cross Validation**

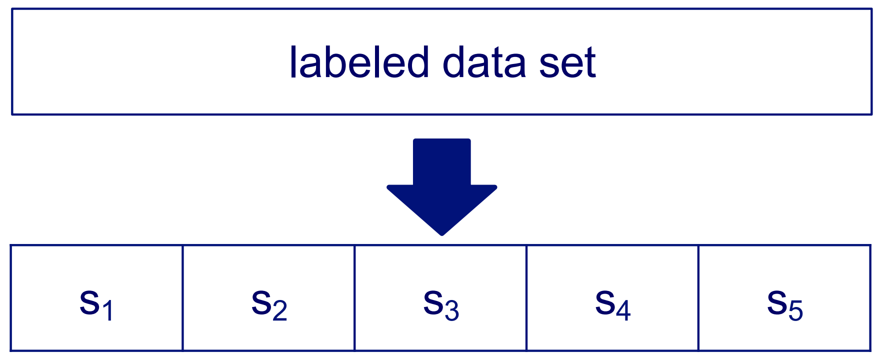


# Cross Validation

## Cross Validation

- k-fold cross-validation (k-fold CV) is randomly divides the training data into k groups (folds) of approximately equal size.
  - In practice, typically use k = 5 or k = 10. When k=n, leave-one-out cross validation (LOOCV).

partition data
into *k* subsamples

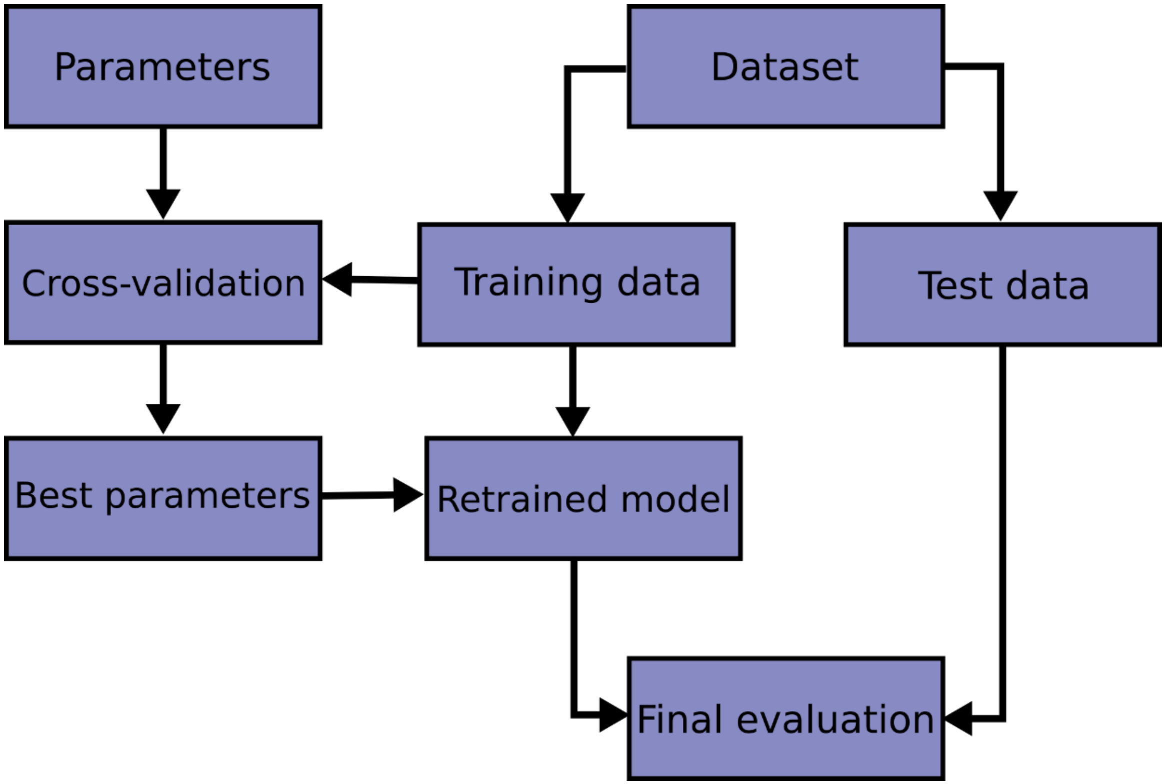



iteratively leave one
subsample out for the test
set, train on the rest

| iteration | train on | test on |
|---|---|---|
| 1 | $s_2$ $s_3$ $s_4$ $s_5$ | $s_1$ |
| 2 | $s_1$ $s_3$ $s_4$ $s_5$ | $s_2$ |
| 3 | $s_1$ $s_2$ $s_4$ $s_5$ | $s_3$ |
| 4 | $s_1$ $s_2$ $s_3$ $s_5$ | $s_4$ |
| 5 | $s_1$ $s_2$ $s_3$ $s_4$ | $s_5$ |

# Cross Validation

- **Cross Validation & Model Selection**
  - After cross-validation, the best-performing set of hyperparameters / algorithm is selected based on validation performance.
  - The model is retrained on the entire training dataset using the best setting found during CV.



# Confusion Matrix

- **Confusion matrices**
  - How can we understand what types of mistakes a learned model makes?

activity recognition from video

actual class

| | bend | jack | jump | pjump | run | side | skip | walk | wave1 | wave2 |
|---|---|---|---|---|---|---|---|---|---|---|
| bend | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| jack | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| jump | 0 | 0 | 89 | 0 | 0 | 0 | 11 | 0 | 0 | 0 |
| pjump | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 |
| run | 0 | 0 | 0 | 0 | 89 | 0 | 11 | 0 | 0 | 0 |
| side | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 |
| skip | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 0 |
| walk | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 |
| wave1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 67 | 33 |
| wave2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |

predicted class

# ROC Curve & AUC

- Metrics like accuracy can fail to provide a clear picture of the model's performance with imbalanced data. Additionally, metrics such as precision, recall, and F1-score are calculated based on a specific threshold, and changing this threshold can dramatically alter the model's evaluation.

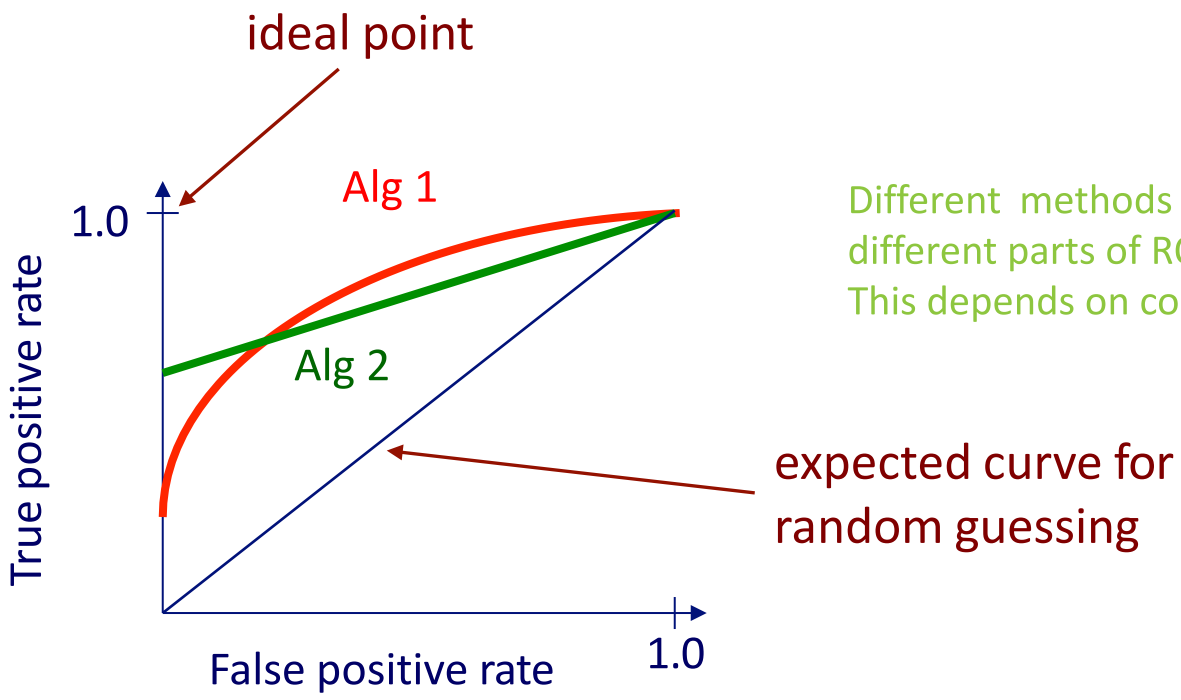
| | | Predicted Class: positive | Predicted Class: negative |
|---|---|---|---|
| Actual Class | positive | true positives (TP) | false negative (FN) |
| | negative | false positive (FP) | true negatives (TN) |

**True Positive Rate (TPR; recall)** $\frac{TP}{\text{actual pos}} = \frac{TP}{TP + FN}$

**False Positive Rate (FPR)** $\frac{FP}{\text{actual neg}} = \frac{FP}{TN + FP}$

# ROC Curve

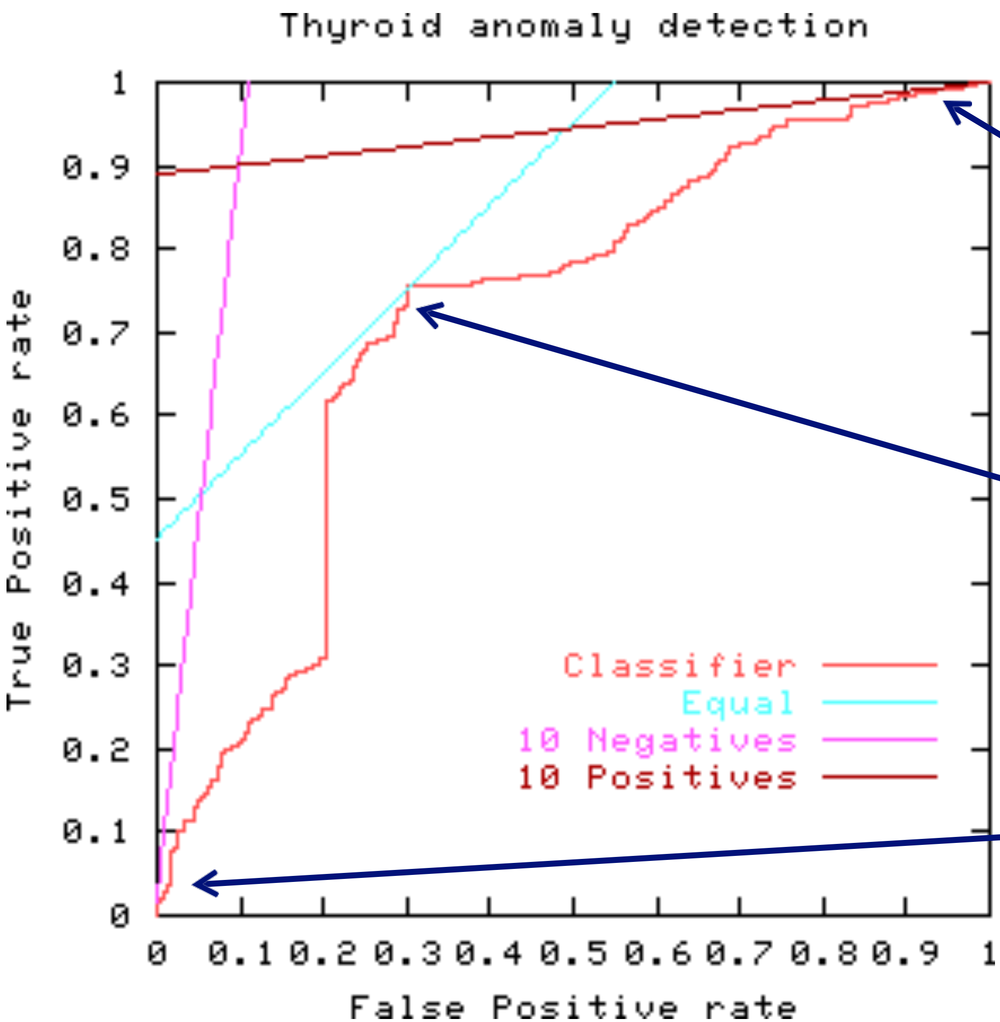
- **ROC Curve**
  - A **Receiver Operating Characteristic (ROC) curve** plots the TPR vs. the FPR as a threshold for the confidence of an instance being positive is varied




Different methods can work better in different parts of ROC space.
This depends on cost of false + vs. false -

# ROC Curve

- **ROC curves and misclassification costs**




best operating point when FN costs 10× FP

best operating point when cost of misclassifying positives and negatives is equal

best operating point when FP costs 10× FN

# ROC Curve

- **Algorithm for creating an ROC curve**

  1. sort test-set predictions according to confidence that each instance is positive
  2. step through sorted list from high to low confidence

     I. locate a threshold between

     instances with opposite classes (keeping instances with the same confidence

     value on the same side of threshold)

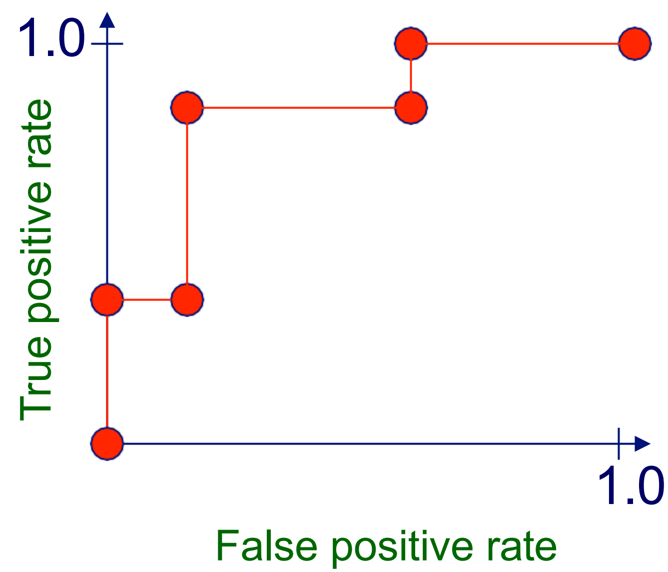
     II. compute TPR, FPR for instances above threshold

     III. output (FPR, TPR) coordinate

# ROC Curve

- **Plotting an ROC curve**

| instance | confidence positive | | correct class |
|---|---|---|---|
| Ex 9 | .99 | | + |
| Ex 7 | .98 | TPR= 2/5, FPR= 0/5 | + |
| Ex 1 | .72 | TPR= 2/5, FPR= 1/5 | - |
| Ex 2 | .70 | | + |
| Ex 6 | .65 | TPR= 4/5, FPR= 1/5 | + |
| Ex 10 | .51 | | - |
| Ex 3 | .39 | TPR= 4/5, FPR= 3/5 | - |
| Ex 5 | .24 | TPR= 5/5, FPR= 3/5 | + |
| Ex 4 | .11 | | - |
| Ex 8 | .01 | TPR= 5/5, FPR= 5/5 | - |



# AUC

- **Area Under Curve (AUC)**
  - **AUC** represents the **area under the ROC curve** and summarizes the performance of a classification model as a single value.
  - The closer the AUC is to 1, the better the model's performance.