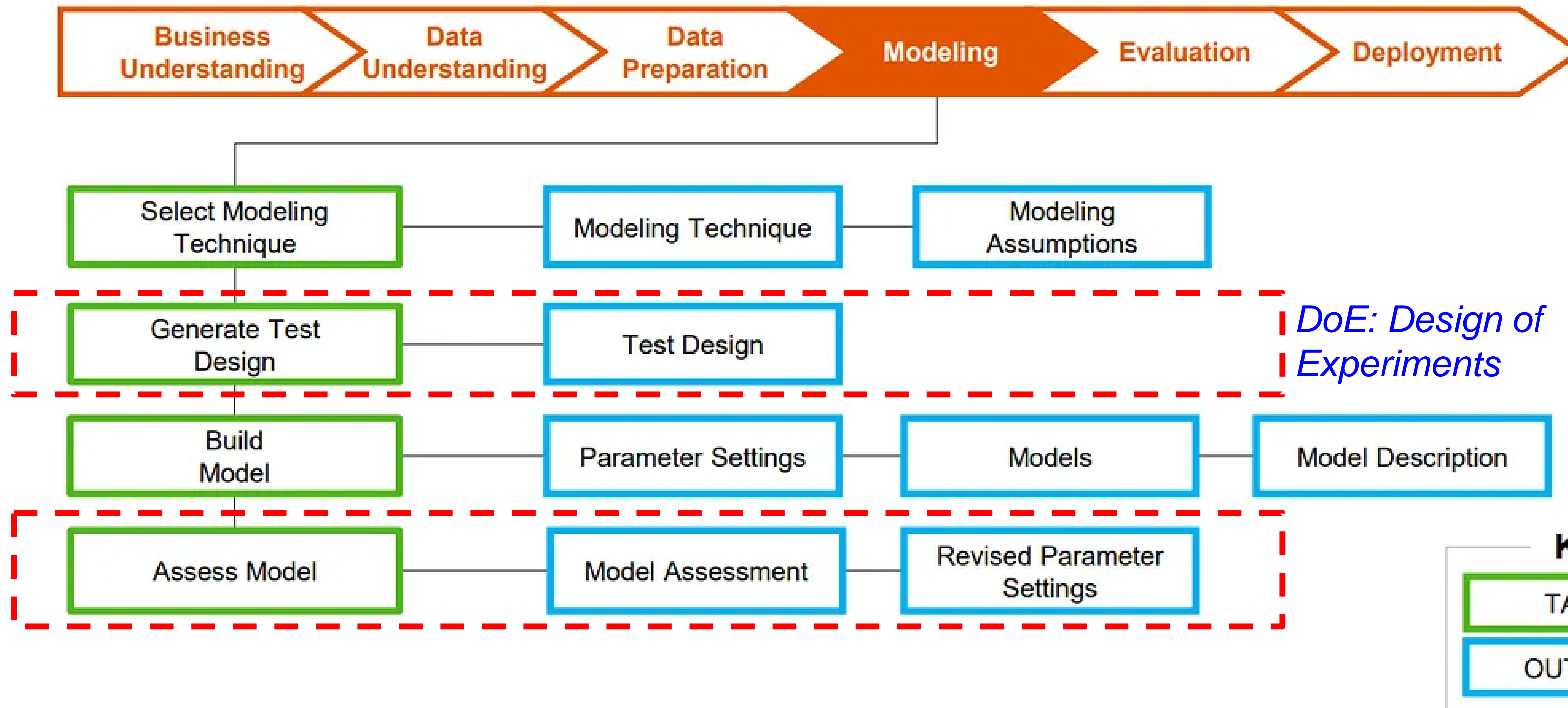


Pengukuran Kinerja Model

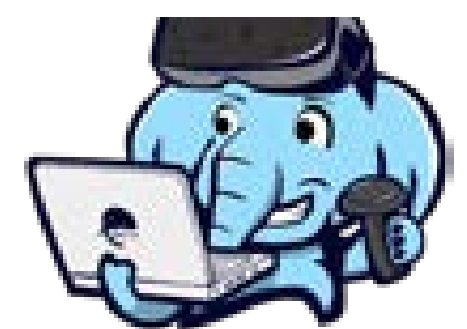
Tim Dosen IF3170 Teknik Informatika ITB

CRISP-DM – Phase 4: Modeling



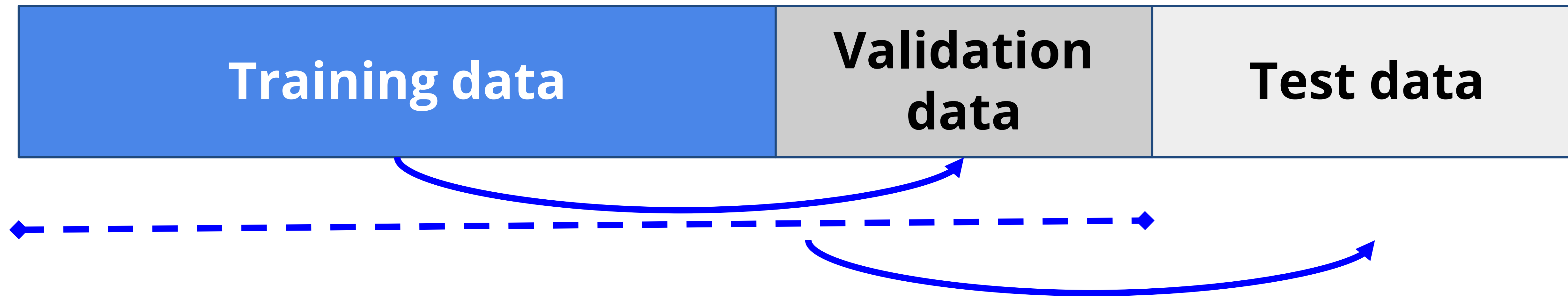
Pemilihan Skema Validasi

- Hold-out validation
- K-fold cross validation
- Bootstrapping Validation



Pemilihan Skema Validasi:

Hold-out Validation *random*



1. Random split dataset menjadi training data, validation data, dan test data.
2. Untuk setiap konfigurasi parameter, lakukan pembelajaran, lalu hitung kinerja terhadap validation data.
3. Menggunakan model dengan konfigurasi terbaik hasil (2) (training dengan training data saja atau training+validation data), hitung kinerjanya terhadap test data.



Contoh Implementasi Python: Hold-out Validation

```
from sklearn.model_selection import train_test_split
from sklearn import datasets
from sklearn import svm

X, y = datasets.load_iris(return_X_y=True)
print(X.shape, y.shape)

#split 60:20:20
X_hold, X_test, y_hold, y_test = train_test_split(X, y, test_size=0.2,
random_state=0)
X_train, X_val, y_train, y_val = train_test_split(X_hold, y_hold,
test_size=0.25, random_state=0)

print('Training data: ',X_train.shape, y_train.shape)
print('Validation data: ',X_val.shape, y_val.shape)
print('Testing data: ',X_test.shape, y_test.shape)

clf = svm.SVC(kernel='linear', C=1).fit(X_train, y_train)
print('Validation score:',clf.score(X_val, y_val))
```

5

(150, 4) (150,)
Training data: (90, 4) (90,)
Validation data: (30, 4) (30,)
Testing data: (30, 4) (30,)
Validation score: 0.933333
Test score: 1.0

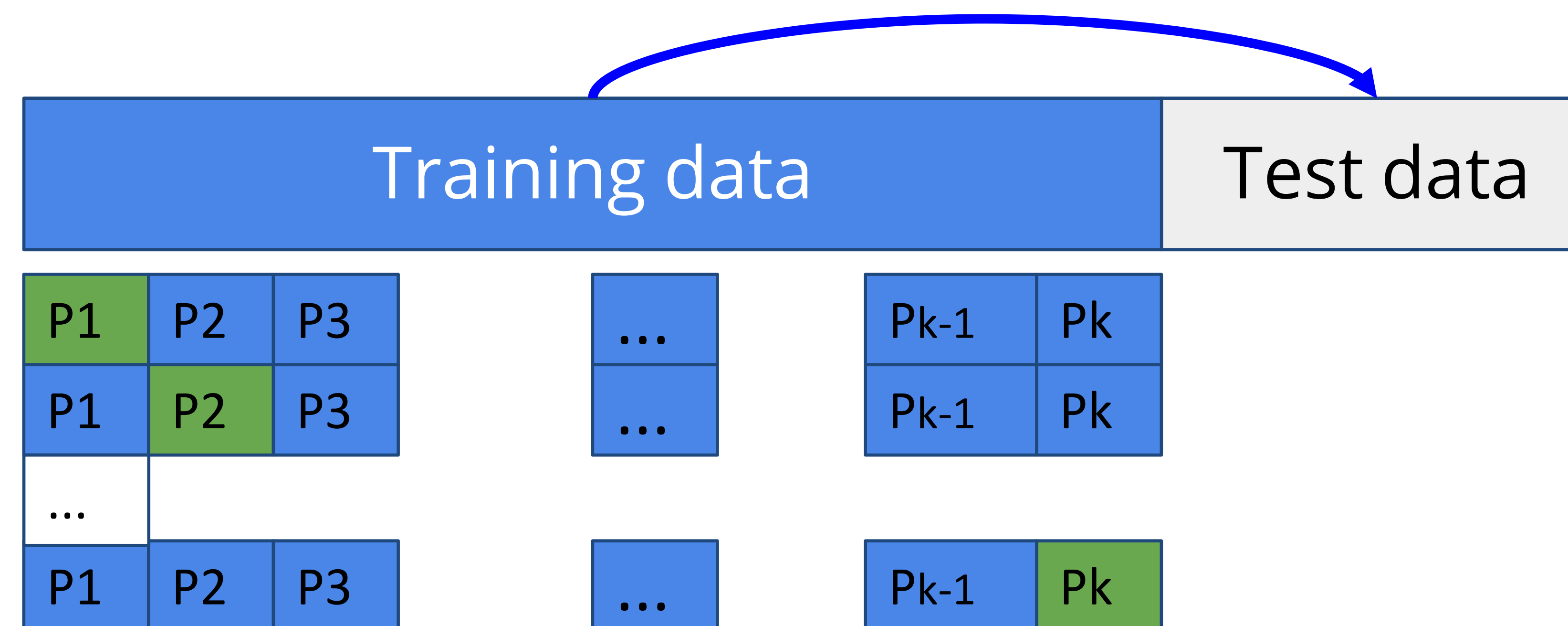
```
# tahap pengujian, setelah eksperimen semua konfigurasi selesai
clf = svm.SVC(kernel='linear', C=1).fit(X_hold, y_hold)
print('Test score:',clf.score(X_test, y_test))
```



Pemilihan Skema Validasi: K-fold Cross Validation

random train sm test
train < train (k-1)
validate (1)

1. Random split dataset menjadi training dan test data.
2. Untuk setiap konfigurasi lakukan k-fold cross validation (1 partisi validation data, k-1 partisi training data).
3. Menggunakan model dengan konfigurasi terbaik hasil (2) (all training data), hitung kinerjanya terhadap test data.



Keterangan:

Validation data



Contoh Implementasi Python: K-Fold Cross Validation

```
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score

X_hold, X_test, y_hold, y_test = train_test_split(X, y,
test_size=0.2, random_state=0)
clf = svm.SVC(kernel='linear', C=1)
scores=cross_val_score(clf, X, y, cv=5)
print('Validation scores:',scores)

clf = svm.SVC(kernel='linear', C=1).fit(X_hold, y_hold)
print('Test score:',clf.score(X_test, y_test))
```

Validation scores: [0.96666667 1. 0.96666667 0.96666667 1.]

Test score: 1.0

```
sklearn.model_selection.cross_val_score(estimator, X, y=None, *, groups=None, scoring=None, cv=None,
n_jobs=None, verbose=0, fit_params=None, params=None, pre_dispatch='2*n_jobs', error_score=nan)
```

https://scikit-learn.org/1.5/modules/cross_validation.html



Contoh Implementasi Python: K-Fold Cross Validation

```
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
```

```
X_hold, X_test, y_hold, y_test = train_test_split(X, y, test_size=0.2,
random_state=0)
kf = KFold(n_splits=5)
for train, val in kf.split(X_hold):
    X_train, X_val, y_train, y_val = X[train], X[val], y[train], y[val]
    print('Train vs valid: ', y_train.shape[0], ' - ',
y_val.shape[0],'; Indeks data-1 val: ',val[0])
```

```
clf = svm.SVC(kernel='linear', C=1)
scores=cross_val_score(clf, X, y, cv=kf)
print('Validation scores:',scores)
```

```
Train vs valid: 96 - 24 ; Indeks data-1 val: 0
Train vs valid: 96 - 24 ; Indeks data-1 val: 24
Train vs valid: 96 - 24 ; Indeks data-1 val: 48
Train vs valid: 96 - 24 ; Indeks data-1 val: 72
Train vs valid: 96 - 24 ; Indeks data-1 val: 96
Validation scores: [1.  1.  0.86666667 1.  0.8667]
Test score: 1.0
```

```
# tahap pengujian, setelah eksperimen semua konfigurasi selesai
clf = svm.SVC(kernel='linear', C=1).fit(X_hold, y_hold)
print('Test score:',clf.score(X_test, y_test))
```



Pemilihan Skema Validasi: Bootstrapping Validation

bikin bootstrap sampling
dr dataset awal

- Sampel bootstrap mungkin tumpang tindih lebih banyak dibandingkan dengan sampel cross-validation
- Cara terbaik untuk melakukan resampling untuk dataset yang sangat kecil

estimasi nya lbh bergantung



di dlm sini
ada validation set yg



Contoh Implementasi Python: Bootstrapping Validation

```
num_splits = 5

X = np.arange(10)
n_samples = len(X)
test_size = int(n_samples * 0.2)

for i in range(num_splits):
    # Bootstrap sampling
    bootstrap_idx = resample(np.arange(n_samples), replace=True, n_samples=n_samples)
    test_idx = np.setdiff1d(np.arange(n_samples), bootstrap_idx)[:test_size]
    print("%s %s" % (bootstrap_idx, test_idx))
```

[1 0 8 3 4 8 0 8 0 3] [2 5]

[7 2 3 3 7 8 4 2 8 1] [0 5]

[2 4 1 3 6 4 8 8 5 9] [0 7]

[9 2 2 9 8 6 7 8 0 3] [1 4]

[8 0 6 0 7 8 5 2 3 8] [1 4]



Metrik Evaluasi

Metrik untuk Klasifikasi

- Akurasi (Accuracy)
- Precision
- Recall (Sensitivity atau True Positive Rate)
- F1-Score
- ROC-AUC (Receiver Operating Characteristic - Area Under Curve)

Metrik untuk Clustering

- Silhouette Score
- Davies-Bouldin Index
- Inertia (Sum of Squared Distances)

Metrik untuk Regresi

- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- R-squared (R^2)

Metrik untuk Pemodelan Waktu (Time Series Forecasting)

- Mean Absolute Percentage Error (MAPE)
- Mean Squared Logarithmic Error (MSLE)



Recall: Prediction Measurements - Klasifikasi

		Prediction	
		True	False
Reality	True	Tp True-positive	Fn False-negative
	False	Fp False-positive	Tn True-negative

$$Accuracy = \frac{Tp + Tn}{Tp + Fp + Tn + Fn}$$

$$Precision = \frac{Tp}{Tp + Fp} \quad Recall = \frac{Tp}{Tp + Fn}$$

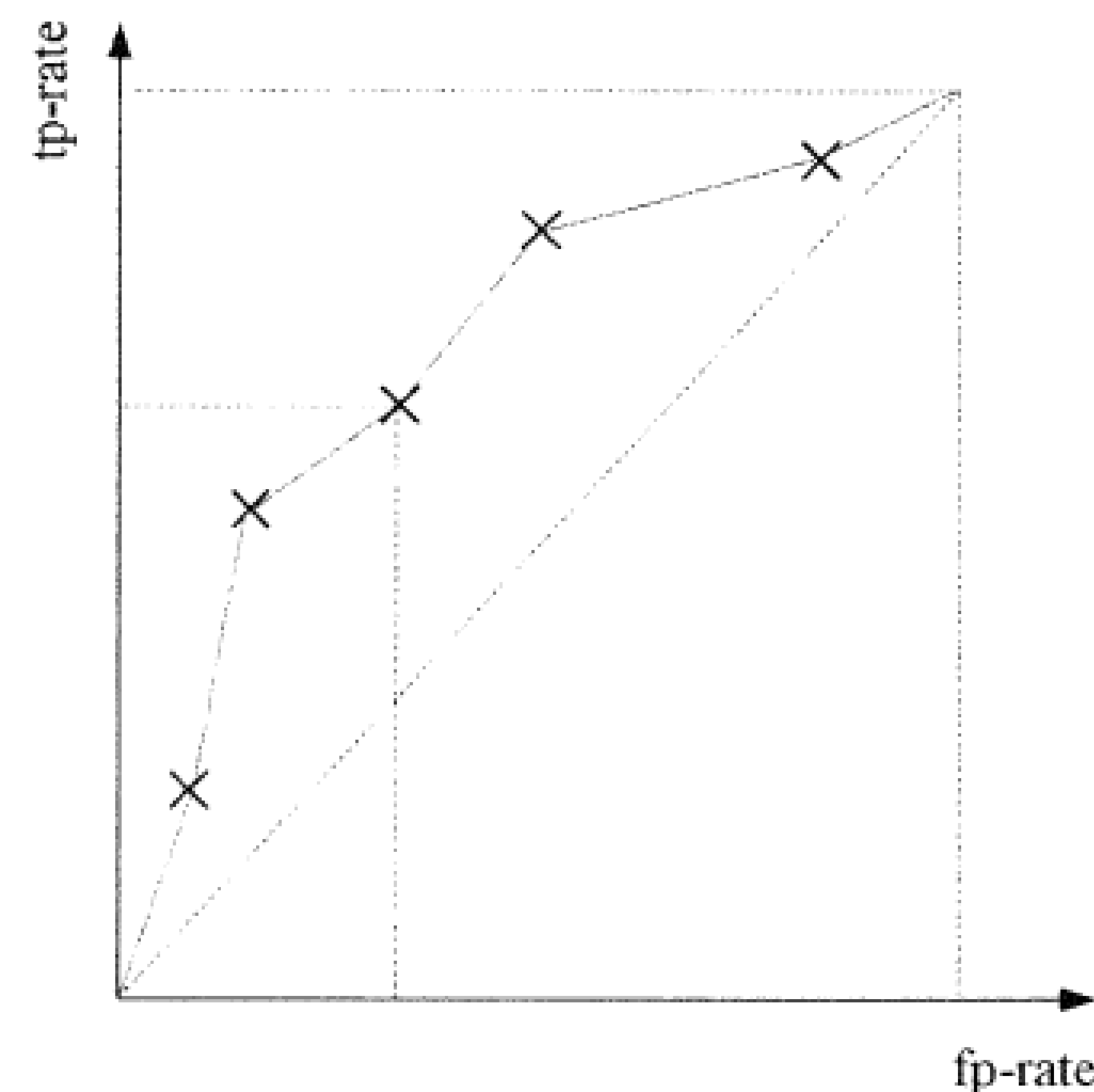
$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

- **Akurasi:**
Persentase prediksi yang benar dari total prediksi yang dilakukan oleh model.
- **Precision:**
Persentase prediksi positif yang benar dari semua prediksi positif yang dibuat oleh model.
- **Recall:**
Persentase kasus positif yang benar-benar diidentifikasi sebagai positif oleh model.
- **F1-Score:**
Harmonic mean dari precision dan recall. Memberikan keseimbangan antara precision dan recall.

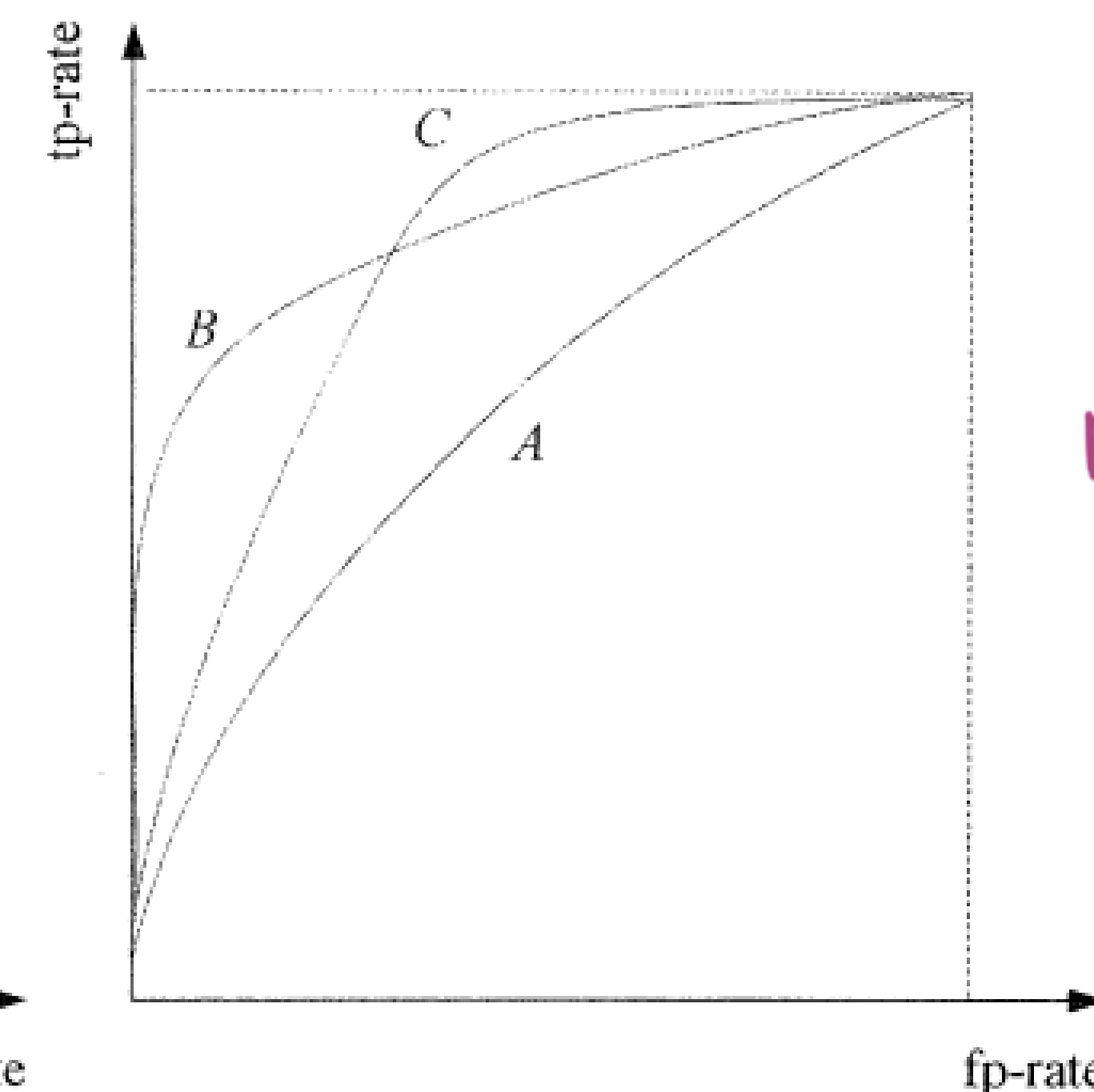


Metrik Evaluasi: ROC (Receiver operating characteristic) - AUC

- AUC mengukur area di bawah kurva ROC, yang merupakan grafik dari true positive rate (Recall) versus false positive rate.
- **Nilai AUC:**
 - 1.0: Kinerja sempurna.
 - 0.5: Kinerja acak (setara dengan tebakan acak).
- **Kelebihan** : Berguna untuk mengevaluasi model pada berbagai threshold klasifikasi.
- **Kekurangan** : Lebih sulit diinterpretasikan daripada metrik sederhana seperti akurasi.



(a) Example ROC curve



(b) Different ROC curves for different classifiers

		Predict	
		T	F
Real	T	TP	FN
	F	FP	TN

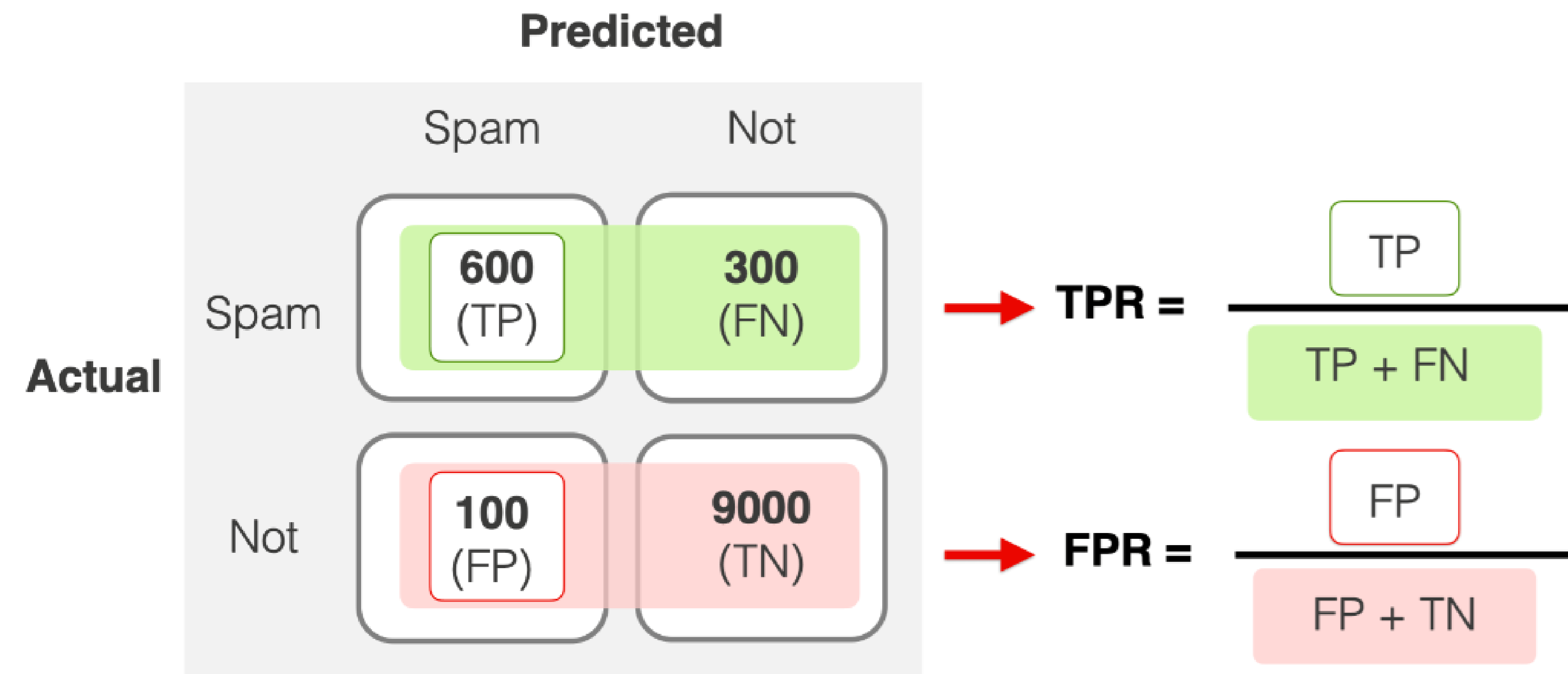
$$\text{ACC} = \frac{TP + TN}{TP + TN + FP + FN}$$
$$\text{PREC} = \frac{TP}{TP + FP}$$
$$\text{REC} = \frac{TP}{TP + FN}$$
$$F1 = 2 \times \frac{\text{prec} \times \text{rec}}{\text{prec} + \text{rec}}$$



Metrik Evaluasi:

ROC (Receiver operating characteristic) - AUC








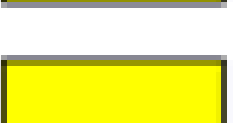

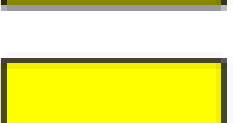
- The ROC curve plots the **True Positive rate (TPR)** against the **False Positive rate (FPR)** at various classification thresholds.



Metrik Evaluasi: ROC (Receiver operating characteristic) - AUC

- The ROC curve plots the True Positive rate (TPR) against the False Positive rate (FPR) at various classification thresholds.

For a probabilistic classification model, you can vary the decision threshold that defines how to convert the model predictions into labels.

	data	label	predicted	confidence
TP		1	1	[0.20,0.80]
TN		0	0	[0.40,0.60]
FN		1	0	[0.50,0.50]
TN		0	0	[0.50,0.50]
TN		0	0	[0.60,0.40]
TN		0	0	[0.60,0.40]
TN		0	0	[0.70,0.30]
TN		0	0	[0.75,0.25]
TN		0	0	[0.80,0.10]
TN		0	0	[1.00,0.00]

put most confident positive predictions at top

put most confident negative predictions at bottom

calculate precision/recall at each break point/threshold

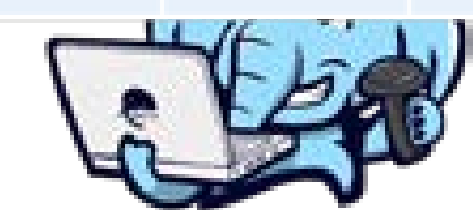
classify everything above threshold as positive and everything else negative

Akurasi = 9/10
Precision₀ = 8/9
Recall₀ = 8/8
Precision₁ = 1/1
Recall₁ = 1/2

Label\Pred	0	1
0	8	0
1	1	1

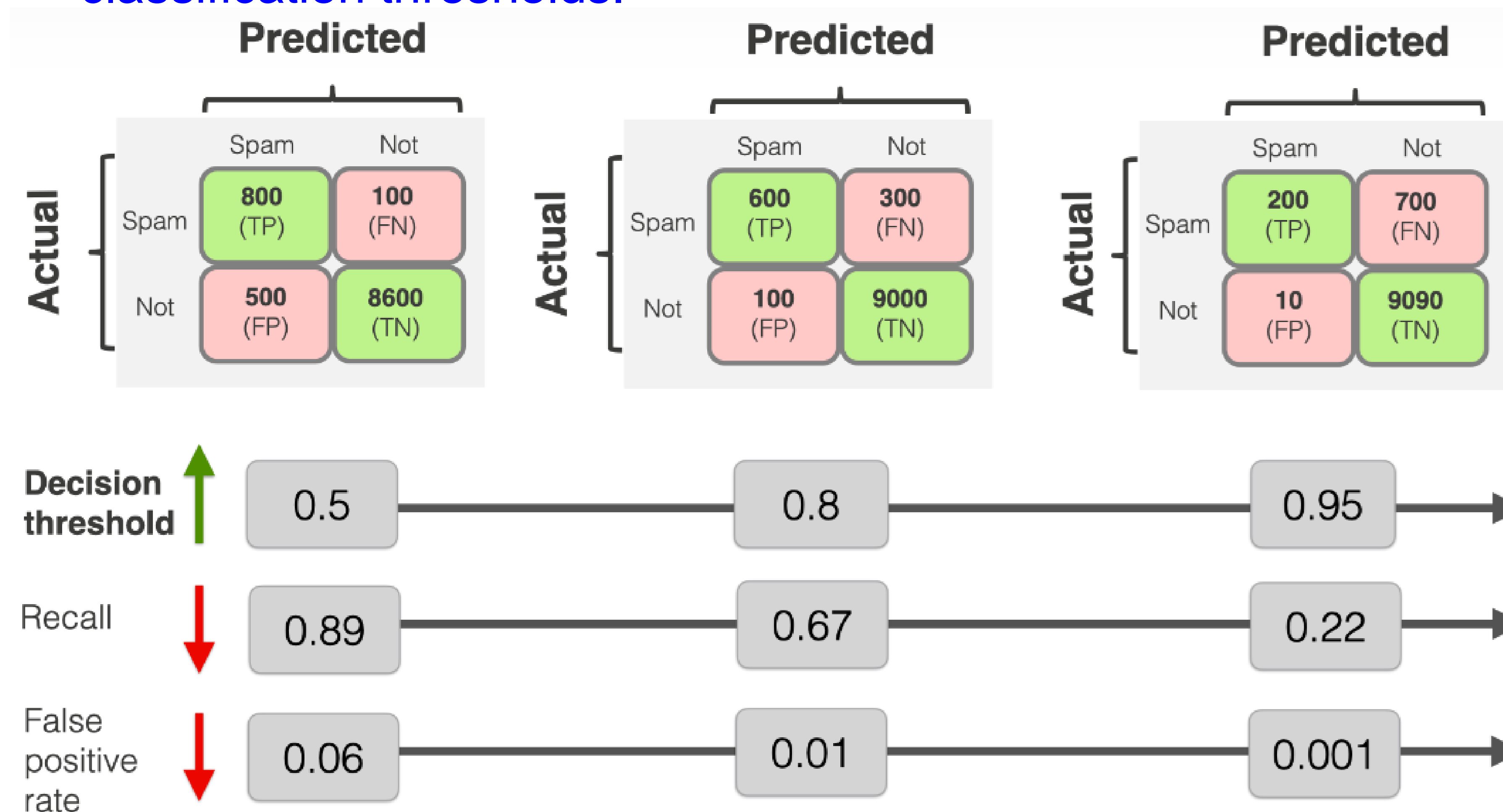
<https://www.evidentlyai.com/classification-metrics/explain-roc-curve>

$$\text{acc} = \frac{9}{10} \quad \text{prec} = \frac{1}{1} \quad \text{rec} = \frac{1}{2}$$
$$F1 = 2 \cdot \frac{\frac{1}{1} \cdot \frac{1}{2}}{\frac{1}{1} + \frac{1}{2}} = \frac{2}{3}$$



Metrik Evaluasi: ROC (Receiver operating characteristic) -AUC

- The ROC curve plots the True Positive rate (TPR) against the False Positive rate (FPR) at **various classification thresholds**.



0.5 threshold:
TPR: $800/(800+100)=0.89$
FPR: $500/(500+8600)=0.06$

0.8 threshold:
TPR: $600/(600+300)=0.67$
FPR: $100/(100+9000)=0.01$

0.95 threshold:
TPR: $200/(200+700)=0.22$
FPR: $10/(10+9090)=0.001$

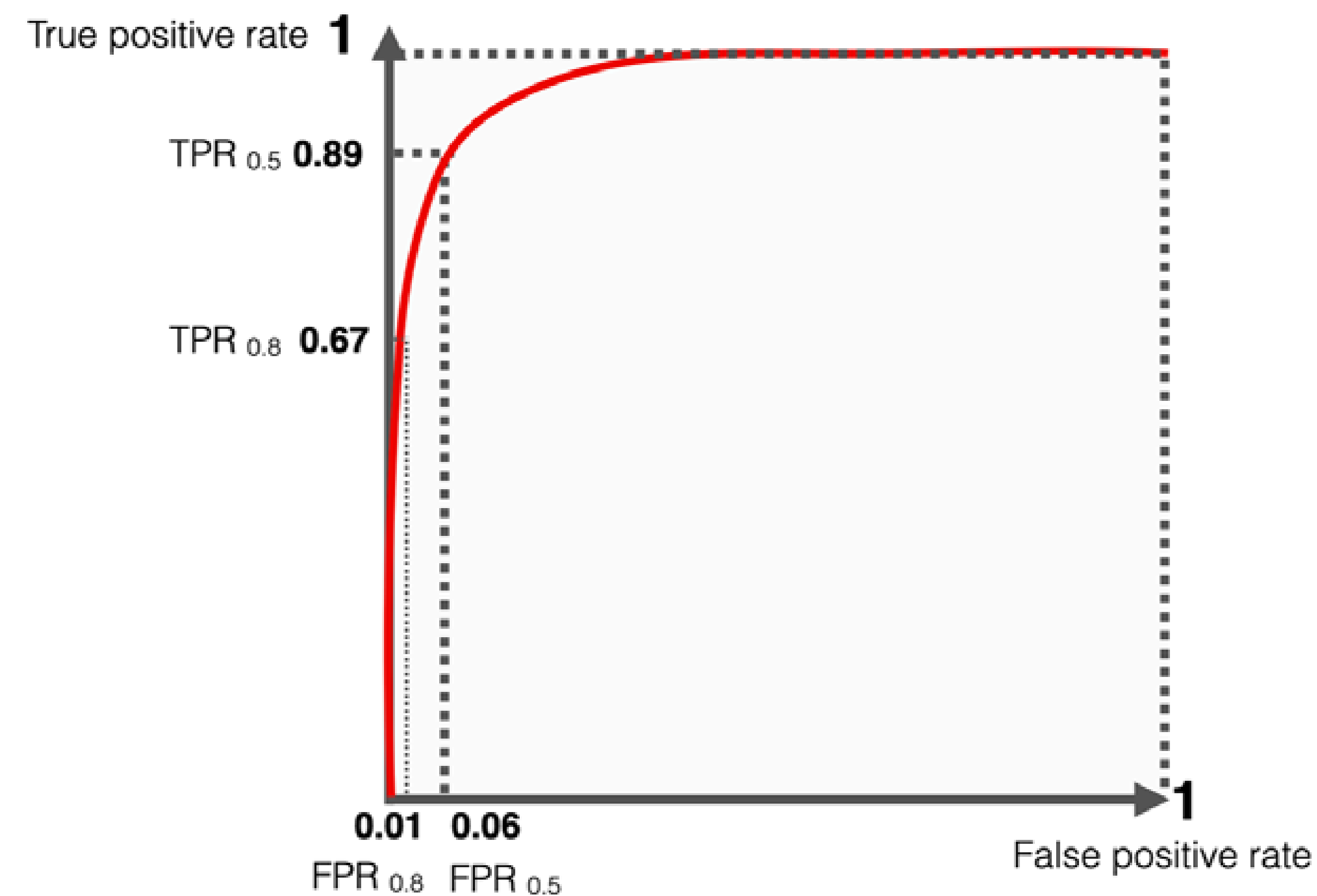
- The **left side** of the curve corresponds to the more "confident" thresholds: a higher threshold leads to lower recall and fewer false positive errors.
- The **right side** of the curve represents the "less strict" scenarios when the threshold is low.



Metrik Evaluasi:

ROC (Receiver operating characteristic) - AUC

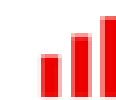
- The ROC curve plots the True Positive rate (TPR) against the False Positive rate (FPR) at **various classification thresholds**.



0.5 threshold:
TPR: 0.89
FPR: 0.06

0.8 threshold:
TPR: 0.67
FPR: 0.01

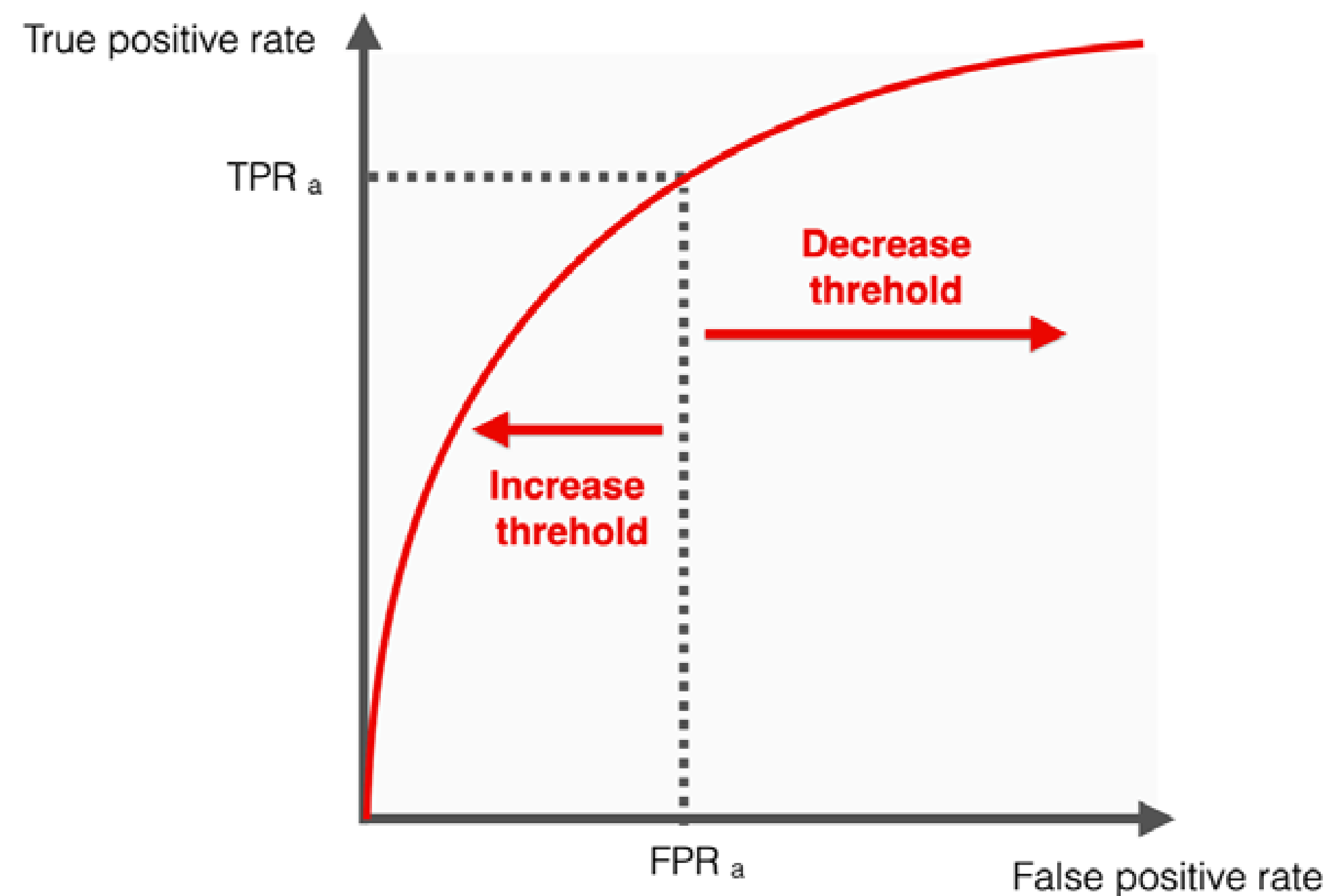
0.95 threshold:
TPR: 0.22
FPR: 0.001



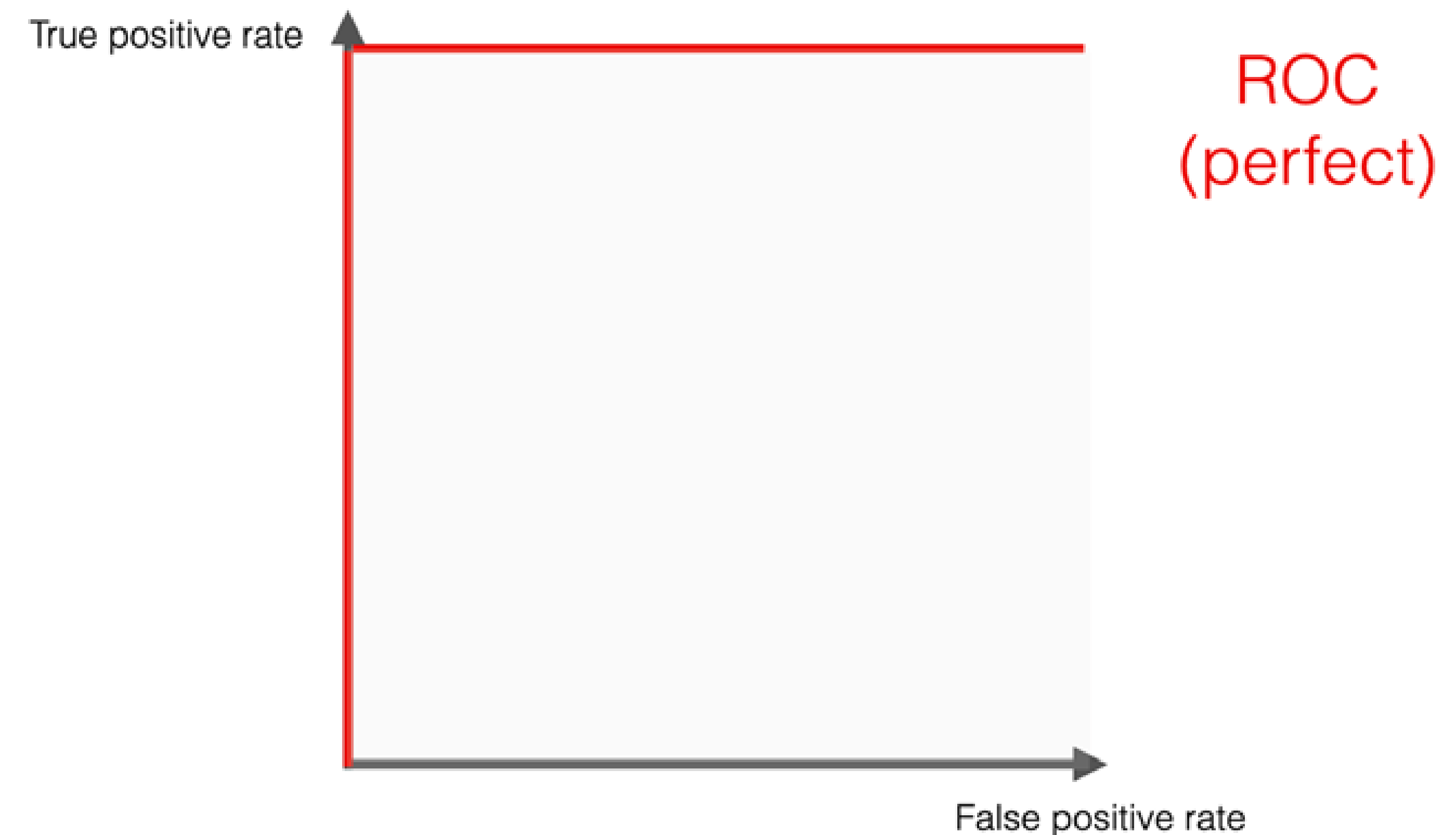
Metrik Evaluasi:

ROC (Receiver operating characteristic) - AUC

- The ROC curve plots the True Positive rate (TPR) against the False Positive rate (FPR) at **various classification thresholds**.



If our model is correct in all the predictions, all the time, it means that the TPR is always 1.0, and FPR is 0.



Recall: Prediction Measurements - Regresi

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

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

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

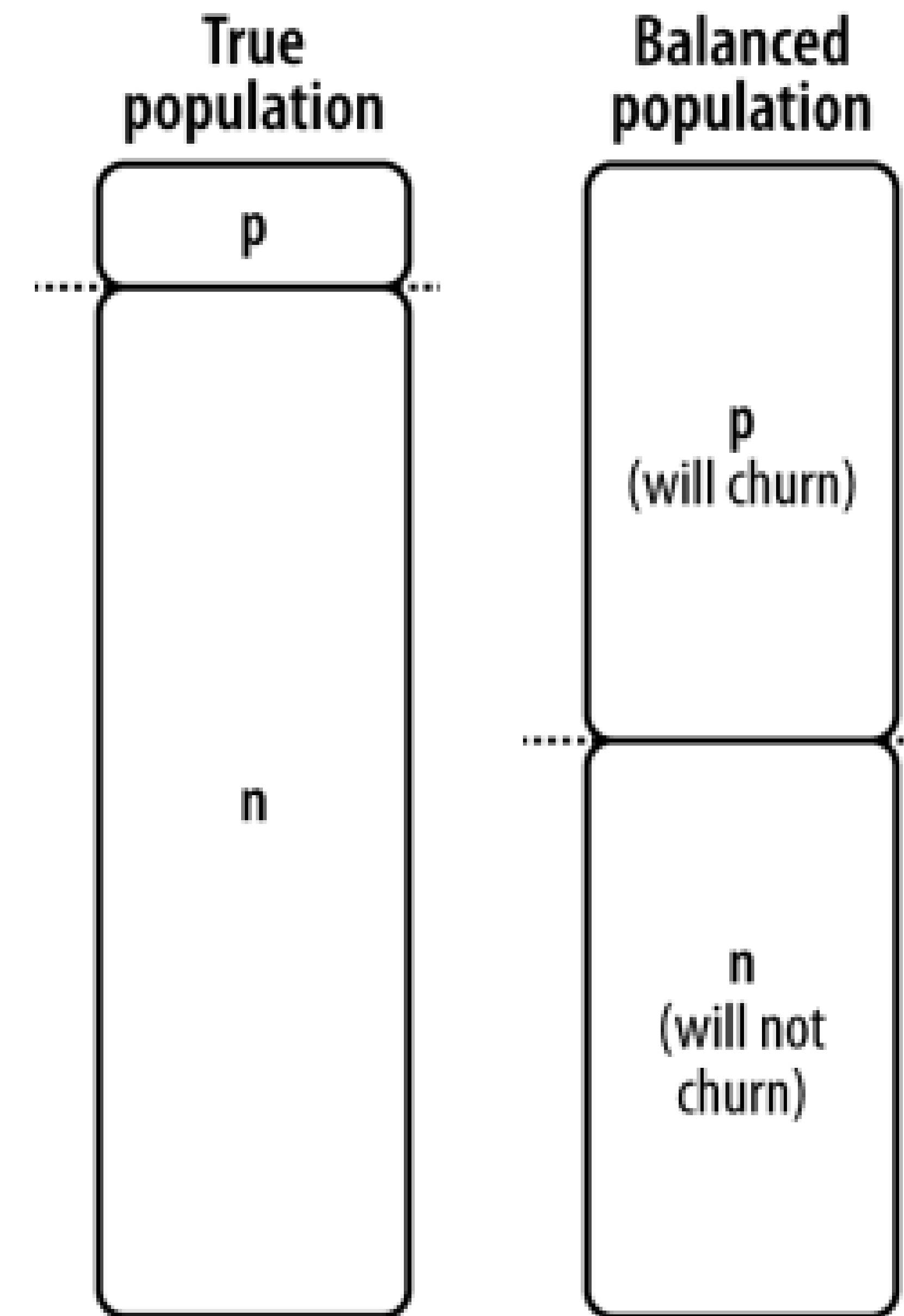
$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

Data	Target	Prediksi	Abs Err	Abs % Err	Sq Err
R1	1	0.99	0.01	1.00	0.00
R2	1	0.48	0.52	52.00	0.27
R3	0.5	0.51	0.01	2.00	0.00
R4	1	0.9	0.1	10.00	0.01
R5	1	0.4	0.6	60.00	0.36
R6	0.5	0.6	0.1	20.00	0.01
			0.22	24.17	0.11
			MAE	MAPE	MSE



Example: Churn Prediction

- Let's consider a customer who churns to be a positive example, so within our population of customers we expect a positive to negative class ratio of 1:9.
- We have two datasets:
 - Representative dataset. If the training population is 1000 customers, a representative sample from the population have 100 positive examples and 900 negative examples.
 - Artificially balanced dataset. If the training population is 1000 customers, a balanced sample from the population have 500 positive examples and 500 negative examples.



Comparing Models: Description



Classifier A



Classifier B

- Both models correctly classify 80% of the balanced population (80% accuracy).
- Classifier A often falsely predicts that customers will churn when they will not, while classifier B makes many opposite errors of predicting that customers will not churn when in fact they will.
- Training with balanced dataset, and testing with representative dataset.

Comparing Models: Balanced Dataset

- Classifier A achieves 80% accuracy on the balanced sample by correctly identifying **all positive** examples but only **30% of the negative** examples.
- Classifier B also achieves 80% accuracy on the balanced sample by correctly identifying **all the negative** examples but only **30% of the positive** examples.

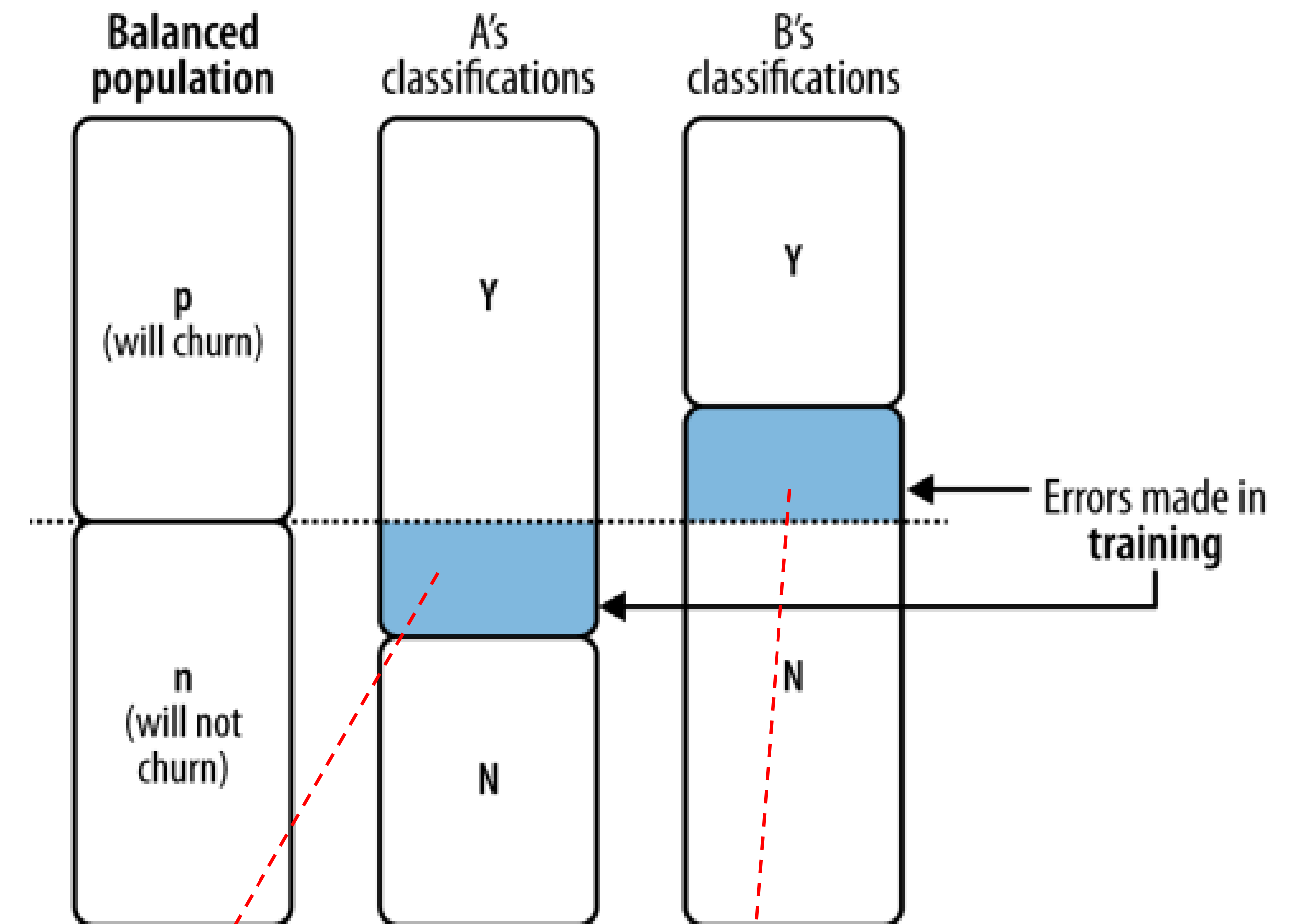


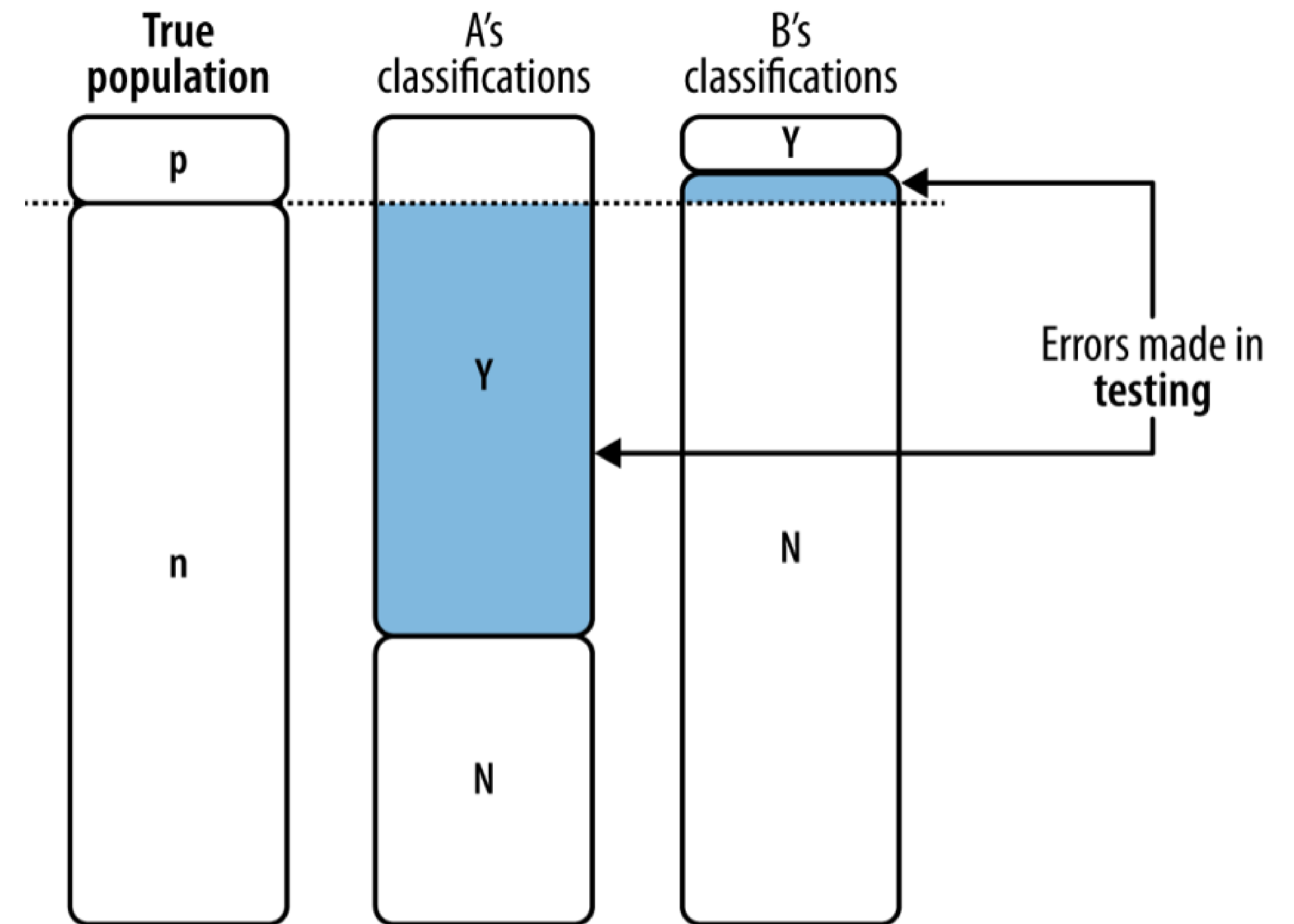
Table 7-2. Confusion matrix of A Table 7-3. Confusion matrix of B

	churn	not churn
Y	500	200
N	0	300

	churn	not churn
Y	300	0
N	200	500

Comparing Models: Representative Dataset

- Both models correctly classify 80% of the balanced population (80% accuracy).
- When applied to the original, unbalanced population of customers, model A's accuracy declines to 37% while model B's rises to 93%.
- Which model is better?
 - model (B) now appears to be better than A because B seems to have greater performance on the true population



Evaluasi & Analisis Hasil

Setelah eksperimen selesai dikerjakan, lakukan:

- **Tinjau hasil evaluasi model berdasarkan metrik kinerja yang dipilih.**
 - Apakah model mencapai target kinerja yang diinginkan?
 - Jika tidak, mengapa?
 - Apakah model cukup baik untuk digunakan dalam produksi?
- **Jika menggunakan beberapa model atau konfigurasi dalam eksperimen, bandingkan kinerja model-model tersebut.**
 - Model mana yang terbaik berdasarkan kinerja keseluruhan?
 - Tinjau pentingnya fitur yang digunakan oleh model untuk membuat prediksi. Fitur mana yang paling berpengaruh pada hasil model?
 - Apakah ada trade-off yang harus dipertimbangkan, seperti model dengan kinerja terbaik tetapi dengan waktu komputasi yang lebih lama?



Contoh Penggunaan Statistical Test

Statistical Test bisa digunakan untuk mengecek apakah hasil dua model klasifikasi kategorikal itu berbeda secara signifikan atau tidak

1. Tulis null hypothesis H_0 yang berisi perbedaan tidak signifikan,
2. Pilih level of significance $\alpha \in (0, 1)$,
3. Tetapkan uji statistik yang sesuai. Misalnya menggunakan McNemar Test
4. Hitung nilai statistik dan bandingkan dengan nilai dengan critical value pada tabel (misalnya untuk McNemar, menggunakan tabel distribusi chi-square dengan derajat kebebasan = 1). Jika nilai statistik lebih besar dari critical value untuk level of significance α yang dipilih, maka null hypothesis ditolak dan perbedaan antara dua model dianggap signifikan

Contoh Penggunaan Uji Statistik McNemar

1. Buat confusion matrix yang membandingkan hasil dari kedua model
2. Hitung nilai statistik
3. Bandingkan dengan tabel chi-square

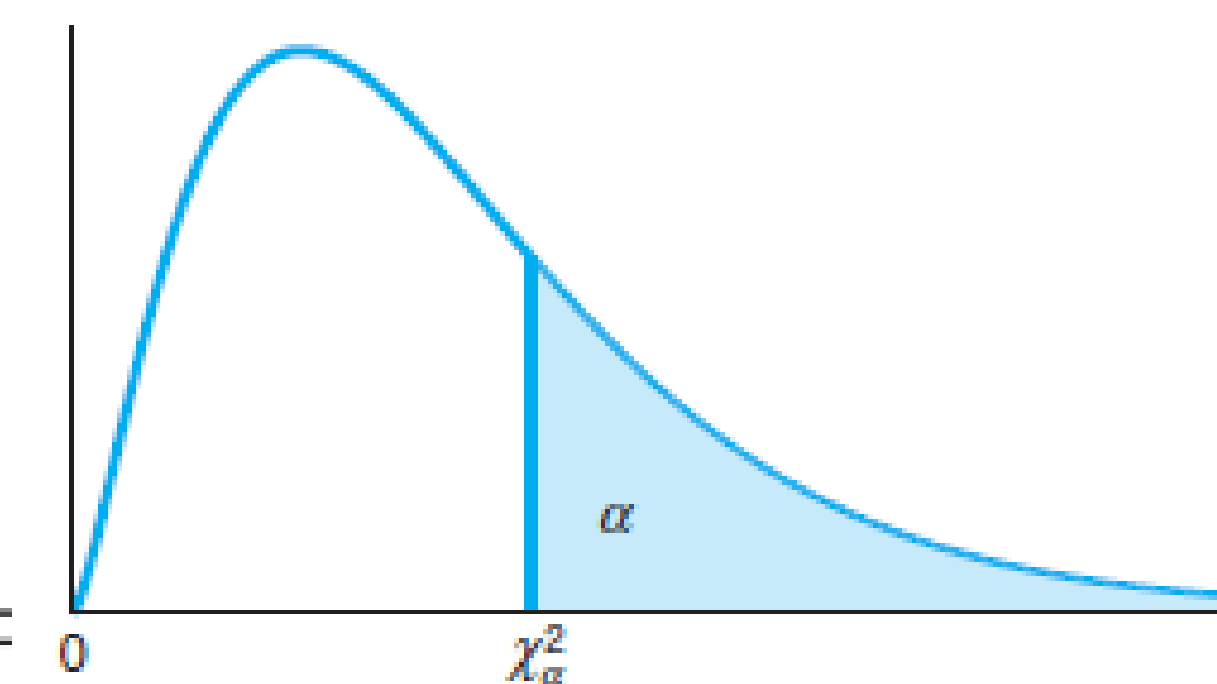
	Test 2 positive	Test 2 negative	Row total
Test 1 positive	a	b	$a + b$
Test 1 negative	c	d	$c + d$
Column total	$a + c$	$b + d$	N

- a. $\chi^2 > \chi^2_{\alpha}$: H0 ditolak
- b. $\chi^2 \leq \chi^2_{\alpha}$: H0 diterima

$$\chi^2 = \frac{(b - c)^2}{b + c}$$

Table A.5 (continued) Critical Values of the Chi-Squared Distribution

v	α									
	0.30	0.25	0.20	0.10	0.05	0.025	0.02	0.01	0.005	0.001
1	1.074	1.323	1.642	2.706	3.841	5.024	5.412	6.635	7.879	10.827
2	1.386	1.578	1.876	2.779	3.841	5.024	5.412	6.635	7.879	10.827



Contoh Penggunaan Uji Statistik McNemar

Table A.5 (continued) Critical Values of the Chi-Squared Distribution

$$\chi^2 = \frac{(b - c)^2}{b + c}$$

v	α									
	0.30	0.25	0.20	0.10	0.05	0.025	0.02	0.01	0.005	0.001
1	1.074	1.323	1.642	2.706	3.841	5.024	5.412	6.635	7.879	10.827
2	0.446	0.578	0.810	1.602	2.001	2.278	2.304	2.910	3.007	3.841

	M1: correct	M1: incorrect	Row total
M2: correct	101	121	222
M2: incorrect	59	33	92
Column total	160	154	314

Statistics = $(121-59)^2 / 121+59 = 21.35$
 Untuk alpha = 0.001, batas p-value adl 10.827 maka H_0 ditolak (ada perbedaan yang signifikan)

	M1: correct	M1: incorrect	Row total
M2: correct	59	6	65
M2: incorrect	16	80	96
Column total	75	86	161

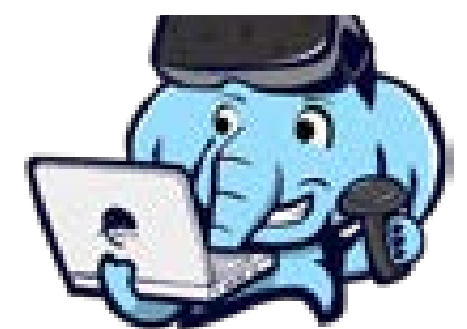
Statistics = $(6-16)^2 / 6+16 = 4.55$
 untuk alpha = 0.01, batas p-value adl 6.635
 maka H_0 diterima (tidak ada perbedaan yang signifikan)



Evaluasi dan Analisis Hasil (2)

Setelah eksperimen selesai dikerjakan, lakukan:

- **Analisis Error**
 - Analisis kesalahan yang dibuat oleh model, misalnya dengan melihat confusion matrix untuk klasifikasi atau dengan meninjau residuals untuk regresi.
 - Apakah ada pola dalam kesalahan model? Apakah model salah dalam kasus-kasus tertentu yang bisa dihindari? Apakah model mengalami overfitting atau underfitting?



Evaluasi dan Analisis Hasil (3)

- Example: confusion matrix of cat classifier
- But it does not help understand what is being misclassified, so we isolate the 50 and 100 images respectively, so we look at the images that have been misclassified and create some hypotheses/ observations on grouping these errors.
- Identify common errors across all erroneous cases:
 - 1. White cats not being correctly identified
 - 2. Dogs like cats being identified as cats
 - 3. Dark background leads to misclassification
 - and so on
- Map the exact distribution of the errors based on common errors
- Improve model by working on top common errors data correction first to have the most impact.

Cat Classifier - Confusion Matrix			
		Actual	
		Cat	Not Cat
Predicted	Cat	700	100
	Not Cat	50	150

Cat Classification Model Error Distribution		
Possible Hypothesis for error	Count	% of total Errors
1. White cats	27	18.0%
2. Dogs like cats being identified as cats	28	18.7%
3. Dark background	61	40.7%
4. Sphinx Cats !	22	14.7%

Evaluasi dan Analisis Hasil (6)

- **Konsistensi dengan Tujuan Bisnis**
 - Evaluasi hasil model dalam konteks tujuan bisnis atau penelitian awal. Apakah model membantu mencapai tujuan bisnis? Jika tidak, apa yang perlu diperbaiki?
- **Risiko dan Mitigasi**
 - Identifikasi potensi risiko atau masalah yang mungkin muncul jika model diimplementasikan dalam lingkungan nyata, seperti bias model atau sensitivitas terhadap outliers.
 - Apakah ada risiko yang perlu diperhatikan?
 - Bagaimana cara mitigasinya?



Menarik Kesimpulan

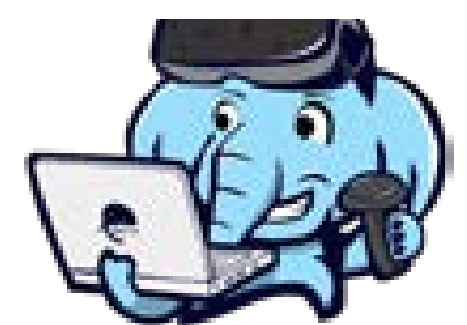
Berdasarkan hasil evaluasi dan analisis hasil, buat kesimpulan tentang:

- Hasil evaluasi kinerja model, termasuk evaluasi fitur dalam model dan identifikasi fitur yang signifikan.
- Analisis Error
- Evaluasi apakah hasil model mendukung tujuan bisnis atau eksperimen.
- Risiko dalam implementasi model



Hal-hal yang Perlu Diperhatikan

- Pembelajaran mesin bersifat iteratif
 - Pada eksperimen pertama, tidak lebih dari 25% sumber daya
 - *It is good idea not to start with high expectations, or promises.*
- Pengujian statistik tidak menunjukkan bahwa hipotesis yang dibangun benar, tetapi menunjukkan sebanyak apa samples yang mendukung hipotesis tersebut.
- Dokumentasikan seluruh proses eksperimen, hasil, dan kesimpulan



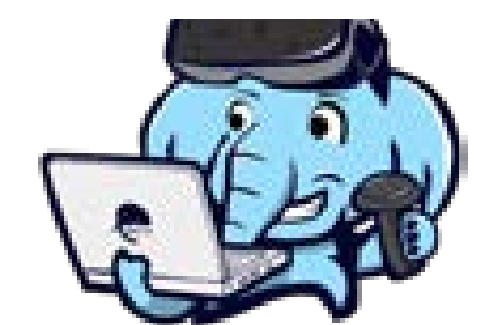
Hal-hal yang Perlu Diperhatikan

- Secara umum, kita membandingkan algoritma pembelajaran dengan menggunakan nilai error atau akurasi. Terdapat kriteria lain selain nilai error dalam memutuskan pemilihan algoritma:
 - Risiko
 - Waktu pelatihan dan kompleksitas ruang
 - Waktu pengujian dan kompleksitas ruang
 - Interpretabilitas (kemudahan untuk dipahami)
 - Kemudahan untuk diprogram



Hal-hal yang Perlu Diperhatikan

- Perhatikan aspek etis dalam pembangunan model *machine learning*, perhatikan bias yang terjadi yang berkaitan dengan etis, misal diskriminasi atau stereotyping kelompok tertentu



Pertanyaan ?