Machine Learning

Error metrics for Imbalance classes

Dr. Mehran Safayani

safayani@iut.ac.ir

safayani.iut.ac.ir

https://www.aparat.com/mehran.safayani

https://github.com/safayani/machine_learning_course

Cancer classification example

Train logistic regression model $h_\theta(x)$. ($y = 1$ if cancer, $y = 0$ otherwise)

$$Accuracy = \frac{\#correctly\ classified}{\#total\ number}$$

Find that you got 99% accuracy (1% error rate) on test set.

Only 0.50% of patients have cancer.

```
function y = predictCancer(x)
    y = 0; %ignore x!
return
```

99.5% accuracy

1000 → 995 correct, 5 wrong

$\frac{995}{1000} = 0.995$ → 99.5%

Precision/Recall

$y = 1$ in presence of rare class that we want to detect

confusion matrix

Predicted Value (predicted by the test)		Actual Value (as confirmed by experiment): positives	Actual Value (as confirmed by experiment): negatives
	positives	**TP** True Positive	**FP** False Positive
	negatives	**FN** False Negative	**TN** True Negative

Precision

(Of all patients where we predicted $y = 1$, what fraction actually has cancer?)

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

Recall

(Of all patients that actually have cancer, what fraction did we correctly detect as having cancer?)

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

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

	Actual positives	Actual Negatives
Predict positives	0	0
Predict negatives	10	90

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{90}{100} \quad Precision = \frac{TP}{TP + FP} = \frac{0}{0} \quad Recall = \frac{TP}{TP + FN} = \frac{0}{10}$$

	Actual positives	Actual Negatives
Predict positives	1	0
Predict negatives	9	90

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{91}{100} \quad Precision = \frac{TP}{TP + FP} = \frac{1}{1} \quad Recall = \frac{TP}{TP + FN} = \frac{1}{10}$$

	Actual positives	Actual Negatives
Predict positives	10	90
Predict negatives	0	0

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{10}{100} \quad Precision = \frac{TP}{TP + FP} = \frac{10}{100} \quad Recall = \frac{TP}{TP + FN} = \frac{10}{10}$$

- **high recall + high precision**：the class is perfectly handled by the model

- **low recall + high precision**：the model can't detect the class well but is highly trustable when it does

  Suppose we want to predict $y = 1$ (cancer) only if very confident.

- **high recall + low precision**：the class is well detected but the model also include points of other classes in it

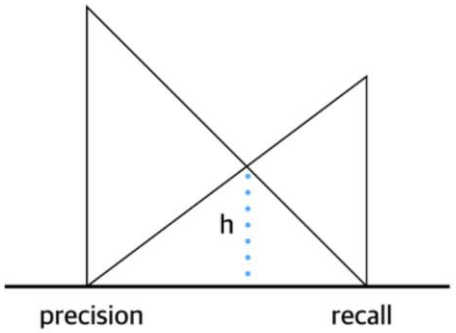
  Suppose we want to avoid missing too many cases of cancer

- **low recall + low precision**：the class is poorly handled by the model

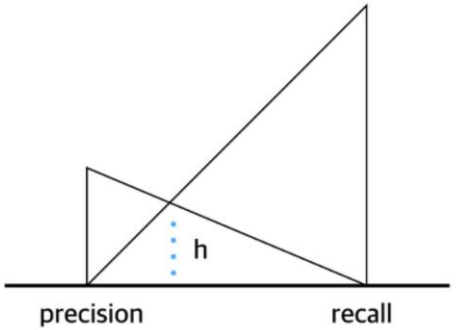
F_1 Score (F score)

How to compare precision/recall numbers?

	Precision(P)	Recall (R)		
Algorithm 1	0.5	0.4	0.45	0.44
Algorithm 2	0.7	0.1	0.4	0.18
Algorithm 3	0.02	1.0	y=1 → 0.5	0.04



h is half the harmonic mean



Average: $\frac{P+R}{2}$

F_1 Score: $2\frac{PR}{P+R}$ harmonic mean

Trading off precision and recall

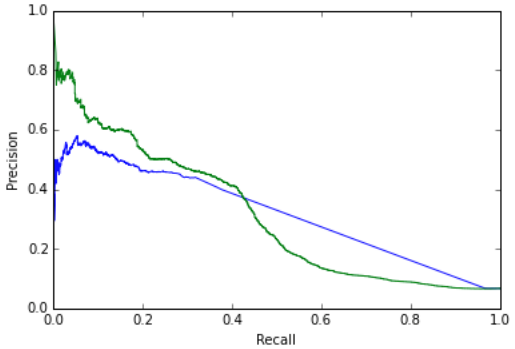
the precision-recall curve shows how the recall vs precision relationship changes as we vary the threshold for identifying a positive in our model.

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

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

Logistic regression: $0 \leq h_\theta(x) \leq 1$
Predict 1 if $h_\theta(x) \geq 0.5$
Predict 0 if $h_\theta(x) < 0.5$

More generally: Predict 1 if $h_\theta(x) \geq$ threshold.



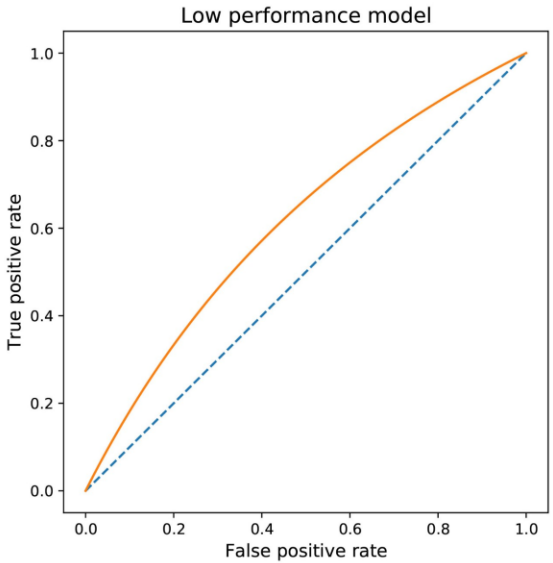
Suppose we want to predict $y = 1$ (cancer) only if very confident. (higher threshold; higher precision; lower recall)

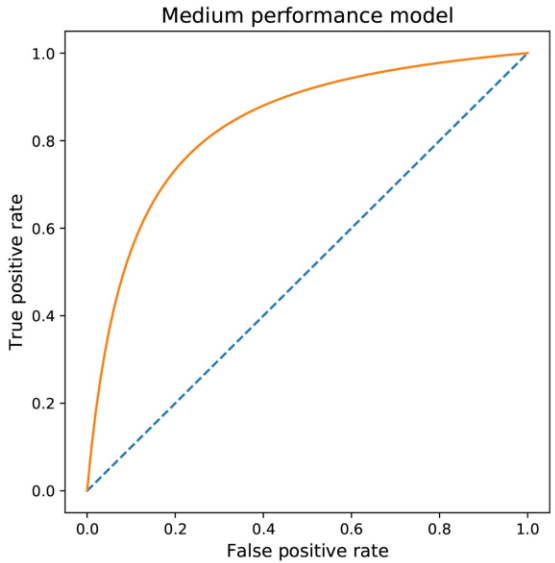
Suppose we want to avoid missing too many cases of cancer (avoid false negatives). (lower threshold; higher recall; lower precision)

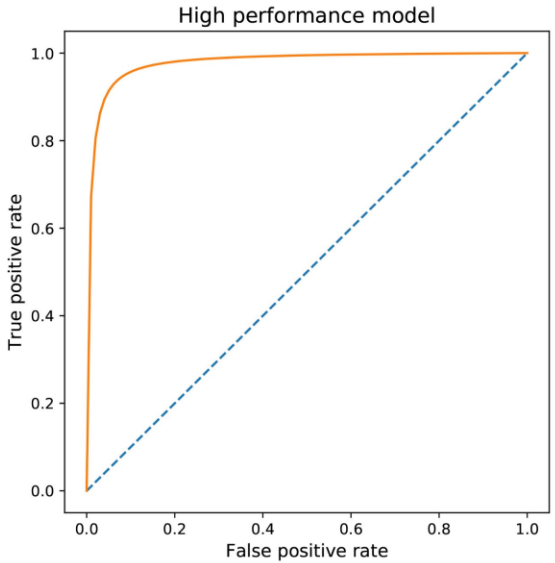
Receiver Operating Characteristic (ROC)

		Actual Value (as confirmed by experiment)	
		positives	negatives
Predicted Value (predicted by the test)	positives	**TP** True Positive	**FP** False Positive
	negatives	**FN** False Negative	**TN** True Negative

$$True\ positive\ rate = \frac{TP}{TP + FN} \quad False\ positive\ rate = \frac{FP}{FP + TN}$$







F_β-Measure

$$F_\beta = \frac{(1+\beta^2) * precision * recall}{\beta^2 * precision + recall}$$

- **F0.5-Measure** (β=0.5): More weight on precision, less weight on recall.
- **F1-Measure** (β=1.0): Balance the weight on precision and recall.
- **F2-Measure** (β =2.0): Less weight on precision, more weight on recall

- Let's take a case with **low Recall**.
- Precision = 0.9 (High)
- Recall = 0.2 (Low)
- **F1** = 2 * (0.9 * 0.2) / (0.9 + 0.2) = 0.36 / 1.1 ≈ **0.33**
- **F2** = 5 * (0.9 * 0.2) / (4*0.9 + 0.2) = 0.9 / (3.6 + 0.2) = 0.9 / 3.8 ≈ **0.24**

Now, let's take a case with **low Precision** (the reverse).

- Precision = 0.2 (Low)
- Recall = 0.9 (High)
- **F1** = 2 * (0.2 * 0.9) / (0.2 + 0.9) = 0.36 / 1.1 ≈ **0.33** (Same F1 as before)
- **F2** = 5 * (0.2 * 0.9) / (4*0.2 + 0.9) = 0.9 / (0.8 + 0.9) = 0.9 / 1.7 ≈ **0.53**

Classic Use Cases:

- **Medical Screening (e.g., for a serious disease):**
    1. **False Negative (Low Recall):** A sick patient is told they are healthy. They don't get treatment. This is **very bad**.
    2. **False Positive (Low Precision):** A healthy patient is flagged for more tests. This causes stress and extra cost, but the mistake can be caught later. This is **less bad**.
    3. **Goal:** Maximize Recall → Use F2.
- **Legal Document Review (e-discovery):**
    1. **False Negative (Low Recall):** A critical, relevant document is missed and not presented as evidence. This could lose the case.
    2. **False Positive (Low Precision):** A few irrelevant documents are sent for a lawyer to review. They waste time but can discard them.
    3. **Goal:** Maximize Recall → Use F2.

- **Spam Email Filtering:**

- **False Positive (Low Precision):** A legitimate, important email (like a job offer, flight confirmation, or receipt) is incorrectly marked as spam and sent to the junk folder. The user might never see it, potentially missing critical opportunities. This is **very bad** for user trust and satisfaction.

- **False Negative (Low Recall):** Some spam emails make it into the primary inbox. The user can quickly identify and delete them. This is **annoying, but less bad.**

- **Goal:** Be extremely certain when labeling something as spam → **Maximize Precision** → Use **F0.5-Measure**.

Multi-Class Metrics

		Actual Values			
		A	**B**	**C**	**D**
Predictions	**A**	9	1	5	0
	B	1	15	0	4
	C	0	3	24	1
	D	0	1	1	15

True Positive

		Actual Values			
		A	**B**	**C**	**D**
Predictions	**A**	9	1	5	0
	B	1	15	0	4
	C	0	3	24	1
	D	0	1	1	15

correctly identified prediction for each class

True Negative for class A

		Actual Values			
		A	**B**	**C**	**D**
Predictions	**A**	9	1	5	0
	B	1	15	0	4
	C	0	3	24	1
	D	0	1	1	15

correctly rejected prediction for certain class A

True Negative for class D

		Actual Values			
		A	**B**	**C**	**D**
Predictions	**A**	9	1	5	0
	B	1	15	0	4
	C	0	3	24	1
	D	0	1	1	15

correctly rejected prediction for certain class D

False Positive for class A

		Actual Values			
		A	**B**	**C**	**D**
Predictions	**A**	9	1	5	0
	B	1	15	0	4
	C	0	3	24	1
	D	0	1	1	15

Incorrectly identified prediction for certain class A

False Positive for class B

		Actual Values			
		A	**B**	**C**	**D**
Predictions	**A**	9	1	5	0
	B	1	15	0	4
	C	0	3	24	1
	D	0	1	1	15

Incorrectly identified prediction for certain class B

False Negative for class A

		Actual Values			
		A	**B**	**C**	**D**
Predictions	**A**	9	1	5	0
	B	1	15	0	4
	C	0	3	24	1
	D	0	1	1	15

Incorrectly rejected for certain class A

Accuracy

Accuracy is calculated as the total number of correct predictions divided by the total number of datasets

		Actual Values			
		A	**B**	**C**	**D**
Predictions	**A**	9	1	5	0
	B	1	15	0	4
	C	0	3	24	1
	D	0	1	1	15

Accuracy=(9+15+24+15)/80=0.78

Balance Data

Accuracy=32/40=0.8

		Actual Values			
		A	B	C	D
Predictions	A	10	0	0	0
	B	0	5	1	1
	C	0	3	8	0
	D	0	2	1	9

Accuracy=29/40=0.725

		Actual Values			
		A	B	C	D
Predictions	A	8	1	0	2
	B	2	7	0	3
	C	0	0	9	0
	D	0	2	1	5

Imbalance Data

Accuracy=126/230=0.547

		Actual Values			
		A	B	C	D
Predictions	A	100	0	0	0
	B	80	9	1	1
	C	10	0	8	0
	D	10	1	1	9

Accuracy=201/230=0.87

		Actual Values			
		A	B	C	D
Predictions	A	198	7	0	2
	B	2	1	8	3
	C	0	0	1	4
	D	0	2	1	1

Precision for Model 1 (Macro Average)

		Actual Values					
		A	B	C	D		
Predictions	A	100	0	0	0	TP=100	FP=0
	B	80	9	1	1	TP=9	FP=82
	C	10	0	8	0	TP=8	FP=10
	D	10	1	1	9	TP=9	FP=12

Precision=TP/(TP+FP) P(A)=1 P(B)=9/91 P(C)=8/18 P(D)=9/21

Macro Average Precision=[P(A)+P(B)+P(C)+P(D)]/4=0.492

Recall for Model 1 (Macro Average)

		Actual Values			
		A	**B**	**C**	**D**
Predictions	**A**	100	0	0	0
	B	80	9	1	1
	C	10	0	8	0
	D	10	1	1	9
		TP=100 FN=100	TP=9 FN=1	TP=8 FN=2	TP=9 FN=1

Recall=TP/(TP+FN) R(A)=100/200 R(B)=9/10 R(C)=8/10 R(D)=9/10

Macro Average Recall=[R(A)+R(B)+R(C)+R(D)]/4=0.775

F1 Score for Model 1

		Actual Values			
		A	**B**	**C**	**D**
Predictions	**A**	100	0	0	0
	B	80	9	1	1
	C	10	0	8	0
	D	10	1	1	9

F1 Score=2*(Recall*Precision)/(Precision +Recall)

F1 Score=2*[0.492*0.775]/[0.492+0.775]=0.601

Imbalance Data

Accuracy=0.547
F1 Score=0.601

		Actual Values			
		A	B	C	D
Predictions	A	100	0	0	0
	B	80	9	1	1
	C	10	0	8	0
	D	10	1	1	9

Accuracy=0.87
F1 Score=0.342

		Actual Values			
		A	B	C	D
Predictions	A	198	7	0	2
	B	2	1	8	3
	C	0	0	1	4
	D	0	2	1	1

Learning for Imbalance Data:**Undersampling, oversampling and generating synthetic data**

•undersampling consists in sampling from the majority class in order to keep only a part of these points

•oversampling consists in replicating some points from the minority class in order to increase its cardinality

•generating synthetic data consists in creating new synthetic points from the minority class (see SMOTE method for example) to increase its cardinality

References and further readings

Andrew NG., Machine Learning Course, Coursera, slide: Error metrics for skewed classes

Minsuk Heo. “Performance measure on multiclass classification [accuracy, f1 score, precision, recall] .” *YouTube*, 3 May. 2020, https://www.youtube.com/watch?v=HBi-P5j0Kec

Baptiste Rocca, “Handling imbalanced datasets in machine learning”, 3 may 2020, https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28