Metrics-Card.xlsm

Agenda:
1. ROC
2. F1 Score
Extra:
3. Naive Bayes Classifier
4. The algorithm
5. Numpy implementation
6. Time Space Complexity

---

# ROC

(Receiver operating characteristic)

→ originally for obtaining & receiving signals

Context: Binary Classification (1/0)

ROC → ML

Step 1: Sort $\hat{y}$ → Probabilities in desc. order.

$O(n \log n)$

Step 2: For each $\hat{y}$ (i) use as a threshold ($t$)

(a) $t_i = 0.95$ if $\hat{y}^{(i)} \geq 0.95$ → 1, else → 0

| X | $y_{true}$ | $\hat{y}$ |
|---|---|---|
| x¹ | 1 | 0.95 |
| x² | 1 | 0.86 |
| x³ | 0 | 0.81 |
| x⁴ | 1 | 0.72 |
| x⁵ | 1 | 0.32 |
| | | 0.06 |

← Model(x⁽ⁱ⁾)

$y_{true}$ & $y_{pred}$

$TPR_2$, $FPR_2$

$\frac{TP}{P}$, $\frac{FP}{N}$

(b) $t_2 = 0.86$

Get $y_{pred}$ if $\hat{y} \geq 0.86$ → 1, else → 0

$y_{true}$ & $y_{pred}$

$TPR_2$, $FPR_2$

$t_5 = 0.32$

| X | $y_{true}$ | $\hat{y}$ | $y_{pred}$ | $y_{true}$ & $y_{pred}$ |
|---|---|---|---|---|
| x¹ | 1 | 0.95 | 1 | TP |
| x² | 1 | 0.86 | 1 | TP |
| x³ | 0 | 0.81 | 1 | FP |
| x⁴ | 1 | 0.72 | 1 | TP |
| x⁵ | 1 | 0.32 | 1 | TP |
| | | 0.06 | 0 | TN |

$TPR_2$, $FPR_2$

$TPR_5$, $FPR_5$

Tuples of $(TPR_i, FPR_i)$

| | | |
|---|---|---|
| 0.95 | $TPR_1$, $FPR_1$ | |
| 0.86 | $TPR_2$, $FPR_2$ | |
| | $TPR_3$, $FPR_3$ | |
| | $TPR_4$, $FPR_4$ | |
| | $TPR_5$, $FPR_5$ | |

Plot

ROC

$TPR_{NV} = 1$

$FPR_{NV} = 0$




→ AU-ROC or AUC

Model A > Model B

AUC (Best Model) → 1

Fundamental shift:

$f_1$ (Recall / Precision)

↳ fixed threshold $t = 0.5$

"ROC is all possible thresholds"

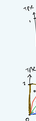
# Random model:

AUC = 0.5

# Properties of AUC ROC

Step ① Deport work with unbalanced data → Sensitive better & accurate

Step ② Random models (AUC = 0.5) (bad)

③ 


Inverse model

$\begin{bmatrix} 1 & \to & 0 \\ 0 & \to & 1 \end{bmatrix}$

"Flip"

④ AUC-ROC doesn't depend on $\hat{y}^{(i)}$ value.

- rather depends on its order of $\hat{y}^{(i)}$

| Sample | $y_{true}$ | $y_{pred}$ | $y_{true}$ & $y_{pred}$ | $y_{true}$ & $y_{pred}$ (mix) |
|---|---|---|---|---|
| a | 1 | 0.95 | 1 | 1 |
| b | 0.91 | 0.6 | 0 | 0 |
| c | 0.81 | 0.6 | 0 | 0 |
| d | 0.72 | 0.6 | 0 | 0 |
| e | 0.32 | 0.6 | 0 | 0 |
| f | 0.11 | 0.06 | 0 | 0 |
| g | | | 0 | 0 |
| h | | | 0 | 0 |
| i | | | 0 | 0 |
| j | | | 0 | 0 |

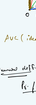
Time wise ROC.

↳ Doesn't work with imbalanced data

Highly Imbalanced

↳ $f_1$, Recall, Precision.

# P.R. Curve (Precision Recall Curve) (PRC)




---

# KNN

K - nearest neighbours




KNN

→ Supervised ML

→ for classification & Regression.

→ Non-linear model

Classwise Intuition.




→ ground-truth

→ neighbors

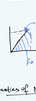
→ Test-sample

K = 3

Tgoung → get prediction

# Algorithm:

Inputs, Dtrain, $X_{train}$, $y_{train}$, Tgoung, k = 5



Algo:

① For each data point in Dtrain.

→ Compute distance ($d(x, x^{(i)})$)

→ store in distance list.

② Sort the dist list in asc. order

③ Pick top k nearest neighbors.

④ Majority Vote.

$y_{pred} \to 3$ Positive

$\to -1$ $y_{true} = -1$

# Binary classification: K = odd number.

---

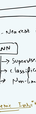
# Cross-Validation

⇒ You need big data for it to be fancy.

# K-Fold CV (Cross-validation)

---

# Dataset

Polynomial Reg → Poly features + linear reg.

Polynomial feat.