Evaluating

Friday, October 18, 2019 9:10 PM

		INDUCER PREDICTION (IP)	
		-1	+1
ACTUAL CLASS (AC)	-1	TN	FP
	+1	FN	TP

Metriche di valutazione

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

← numero delle predizioni corrette (diagonale principale)

← Totale dei records (delle predizioni)

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

← n° predizioni errate (diagonale secondaria)

← Totale dei records (delle predizioni)

Inoltre: ERROR = 1 − ACCURACY → Error e Accuracy sono complementari.

Un modello di classificazione che classifica la maggior parte dei records con una classe 0 che nel caso del dominio non è significativa, nonostante abbia un'accuratezza alta è comunque utile?

No, in questo caso non è utile, in quanto verrà applicata la: ZeroR Rule

Ovvero il modello classificherà i dati solamente secondo la più frequente classe, e non secondo le classi più rare.

In questo caso quindi come fare? Come procedere?
Come si può sviluppare un modello di classificazione che tenga in considerazione anche le classi meno importanti con un certo grado di significatività?

L'accuratezza misurata tratta ogni classe con la stessa importanza, e quindi non è una metrica efficiente per valutare le performance di un modello su un dataset sbilanciato. In quanto la classe più rara può essere più interessante per il dominio rispetto alla classe più frequente.

Per la classificazione binaria, la classe rara è solitamente denotata come la **classe** più **positiva**, mentre la classe più frequente è solitamente considerata come la **classe negativa**

Consideriamo quindi la matrice di confusione:

		Inducer Prediction (IP)	
		-1	+1
Actual Class (AC)	-1	TN	FP
	+1	FN	TP

Possiamo quindi generare le seguenti nuove metriche percentuali:

- **TNR, True Negative Rate or Specificity**
  Fraction of negative records predicted correctly by the Classification Model.

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

- **TPR, True Positive Rate or Sensitivity**
  Fraction of positive records predicted correctly by the Classification Model.

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

=> Recall

- **FPR, False Positive Rate**
  Fraction of negative records predicted as a positive class by the Classification Model.

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

- **FNR, False Negative Rate**
  Fraction of positive records predicted as a negative class by the Classification Model.

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

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

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

Recall and Precision

Recall and Precision sono due metriche che vengono largamente impiegate in applicazioni dove il rilevamento e l'interpretazione di una delle classi è considerato più importante rispetto alle altre classe all'interno del dataset.

$$\text{Precision, } p = \frac{TP}{TP + FP}$$

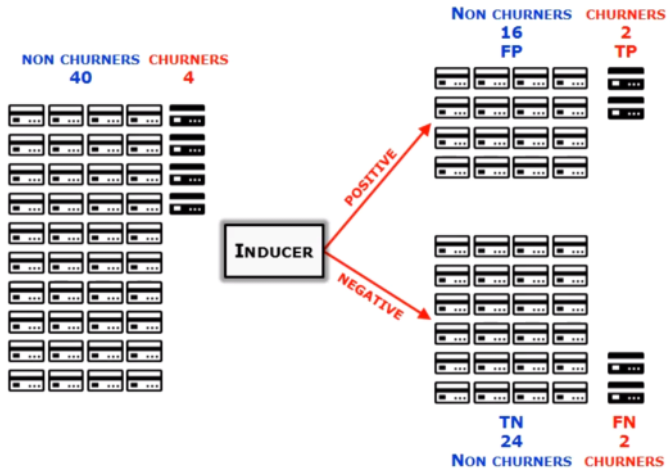
La Precision determina quindi: la frazione di records che risultano effettivamente positivi nel gruppo che il modello di classificazione ha dichiarato come

classe positiva.
Più **alta è la precisione**, **minore è il numero degli errori** falsi positivi comessi dal modello di classificazione.

$$\text{Recall, } r = \frac{TP}{TP + FN}$$

Recall misura la frazione dei records positivi correttamente predetti dal modello di classificazione.
Avere un **alta recall** significa che molto pochi record positivi sono erroneamente classificati come classe negativa.
In realtà la Recall è equivalente al True Positive Rate (TPR)

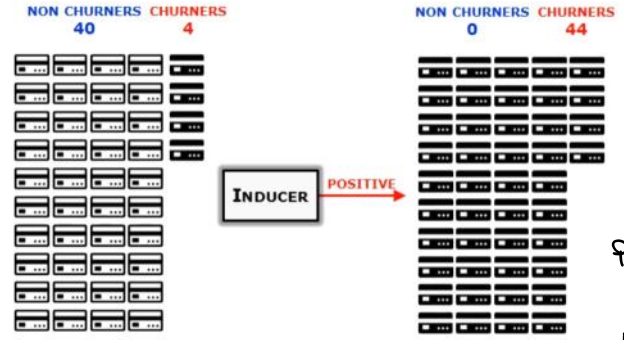
Esempio:



$$p = \frac{TP}{TP+FP} \rightarrow \frac{2}{2+16} = 0,11$$

$$r = \frac{TP}{TP+FN} \rightarrow \frac{2}{2+2} = 0,5$$

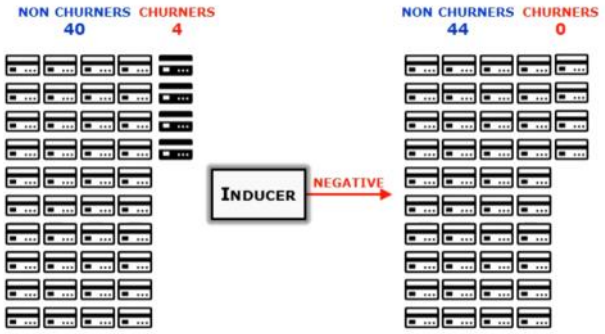
Esempio:



$$p = \frac{TP}{TP+FP} = \frac{4}{4+40} = 0,09$$

$$r = \frac{TP}{TP+FN} = \frac{4}{4+0} = 1$$

Oppure:



$$p = \frac{TP}{TP+FP} = \frac{0}{0+0} = ?$$

$$r = \frac{TP}{TP+FN} = \frac{0}{0+4} = 0$$

F1 Measure

È spesso possibile costruire modelli Baseline che massimizzano una metrica, ma non l'altra. (le due metriche sono quindi dipendenti)

Precision and Recall quindi sono solitamente sommati in un'altra misura chiamata: F1 Measure che rappresenta la loro media armonica.

$$F_1 = \frac{2 \cdot r \cdot p}{r + p}$$

In particolare un alto valore di F1 Measure assicura che sia la Precision che la Recall siano ragionevolmente alte, e quindi che il modello performi bene anche sulle classi minori e meno frequenti all'interno del modello.

Esiste una generalizzazione della F1 Measure che consente di esaminare l'effettivo tradeoff tra Recall e Precision

$$F_\beta = \frac{(\beta^2 + 1) \cdot r \cdot p}{r + \beta^2 \cdot p}$$ = F_β measure

- F_β with $\beta = 0$ → Precision
- F_β with $\beta = \infty$ → Recall

Counting the cost

Si vuole partizionare un dataset per ottenere un subset e successivamente calcolare:

- Training dataset
- Testing dataset

Si vuole quindi pianificare uno studio di classificazione usando il training data set per sviluppare un differente modello di classificazione che verrà poi comparato sul test set.

Si seleziona quindi il modello di classificazione M che fornisce l'ottimo inducer da usare per costruire la decisione di fare o non fare una particolare campagna promozionale su un determinato utente.
Per verificare le performance del modello, si prova a confrontare i risultati con un altro modello fatto precedentemente su un task simile, ottenendo le matrici di confusione finali:

SMP		PREDICTED CLASS	
		-1	+1
ACTUAL CLASS	-1	830	111
	+1	45	114

ACCURACY = 0.858

M		PREDICTED CLASS	
		-1	+1
ACTUAL CLASS	-1	931	10
	+1	57	102

ACCURACY = 0.939

Possiamo quindi verificare che il modello di classificazione M permette di ottenere una performance migliore sul Test Data Set indipendentemente.

Tuttavia il costo di classificare un customer churner come un non churner è molto più alto del classificare il costo del non churner come churner.

Possiamo quindi generare la seguente matrice di Costo e calcolare i costi per il modello SMP e per il modello M

	PREDICTED CLASS	
	-1	+1
ACTUAL CLASS -1	0	1
ACTUAL CLASS +1	100	-1

COST(SMP) = 4,497

COST(M) = 5,608

Siamo quindi nella situazione in cui l'accuratezza è migliore per il modello M, ma il costo è superiore, preferendo il modello SMP

Come fare quindi per selezionare il migliore modello di classificazione con queste metriche considerate?

Per ovviare a questo problema, utilizziamo le seguenti metriche.
Consideriamo le seguenti tabelle:

		PREDICTED CLASS	
		-1	+1
ACTUAL CLASS	-1	TN	FP
	+1	FN	TP

		PREDICTED CLASS	
		-1	+1
ACTUAL CLASS	-1	C_{--}	C_{-+}
	+1	C_{+-}	C_{++}

$$\text{Cost} = C_{--} \cdot TN + C_{-+} \cdot FP + C_{+-} \cdot FN + C_{++} \cdot TP$$

Un esempio del calcolo del costo:

$(TN_2 \cdot TN_1) + (FP_2 \cdot FP_1) + (FN_2 \cdot FN_1) + (TP_1 \cdot TP_2)$ = costo totale

$$\text{Cost} = 0 \cdot 830 + 1 \cdot 111 + 100 \cdot 45 + (-1) \cdot 114 = 4{,}497$$

		PREDICTED CLASS	
		-1	+1
ACTUAL CLASS	-1	830	111
	+1	45	114

		PREDICTED CLASS	
		-1	+1
ACTUAL CLASS	-1	0	1
	+1	100	-1

Attenzione! I coefficienti di costo sono simmetrici, ma il costo è proporzionale all'accuratezza.

		PREDICTED CLASS	
		-1	+1
ACTUAL CLASS	-1	TN	FP
	+1	FN	TP

		PREDICTED CLASS	
		-1	+1
ACTUAL CLASS	-1	p	q
	+1	q	p

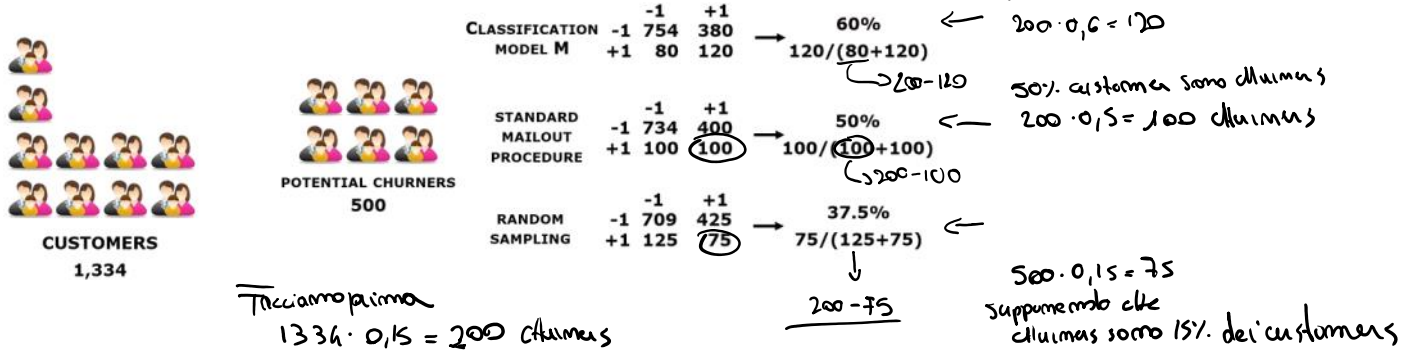
$$N = TP + FN + FP + TN$$

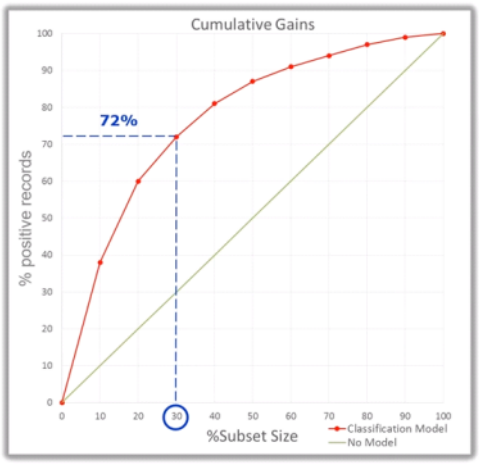
$$\text{ACCURACY} = (TP+TN)/N$$

$$\begin{aligned} \text{COST} &= p \cdot (TP+TN) + q \cdot (FN+FP) \\ &= p \cdot (TP+TN) + q \cdot (N-TP-TN) \\ &= q \cdot N - (q-p) \cdot (TP+TN) \\ &= N \cdot [q - (q-p) \cdot \text{ACCURACY}] \end{aligned}$$

Cumulative Gains

https://www.ibm.com/support/knowledgecenter/en/SSLVMB_23.0.0/spss/tutorials/mlp_bankloan_outputtype_02.html
Si può anche verificare la situazione in cui abbiamo delle labels all'interno del nostro dataset che sono molto sbilanciate tra di loro, a tal proposito è importante ad esempio estrarre delle informazioni utili dal dataset per selezionare solamente la classe più sbilanciata a campione.
Per fare questo potremmo utilizzare alcune tecniche:





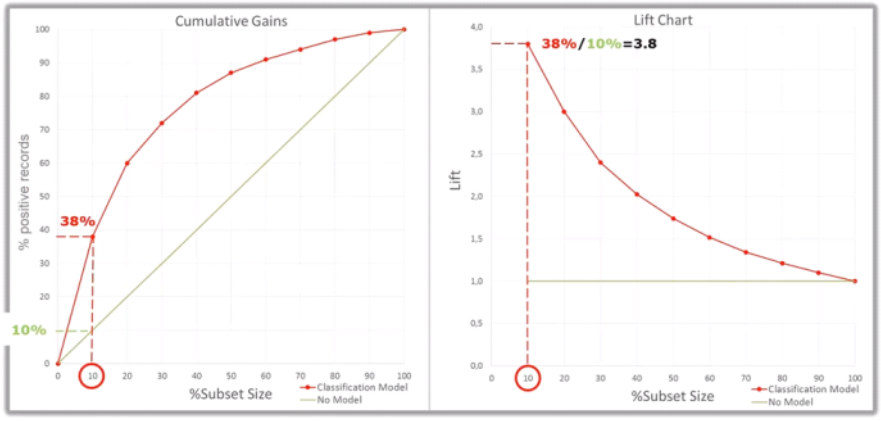
YOU SELECT a given number of customers to mailout, i.e. **1,000 CUSTOMERS**.

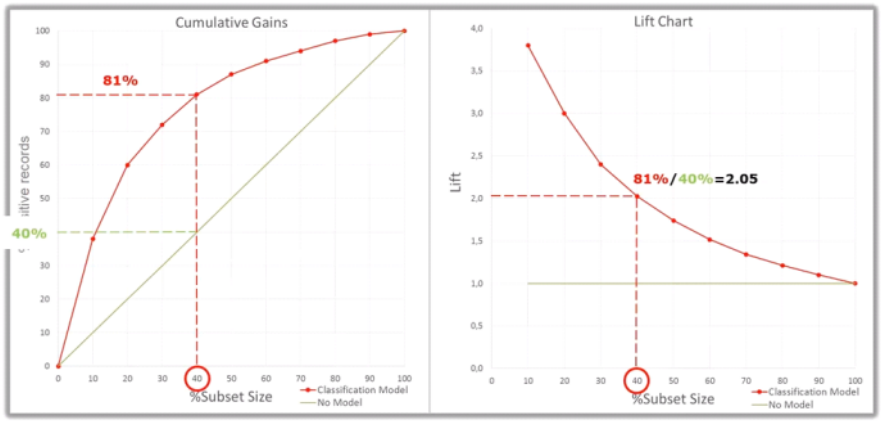
Assume it corresponds to the **10% OF THE CUSTOMERS SET**.

IF YOU SELECT 3,000 CUSTOMERS then it corresponds to the **30% OF THE CUSTOMERS SET.**

Cumulative Gains = selezionando il 30% di records, il modello performa il 72% sui records positivi

Lift Chart





LIFT

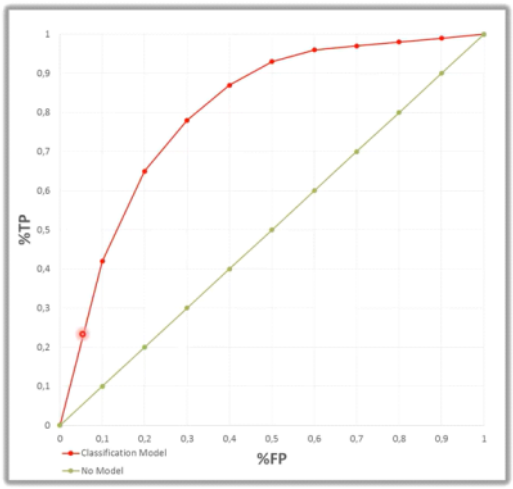
$$\frac{\text{Cumulative Gain (Positive Records)}}{\text{Lift (\% subset size)}}$$

Lift Chars in realtà è simile ad una rappresentazione grafica per valutare i modelli di classificazione chiamata: ROC Curve

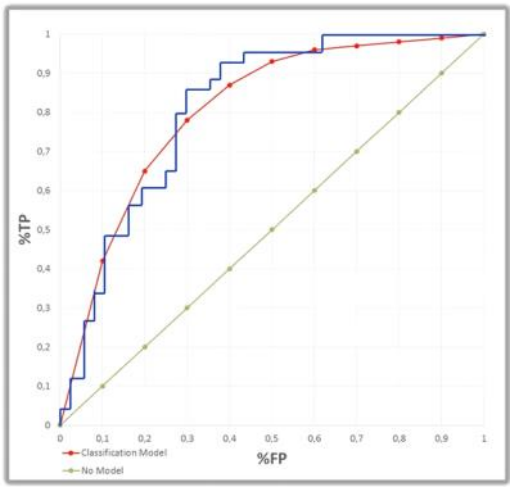
ROC Curve

ROC = Receiver Operating Characteristic Curve

La curva di Roc plotta il **numero di record positivi** in un determinato **subset di records** sull'asse verticale (Y) espressi come la **percentuale del numero totale di record positivi** (% TP, TPR).
Mentre nell'asse delle X orizzontale si esprime il **numero dei record negativi** in un determinato subset di records, espresso come la **percentuale del numero dei record negativi** (%FP, FPR).

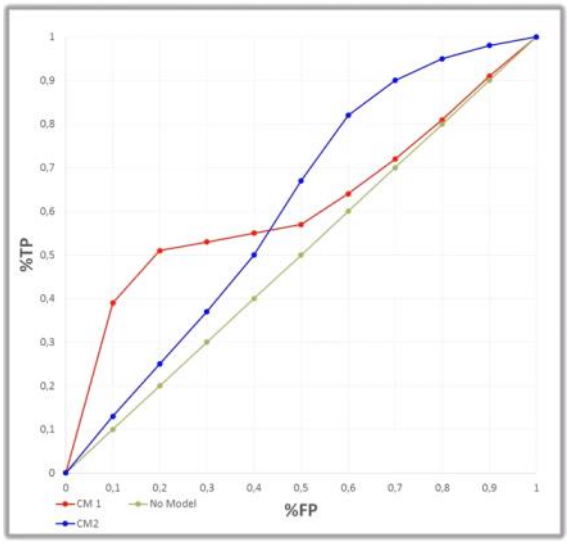


Piuttosto che la curva rossa espressa nell'esempio precedente, è più facile ottenere una **Jagged ROC Curve** che rappresenta meglio i dettagli di un particolare dataset.
Questa dipendenza e questa curva può essere ridotta applicando la **Cross Validation**



ATTENZIONE = la curva di ROC rappresenta le performance di un classificatore senza alcun riguardo o senza alcuna considerazione della distribuzione delle classi oppure del costo degli errori!

Come interpretare una curva di ROC?



Consideriamo questa immagine e in particolare i tre modelli raffigurati.

In questo esempio CM1 eccelle se si considera un piccolo e determinato subset di records.