Lecture 12: Classification error metrics

Madeleine Udell and Josh Grossman
Stanford University

Model inspection & evaluation

Calibration
Accuracy
Precision
Recall
Specificity
AUC
ROC curve

Model inspection & evaluation

Calibration

When you predict an event happens with probability ***p***, how often does it actually occur?

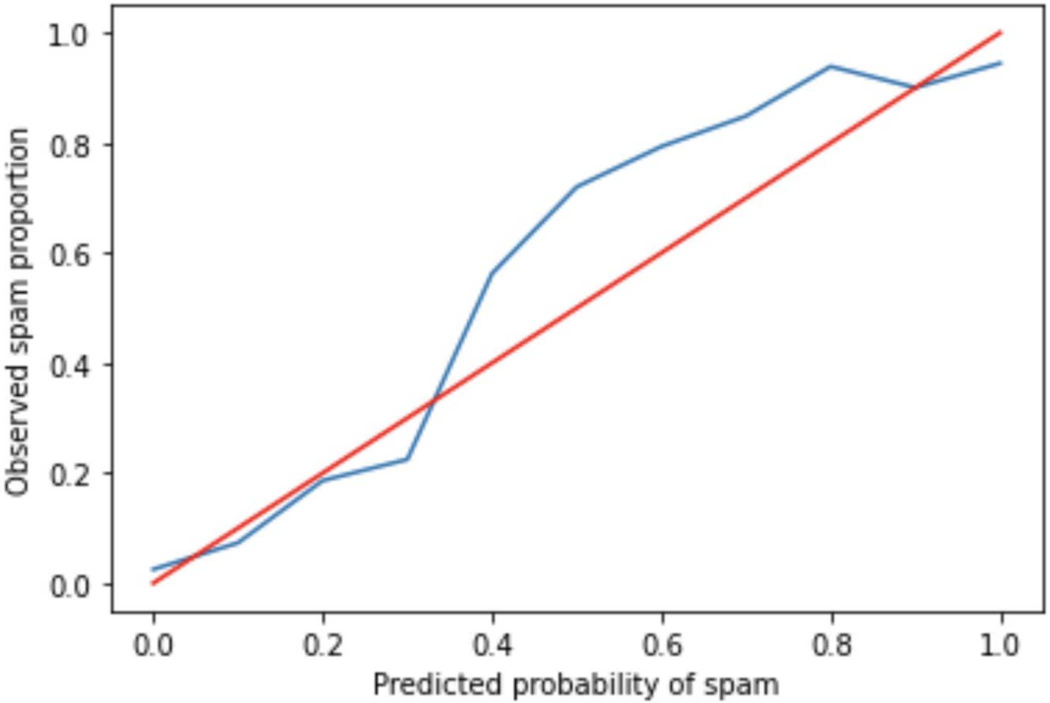
```python
spam['pred'] = pred
spam['rounded_pred'] = np.round(pred, 1)

spam[['pred', 'rounded_pred', 'is_spam']].head()
```

	pred	rounded_pred	is_spam
0	0.256057	0.3	1
1	0.871883	0.9	1
2	0.872221	0.9	1
3	0.256057	0.3	1
4	0.256057	0.3	1

```python
spam.groupby('rounded_pred')['is_spam'].mean().plot()
plt.plot([0,1], [0,1], color='red')
plt.xlabel("Predicted probability of spam")
plt.ylabel("Observed spam proportion")
```

```
Text(0, 0.5, 'Observed spam proportion')
```



Model inspection & evaluation

Accuracy

Percent of predictions that are “correct.”

To determine this, we must convert probabilistic predictions to binary predictions. For example, can convert probabilities to the most likely binary outcome.

60% → 1
45% → 0

```
np.mean(spam['is_spam'] == (spam['pred'] > 0.5))
```

```
0.8072158226472506
```

What is the lowest possible accuracy for any model predicting a binary outcome?

A) 50%
B) 0%
C) Depends on the problem
D) The frequency of the most common class (e.g., 60%, if 60% of the outcomes are 0's)

What is the lowest possible accuracy for any model predicting a binary outcome *after training*?

A) 50%
B) 0%
C) Depends on the problem
D) The frequency of the most common class (e.g., 60%, if 60% of the outcomes are 0's)

```
np.mean(spam['is_spam'] == 0)
```

```
0.6059552271245382
```

50% accuracy is not the baseline! We can get 61% accuracy by always predicting FALSE.

Model inspection & evaluation

Type I and Type II errors

Type I errors FP [Model erroneously calls it spam]

Type II error FN [Model erroneously calls it legit email]

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

Model inspection & evaluation

Precision

Proportion of all “positive” predictions that are true.

[When the model says “positive”, how often it is correct.]

Percent of messages that are labeled “spam” that actually are.

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

```
np.mean(spam['is_spam'][spam['pred'] > 0.5])
```

```
0.901213171577123
```

Model inspection & evaluation

Recall [sensitivity, true positive rate]

Proportion of all true instances that are positive.

[Proportion of true instances the model correctly identifies.]

Percent of actual spam messages correctly labeled as “spam”.

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

```
np.mean(spam['pred'][spam['is_spam'] == 1] > 0.5)
```

```
0.573634859349145
```

Model inspection & evaluation

Specificity

Proportion of all false instances that are negative.

[Proportion of false instances the model correctly identifies.]

Percent of non-spam messages correctly labeled “non-spam”.

$$\frac{TN}{TN + FN}$$

```
np.mean(spam['pred'][spam['is_spam'] == 0] <= 0.5)
```

```
0.9591104734576757
```

Model inspection & evaluation

Trade offs

There is a trade-off between precision and recall.

[And between specificity and sensitivity.]

Model inspection & evaluation

Trade offs

How can you achieve perfect recall?

[Discuss with neighbors]

How can you achieve perfect recall on a binary prediction problem?

A) You must always make perfectly accurate predictions
B) Always predict 0
C) Always predict 1
D) Impossible to achieve perfect recall in every case

Model inspection & evaluation

Trade offs

How can you achieve perfect recall?

Call everything “positive”

[Set a low bar for calling instances “positive”.]

Model inspection & evaluation

Trade offs

How can you achieve perfect recall?

Call everything “positive”

[Set a low bar for calling instances “positive”.]

This strategy leads to many false positives.

[Low precision.]

Model inspection & evaluation

Trade offs

How can you achieve high precision?

How can you achieve high precision?

A) Always predict 1
B) Always predict 0
C) Only predict 1 when your estimated probability of a positive outcome is high, otherwise predict 0
D) Only predict 0 when your estimated probability of a negative outcome is high, otherwise predict 1

Model inspection & evaluation

Trade offs

How can you achieve high precision?

Set a high bar for calling instances “positive”.

Model inspection & evaluation

Trade offs

How can you achieve high precision?

Set a high bar for calling instances “positive”.

This strategy leads to many false negatives.
[Low recall.]

Probabilities to binary labels

Selecting the threshold

50% is not always the appropriate cutoff.

```python
def precision(threshold):
    return(np.mean(spam['is_spam'][spam['pred'] > threshold]))

def sensitivity(threshold):
    return(np.mean(spam['pred'][spam['is_spam'] == 1] > threshold))
```

What happens to precision and recall/sensitivity, relative to a 50% threshold?

[Discuss with neighbors]

What happens to sensitivity/recall when the threshold is raised?

A) Recall must stay the same or go up

B) Recall must stay the same or go down

C) Impossible to know without seeing the prediction problem

What happens to precision when the threshold is raised?

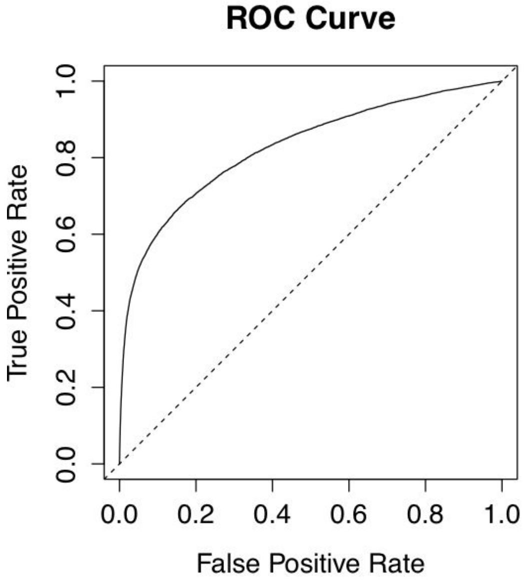
A) Precision must stay the same or go up
B) Precision must stay the same or go down
C) Impossible to know without seeing the prediction problem

But in most cases, (A) is true!

	threshold	accuracy	precision	sensitivity	specificity
0	0.0	0.394045	0.394045	1.0	0.0
1	0.25	0.52945	0.453291	0.942085	0.261119
2	0.5	0.807216	0.901213	0.573635	0.95911
3	0.75	0.76592	0.932941	0.437397	0.979555
4	1.0	0.605955	NaN	0.0	1.0

Model inspection and evaluation

ROC (Receiver Operating Characteristic) curve



$$\text{TPR} = \frac{\text{true positives}}{\text{total true}}$$

$$\text{FPR} = \frac{\text{false positives}}{\text{total false}}$$

What model achieves the dashed line?
[Discuss with neighbors]

What model achieves the dashed line?

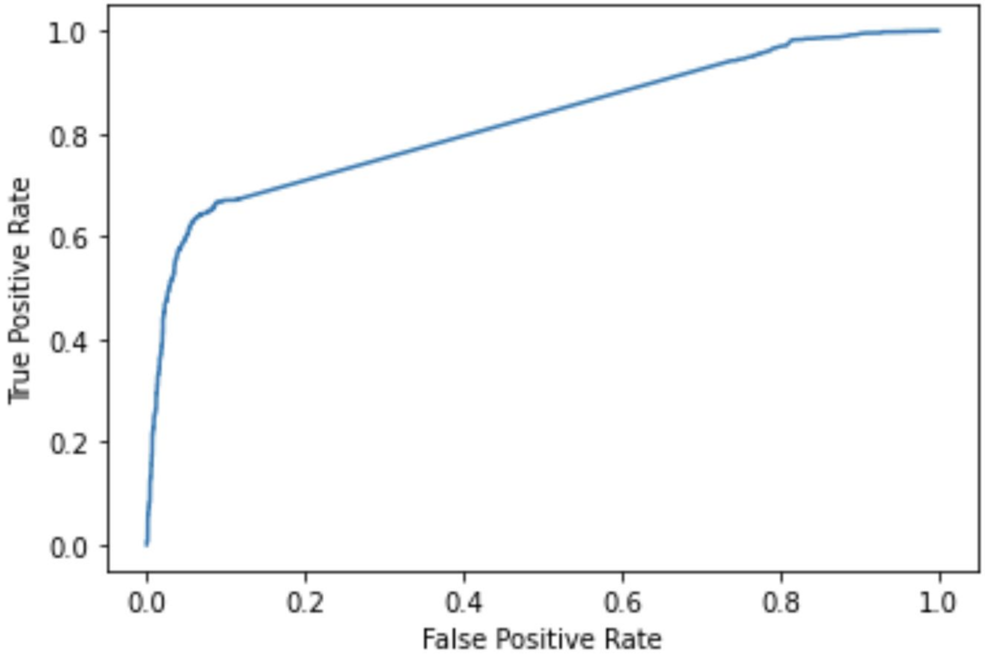
A) Always predict 0
B) Always predict 1

C) Choose a probability uniformly at random in [0,1]

D) Choose a probability iid N(0,1)

```
from sklearn.metrics import roc_curve
fpr, tpr, thresholds = roc_curve(spam['is_spam'], spam['pred'])
plt.plot(fpr, tpr)
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
```

```
Text(0, 0.5, 'True Positive Rate')
```



Model inspection & evaluation

Area under the ROC curve [AUC]

The probability that a classifier will score a randomly chosen true instance higher than a randomly chosen false one.
[Model correctly identifies the true instance in the pair.]

```python
from sklearn.metrics import auc
auc(fpr, tpr)
```

```
0.8230103841140938
```