Performance Metrics

Part I

Input to Evaluate the Performance

- Given:
  - y_test: the real class of each element of the test set
  - y_pred: the binary predictions for each element of the test set
  - y_proba: the class-membership probability of each element of the test set

- How to measure the predictive performance?

Predict on the Test set

Method ***predict:***
Returns an array of binary predictions
(one for each element of the test set)

Method ***predict_proba:***
Returns a *n*-by-2 matrix of probabilities of belonging to each class.
(i,0) is the probability that element i belongs to class 0
(i,1) is the probability that element i belongs to class 1

```
y_pred = cl.predict(X_test)
y_proba = cl.predict_proba(X_test)
```

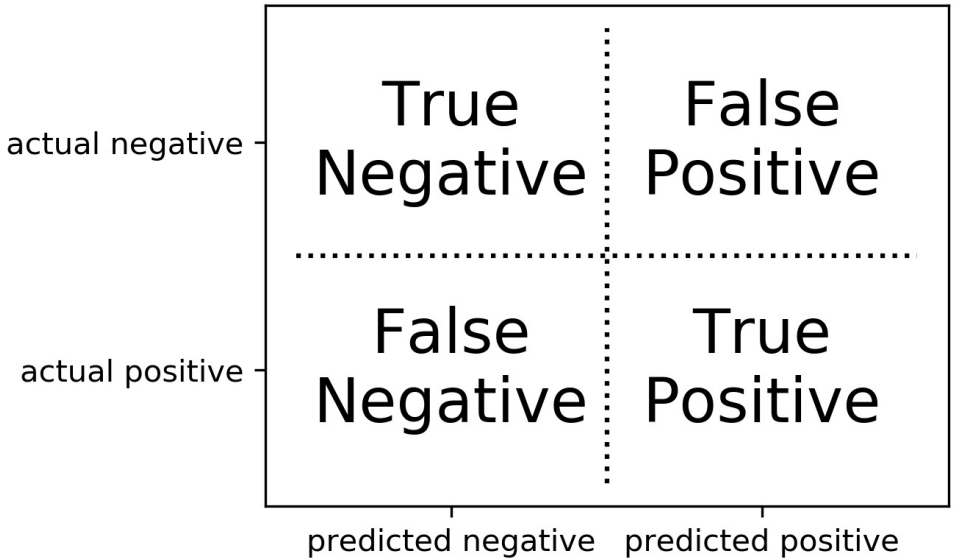
```
y_pred[:40]
```

```
array([ 1.,  1.,  0.,  0.,  1.,  0.,  0.,  0.,  0.,  1.,  0.,  0.,  0.,
        0.,  1.,  1.,  0.,  0.,  0.,  0.,  1.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  1.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  1.,  1.,  1.])
```

```
y_proba[:40]
```

```
array([[ 0.3       ,  0.7       ],
       [ 0.48833333,  0.51166667],
       [ 1.        ,  0.        ],
       [ 0.83333333,  0.16666667],
       [ 0.25714286,  0.74285714],
       [ 0.5       ,  0.5       ],
       [ 0.775     ,  0.225     ],
       [ 0.775     ,  0.225     ],
```

Confusion Matrix



POSITIVE AND NEGATIVE:
In most binary classification problems, we are interested in detecting one class, which is usually the minority class. That class is called POSITIVE; the other one is negative.

- Pregnancy test (positive = you are pregnant)
- Radar system (positive = threat detected)
- Searching for documents about bitcoin (positive = document is about bitcoin)

Confusion Matrix, Continued

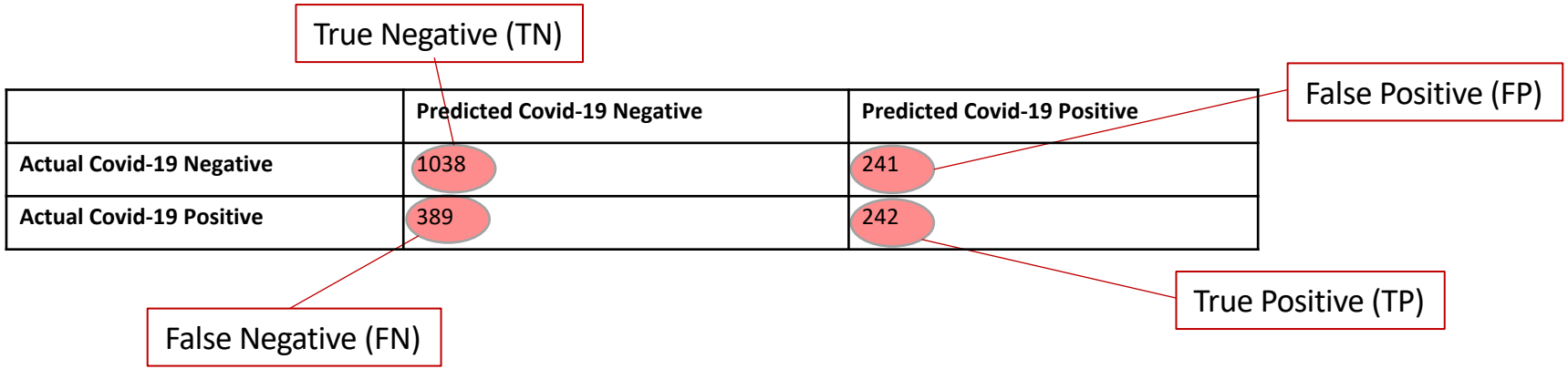
- **Confusion Matrix**: a matrix of all outcomes on the test set:

	Predicted Covid-19 Negative	**Predicted Covid-19 Positive**
Actual Covid-19 Negative	1038	241
Actual Covid-19 Positive	389	242

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

```
array([[1038,  241],
       [ 389,  242]])
```

True/False Positive/Negative



	Predicted Covid-19 Negative	Predicted Covid-19 Positive
Actual Covid-19 Negative	1038	241
Actual Covid-19 Positive	389	242

Accuracy

$$\text{Accuracy} = \frac{TN+TP}{n}$$

ACCURACY IS A BAD METRIC FOR IMBALANCED DATA

True Negative (TN)

	Predicted Covid-19 Negative	Predicted Covid-19 Positive
Actual Covid-19 Negative	1038	241
Actual Covid-19 Positive	389	242

True Positive (TP)

```
sklearn.metrics.accuracy_score(y_test,y_pred)
```

```
0.67015706806282727
```

Precision

Precision = $\frac{TP}{TP+FP}$. Out of the retrieved elements, how many are actually positive?

High precision means low “false alarm rate” (if you test positive, you’re probably positive)

	Predicted Covid-19 Negative	**Predicted Covid-19 Positive**
Actual Covid-19 Negative	1038	241
Actual Covid-19 Positive	389	242

False Positive (FP)

True Positive (TP)

```
sklearn.metrics.precision_score(y_test,y_pred)
```

```
0.50103519668737062
```

Recall (True Positive Rate/Sensitivity/Probability of detection)

Recall = $\frac{TP}{TP+FN}$. Among the relevant elements, how many did I retrieve?

High recall means you're not missing many positives.

	Predicted Covid-19 Negative	Predicted Covid-19 Positive
Actual Covid-19 Negative	1038	241
Actual Covid-19 Positive	389	242

False Negative (FN)

True Positive (TP)

```
sklearn.metrics.recall_score(y_test,y_pred)
```

```
0.38351822503961963
```

Cost of Different Types of Mistakes can be Different (and High) in Some Applications

	Spam filtering	Medical diagnosis
False negative	Annoying	Disease not treated
False positive	Email lost	Wasteful treatment

Tradeoff Between Precision and Recall

- Recall-oriented machine learning tasks:
  - Search and information extraction in legal discovery
  - Tumor detection
  - Often paired with a human expert to filter out false positives
- Precision-oriented machine learning tasks:
  - Search engine ranking, query suggestion
  - Document classification
  - Many customer-facing tasks (users remember failures!)

F1-Score and F-Score (Harmonic Mean of Precision and Recall)

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

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

	Precision	Recall	F_1	$F_{0.5}$	F_2
1	0.01	0.99	0.02	0.01	0.05
2	0.20	0.80	0.32	0.24	0.50
3	0.40	0.90	0.55	0.45	0.72
4	0.60	0.62	0.61	0.60	0.62
5	0.90	0.95	0.92	0.91	0.94

β allows adjustment of the metric to control the emphasis on recall vs precision:

- **Precision-oriented users: $\beta = 0.5$ (false positives hurt performance more than false negatives)**
- **Recall-oriented users: $\beta = 2$ (false negatives hurt performance more than false positives)**