

False positives = Test +, actually -

False positives = Test +, actually -

False positive rate (FPR, Type I error rate):

False positives = Test +, actually -

False positive rate (FPR, Type I error rate):

$$\alpha = P(\text{test says } + \mid \text{input is } -)$$

False positives = Test +, actually -

False positive rate (FPR, Type I error rate):

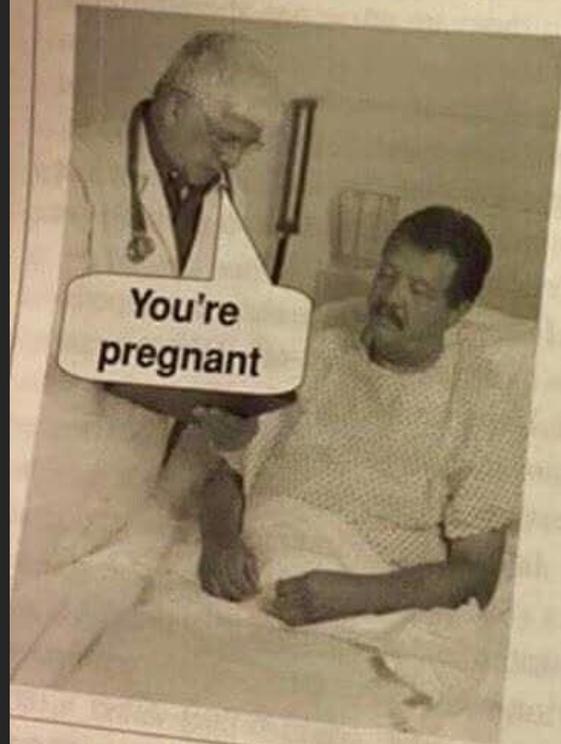
$$\alpha = P(\text{test says } + \mid \text{input is } -)$$

False negative = Test -, actually +

False negative rate (FNR, Type II error rate):

$$\beta = P(\text{test says } - \mid \text{input is } +)$$

Type I error
(false positive)



Type II error
(false negative)

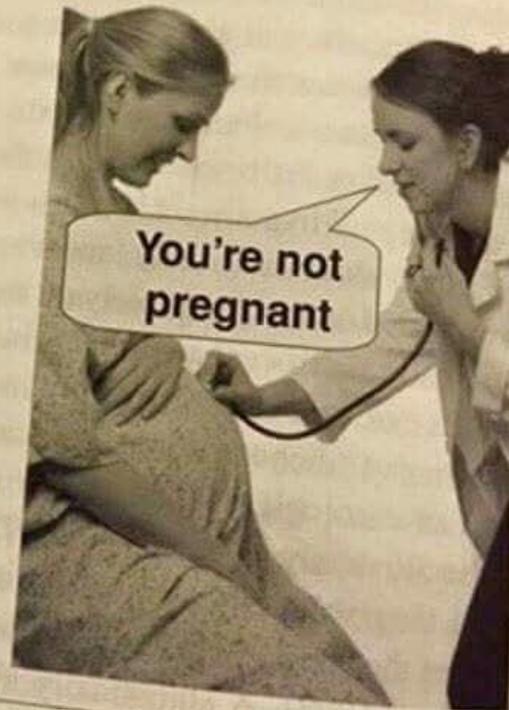


Figure 3.1 Type I and Type II errors

levels to .01 or even .001
%



Shobhit Bakliwal

@shobhitic



really efficient algorithm that tells whether a number is prime or not very very quickly with > 95% accuracy

```
6  √ bool is_prime(int x)
7
8  }
```

```
test no.99989: passed
test no.99990: passed
test no.99991: failed
test no.99992: passed
test no.99993: passed
test no.99994: passed
test no.99995: passed
test no.99996: passed
test no.99997: passed
test no.99998: passed
test no.99999: passed
95.121% tests passed
```



Shobhit Bakliwal

@shobhitic



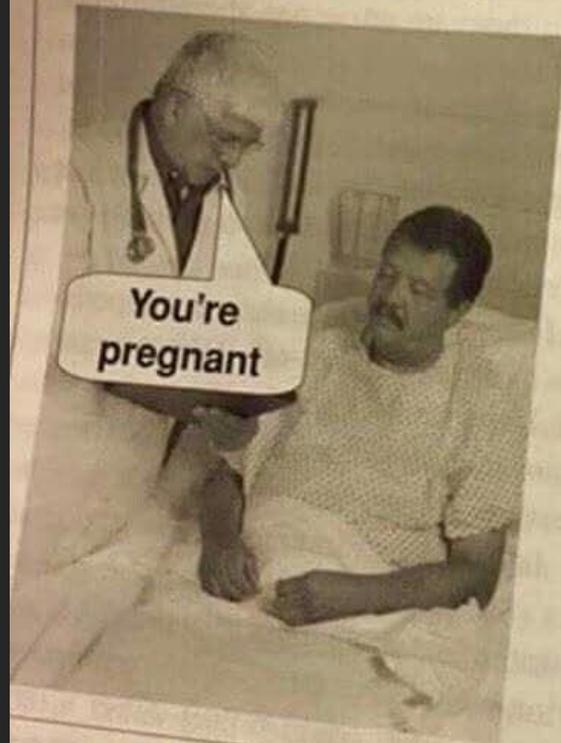
really efficient algorithm that tells whether a number is prime or not very very quickly with > 95% accuracy

```
6  ∨ bool is_prime(int x)
7      return false;
8  }
```

```
test no.99989: passed
test no.99990: passed
test no.99991: failed
test no.99992: passed
test no.99993: passed
test no.99994: passed
test no.99995: passed
test no.99996: passed
test no.99997: passed
test no.99998: passed
test no.99999: passed
95.121% tests passed
```

```
6  ✓ bool is_prime(int x)
7      return false;
8  }
```

Type I error
(false positive)



Type II error
(false negative)

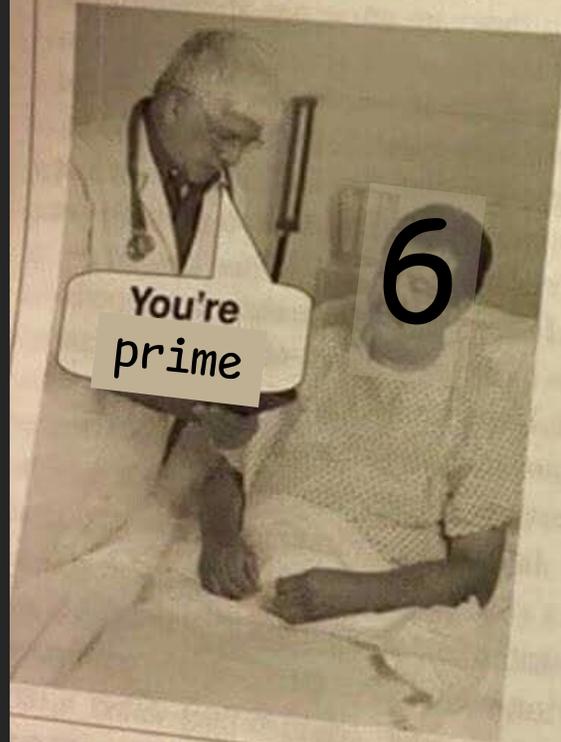


Figure 3.1 Type I and Type II errors

levels to .01 or even .001
%

```
6  ✓ bool is_prime(int x)
7      return false;
8  }
```

Type I error
(false positive)



Type II error
(false negative)

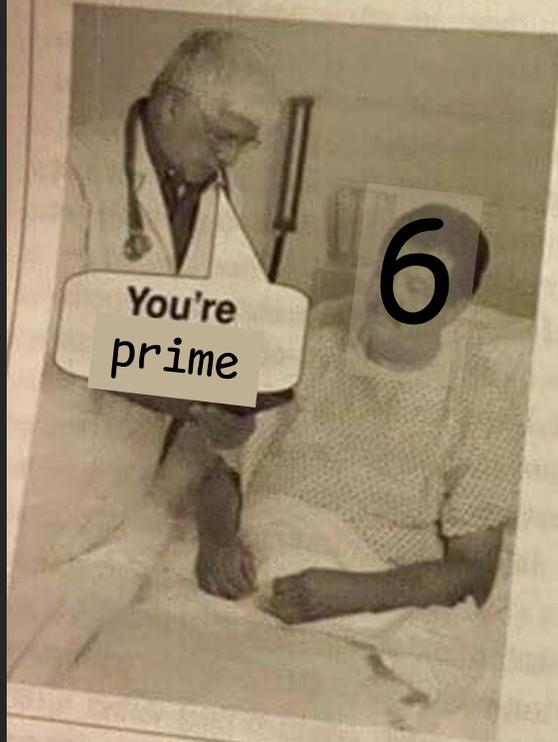


Figure 3.1 Type I and Type II errors

levels to .01 or even .001
%

```
6  ✓ bool is_prime(int x)
7      return false;
8  }
```

Type I error
(false positive)



Type II error
(false negative)

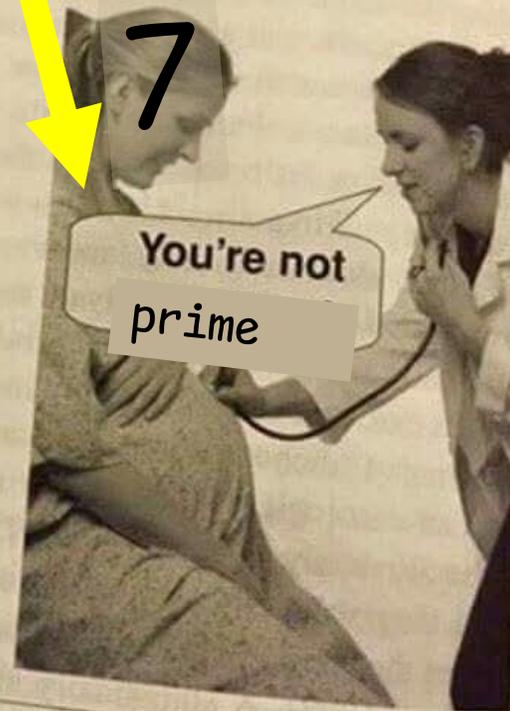
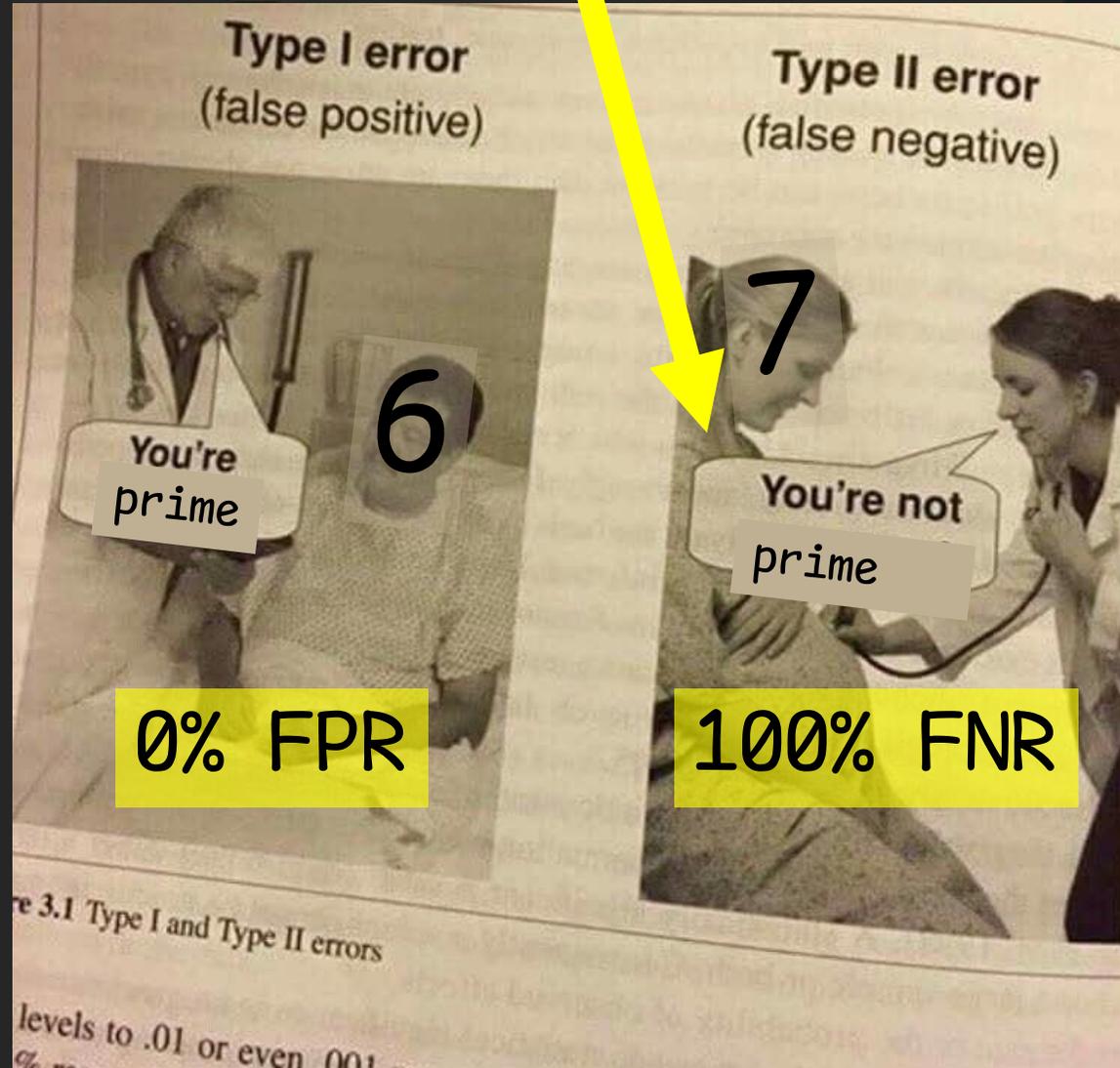


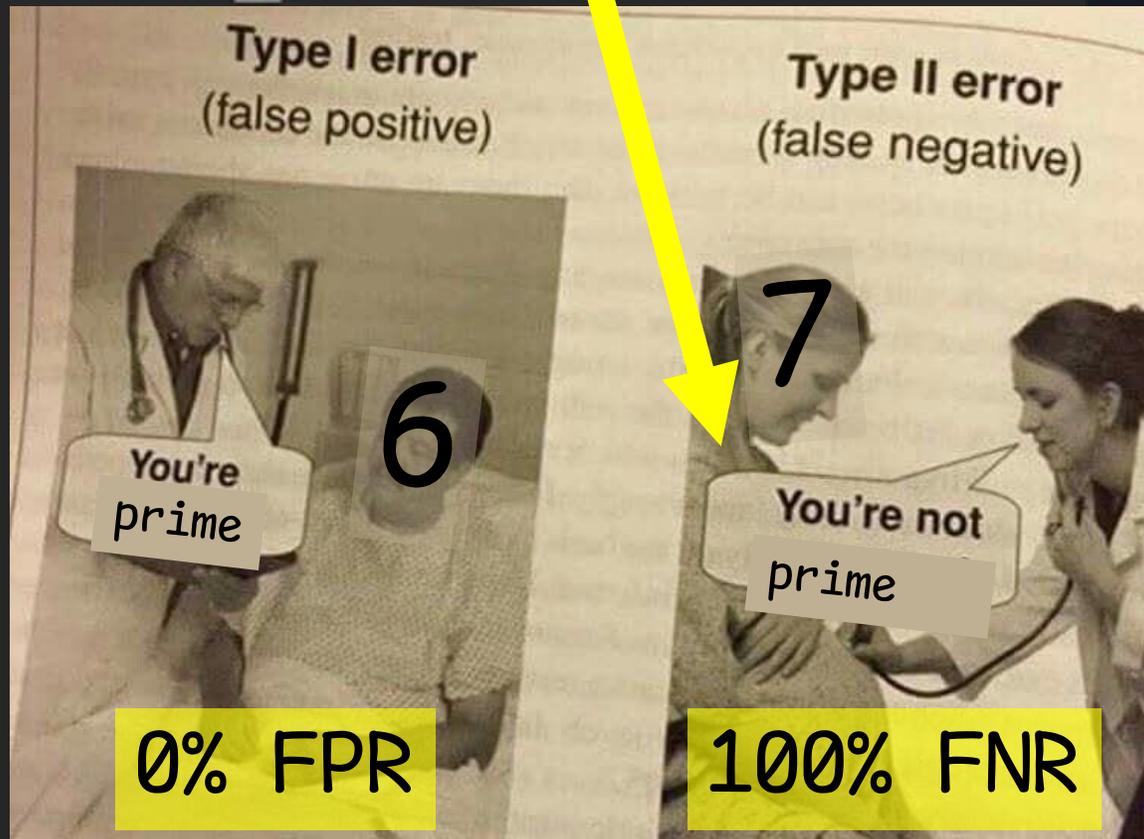
Figure 3.1 Type I and Type II errors

levels to .01 or even .001
%

```
6  ✓ bool is_prime(int x)
7      return false;
8  }
```



```
6  ✓ bool is_prime(int x)
7      return false;
8  }
```



Moral: accuracy is bad metric on very imbalanced data. Use FPR, FNR, F1-score instead.