Day 4: Evaluating Models Part 2

Micro-Averaging

$\hat{y}$ \ y	0	1	2
0	8	10	1
1	5	60	50
2	3	30	200

Class 0:

$\hat{y}$ \ y	0	not 0
0	8	11
not 0	8	340

Class 1:

$\hat{y}$ \ y	1	not 1
1	60	55
not 1	40	212

Class 2:

$\hat{y}$ \ y	2	not 2
2	200	33
not 2	51	83

Pooled Confusion Matrix:

	y yes	y no
$\hat{y}$ yes	268	99
$\hat{y}$ no	99	635

$$\text{Precision} = \frac{268}{268 + 99} = \boxed{.73}$$

F1 Score: harmonic mean of precision and recall

$$2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Evaluation Process

learn θ

test θ on "unseen" data

In pseudocode:

In ps...

```
X_data = [[3, 4, 2],     data point (d.p.) 1
          [3, 2, 1],     dp 2
          [4, 8, 3],     dp 3
          [1, 1, 1],     dp 4
          [3, 8, -5]]    dp 5

y_data = [0,    dp 1
          0,    dp 2
          1,    dp 3
          0,    dp 4
          1]    dp 5
```

Split data into train and test set:

```
X_train = [[3,4,2],          y_train = [0,
           [3,2,1],                     0,
           [4,8,3]]                     1]

X_test = [[1,1,1],           y_test = [0,
          [3,8,-5]]                    1]
```

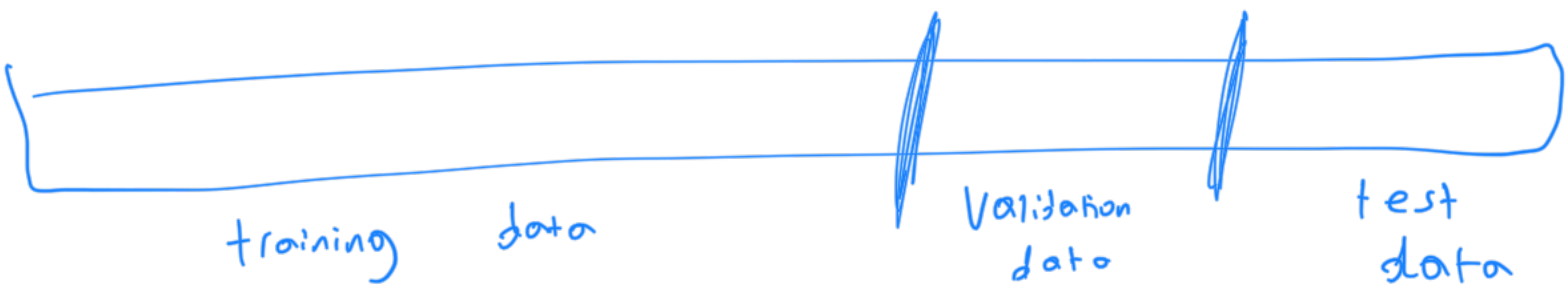
We train model on train data, test on test data:

```
model = LogisticRegression()
model.fit(X_train, y_train)
```

y_pred = model.predict(X_test)

f1_score = get_f1_score(y_test, y_pred)

Real World: Train / Validation / Test



" " tune hyperparameters (e.g., regularization strength λ, LR decision threshold, ...) " "

Receiver Operating Characteristic (ROC) Curves

Default LR cutoff = 0.5

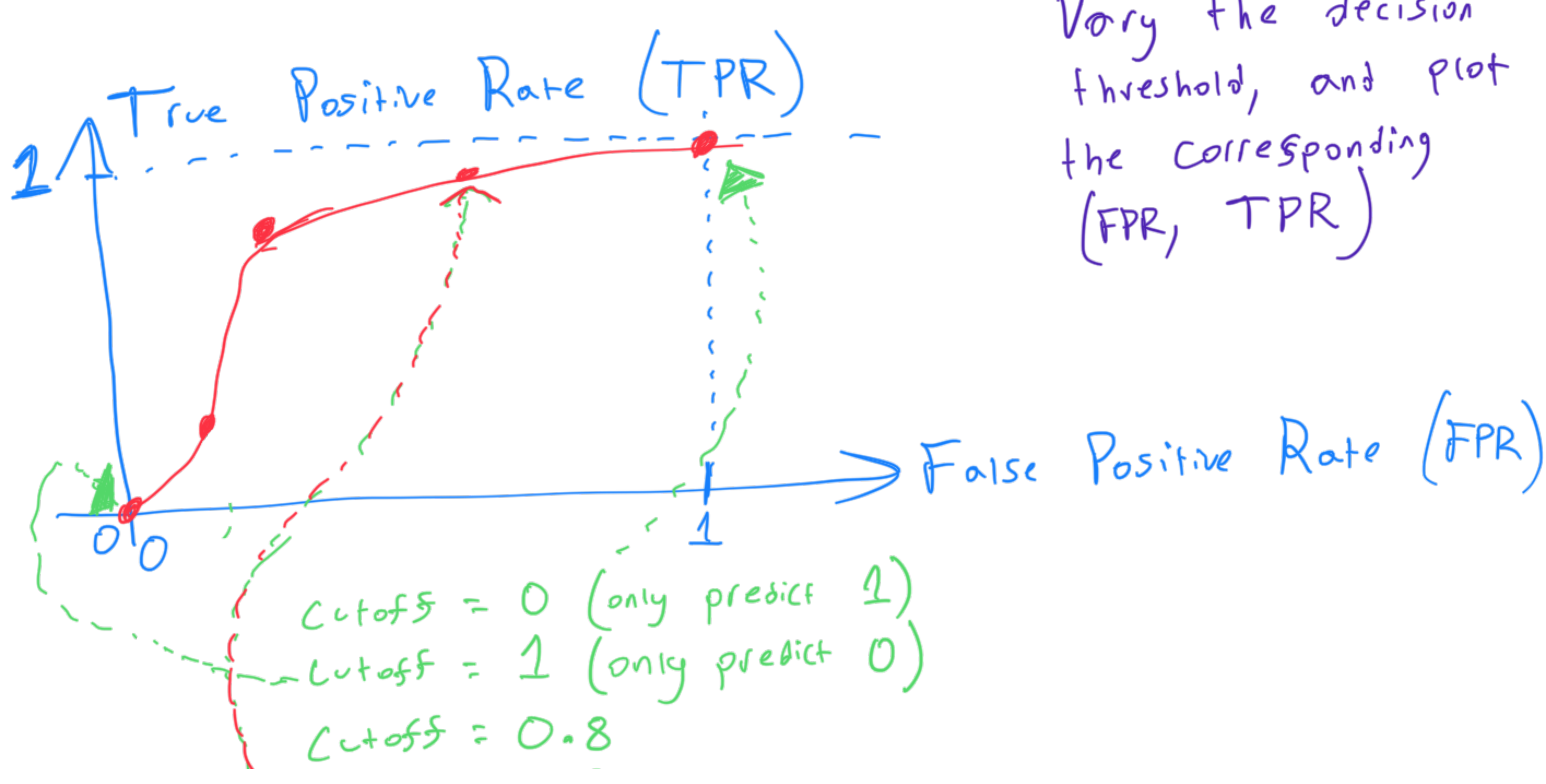
is $\hat{y} \geq 0.5 \longrightarrow$ "predict 1"

else $\longrightarrow$ "predict 0"

cutoff = threshold

Doesn't have to be 0.5

Vary the decision threshold, and plot the corresponding (FPR, TPR)

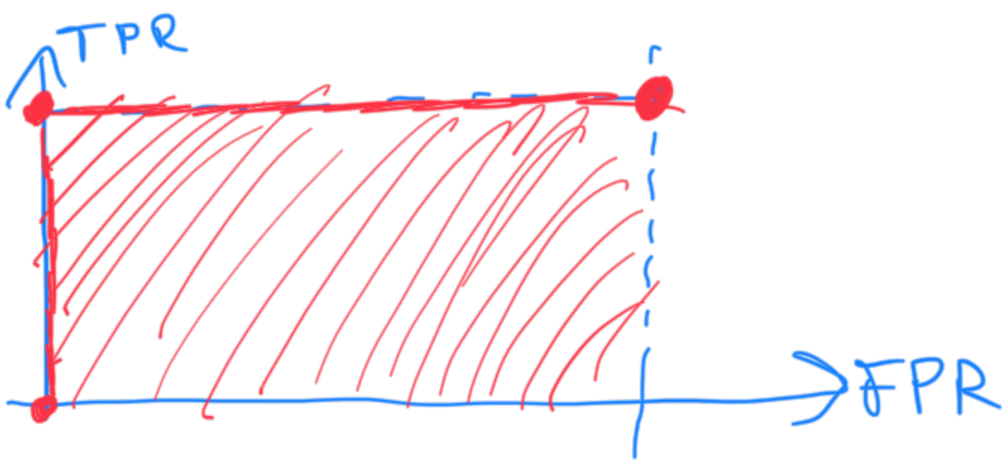


cutoff = 0.2

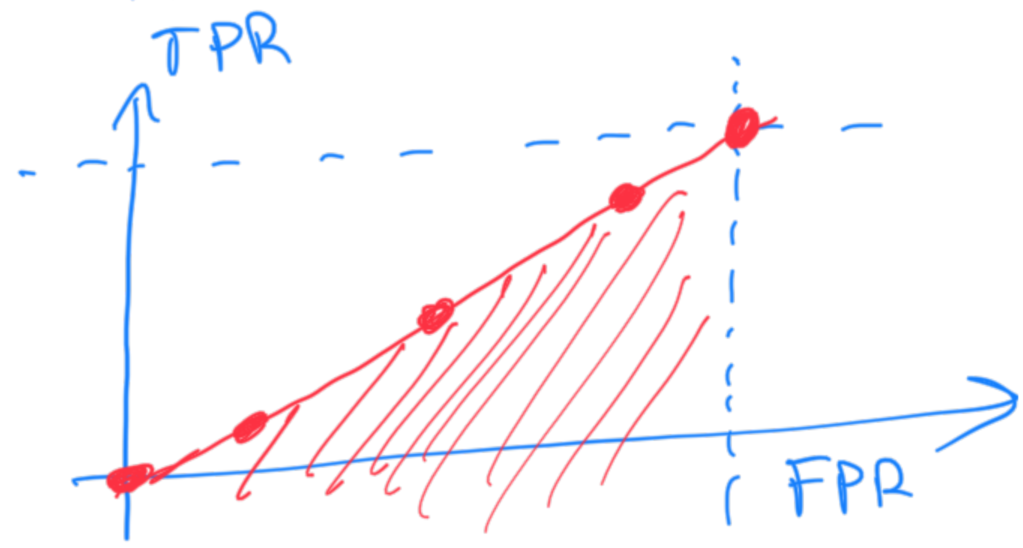
$$TPR = \text{sensitivity} = \text{recall} = \frac{TP}{TP + FN}$$

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

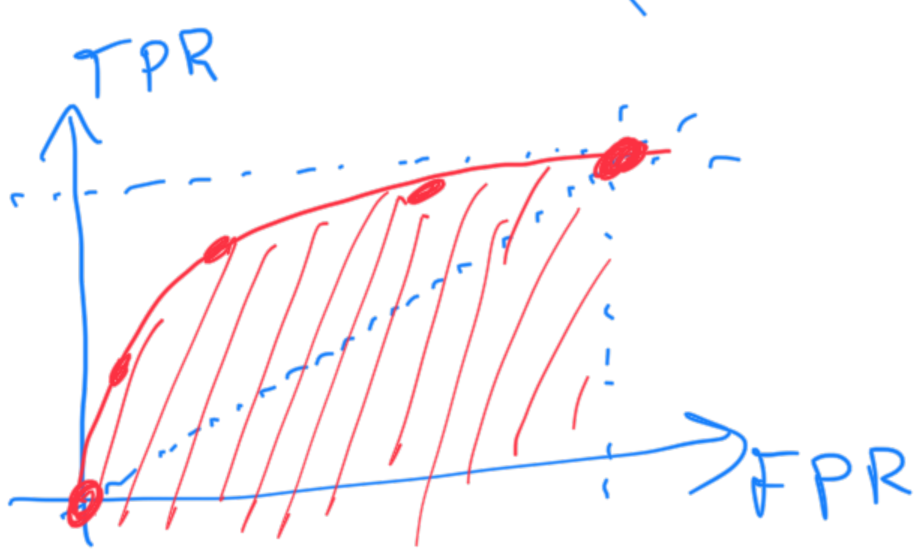
Perfect Model:



Random Model:



The Usual Case (better than random, not perfect):



Area Under the ROC (AUROC):

area under the ROC curve

Random model: AUROC = 0.5

Perfect model: AUROC = 1.0

In practice: 0.5 < AUROC < 1

with higher AUROC → better performance

Cross-Validation

dataset:

fold 1	train	train	train	train	test
fold 2	train	train	train	test	train
fold 3	train	train	test	train	train
fold 4	train	test	train	train	train
fold 5	test	train	train	train	train

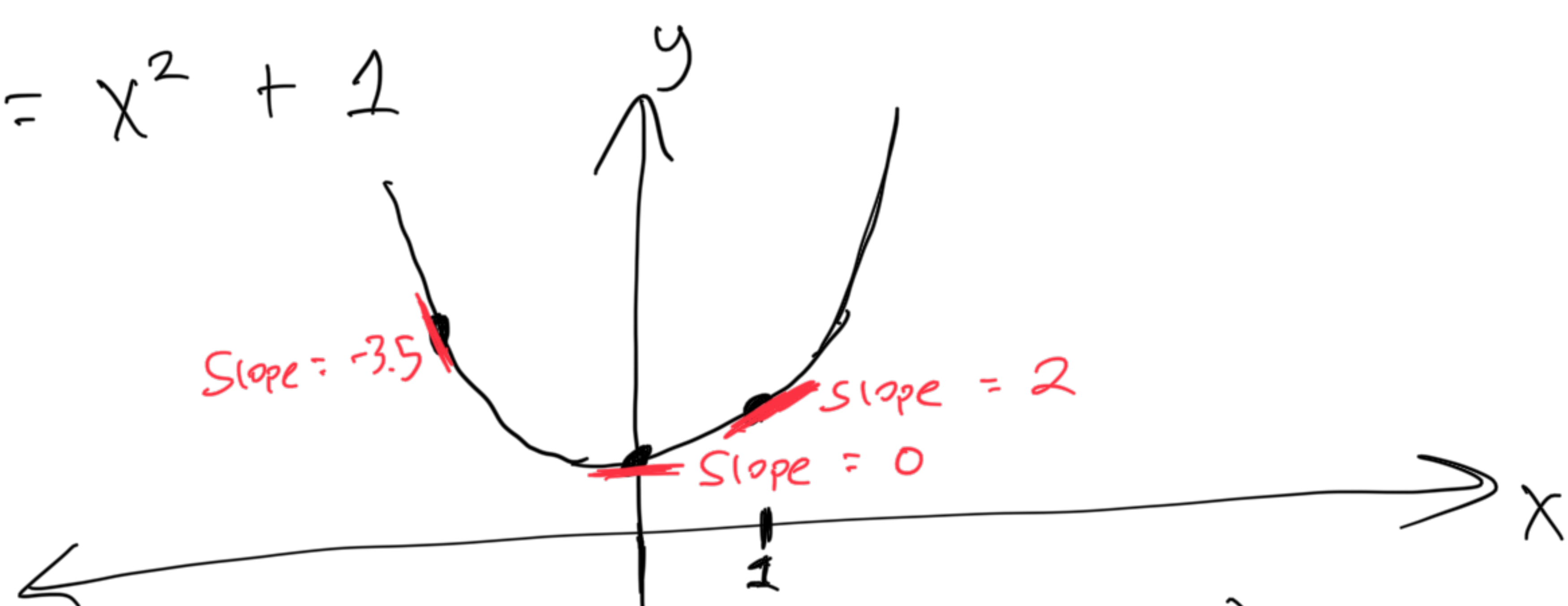
"K-Fold Cross-Validation"

\# of folds

To evaluate the model: average the performance over the K folds

Calculus Review

$y = x^2 + 1$



Derivative $= y' = \dfrac{dy}{dx} = \dfrac{d(x^2+1)}{dx}$

$\longrightarrow$ instantaneous rate of change of y with respect to x

$\dfrac{d(x^2+1)}{dx} = 2x$ (see calculus class

dX for derivative rules)

Partial Derivative

$$f(x_1, x_2, x_3) = 3x_1^5 - 12x_1x_2 + 2x_3 - 5$$

$$\frac{\partial f}{\partial x_1} = 15x_1^4 - 12x_2$$

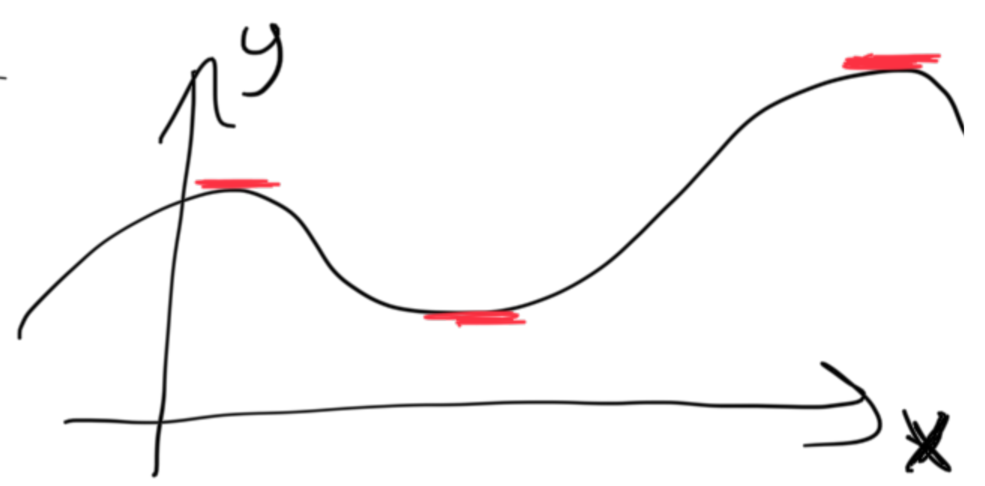
$$\frac{\partial f}{\partial x_2} = -12x_1$$

$$\frac{\partial f}{\partial x_3} = 2$$

$$\left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3}\right] = \text{Gradient}$$

(as in "gradient descent")

Optimizing functions



Procedure is to

Set derivative $= 0$

Solve for X

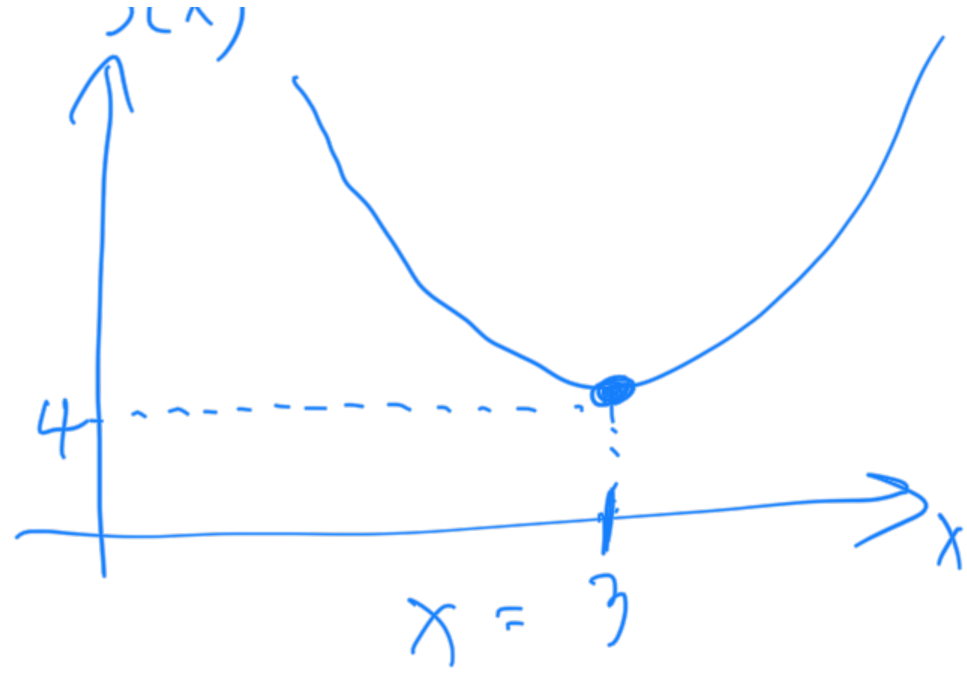
Example: $f(x) = (x-3)^2 + 4$

$$\frac{df}{dX} = 2(X-3)$$

$$\boxed{\text{deriv}(x^n) = nx^{n-1}}$$

$f(x)$

$$2(x-3) = 0$$
$$x - 3 = 0$$
$$x = 3$$



In ML;