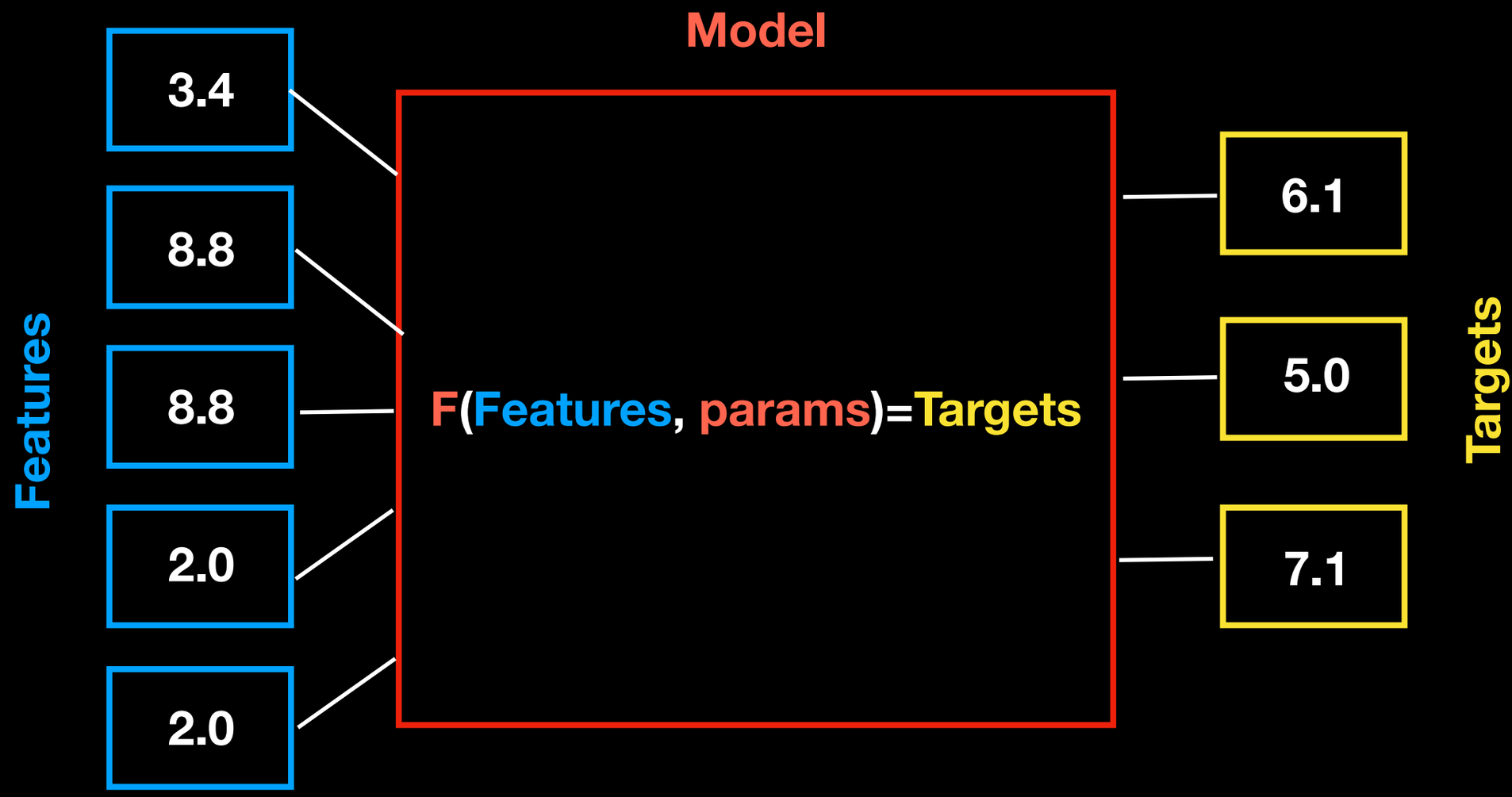
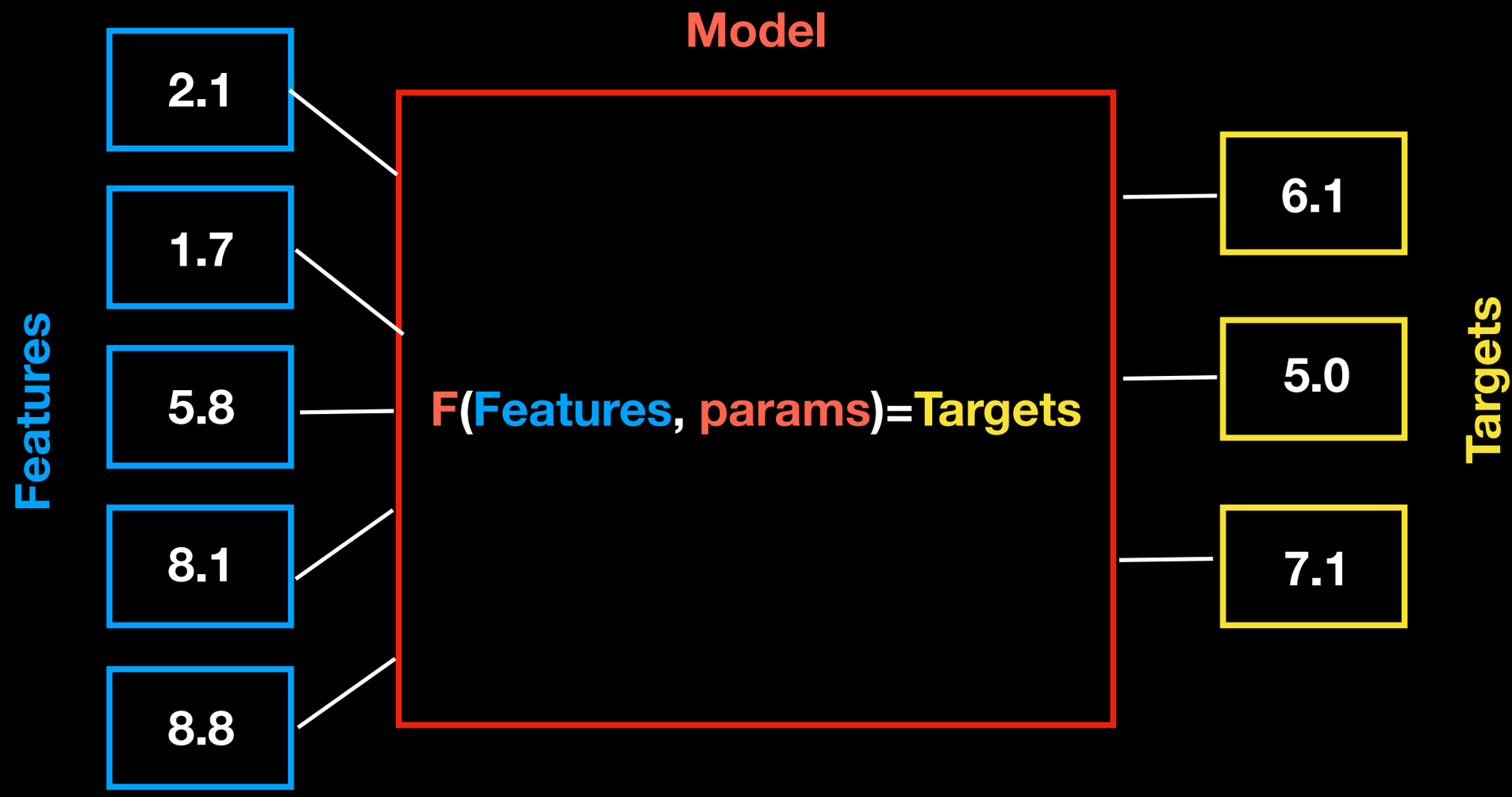
Regression Problem


Model
3.4
8.8
2.0
2.0
Features
F(Features, params)=Targets
6.1
5.0
7.1
Targets


Model
2.1
1.7
5.8
8.1
8.8
F(Features, params)=Targets
6.1
5.0
7.1
Features
Targets

Loss Function

F(Features, params)=Targets

DistanceFunction(RealTarget, EstimatedTarget)

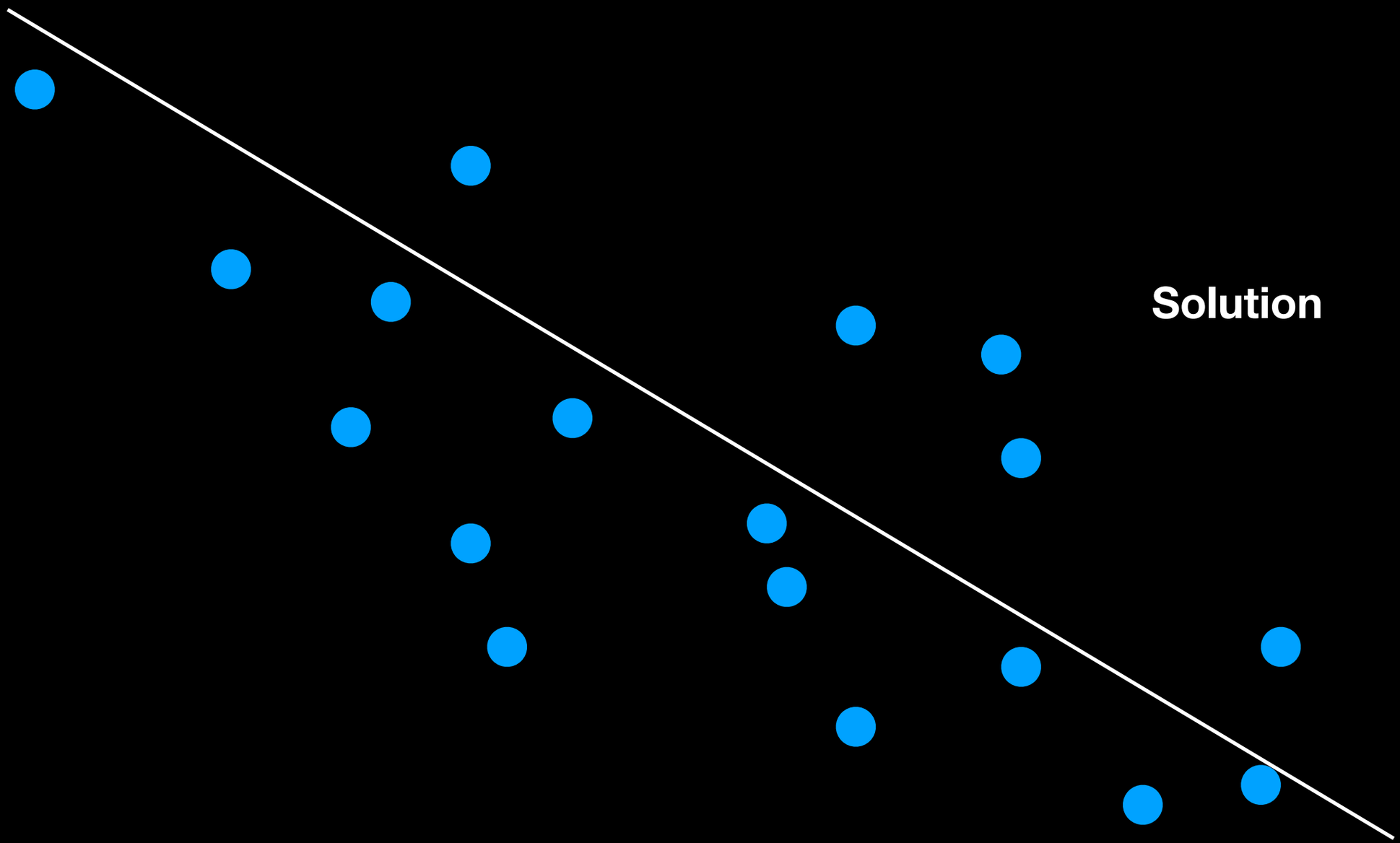
Min(Σ(DistanceFunction(RealTarget[i],EstimatedTarget[i]))

EstimatedTarget[i]=F(Features[i],params)

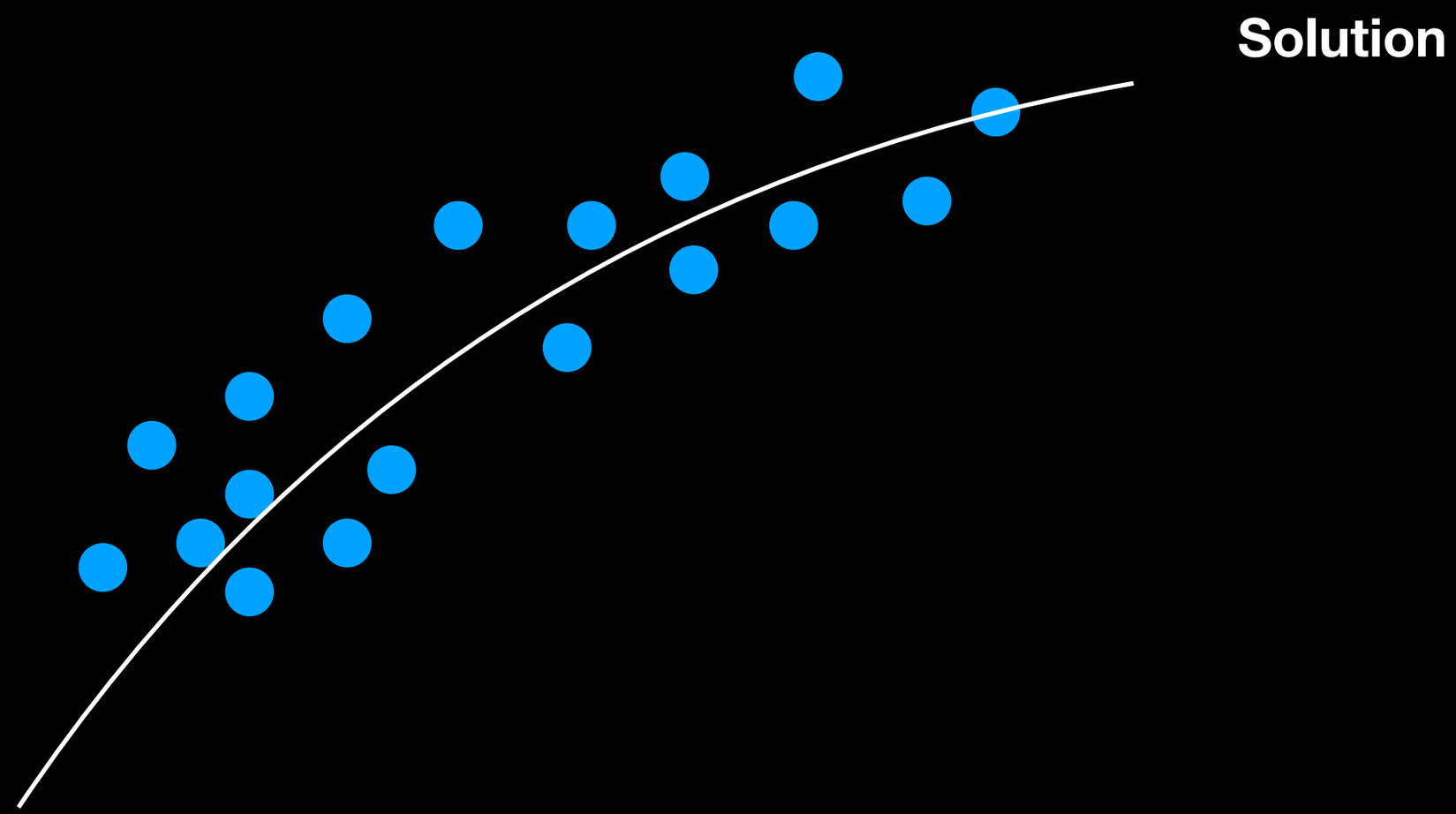
Min(Σ(DistanceFunction(RealTarget[i],F(Features[i],params))

=> use params of F to find optimal Targets
that are very close to real targets

Example:
Linear Regression



Example:
Polynomal Fit



Regularisation

Linear Regression

Feature Vector

$Ax=b$

Matrice

Target Vector

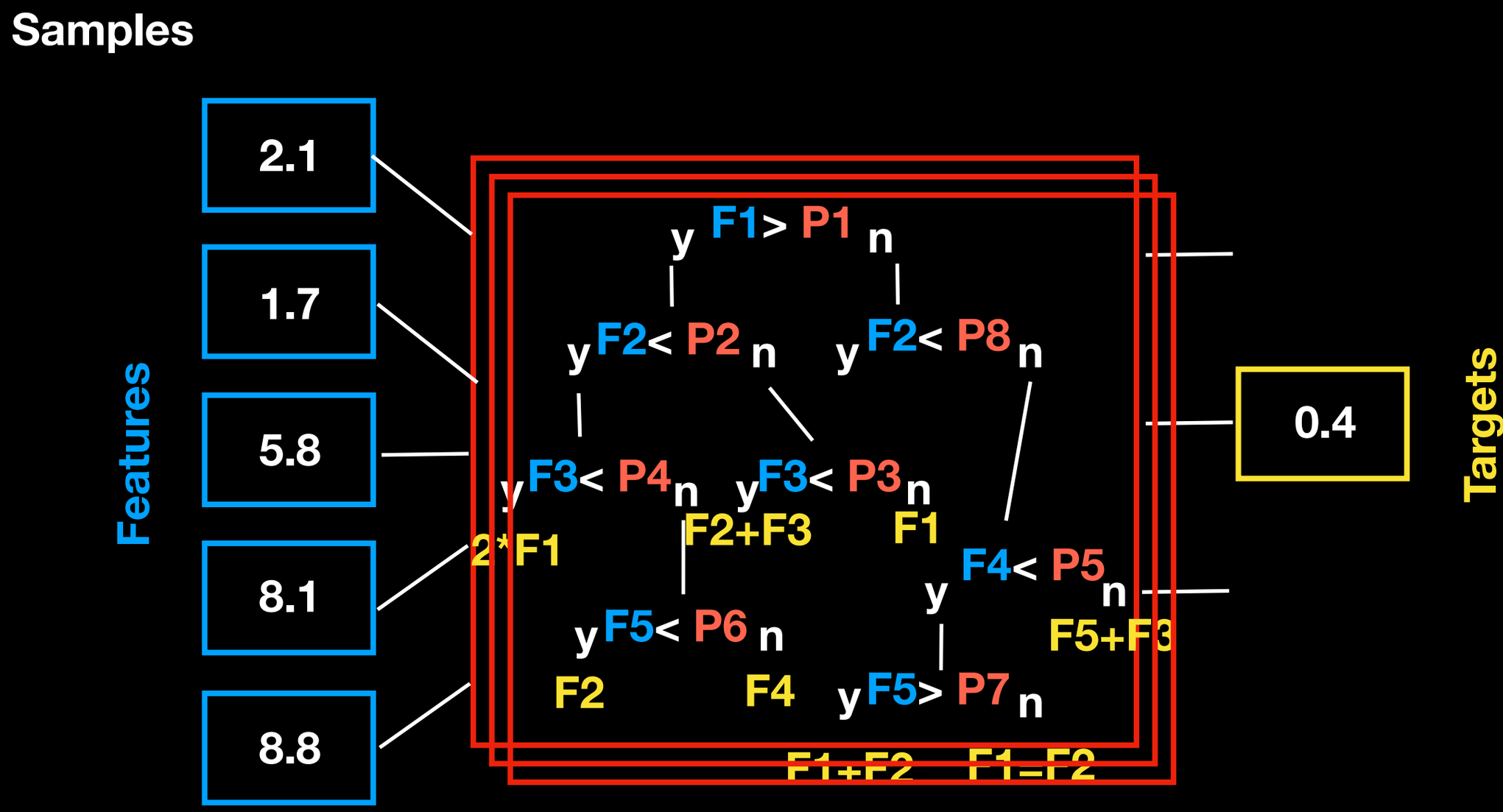
$\Sigma||Ax[i]-b[i]||$

$\Sigma(||Ax[i]-b[i]||+|Lx[i]|+||Lx[i]||)$

Regaluarisation

Ridge $\lambda||x[i]||$ / Lasso $\lambda|x[i]|$ / Elastic Net $\lambda1||x[i]||+\lambda2|x[i]|$
-> feature selection

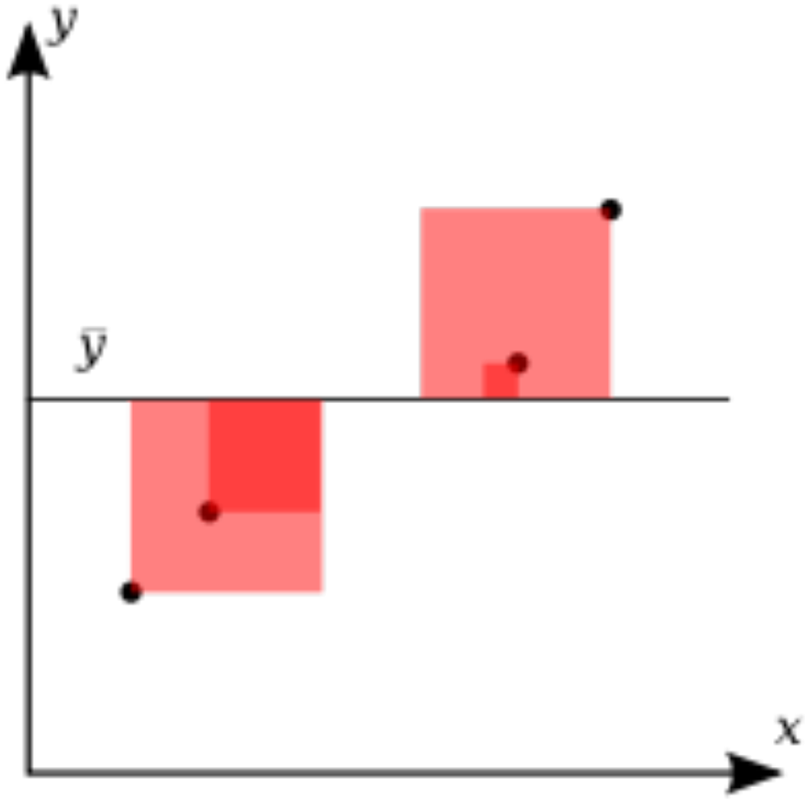
Example:
(Random) Forest

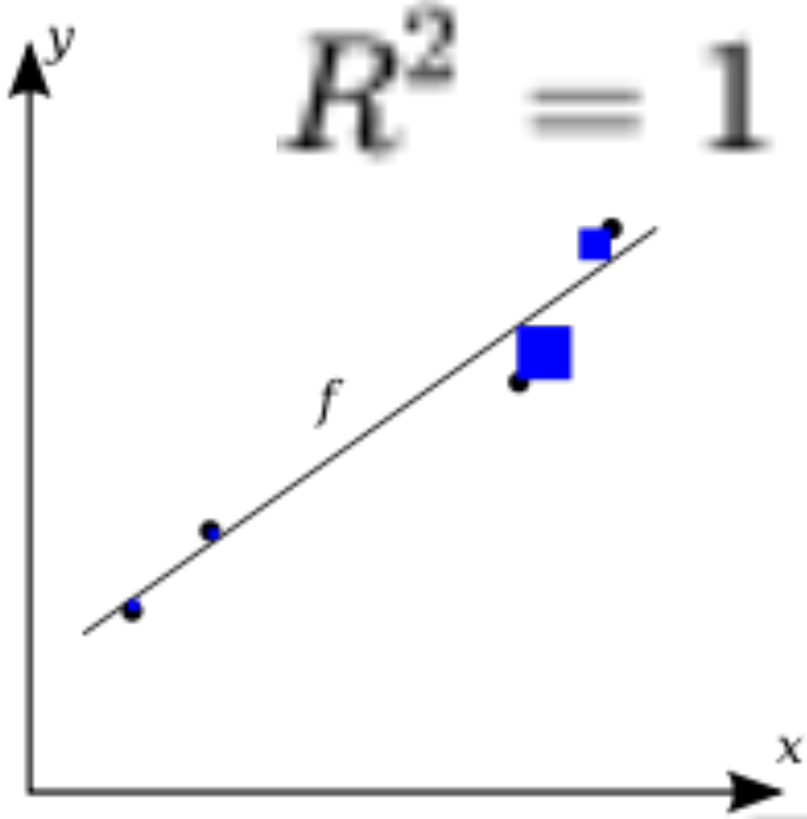


Finding optimal parameters PN

+ Decision parameter which tree wins

R2 Score





$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2, \quad SS_{\text{reg}} = \sum_i (f_i - \bar{y})^2,$$

Model Validation

n-fold cross validation

Test On \ Train On	1	2	3	4
1	0.9	0.7	0.5	0.9
2	0.5	0.8	0.6	0.6
3	0.6	0.6	0.9	0.7
4	0.7	0.2	0.5	0.9

yields a confidence SCORE +- error
->How often will I hit the right target

Final Model Validation only with Testset

if possible cross validated

cross validation datasets can overlap and be chosen randomly

but know what you are doing!