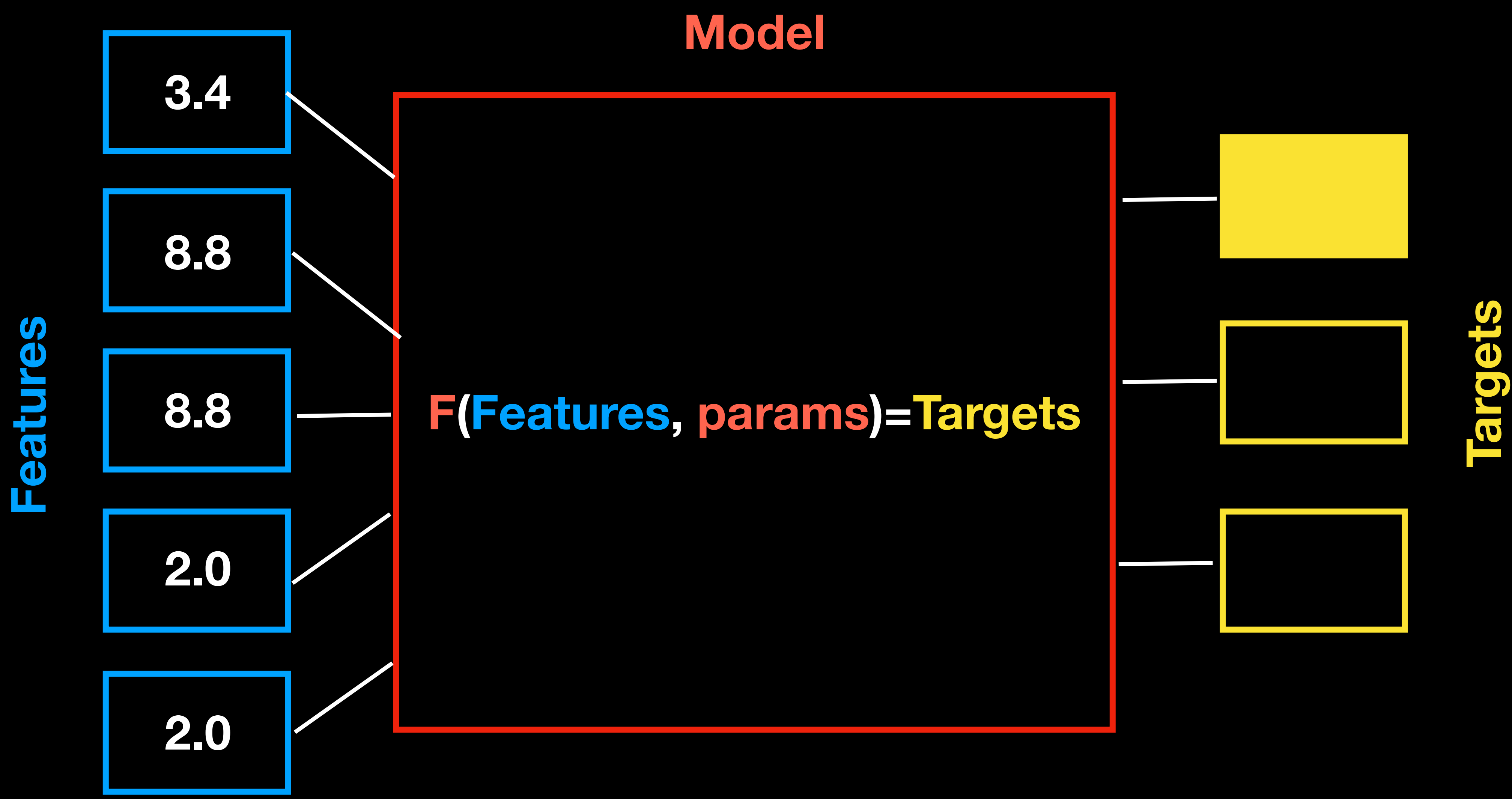
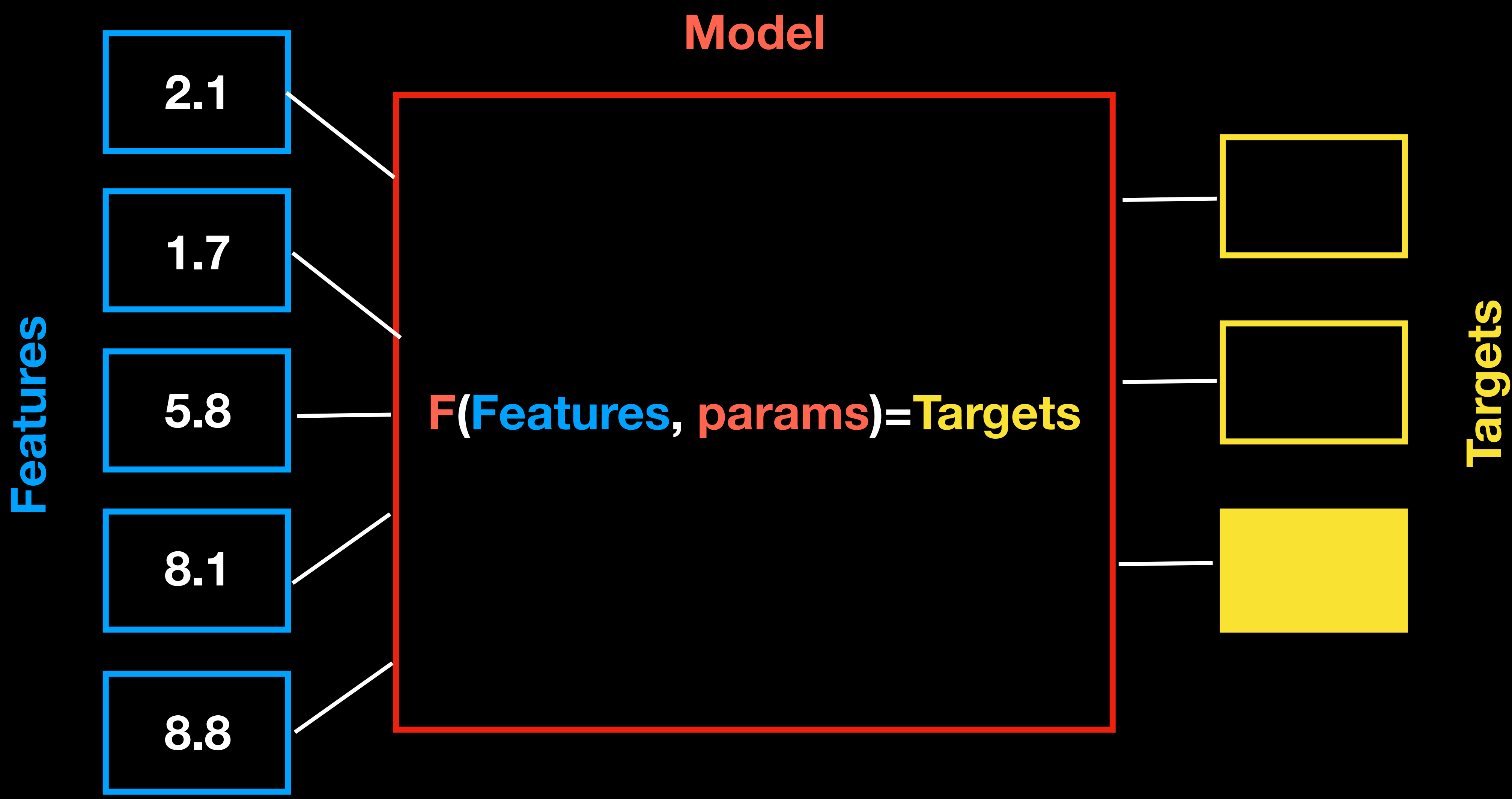
Classification Problem


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


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

Loss Function

F(Features, params)=Targets

DistanceFunction(RealTarget, EstimatedTarget)

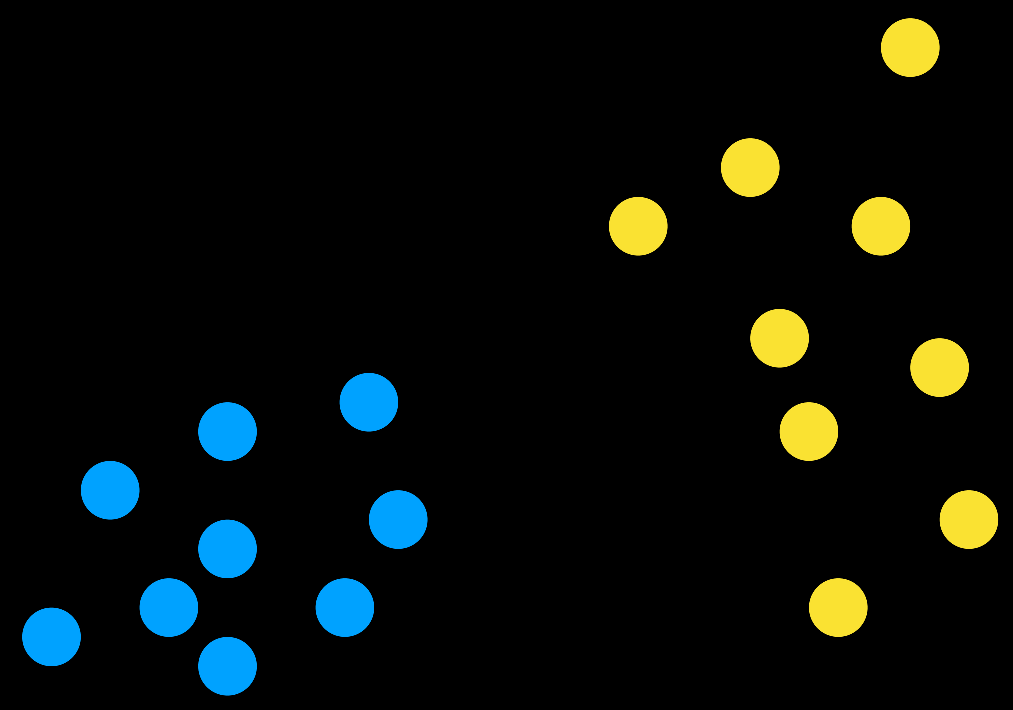
Min(Σ(DistanceFunction(RealTarget[i],EstimatedTarget[i]))

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

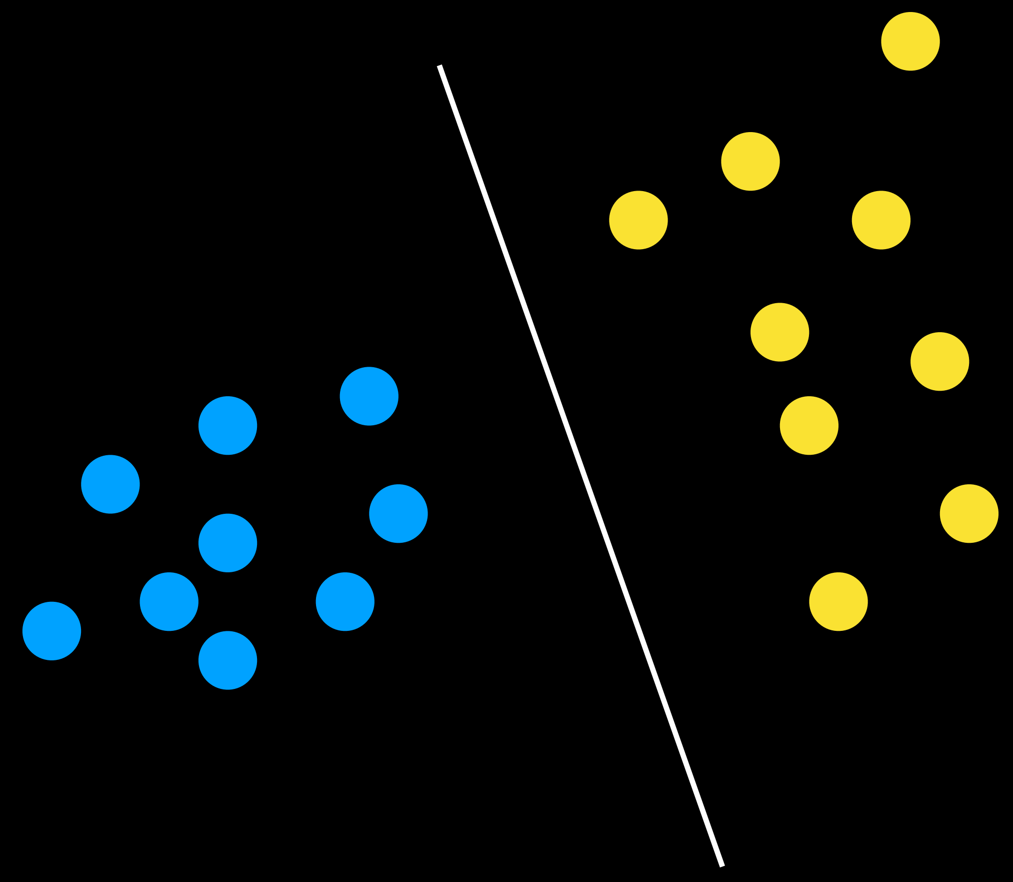
Min(Σ(DistanceFunction(RealTarget[i],F(Features[i],params))

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

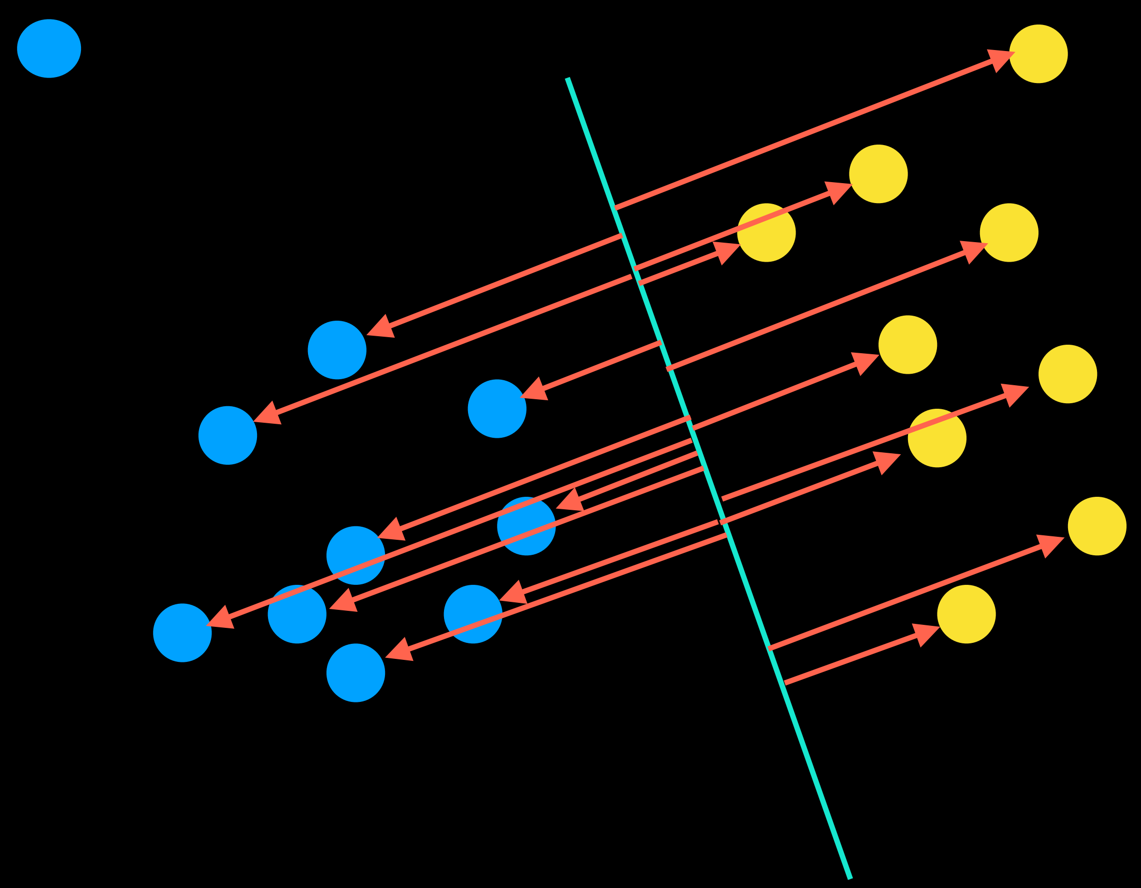
Example:
Support Vector Machine



Example:
Support Vector Machine



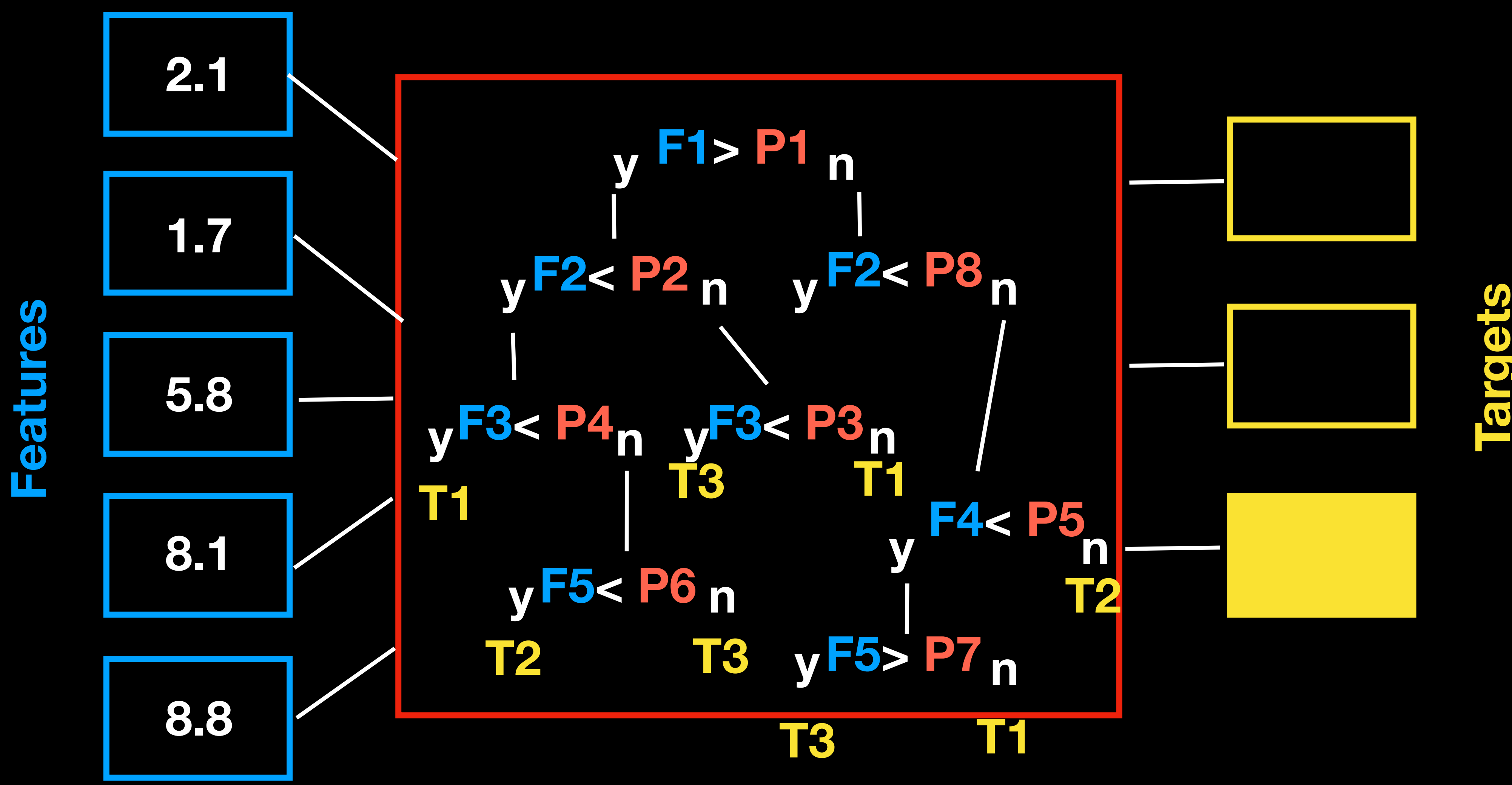
Example: Support Vector Machine



Find Plane so that support vectors are the largest and hence the dataset is most separated

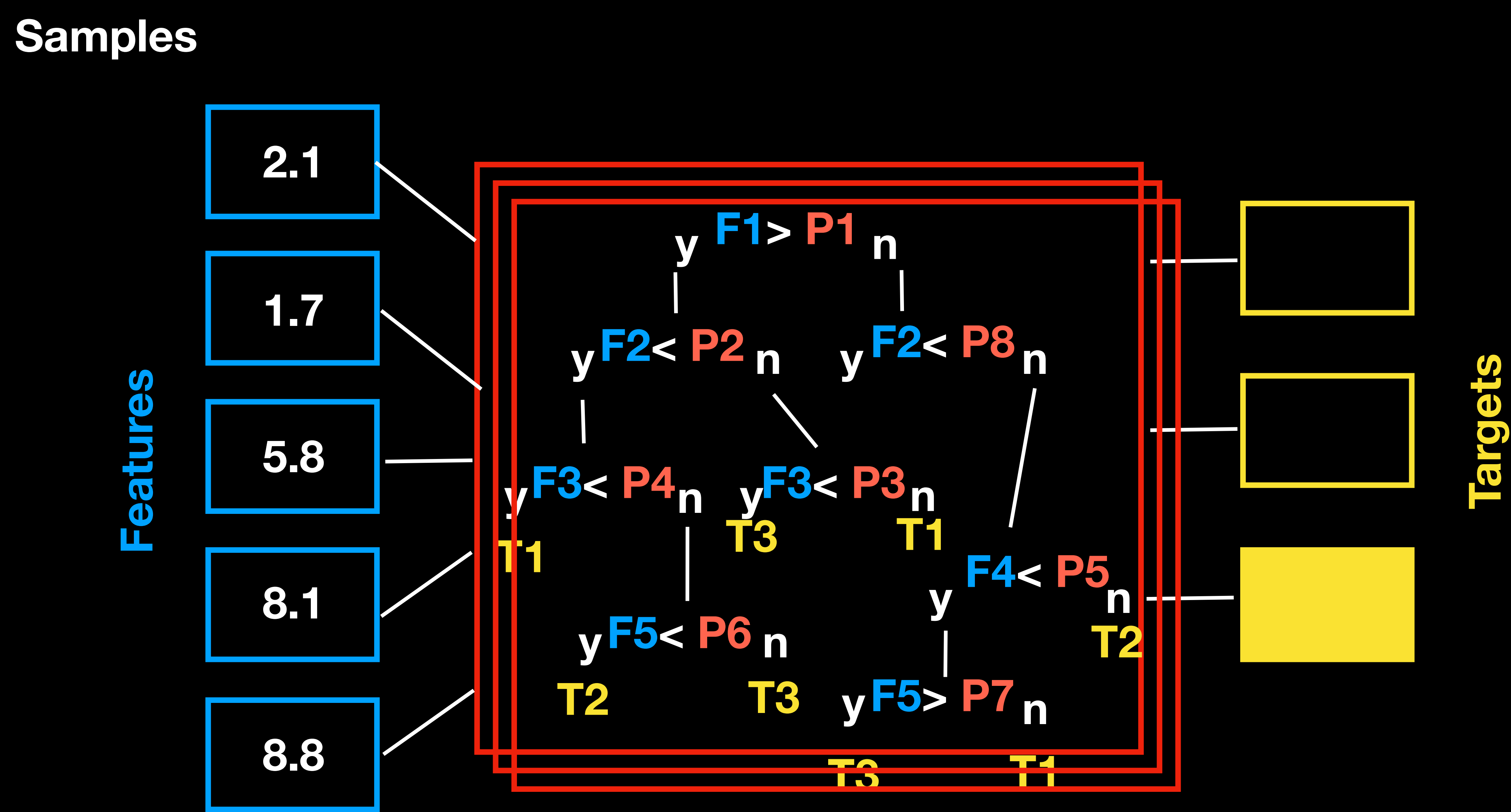
DEMO: https://www.csie.ntu.edu.tw/~cjlin/libsvm/

Example: Decision Tree



Finding optimal parameters PN

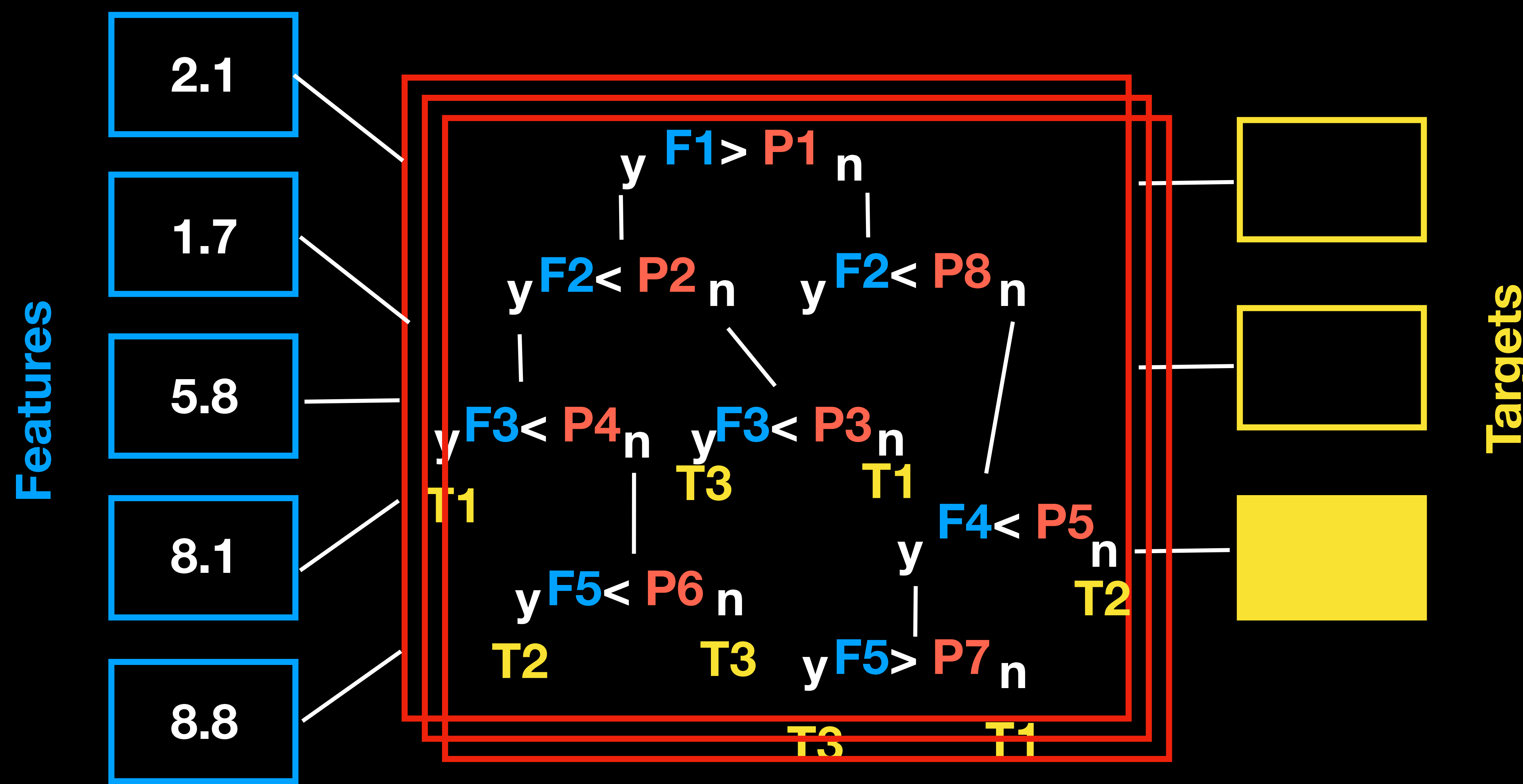
Example: (Random) Forest



Finding optimal parameters PN

+ Decision parameter which tree wins (i.e. majority voting)

Example: (Random) Forest



Finding optimal parameters PN

+ Decision parameter which forest wins (i.e. majority voting)

Model Validation

Dataset

Trainingset | Testset

1 | 2 | 3 | 4

n-fold cross validation

Train On

Test 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 probability +- 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!