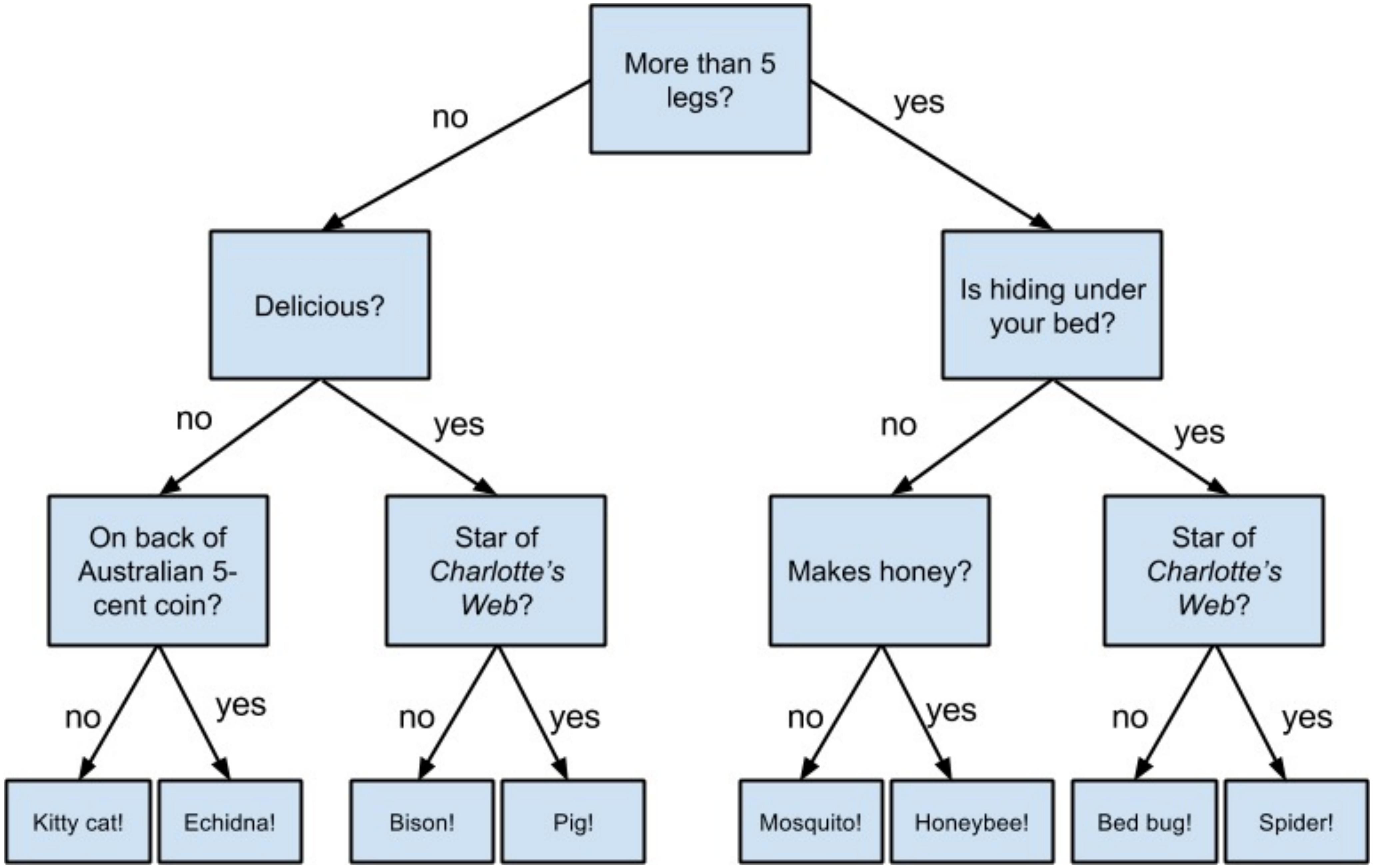
machine learning technique to build a model

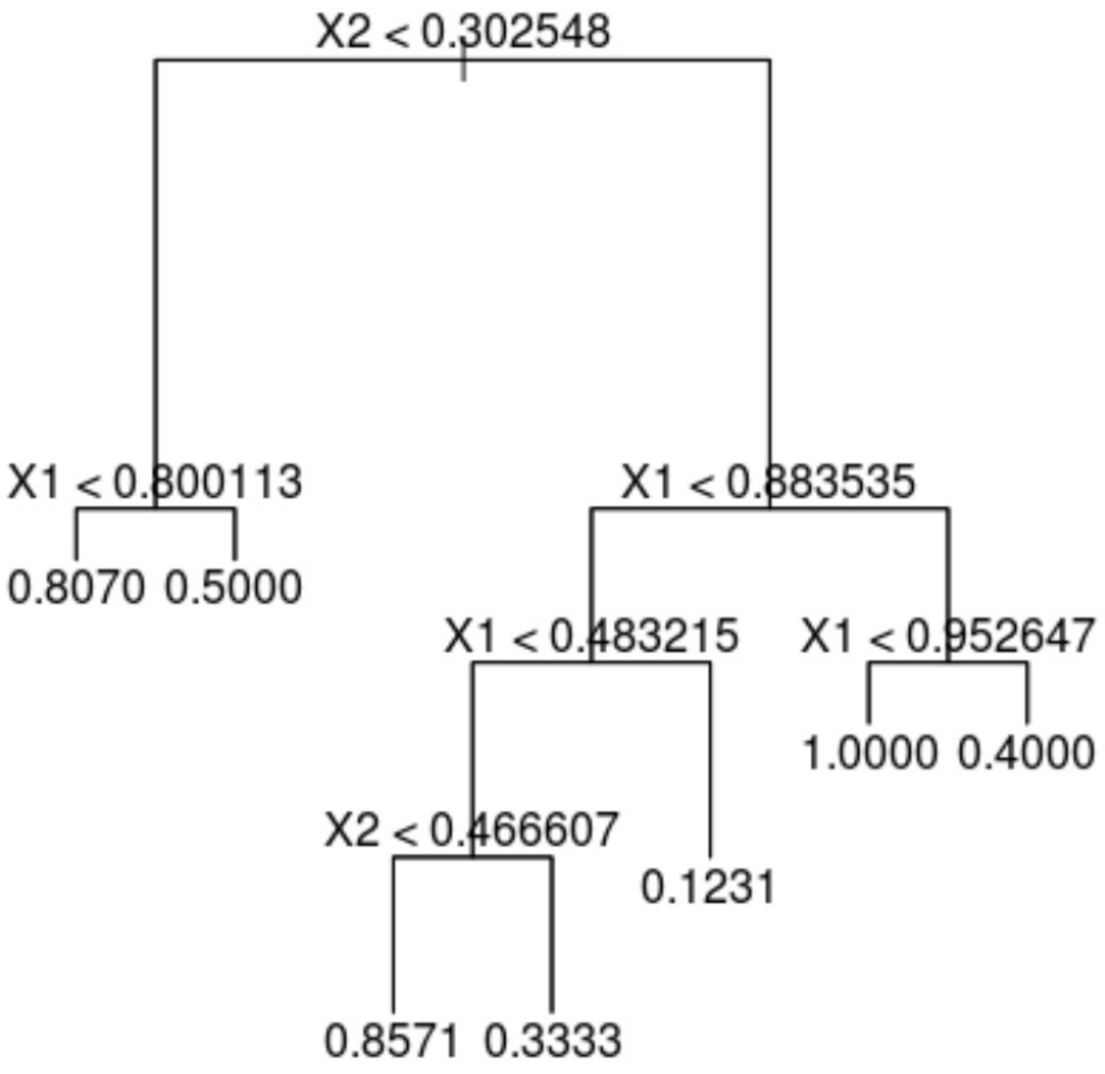
Decision tree based predictive modeling (continued)

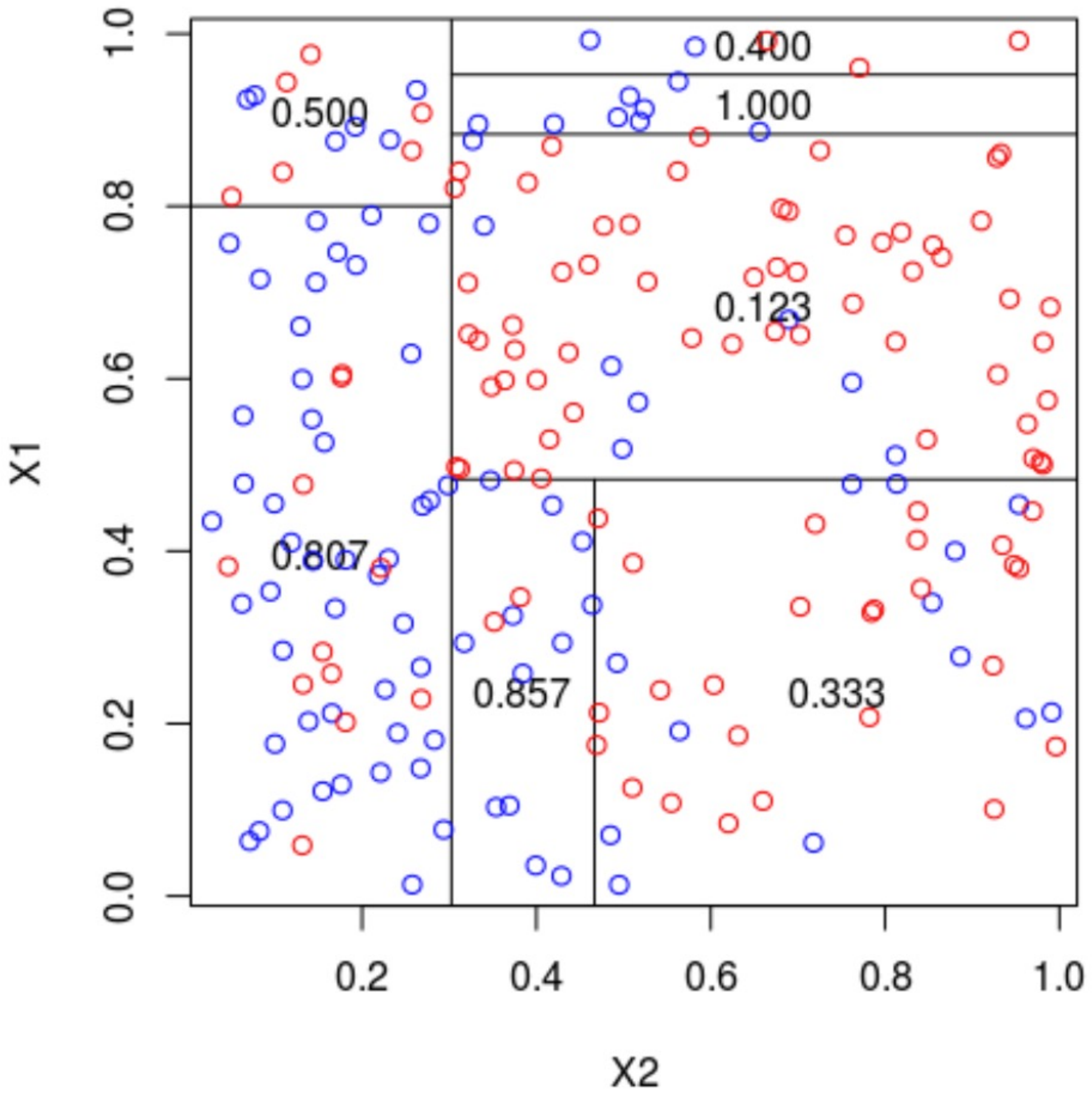
decision trees and random forests

decision tree



decision tree





how to grow a decision tree

- pick the split (variable and cutoff) that best separates the data at current node

- for classification, best is typically determined by gini impurity

- stop growing tree when all nodes are “pure” (all one label) or node contains a low number of data points

bootstrap aggregation (bagging)

- resample (with replacement) a dataset of the same size as original data

- repeat, compute a ***model*** on each sampled dataset

- ***aggregate*** the model over the sampled datasets to reduce overfitting or quantify uncertainty

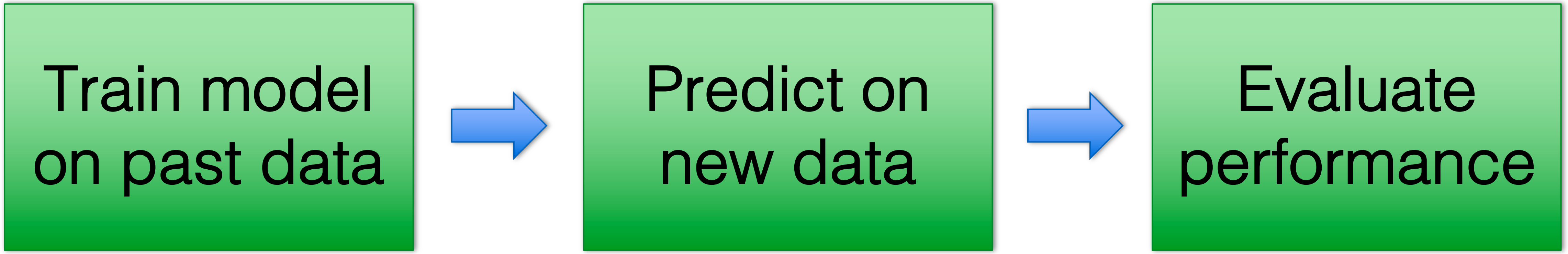
random forests:
bagging decision trees

- a single decision tree fully grown overfits the data (the prediction will be perfect on training data)

- random forests grow many trees over randomly sampled subsets of the data and then aggregate (average or majority vote)

- to reduce variance further, choose splits at each node from a random subset of predictor variables

Machine Learning



Train model on past data

Predict on new data

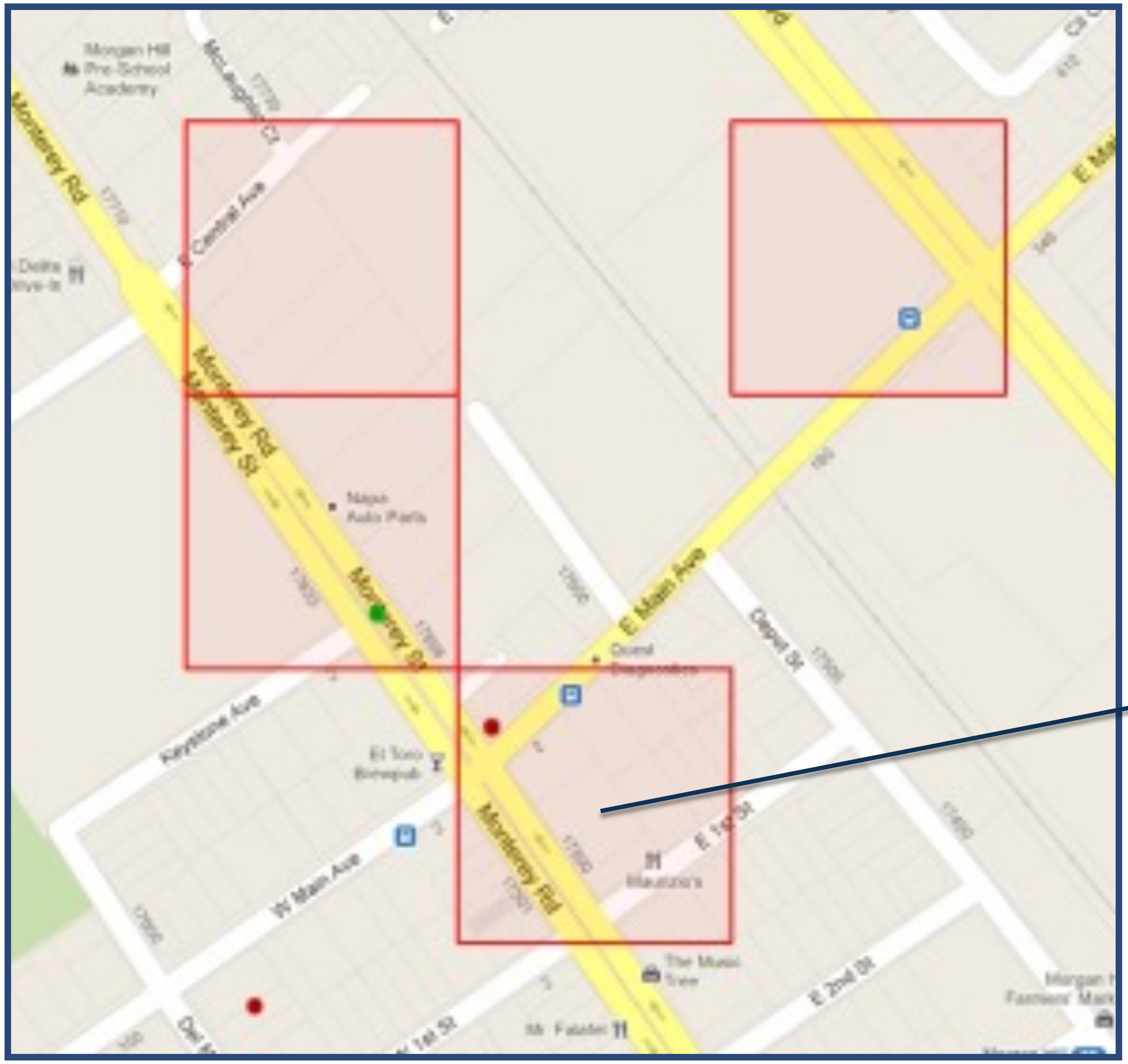
Evaluate performance

$\tilde{L}(y, f(x))$

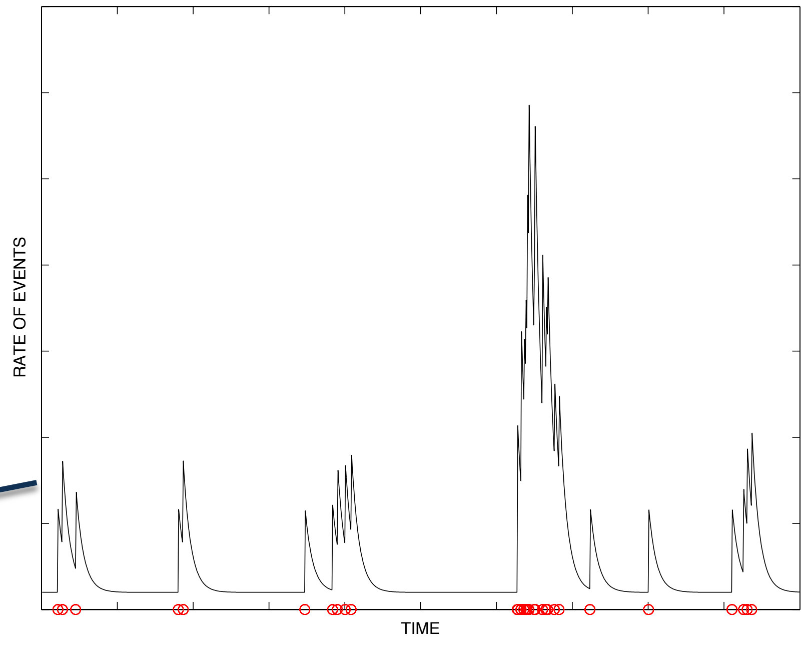
$y = f(x)$

$L(y, f(x))$

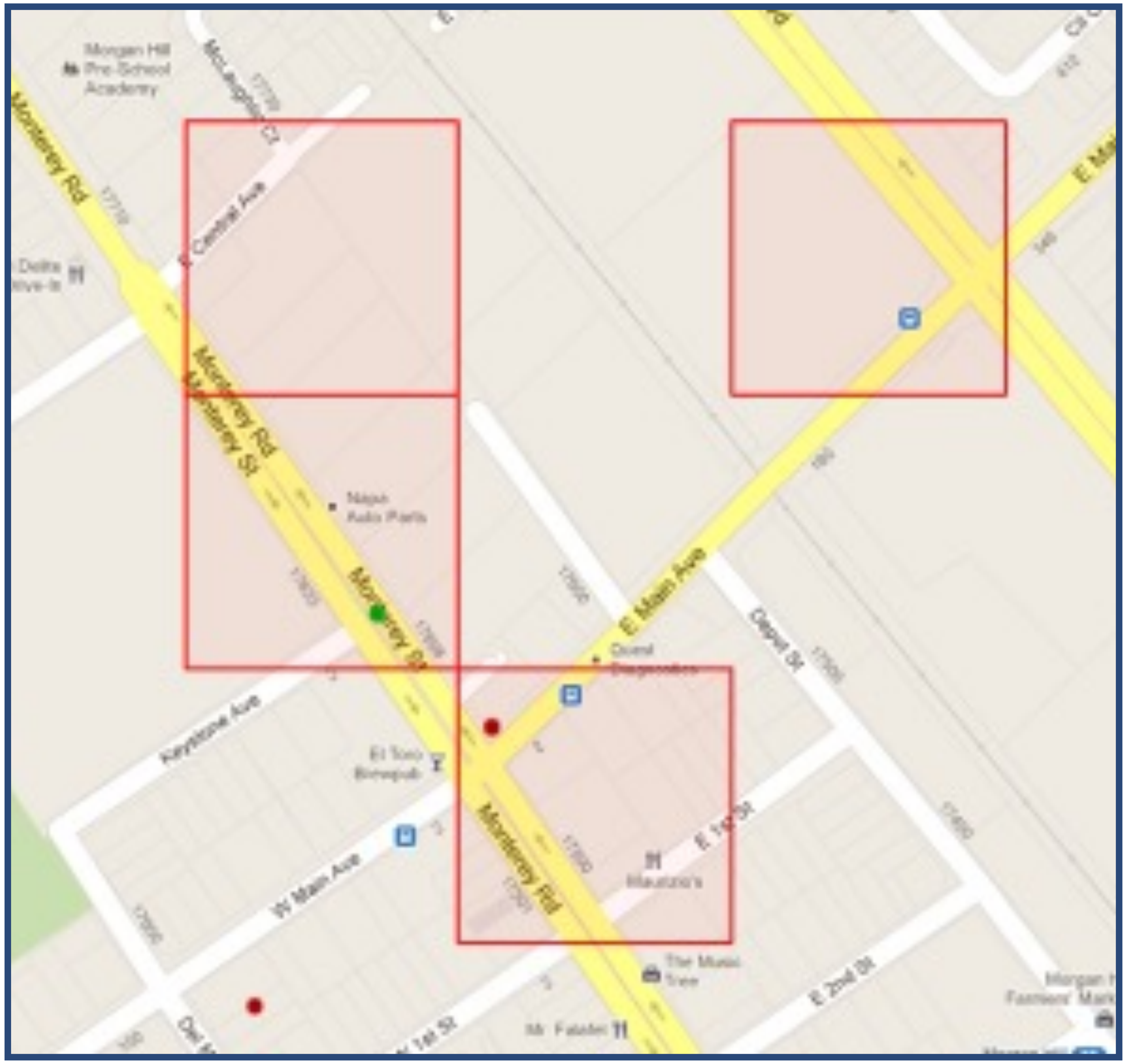
Crime hotspot prediction



$$\lambda(t) = \mu + \sum_{t>t_k} g(t - t_k)$$



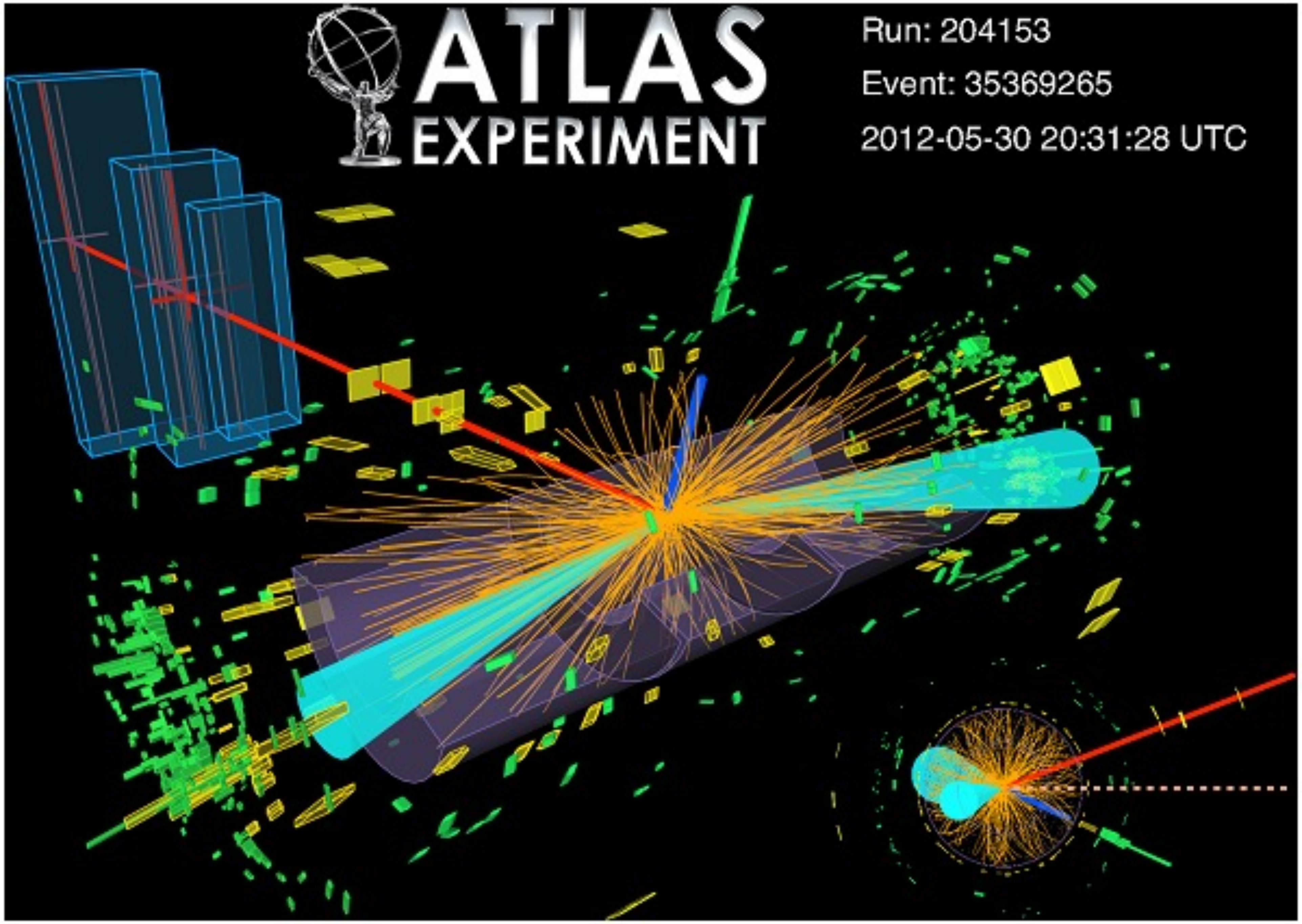
Crime hotspot prediction



Predictive Accuracy Index:

% of crime predicted in top k ranked hotspots

Higgs Boson Detection



Approximate Median Significance

$$\sqrt{2((s+b+10)\log(1+s/(b+10))-s)}$$

s, number of true positives

b, number false positives

Boosting: stagewise additive modeling with "weak" learners

$$\min_{f} \sum_{i=1}^{N} L(y_i, f(x_i))$$

$$f(x) = f_1(x) + f_2(x) + \ldots$$

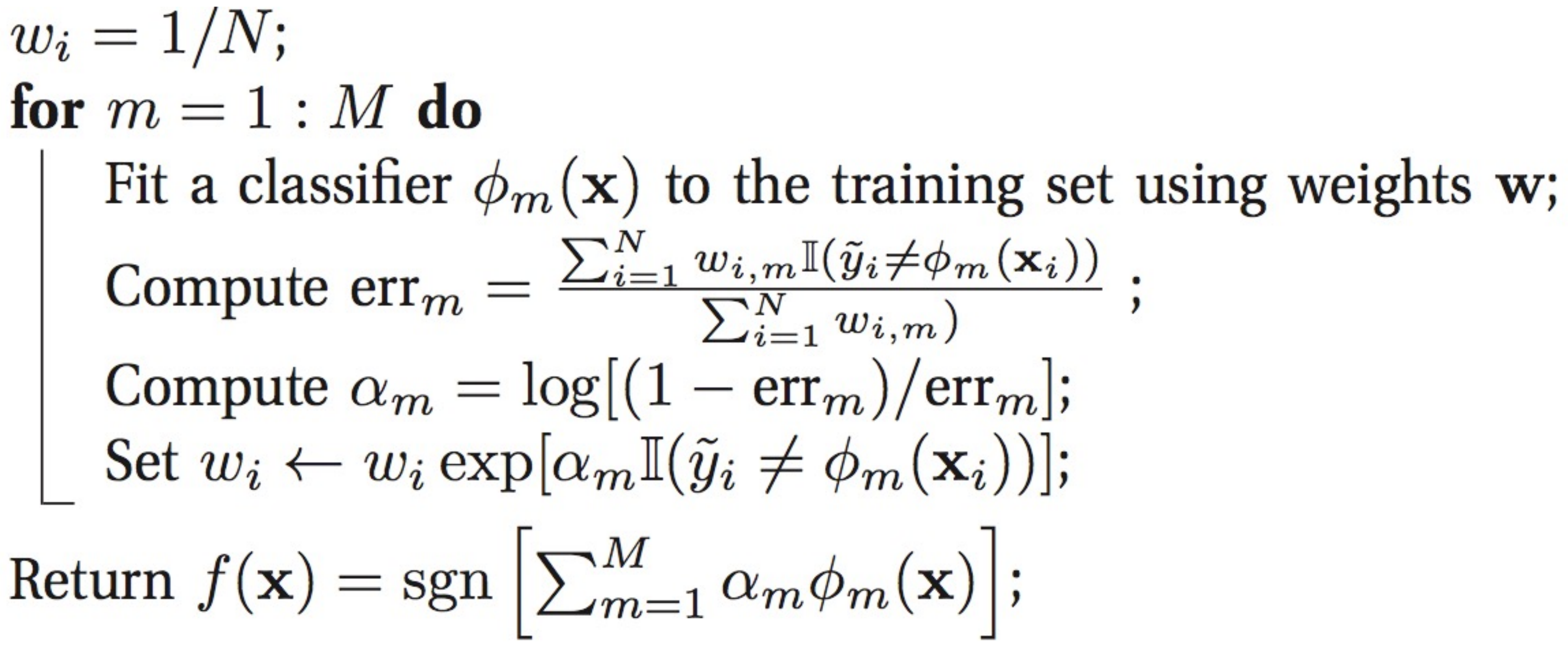
Boosting: stagewise additive modeling with “weak” learners

$$\min_{\beta,\gamma} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + \beta\phi(x_i;\gamma))$$

$$f_m(x) = f_{m-1}(x) + \beta\phi(x;\gamma)$$

Adaboost

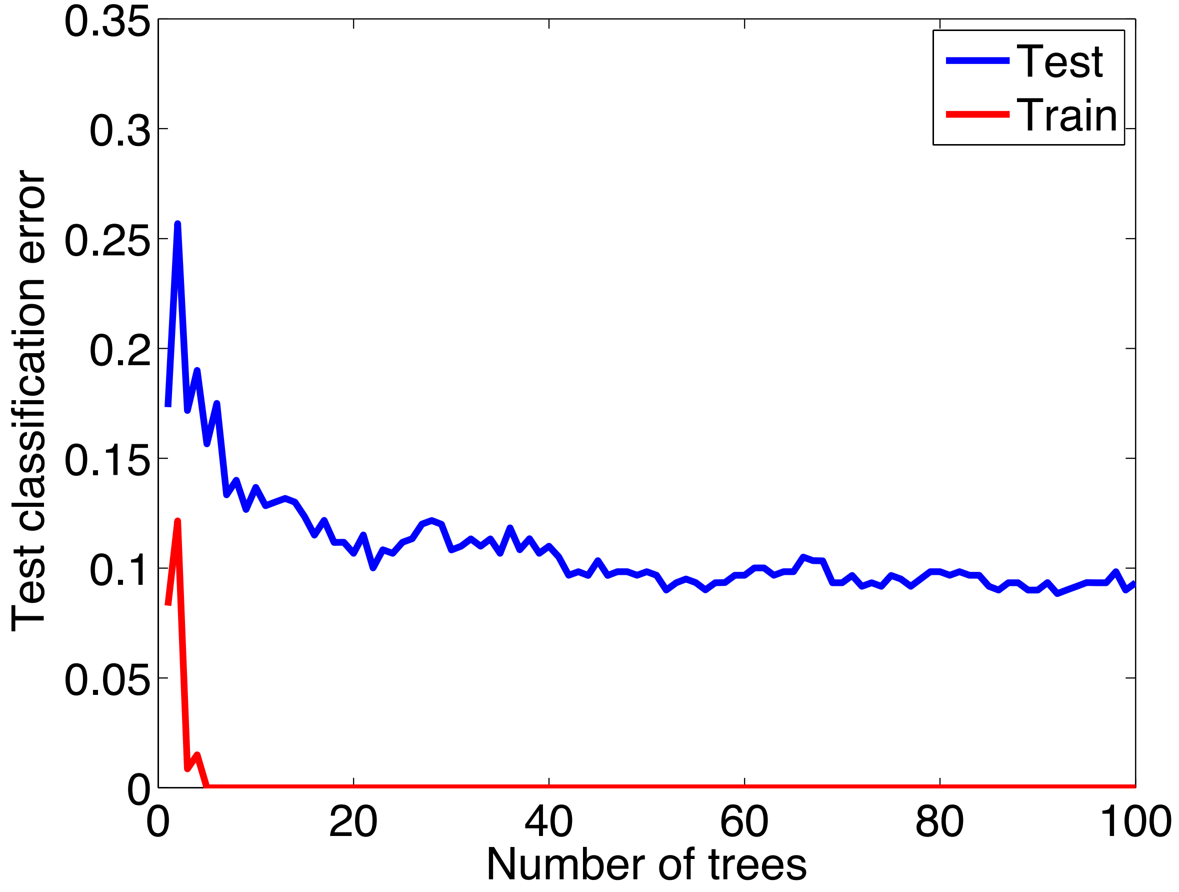
(Freund & Shapire)



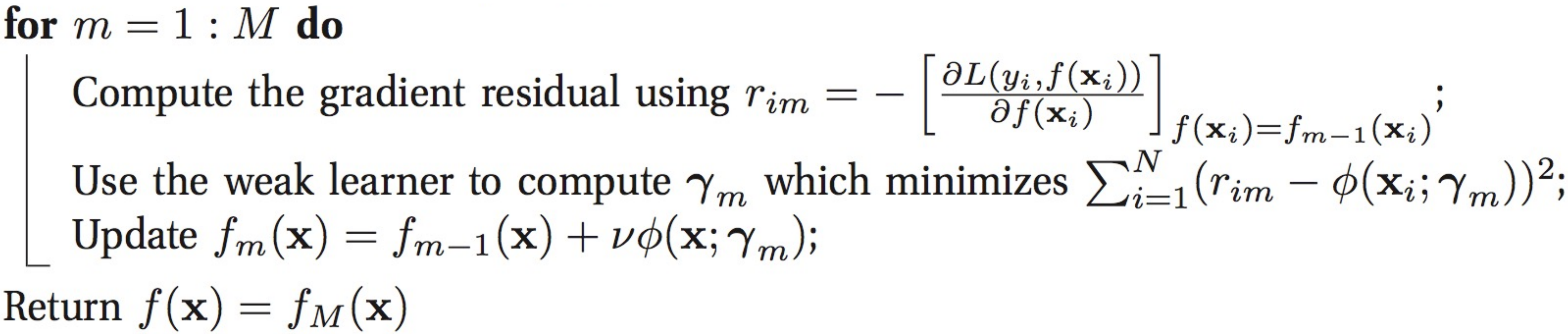
```
$w_i = 1/N$;
for $m = 1 : M$ do
    Fit a classifier $\phi_m(\mathbf{x})$ to the training set using weights $\mathbf{w}$;
    Compute $\mathrm{err}_m = \frac{\sum_{i=1}^{N} w_{i,m} \mathbb{I}(\tilde{y}_i \neq \phi_m(\mathbf{x}_i))}{\sum_{i=1}^{N} w_{i,m})}$;
    Compute $\alpha_m = \log[(1 - \mathrm{err}_m)/\mathrm{err}_m]$;
    Set $w_i \leftarrow w_i \exp[\alpha_m \mathbb{I}(\tilde{y}_i \neq \phi_m(\mathbf{x}_i))]$;
Return $f(\mathbf{x}) = \mathrm{sgn}\left[\sum_{m=1}^{M} \alpha_m \phi_m(\mathbf{x})\right]$;
```

Adaboost

- Adaboost maximizes margin on training data
- Sensitive to mislabeled data
- Does not output probability
- Logitboost is alternative

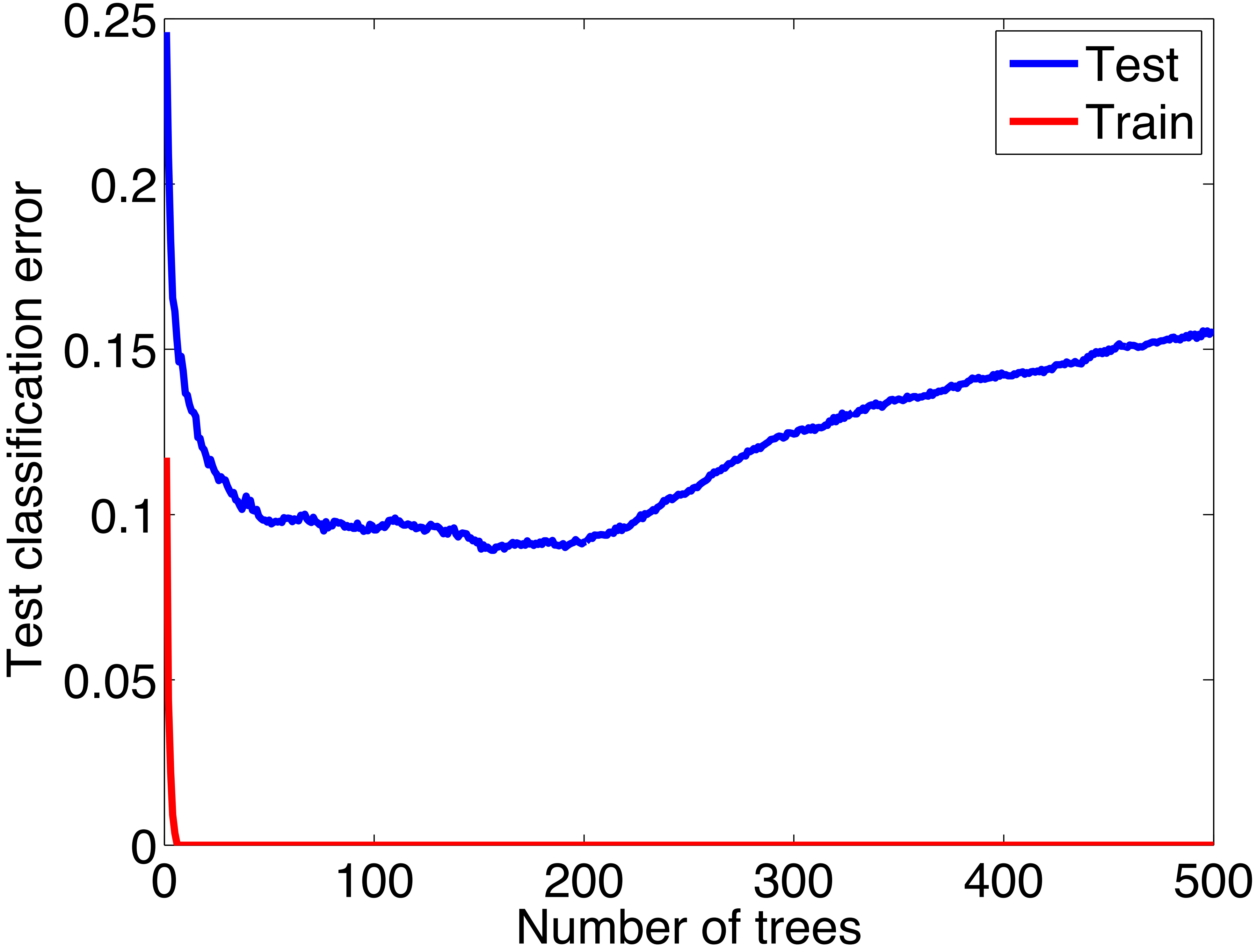


Gradient boosting



```
for $m = 1 : M$ do
    Compute the gradient residual using $r_{im} = -\left[\frac{\partial L(y_i, f(\mathbf{x}_i))}{\partial f(\mathbf{x}_i)}\right]_{f(\mathbf{x}_i)=f_{m-1}(\mathbf{x}_i)}$;
    Use the weak learner to compute $\boldsymbol{\gamma}_m$ which minimizes $\sum_{i=1}^{N}(r_{im} - \phi(\mathbf{x}_i; \boldsymbol{\gamma}_m))^2$;
    Update $f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + \nu\phi(\mathbf{x}; \boldsymbol{\gamma}_m)$;
Return $f(\mathbf{x}) = f_M(\mathbf{x})$
```

Boosting in practice



Boosting in practice

- Cross-validation: monitor error on held out data
- Reduce learning rate Δt (more iterations needed)
- Restrict tree depth
- “Stochastic gradient boosting”: subsample training data at each iteration
- Loss function itself is a “parameter”

Advantages of boosting

- Weak learners are efficient, easy to train
- Performs variable selection w/out forward selection or backward elimination
- Allows for loss function to be optimized with decision trees

Disadvantages of boosting

- Sequential: unlike random forests, estimating weak learners can’t be done in parallel

- More parameters to tune compared to random forests (usually done by grid search)

- Similar to random forests in that they (often) don’t “learn features”

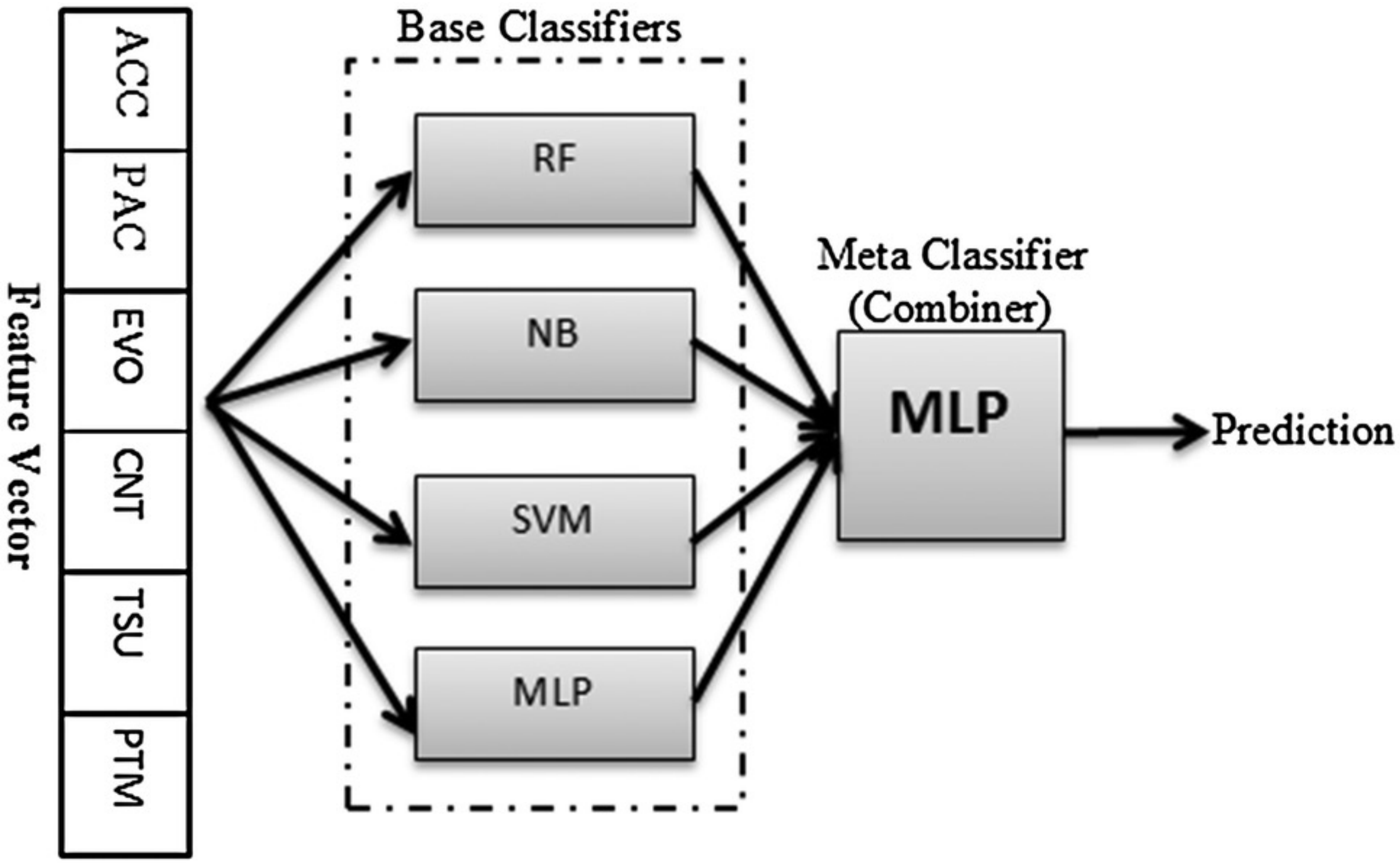
Boosting software

- gbm
- xgboost
- LightGBM
- catboost

ensemble models

- no model is perfect
- why not combine models to improve accuracy
- another way of looking at it, let the output of different models be new features

stacking


Feature Vector
ACC
PAC
EVO
CNT
TSU
PTM
Base Classifiers
RF
NB
SVM
MLP
Meta Classifier
(Combiner)
MLP
Prediction

linear weighted stacking

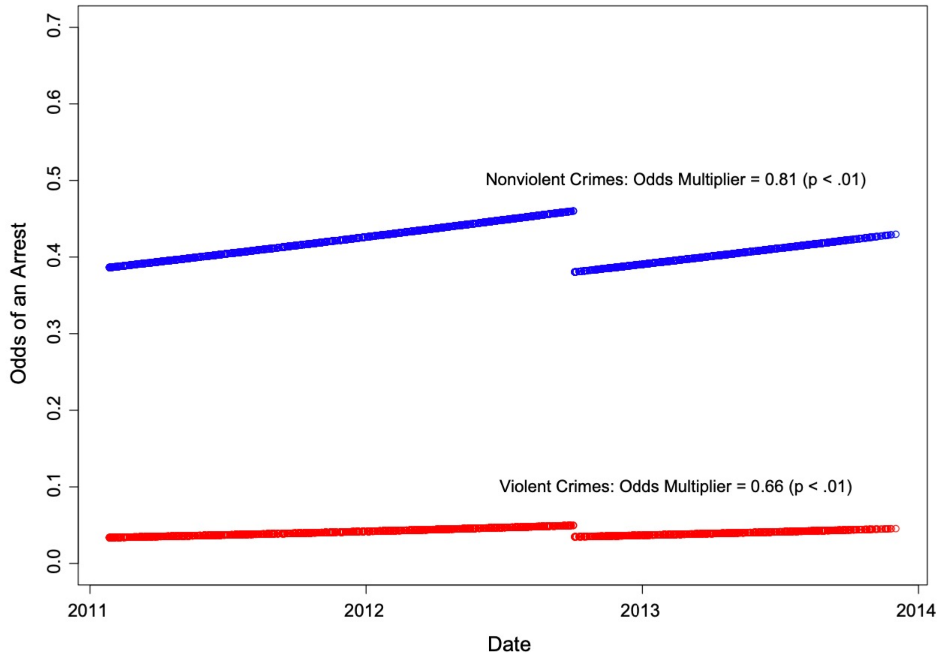
- divide training data into 2 parts, call them train1 and train2
- on train1, train rf, glm, gbm, svm
- score each model on train2 and use those scores as four new features
- combine four new features with original features on train2 and fit a glm model (with interactions between original and new features)
- run test data through the stacked model pipeline

Example: forecasting recidivism

- Risk score used in pre-trial bail and parole from prison decisions
- Features
  - Demographics: age, sex, race/ethnicity, education, employment
  - Record: prior arrests, convictions
  - Event-level: drug tests, probation reports
- Output is a probability of recidivism

Example: Pennsylvania Board of Probation and Parole

- Random forest used to assign risk of recidivism
- Quasi-experimental
- Change in distribution of parolees
- No change in avg rate of parole approval
- Estimated risk of recidivism was lower in treatment group

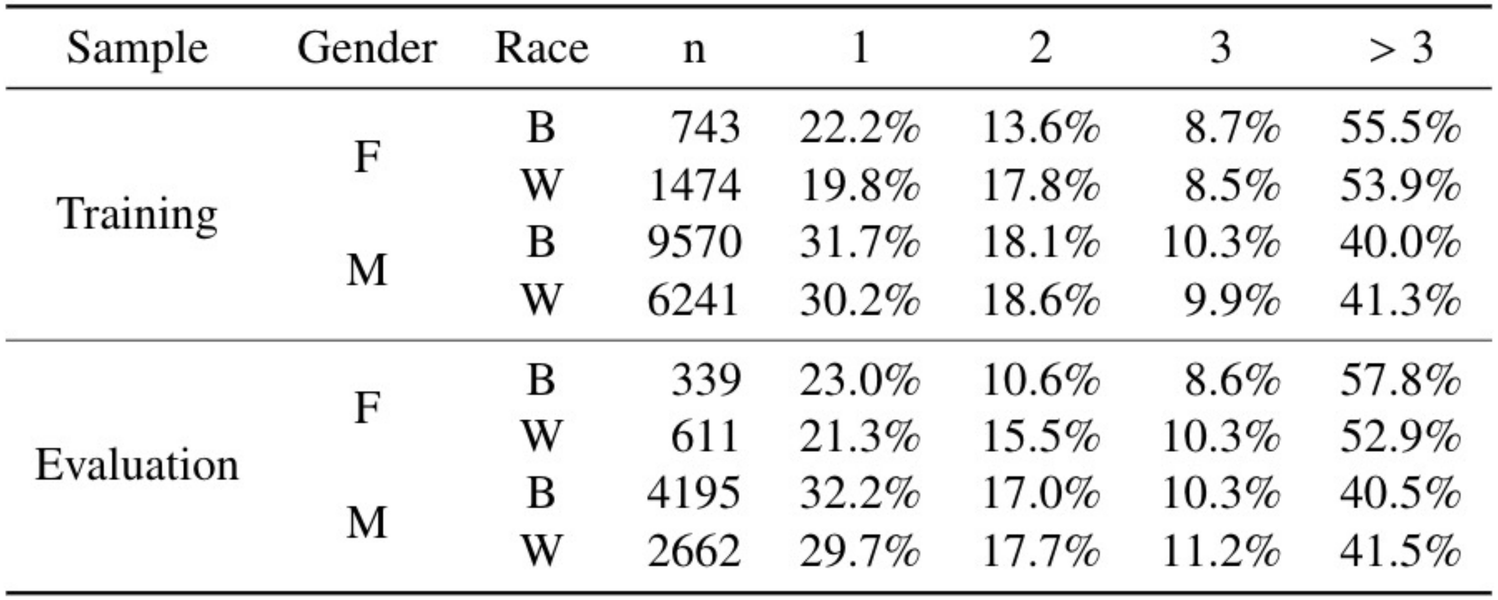


Berk, Richard. "An impact assessment of machine learning risk forecasts on parole board decisions and recidivism." *Journal of Experimental Criminology* 13.2 (2017): 193-216.

NIJ Recidivism Forecasting Challenge

$$\mathrm{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{p}_i)^2,$$

$$\mathrm{AF} = (1 - \mathrm{MSE})(1 - \Delta\mathrm{FPR})$$



Sample	Gender	Race	n	1	2	3	> 3
Training	F	B	743	22.2%	13.6%	8.7%	55.5%
		W	1474	19.8%	17.8%	8.5%	53.9%
	M	B	9570	31.7%	18.1%	10.3%	40.0%
		W	6241	30.2%	18.6%	9.9%	41.3%
Evaluation	F	B	339	23.0%	10.6%	8.6%	57.8%
		W	611	21.3%	15.5%	10.3%	52.9%
	M	B	4195	32.2%	17.0%	10.3%	40.5%
		W	2662	29.7%	17.7%	11.2%	41.5%

Modeling approach

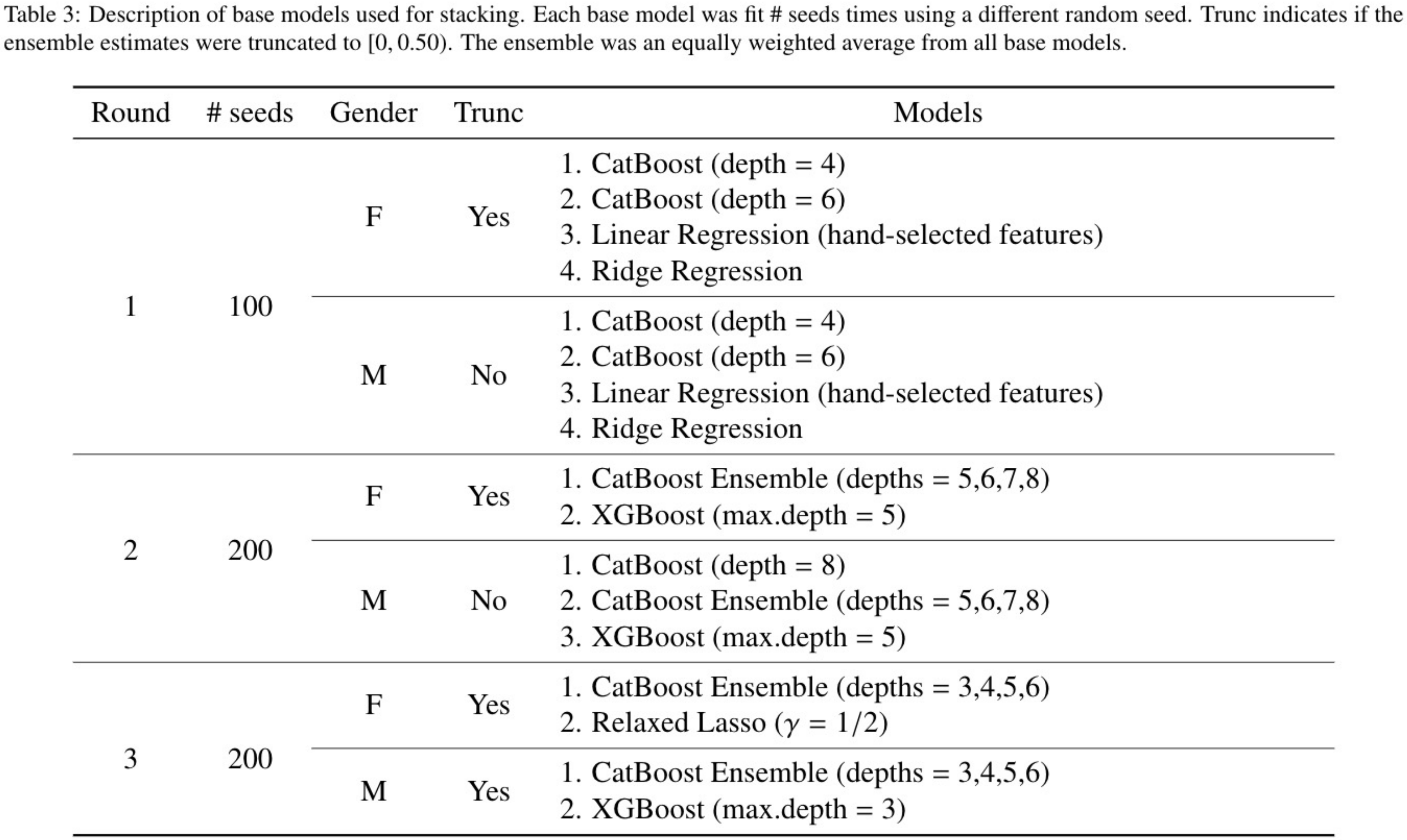


Table 3: Description of base models used for stacking. Each base model was fit # seeds times using a different random seed. Trunc indicates if the ensemble estimates were truncated to $[0, 0.50)$. The ensemble was an equally weighted average from all base models.

Round	# seeds	Gender	Trunc	Models
1	100	F	Yes	1. CatBoost (depth = 4) 2. CatBoost (depth = 6) 3. Linear Regression (hand-selected features) 4. Ridge Regression
		M	No	1. CatBoost (depth = 4) 2. CatBoost (depth = 6) 3. Linear Regression (hand-selected features) 4. Ridge Regression
2	200	F	Yes	1. CatBoost Ensemble (depths = 5,6,7,8) 2. XGBoost (max.depth = 5)
		M	No	1. CatBoost (depth = 8) 2. CatBoost Ensemble (depths = 5,6,7,8) 3. XGBoost (max.depth = 5)
3	200	F	Yes	1. CatBoost Ensemble (depths = 3,4,5,6) 2. Relaxed Lasso ($\gamma = 1/2$)
		M	Yes	1. CatBoost Ensemble (depths = 3,4,5,6) 2. XGBoost (max.depth = 3)

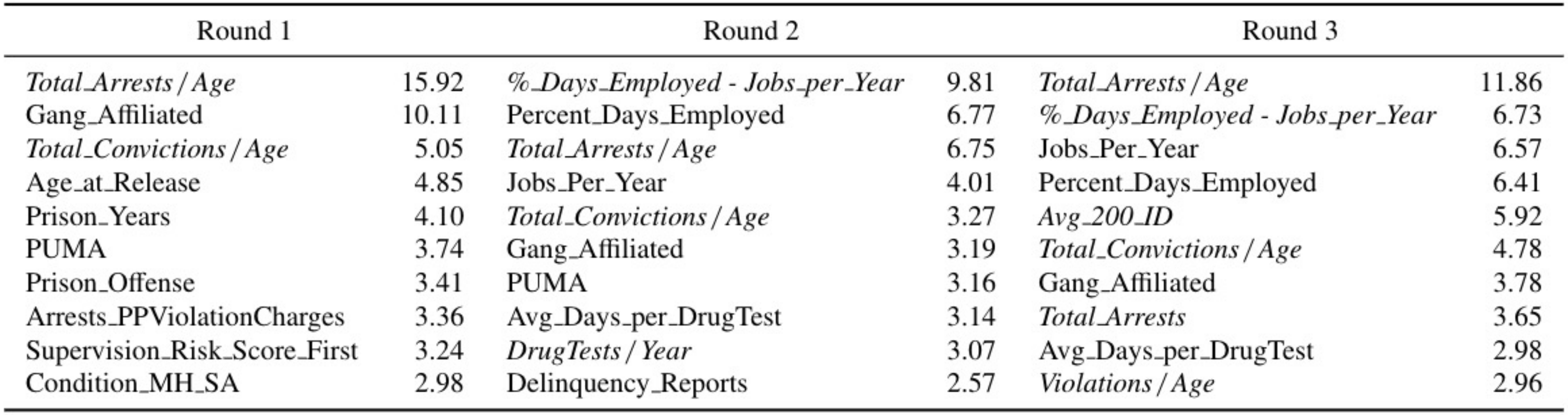
Models that did not improve score

- Random Forest
- Generalized Additive Models
- Neural Networks
- LightGBM/GBM

Important features

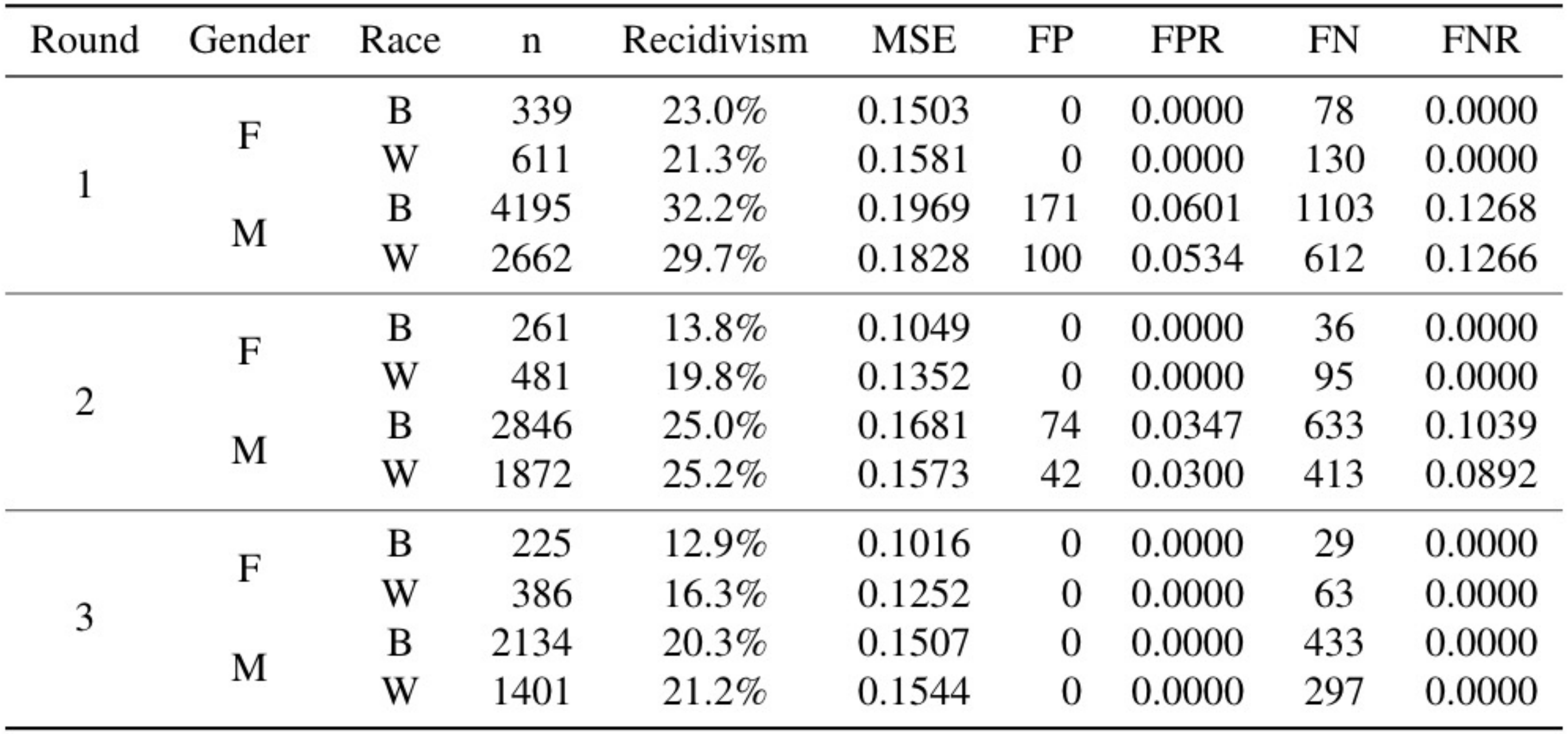
- Total arrests / Age at Release
- Percent Days Employed – Jobs Per Year
- Total convictions / Age at Release
- Smoothing recidivism label over ID
- Gang affiliated
- # drug tests=365/Average days per Drug Test
- Total arrests
- Total violations / Age at Release

Important features



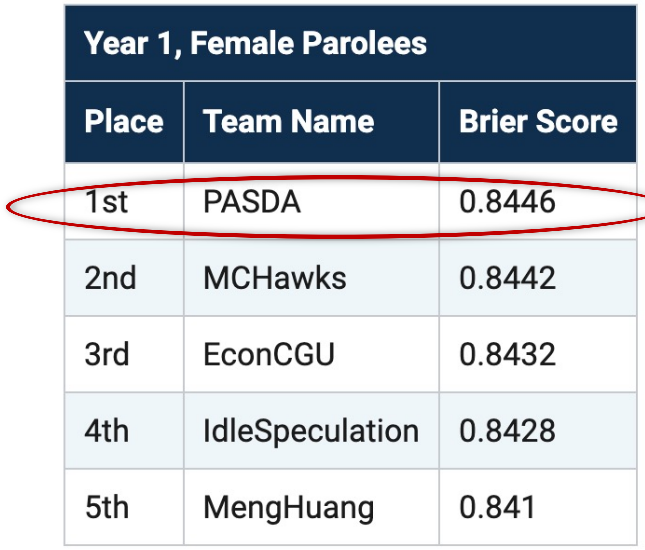
Round 1		Round 2		Round 3	
Total_Arrests / Age	15.92	*%_Days_Employed - Jobs_per_Year*	9.81	*Total_Arrests / Age*	11.86
Gang_Affiliated	10.11	Percent_Days_Employed	6.77	*%_Days_Employed - Jobs_per_Year*	6.73
Total_Convictions / Age	5.05	*Total_Arrests / Age*	6.75	Jobs_Per_Year	6.57
Age_at_Release	4.85	Jobs_Per_Year	4.01	Percent_Days_Employed	6.41
Prison_Years	4.10	*Total_Convictions / Age*	3.27	*Avg_200_ID*	5.92
PUMA	3.74	Gang_Affiliated	3.19	*Total_Convictions / Age*	4.78
Prison_Offense	3.41	PUMA	3.16	Gang_Affiliated	3.78
Arrests_PPViolationCharges	3.36	Avg_Days_per_DrugTest	3.14	*Total_Arrests*	3.65
Supervision_Risk_Score_First	3.24	*DrugTests / Year*	3.07	Avg_Days_per_DrugTest	2.98
Condition_MH_SA	2.98	Delinquency_Reports	2.57	*Violations / Age*	2.96

Overall Performance

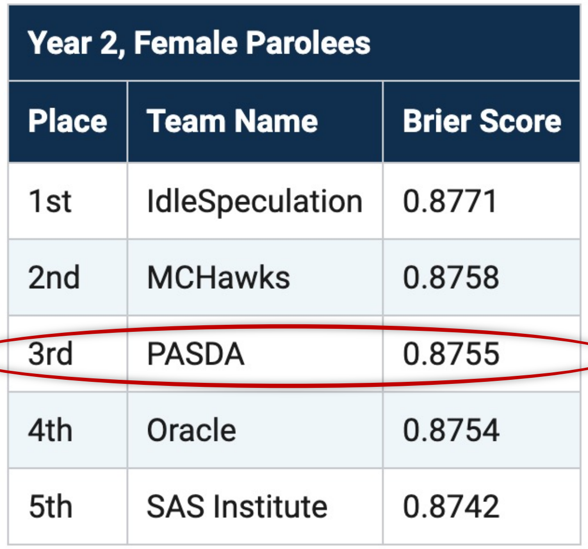


Round	Gender	Race	n	Recidivism	MSE	FP	FPR	FN	FNR
1	F	B	339	23.0%	0.1503	0	0.0000	78	0.0000
		W	611	21.3%	0.1581	0	0.0000	130	0.0000
	M	B	4195	32.2%	0.1969	171	0.0601	1103	0.1268
		W	2662	29.7%	0.1828	100	0.0534	612	0.1266
2	F	B	261	13.8%	0.1049	0	0.0000	36	0.0000
		W	481	19.8%	0.1352	0	0.0000	95	0.0000
	M	B	2846	25.0%	0.1681	74	0.0347	633	0.1039
		W	1872	25.2%	0.1573	42	0.0300	413	0.0892
3	F	B	225	12.9%	0.1016	0	0.0000	29	0.0000
		W	386	16.3%	0.1252	0	0.0000	63	0.0000
	M	B	2134	20.3%	0.1507	0	0.0000	433	0.0000
		W	1401	21.2%	0.1544	0	0.0000	297	0.0000

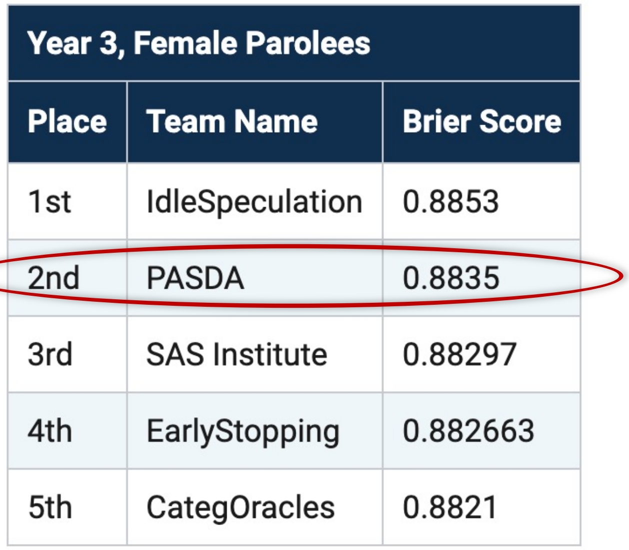
Overall Performance



Year 1, Female Parolees		
Place	**Team Name**	**Brier Score**
1st	PASDA	0.8446
2nd	MCHawks	0.8442
3rd	EconCGU	0.8432
4th	IdleSpeculation	0.8428
5th	MengHuang	0.841



Year 2, Female Parolees		
Place	**Team Name**	**Brier Score**
1st	IdleSpeculation	0.8771
2nd	MCHawks	0.8758
3rd	PASDA	0.8755
4th	Oracle	0.8754
5th	SAS Institute	0.8742



Year 3, Female Parolees		
Place	**Team Name**	**Brier Score**
1st	IdleSpeculation	0.8853
2nd	PASDA	0.8835
3rd	SAS Institute	0.88297
4th	EarlyStopping	0.882663
5th	CategOracles	0.8821