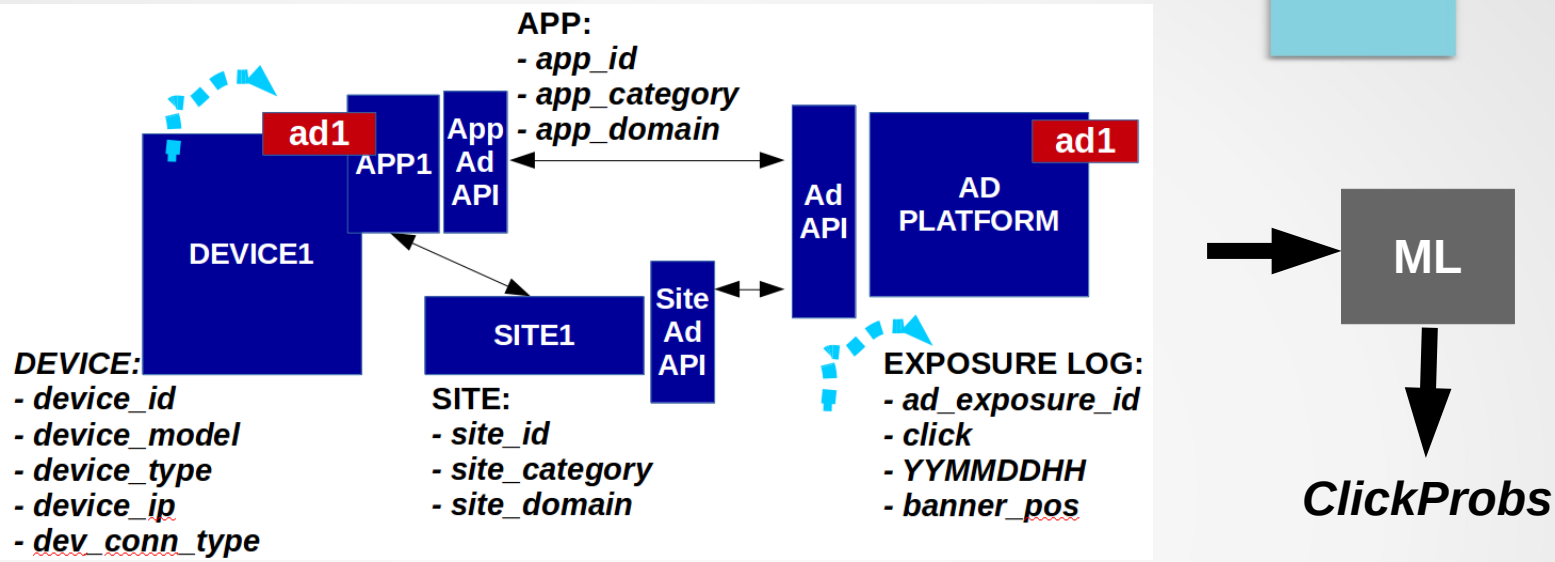
Ensemble Classifiers

A Flexible Bagging Ensemble Classifier (Applied To The Click-Through Problem)

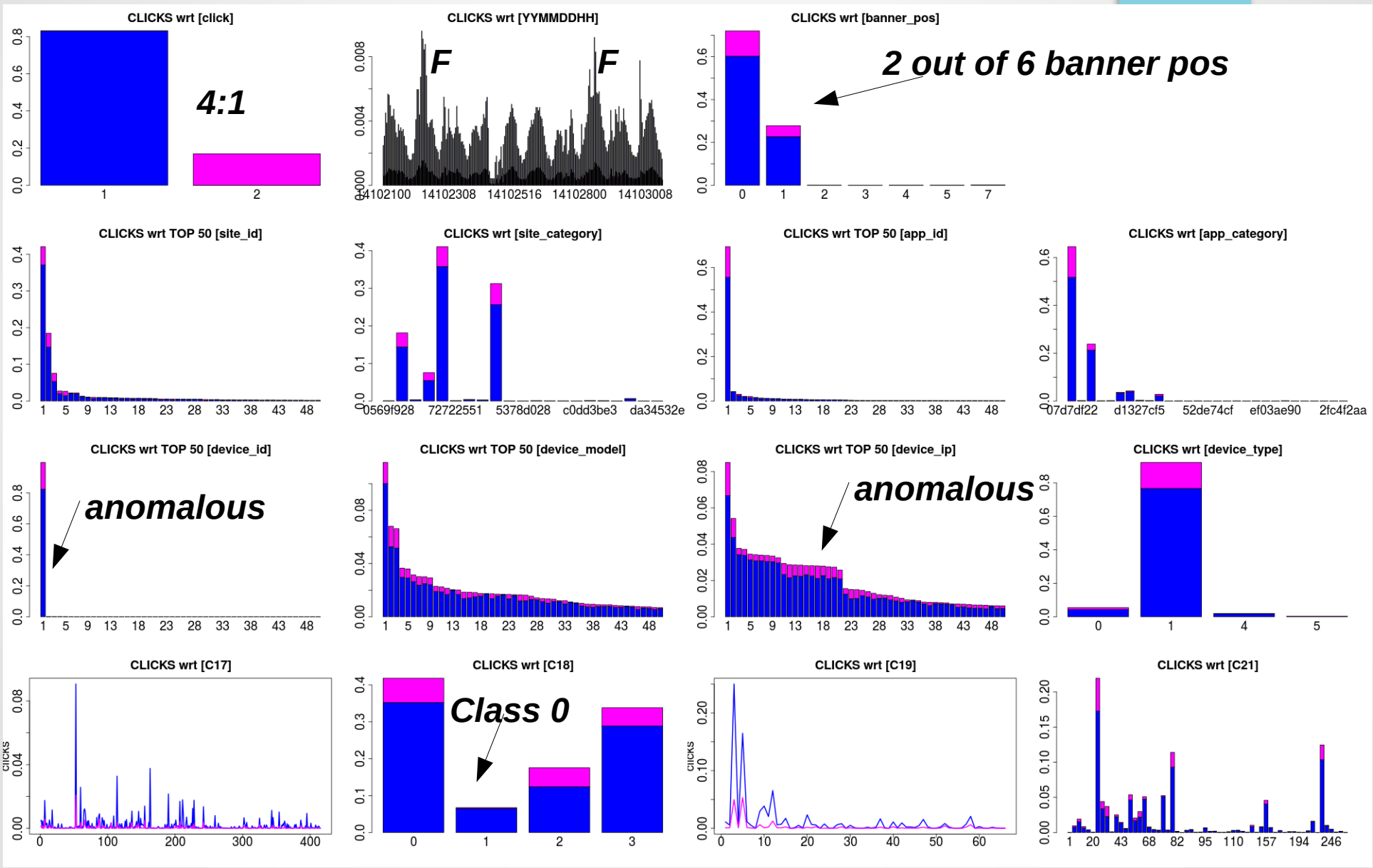
Nelson R. Manohar

Application (Click Through Rate)



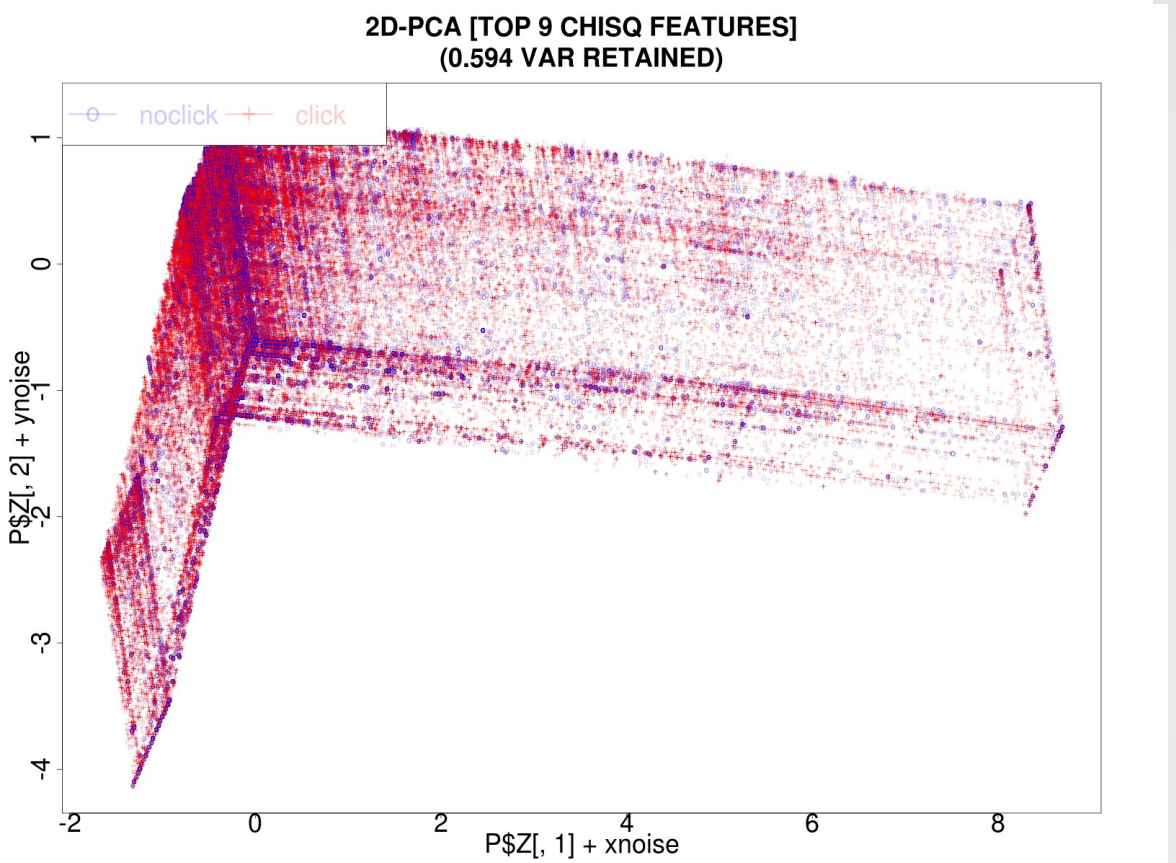
- GIVEN:
  - **Log** of millions of **ad impressions** (presented to mobile users by this AdPlatform) along with a bit indicating whether each impression resulted in Click or NoClick
- FEATURES (how each ad impression was exposed to the user):
  - ***anonymous*** numerical features and factors accounting for ***site*** visited, ***app*** used, type of ***device*** and connectivity, ***hour***, etc.
- PREDICT:
  - the **Click Probability (*ClickProb*)** of ad impression for test samples

Quick Look at the Features



Underlying Problem

- With respect to given features:
  - Domain knowledge missing (features are anonymous and no details given)
  - Clicks samples are quite similar to NoClicks samples



- Predictive Baseline:
  - How well do Dummy Classifiers do with this data?

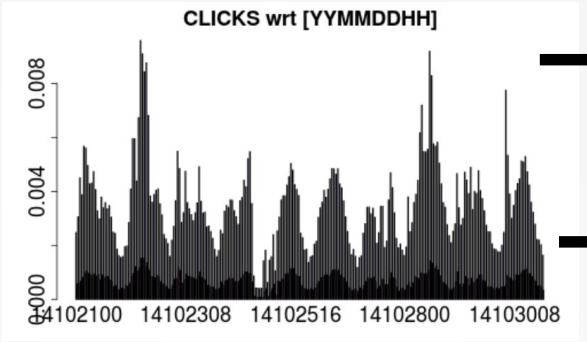
CLASSIFIER	LOG LOSS
DUMMY(0.999)	5.730061
DUMMY(0.5)	0.693147
DUMMY(0.001)	1.178694

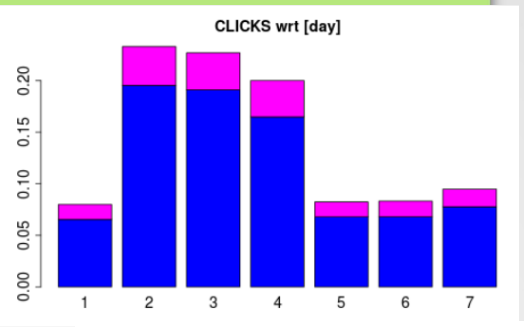
$logloss(y, \hat{y})$:

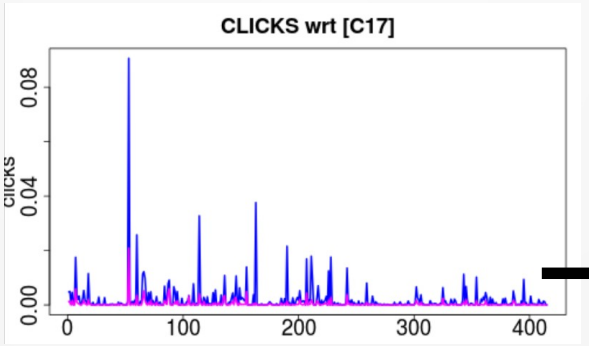
$$\frac{\sum_{i=1}^{n} (y \cdot \log(\hat{y})) + (1-y) \cdot \log(1-\hat{y})}{n}$$

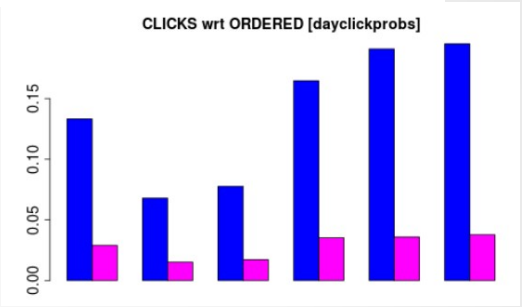
Basic Feature Generation

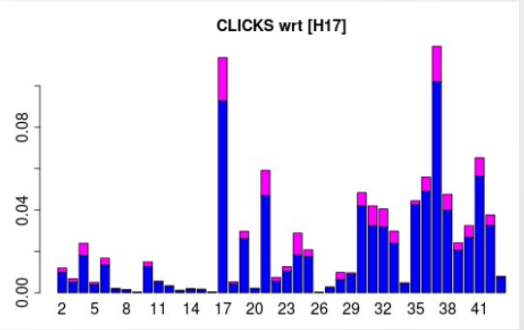
- Temporal
  - Day, Hour
- Probabilities
  - P(Click | Factor_Level)
- Interval Cuts:
- Num. Factor Levels
  - Top N

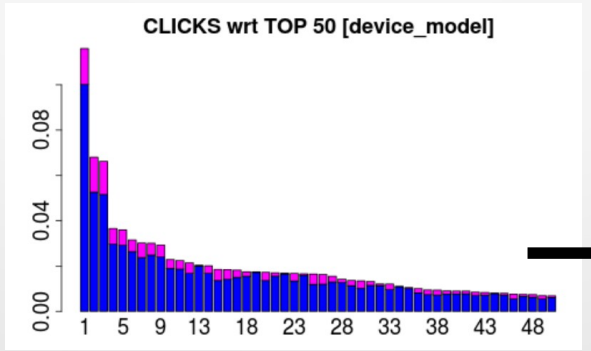


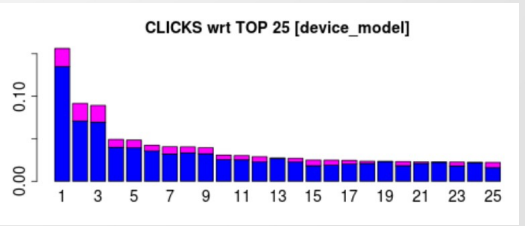




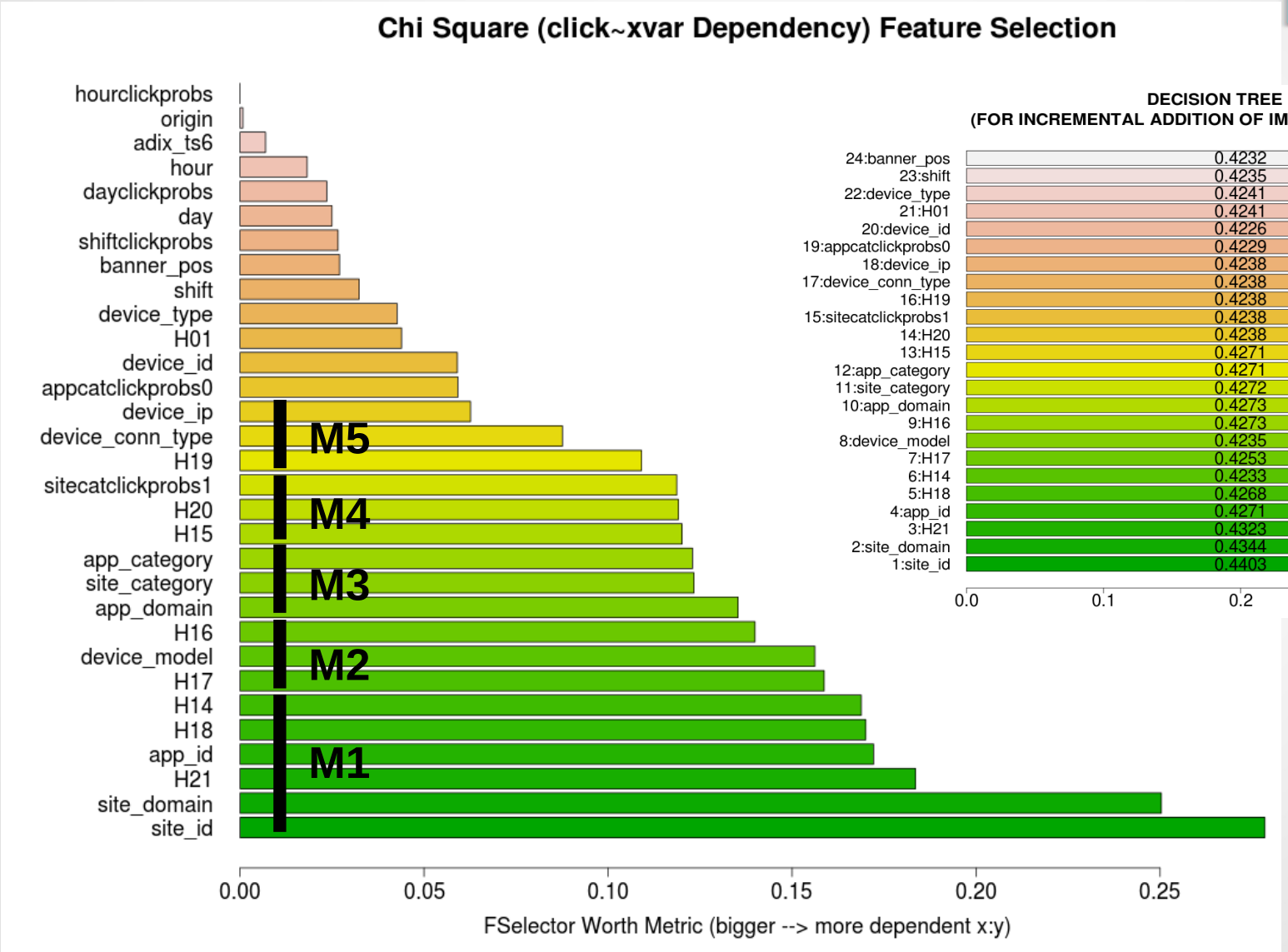






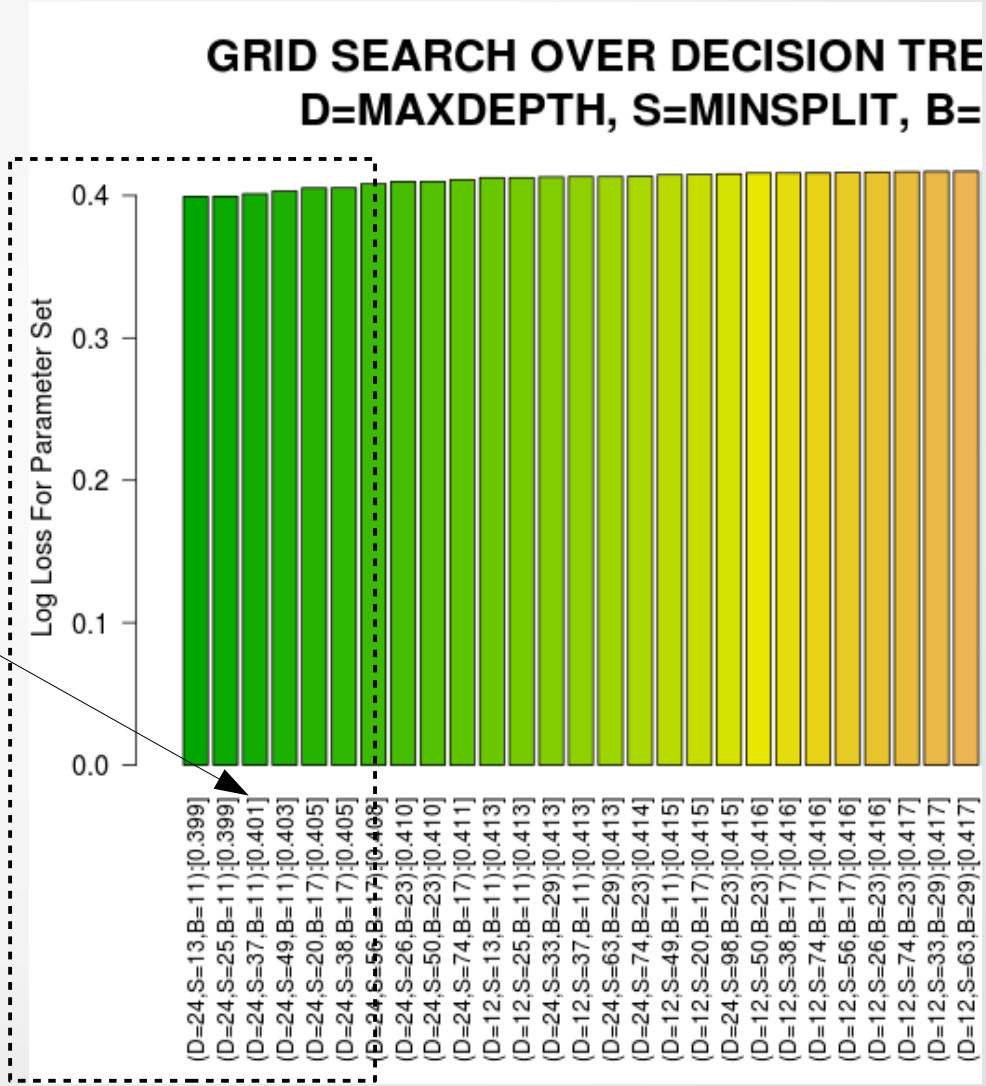


Feature Selection

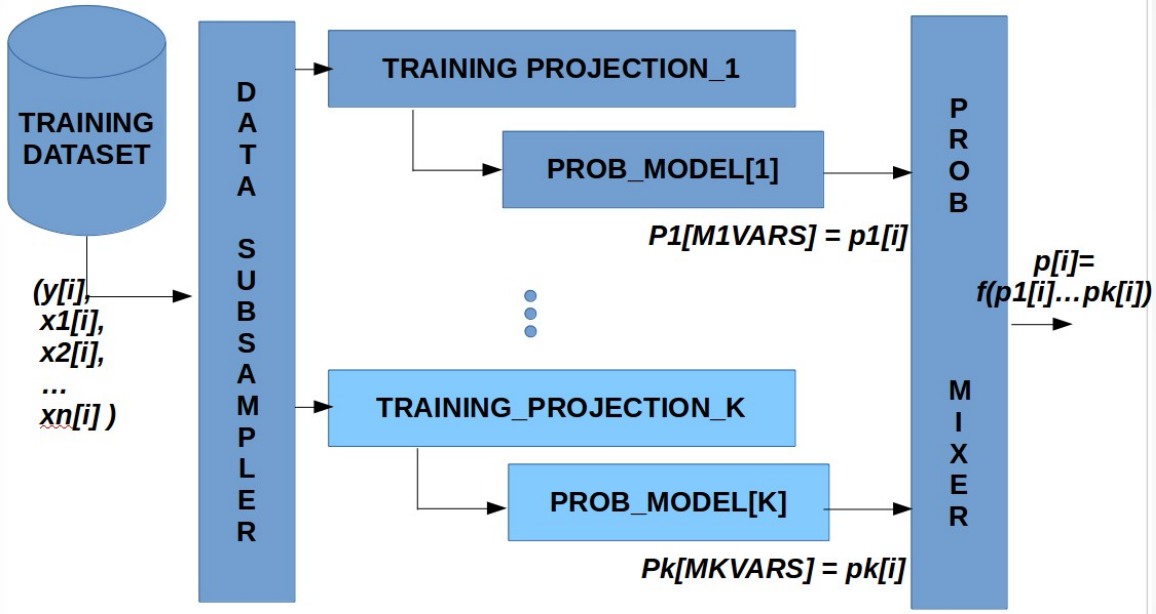


Grid Parameter Search (using Baseline Classifier)

- Problem is affine to Decision Trees
- Parameters for DT/RF:
  - **D=Max Depth**
  - **S=Min Split**
  - **B=Min Bucket**
- Region on
  - **D=24,S=20-50,B=11-17**



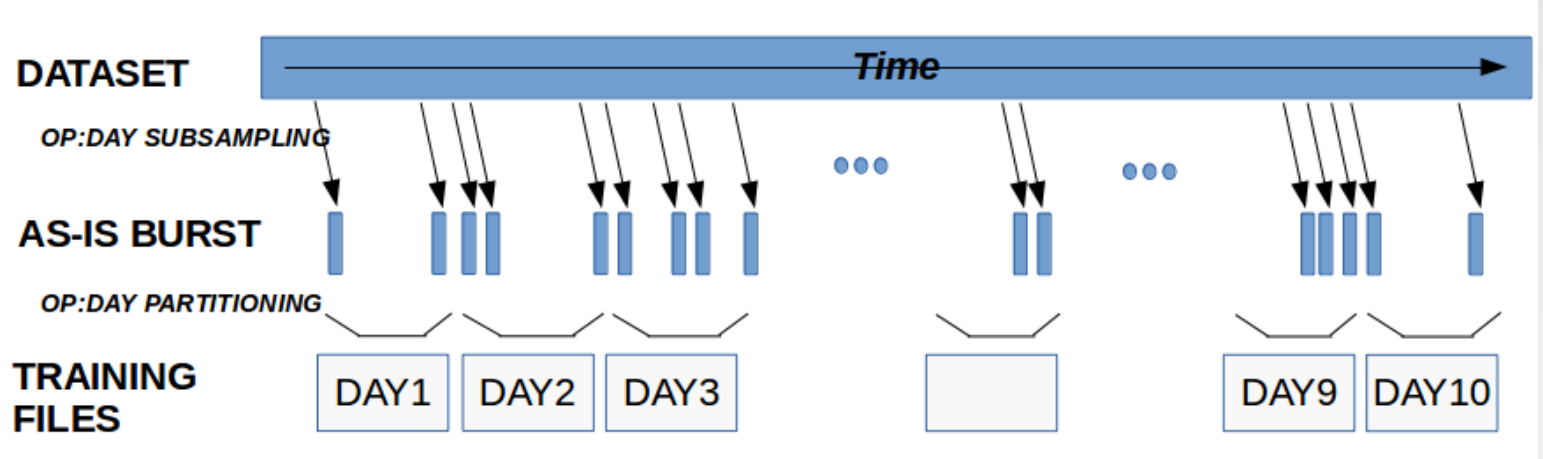
Ensemble Modeling (Bagging Framework): *ClickProbs = DT(click ~ X[S, xvars])*



- **GENERATE some candidate models:**
  - **M1 = DT(FSELECT_VARS[1:3]**
  - **M2 = DT(FSELECT_VARS[1:6]**
  - **M3 = DT(FSELECT_VARS[1:9]**
  - **M4 = DT(FSELECT_VARS[1:12]**
  - **M5 = DT(FSELECT_VARS[1:15]**

- (Up to) 9 models are specified; models can be replicated
- Prob. Mixer computes a weighted average of model's outputs
- Each model learns with different dataset slice **S** of size N
- Models *can* have different DT parameters (e.g, MaxDepth)
- For *these* slides, all models *have* same DT parameters

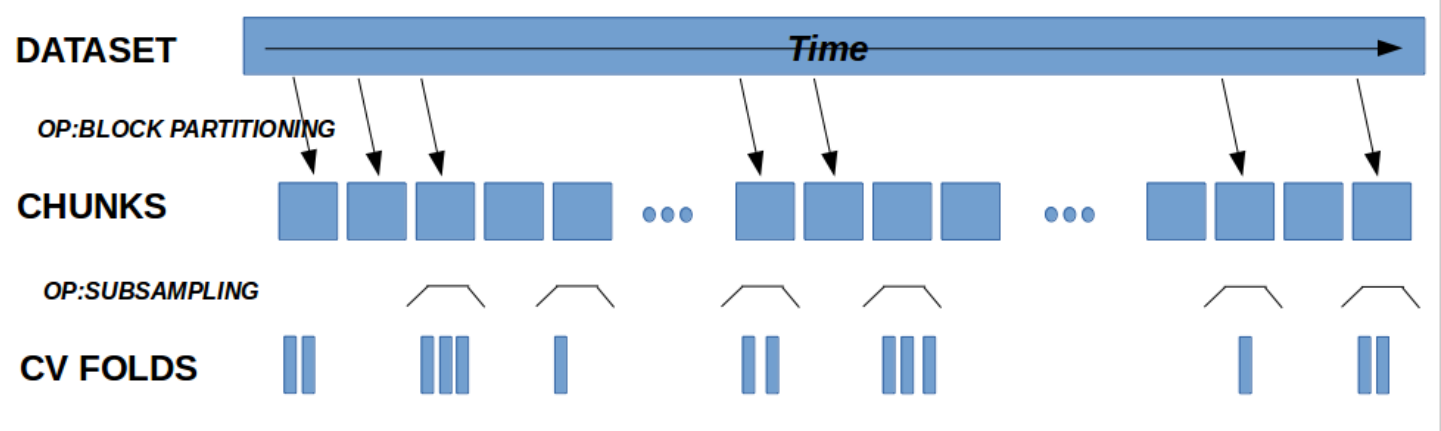
Subsampling Impact



- Dataset has both:
  - Class 1/0 imbalance
  - Temporal patterns
- Two Sub-sampling Choices:
  - Balance Class 1/0
  - Temporal Burst
- Class 1/0 balanced sub-sampling **increases** LogLoss

Training File ~600K training samples	mean(CV Log Loss) 540 CV folds
Balanced Classes	0.628399
Temporal Burst	0.410386

Cross-Validation (CV) Strategy



Y_TRUE \ YP_CLASS	0	1
0	812	8
1	167	13

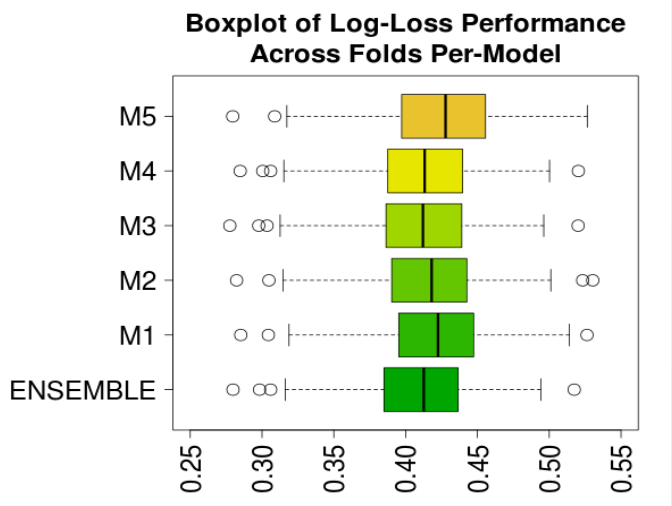
F_RATIO	FPR	FNR	ACCURACY	BER	MSE	PRECISION	RECALL	TPR	TNR	LOG_LOSS
0.13	0.0098	0.93	0.82	0.53	1.5	0.62	0.072	0.072	0.99	0.43

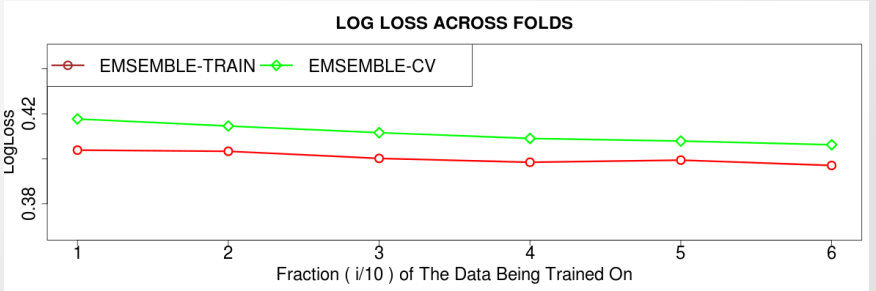
- Dataset partitioned into chunks of 100K samples
- Each chunk sampled with 3 random bursts of 1K samples
- Each temporally cohesive sample represented a CV slice
- Log Loss computed for each CV fold; then average reported

Ensemble-1: Cross Validation

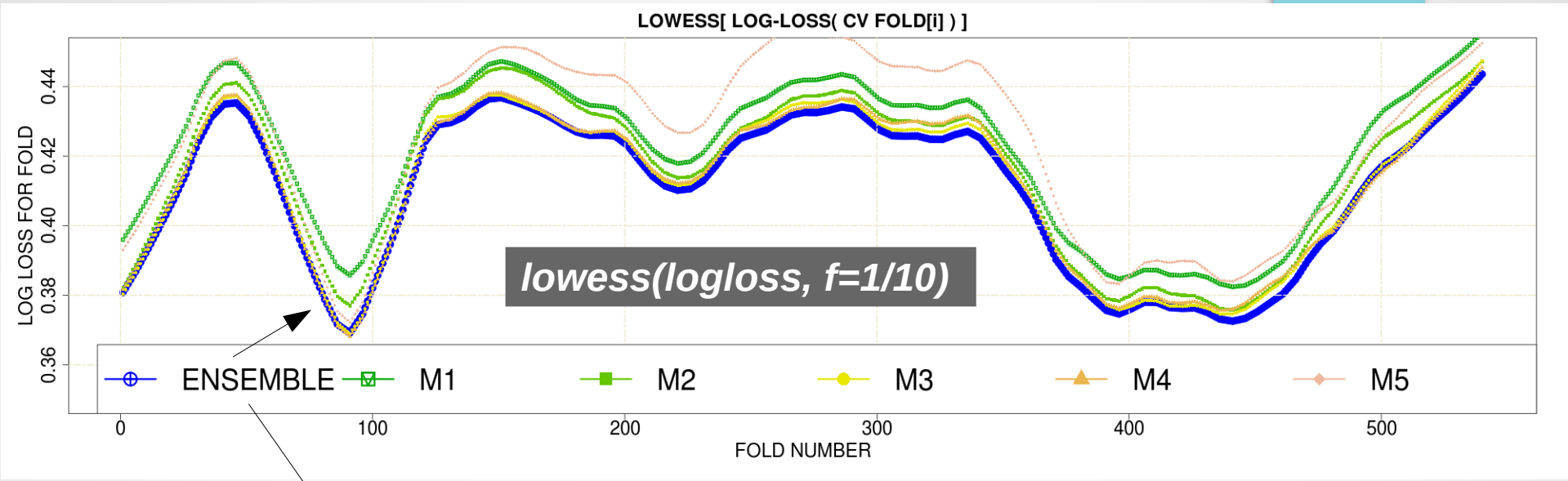
- First, using 540 CV folds,
  - Apply models to fold
  - Obtain each model output's click probabilities
  - Compute ensemble's click probabilities from those
  - Compute Log-Loss values for all models
- Then, to compare ***candidate*** models, examine range of Log Loss of the models
- Then, assess the ensemble's Learning Curve operating region

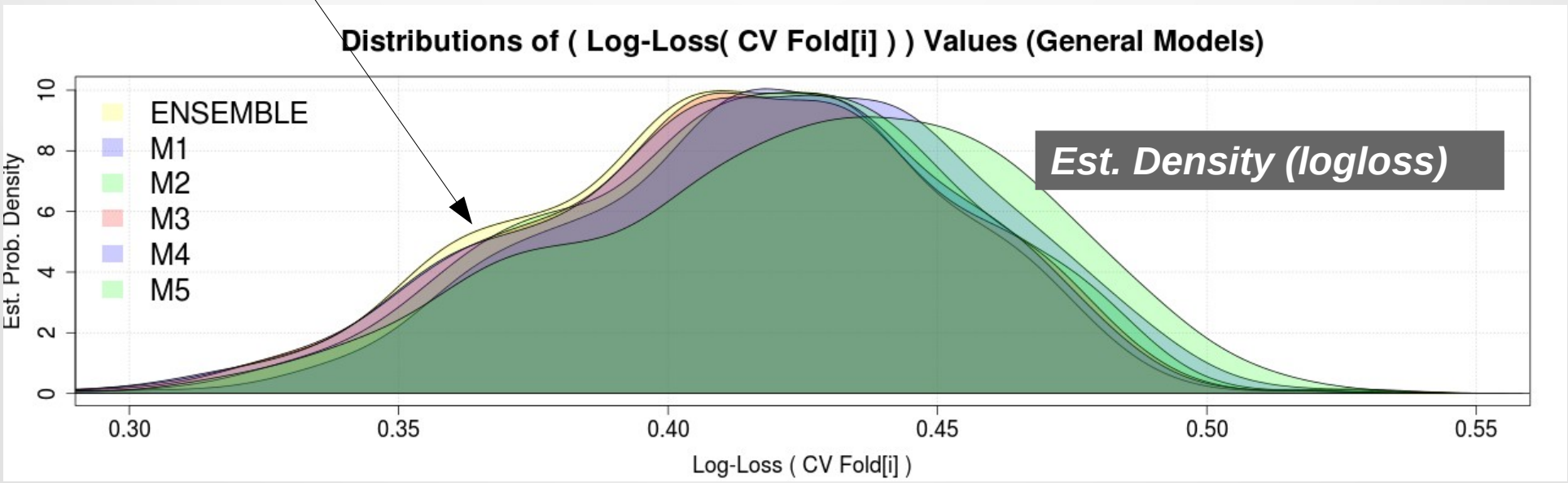
	"AVGLL"	"MINLL"	"MAXLL"	"MEDLL"
" M1"	0.420784	0.2852	0.5263	0.42255
" M2"	0.415791	0.2823	0.5303	0.4181
" M3"	0.41206	0.2776	0.5201	0.41205
" M4"	0.412132	0.2848	0.5203	0.41325
" M5"	0.424618	0.2796	0.5265	0.4278
"ENSEMB"	0.410386	0.2798	0.5174	0.41265





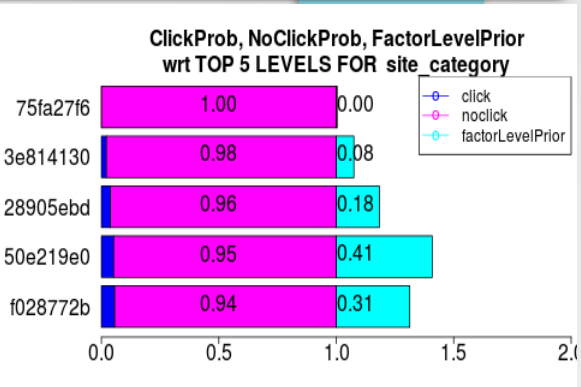
Ensemble Log Loss & Click Probabilities

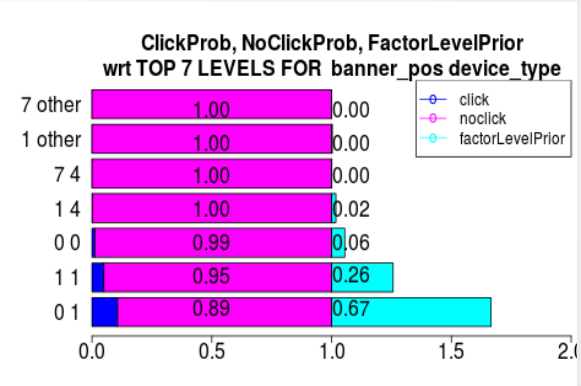


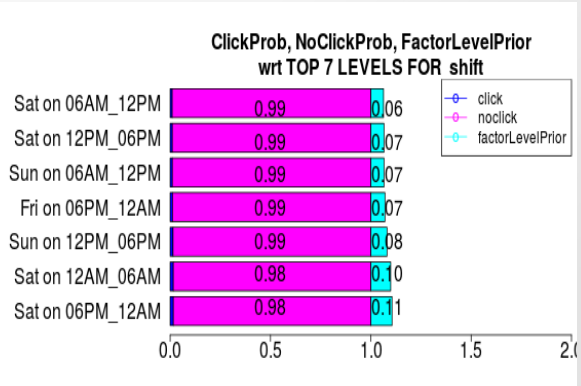


Observations: Factor Levels & ClickProbs

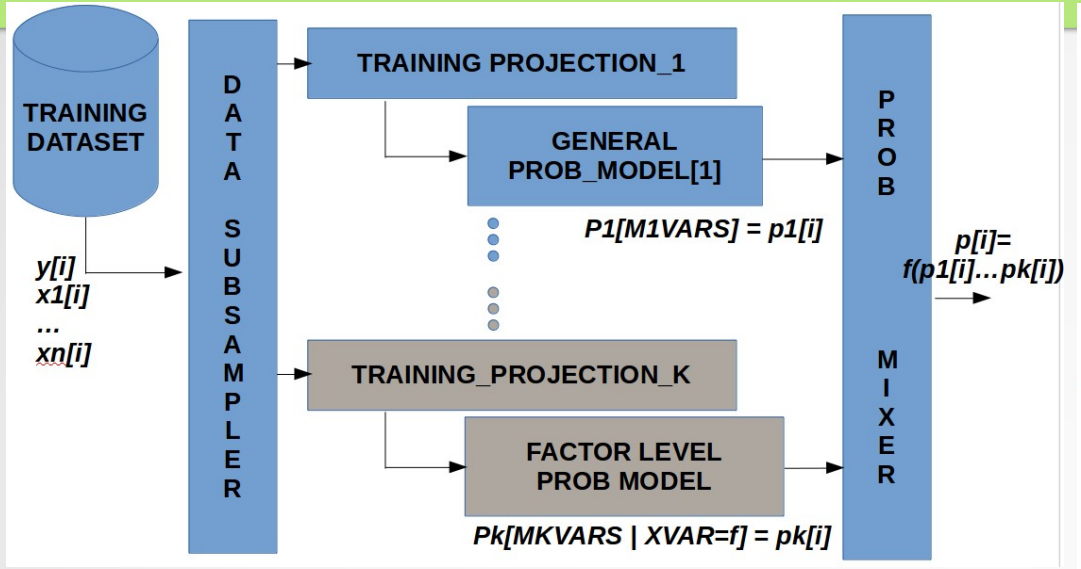
- Explore/Exploit Customized Modeling to:
  - Automatically exploit narrowly targeted classifiers to boost performance
  - by specifying a **targeting hypothesis of the problem-domain, e.g.,**
    - Which sites are click more likely?
    - Are there combinations of device type & ad placement for which clicks are more likely?
    - Are there times of the week at which clicks are more likely?
- Not boosting (i.e, not iterative, weight refinement yet)







Adding Customized (Per Factor-Level) Models



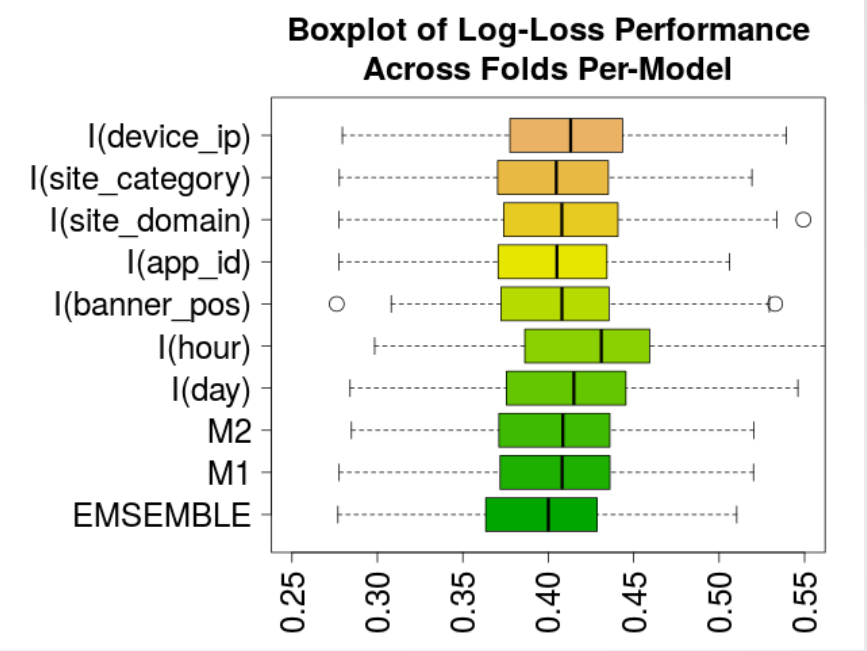

```
BuildFactorLevelPredictorFor( XY, V ):
    P = list()
    for each factor f in factors(V):
        subset = which(XY$V == f)
        if |subset| too small: skip
            pf = rpart("click ~ .", XY[subset,-V]))
            training_logloss = …
            if ( training_logloss too large ): skip
            P[f] = pf
    return (P)

PredictForFactorLevelPredictor( XY, V, P ):
    yp = predict(default_predictor)
    for each factor f in factors(V):
        if f in P:
            subset = which(XY$V == f)
            yp[subset] = rpart.predict( P[f], XY[subset,])
    return (yp)
```

- Train a predictive model for each factor-level model ***F=flevel***
  - ***DT(click~ X[DatasetSlice[F=****flevel*,],])**
  - **Customized Training,** only with samples in slice assoc. with factor level ***flevel***.
  - **Customized Feature Selection** done for each factor-level model ***F=flevel***.
  - Selects only models with **logloss** performance above a threshold.
- During prediction, **predict** if matching model exist, else, output default probs.
- Number of models does not fit in memory so stored on disk; later on DB (for MR).

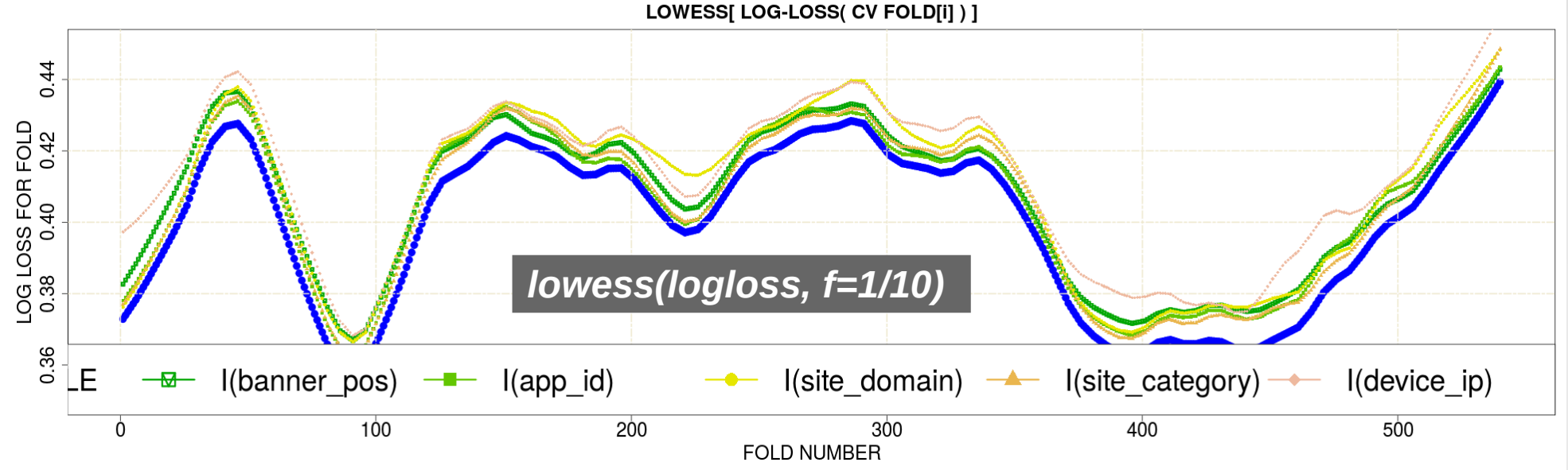
… Applied to App, Sites, Devices Hypothesis

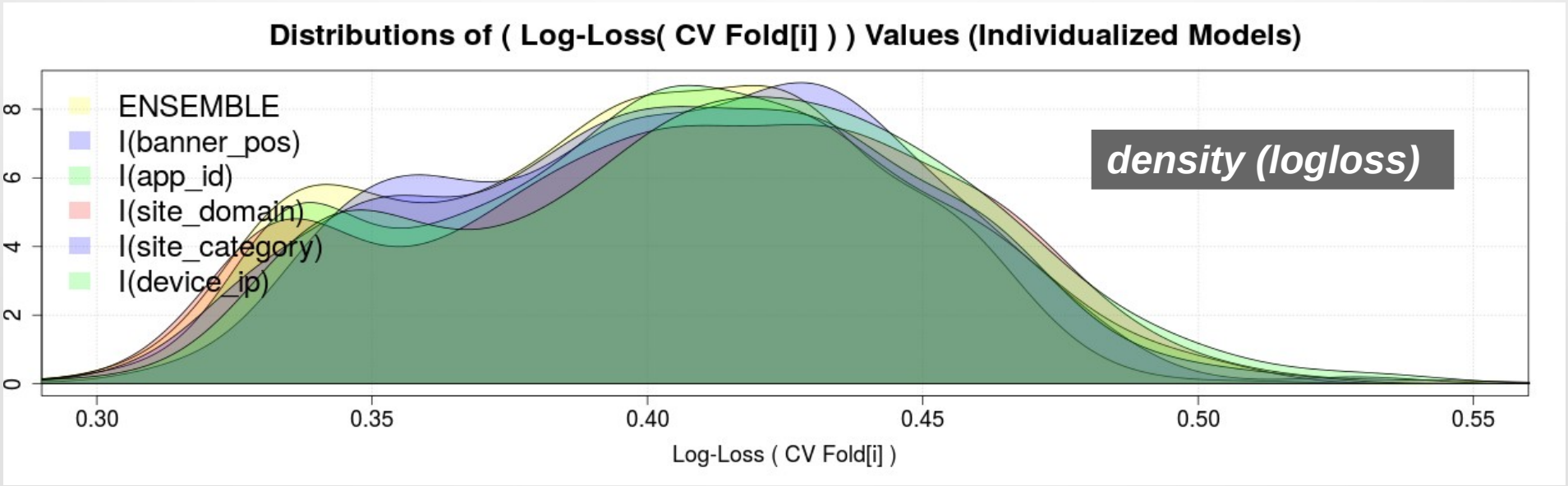
	"AVGLL"	"MINLL"	"MAXLL"
" M1<--M3"	0.404667	0.2776	0.5201
" M2<--M4"	0.405016	0.2848	0.5203
"I(day)"	0.412685	0.284	0.5461
"I(hour)"	0.427666	0.2984	0.5791
"I(banner_pos)"	0.405592	0.2762	0.5328
"I(app_id)"	0.403019	0.2775	0.5061
"I(site_domain)"	0.406195	0.2775	0.5493
"I(site_category)"	0.40301	0.2777	0.5192
"I(device_ip)"	0.410526	0.2795	0.5393
"ENSEMB"	0.396831	0.2769	0.5103



- Exploits predictive power of individual factor levels (e.g., ***F*** = *site_id*, ***fl*** = *mail.yahoo.com*)
- Facilitates comparative evaluation of the predictive effect against baseline (e.g., paired *t-test* between CV LogLoss value populations of ***Hour*** vs. ***M1***)
- Facilitates exploring domain hypotheses (*site_id, hour, banner_pos)*
- Can be extended into parallelization as well as boosting

CV Log Loss Values For Custom Models





Ensemble Performance Trajectory

	"AVGLL"	"MINLL"	"MAXLL"
" M1"	0.420784	0.2852	0.5263
" M2"	0.415791	0.2823	0.5303
" M3"	0.41206	0.2776	0.5201
" M4"	0.412132	0.2848	0.5203
" M5"	0.424618	0.2796	0.5265
"ENSEMB"	0.410386	0.2798	0.5174

	"AVGLL"	"MINLL"
"M1<-M3"	0.404667	0.2776
"M2<-M4"	0.405016	0.2848
"I(H21)"	0.431589	0.2873
"I(H18)"	0.409894	0.2792
"I(H19)"	0.406379	0.2775
"I(H17)"	0.411549	0.2822
"I(H14)"	0.410192	0.2744
"I(H15)"	0.410534	0.2795
"I(H20)"	0.408787	0.2789
"ENSEMB"	0.399117	0.2752

	"AVGLL"	"MINLL"
" M1<--M3"	0.404667	0.2776
" M2<--M4"	0.405016	0.2848
"I(day)"	0.412685	0.284
"I(hour)"	0.427666	0.2984
"I(banner_pos)"	0.405592	0.2762
"I(app_id)"	0.403019	0.2775
"I(site_domain)"	0.406195	0.2775
"I(site_category)"	0.40301	0.2777
"I(device_ip)"	0.410526	0.2795
"ENSEMB"	0.396831	0.2769

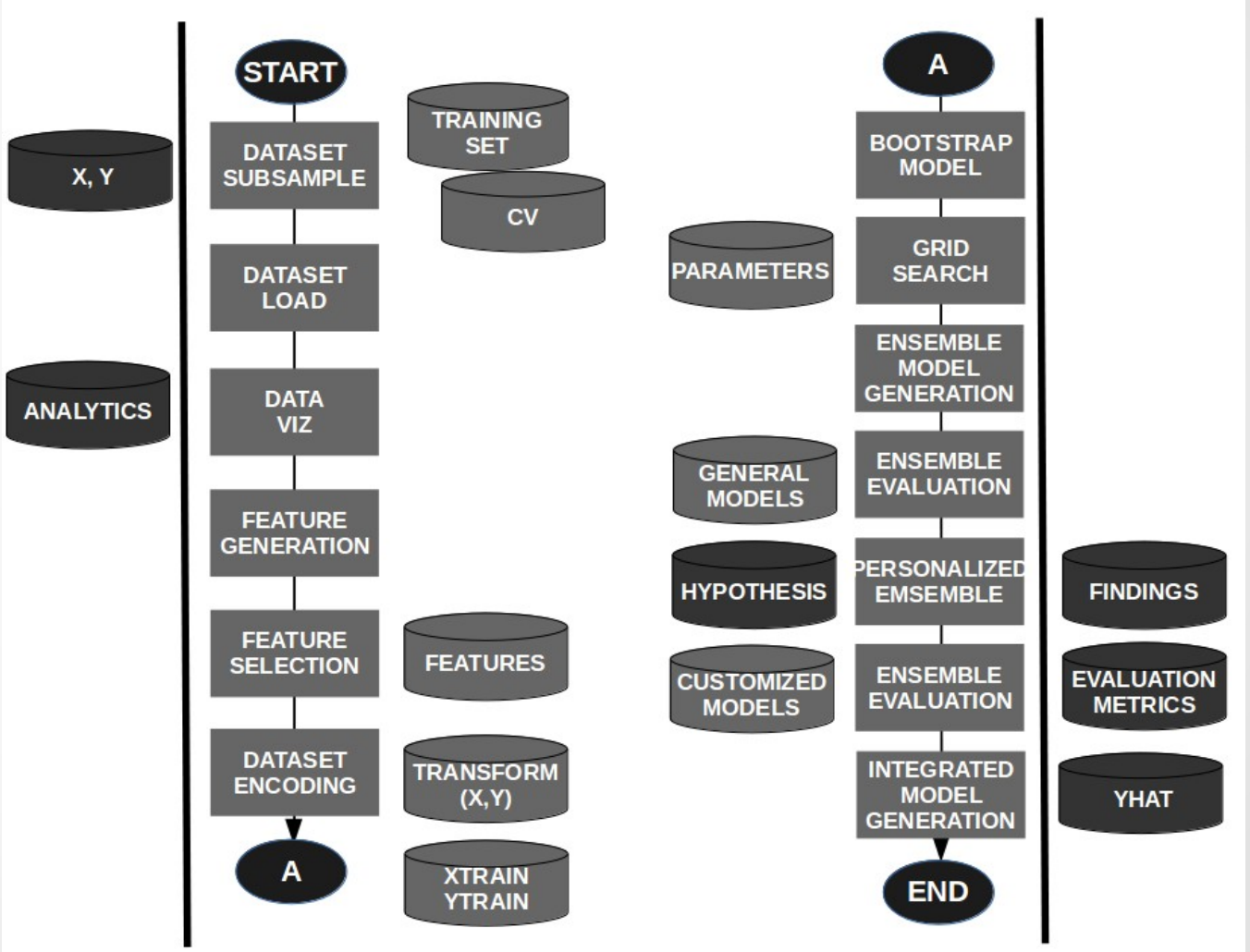
- Training done with 0.5% of the data
  - Yet by trading 10x compute time for 1/100 training size, we reduced Log Loss to Kaggle levels.

Yet all 10x compute tasks are coarse-grain.

Very amenable to MR.

Data Pipeline

- A flexible ensemble classifier pipeline in R:
  - For exploration of Factor/DT oriented ensemble modeling
  - Reusable and modular prototyping code base in R.
  - Ensembles positioned to be parallelized
- Code at bitbucket.org/nelsonmanohar/machinelearning



Concluding Remarks

- **Future Work:**
  - Moving models and factor-level models into **network DB**
  - **MR'd execution** & throttling of model building/execution
  - **Visualization & meta-analysis** of exploratory modeling results over time
  - Autonomous **feature exploration and generation** of conditional probability features, clustering (e.g, social network features)
  - **Boosting** of factor-level models and/or ensembles
  - Integration of **heterogeneous classifiers** for special cases
  - **Python**, SQL, **MR**, Spark, More **Datasets** to Examine

Appendix Section

End of Presentation

Misc Slides: Background

- Software Systems
  - Telephony and Intelligent (1-800) Networks
  - Multimedia Groupware (Asynchronous Collaboration)
  - Scalable Cloud Systems for Media (IP)
  - Measurements (Stationary Detection)
- Data Mining and Machine Learning
  - DNA alignment prototyping
  - Retraining Post Doc in Social Network Analysis of Newsgroups
  - Malware Detection system (Windows and Android)
- Education
  - Ph.D Computer Science & Engineering (1997)

Dataset Details & Illustrative Samples

- Some details of the Dataset

Attribute	*Description/Value*
Name	Click Through Rate
Description	Click/NoClicks response to mobile device ad exposures
Source	Kaggle
Size	5.9 GB
Number of records	40428968
Number of original features	24 features, 1 binary predictor class
Timespan covered	10 days to predict 1 full day, offline
Class imbalance	0.18 click density (approx, 5 to 1)
Classification type	Binary classification, Click/NoClick
Type of features	Anonymous numerical and categorical
Feature sets	Device, ClickExposure, Network, Site, App
Performance measure	Log loss of the resulting probabilities
Misc. remarks	Ad id not provided.

- A look at the raw data...

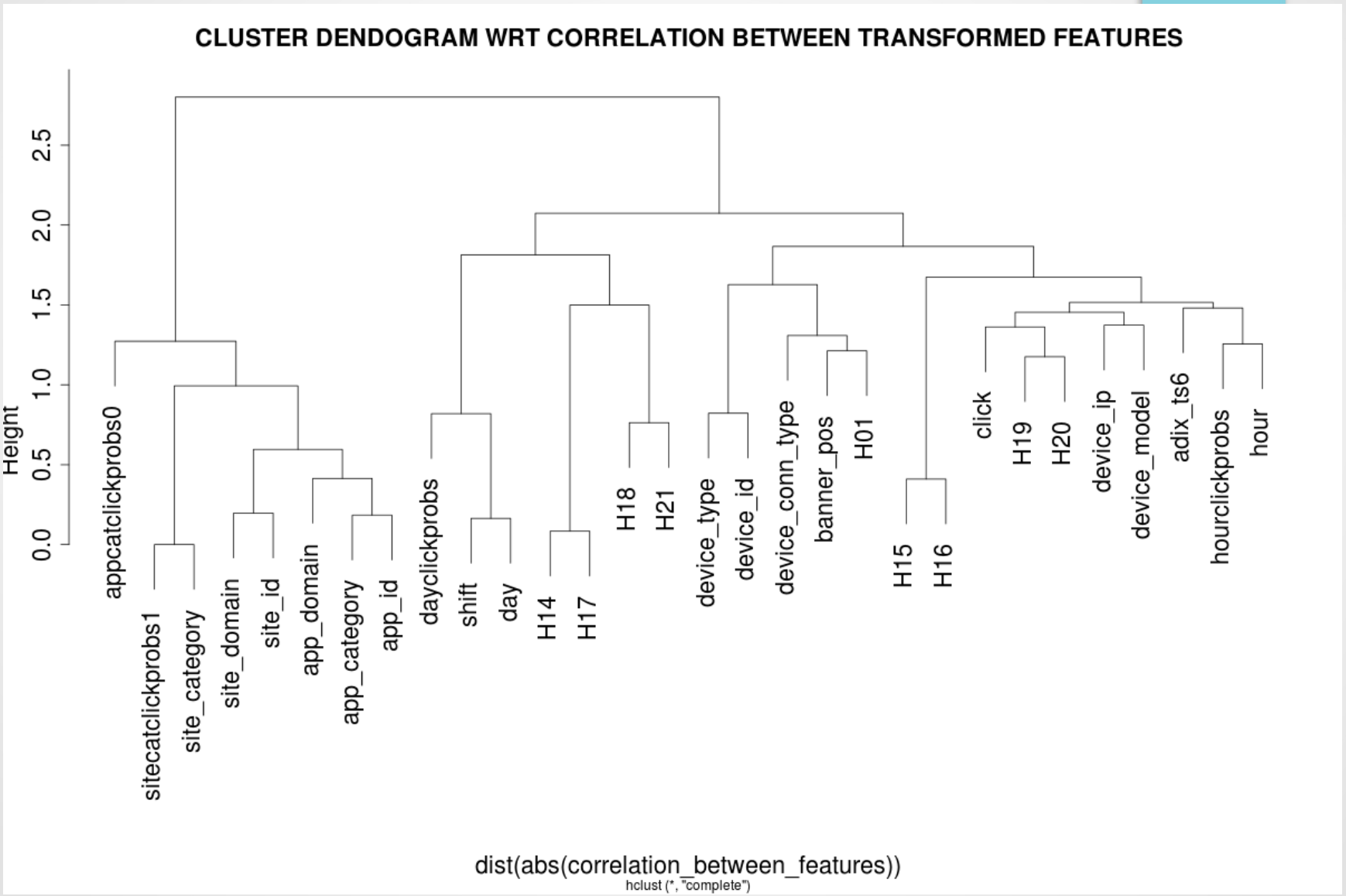
```
X = read.csv( TRAINSET, nrows=5, header=TRUE, stringsAsFactors=TRUE, colClasses=c('character',rep('integer',4)
```

	id	click	hour	C1	banner_pos	site_id	site_domain	site_category	app_id	app_domain
6	10000720757801103869	0	14102100	1005	0	d6137915	bb1ef334	f028772b	ecad2386	7801e8d9
7	10000724729988544911	0	14102100	1005	0	8fda644b	25d4cfcd	f028772b	ecad2386	7801e8d9
8	10000918755742328737	0	14102100	1005	1	e151e245	7e091613	f028772b	ecad2386	7801e8d9
9	10000949271186029916	1	14102100	1005	0	1fbe01fe	f3845767	28905ebd	ecad2386	7801e8d9
10	10001264480619467364	0	14102100	1002	0	84c7ba46	c4e18dd6	50e219e0	ecad2386	7801e8d9

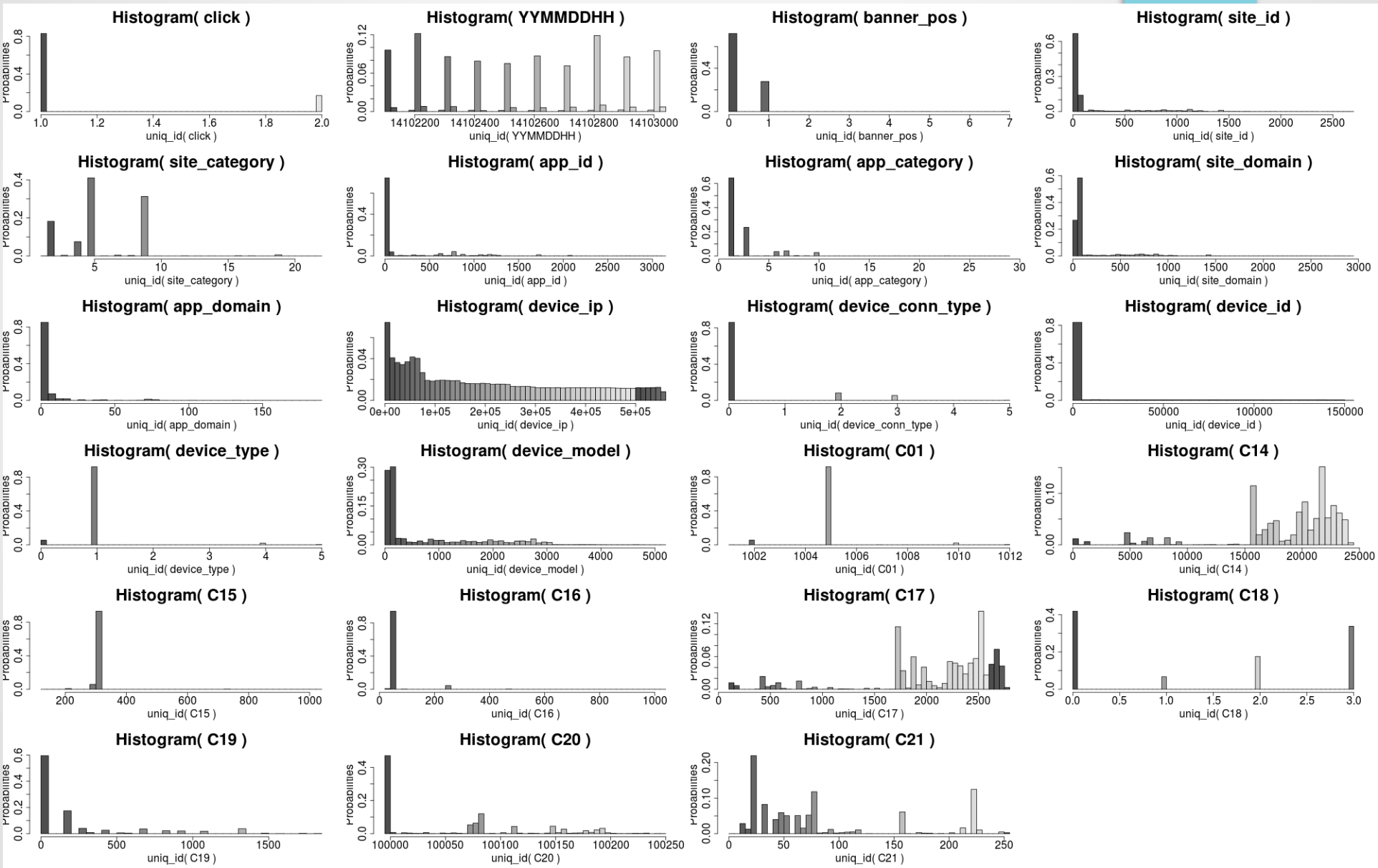
```
), rep('factor',9), rep('integer',10)))
```

app_category	device_id	device_ip	device_model	dev_type	dev_conn_type	C14	C15	C16	C17	C18	C19	C20	C21
07d7df22	a99f214a	05241af0	8a4875bd	1	0	16920	320	50	1899	0	431	100077	117
07d7df22	a99f214a	b264c159	be6db1d7	1	0	20362	320	50	2333	0	39	-1	157
07d7df22	a99f214a	e6f67278	be74e6fe	1	0	20632	320	50	2374	3	39	-1	23
07d7df22	a99f214a	37e8da74	5db079b5	1	2	15707	320	50	1722	0	35	-1	79
07d7df22	c357dbff	f1ac7184	373ecbe6	0	0	21689	320	50	2496	3	167	100191	23

Miscellaneous Appendix Slides: Correlation Between Features



Misc. Slides: Dataset Details (1% subsample)



Baseline Classifier & Performance

- Problem affine to ensemble of DT to address over-fitting

Classifier	Disadvantages
NB	Generative: decision function based solely on joint distribution model
KNN	Requires numerical data
Radial SVM	Computational complexity and requires numerical data
Logistic Regression	Factor interactions quickly add up complexity
LDA	Generative and requires numerical data
Decision Trees	Prone to overfit, can be addressed via tree controls (maxdepth, etc), pruning, and feature selection.
Random trees	Scale. Probabilities derived from votes requiring many trees.

- Log Loss metric is intuitively different to F-ratio

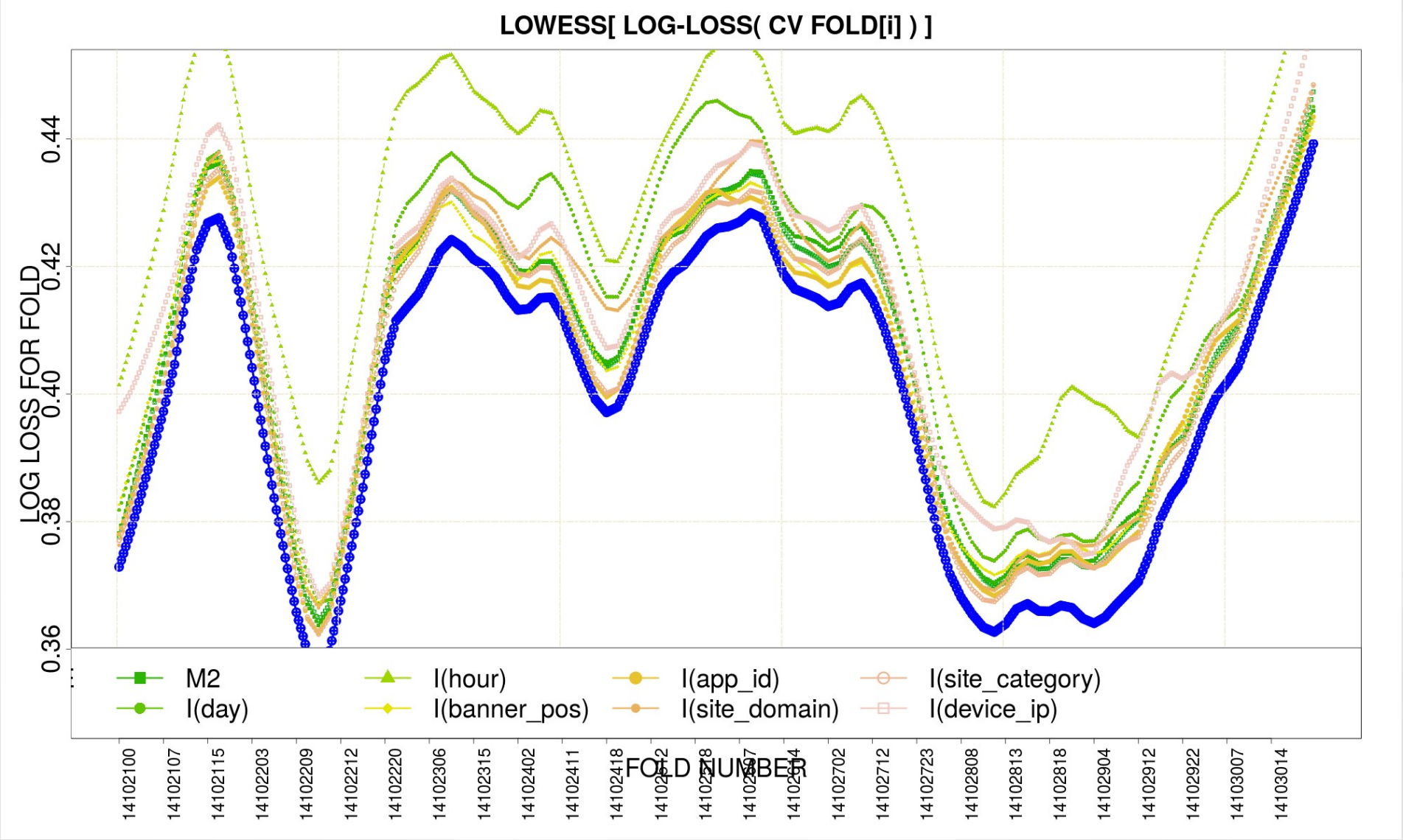
	YP_CLASS	
Y_TRUE	0	1
0	812	8
1	167	13

F_RATIO	FPR	FNR	ACCURACY	BER	MSE	PRECISION	RECALL	TPR	TNR	LOG_LOSS
0.13	0.0098	0.93	0.82	0.53	1.5	0.62	0.072	0.072	0.99	0.43

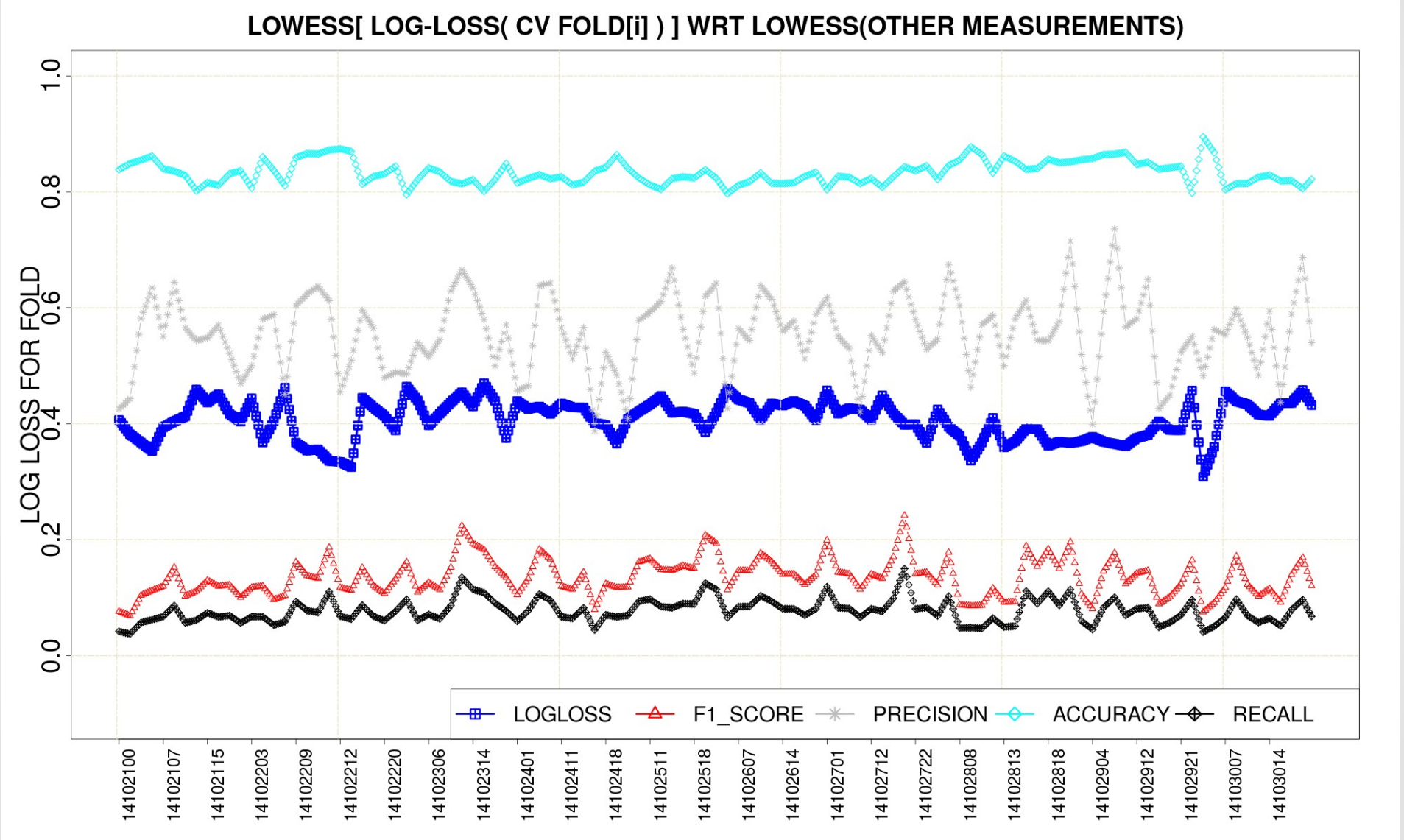
	YP_CLASS	
Y_TRUE	0	1
0	857	7
1	133	3

F_RATIO	FPR	FNR	ACCURACY	BER	MSE	PRECISION	RECALL	TPR	TNR	LOG_LOSS
0.041	0.0081	0.98	0.86	0.51	1.4	0.3	0.022	0.022	0.99	0.35

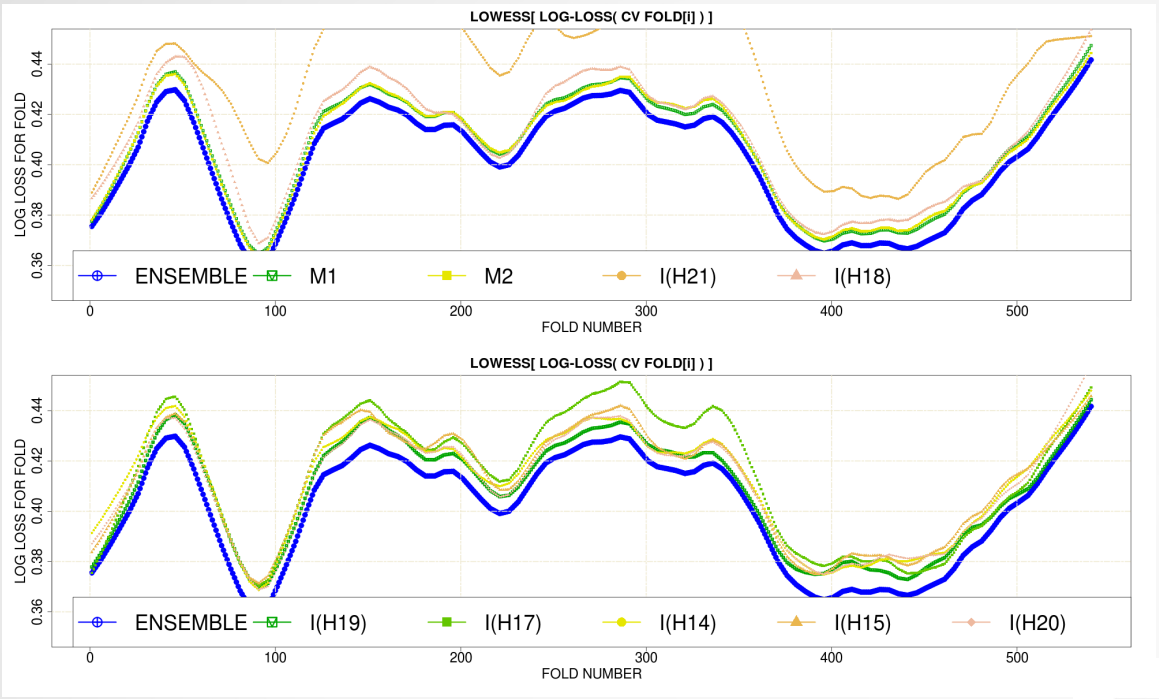
Miscellaneous Appendix Slides
Time Span of the CV Folds[1:540]

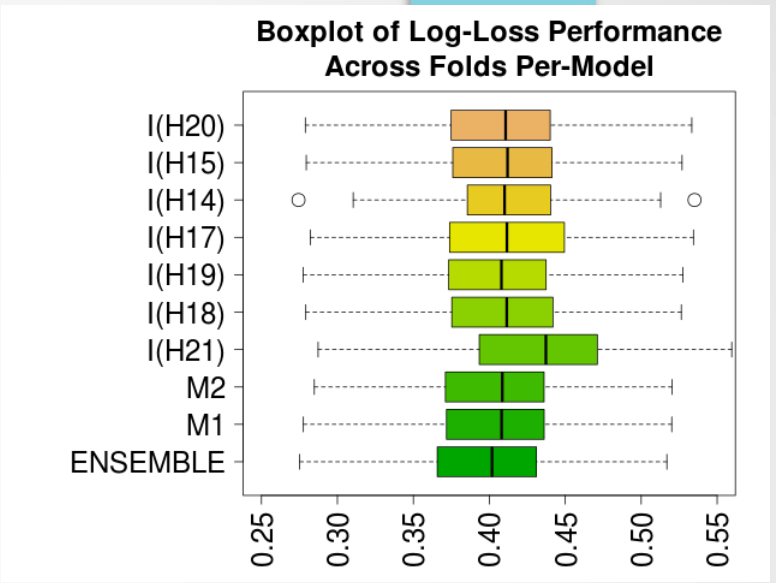


CV Log Loss vs. Other Metrics
(lowess(x, f=1/100))

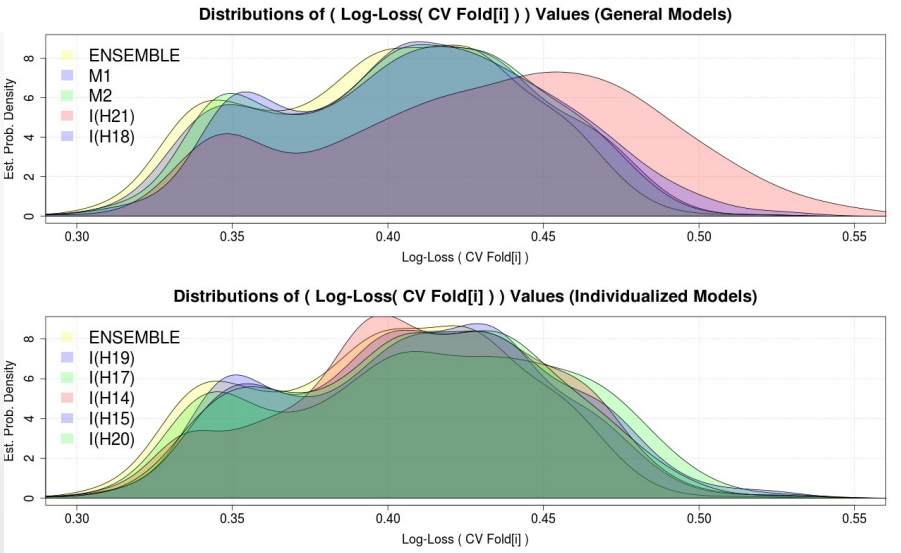


Exploration modeling applied to H[i] factors

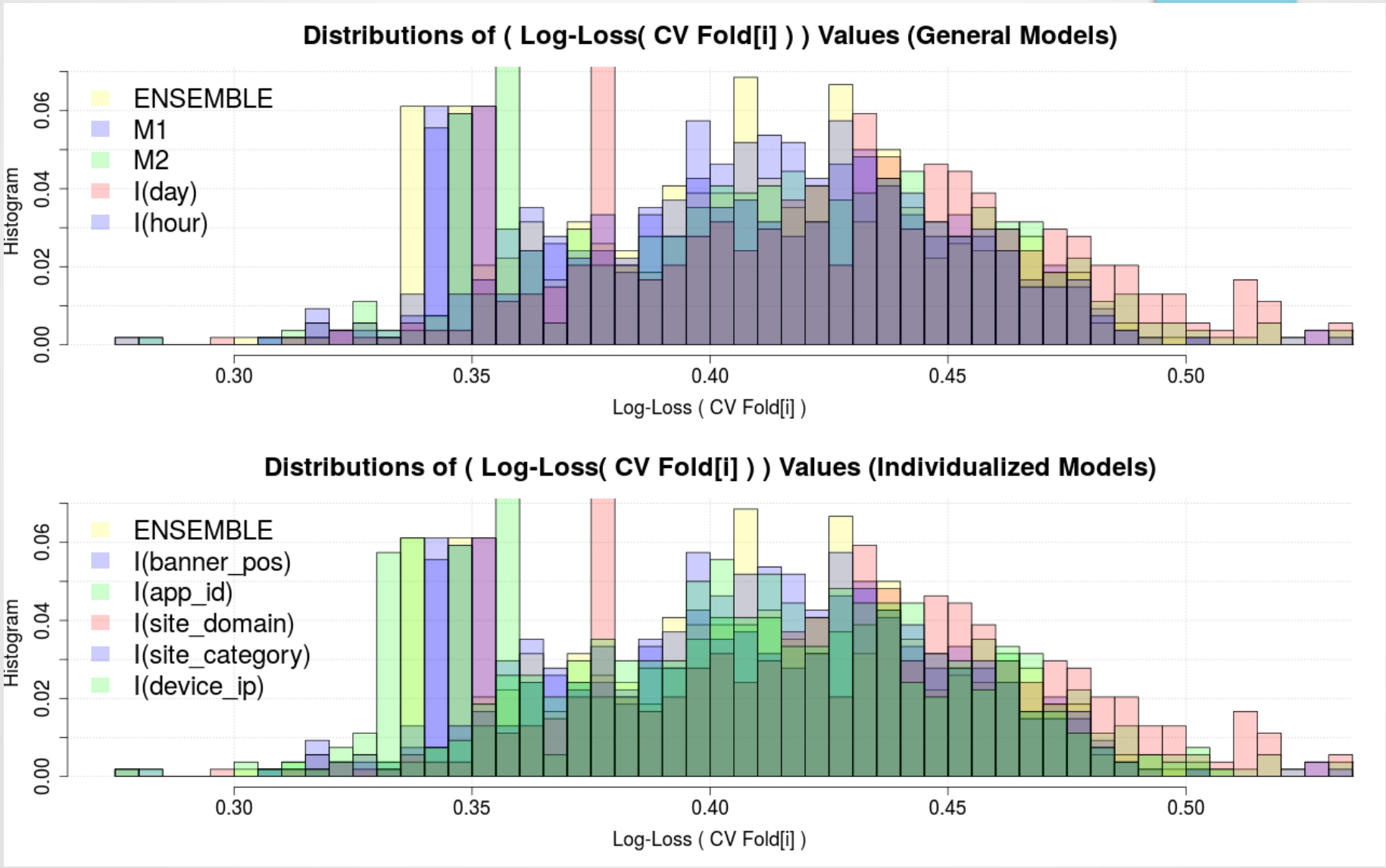




	"AVGLL"	"MINLL"	"MAXLL"
"M1<-M3"	0.404667	0.2776	0.5201
"M2<-M4"	0.405016	0.2848	0.5203
"I(H21)"	0.431589	0.2873	0.6082
"I(H18)"	0.409894	0.2792	0.5264
"I(H19)"	0.406379	0.2775	0.5275
"I(H17)"	0.411549	0.2822	0.5344
"I(H14)"	0.410192	0.2744	0.5351
"I(H15)"	0.410534	0.2795	0.5269
"I(H20)"	0.408787	0.2789	0.5333
"ENSEMB"	0.399117	0.2752	0.5169



Misc. Slides:
Histograms of Model's LogLoss (CVFold[i])



Miscellaneous Slides: Reference

- Research Papers:
  - ClickThru: Yahoo, Google, Microsoft
  - Ensembles: Boosting, Bagging
- Kaggle Avazu: https://www.kaggle.com/c/avazu-ctr-prediction
- R code: http://bitbucket.org/nelsonmanohar/machinelearning/