Ensemble methods: StackNet

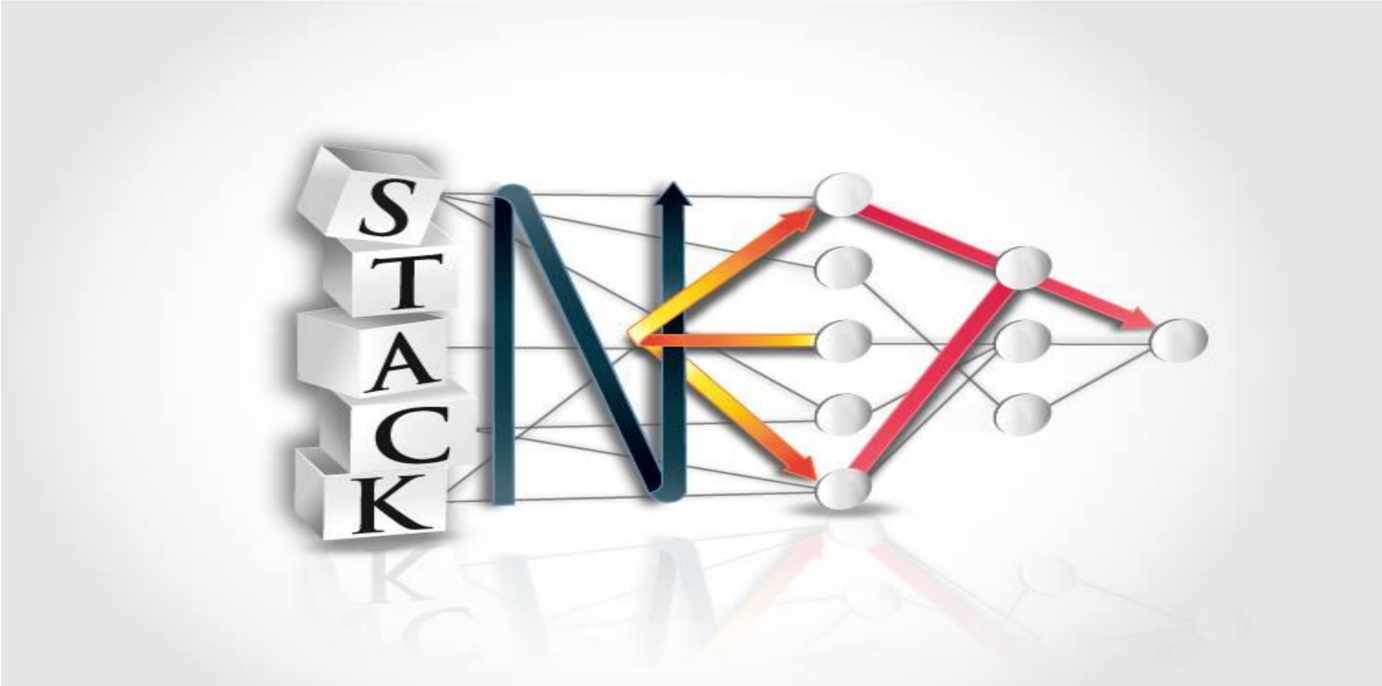
By Marios Michailidis

Examined ensemble methods

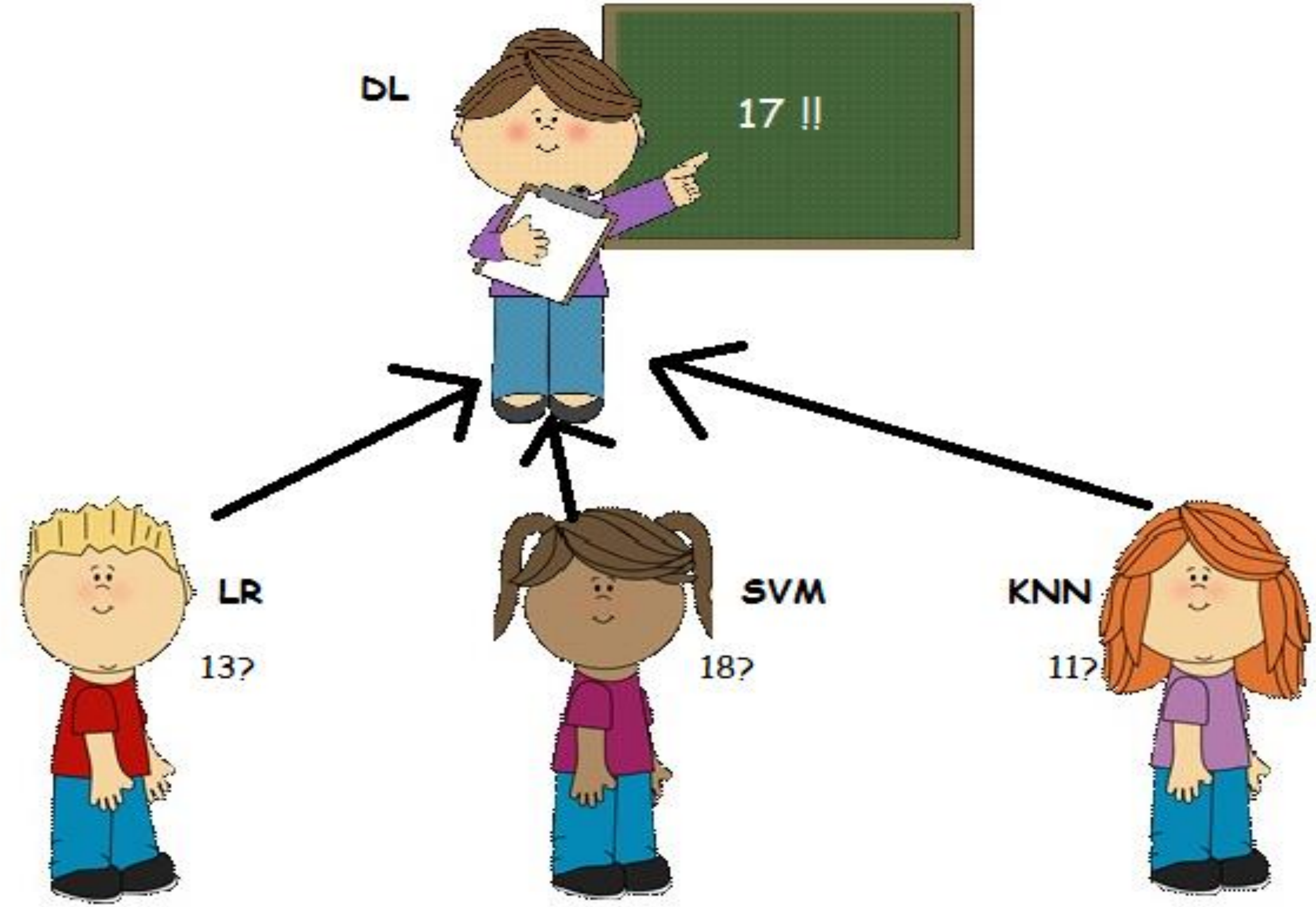
- Averaging (or blending)
- Weighted averaging
- Conditional averaging
- Bagging
- Boosting
- Stacking
- StackNet

What is StackNet

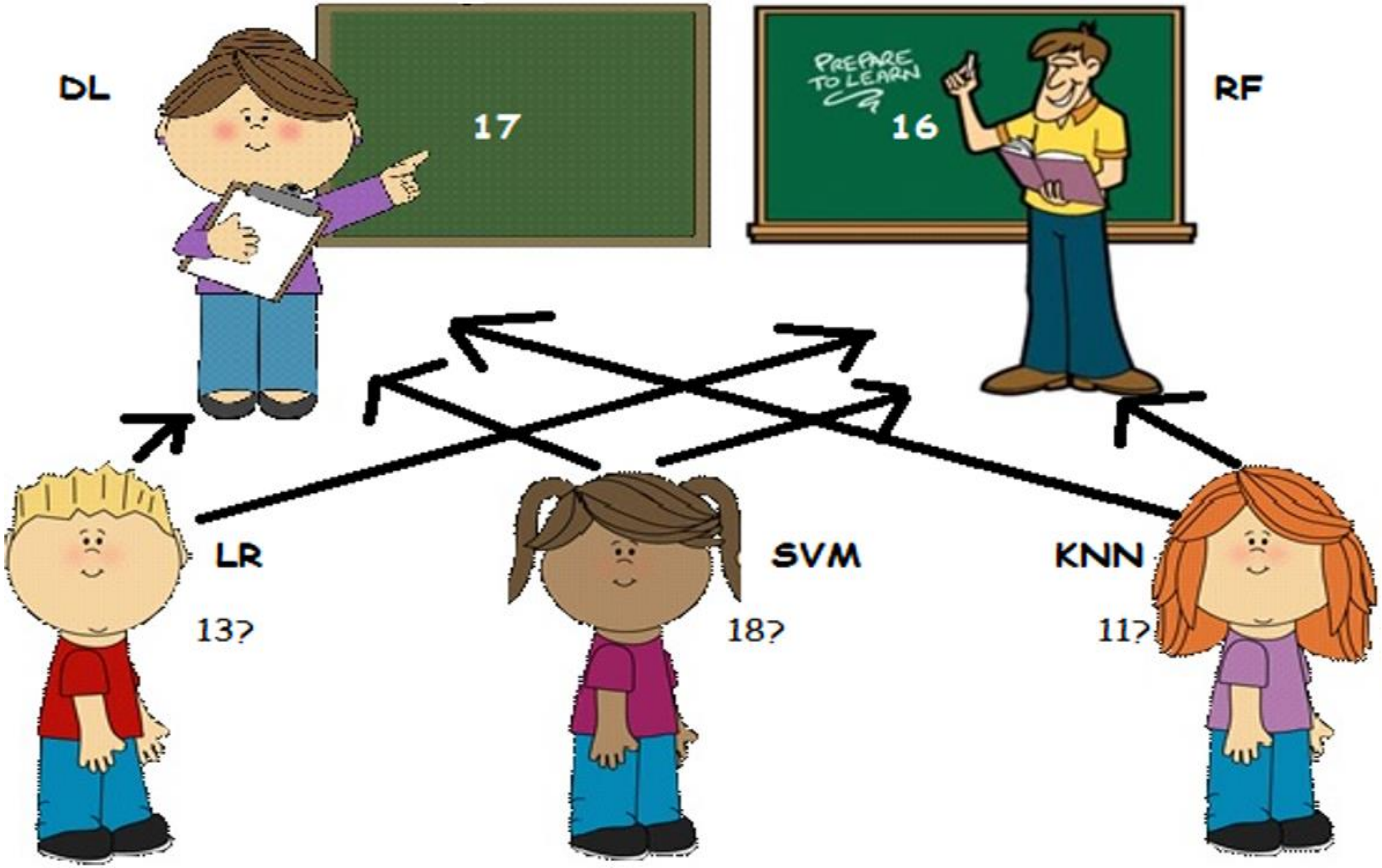
A scalable meta modelling methodology that utilizes stacking to combine multiple models in a neural network architecture of multiple levels.



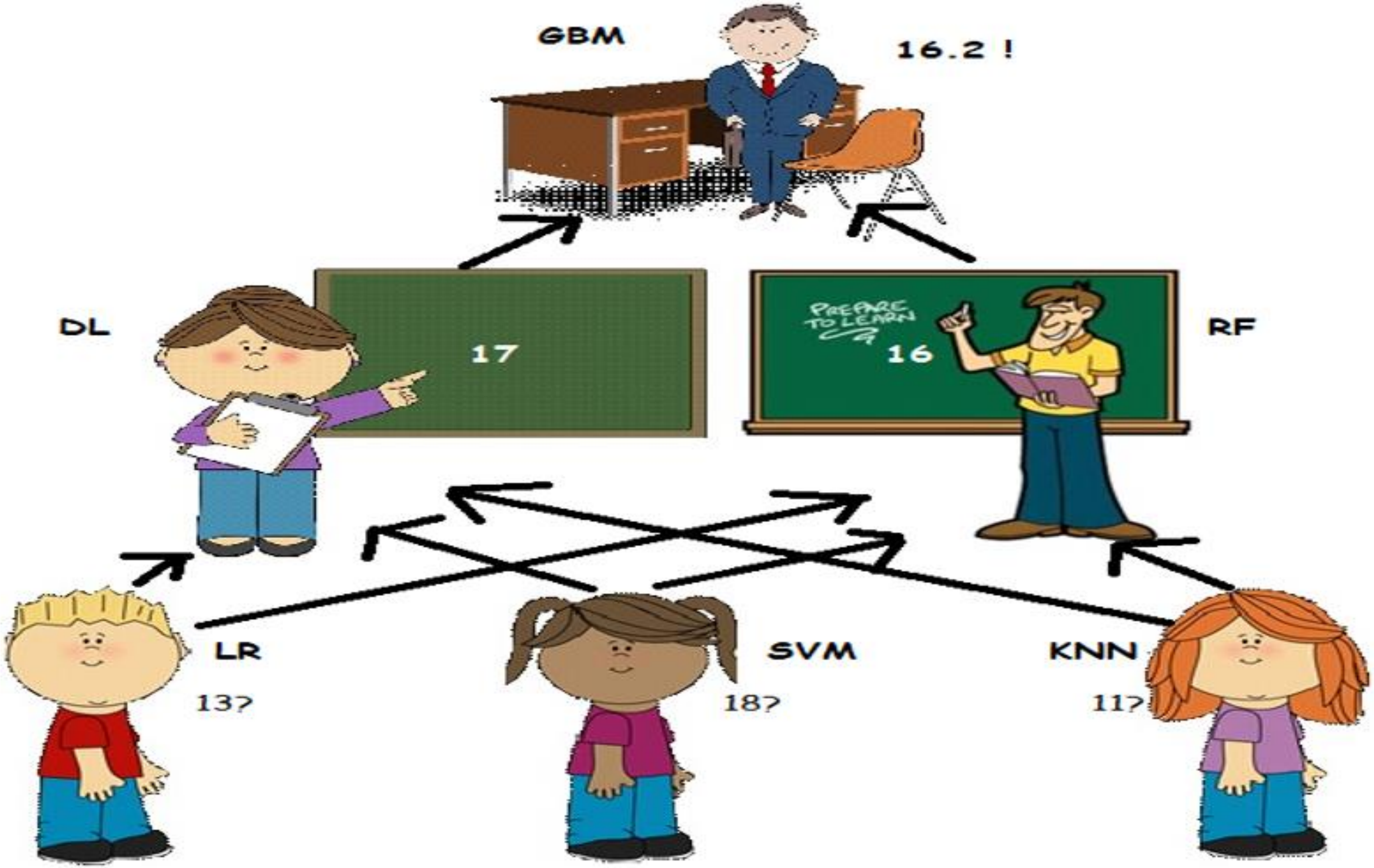
(Continuing) Naïve example



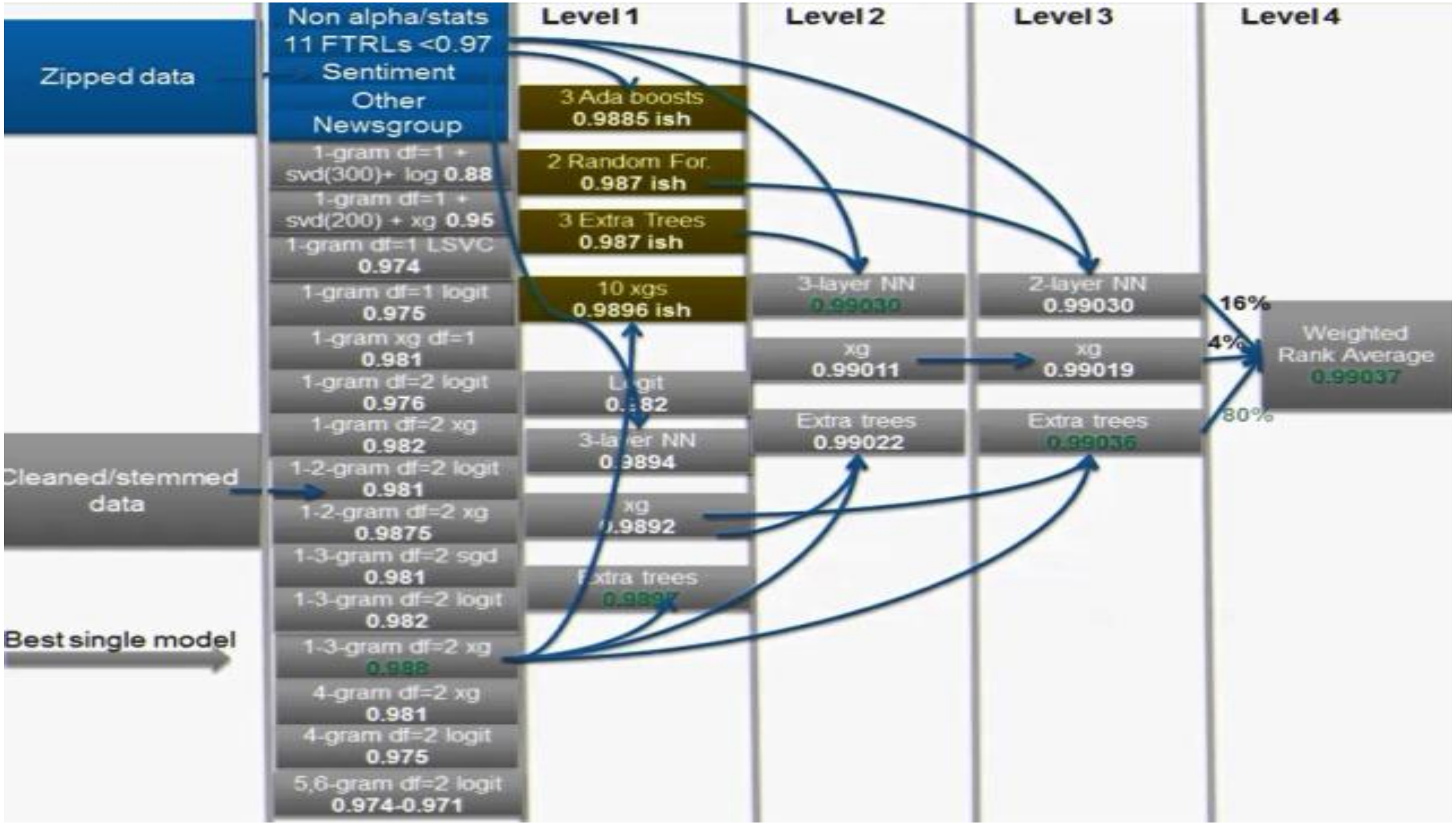
(Continuing) Naïve example



(Continuing) Naïve example



Why would this be of any use

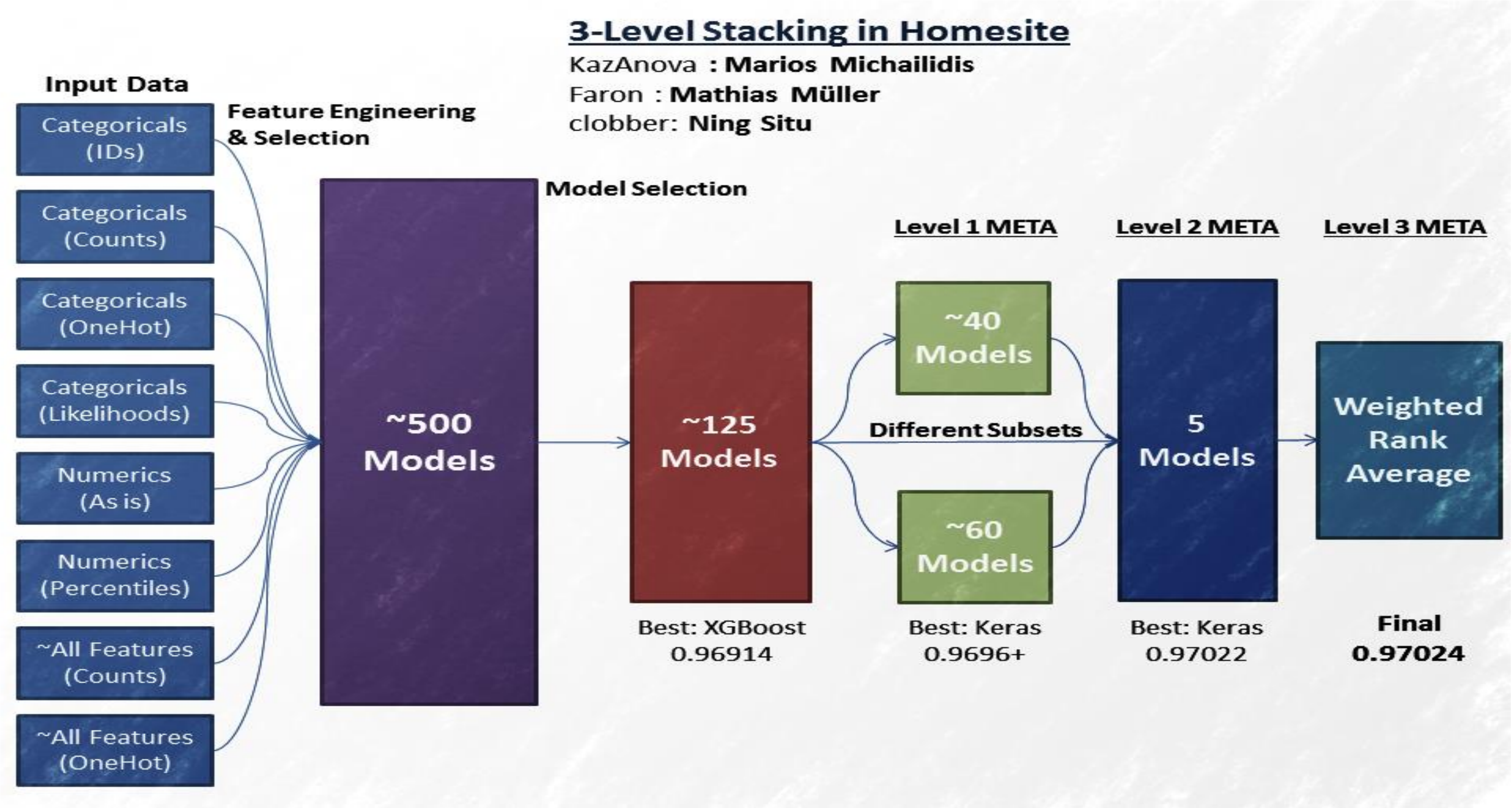


I'm giving you an example of a competition my team used, four layer of stacking, in order to win. And we used two different sources of input data.

We generated multiple models. Normally, exit boost and logistic regressions, and then we fed those into a four-layer architecture in order to get the top score. And although we could have escaped without using that fourth layer, we still need it up to level three in order to win. So you can understand the usefulness of deploying deep stacking.

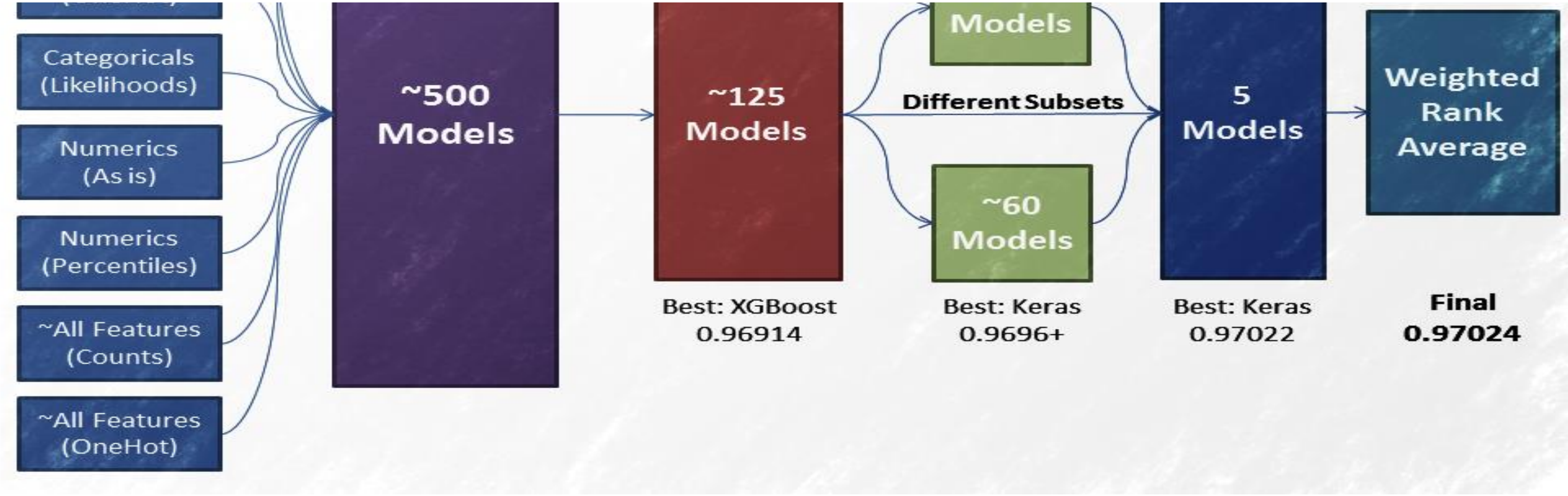
Another example is the Homesite competition organized by Homesite insurance where again, we created many different views of the data. So we had different transformations. We generated many models. We fed those models into a three-level architecture. I think we didn't need the third layer again. Probably, we could have escaped with only two levels but again, deep stacking was necessary in order to win. So there is your answer, deep stacking on multiple levels really helps you to win competitions.

Why would this be of any use



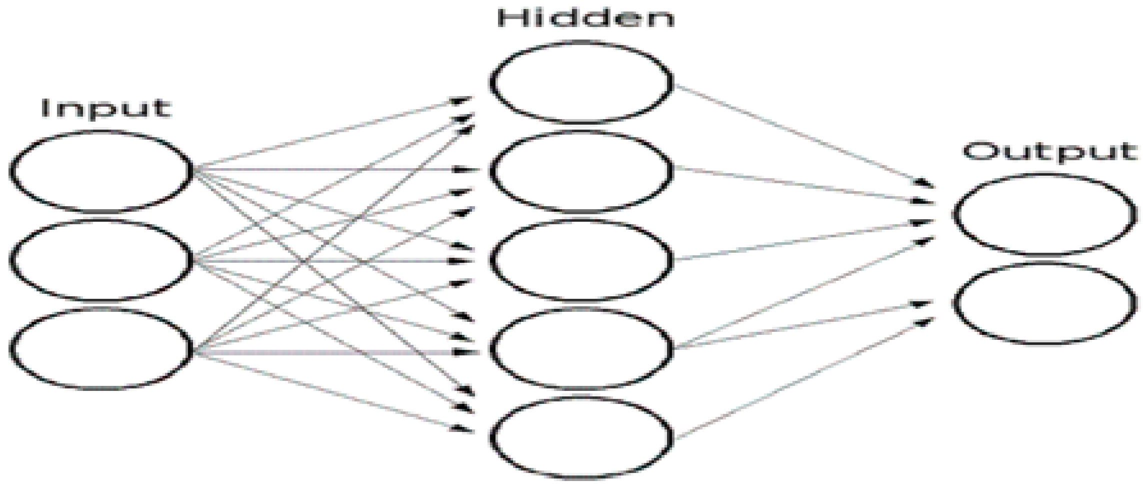
Why would this be of any use

These contests that are so close to 100% scores encourage massive, ugly ensembles consisting of old tech that's existed for many years, just to shave off those last fractions of a percent. They result in virtually no commercial value and definitely no academic value. They win the contest and that's it.



StackNet as a neural network

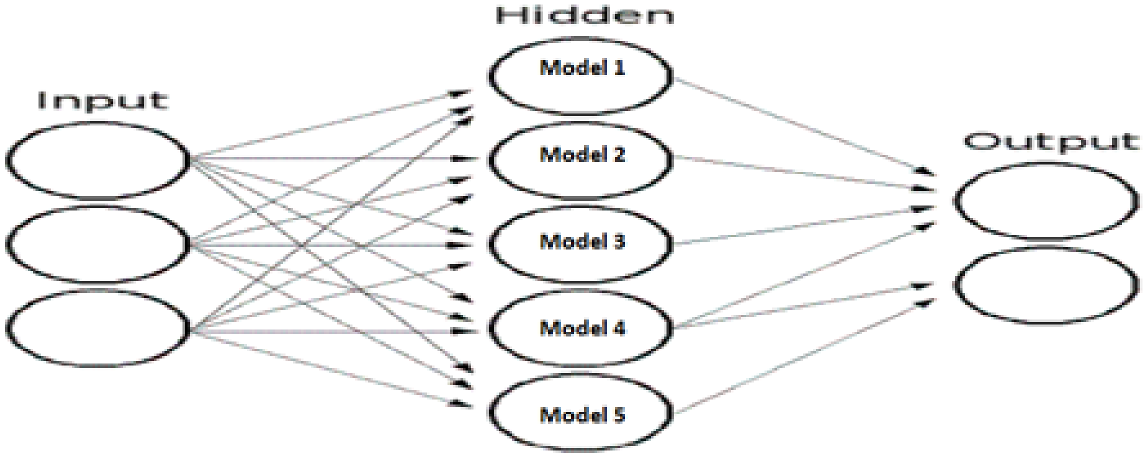
- In a neural network , every node is a **simple linear model** (like linear regression) with some non linear transformation.
- Instead of a linear model we could use **any model.**



$$f_1(x_i) = \sum_{h=1}^{H} (g_1(x_i) beta_{1h} + bias_{1h})$$

StackNet as a neural network

- In a neural network , every node is a **simple linear model** (like linear regression) with some non linear transformation.
- Instead of a linear model we could use **any model.**



$$f_1(x_i) = \sum_{h=1}^{H} (g_1(x_i) beta_{1h} + bias_{1h})$$

$$f_1(x_i, s) = g_1(x_i) s$$

How to train

- We cannot use **BP** (not all models are differentiable)
- We use **stacking** to link each model/node with target

How to train

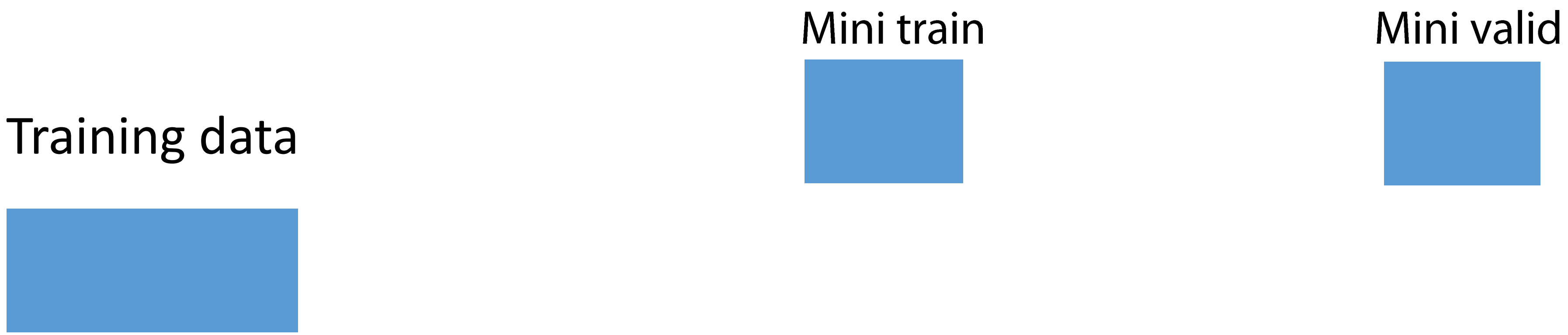
Train data

How to train

Training data

Valid data

How to train



How to train

x0	x1	x2	x3	y
0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

How to train

K=4

x0	x1	x2	x3	y
0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

How to train

K=4

x0	x1	x2	x3	y
0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

pred
0.00
0.00
0.00
0.00
0.00
0.00
0.00
0.00

How to train

Fold : 1

x0	x1	x2	x3	y
0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

pred
0.00
0.00
0.00
0.00
0.00
0.00
0.00
0.00

How to train

Fold : 1

x0	x1	x2	x3	y
0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

Train

pred
0.00
0.00
0.00
0.00
0.00
0.00
0.00
0.00

How to train

Fold : 1

x0	x1	x2	x3	y
0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

Predict

0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0

Train

0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

pred
0.00
0.00
0.00
0.00
0.00
0.00
0.00
0.00

How to train

Fold : 1

x0	x1	x2	x3	y
0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

Predict

0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0

Train

0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

pred
0.96
0.03
0.00
0.00
0.00
0.00
0.00
0.00

How to train

Fold : 2

Predict

0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0

Train

0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

x0	x1	x2	x3	y
0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

pred
0.96
0.03
0.00
0.00
0.00
0.00
0.00
0.00

How to train

Fold : 2

x0	x1	x2	x3	y
0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

Predict

0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0

Train

0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

pred
0.96
0.03
0.90
0.12
0.00
0.00
0.00
0.00

How to train

Fold : 3

x0	x1	x2	x3	y
0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

Predict

0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1

Train

0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

pred
0.96
0.03
0.90
0.12
0.00
0.00
0.00
0.00

How to train

Fold : 3

Predict

0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1

x0	x1	x2	x3	y
0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

Train

0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

pred
0.96
0.03
0.90
0.12
0.03
0.77
0.00
0.00

How to train

Fold : 4

x0	x1	x2	x3	y
0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

Predict

0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

Train

0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1

pred
0.96
0.03
0.90
0.12
0.03
0.77
0.00
0.00

How to train

Fold : 4

x0	x1	x2	x3	y
0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

Predict

0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

Train

0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1

pred
0.96
0.03
0.90
0.12
0.03
0.77
0.18
0.91

How to train

Fold : 4

x0	x1	x2	x3	y
0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

Predict

0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

Train

0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1

pred
0.96
0.03
0.90
0.12
0.03
0.77
0.18
0.91

How to train

Fold : 4

x0	x1	x2	x3	y
0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

Predict

0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

Train

0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1

test
0.43
0.03
0.90
0.12
0.03
0.77
0.18
0.91

pred
0.96
0.03
0.90
0.12
0.03
0.77
0.18
0.91

How to train

Fold : 4

x0	x1	x2	x3	y
0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1
0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

Predict

0.08	0.72	0.97	0.04	0
0.84	0.79	0.89	0.05	1

Train

0.94	0.27	0.80	0.34	1
0.02	0.22	0.17	0.84	0
0.83	0.11	0.23	0.42	1
0.74	0.26	0.03	0.41	0
0.08	0.29	0.76	0.37	0
0.71	0.76	0.43	0.95	1

test
0.43
0.03
0.90
0.12
0.03
0.77
0.18
0.91

pred	pred
0.96	0.00
0.03	0.00
0.90	0.00
0.12	0.00
0.03	0.00
0.77	0.00
0.18	0.00
0.91	0.00

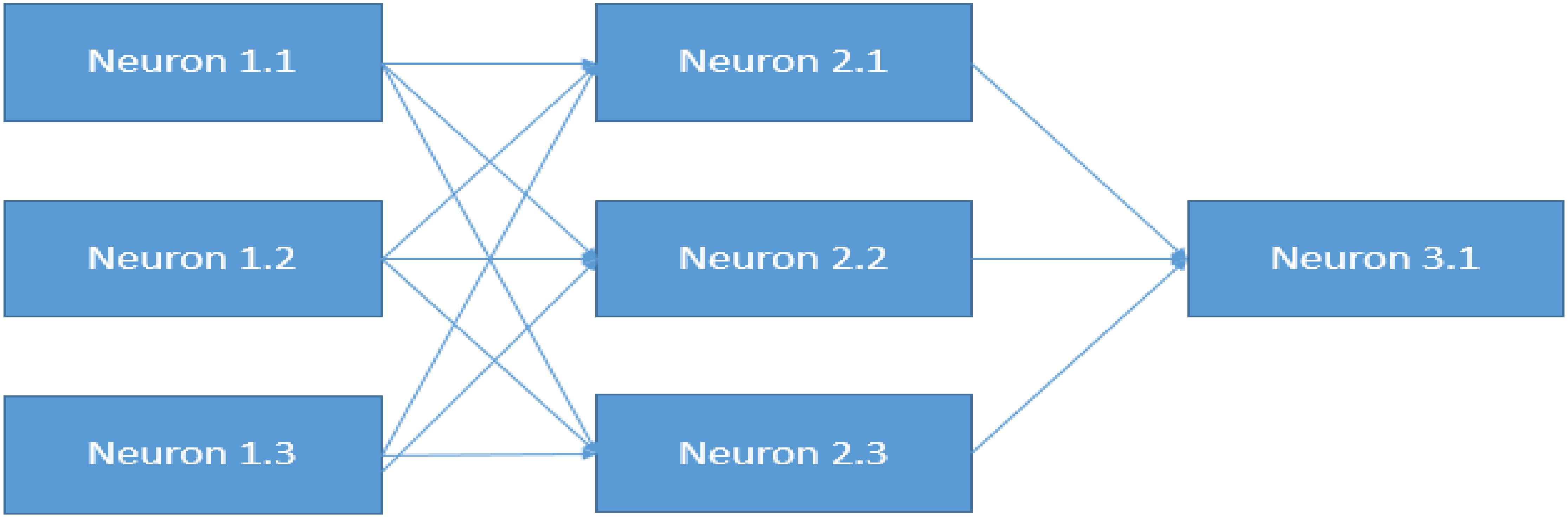
How to train

- We cannot use **BP** (not all models are differentiable)
- We use **stacking** to link each model/node with target
- To extend to many levels, we can use a **Kfold** paradigm

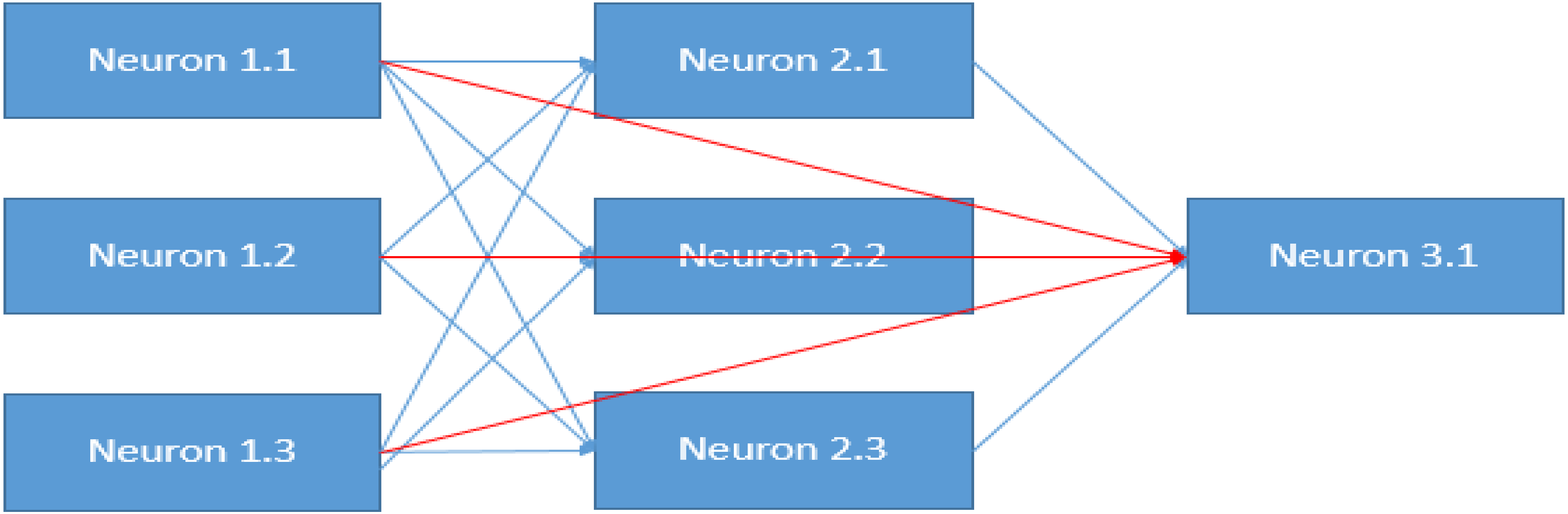
How to train

- We cannot use **BP** (not all models are differentiable)
- We use **stacking** to link each model/node with target
- To extend to many levels, we can use a **Kfold** paradigm
- No epochs – different connections instead.

How to train


Neuron 1.1
Neuron 1.2
Neuron 1.3
Neuron 2.1
Neuron 2.2
Neuron 2.3
Neuron 3.1

How to train


Neuron 1.1
Neuron 1.2
Neuron 1.3
Neuron 2.1
Neuron 2.2
Neuron 2.3
Neuron 3.1

1^{st} level tips

- Diversity based on algorithms:
  - 2-3 gradient boosted trees (lightgb, xgboost, H2O, catboost)
  - 2-3 Neural nets (keras, pytorch)
  - 1-2 ExtraTrees/Random Forest (sklearn)
  - 1-2 linear models as in logistic/ridge regression, linear svm (sklearn)
  - 1-2 knn models (sklearn)
  - 1 Factorization machine (libfm)
  - 1 svm with nonlinear kernel if size/memory allows (sklearn)
- Diversity based on input data:
  - Categorical features: One hot, label encoding, target encoding, frequency
  - Numerical features: outliers, binning, derivatives, percentiles, scaling
  - Interactions : col1*/+-col2, groupby, unsupervised

Subsequent level tips

- Simpler (or shallower) Algorithms:
  - ❑ gradient boosted trees with small depth (like 2 or 3)
  - ❑ Linear models with high regularization
  - ❑ Extra Trees
  - ❑ Shallow networks (as in 1 hidden layer)
  - ❑ knn with BrayCurtis Distance
  - ❑ Brute forcing a search for best linear weights based on cv
- Feature engineering:
  - ❑ pairwise differences between meta features
  - ❑ row-wise statistics like averages or stds
  - ❑Standard feature selection techniques
- For every 7.5 models in previous level we add 1 in meta (empirical)
- Be mindful of target leakage

Software for Stacking

- StackNet (https://github.com/kaz-Anova/StackNet)
- Stacked ensembles from H2O
- Xcessiv (https://github.com/reiinakano/xcessiv)

Tips about StackNet (Software)

- It supports many prominent tools (xgboost, lightgbm, H2O, keras…)
- Can run classifiers in regression and vice versa.
- It has several top 10s in competitions.

Tips about StackNet (Software)

Your submission scored 0.92256.

Submission and Description	Private Score	Public Score	Use for Final Score
sub_70_30.7z 6 hours ago by Μαριος Μιχαηλιδης KazAnova *add submission details*	0.91923	0.92256	☐

#	△pub	Team Name ✱ in the money	Kernel	Team Members	Score	Entries	Last
1	▲ 2	✱ Paul Duan & BS Man			0.92360	122	4y
2	▼ 1	✱ Owen Zhang			0.92273	54	4y
3	▲ 1	✱ Dmitry&Leustagos			0.92255	110	4y
4	▲ 1	Tim			0.92189	24	4y
5	▲ 2	Chaotic Experiments			0.92154	77	4y
6	▲ 2	Murashka			0.92106	124	4y
7	▲ 3	Alexander Larko			0.92105	102	4y
8	▼ 6	Gxav			0.92013	34	4y
9	▼ 3	beginnersLuck			0.91961	76	4y
10	▲ 2	IzuiT			0.91942	32	4y

Tips about StackNet (Software)

- It supports many prominent tools (xgboost, lightgbm, H2O, keras…)
- Can run classifiers in regression and vice versa.
- It has several top 10s in competitions.
- The parameters' section.

Tips about StackNet (Software)

XgboostClassifier

The original parameters can be found here

```
XgboostClassifier booster:gbtree num_round:1000 eta:0.005 max_leaves:0 gamma:1. max_depth:5 min_child_weight:1.0 subs
```

Parameter	Explanation
scale_pos_weight	used for imbalanced classes(double)
num_round	Number of estimators to build (int) . **This is important.**
max_leaves	Maximum leaves in a tree (int).
eta	Penalty applied to each estimator. Needs to be between 0 and 1 (double). **This is important.**
max_depth	Maximum depth of the tree (int). **This is important.**
subsample	Proportion of observations to consider (double). **This is important.**
colsample_bylevel	Proportion of columns (features) to consider in each level (double).
colsample_bytree	Proportion of columns (features) to consider in each Tree (double) **This is important.**
max_delta_step	controls optimization step (double).

Before we say goodbye…

- Apply what you have learnt (in competitions).
- It takes some time to adjust.
- Always save your code and re-use it
- Seek collaborations
- Read forums/kernels


thank you
danke
рахмат
謝謝
ngiyabonga
teşekkür ederim
gracias
Баярлалаа
спасибо
faafetai lava
mersi
kia ora
barka
welalin
tack
spas
dank je
misaotra
matondo
paldies
grazzi
mahalo
tapadh leat
хвала
asante
manana
obrigada
tenki
vinaka
спасибі
blagodaram
kiitos
dankie
nanni
nandri
dhanyavad
maururu
hvala
bedankt
bayarlalaa
gràçie
köszönöm
enkosi
dziękuję
dankon
aćiū
akun
djiere dieuf
tau
mochchakkeram
murakoze
mamnun
дякую
chnorakaloutioun
gratias ago
gràcies
sulpáy
taiku
go raibh maith agat
chokrane
sobodi
dĕkuji
obrigado
sagolun
sukriya
kop khun krap
arigatô
takk
dakujem
trugarez
mèsi
didi madloba
najis tuke
ありがとう
tanemirt
grazie
rahmet
terima kasih
kam sah hamnida
rahmat
diolch
dhanyavadagalu
shukriya
mercé
мерси
তোমাকে ধন্যবাদ
감사합니다
xièxie
ευχαριστώ
merci

Rank	Tier	User		Medals			Points
1		You	joined a year ago	999	0	0	994,882
2		Stanislav Semenov	joined 4 years ago	28	9	0	190,356
3		Μαριος Μιχαηλιδης KazAnova	joined 4 years ago	26	23	21	168,976
4		Faron	joined 3 years ago	14	4	3	132,862
5		Eureka	joined 4 years ago	16	13	3	131,759
6		raddar	joined 2 years ago	9	6	3	119,285
7		idle_speculation	joined 4 years ago	7	8	6	116,367
8		weiwei	joined a year ago	5	3	1	108,836
9		bestfitting	joined a year ago	5	3	0	107,497
10		Silogram	joined 5 years ago	10	24	9	97,850
11		utility	joined 3 years ago	13	7	3	95,855