Unobservable model errors, metalearning

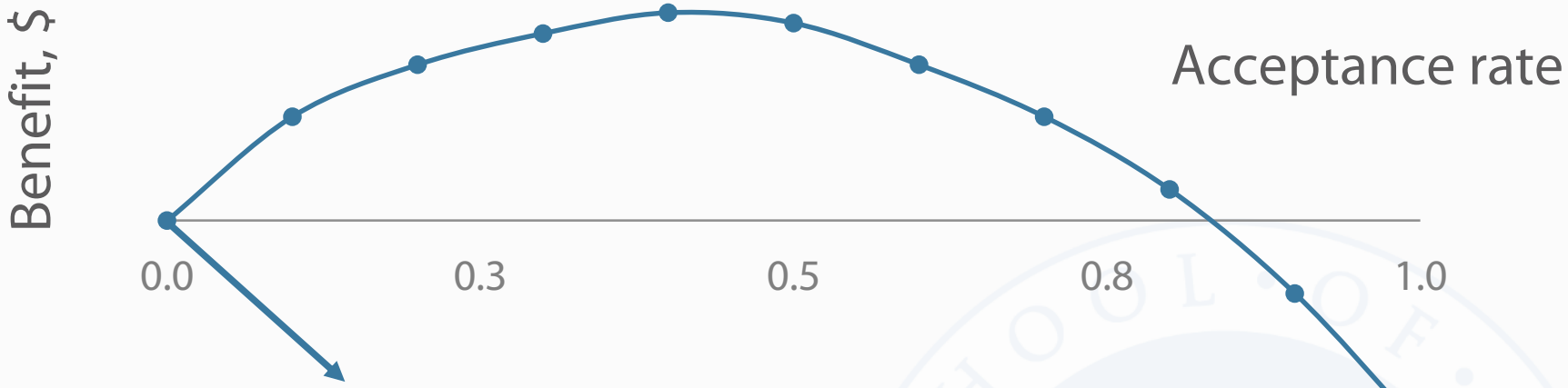
Course roadmap

1. Project valuation: valuation metrics, planning and rules
2. Model quality and decision making. Benefit curve
3. Estimating model risk discounts
4. A/B testing and financial result verification
5. Unobservable model errors, metalearning
   - Unobservable model errors
   - Control groups in A/B testing
   - Reject inference and semi-supervised learning
   - Expected benefit in case of unobservable FP and FN
   - Metalearning. Reconstructing target event using all available information

Unobservable model errors

What is the problem?

Remember Week 2 about benefit curves construction?

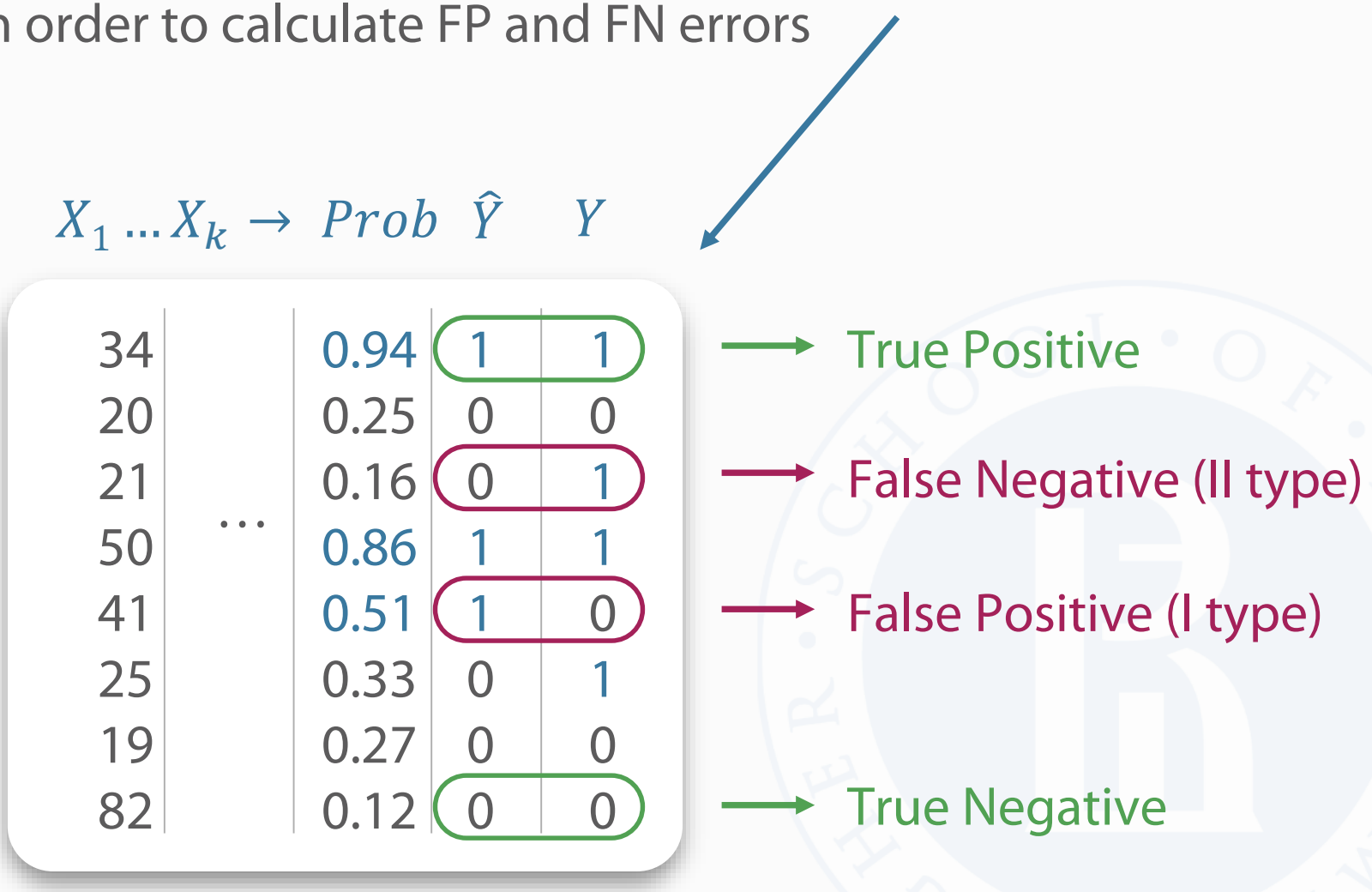


Starting point:
Threshold level $a = 0$
Acceptance rate $c = 0$

Reject if *Prob* > 0,
i.e. no loss and no gain
Zero benefit

What is the problem?

The key assumption was having **representative** historical data in order to calculate FP and FN errors



What is the problem?

Available historical data sometimes can be **not representative**:

- There is a considerable number of rejected clients
- A/B testing is too expensive

Why?

Rejected clients

Every running business-process affects the historical data

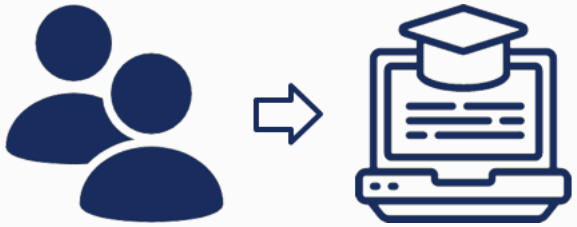
$X_1 \dots X_k$

Client

Rejected clients

Every running business-process affects the historical data

$X_1 \dots X_k$ $Prob$

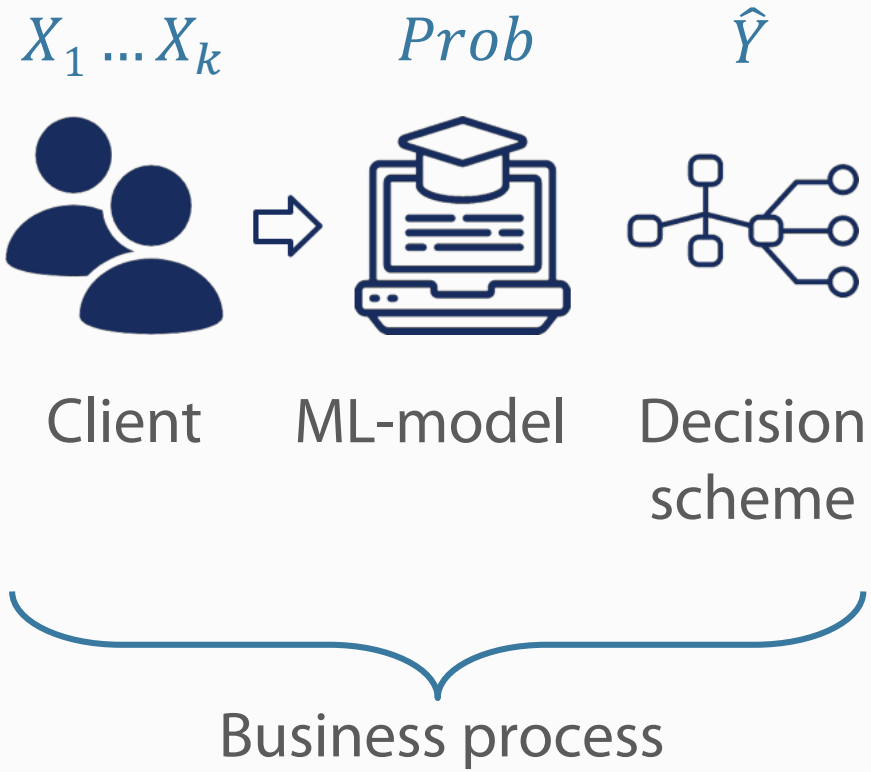


Client ML-model

thenounproject.com

Rejected clients

Every running business-process affects the historical data



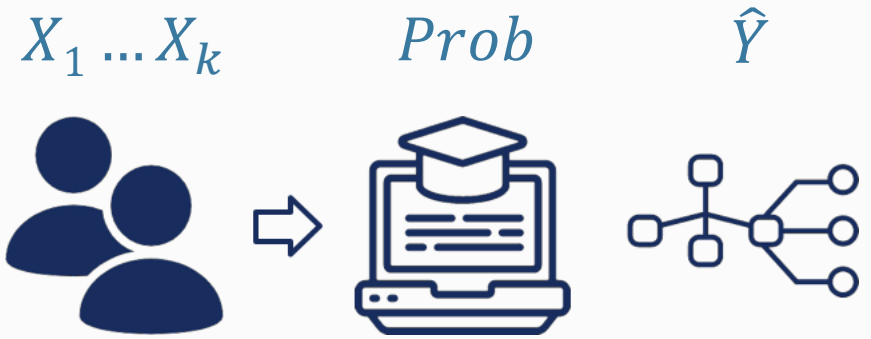
Accept
(i.e. grant loan/
make an offer/
provide discount

Reject
(i.e. reject loan/
make no offer/
withdraw discount

thenounproject.com

Rejected clients

Every running business-process affects the historical data



Accept
(i.e. grant loan/ make an offer/ provide discount

Y
(target event)
is known
(0/1)

Reject
(i.e. reject loan/ make no offer/ withdraw discount

unknown
(?)

Rejected clients. True drama

1. There is always a decision rule (expert or model based) within business process

Rejected clients. True drama

1. There is always a decision rule (expert or model based) within business process
2. If you build a new ML model you can straightforwardly use only that part of client data which was accepted by previously settled decision rules

Rejected clients. True drama

1. There is always a decision rule (expert or model based) within business process
2. If you build a new ML model you can straightforwardly use only that part of client data which was accepted by previously settled decision rules
3. At the same time, when your new ML model is ready to be deployed, it will be implemented for the whole client base

Rejected clients. True drama

1. There is always a decision rule (expert or model based) within business process
2. If you build a new ML model you can straightforwardly use only that part of client data which was accepted by previously settled decision rules
3. At the same time, when your new ML model is ready to be deployed, it will be implemented for the whole client base
4. **Here is the drama**: you always train your model on ***biased*** data

Rejected clients. True drama

1. There is always a decision rule (expert or model based) within business process
2. If you build a new ML model you can straightforwardly use only that part of client data which was accepted by previously settled decision rules
3. At the same time, when your new ML model is ready to be deployed, it will be implemented for the whole client base
4. **Here is the drama**: you always train your model on ***biased*** data

The same holds true if you want just to evaluate your new model performance in production: you cannot observe real FP and TN errors

Rejected clients. View data

There can be a considerable part of historical data which cannot be used to train ML model straightforwardly

$X_1 \dots X_k \rightarrow Prob \quad \hat{Y} \quad Y$

34		0.94	1	1
20		0.25	0	0
21		0.16	0	1
50	…	0.86	1	1
41		0.51	1	0
25		0.33	0	1
19		0.27	0	0
82		0.12	0	0
15		0.87	0	?
19		0.83	1	?
27		0.75	0	?
11		0.99	1	?

Rejected clients. View data

There can be a considerable part of historical data which cannot be used to train ML model straightforwardly

$X_1 \dots X_k \rightarrow Prob \quad \hat{Y} \quad Y$

34		0.94	1	1
20		0.25	0	0
21		0.16	0	1
50	…	0.86	1	1
41		0.51	1	0
25		0.33	0	1
19		0.27	0	0
82		0.12	0	0
15		0.87	0	?
19		0.83	1	?
27		0.75	0	?
11		0.99	1	?

$Prob$ and $\hat{Y}$ are outputs of new ML model

Accepted clients:
(based on prev. decision rules)
full info about target event

Rejected clients:
no info about target event

Rejected clients. View data

The historical data that can be used for model training is subject to bias

$X_1 \dots X_k \rightarrow Prob \quad \hat{Y} \quad Y$

34		0.94	1	1
20		0.25	0	0
21		0.16	0	1
50	…	0.86	1	1
41		0.51	1	0
25		0.33	0	1
19		0.27	0	0
82		0.12	0	0
15		0.87	0	?
19		0.83	1	?
27		0.75	0	?
11		0.99	1	?

Biased data

Accepted clients:
(based on prev. decision rules)
full info about target event

Rejected clients:
no info about target event

Rejected clients. View data

We cannot calculate real FP and FN errors due to unknown target event for rejected clients

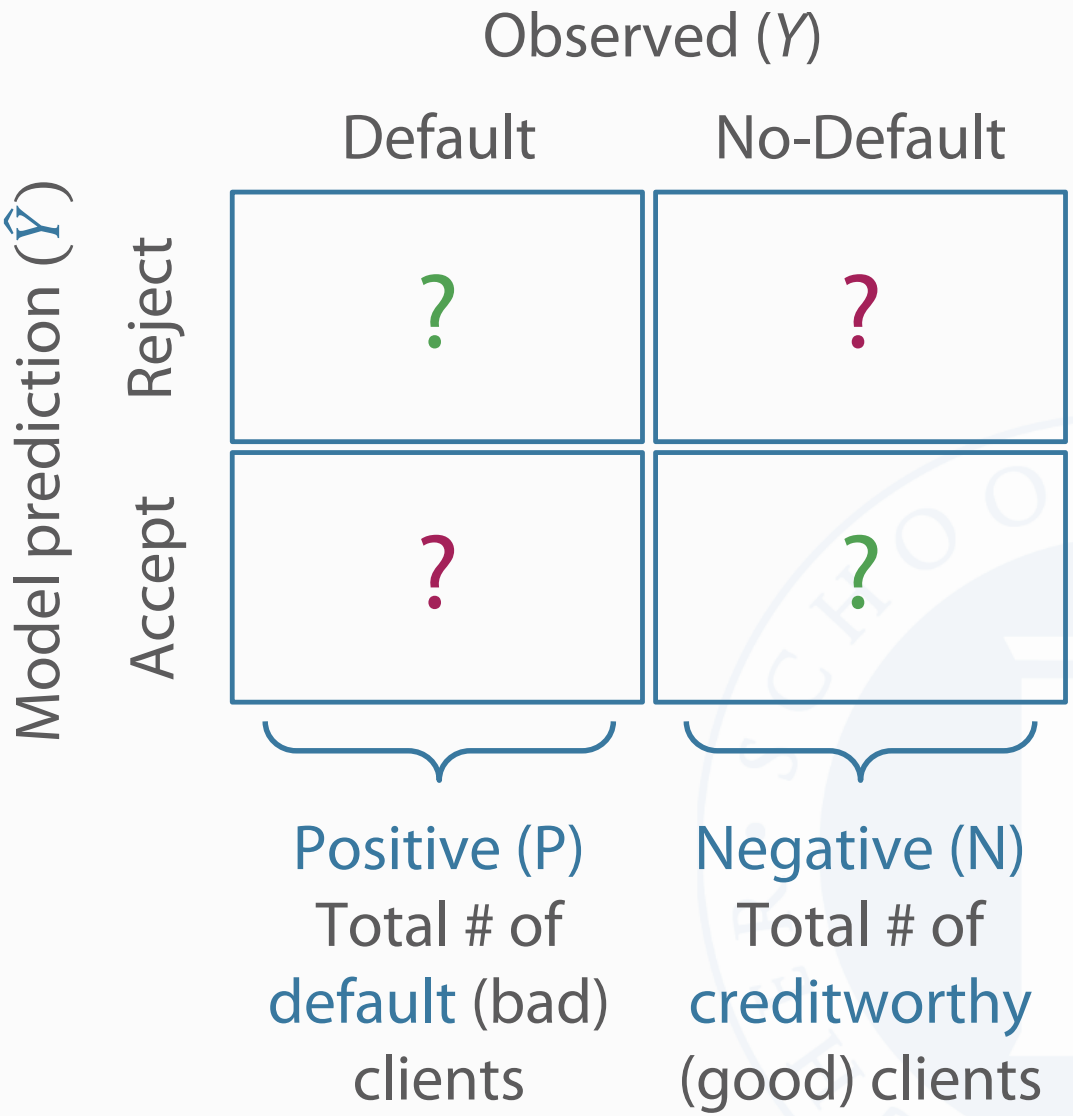
$X_1 \dots X_k \rightarrow Prob \quad \hat{Y} \quad Y$

34		0.94	1	1
20		0.25	0	0
21		0.16	0	1
50	…	0.86	1	1
41		0.51	1	0
25		0.33	0	1
19		0.27	0	0
82		0.12	0	0
15		0.87	0	?
19		0.83	1	?
27		0.75	0	?
11		0.99	1	?

No real FP and FN – no benefit curve!

FP, FN, TP, TN?

Rejected clients. Model prediction errors matrix



Wrap-up

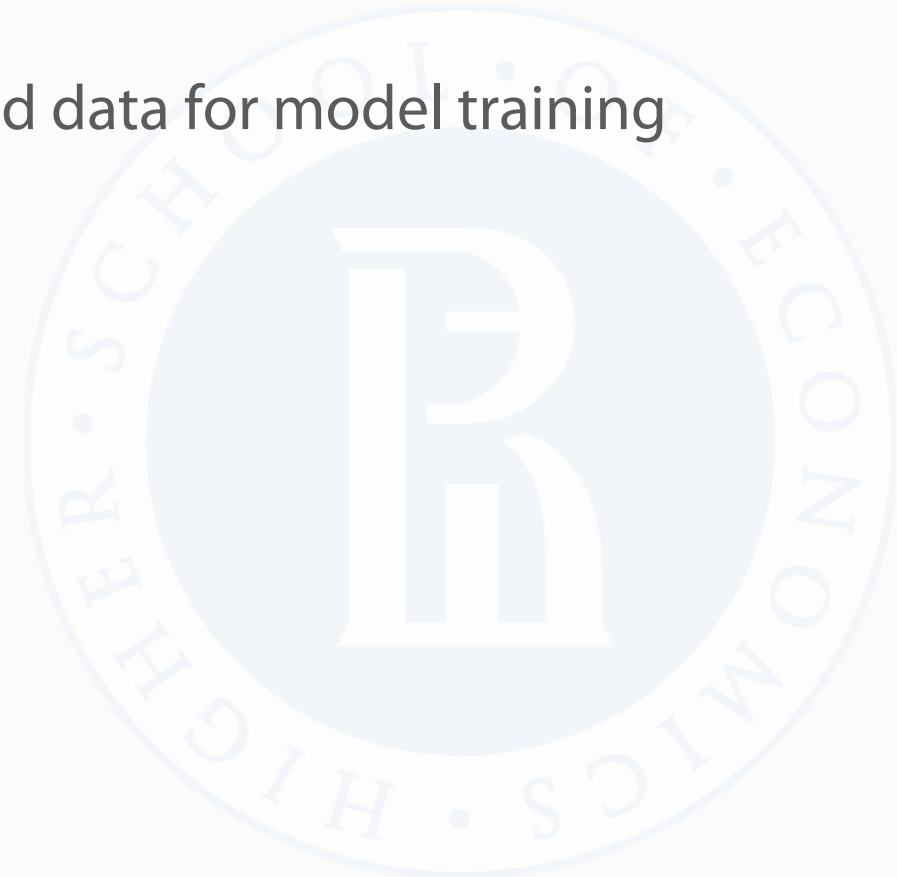
1. Historical data is always generated by some non-random business process

Wrap-up

1. Historical data is always generated by some non-random business process
2. Either expert or model-based decisions within business process can lead to unobservable target event for some clients

Wrap-up

1. Historical data is always generated by some non-random business process
2. Either expert or model-based decisions within business process can lead to unobservable target event for some clients
3. Therefore, we dealt with biased data for model training and performance evaluation



Control groups in A/B testing

Why A/B testing can help?

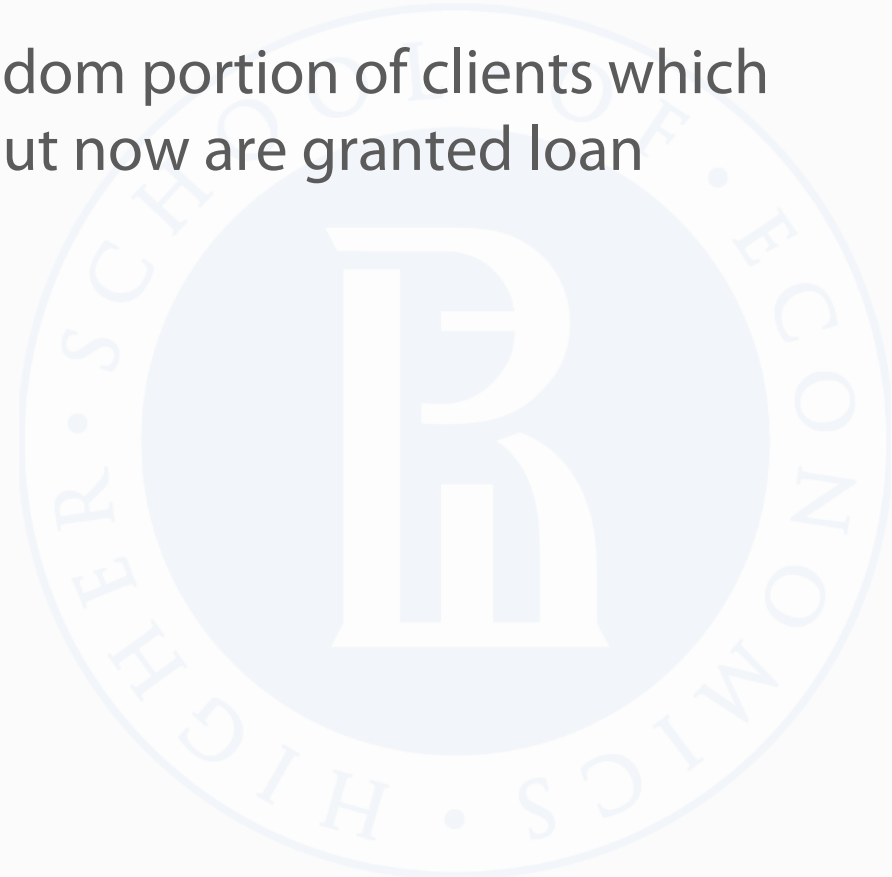
1. Usually A/B implies some control group
2. Control group allows us to gain representative data about clients

Why A/B testing can help?

1. Usually A/B implies some control group
2. Control group allows us to gain representative data about clients

Control group is designed:

- In credit scoring: as a small random portion of clients which otherwise would be rejected but now are granted loan



Why A/B testing can help?

1. Usually A/B implies some control group
2. Control group allows us to gain representative data about clients

Control group is designed:

- In credit scoring: as a small random portion of clients which otherwise would be rejected but now are granted loan
- In response models: as a small random portion of clients which are (or are not) offered the product

Why A/B testing can help?

1. Usually A/B implies some control group
2. Control group allows us to gain representative data about clients

Control group is designed:

- In credit scoring: as a small random portion of clients which otherwise would be rejected but now are granted loan
- In response models: as a small random portion of clients which are (or are not) offered the product
- In collection: as a small random portion of clients which are communicated within random channels, etc.

Regardless of model predictions!

Rejected clients. Control group

As soon as control group is random, it is representative with respect to rejected clients

$X_1 \dots X_k \rightarrow$		$Prob$	$\hat{Y}$	Y
34		0.94	1	1
20		0.25	0	0
21		0.16	0	1
50	…	0.86	1	1
41		0.51	1	0
25		0.33	0	1
19		0.27	0	0
82		0.12	0	0
15		0.87	0	1
19		0.83	1	1
27		0.75	0	?
11		0.99	1	?

Control group is usually small

Rejected clients. Control group

Now we can append historical data and control group results and calculate FP and FN on representative data

$X_1 \dots X_k \rightarrow$		$Prob$	$\hat{Y}$	Y	w
34		0.94	1	1	1
20		0.25	0	0	1
21		0.16	0	1	1
50	…	0.86	1	1	1
41		0.51	1	0	1
25		0.33	0	1	1
19		0.27	0	0	1
82		0.12	0	0	1
15		0.87	0	1	2
19		0.83	1	1	2
27		0.75	0	?	-
11		0.99	1	?	-

But we also have to assign some larger weight w to control group

Rejected clients. FP and FN calculation

How should we define weight w?

For initial clients with known target variable weight should be 1.

For clients within control group it should be a ratio between total number of clients with unknown target event and number of clients in control group.

$$w = \frac{N_{rejected}}{N_{control\_group}}$$

Therefore, $\sum_{i=1}^{N} w_i = N$, the total number of clients in our base

Rejected clients. FP and FN calculation

Now, we can derive the formulas for FP and FN rates in model prediction errors matrix:

$$FPR = \frac{\sum_{i=1}^{N} w_i \cdot I\{\hat{Y}_i = 1\} \cdot I\{Y_i = 0\}}{\sum_{i=1}^{N} w_i \cdot I\{Y_i = 0\}}$$

$$FNR = \frac{\sum_{i=1}^{N} w_i \cdot I\{\hat{Y}_i = 0\} \cdot I\{Y_i = 1\}}{\sum_{i=1}^{N} w_i \cdot I\{Y_i = 1\}}$$

As soon as, we calculate FP and FN, we can build benefit curves

Rejected clients. FP and FN calculation

Resume to our example:

$X_1 \dots X_k \rightarrow$		$Prob$	$\hat{Y}$	Y	w
34		0.94	1	1	1
20		0.25	0	0	1
21		0.16	0	1	1
50	…	0.86	1	1	1
41		0.51	1	0	1
25		0.33	0	1	1
19		0.27	0	0	1
82		0.12	0	0	1
15		0.87	0	1	2
19		0.83	1	1	2
27		0.75	0	?	-
11		0.99	1	?	-

$$w = \frac{N_{rejected}}{N_{control\_group}} = \frac{4}{2} = 2$$

$$FPR = \frac{1}{4} = 25\%$$

$$FNR = \frac{1+1+2}{4+2+2} = 50\%$$

Problems with A/B testing

A/B testing sometimes is not affordable:

1. For certain business processes too expensive.
   Credit scoring — high FP error cost leads to severe losses when running a control group

2. IT-infrastructure may not allow one to perform A/B testing easily and on demand

Wrap-up

1. Designing control groups appropriately may allow one to overcome bias in historical data
2. Weights of observations should be adjusted when calculating FP and FN using control group
3. Running control groups, however, maybe too expensive for some business processes

Reject inference and semi-supervised learning

Still open question.

There is no straightforward technique to handle unobserved FP.

However, there are considerable advances in ML studies in order to cope with it:

1. **Ratsaby J., Venkatesh S. S. (1995)**
   Learning from a mixture of labeled and unlabeled examples with parametric side information

2. **Zhu Xiaojin (2008)**
   Semi-supervised learning literature survey

3. **Kozodoi N. et al (2019)**
   Shallow Self-Learning for Reject Inference in Credit Scoring

Reject inference. Basic idea

Data with known target event

Data with target event unknown

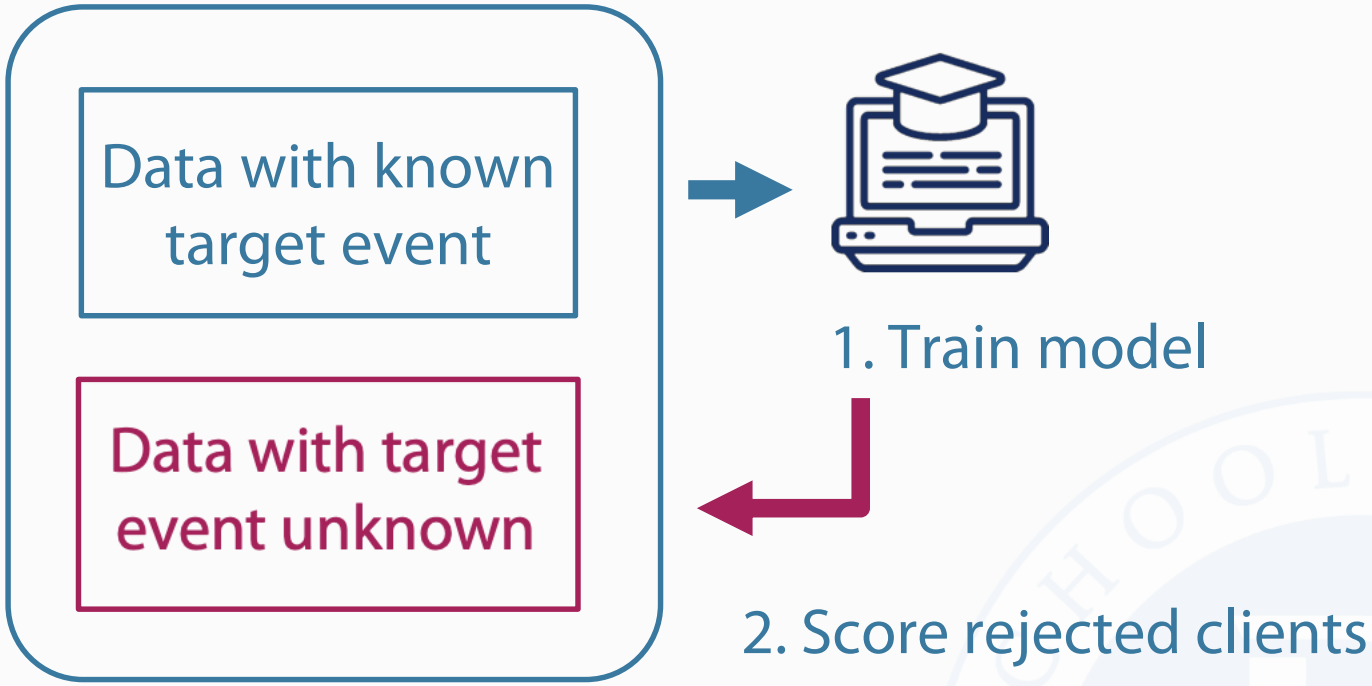
Reject inference. Basic idea

Data with known target event

1. Train model

Data with target event unknown

Reject inference. Basic idea



thenounproject.com

Reject inference. Basic idea

1. Train model

2. Score rejected clients

3. Simulate target event for rejected clients based on Prob at step 2

Reject inference. Basic idea

1. Train model

2. Score rejected clients

3. Simulate target event for rejected clients based on Prob at step 2

4. Retrain model on united dataset

thenounproject.com

Reject inference. Basic idea

1. Train model

2. Score rejected clients

3. Simulate target event for rejected clients based on Prob at step 2

4. Retrain model on united dataset

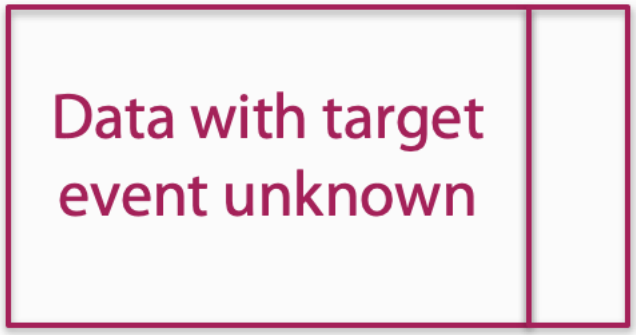
Repeat 1-4 until models at Step 1 and Step 4 are close in terms of parameters value

thenounproject.com

Reject inference. Target event simulation

At step 3 we simulate target event for rejected clients

How? $Prob_i$



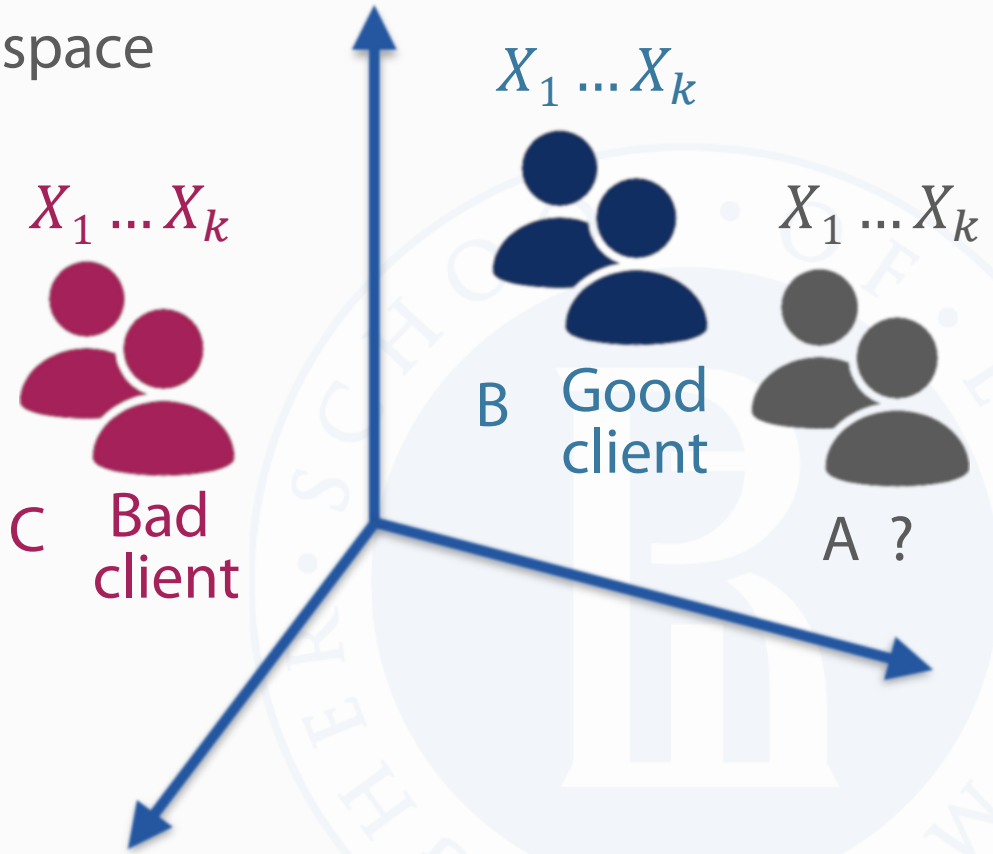
Different methods:

- Initialize random binary variable with success probability equal to $Prob_i$ for each client
- Or, duplicate the rows and assign each target equal 0 and 1, also assign weight equal to $Prob_i$ and $(1 - Prob_i)$ for every duplicated row

Semi-supervised learning. Basic idea

Please, do not miss 60+ pages of Zhi Xiaojin (2008) survey.

1. Target event is reconstructed as average *Y* of closest clients within feature space (metrical approach)
2. Model is trained on united dataset with simulated target



thenounproject.com

Expected benefit in case of unobservable FP and FN

Expected benefit

Applying reject inference or semi-supervised learning technique, we obtain probabilistic estimate of Y for rejected clients

$X_1 \dots X_k \rightarrow$		$Prob$	$\hat{Y}$	Y
34		0.94	1	1
20		0.25	0	0
21		0.16	0	1
50	…	0.86	1	1
41		0.51	1	0
25		0.33	0	1
19		0.27	0	0
82		0.12	0	0
15		0.87	0	0.87
19		0.83	1	0.83
27		0.75	0	0.75
11		0.99	1	0.99

Often, probabilistic estimates are equal to model predictions (due to procedure in reject inference), but generally they can be different

Expected benefit

After iterating through various threshold levels a (and corresponding acceptance rates c), we can calculate cumulative expected benefit

$$Benefit(c) = \sum_{i:Prob_i \leq a} (1 - Y_i) \cdot e_{FP} - Y_i \cdot e_{FN}$$

One could also mention here:

$$FP(c) = \sum_{i:Prob_i > a} (1 - Y_i)$$

$$FN(c) = \sum_{i:Prob_i \leq a} Y_i$$

As soon as Y_i is probabilistic for rejected clients all formulas give us expected rather than precise result

Expected benefit. Open question

1. Suppose you have two different credit scoring models: logistic regression and XGBoost

Pr_{reg}	Pr_{xgb}	Y_{reg}	Y_{xgb}
0.75	0.94	1	1
0.31	0.25	0	0
0.19	0.16	1	1
0.84	0.86	1	1
0.58	0.51	0	0
0.49	0.33	1	1
0.22	0.27	0	0
0.06	0.12	0	0
0.90	0.87	0.90	0.87
0.67	0.83	0.67	0.83
0.81	0.75	0.81	0.75
0.95	0.99	0.95	0.99

Expected benefit. Open question

2. For each algorithm you will produce probabilistic estimates of target event for rejected clients

Pr_{reg}	Pr_{xgb}	Y_{reg}	Y_{xgb}
0.75	0.94	1	1
0.31	0.25	0	0
0.19	0.16	1	1
0.84	0.86	1	1
0.58	0.51	0	0
0.49	0.33	1	1
0.22	0.27	0	0
0.06	0.12	0	0
0.90	0.87	0.90	0.87
0.67	0.83	0.67	0.83
0.81	0.75	0.81	0.75
0.95	0.99	0.95	0.99

You can calculate expected benefit, but each model has its own imputed target events, which makes model comparison difficult

Wrap-up and one open question

1. Reject inference and semi-supervised learning may help to extract maximum information from available data to train model

Wrap-up and one open question

1. Reject inference and semi-supervised learning may help to extract maximum information from available data to train model

2. Both control groups and reject reference allow (explicitly or implicitly) us to reconstruct target event for rejected clients

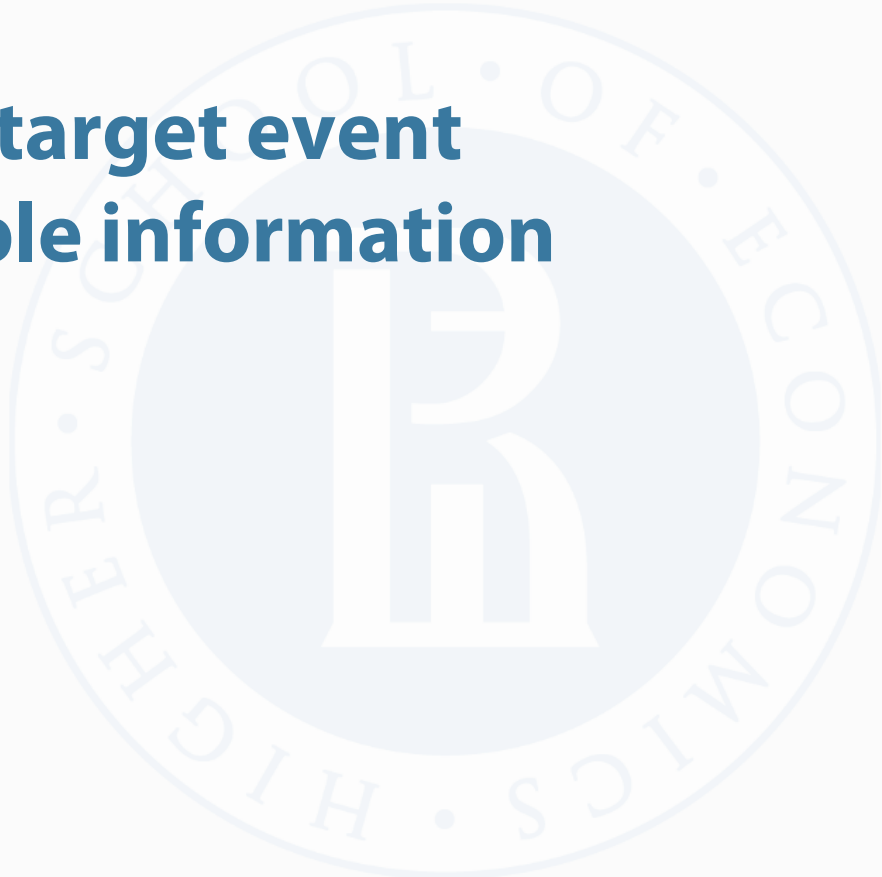
Wrap-up and one open question

1. Reject inference and semi-supervised learning may help to extract maximum information from available data to train model
2. Both control groups and reject reference allow (explicitly or implicitly) us to reconstruct target event for rejected clients
3. However, there is no universal rule or theorem that reject inference guarantees better model. Each dataset and each process has to be examined separately

Wrap-up and one open question

1. Reject inference and semi-supervised learning may help to extract maximum information from available data to train model
2. Both control groups and reject reference allow (explicitly or implicitly) us to reconstruct target event for rejected clients
3. However, there is no universal rule or theorem that reject inference guarantees better model. Each dataset and each process has to be examined separately
4. More than that, it is an open question how to compare two models after applying reject inference independently

Metalearning.
Reconstructing target event using all available information

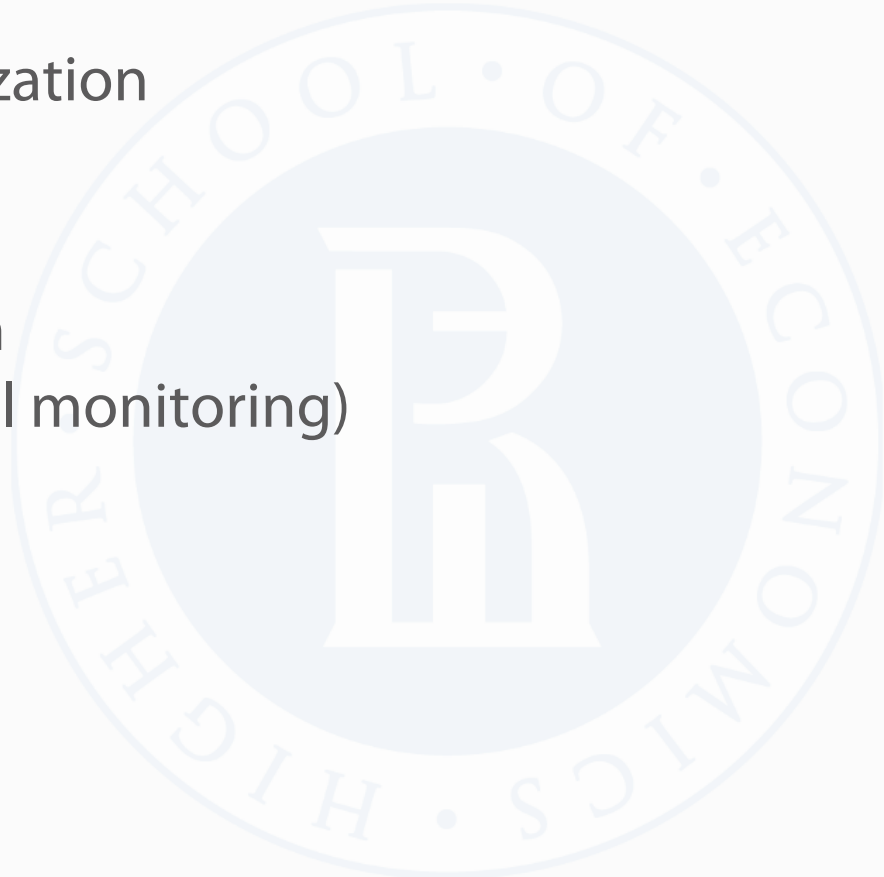


Metalearning. Why bother?

Metalearning — is a subfield of machine learning where ML methods are applied to metadata about ML models performance

Examples:

1. Smart hyperparameter optimization
2. AutoML libraries
3. Model performance prediction
   (early warning signals in model monitoring)



Metalearning. Why bother?

The last obstacle — two different models and we do not know which performs better on rejected clients' data

It seems that subfield of machine learning that analyses meta data about model performance may help

Metalearning. Reconstructing target event

Resume to our example. We need to produce one probabilistic estimate of Y for rejected clients in order to compare models

Pr_{reg}	Pr_{xgb}	Y_{reg}	Y_{xgb}
0.75	0.94	1	1
0.31	0.25	0	0
0.19	0.16	1	1
0.84	0.86	1	1
0.58	0.51	0	0
0.49	0.33	1	1
0.22	0.27	0	0
0.06	0.12	0	0
0.90	0.87	0.90	0.87
0.67	0.83	0.67	0.83
0.81	0.75	0.81	0.75
0.95	0.99	0.95	0.99

Metalearning. Reconstructing target event

Intuition. We must use all information, so we pick predictions from both models

Pr_{reg}	Pr_{xgb}	Y_{reg}	Y_{xgb}
0.75	0.94	1	1
0.31	0.25	0	0
0.19	0.16	1	1
0.84	0.86	1	1
0.58	0.51	0	0
0.49	0.33	1	1
0.22	0.27	0	0
0.06	0.12	0	0
0.90	0.87	0.90	0.87
0.67	0.83	0.67	0.83
0.81	0.75	0.81	0.75
0.95	0.99	0.95	0.99

It seems we can blend them using some weights.
Do we want the more precise model to have larger weight?

Metalearning. Reconstructing target event

Enrich our dataset with meta information which model was more precise on data segment with observed target event

Pr_{reg}	Pr_{xgb}	Y_{reg}	Y_{xgb}
0.75	0.94	1	1
0.31	0.25	0	0
0.19	0.16	1	1
0.84	0.86	1	1
0.58	0.51	0	0
0.49	0.33	1	1
0.22	0.27	0	0
0.06	0.12	0	0
0.90	0.87	0.90	0.87
0.67	0.83	0.67	0.83
0.81	0.75	0.81	0.75
0.95	0.99	0.95	0.99

Here XGBoost prediction was closer to correct answer $Y_i = 1$

Metalearning. Reconstructing target event

Enrich our dataset with meta information which model was more precise on data segment with observed target event

Pr_{reg}	Pr_{xgb}	Y_{reg}	Y_{xgb}
0.75	0.94	1	1
0.31	0.25	0	0
0.19	0.16	1	1
0.84	0.86	1	1
0.58	0.51	0	0
0.49	0.33	1	1
0.22	0.27	0	0
0.06	0.12	0	0
0.90	0.87	0.90	0.87
0.67	0.83	0.67	0.83
0.81	0.75	0.81	0.75
0.95	0.99	0.95	0.99

Here XGBoost prediction was closer to correct answer $Y_i = 1$

So, we can use logloss to discriminate:

$$-Y_i \log(P_i) - (1 - Y_i)\,(1 - \log(P_i))$$

where we input Pr_{xgb}, Pr_{reg} as P_i

Metalearning. Reconstructing target event

Enrich our dataset with meta information which model was more precise on data segment with observed target event

Pr_{reg}	Pr_{xgb}	Y_{reg}	Y_{xgb}	$Flag$
0.75	0.94	1	1	1
0.31	0.25	0	0	1
0.19	0.16	1	1	0
0.84	0.86	1	1	1
0.58	0.51	0	0	1
0.49	0.33	1	1	0
0.22	0.27	0	0	0
0.06	0.12	0	0	0
0.90	0.87	0.90	0.87	-
0.67	0.83	0.67	0.83	-
0.81	0.75	0.81	0.75	-
0.95	0.99	0.95	0.99	-

Meta data

$Flag = 1$ if XGBoost prediction is closer to correct answer than log reg prediction

$Flag = 0$, otherwise

Metalearning. Reconstructing target event

Train an auxiliary classifier which takes all client $X_1, \ldots, X_k$ as input and predicts $Flag$ as target.

Pr_{reg}	Pr_{xgb}	Y_{reg}	Y_{xgb}	$Flag$
0.75	0.94	1	1	1
0.31	0.25	0	0	1
0.19	0.16	1	1	0
0.84	0.86	1	1	1
0.58	0.51	0	0	1
0.49	0.33	1	1	0
0.22	0.27	0	0	0
0.06	0.12	0	0	0
0.90	0.87	0.90	0.87	-
0.67	0.83	0.67	0.83	-
0.81	0.75	0.81	0.75	-
0.95	0.99	0.95	0.99	-

Meta as a new target

As a result we expect to answer, which clients are better predicted by first and second model

Metalearning. Reconstructing target event

Prediction of an auxiliary classifier Pr_{aux} is in fact a probability that XGBoost is likely to be more precise for this particular client

Pr_{reg}	Pr_{xgb}	Y_{reg}	Y_{xgb}	$Flag$	Pr_{aux}
0.75	0.94	1	1	1	0.98
0.31	0.25	0	0	1	0.92
0.19	0.16	1	1	0	0.04
0.84	0.86	1	1	1	0.67
0.58	0.51	0	0	1	0.73
0.49	0.33	1	1	0	0.21
0.22	0.27	0	0	0	0.34
0.06	0.12	0	0	0	0.03
0.90	0.87	0.90	0.87	-	0.78
0.67	0.83	0.67	0.83	-	0.43
0.81	0.75	0.81	0.75	-	0.79
0.95	0.99	0.95	0.99	-	0.99

We are really interested in these values: they serve as weights for blending

Metalearning. Reconstructing target event

Finally, we produce one probabilistic estimate Y_{aprox} for rejected clients in order to compare models

Pr_{reg}	Pr_{xgb}	Y_{reg}	Y_{xgb}	$Flag$	Pr_{aux}	Y_{aprox}
0.75	0.94	1	1	1	0.98	1
0.31	0.25	0	0	1	0.92	0
0.19	0.16	1	1	0	0.04	1
0.84	0.86	1	1	1	0.67	1
0.58	0.51	0	0	1	0.73	0
0.49	0.33	1	1	0	0.21	1
0.22	0.27	0	0	0	0.34	0
0.06	0.12	0	0	0	0.03	0
0.90	0.87	0.90	0.87	-	0.78	0.88
0.67	0.83	0.67	0.83	-	0.43	0.74
0.81	0.75	0.81	0.75	-	0.79	0.76
0.95	0.99	0.95	0.99	-	0.99	0.99

$$Y_{aprox} = Y_{reg}(1 - Pr_{aux}) + Y_{xgb}Pr_{aux}$$

With Y_{aprox} we can now calculate expected FP, FN and benefit

Wrap-up

1. Metalearning uses models that predict models' performance
2. This can be useful if we want to compare two models given considerable number of rejected clients with unknown target event
3. Reconstructing target event can be thought as a special blending technique and can have more general applications
4. No matter how sophisticated target reconstruction techniques are, the best option (from statistical standpoint) is to run a solid control group