Validation and overfitting

Restaurant Revenue Prediction

Predict annual restaurant sales based on objective measurements

$30,000 · 2,257 teams · 2 years ago

Public Leaderboard | Private Leaderboard

This leaderboard is calculated with approximately 30% of the test data.

The final results will be based on the other 70%, so the final standings may be different.

Raw Data | Refresh

#	△priv	Team Name	Kernel	Team Members	Score	Entries	Last
1	▼19...	BAYZ, M.D.			0.00000	115	2y
2	▼16...	Will Iam			710063.76...	116	2y
3	▼10...	Scott Lowe			1462479.4...	106	2y
4	▼935	AMAR_PREM_AnandAkela_Teja			1464692.1...	97	2y
5	▼683	Analytic Bastard			1492787.0...	115	2y

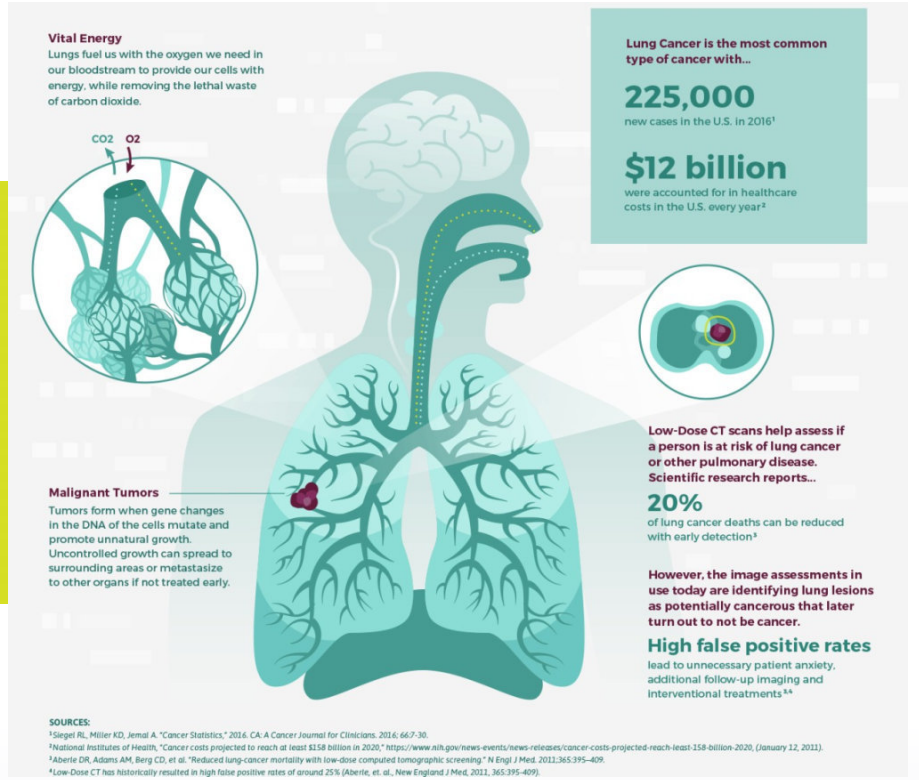
Next videos

1. We will understand the concept of validation and overfitting
2. We will identify the number of splits that should be done to establish stable validation
3. We will break down most frequent ways to repeat train test split
4. We will discuss most often validation problems

Validation: example

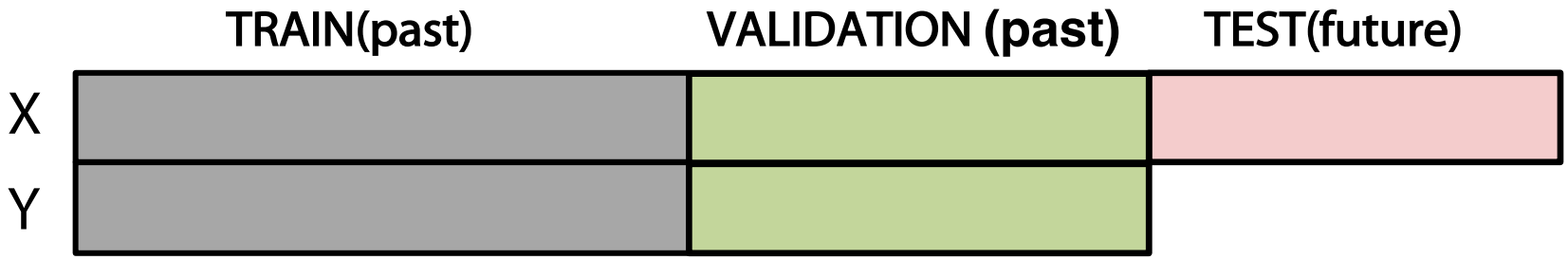
DATA SCIENCE BOWL

Passion. Curiosity. Purpose.

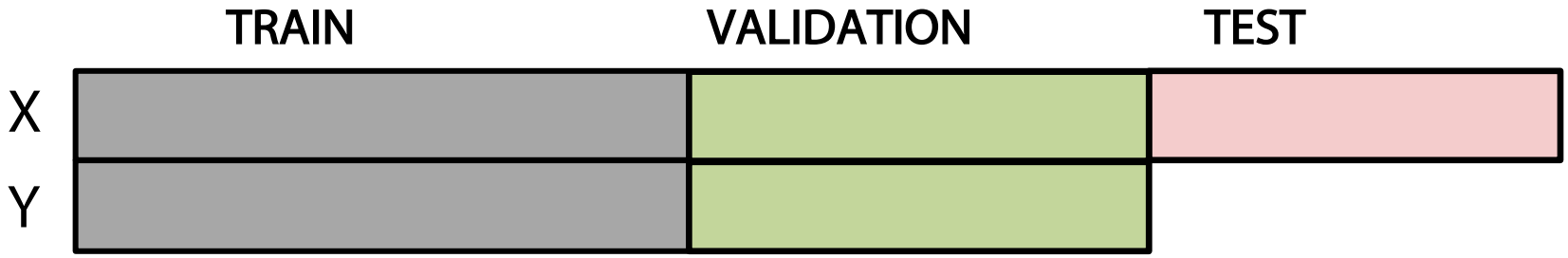


Validation: example

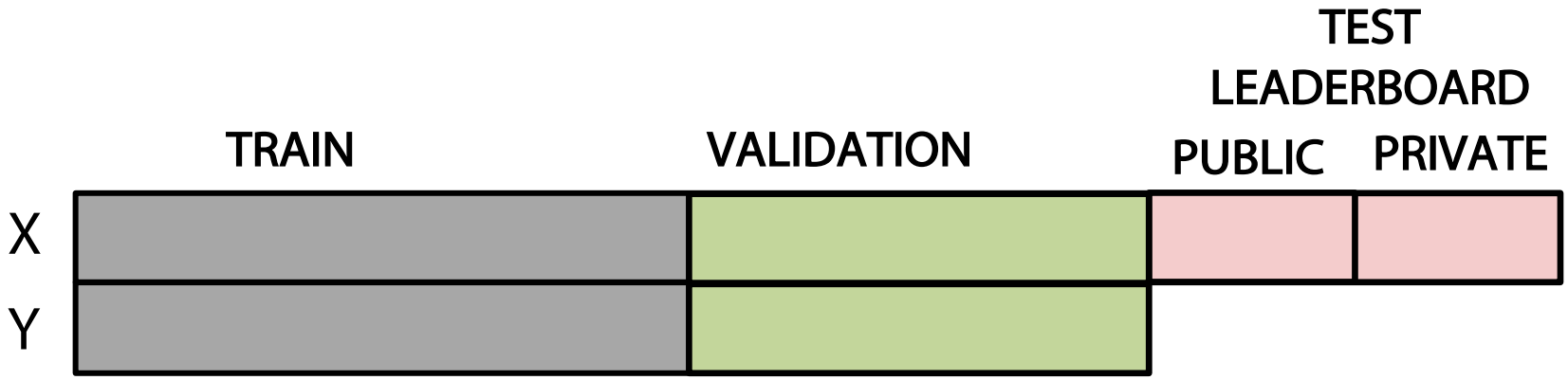
Validation: example



Validation: competitions

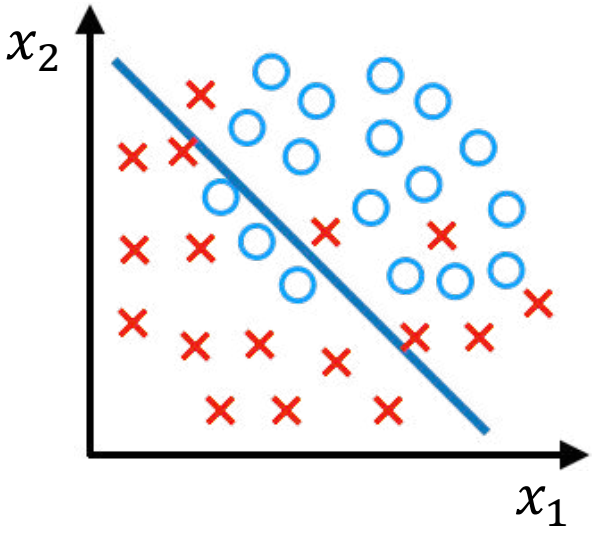


Validation: competitions



Validation: underfitting and overfitting

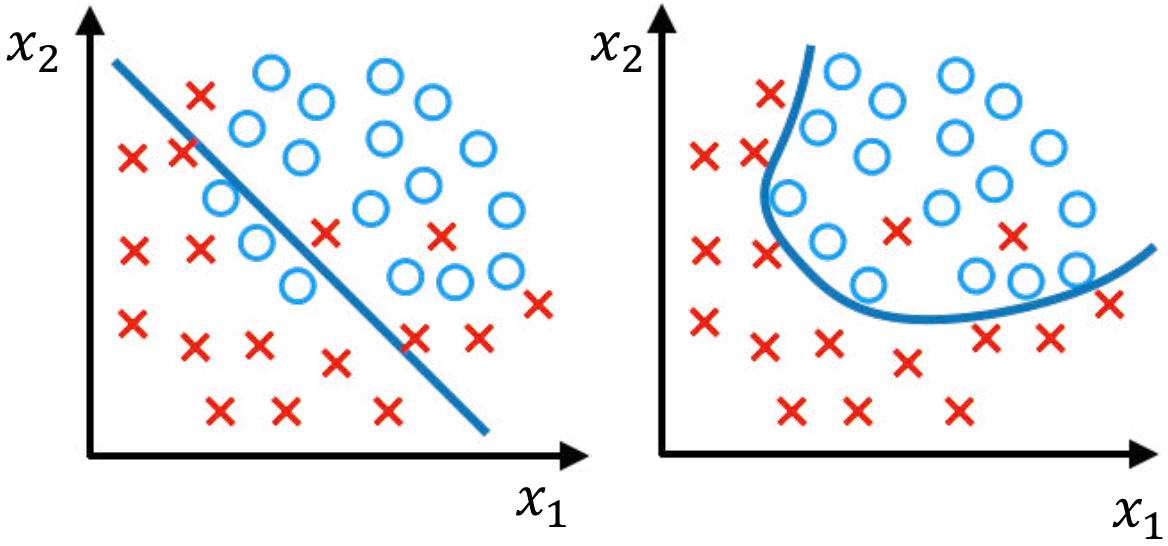
UNDERFITTING



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$
$$(g = sigmoid\ function)$$

Validation: underfitting and overfitting

UNDERFITTING

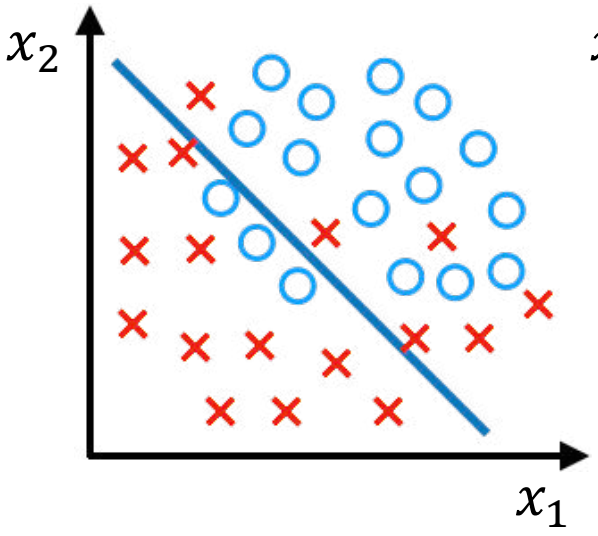


$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$
$(g = sigmoid\ function)$

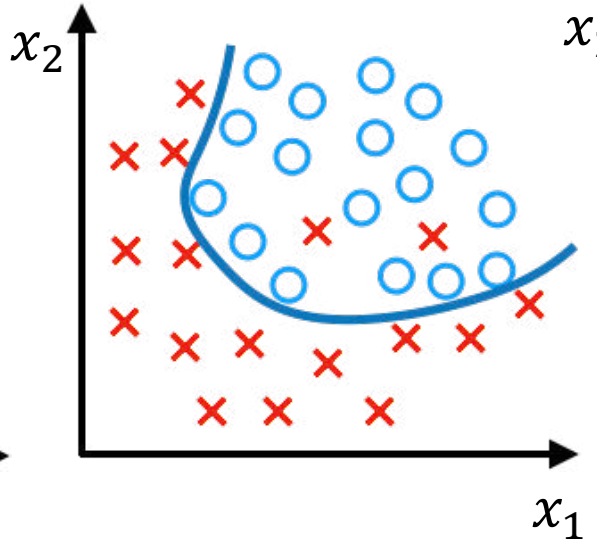
$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2$
$+\theta_3 x_1^2 + \theta_4 x_2^2$
$+\theta_5 x_1 x_2)$

Validation: underfitting and overfitting

UNDERFITTING

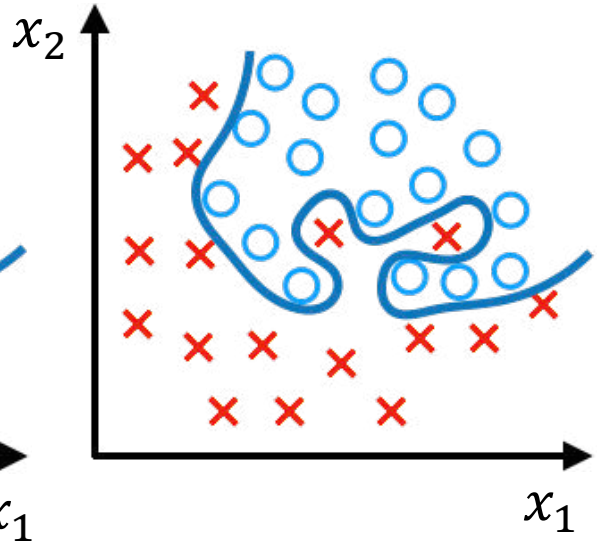


$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$
$$(g = sigmoid\ function)$$



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$$

OVERFITTING



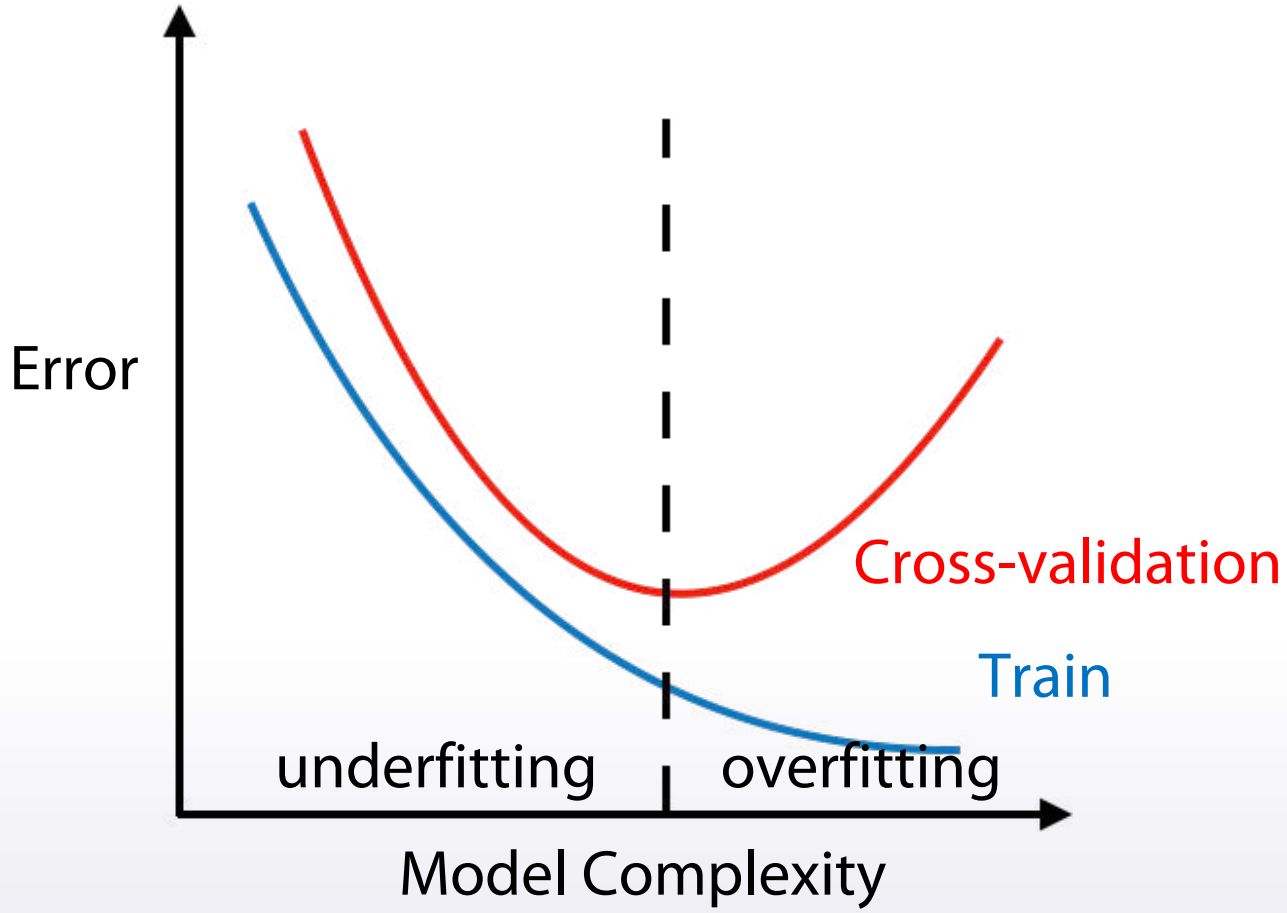
$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2)$$

Validation: underfitting and overfitting

Overfitting in general != overfitting in competitions

Validation: underfitting and overfitting

Overfitting in general != overfitting in competitions



Conclusion

1. Validation helps us evaluate a quality of the model
2. Validation helps us select the model which will perform best on the unseen data
3. Underfitting refers to not capturing enough patterns in the data
4. Generally, overfitting refers to
    a. capturing noize
    b. capturing patterns which do not generalize to test data
5. In competitions, overfitting refers to
    a. low model's quality on test data,
    which was unexpected due to validation scores

Validation strategies

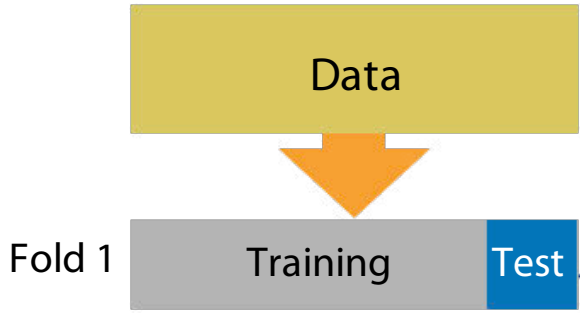
Validation types

- Holdout
- K-fold
- Leave-one-out

Validation types

- Holdout: ngroups = 1

  `sklearn.model_selection.ShuffleSplit`



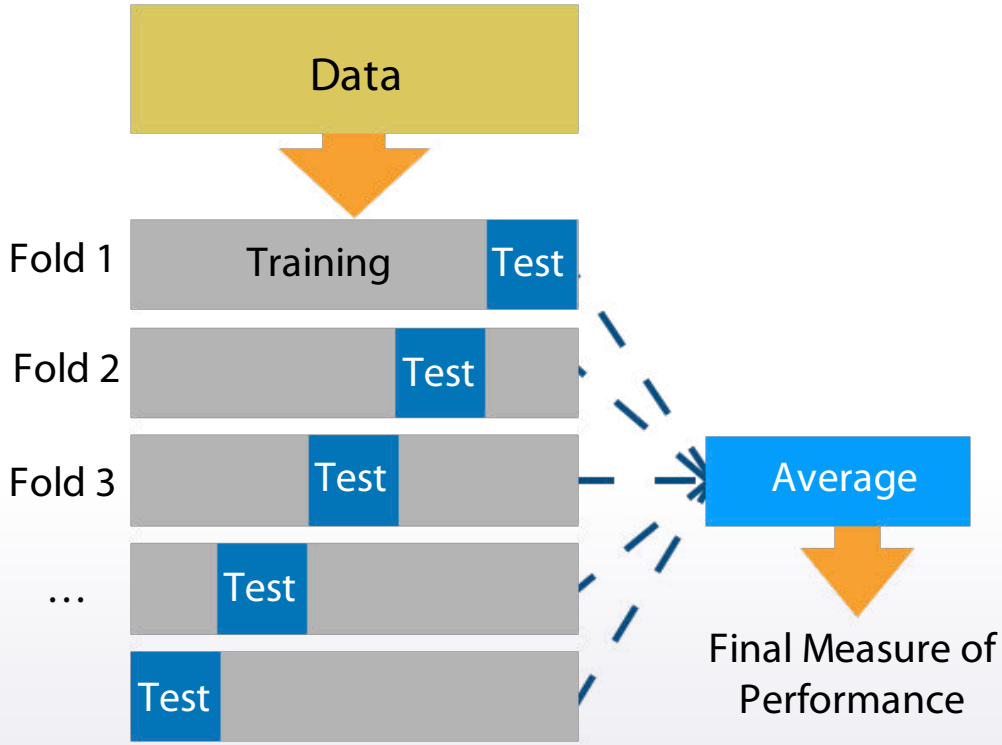
Validation types

- Holdout: ngroups = 1

  `sklearn.model_selection.ShuffleSplit`

- K-fold: ngroups = k

  `sklearn.model_selection.Kfold`



Validation types

- Holdout: ngroups = 1

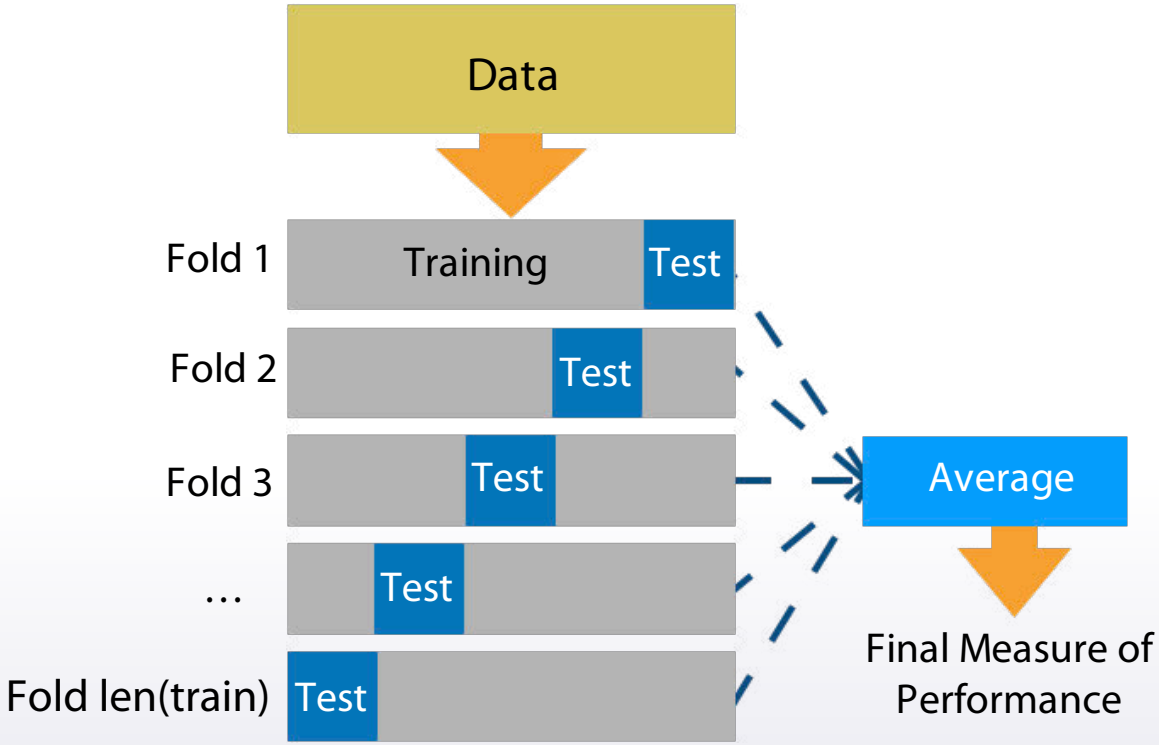
  `sklearn.model_selection.ShuffleSplit`

- K-fold: ngroups = k

  `sklearn.model_selection.Kfold`

- Leave-one-out: ngroups = len(train)

  `sklearn.model_selection.LeaveOneOut`



Stratification

Samples and their target values

0	1	0	0	1	1	1	0

Stratification

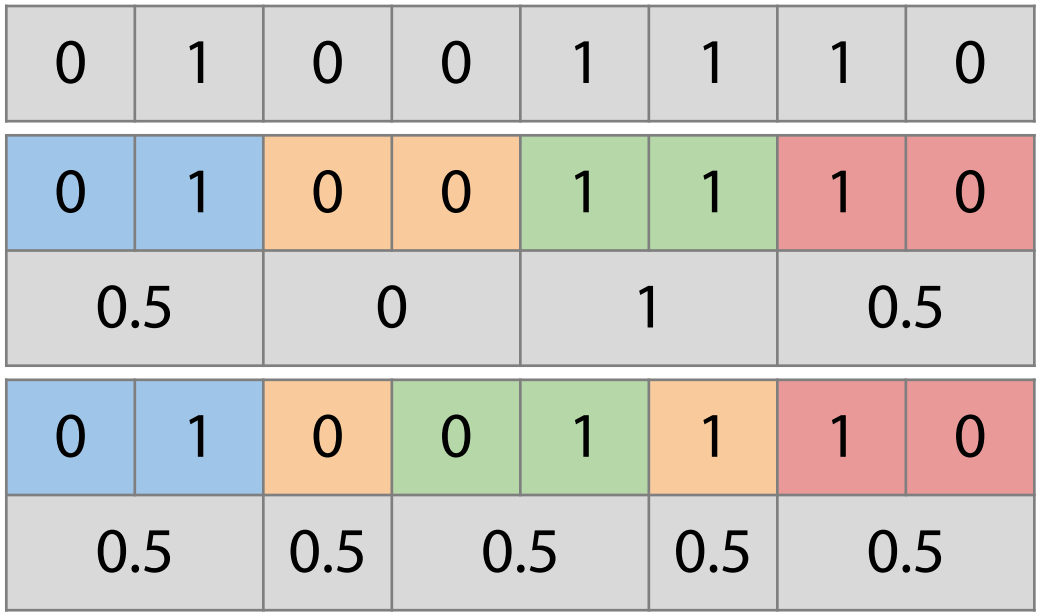
Samples and their target values

0	1	0	0	1	1	1	0

0	1	0	0	1	1	1	0
0.5		0		1		0.5	

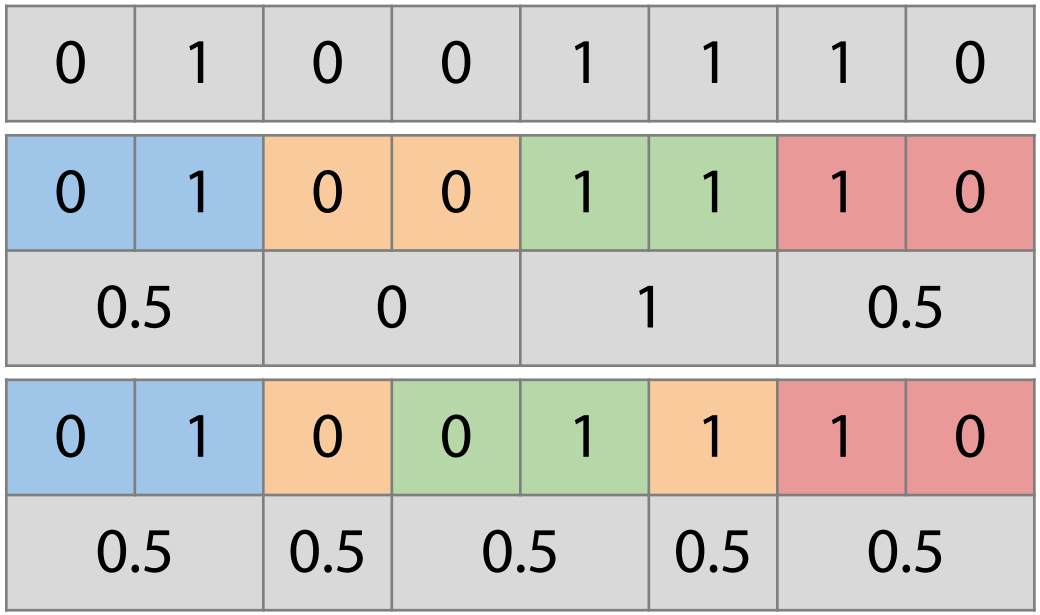
Stratification

Samples and their target values



Stratification

Samples and their target values



Stratification is useful for:

- Small datasets
- Unbalanced datasets
- Multiclass classification

Conclusion

There are three main validation strategies:

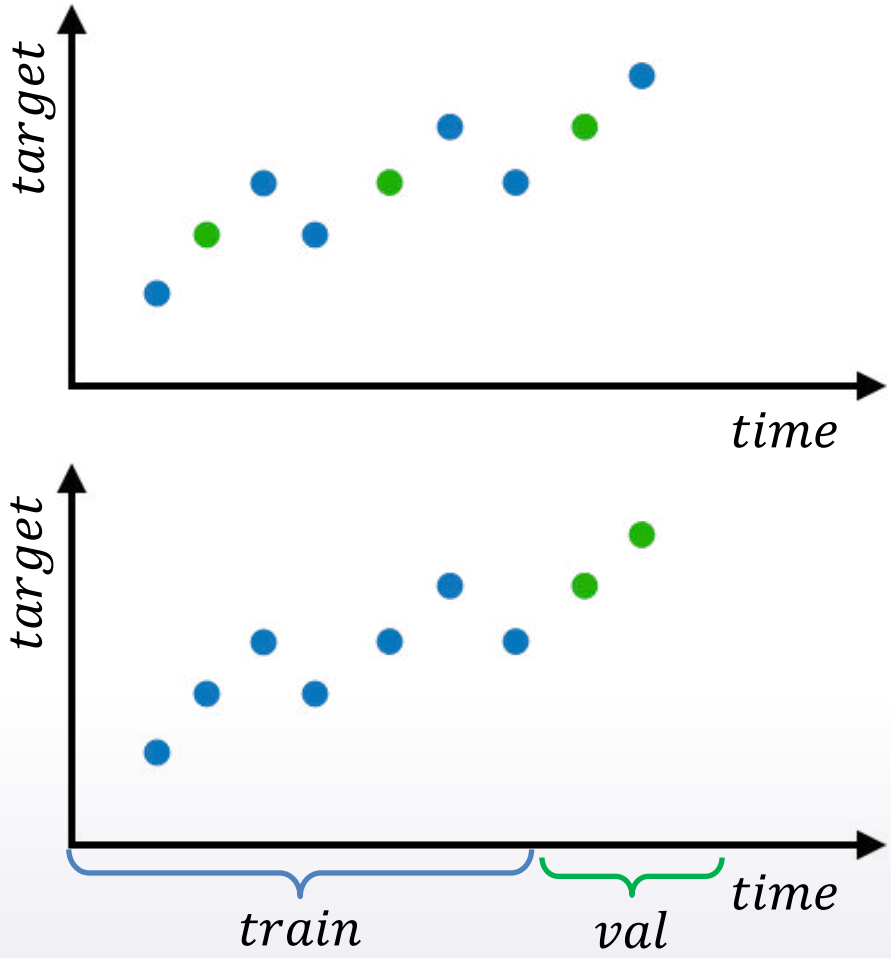
1. Holdout
2. KFold
3. LOO

Stratification preserve the same target distribution over different folds

Data splitting strategies

Different approaches to validation

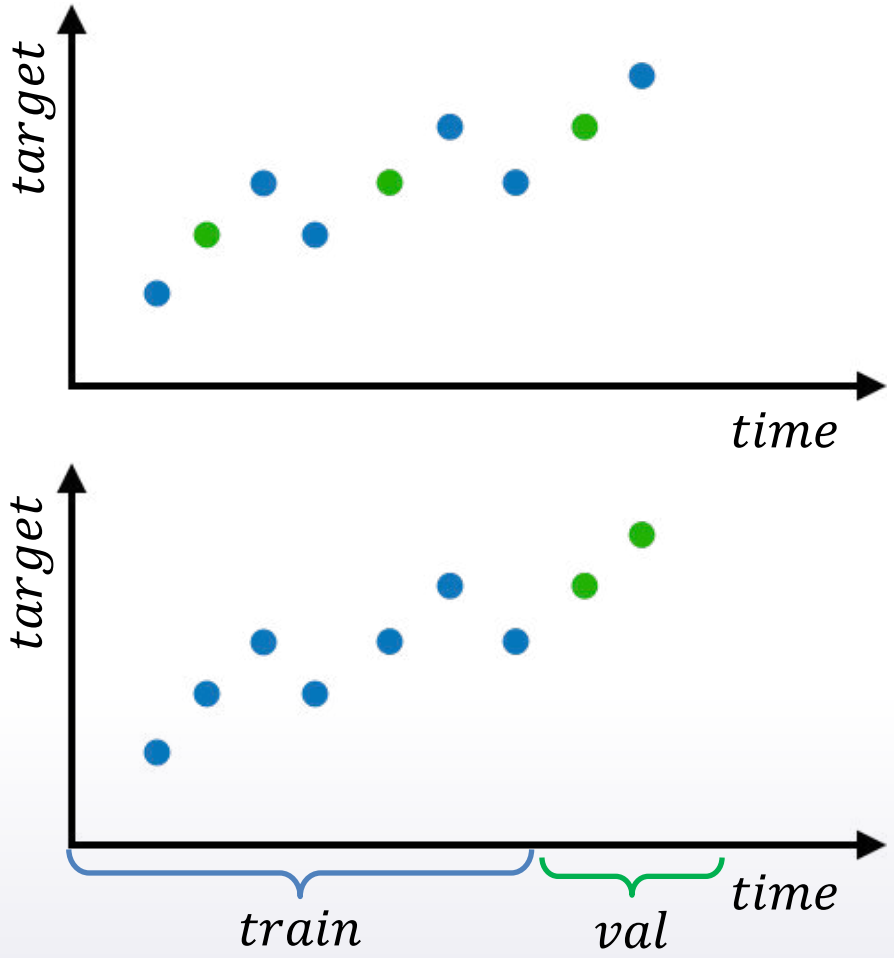
set up validation to replicate train/test split



Different approaches to validation

Important features:

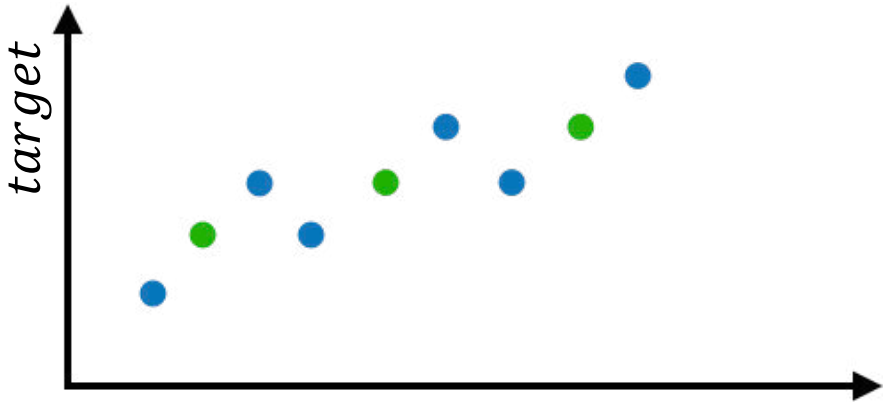
1. Previous and next target values



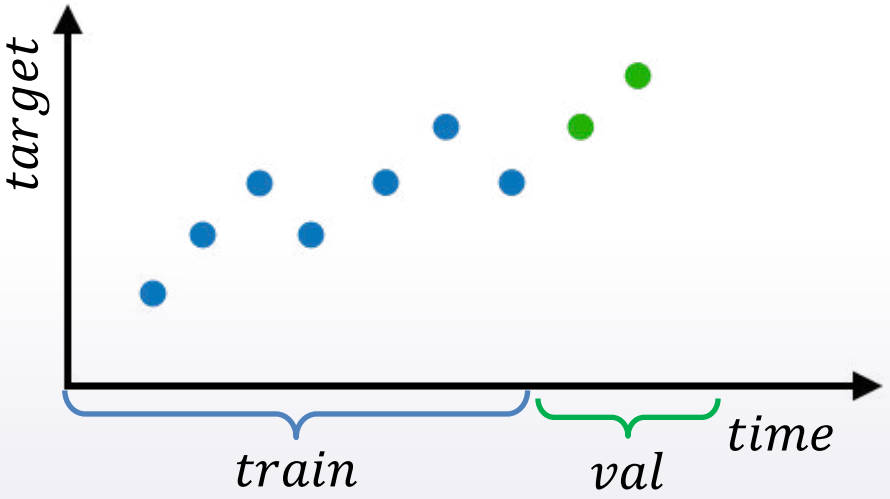
Different approaches to validation

Important features:

1. Previous and next target values



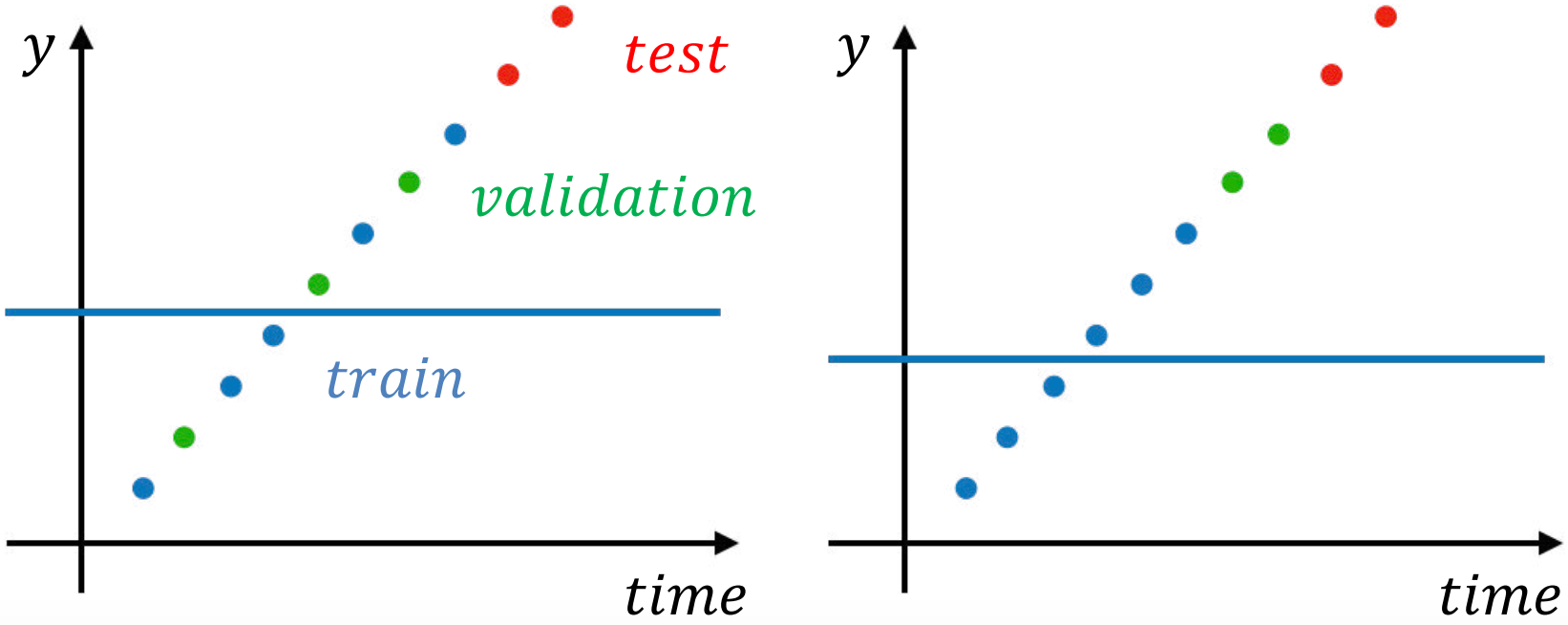
2. Time-based trend



Question screen

If we carefully generate features that are drawing attention to time-based patterns, will we get a reliable validation with a random-based split?

Different approaches to validation



Time-based splits

- “Rossman Store Sales”

- “Grupo Bimbo Inventory Demand”

Important outcome

Different splitting strategies can differ significantly

1. in generated features
2. in a way the model will rely on that features
3. in some kind of target leak

Splitting data into train and validation

1. Random, rowwise
2. Timewise
3. By id

Splitting data into train and validation

1. Random, rowwise
2. Timewise
3. By id

Splitting data into train and validation

1. Random, rowwise
2. Timewise
3. By id

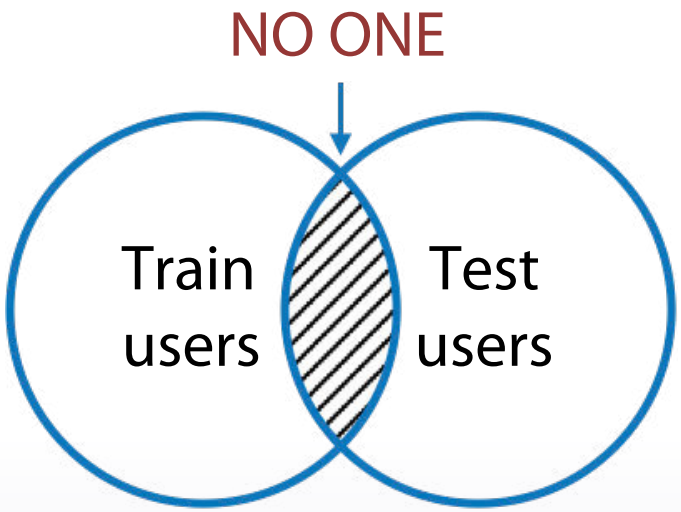
Moving window

Moving window validation

week1	week2	week3	week4	week5	week6
train			validation		
train				validation	
train					validation

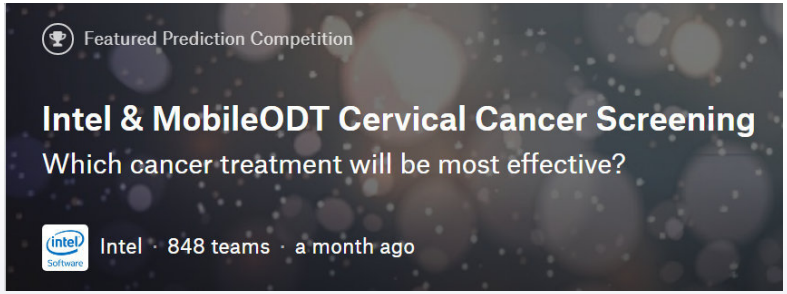
Splitting data into train and validation

1. Random, rowwise
2. Timewise
3. By id



Splitting data into train and validation

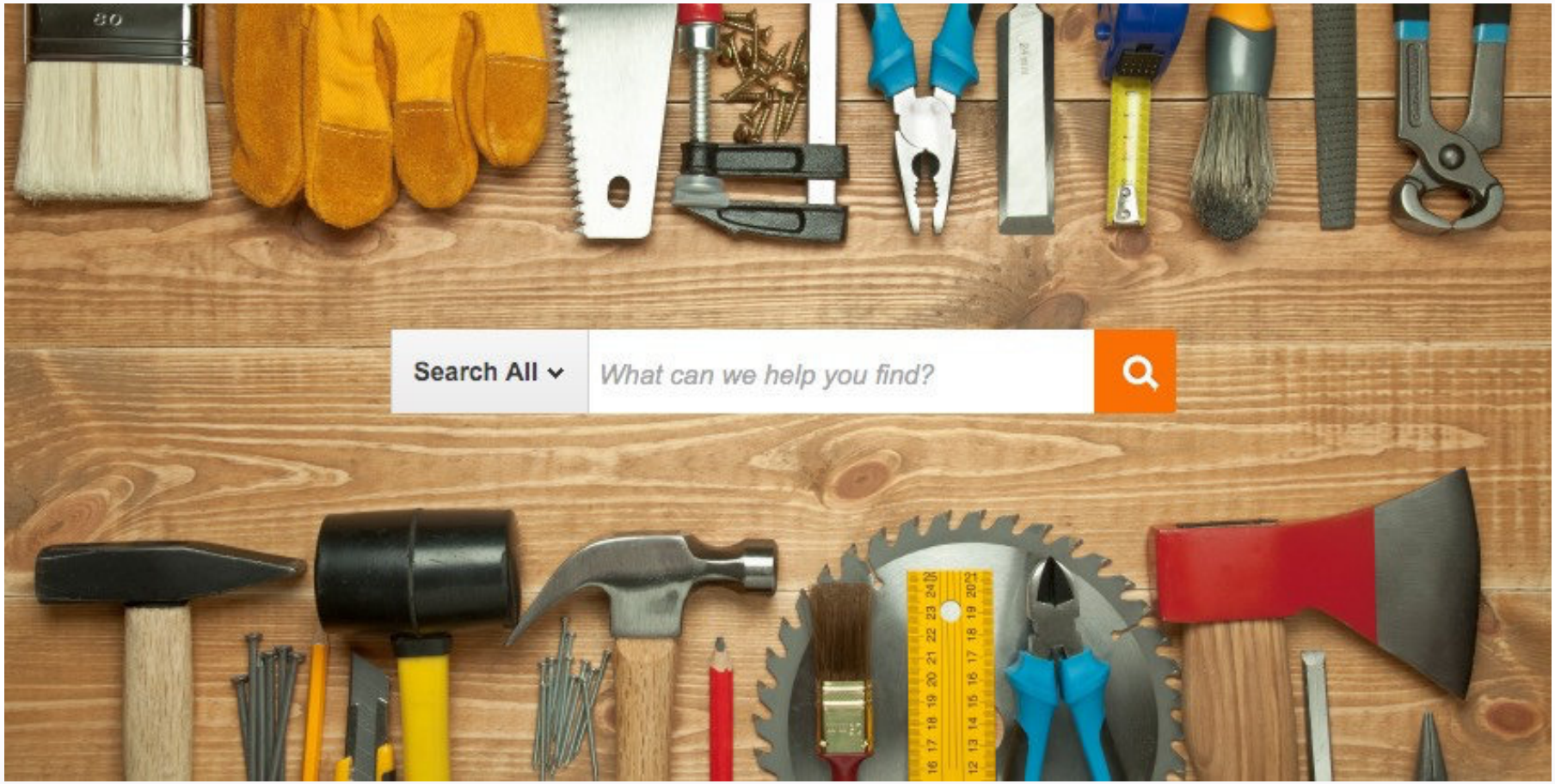
1. Random, rowwise
2. Timewise
3. By id



Splitting data into train and validation

1. Random, rowwise
2. Timewise
3. By id
4. Combined

Home Depot Product Search Relevance



Conclusion

1. In most cases data is split by
    a. Row number
    b. Time
    c. Id
2. Logic of feature generation depends on the data splitting strategy
3. Set up your validation to mimic the train/test split of the competition

Common validation problems

Validation

1. We discussed the concept of validation and overfitting
2. We understood how to choose validation strategy
3. We learned to identify data split made by organizers.

Validation

1. We discussed the concept of validation and overfitting
2. We understood how to choose validation strategy
3. We learned to identify data split made by organizers.
4. Validation problems
    a. Validation stage
    b. Submission stage

Validation stage

Holidays in Russia

January

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

February

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

8 holidays

14 weekend

12 working days

Validation stage

Causes of different scores and optimal parameters

1. Too little data

Validation stage

Causes of different scores and optimal parameters

1. Too little data
2. Too diverse and inconsistent data

Validation stage

Causes of different scores and optimal parameters

1. Too little data
2. Too diverse and inconsistent data

We should do extensive validation

1. Average scores from different KFold splits
2. Tune model on one split, evaluate score on the other

Validation stage: extensive validation

- Liberty Mutual Group:
  Property Inspection Prediction

- Santander Customer Satisfaction

Submission stage

We can observe that:

- LB score is consistently higher/lower that validation score
- LB score is not correlated with validation score at all

Submission stage

0. We may already have quite different scores in Kfold

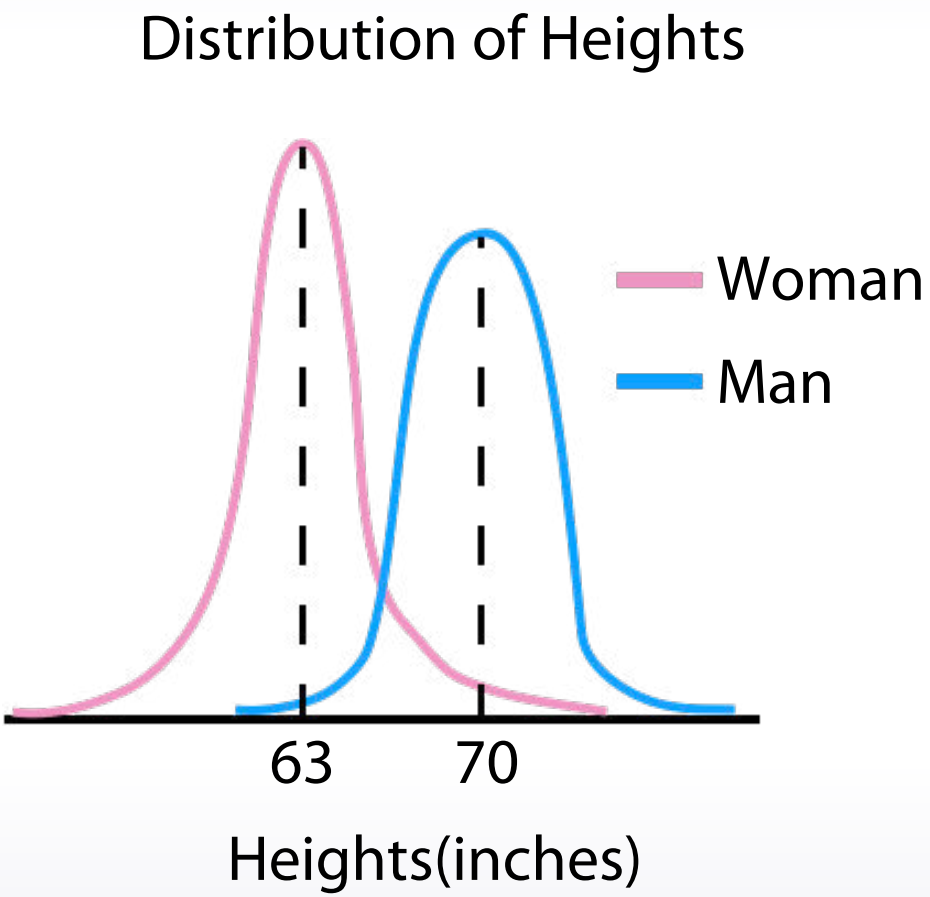
Submission stage

0. We may already have quite different scores in Kfold

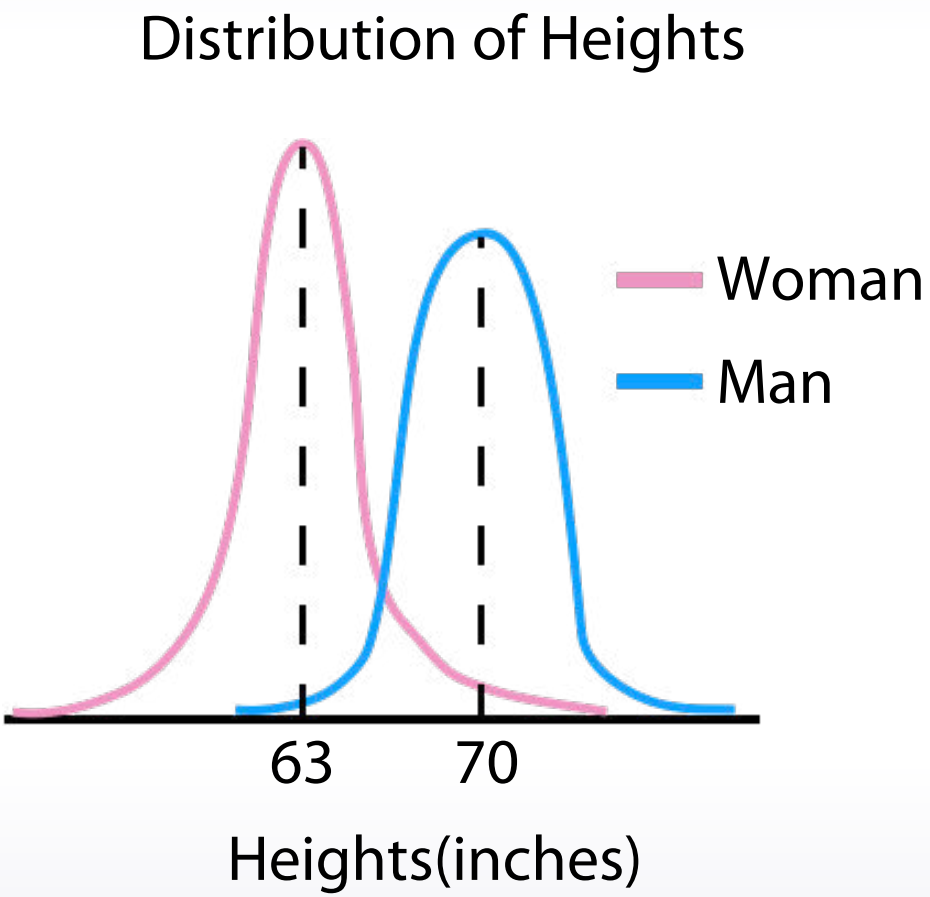
Other reasons:

1. too little data in public leaderboard
2. train and test data are from different distributions

Submission stage: different distributions

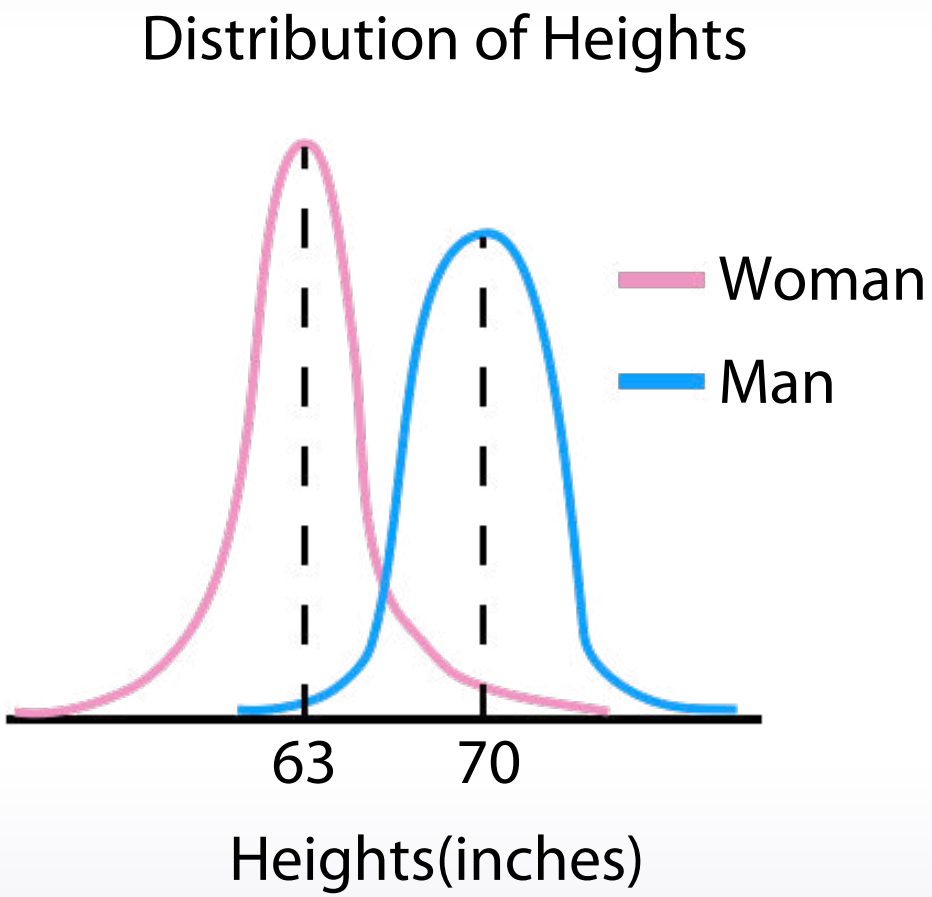


Submission stage: different distributions



- Mean for train: Calculate from the train data
- Mean for test: Leaderboard probing

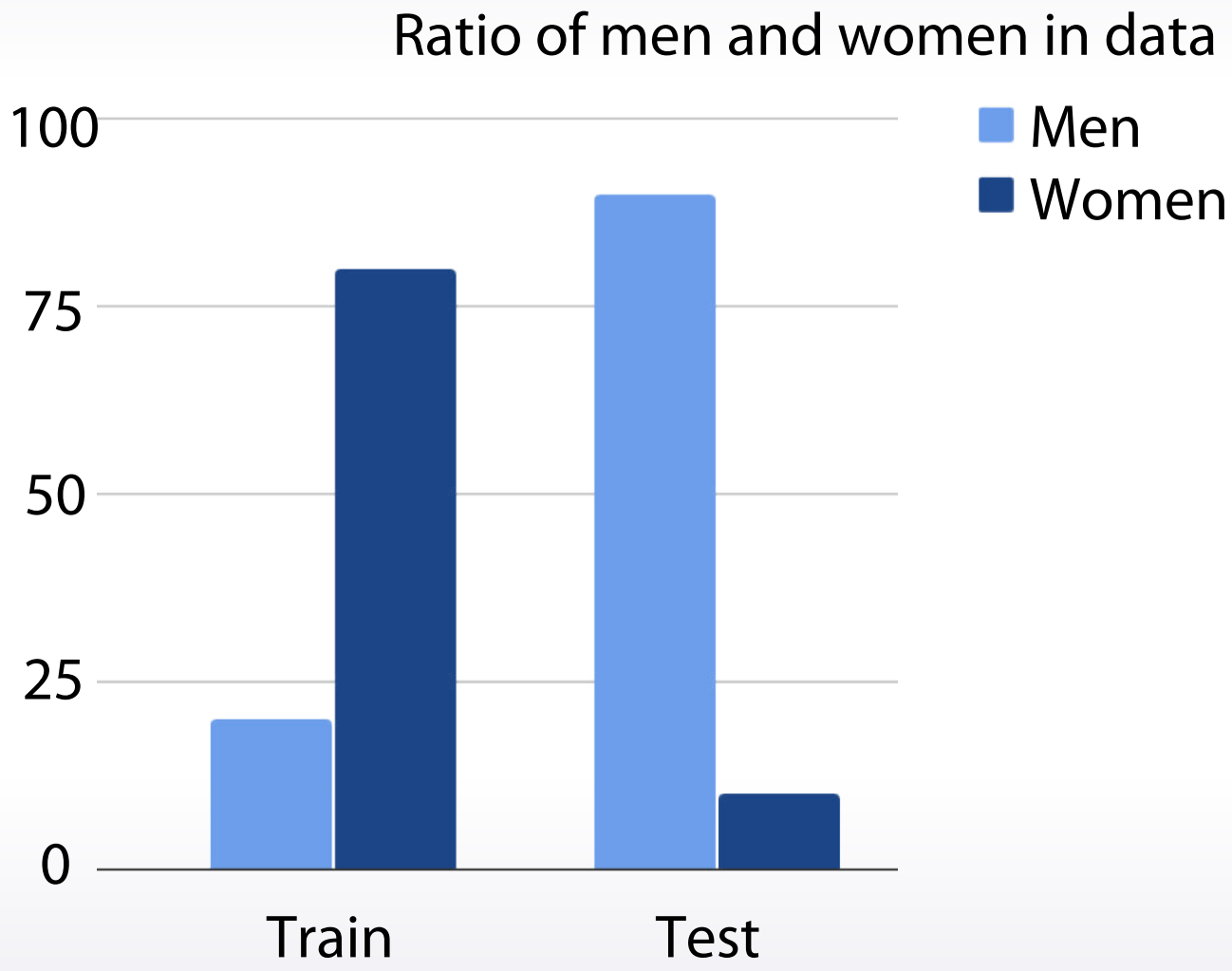
Submission stage: different distributions



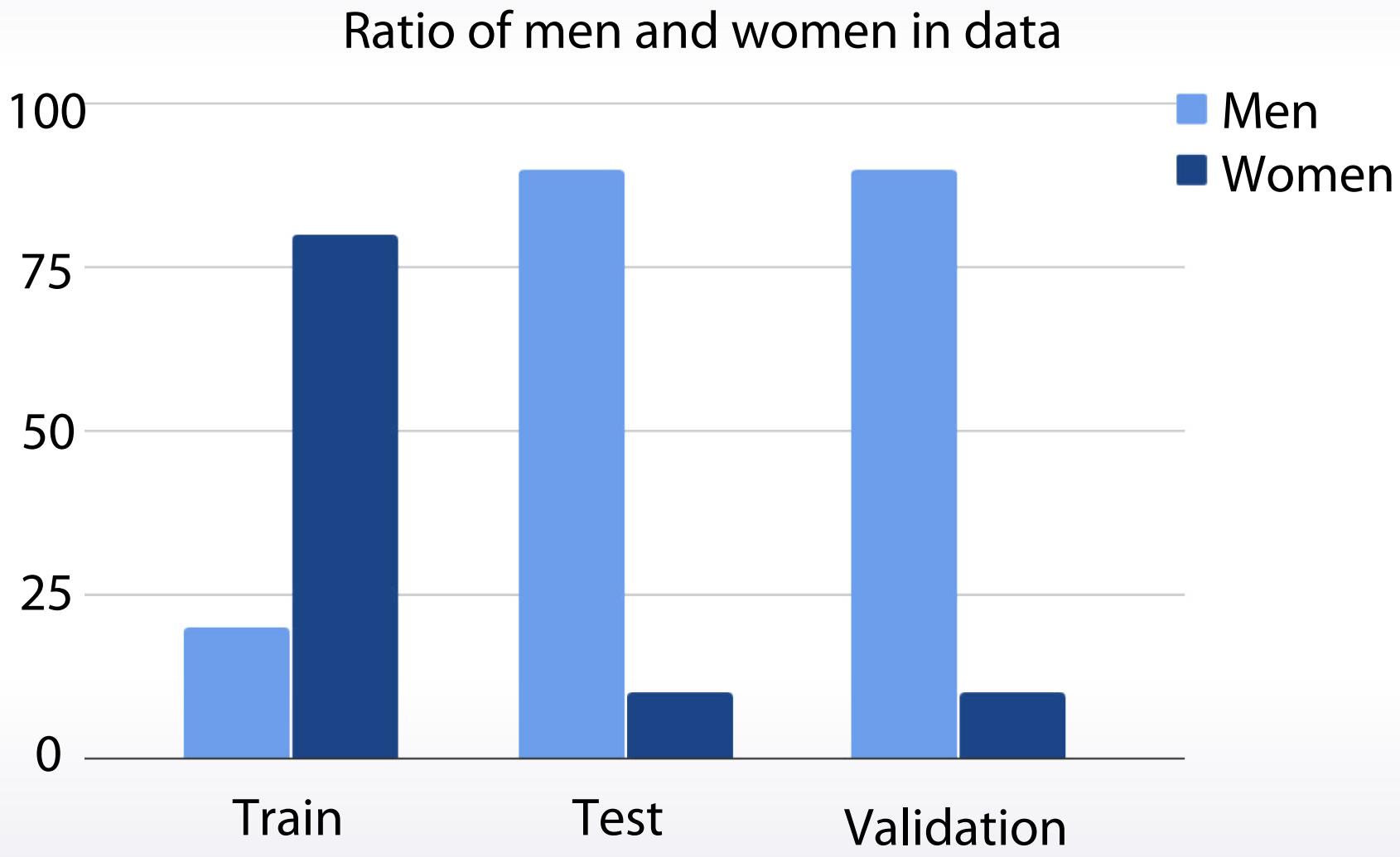
Quora Question Pairs

Quora

Submission stage: different distributions



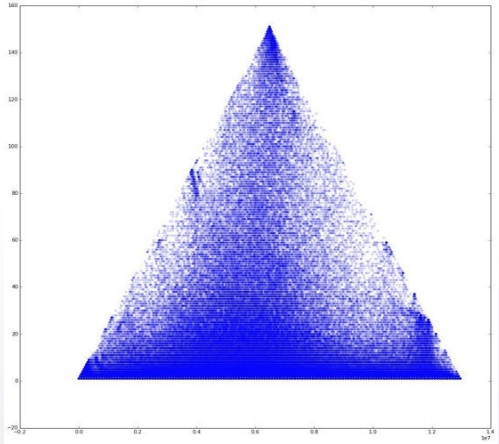
Submission stage: different distributions



Submission stage: different distributions

- Data Science Game 2017 Qualification phase: Music recommendation

- CTR prediction task from EDA



Submission stage

Causes of validation problems:

- too little data in public leaderboard
- incorrect train/test split
- different distributions in train and test

LB shuffle

#	△pub	Team Name	Kernel	Team Members	Score	Entries	Last
1	▲ 35	Dr. Knope			0.0382244	26	5mo
2	▲ 11	NimaShahbazi & mchahhou			0.0369387	170	5mo
3	▲ 389	rnrq			0.0343235	70	5mo
4	▲ 6	Data Finance			0.0323850	100	5mo
5	—	best fitting			0.0320763	172	5mo
6	▲ 33	NIWATORI			0.0301690	31	5mo
7	▲ 8	E2			0.0291539	43	5mo
8	▲ 11	John Ma			0.0289587	97	5mo
9	▲ 25	Pradeep and Arthur			0.0287992	111	5mo
10	▼ 4	William Hau			0.0287899	165	5mo

Expect LB shuffle because of

- Randomness
- Little amount of data
- Different public/private distributions

Expect LB shuffle because of

- Randomness

- Little amount of data

- Different public/private distributions

Expect LB shuffle because of

- Randomness

- Little amount of data

- Different public/private distributions

Expect LB shuffle because of

- Randomness

- Little amount of data

- Different public/private distributions

Conclusion

- If we have big dispersion of scores on validation stage, we should do extensive validation
  - Average scores from different KFold splits
  - Tune model on one split, evaluate score on the other

Conclusion

- If we have big dispersion of scores on validation stage, we should do extensive validation
  - Average scores from different KFold splits
  - Tune model on one split, evaluate score on the other
- If submission's score do not match local validation score, we should
  - Check if we have too little data in public LB
  - Check if we overfitted
  - Check if we chose correct splitting strategy
  - Check if train/test have different distibutions

Conclusion

- If we have big dispersion of scores on validation stage, we should do extensive validation
  - Average scores from different KFold splits
  - Tune model on one split, evaluate score on the other
- If submission's score do not match local validation score, we should
  - Check if we have too little data in public LB
  - Check if we overfitted
  - Check if we chose correct splitting strategy
  - Check if train/test have different distibutions
- Expect LB shuffle because of
  - Randomness
  - Little amount of data
  - Different public/private distributions

Summary of Validation topic

1. Defined validation and its connection to overfitting
2. Described common validation strategies
3. Demonstrated major data splitting strategies
4. Analysed and learn how to tackle main validation problems

Data leakage

A moment to reflect

- How bad it is?
- Whats the public opinion?
- To exploit or not to exploit

Contents

- Leakage types and examples
- Competition specific. Leaderboard probing
- Concrete walkthroughs

Leaks in time series

- Split should be done on time.
  - In real life we don't have information from future
  - In competitions first thing to look: train/public/private split, is it on time?
- Even when split by time, features may contain information about future.
  - User history in CTR tasks
  - Weather

Unexpected information

- Meta data
- Information in IDs
- Row order

Leaderboard probing

- Types of LB probing
- Categories tightly connected with ‘id’ are vulnerable to LB probing
  - Company of user in RedHat competition
  - Year, Month, Week in WestNile competition

Leaderboard probing

id	…	y
1	…	0
1	…	0
1	…	0
2	…	1
2	…	1
2	…	1

Private

Public

Leaderboard probing

Adapting global mean via LB probing:

$$-L * N = \sum_{i=1}^{N} (y_i \ln C + (1 - y_i) \ln (1 - C))$$

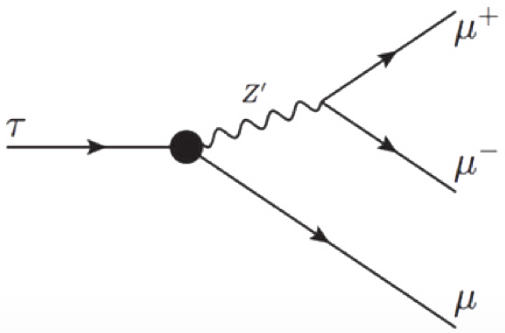
$$-L * N = N_1 \ln C + (N - N_1) \ln (1 - C))$$

$$\frac{N_1}{N} = \frac{-L - \ln (1 - C)}{\ln C - \ln (1 - C)}$$

Peculiar examples

Dato

Expedia®



Q

Truly Native

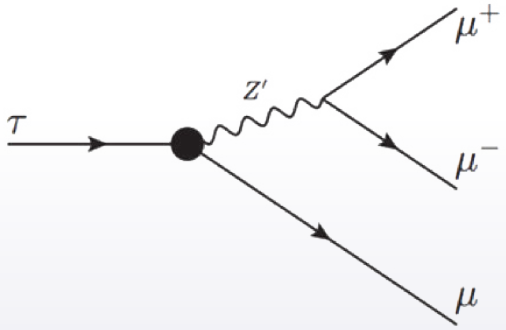
- Predict whether the content in an HTML file is sponsored or not
- Data leak in archive dates. But is it all?
  - Data collection
  - Date proxies

Expedia

- Predict hotel group a user is going to book
- Data leak in distance feature
- Reverse engineering true coordinates

Flavours of physics

- Machine learning problem for something that has never been observed
- Signal events were simulated
- Special statistical tests in order to punish the models that exploit simulation flaws
- However, one could by-pass the tests, fully exploit simulation flaws and get a perfect score on the leaderboard



Pairwise tasks

- Data leakage in item frequencies
- Similarities from connectivity matrix

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Endcard

Expedia Kaggle Competition

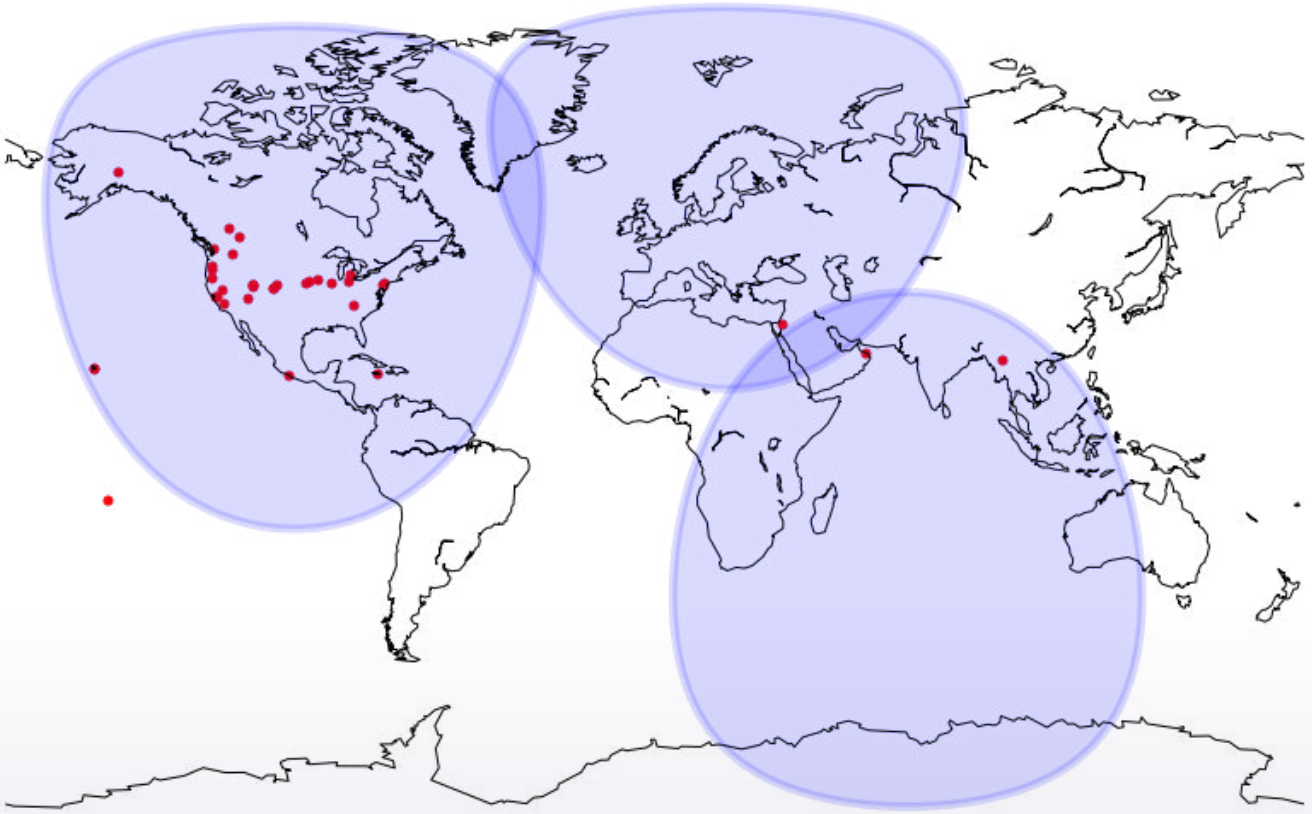
Data leakage

- destination_distance - user_city pair is a leak to true hotel location. A lot of matches between train and test.
- How to improve on that?
- Features based on counts on corteges of such nature
- Try to find the true coordinates

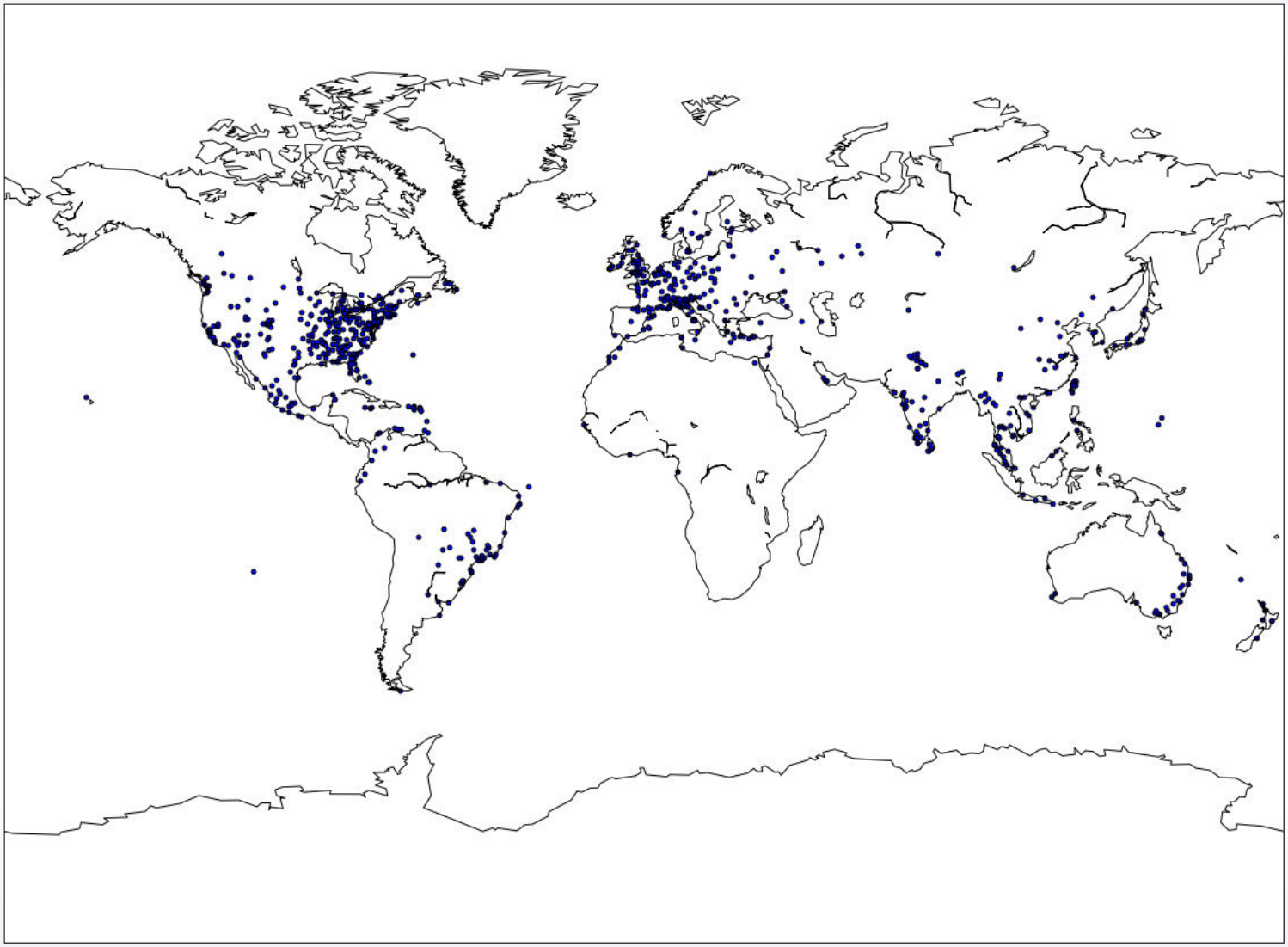
Spherical geometry

$$d = 2r \arcsin\left(\sqrt{\operatorname{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)\operatorname{hav}(\lambda_2 - \lambda_1)}\right)$$

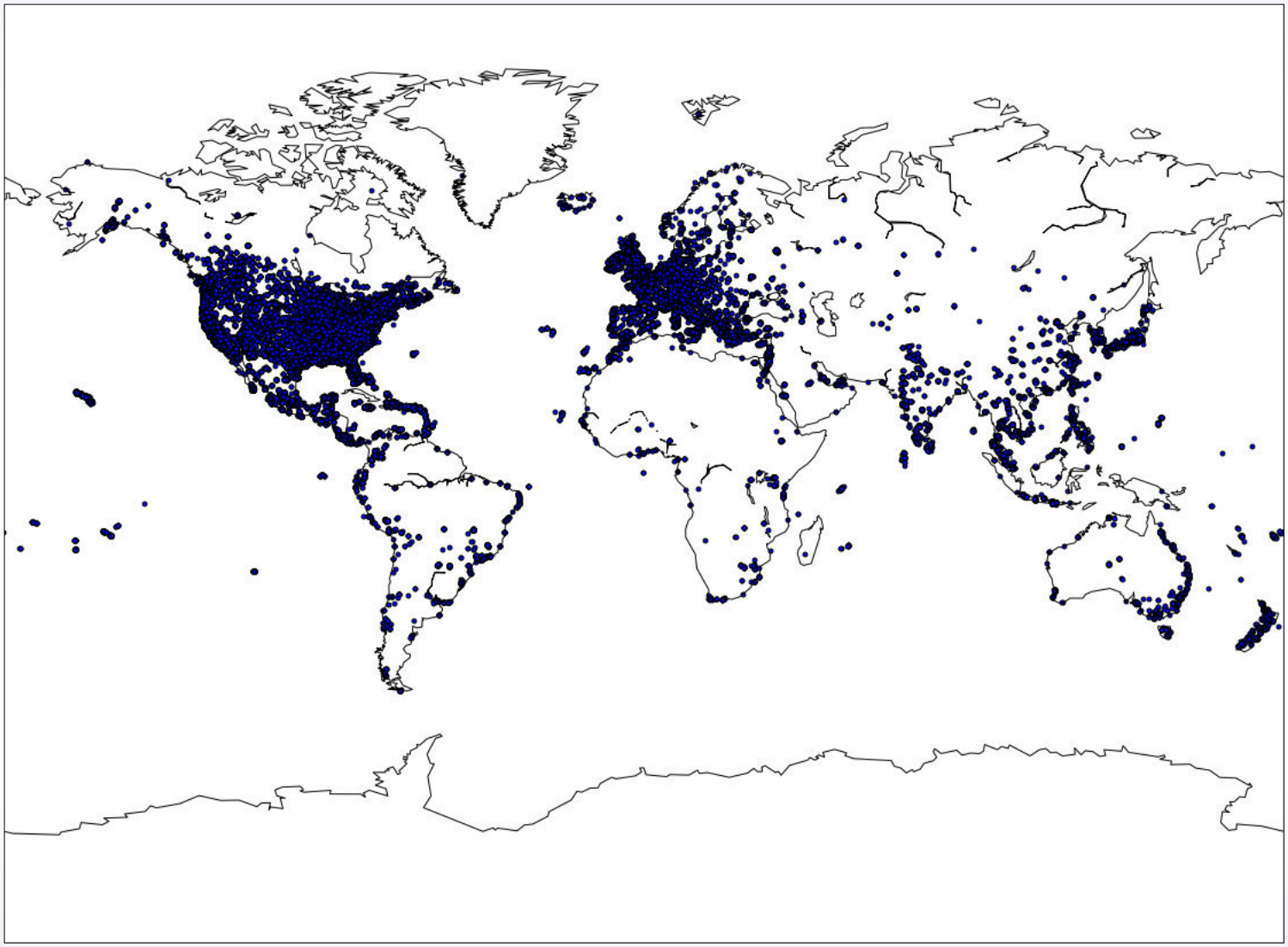
$$= 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$



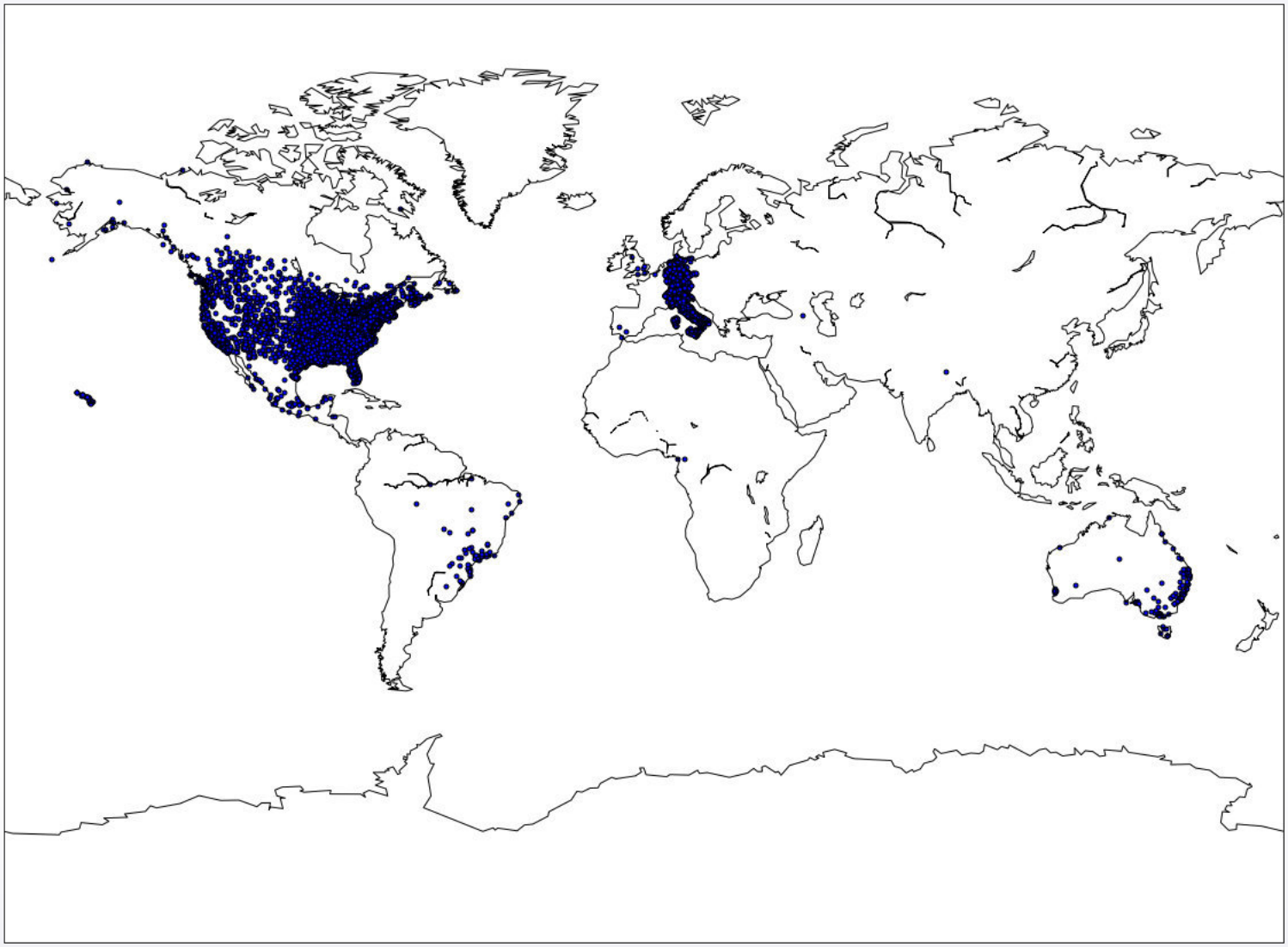
Hotel cities. Old version



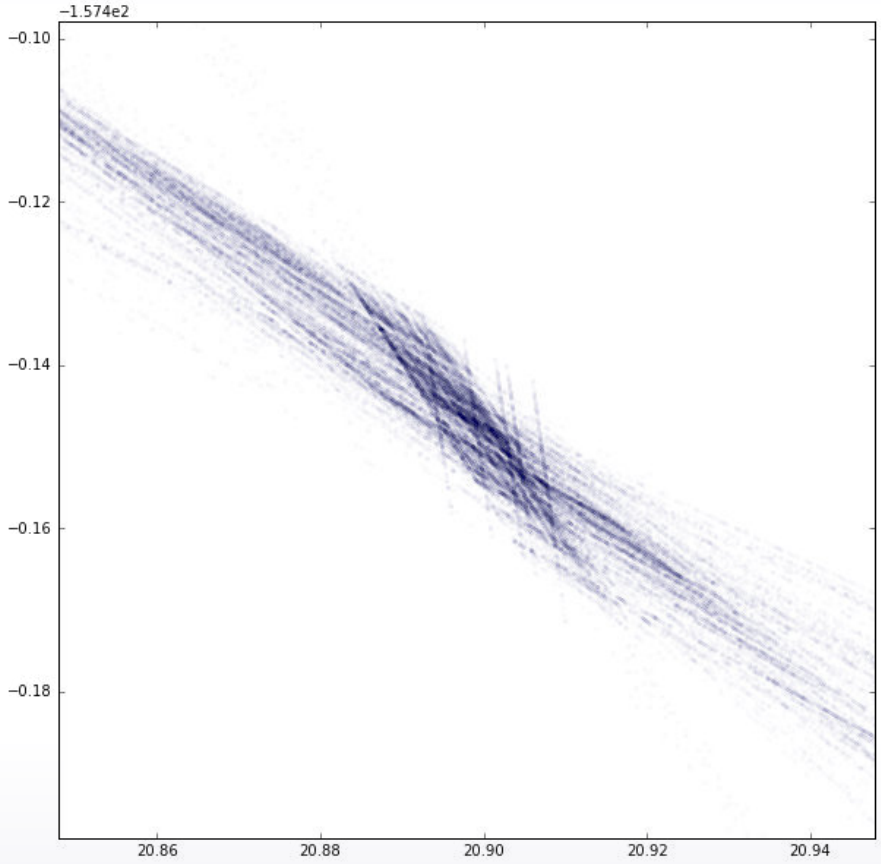
Hotels cities. New version

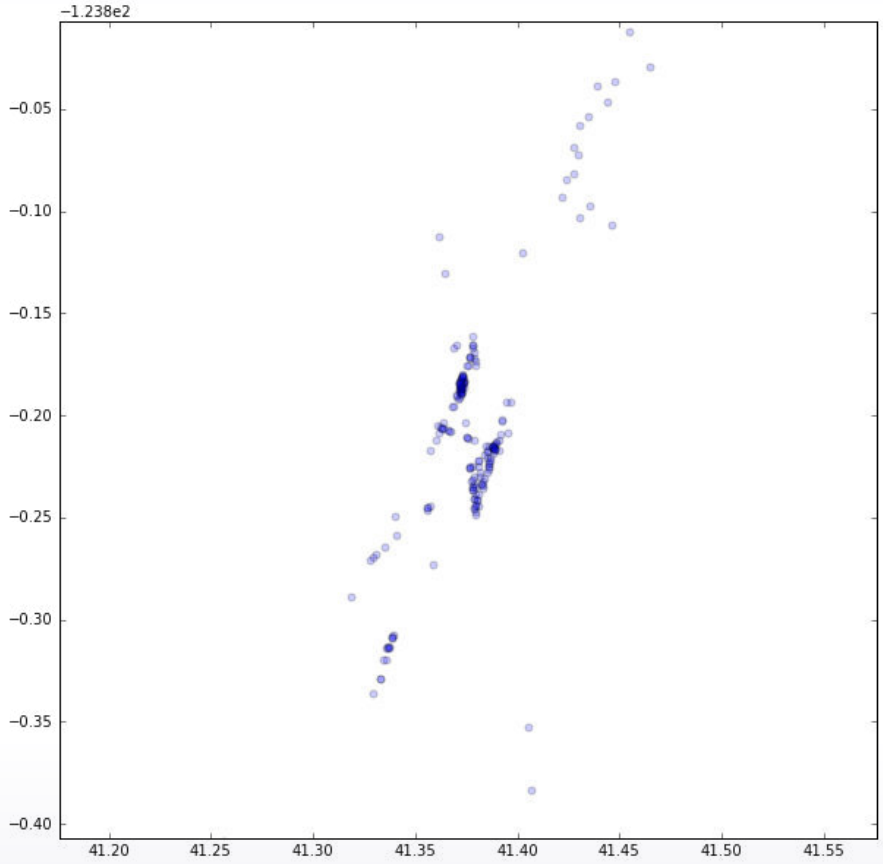


User cities. New version

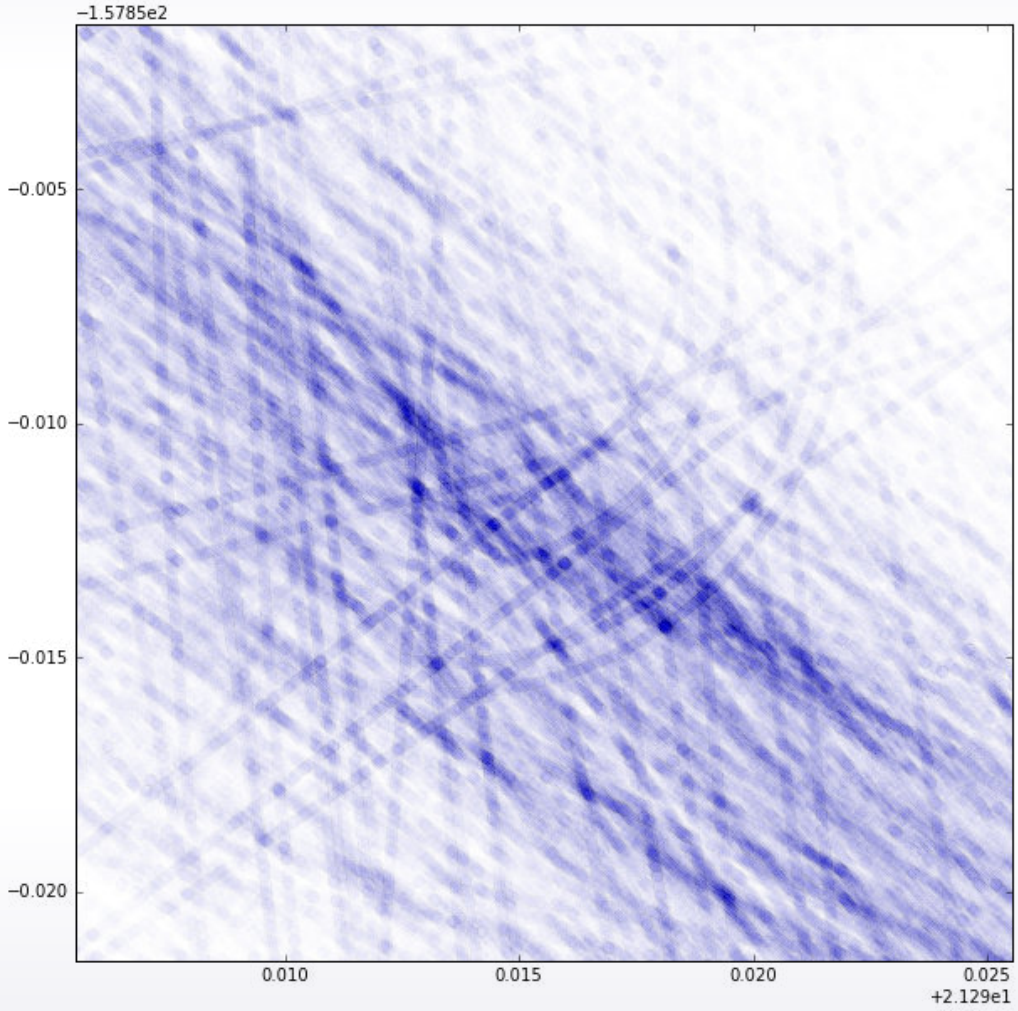


Trying to find the true coordinates of hotels (fail?)

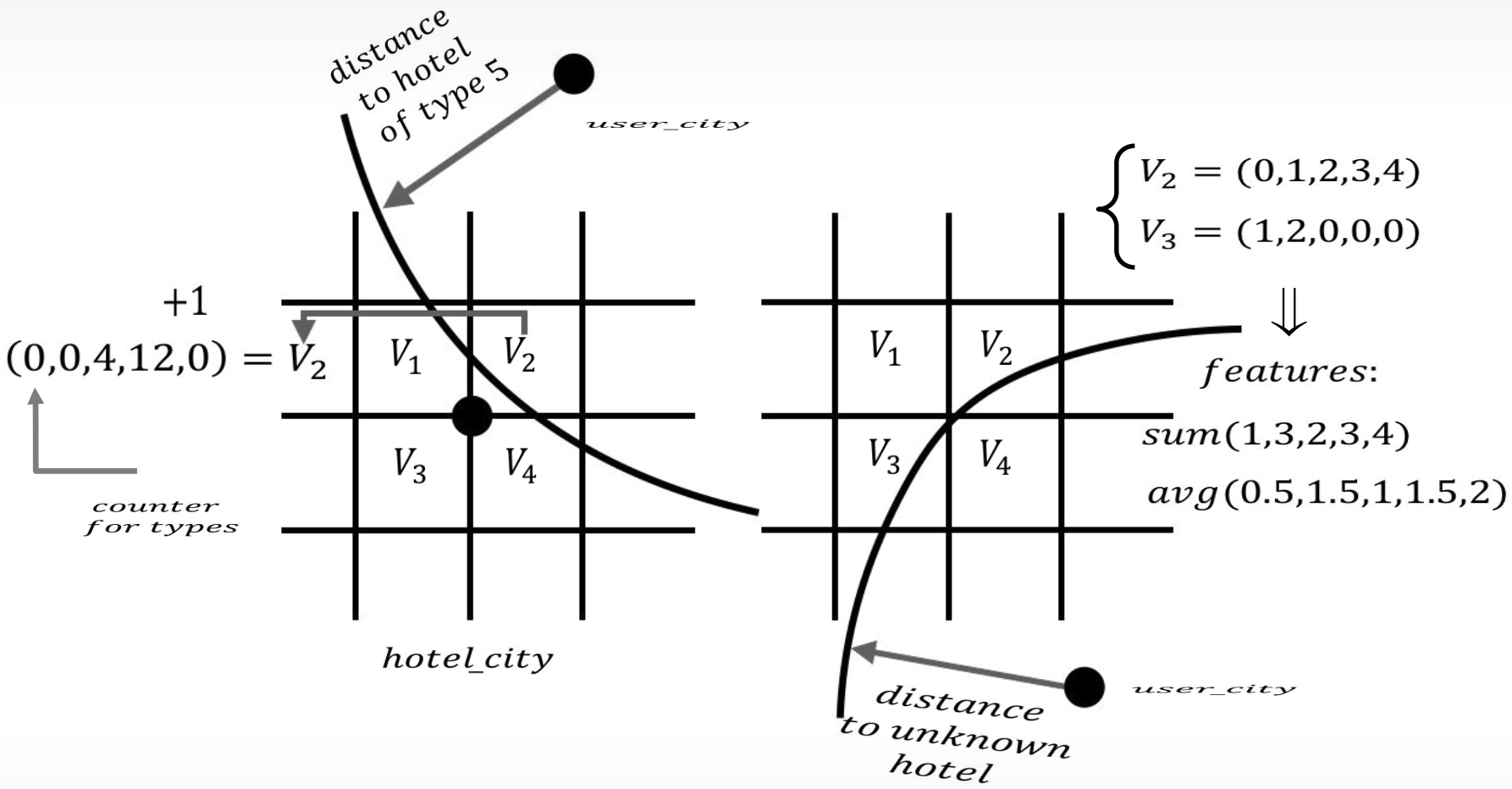




Trying to find the true coordinates of hotels (fail?)



Counters in grid cells



Final model

- Out-of-fold feature generation. 2013<->2014
- Xgboost
- 16 hours of training

Results

- Public – 3rd
- Private - 4th

#	△pub	Team Name	Kernel	Team Members	Score	Entries	Last
1	—	**idle_speculation**			0.60219	1	1y
2	—	**beluga**			0.53218	64	1y
3	▲1	**Victor**			0.53134	50	1y
4	▼1	**Ala Mode**			0.52995	26	1y