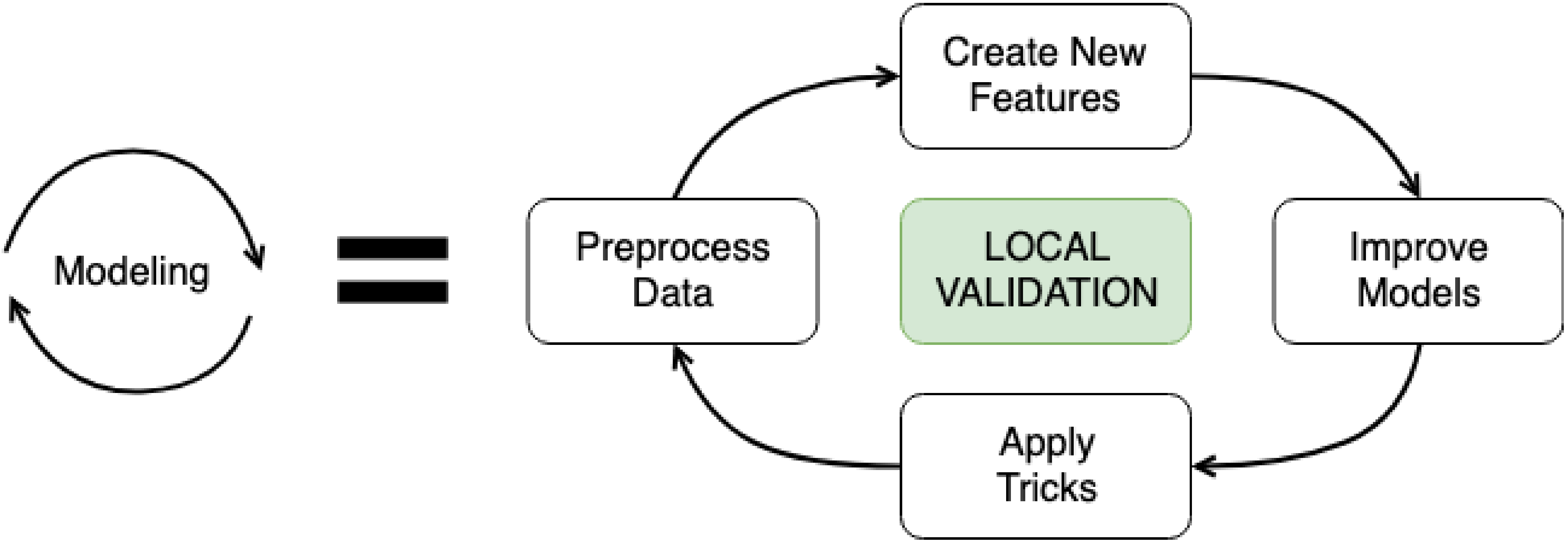
Baseline model

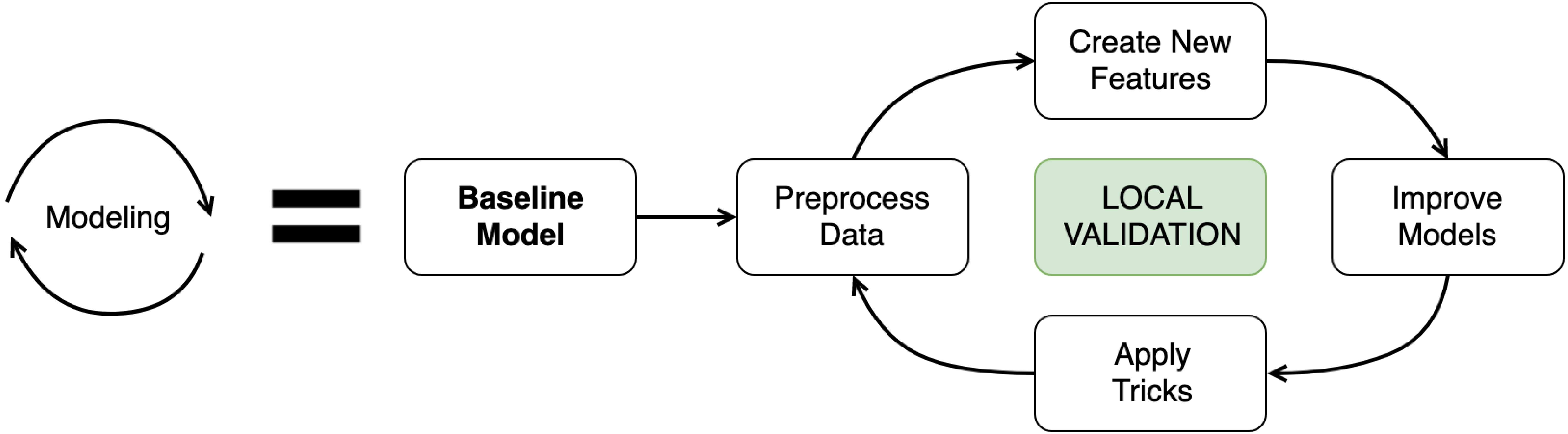
WINNING A KAGGLE COMPETITION IN PYTHON

Yauhen Babakhin
Kaggle Grandmaster

Modeling stage



Modeling stage



New York city taxi validation

```python
# Read data
taxi_train = pd.read_csv('taxi_train.csv')
taxi_test = pd.read_csv('taxi_test.csv')
```

```python
from sklearn.model_selection import train_test_split


# Create local validation
validation_train, validation_test = train_test_split(taxi_train,
                                                     test_size=0.3,
                                                     random_state=123)
```

Baseline model I

```
import numpy as np
# Assign the mean fare amount to all the test observations
taxi_test['fare_amount'] = np.mean(taxi_train.fare_amount)
# Write predictions to the file
taxi_test[['id','fare_amount']].to_csv('mean_sub.csv', index=False)
```

Validation RMSE	Public LB RMSE	Public LB Position
9.986	9.409	1449 / 1500

Baseline model II

```
# Calculate the mean fare amount by group
naive_prediction_groups = taxi_train.groupby('passenger_count').fare_amount.mean()
```

```
# Make predictions on the test set
taxi_test['fare_amount'] = taxi_test.passenger_count.map(naive_prediction_groups)
# Write predictions to the file
taxi_test[['id','fare_amount']].to_csv('mean_group_sub.csv', index=False)
```

Validation RMSE	Public LB RMSE	Public LB Position
9.978	9.407	1411 / 1500

Baseline model III

```python
# Select only numeric features
features = ['pickup_longitude', 'pickup_latitude',
            'dropoff_longitude', 'dropoff_latitude', 'passenger_count']
```

```python
from sklearn.ensemble import GradientBoostingRegressor

# Train a Gradient Boosting model
gb = GradientBoostingRegressor()
gb.fit(taxi_train[features], taxi_train.fare_amount)
# Make predictions on the test data
taxi_test['fare_amount'] = gb.predict(taxi_test[features])
```

Baseline model III

```
# Write predictions to the file
taxi_test[['id','fare_amount']].to_csv('gb_sub.csv', index=False)
```

Validation RMSE	Public LB RMSE	Public LB Position
5.996	4.595	1109 / 1500

Intermediate results

Model	Validation RMSE	Public LB RMSE
Simple Mean	9.986	9.409
Group Mean	9.978	9.407
Gradient Boosting	5.996	4.595

Correlation with Public Leaderboard

Model	Validation RMSE	Public LB RMSE
Model A	3.500	3.800
Model B	3.300	4.100
Model C	3.200	3.900

Model	Validation RMSE	Public LB RMSE
Model A	3.400	3.900
Model B	3.100	3.400
Model C	2.900	3.300

Let's practice!

WINNING A KAGGLE COMPETITION IN PYTHON

Hyperparameter tuning

WINNING A KAGGLE COMPETITION IN PYTHON

Yauhen Babakhin
Kaggle Grandmaster

Iterations

Model	Validation RMSE	Public LB RMSE	Public LB Position
Simple mean	9.986	9.409	1449 / 1500
Group mean	9.978	9.407	1411 / 1500
Gradient Boosting	5.996	4.595	1109 / 1500
Add hour feature	5.553	4.352	1068 / 1500
Add distance feature	5.268	4.103	1006 / 1500
...	...	...	...

Iterations

Model	Validation RMSE	Public LB RMSE	Public LB Position
Simple mean	9.986	9.409	1449 / 1500
Group mean	9.978		
Gradient Boosting	5.996	4.595	1109 / 1500
Add hour feature	5.553		
Add distance feature	5.268	4.103	1006 / 1500
...	...	...	...

Hyperparameter optimization

Competition type	Feature engineering	Hyperparameter optimization
Classic Machine Learning	+++	+
Deep Learning	-	+++

Ridge regression

Least squares linear regression

$$Loss = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \rightarrow \min$$

Ridge regression

Least squares linear regression

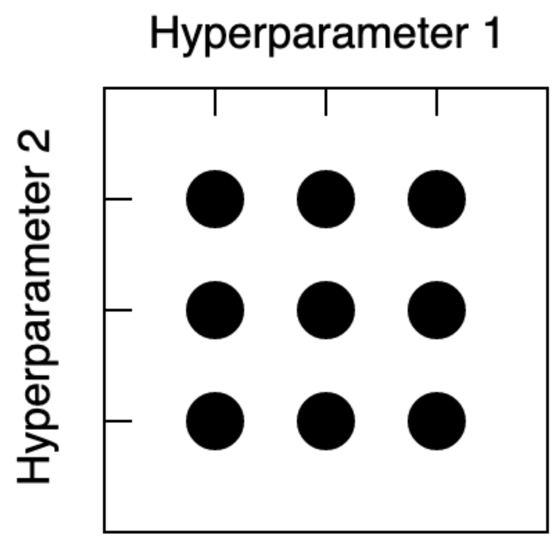
$$Loss = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \rightarrow \min$$

Ridge regression

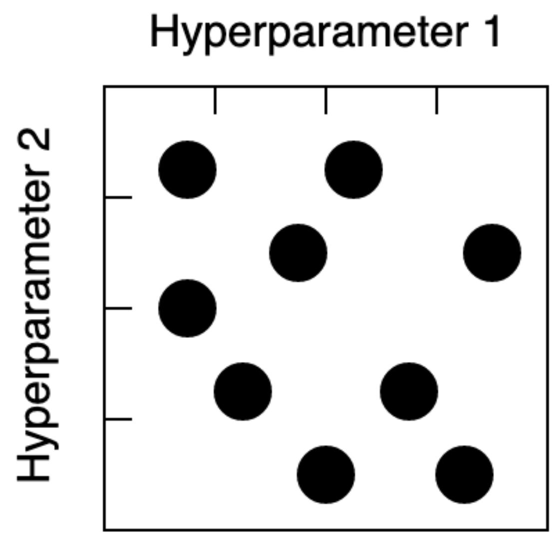
$$Loss = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^{K} {w_j}^2 \rightarrow \min$$

Hyperparameter optimization strategies

- **Grid search.** Choose the predefined grid of hyperparameter values
- **Random search.** Choose the search space of hyperparameter values
- **Bayesian optimization.** Choose the search space of hyperparameter values



Grid search



Random search

Grid search

```
# Possible alpha values
alpha_grid = [0.01, 0.1, 1, 10]
from sklearn.linear_model import Ridge
results = {}
# For each value in the grid
for candidate_alpha in alpha_grid:
    # Create a model with a specific alpha value
    ridge_regression = Ridge(alpha=candidate_alpha)
    # Find the validation score for this model
    # Save the results for each alpha value
    results[candidate_alpha] = validation_score
```

Let's practice!

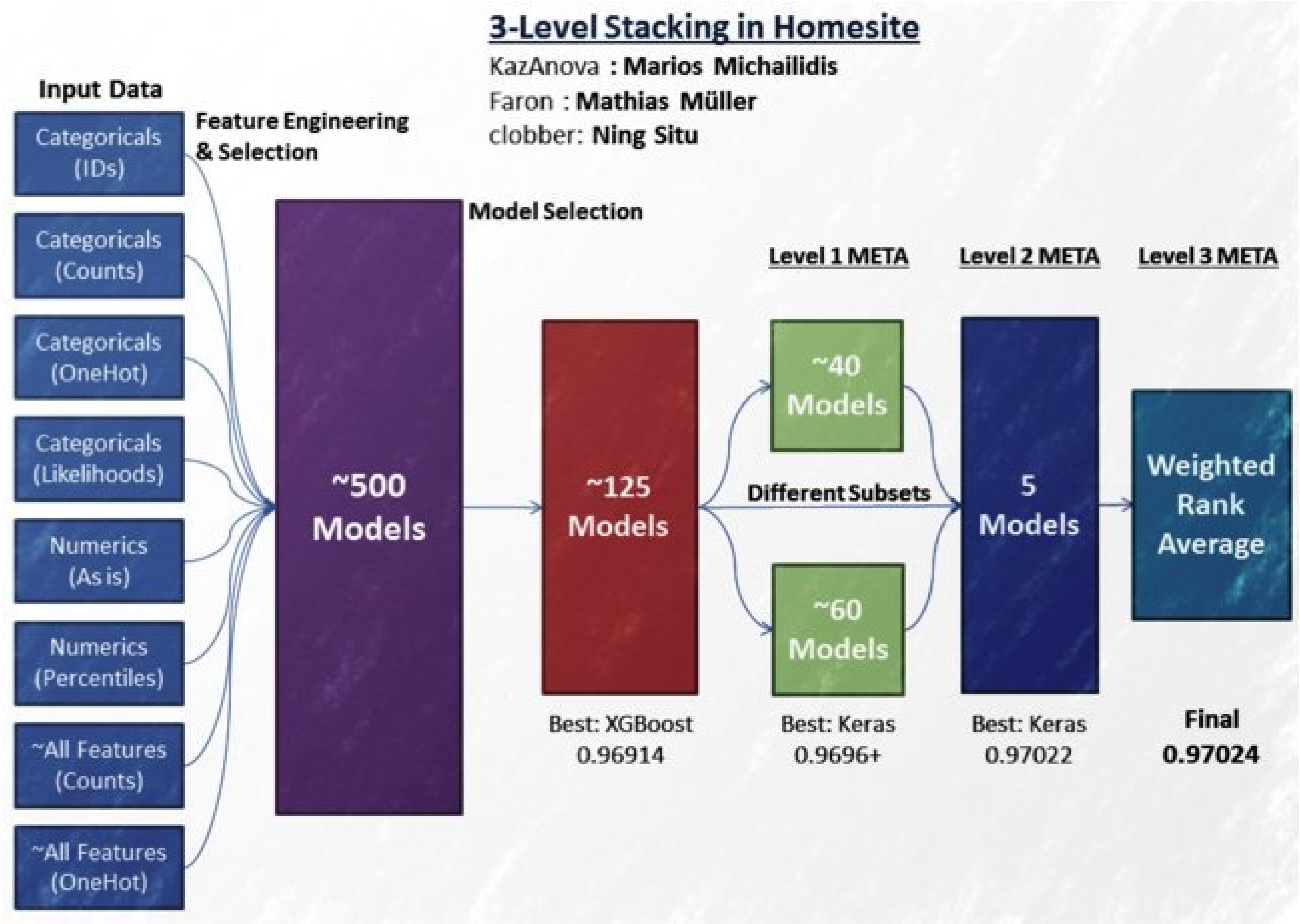
WINNING A KAGGLE COMPETITION IN PYTHON

Model ensembling

WINNING A KAGGLE COMPETITION IN PYTHON

Yauhen Babakhin
Kaggle Grandmaster

Model ensembling



Model blending

- Regression problem
- Train two different models: A and B
- Make predictions on the test data:

Test ID	Model A prediction	Model B prediction
1	1.2	1.5
2	0.1	0.4
3	5.4	7.2

Model blending

Test ID	Model A prediction	Model B prediction	Arithmetic mean
1	1.2	1.5	1.35
2	0.1	0.4	0.25
3	5.4	7.2	6.30

Model blending

Arithmetic mean

$$arithmetic = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Geometric mean

$$geometric = \left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}}$$

Model stacking

1. Split train data into two parts
2. Train multiple models on Part 1
3. Make predictions on Part 2
4. Make predictions on the test data
5. Train a new model on Part 2 using predictions as features
6. Make predictions on the test data using the 2nd level model

Stacking example

Train ID	feature_1	...	feature_N	Target
1	0.55	...	1.37	1
2	0.12	...	-2.50	0
3	0.65	...	3.14	0
4	0.10	...	2.87	1
5	0.54	...	-0.10	0

Test IDs	feature_1	...	feature_N	Target
11	0.49	...	-2.32	?
12	0.32	...	1.15	?
13	0.91	...	0.81	?

Stacking example

Train ID	feature_1	...	feature_N	Target
1	0.55	...	1.37	1
2	0.12	...	-2.50	0
3	0.65	...	3.14	0

Train ID	feature_1	...	feature_N	Target
4	0.10	...	2.87	1
5	0.54	...	-0.10	0

Stacking example

Train ID	feature_1	...	feature_N	Target
1	0.55	...	1.37	1
2	0.12	...	-2.50	0
3	0.65	...	3.14	0

Train ID	feature_1	...	feature_N	Target
4	0.10	...	2.87	1
5	0.54	...	-0.10	0

Train models A, B, C on Part 1

Stacking example

Train ID	feature_1	...	feature_N	Target	A_pred	B_pred	C_pred
4	0.10	...	2.87	1	0.71	0.52	0.98
5	0.54	...	-0.10	0	0.45	0.32	0.24

Test IDs	feature_1	...	feature_N	Target	A_pred	B_pred	C_pred
11	0.49	...	-2.32	?	0.62	0.45	0.81
12	0.32	...	1.15	?	0.31	0.52	0.41
13	0.91	...	0.81	?	0.74	0.55	0.92

Stacking example

Train ID	Target	A_pred	B_pred	C_pred
4	1	0.71	0.52	0.98
5	0	0.45	0.32	0.24

Test IDs	Target	A_pred	B_pred	C_pred
11	?	0.62	0.45	0.81
12	?	0.31	0.52	0.41
13	?	0.74	0.55	0.92

Stacking example

Train ID	Target	A_pred	B_pred	C_pred
4	1	0.71	0.52	0.98
5	0	0.45	0.32	0.24

Test IDs	Target	A_pred	B_pred	C_pred
11	?	0.62	0.45	0.81
12	?	0.31	0.52	0.41
13	?	0.74	0.55	0.92

Train 2nd level model on Part 2

Stacking example

Train ID	Target	A_pred	B_pred	C_pred
4	1	0.71	0.52	0.98
5	0	0.45	0.32	0.24

Test IDs	Target	A_pred	B_pred	C_pred	Stacking prediction
11	?	0.62	0.45	0.81	**0.73**
12	?	0.31	0.52	0.41	**0.35**
13	?	0.74	0.55	0.92	**0.88**

Let's practice!

WINNING A KAGGLE COMPETITION IN PYTHON

Final tips

WINNING A KAGGLE COMPETITION IN PYTHON

Yauhen Babakhin
Kaggle Grandmaster

Save information

1. Save folds to the disk
2. Save model runs
3. Save model predictions to the disk
4. Save performance results

Kaggle forum and kernels

Kaggle forum and kernels

Kaggle forum

- Competition discussion by the participants

Kaggle forum and kernels

Kaggle forum

- Competition discussion by the participants

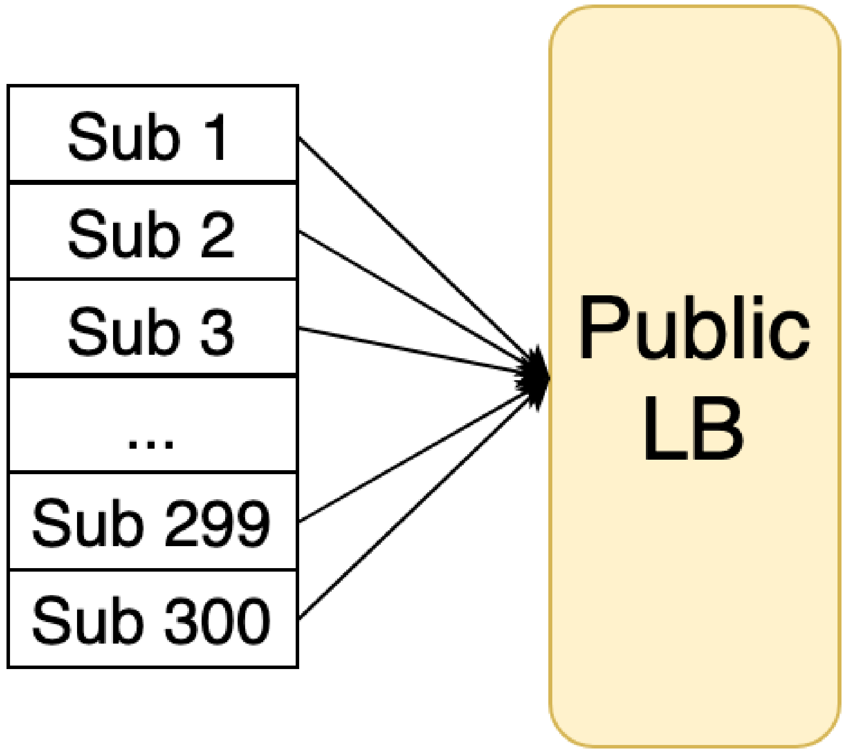
Kaggle kernels

- Scripts and notebooks shared by the participants
- Cloud computational environment

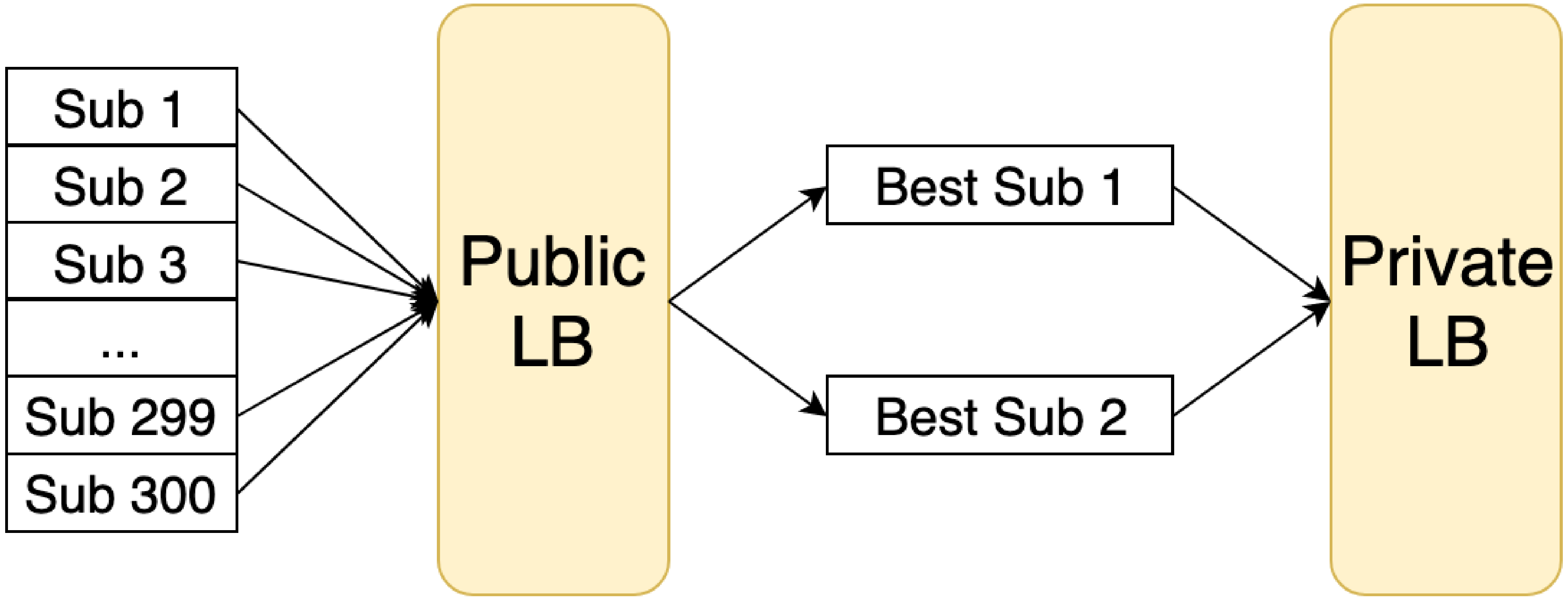
Forum and kernels usage

When?	Forum	Kernels
Before the competition	Read winners' solutions from the past similar competitions	Go through baseline approaches from the past similar competitions
During the competition	Follow the discussion to find the ideas and approaches for the problem	Look at EDA, baseline models and validation strategies used by others
After the competition	Read winners' solutions	Look at the final solutions code sharing

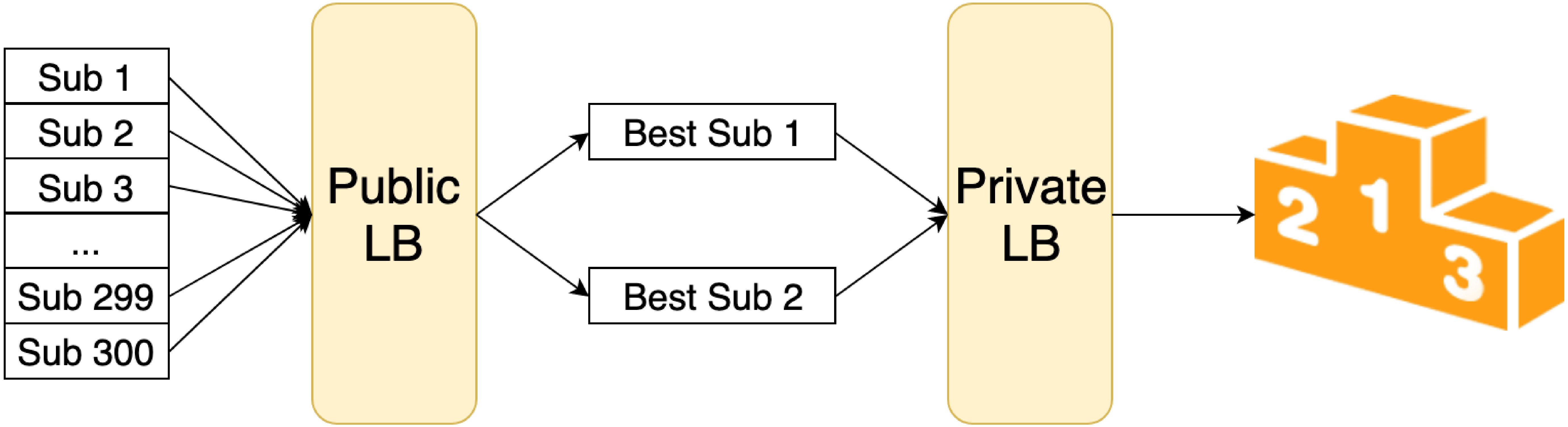
Select final submissions



Select final submissions



Select final submissions



Final submissions

1. Best submission on the local validation
2. Best submission on the Public Leaderboard

Let's practice!

WINNING A KAGGLE COMPETITION IN PYTHON

Final thoughts

WINNING A KAGGLE COMPETITION IN PYTHON

Yauhen Babakhin
Kaggle Granmaster

What we've learned

- What is Kaggle
- Understand the problem
- Make EDA
- Develop local validation
- Generate new features
- Build model ensembles

Kaggle vs Data Science

Kaggle vs Data Science

Data analytics

- Kaggle does not help here

Kaggle vs Data Science

Data analytics

- Kaggle does not help here

Machine learning models

1. Talk to Business. Define the problem
2. Collect the data
3. Select the metric
4. Make train and test split
5. Create the model
6. Move model to the production

kaggle™

Start competing on Kaggle!

WINNING A KAGGLE COMPETITION IN PYTHON