zalando

KAGGLE HOME DEPOT SEARCH RELEVANCE COMPETITION

LESSONS LEARNED

Dr. ANDREAS MERENTITIS
SENIOR DATA SCIENTIST

09-09-2016

Outline

- Background
  - What is Kaggle?
  - Why participate in / host contests?
- Home Depot problem
- Our Model
  - Feature engineering
  - Training methods
  - Best simple model
- Summary
  - Important findings
  - Kaggle recipe

Kaggle is the world's largest community of data scientists.

- They compete with each other to solve complex data science problems
- Thousands of PhDs from industries such as insurance, finance, and technology (from 100 countries and 200 universities)
- Many tens of thousands of registered accounts, out of which a few hundreds are “masters”

Background - The typical setup in a nutshell

kaggle

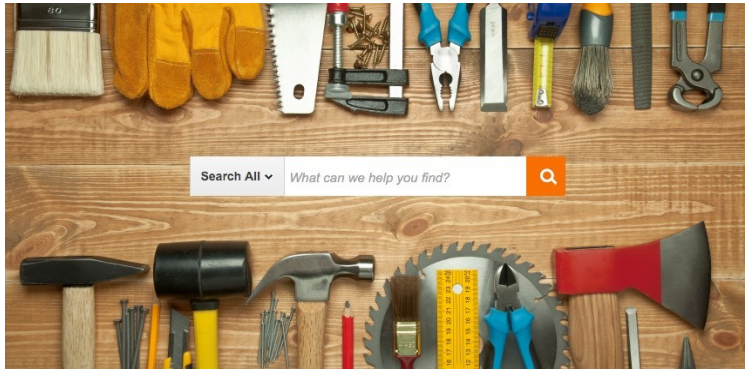
Companies host challenges to get some valuable insights:

- How good can we get with the current data without a breakthrough in theory?
- How does the curve of diminishing returns look?
- What is the single best model and what are the top features?

People play in Kaggle for three main reasons:

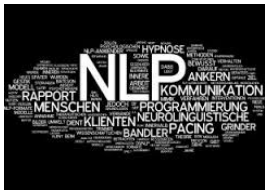
- Mostly to learn. Especially advanced cross-validation approaches and extreme ensembling are key areas for learning
- Interact with the data science community, increase your visibility
- Win money (well not so much really)

The Home - Depot Problem

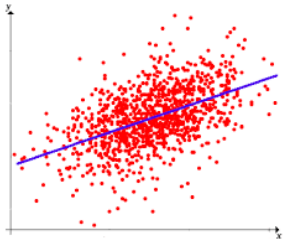


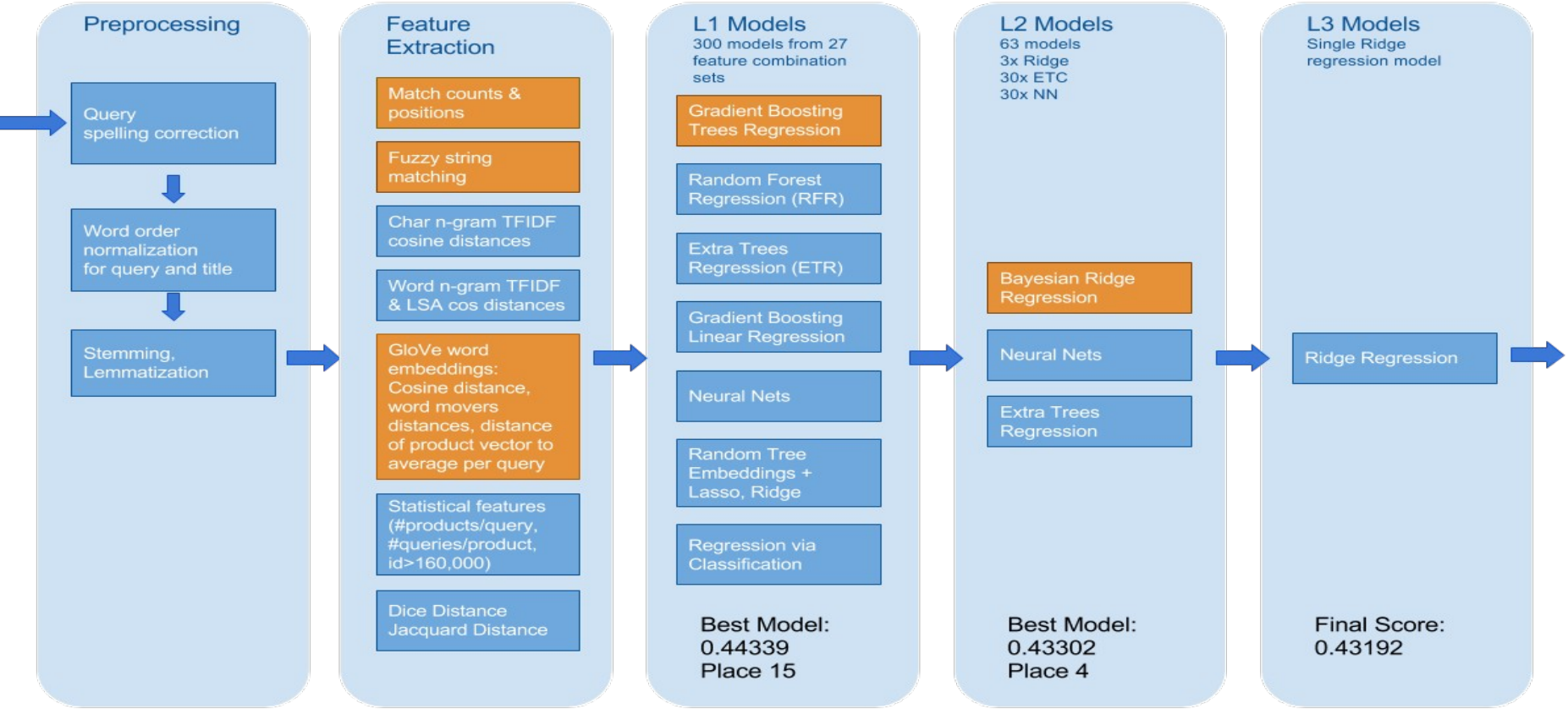
=

+



=




Preprocessing
Query spelling correction
Word order normalization for query and title
Stemming, Lemmatization
Feature Extraction
Match counts & positions
Fuzzy string matching
Char n-gram TFIDF cosine distances
Word n-gram TFIDF & LSA cos distances
GloVe word embeddings: Cosine distance, word movers distances, distance of product vector to average per query
Statistical features (#products/query, #queries/product, id>160,000)
Dice Distance Jacquard Distance
L1 Models
300 models from 27 feature combination sets
Gradient Boosting Trees Regression
Random Forest Regression (RFR)
Extra Trees Regression (ETR)
Gradient Boosting Linear Regression
Neural Nets
Random Tree Embeddings + Lasso, Ridge
Regression via Classification
Best Model:
0.44339
Place 15
L2 Models
63 models
3x Ridge
30x ETC
30x NN
Bayesian Ridge Regression
Neural Nets
Extra Trees Regression
Best Model:
0.43302
Place 4
L3 Models
Single Ridge regression model
Ridge Regression
Final Score:
0.43192

Challenges:

- Misspellings
- Non-distinctive word matches (and, the, in, on, ...)
- Irrelevant products with matching attributes
- Semantic Similarity
- Ambiguous queries
- Exploiting systemic bias

Feature Engineering

Challenge 1:	**Misspellings**
Approach:	**Fuzzy Match**

"*refrigerator*" misspellings

refrigirator	refrigeratore
refrdgerators	refrig
refridegrator	refridgetor

refrigirator

(refr) (efri) (frig) (rigi) (igir) …

Feature Engineering

Challenge 2: **Non-distinctive word matches (and, the, in, on, …)**

Approach: **TF-IDF weighted similarity**

'Delta Vero 1-Handle Shower Only Faucet Trim Kit in Chrome’

Delta	Vero	1-Handle	shower	only	faucet	Trim	Kit	In	Chrome
5.88	9.12	4.24	4.17	3.12	4.03	3.56	3.66	1.14	4.19

Feature Engineering

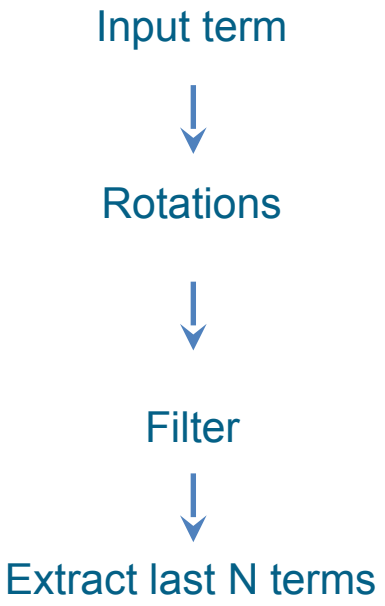
Challenge 3: **Irrelevant products with matching attributes**

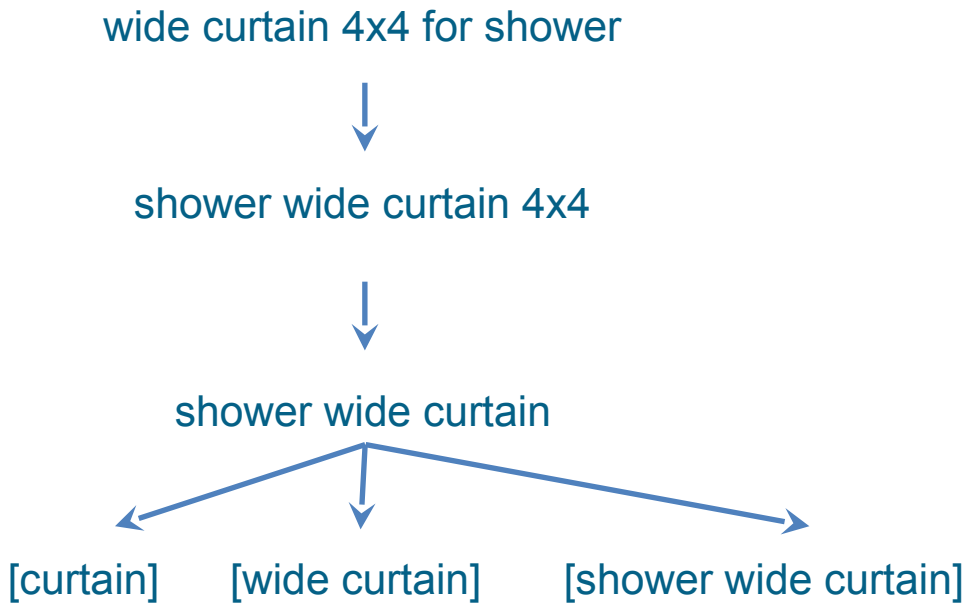
Approach: **Extract noun terms in product descriptions**

For	homeowners	seeking	a	lawn	mower	with	high-quality	,	a	user-friendly	design	and	excellent	mulching	capability.
IN	NNS	VBG	DT	NN	NN	IN	JJ	,	DT	JJ	NN	CC	NN	VBG	NN

Challenge 3: **Irrelevant products with matching attributes**

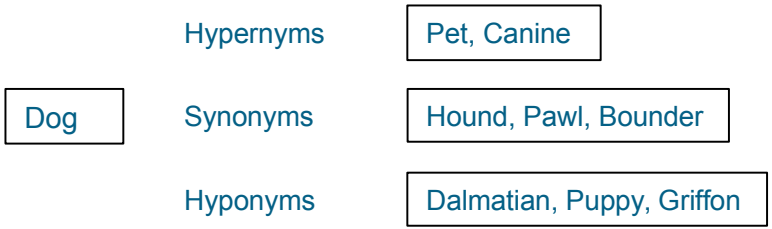
Approach: **Main Entity Detection in search term and product titles**





Challenge 4: **Semantic Similarity**

Approach: **Vocabulary Expansion**



Challenge 4: **Semantic Similarity**

Approach: **Matrix Decomposition (LSA, NMF)**

words

	"Door"	"Lock"	"Mower"	. . .
Product 1	1	1	0	. . .
Product 2	0	0	0	. . .
Product 3	0	0	1	. . .
. . .	. . .	. . .	. . .	

topics

	Furniture	Gardening	Pets	. . .
Product 1	0.98	0.23	0.03	. . .
Product 2	-0.32	0.34	1.56	. . .
Product 3	-1.34	2.08	-0.41	. . .
. . .	. . .	. . .	. . .	. . .

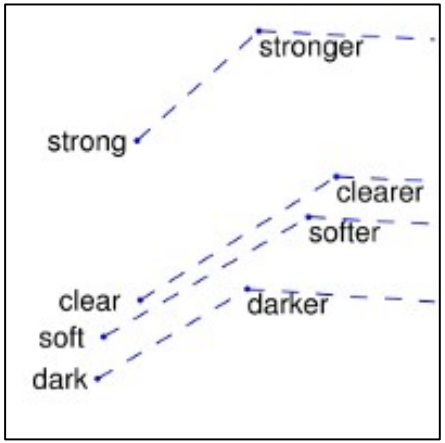
N x D

N x K (K << D)

Challenge 4: **Semantic Similarity**

Approach: **Word Embeddings**

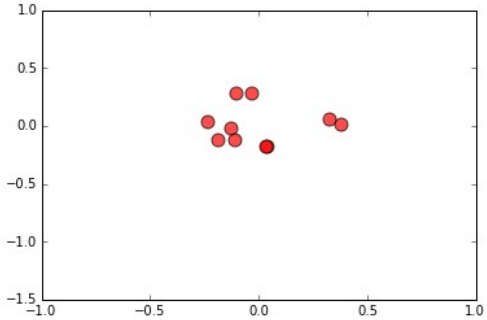
Word Embeddings using GloVe



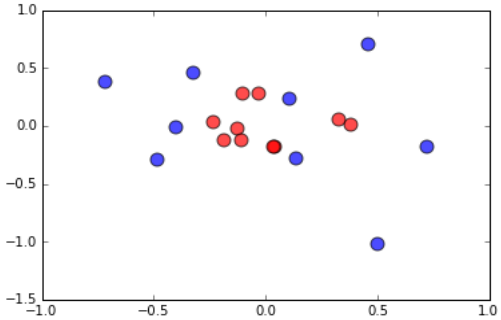
“stronger” – “strong” + “soft” ≈ “softer”

Semantic Representation of Products

“Steel Shelving”

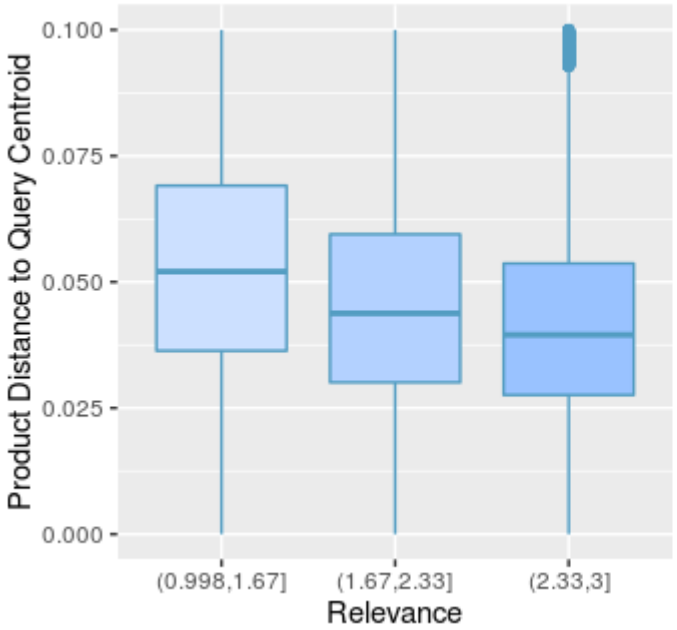


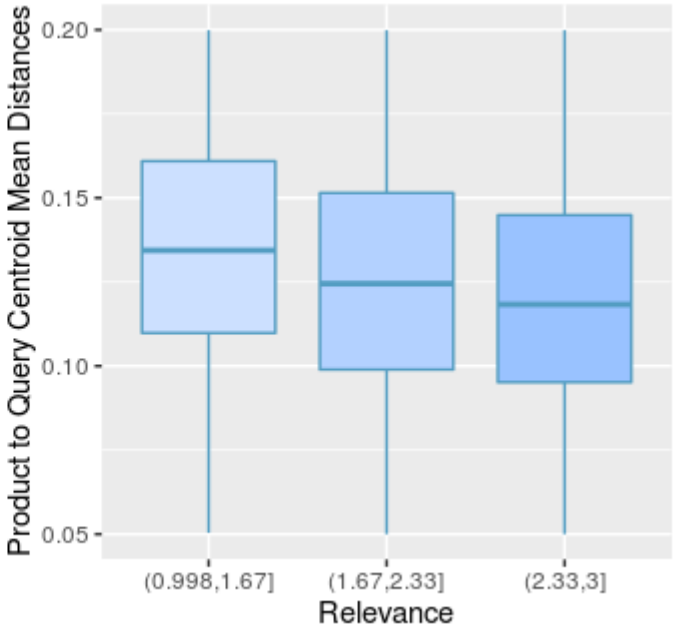
“Manual”



Challenge 4: **Semantic Similarity**

Approach: **Word Embeddings**

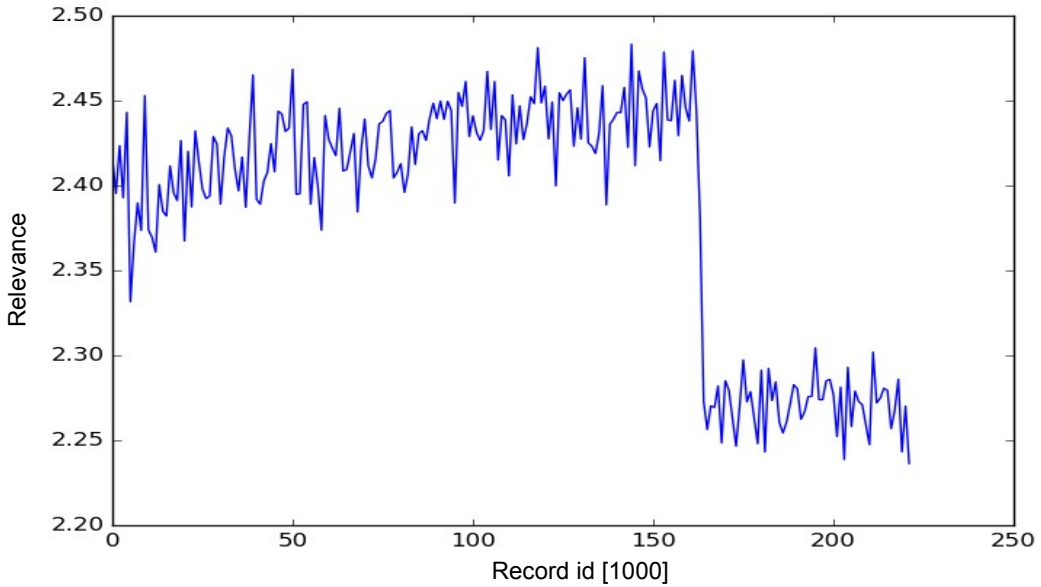




Challenge 5: **Exploiting systematic bias**

Approach: **Visual inspection, indicator function**

- Indicator function for record ids > 160,000



We used ~300 layer-one models, trained on different feature subsets.

- Generalized Linear Models where approximately half of these,
- Most of the others where XGB, ETR, and ETC.

The ensembling was based on a stacking approach:

- Layer-two stacking with Bayesian Ridge, 10X bagged Neural Net and 10X bagged ETR,
- Trained for each layer-one CV run.

Final weighting of the models was done by layer-three stacking with ridge regression.

We applied different combinations of feature transformations and embeddings

Three types of transformations were applied:

- Sparse Random Projections
- Gaussian Projections
- Random Tree Embeddings (RTE)

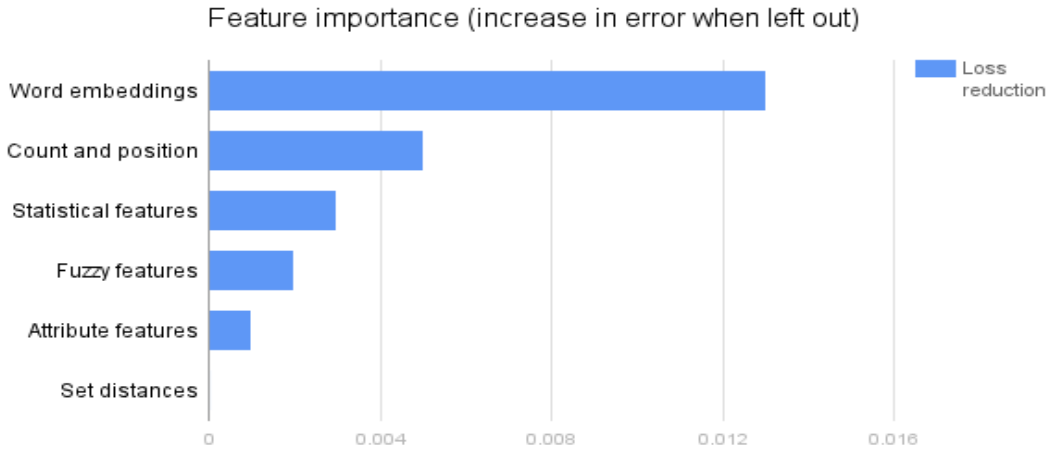
All three helped in reducing correlation between layer-one models

Especially RTE coupled with Lasso and Bayesian Ridge gave some of the best models

Best single model 1/2

Our best single model got a score of 0.44339 on private test set, enough for place 15.

The model is trained with xgboost gradient boosting trees, based on following features:



Training of the model can be done in 10-30 minutes, and predicting takes a few seconds.

Count, fuzzy and word embedding features are necessary to get close to the maximum score for a single model

Word embedding features are needed to get below 0.455 - but perform poorly by themselves (~0.47)

Adding attribute features creates only a small boost

Finding a good cross-validation strategy was difficult but very important in this competition

k-fold cross-validation performs badly on this dataset

- records represent query-product pairs
- multiple product candidates are matched with the same query

This imposes a risk of overfitting:

- algorithms can identify queries from some features
- bias the prediction towards the average relevance for those queries
- we want instead to learn a function predicting relevance for arbitrary pairings

Cross-validation 2/2

We needed an alternative strategy that better reflects the two scenarios in the cross-validation folds:

- 57% of queries in test set did not appear in train set
- 72% of product ids in test set did not appear in train set

To reflect scenario A:

- we split the queries in the train set randomly into three equally sized sets
- collected the corresponding query-product pairs to form the cross-validation folds

To reflect scenario B, we did the same on product uids.

Overfitting was stronger for queries, thus:

- two runs of scenario A using different random seeds
- one run of scenario B
- total of three runs times three folds for cross-validation

We used the mean RMSE across all folds to validate our model.

As indicated before, having a solid cross-validation strategy was critical for this contest

Getting multiple uncorrelated models was key:

- Variety in features
- different feature transformations
- different regression algorithms

Beyond these general remarks, some lessons learned:

- Parsing query and title structure is important
- Word embeddings outperform topic modeling
- Product candidate distribution per query carries information
- Systematic biases/data preparation leakage

There are typically two main types of contest, those that are image or audio related and those that are not

For images (and to some extent audio) deep learning dominates, for all others Xgboost

Cross-validation and ensembling is critical on both types

It really pays to have a good CV strategy that reliably predicts your model's performance before submitting to public leaderboard

To gain most from ensembling you need models that are good on their own and as uncorrelated as possible

Automated unsupervised feature transformations are key for reducing correlation between models

Different loss functions help reduce correlation while keeping outliers in check

Tuning of hyperparameters and feature selection are often less important for the first layer models than in the typical one-model ML case

In every stage of the meta-ensemble have at least some models that optimize for the metric of the contest directly when possible (Xgboost)

Play in teams of 2-5 and have fun!

Special thanks to my team members in the contest:

- Alexander Bauer
- Nurlanbek Duishoev