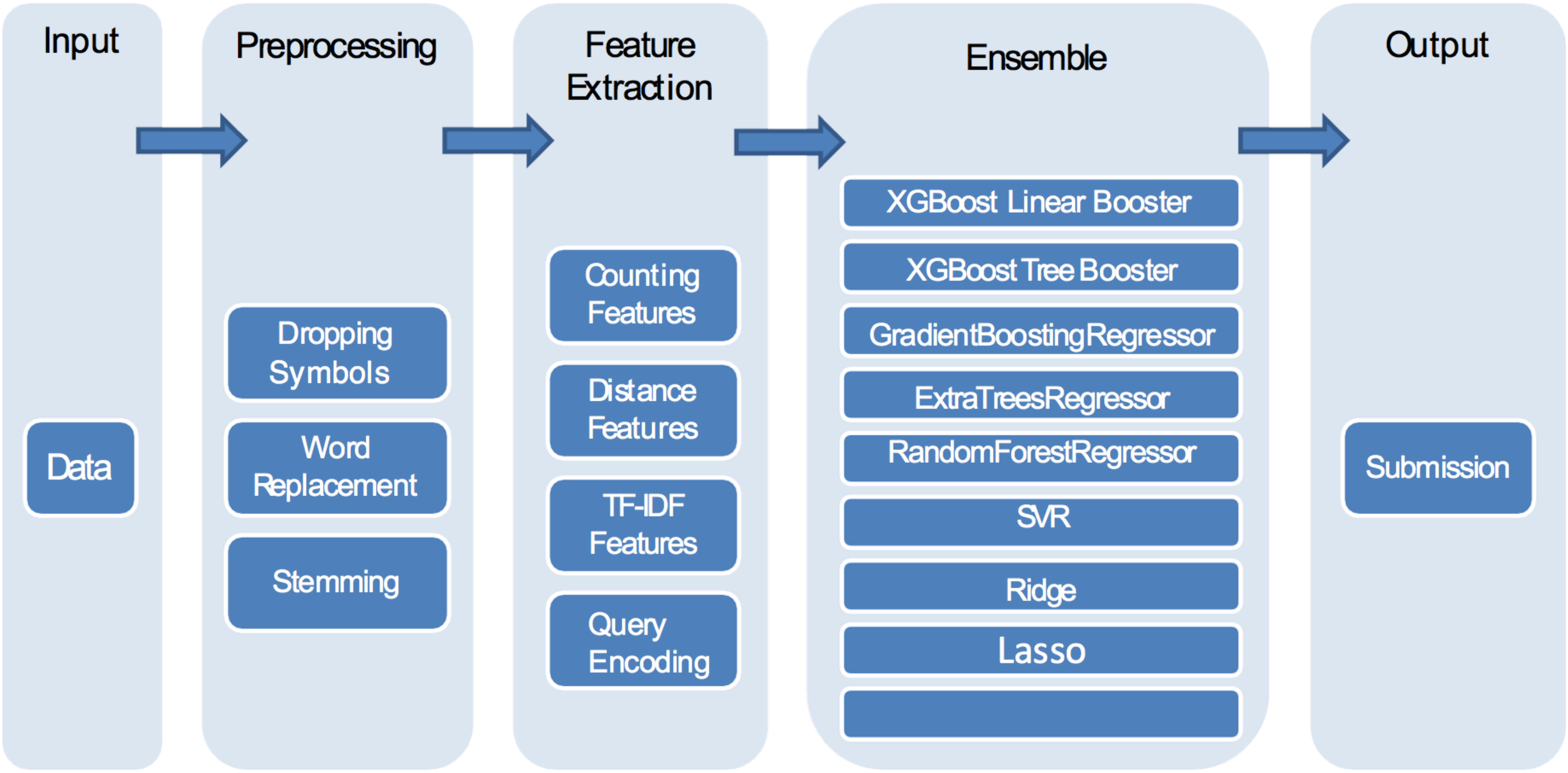
# KAGGLE: HOME DEPOT SEARCH RELEVANCE

# OVERVIEW

- Tasks: Given product title and product description, predict the relative relevance score for the search query and the product
- Example:
  - 2,100001,"Simpson Strong-Tie 12-Gauge Angle","angle bracket",3
  - 3,100001,"Simpson Strong-Tie 12-Gauge Angle","l bracket",2.5
- Relevance Score is slightly right skewed distributed
- Either regression problem or classification problem.
- Reason: NLP problem, less computational burden, challenging
- Time: 3 weeks, 2 weeks for feature engineering
- Results: 25% of total 2125 teams, RMSE 0.47214, Top RMSE 0.43192



Input

Data

Preprocessing

Dropping Symbols

Word Replacement

Stemming

Feature Extraction

Counting Features

Distance Features

TF-IDF Features

Query Encoding

Ensemble

XGBoost Linear Booster

XGBoost Tree Booster

GradientBoostingRegressor

ExtraTreesRegressor

RandomForestRegressor

SVR

Ridge

Lasso

Output

Submission

# STRUGGLING

- Optimistic to this competition
  - Material
  - Many great ideas
- First entry scored 0.485 with only 20 features
  - Basic preprocessing
  - Counting features and distance feature (key word matching)
  - Random forest regressor without fine tune.
- More we tried, lower score we received
  - Add many fancy features e.g. colors, brands, materials
- Computational burden
- Coding ability
  - word2vec

# EDUCATIONAL

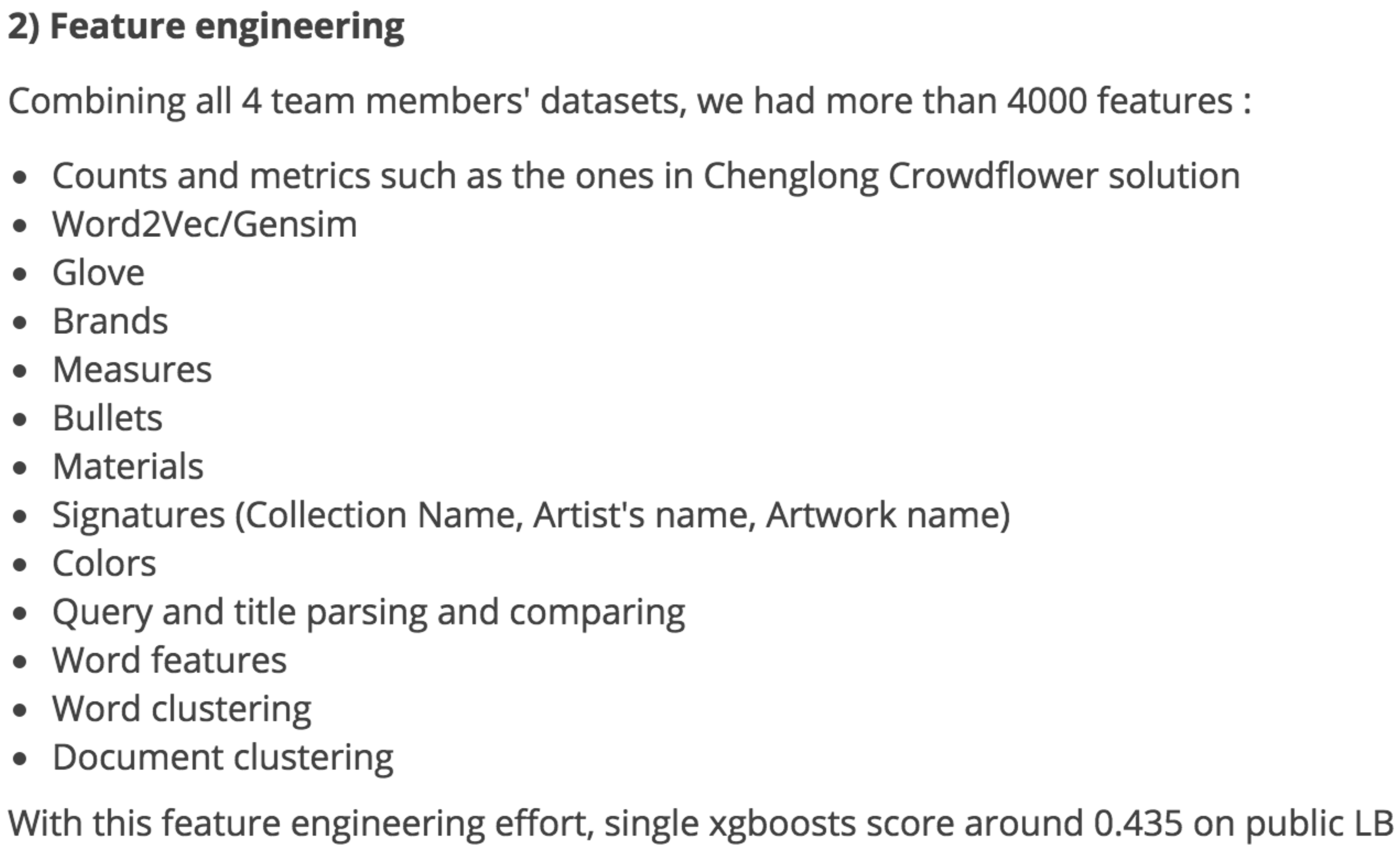
- Taking preprocessing more seriously (200 lines of code)
  - Stemming
  - Check misspelling (huge misspelling dictionary by google API)
  - Synonym replacement
  - Gives 0.05 increase
- Code management (crowdflower winner solution)
  - Separate scripts for functions, generating features and modelling
  - Better for team to communicate
  - Wrap up the whole thing as a pipeline
    - Run script on external services
    - Feature union
    - Grid search on tune model
    - Save physical memory
- Start to write your own functions and class
  - Customize metric function

# THINK MORE..

- More data exploration !
  - In total 240000 pairs of data, only 40000 unique query search
  - More key word matches, higher relevance score

  - 98539,130815,"Screen Tight 36 in. x 80 in. Brookgreen Solid Vinyl White Screen Door","36 screen door",1.67
  - 97918,130541,"Klein Tools Tradesman Pro 10 in. Tote Organizer","klein bag",1.67
  - obvious information-floor
  - Everyone get very close score on leaderboard
- More information should be involved
  - Not 2-gram, 3-gram or even 6-gram
  - Everyone in the team should work on features

# WINNER SOLUTION SCREEN SHOT

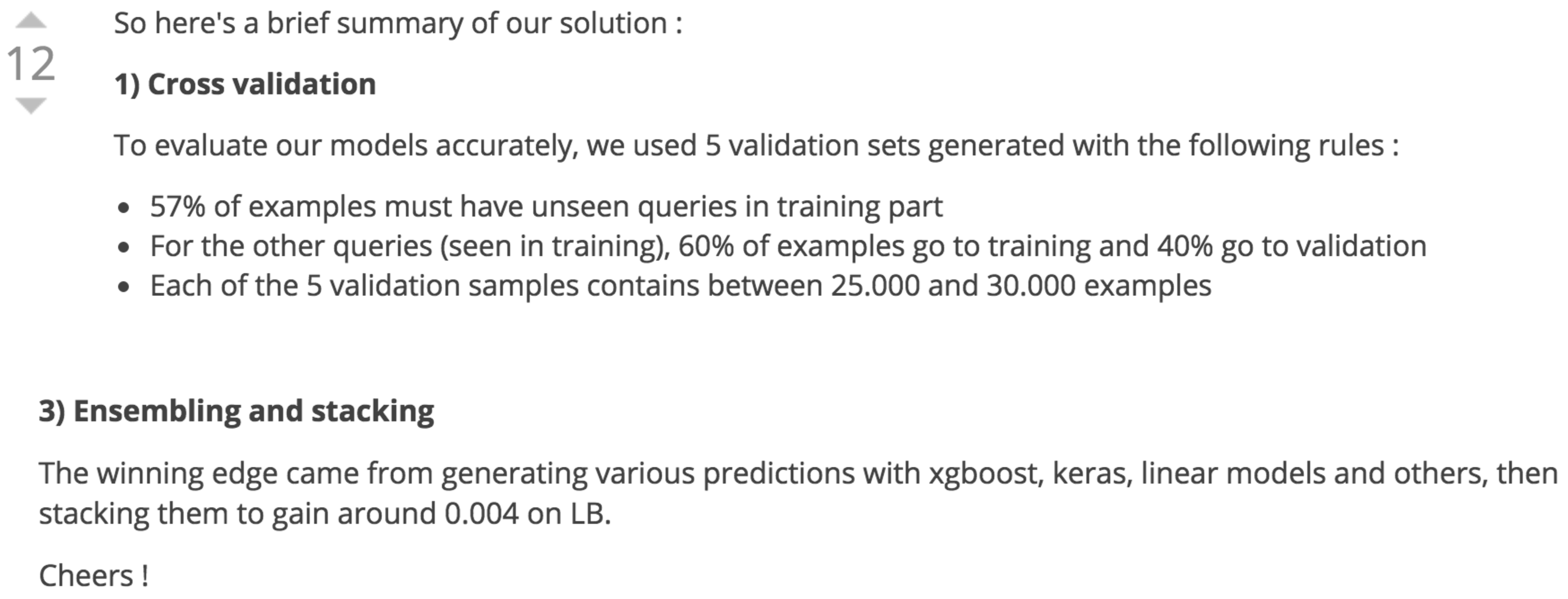


**2) Feature engineering**

Combining all 4 team members' datasets, we had more than 4000 features :

- Counts and metrics such as the ones in Chenglong Crowdflower solution
- Word2Vec/Gensim
- Glove
- Brands
- Measures
- Bullets
- Materials
- Signatures (Collection Name, Artist's name, Artwork name)
- Colors
- Query and title parsing and comparing
- Word features
- Word clustering
- Document clustering

With this feature engineering effort, single xgboosts score around 0.435 on public LB



So here's a brief summary of our solution :

**1) Cross validation**

To evaluate our models accurately, we used 5 validation sets generated with the following rules :

- 57% of examples must have unseen queries in training part
- For the other queries (seen in training), 60% of examples go to training and 40% go to validation
- Each of the 5 validation samples contains between 25.000 and 30.000 examples

**3) Ensembling and stacking**

The winning edge came from generating various predictions with xgboost, keras, linear models and others, then stacking them to gain around 0.004 on LB.

Cheers !

# THANK YOU!

- More you invest, more you will gain!