Sheffield R MeetUp

Compete (and win) on Kaggle

Lukáš Drápal

Senior Data Scientist, Capital One

(lukas.drapal@capitalone.com)

1st November 2016

Agenda

- Introduction
  - I do not Kaggle as my day job ☺
- Kaggle.com
  - How it works?
- Allstate Purchase prediction challenge
  - Challenge description
  - Solution overview
  - Used technique & tools
- Why Kaggle?

How scoring on Kaggle works

- Training set (97 009 customers): response known
- Test set (55 716 customers): response unknown
  - Public leaderboard (30% of test set)
    - Score on public leaderboard is shown immediately after a prediction is uploaded
  - Private leaderboard (70% of test set)
    - Shown after the competition end
    - Only private leaderboard score matters
- “Leaderboard overfitting”
  - Tuning predictions based on the public leaderboard
  - Decreases the ability of predictions to generalized on the private leaderboard

Completed • $50,000 • 1,568 teams

Allstate Purchase Prediction Challenge

Tue 18 Feb 2014 – Mon 19 May 2014 (21 months ago)

Dashboard

Private Leaderboard - Allstate Purchase Prediction Challenge

This competition has completed. This leaderboard reflects the final standings.

See someone using multiple accounts? Let us know.

#	Δrank	Team Name * in the money	Score	Entries	Last Submission UTC (Best – Last Submission)
1	↑10	**Prazaci** *	**0.53743**	**151**	**Mon, 19 May 2014 19:00:09 (-4.4h)**
2	↑2	Alessandro & BreakfastPirate *	0.53715	263	Mon, 19 May 2014 21:13:00 (-20.3h)

Dashboard

Public Leaderboard - Allstate Purchase Prediction Challenge

This leaderboard is calculated on approximately 30% of the test data.
The final results will be based on the other 70%, so the final standings may be different.

See someone using multiple accounts? Let us know.

#	Δ1w	Team Name * in the money	Score	Entries	Last Submission UTC (Best – Last Submission)
1	↑2	Magic Learner *	0.54571	397	Mon, 19 May 2014 23:49:54 (-2.3d)
2	↓1	Owen *	0.54571	71	Mon, 19 May 2014 00:55:50 (-0h)
3	↓1	Finite State Insurance Machines *	0.54565	222	Mon, 19 May 2014 20:02:51 (-3.8d)
4	—	Alessandro & BreakfastPirate *	0.54535	263	Mon, 19 May 2014 21:13:00 (-20.3h)
5	↑5	JWANG	0.54487	56	Mon, 19 May 2014 23:46:56
6	↑11	dynamic24	0.54481	231	Mon, 19 May 2014 23:54:13 (-25.6h)
7	↓1	User Error Structure	0.54463	105	Sun, 18 May 2014 17:37:47 (-39.8h)
8	↑4	Random Predict	0.54445	292	Mon, 19 May 2014 18:02:54 (-43.5h)
9	↓4	Maxim	0.54433	102	Mon, 19 May 2014 18:20:38 (-11.8d)
10	↓2	Peng	0.54415	112	Mon, 19 May 2014 23:42:08 (-23.1h)
11	↑52	**Prazaci**	**0.54403**	**151**	**Mon, 19 May 2014 19:00:09 (-4.4h)**

Business idea

- Business idea:
  - Recommend insurance policy settings for customers
  - Shorter quoting process
  - Better customer experience
  - A customer does not leave to competition within a tedious process

Problem description

- Task: Predict the purchased coverage options
  - "Quote" = a single combination of 7 options
  - Each option has 2 to 4 possible values
- Data for one customer consists of:
  - Demographic information + location + cost of quote
  - Quote history:

	A Collision	B Property Damage	C Medical	D Uninsured	E Under-insured	F Bodily Injury	G Compre-hensive
Quote 1	1	1	4	3	0	1	2
Quote 2	1	1	4	3	0	1	2
Quote 3	1	2	4	3	0	2	2
Quote 4	1	1	4	3	0	2	2
Quote 5	1	1	4	3	0	2	2
Purchase	1	1	4	3	0	2	2

- Last quoted benchmark (LQB) worked really well

Modelling

- Strict evaluation metrics – all policy options (A to G) needs to be predicted correctly (no partial credit)
- What to choose as the response variable?
  - One policy option
    - 7 models
    - Although each individual option is predicted with a high accuracy, last quoted benchmark worked better
  - All policies together
    - Response with level "2143022" corresponds to (A = 2, B = 1, …)
    - Too many levels (> 2000), too little data
  - Some policies together
    - Pick pairs that are correlated (AF, BE, CD, G) and make 4 models

Example: model for AF (1/2)

- Possible values A = {0, 1, 2}; F = {0, 1, 2, 3}
- Created variable AF with 12 levels:

  AF = {00, 01, 02, 03, 10, …, 22, 23}
- Predictors - One row per a customer:

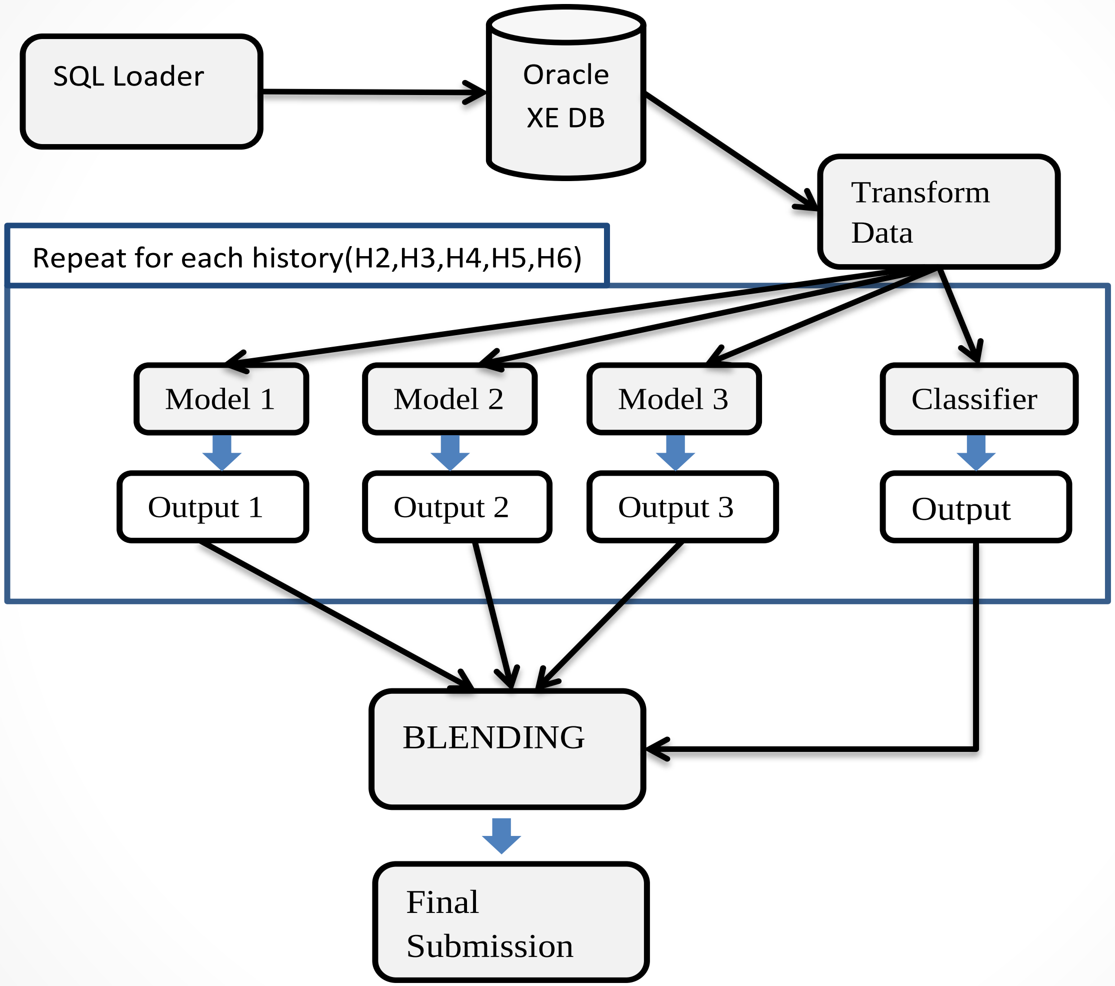
  Demographic information, location, Quote_1A, .. Quote_1G, …., Quote_5A, … Quote_5G ()
- Multinomial response: AF (e.g. 10)

	A Collision	B Property Damage	C Medical	D Uninsured	E Under-insured	F Bodily Injury	G Compre-hensive
Quote 1	1	1	4	3	0	1	2
Quote 2	1	1	4	3	0	1	2
Quote 3	1	2	4	3	0	2	2
Quote 4	1	1	4	3	0	2	2
Quote 5	1	1	4	3	0	2	2
Purchase	1	1	4	3	0	2	2

Example: model for AF (2/2)

- Output: 12 scores describing how is likely a given combination of AF
- Prediction of the model: combination of AF with the highest score

Solution overview

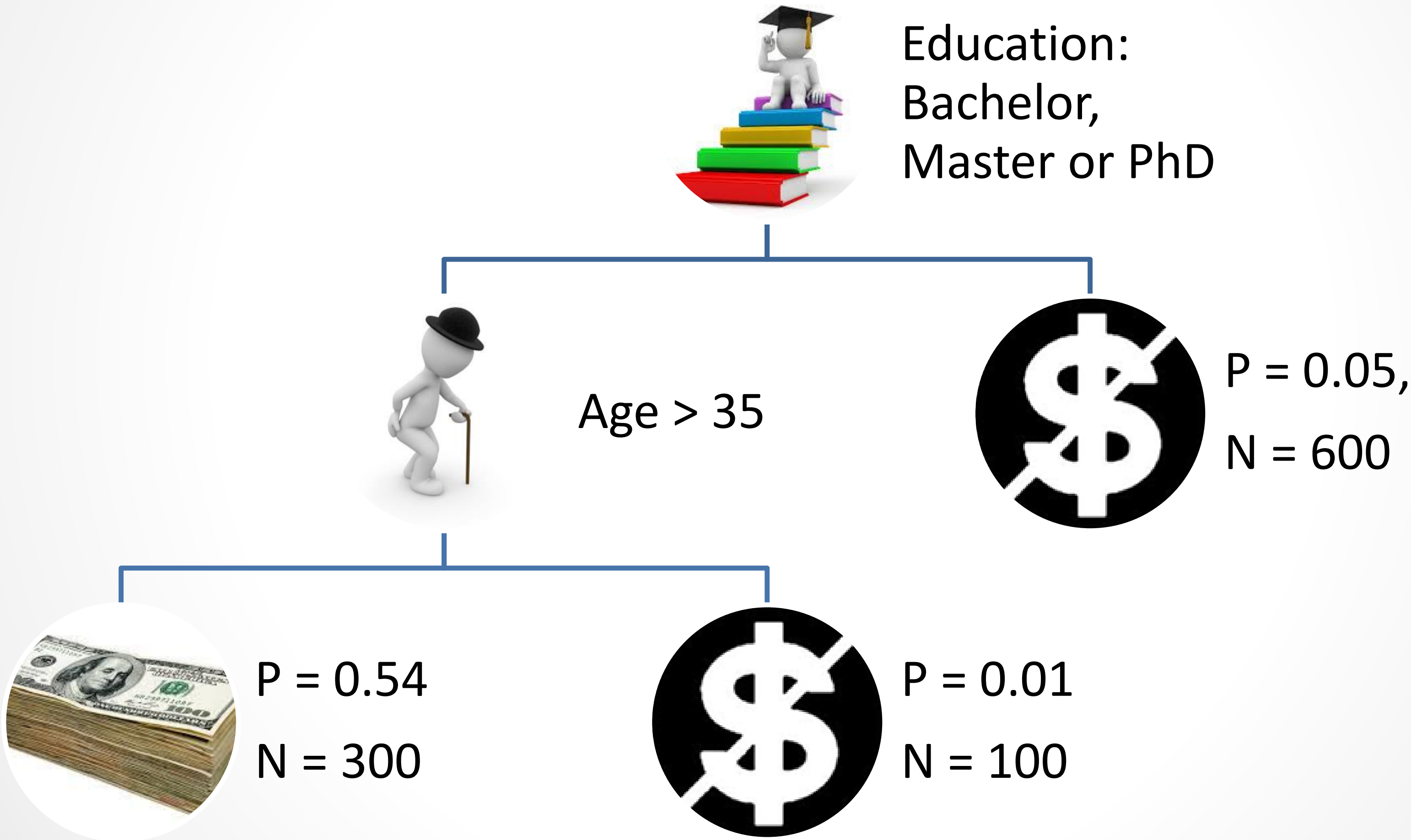


Models

- Averaging models together helps to improve performance (variance of predictions decreases)
  - Especially if models are not correlated, errors tend to cancel out
- 3 models with different complexity were build and combined
- Classifier
  - Models still lack interaction between policies
  - Classify whether LQB or models outcome should be used
  - It can say that LQB should be used even when three models agree on a change
- All models: gradient boosted machines

Decision trees

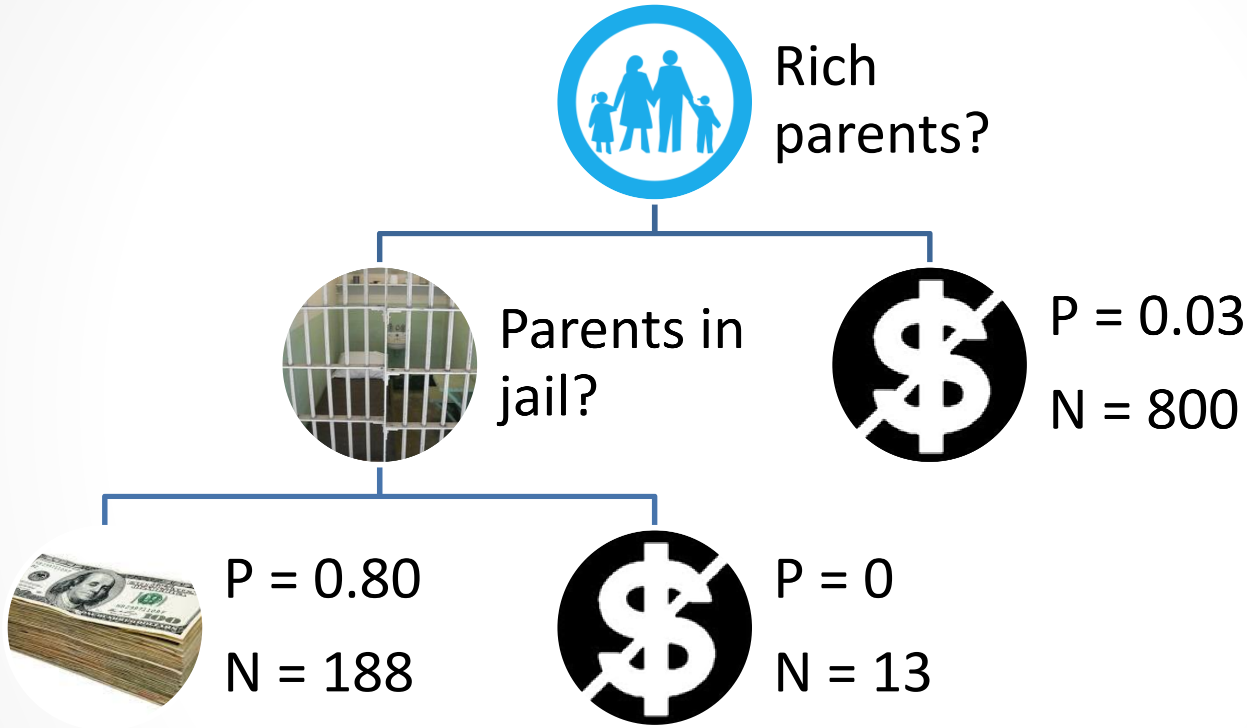
Find out the rich folks on a party with yes/no questions:



P = Probability of being rich
N = Sample size

Decision trees can be very instable

After Paris Hilton walks in:



P = Probability of being rich
N = Sample size

Instability: a small change in a dataset can lead to a completely different structure

General stochastic gradient

- General – any differentiable loss function L
- Observations from training set (X_1, y_1), … (X_n, y_n)

1. Put constant $$F_0(x) = \arg\min_\gamma \sum_{i=1}^{n} L(y_i, \gamma).$$

2. For each tree (m = 1,…,M) do

   a) Bagging (sample rows of dataset)
   b) Compute *pseudresiduals:* $$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$$
   c) Fit a regression tree h_m with K terminal nodes with pseudoresiduals r_{im} as the response using only *the bagged sample*
   d) Find optimal $$\gamma_{mk} = \arg\min_\gamma \sum_{S_k} L(y_i, F_{m-1}(x_i) + \gamma)$$
   where S_k is the set of x_i that define terminal node k.
   e) Update $F_m(x_i) = F_{m-1}(x_i) + \alpha\gamma_{mk(x_i)}$ where *$k(x_i)$* indicates the index of the terminal node into which observation x_i would fall. This step is done only on the bagged sample.

3. Use $F_M(x)$ as the final model

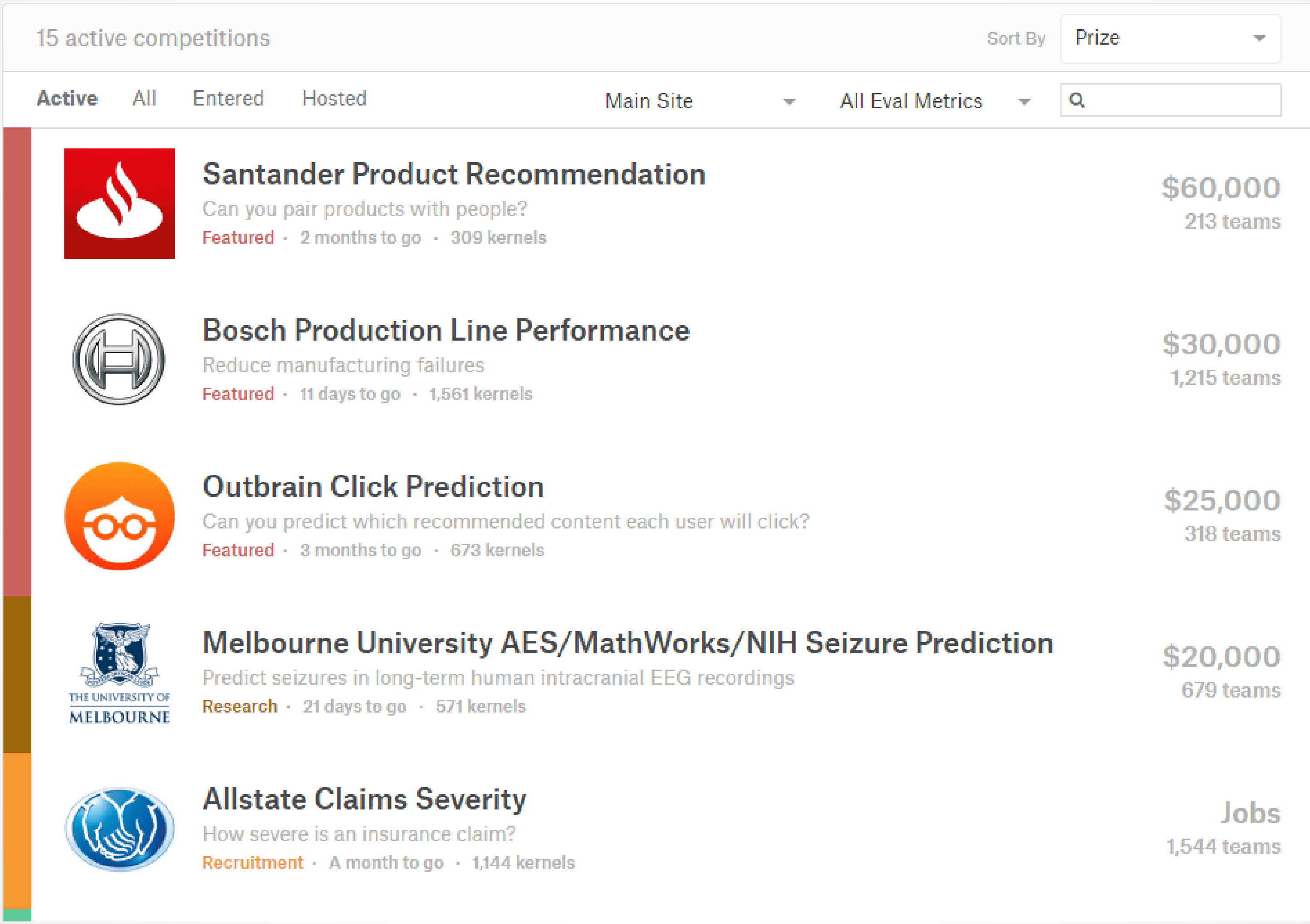
Gradient boosting machines in practise

- Used package gbm, hyperparameters tuned with caret
- Hyperparameters that need to be tuned:
  - Number of trees (M)
  - Depth of trees
  - Minimal number of observations in a node
  - Learning rate (effects finding optimal c)
  - Bagging proportion
  - Loss function
- XGBoost – (new) very flexible implementation
- H2O – modern modelling opensource ML tool with connections (packages/libraries) to Python, R…

Why Kaggle?

- Criticism: Kaggle is not like the real world
  - Problem definition, evaluation and data are not that clear
- Data science is learned by DOING
- Kaggle offers:
  - Great datasets to play with
  - Competitions that can really push You
  - Community that shares the newest tools (H2O, Vowpal Wabbit, …) and top techniques
- Observations:
  - Performance boost is mostly based on feature engineering
  - Averaging predictions based on different algorithms (e.g. gradient boosting + deep learning) helps to get an edge

Warning: competing @ Kaggle is addictive



Q&A