

London R Meetup

Compete (and win) on Kaggle.com

Lukáš Drápal

Senior Data Scientist, Capital One



Kaggle Master

(lukas.drapal@capitalone.com)

31st January 2017

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 generalize 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 <small>* in the money</small>	Score <small>🔒</small>	Entries	Last Submission UTC (Best - Last Submission)
1	↑10	Prazaci <small>👤 *</small>	0.53743	151	Mon, 19 May 2014 19:00:09 (-4.4h)
2	↑2	Alessandro & BreakfastPirate <small>👤 *</small>	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 <small>* in the money</small>	Score <small>🔒</small>	Entries	Last Submission UTC (Best - Last Submission)
1	↑2	Magic Learner <small>👤 *</small>	0.54571	397	Mon, 19 May 2014 23:49:54 (-2.3d)
2	↓1	Owen <small>*</small>	0.54571	71	Mon, 19 May 2014 00:55:50 (-0h)
3	↓1	Finite State Insurance Machines <small>👤 *</small>	0.54565	222	Mon, 19 May 2014 20:02:51 (-3.8d)
4	—	Alessandro & BreakfastPirate <small>👤 *</small>	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 <small>👤</small>	0.54463	105	Sun, 18 May 2014 17:37:47 (-39.8h)
8	↑4	Random Predict <small>👤</small>	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 <small>👤</small>	0.54403	151	Mon, 19 May 2014 19:00:09 (-4.4h)

Business idea

- Business idea:
 - Recommend insurance policy settings to customers visiting website
 - 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)

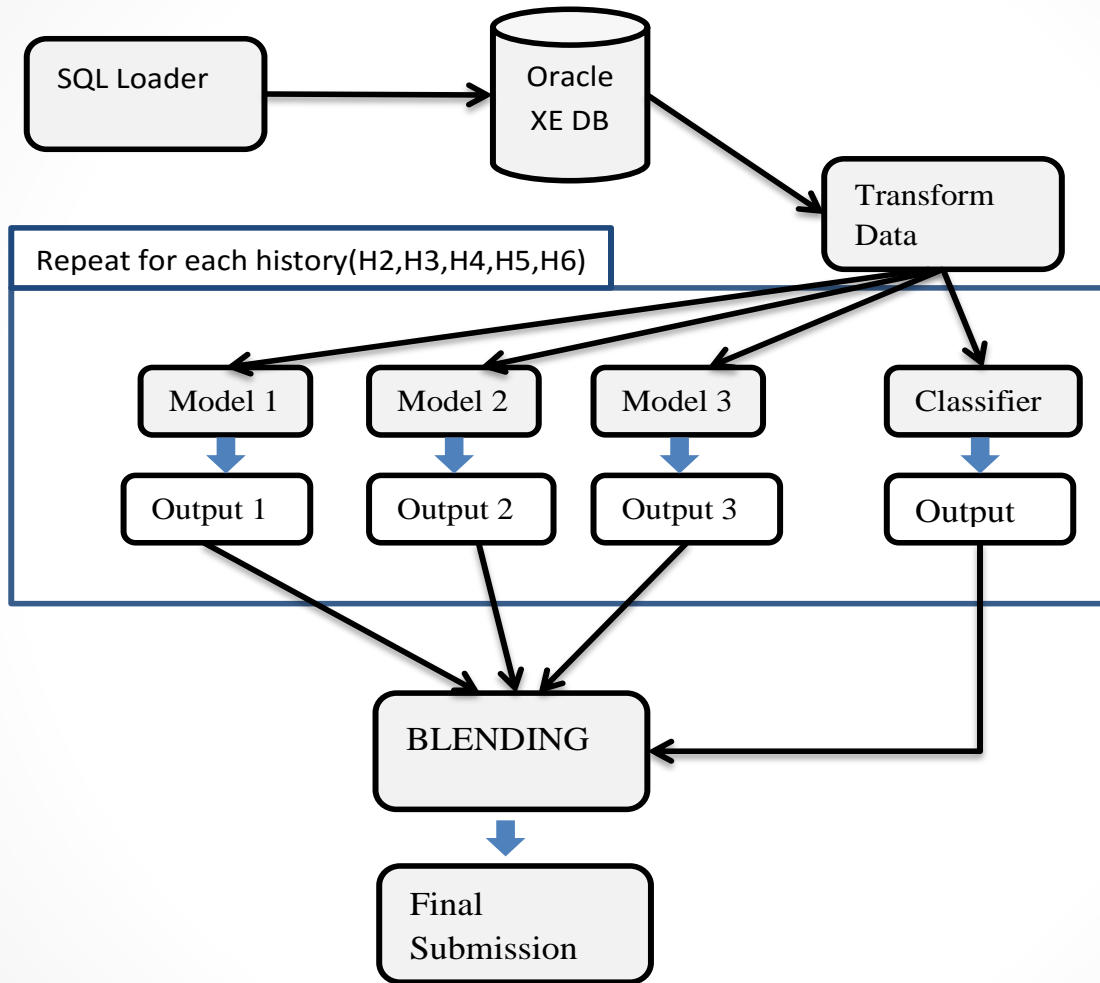
- Possible values $A = \{0, 1, 2\}$; $F = \{0, 1, 2, 3\}$
- Created variable AF with 12 levels:
AF = {00, 01, 02, 03, 10, ..., 22, 23}
- Predictors - One row per a customer:
Demographic information, location, Quote_1A, .. Quote_1G,
....., Quote_5A, ... Quote_5G
- Multinomial response: AF (e.g. 10)

	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

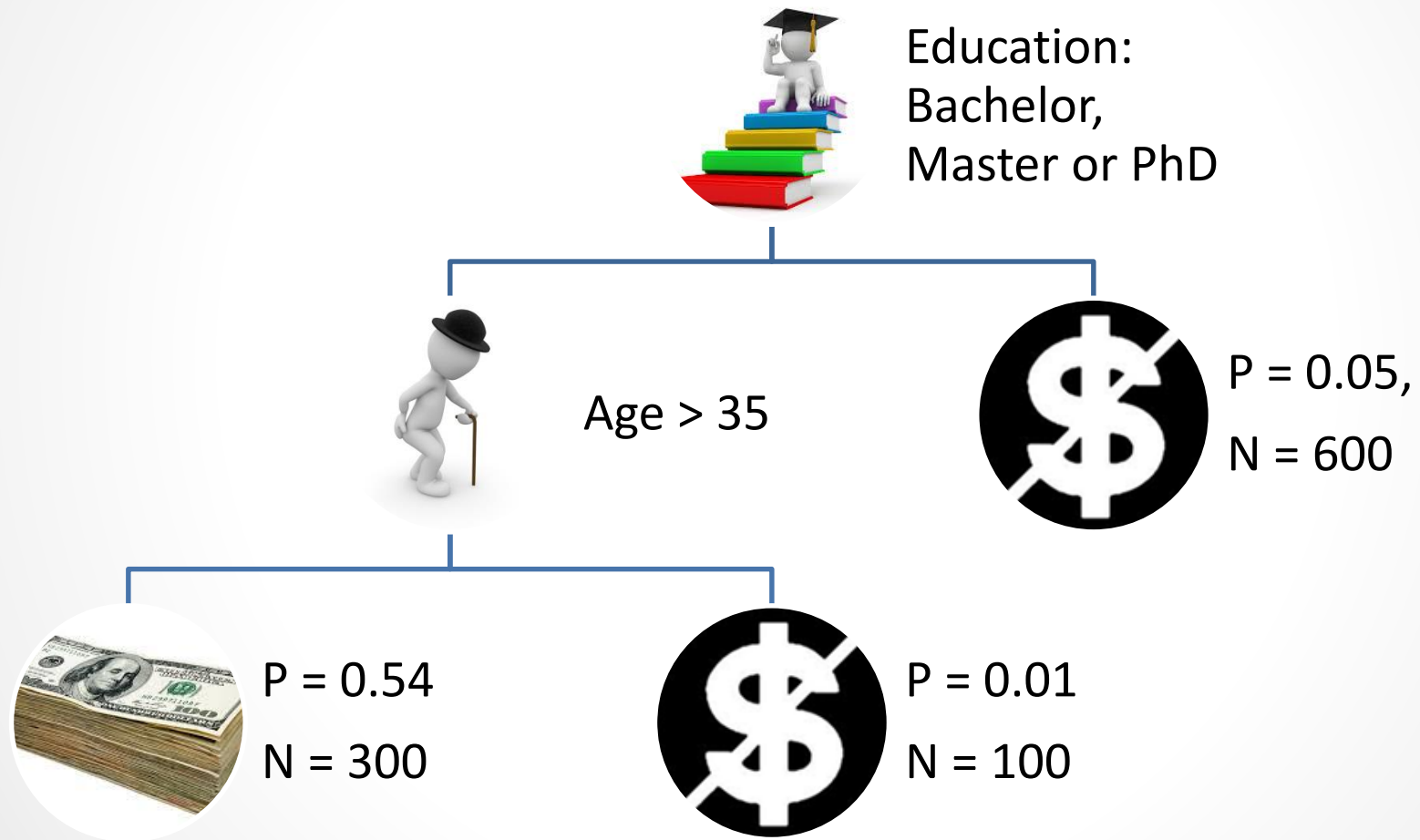


Models

- 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

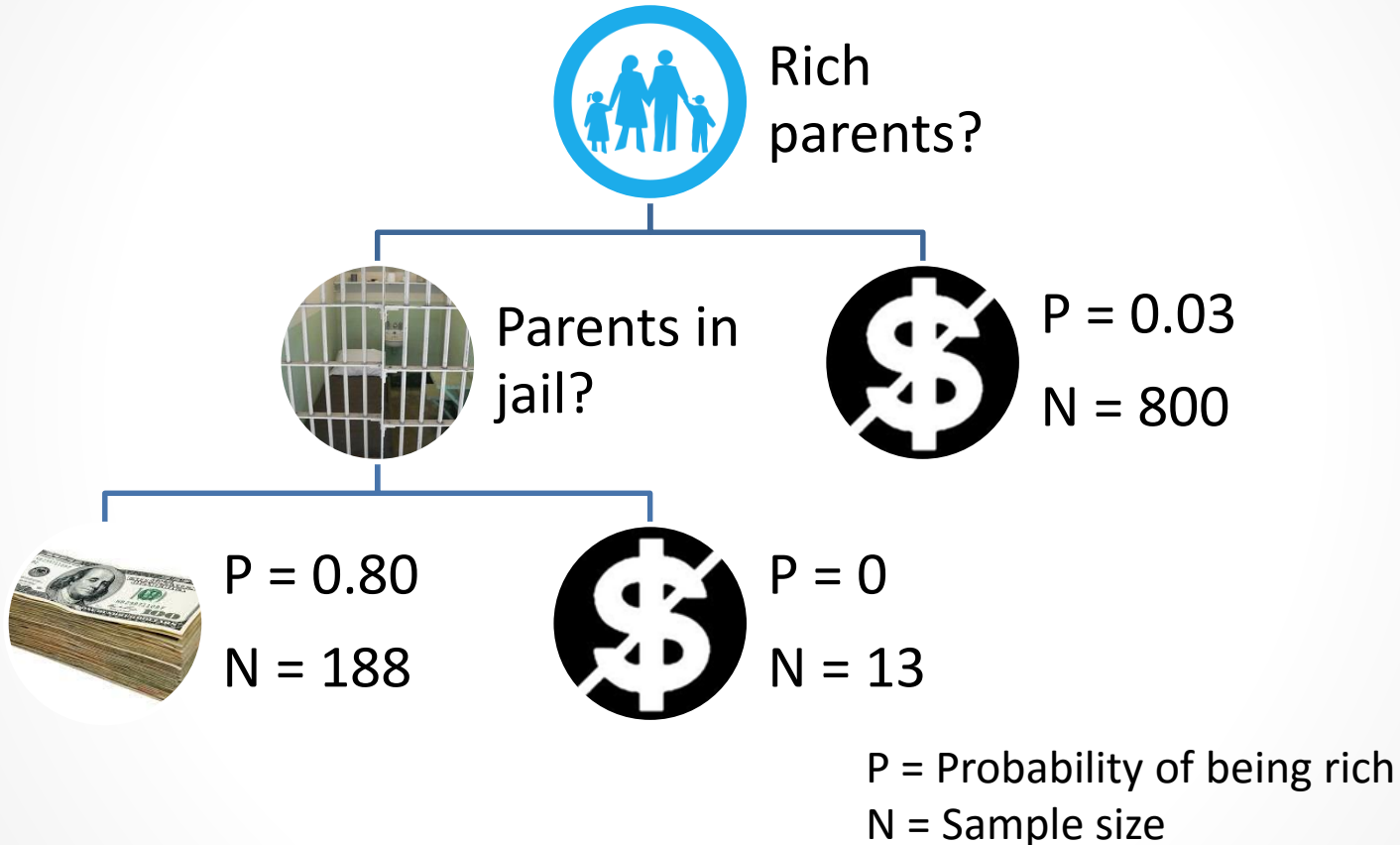
Find out the rich folks on a party with yes/no questions:



P = Probability of being rich
N = Sample size

Decision trees can be very instable

After Paris Hilton walks in:



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

Stochastic gradient boosting: motivation

- Observations from training set $(x_1, y_1), \dots (x_n, y_n)$
- We have built the first tree $F_1(x)$
- Can we build another tree $h_2(x)$ to improve our predictions?
 - Ideally, we want $y = F_1(x) + h_2(x)$
 - Hence, let's build tree $h_2(x)$ to predict $(y - F_1(x))$
 - Combine the new tree with the previous tree to make a model $F_2(x) = F_1(x) + h_2(x)$
- Keeping building trees h_m to a predict $y - F_{m-1}(x)$ as long as it helps predictive power on test/validation set

Stochastic gradient boosting

- L ... Loss function (needs to be differentiable)

1. Put constant

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

2. For each tree ($m = 1, \dots, M$) do

a) *Bagging* (sample rows of dataset)

b) Compute *pseudo residuals*: $r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)}$

(essentially $r_{im} = y_i - F_{m-1}(X_i)$, in the case of MSE loss function)

c) Fit a decision tree h_m on the bagged sample with pseudoresiduals r_{im} as the response

d) Find optimal $\gamma_m = \arg \min_{\gamma} \sum_{i=1}^n L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$. (only using the bagged sample)

e) Update $F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \alpha$, learning rate α e.g. 0.05 reduces overfitting

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

- Stochastic – through bagging
- Gradient – at each point we minimize gradient of a loss function
- Boosting – iterative combination of trees that are weak on their own to make a powerful 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
 - Bagging proportion
 - Loss function
- XGBoost: flexible implementation using regularization
- H2O: scalable ML tool for big data – easy to use R/python library; distributed computing & deployment in java
- LightGBM: new gbm implementation from Microsoft

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
 - Community that shares the newest tools (H2O, Vowpal Wabbit, ...) and top techniques
 - Competitions that really push You to build the best models
- 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

10 active competitions		Sort By	Prize		
Active	All	Entered	Hosted	All Categories	Q
	Data Science Bowl 2017	Can you improve lung cancer detection?	\$1,000,000 898 teams		
	Featured	· 2 months to go · 308 kernels			
	The Nature Conservancy Fisheries Monitoring	Can you detect and classify species of fish?	\$150,000 1,340 teams		
	Featured	· 2 months to go · 232 kernels			
	Dstl Satellite Imagery Feature Detection	Can you train an eye in the sky?	\$100,000 175 teams		
	Featured	· A month to go · 108 kernels			
	Two Sigma Financial Modeling Challenge	Can you uncover predictive value in an uncertain world?	\$100,000 1,609 teams		
	Featured	· A month to go · 165 kernels			
	Dogs vs. Cats Redux: Kernels Edition	Distinguish images of dogs from cats	854 teams		
	Playground	· A month to go · 155 kernels			
	Transfer Learning on Stack Exchange Tags	Predict tags from models trained on unrelated topics	243 teams		
	Playground	· 2 months to go · 82 kernels			

Q&A

