Statistics Finland

How to Make AI do Your Job for Statistical Classification of Industry and Occupation

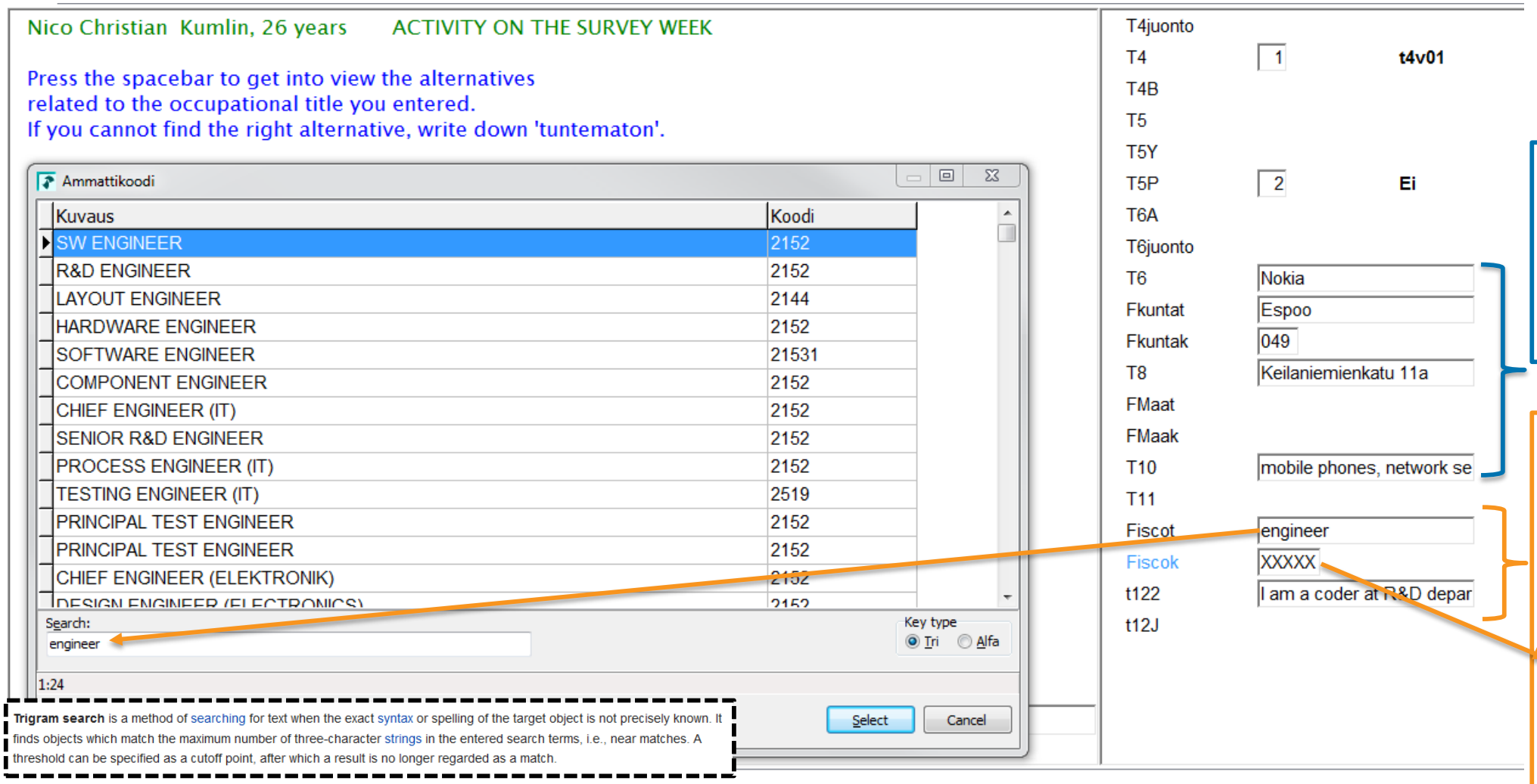
Jukka Kärkimaa, Data scientist
Liisa Larja, Senior Statistician

BigSurv18, 26th October 2018

Problem / Motivation

- Coding of person's occupation and workplace's industry according to the statistical classifications (NACE, ISCO) can be very labour intensive
  - NACE: 800 levels with 5 digits
  - ISCO: 500 levels with 5 digits
- Many times the classifications are so complex, that neither the survey respondent nor the interviewer can choose the right code by herself
- Very often the interviewees are asked various auxiliary questions and the coding to the classification is done manually after the survey interview at the statistical office
- **How to make artificial intelligence do this work for you using machine learning methods?**

Currently, at FI-LFS manual coding takes 15 % of human resources



name, address of the job and open text description of the industry → **manual matching** to the enterprise register NACE-codes after the interview.

name of the occupation = index word for trigram search → interviewer chooses the correct code (coding ready!)

If appropriate code not found → provide job description (open text – to be **manually classified** after the interview).

Statistics Finland

Data

- 1M survey respondents from 2011-2018
  - Industry NACE or occupation ISCO available for 600k respondents
- At FI-LFS we regularly use register data
  - The same variables can also be derived by interviewing!

- X = source used in our analysis (x) = alternative source, that could be used

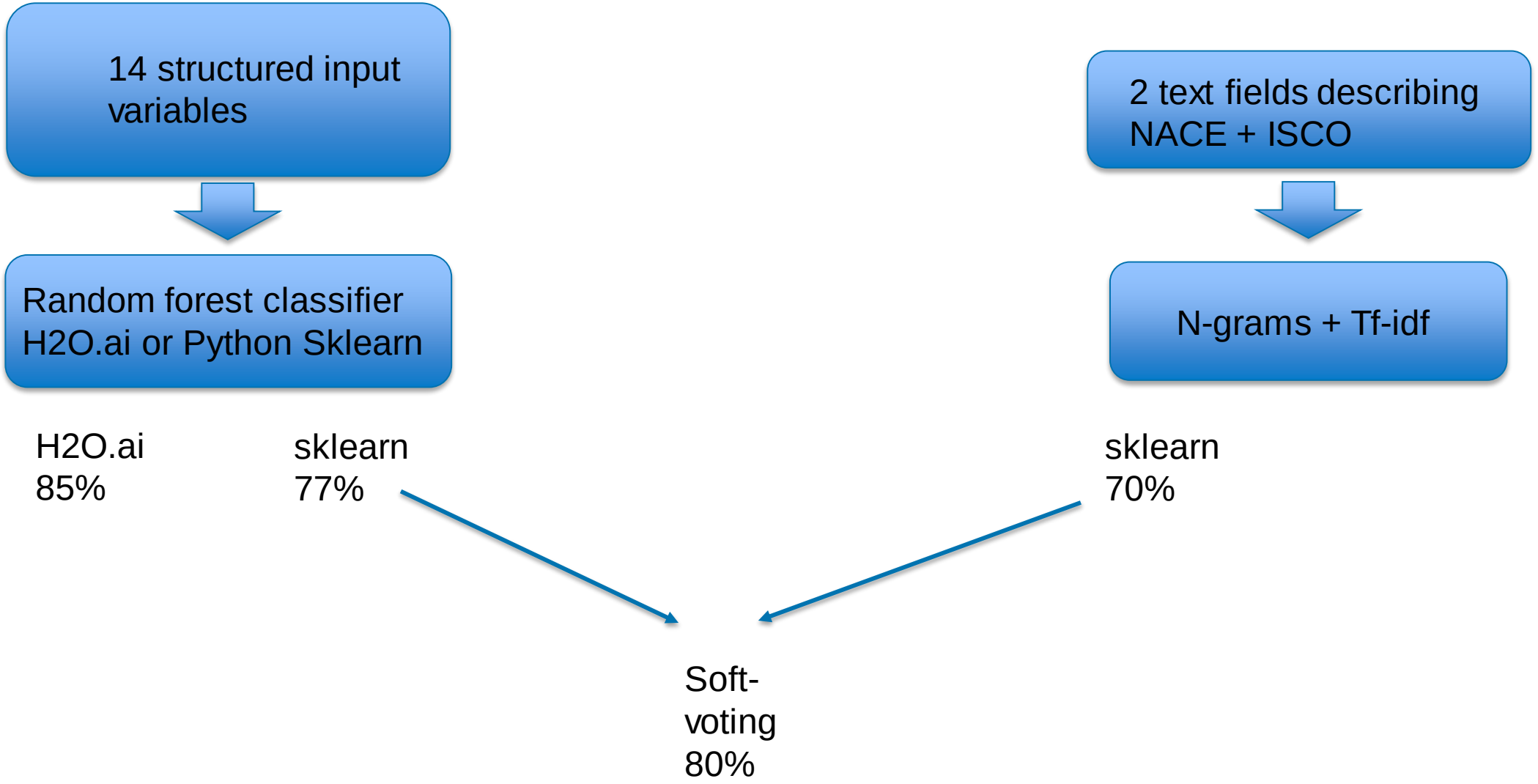
Variable	Source (register; survey)	Type (categorical; continuous; free text)	Cardinality
Age	R (s)	cat.	67, ordinal
Sex	R (s)	cat.	2
Country of birth	R (s)	cat.	149
Registered as unemployed	R (s)	cat.	10
Education (ISCED at 5-digit level)	R (s)	cat.	1728
Education (ISCED at 1-digit level)	R (s)	cat.	7
Postal code of the employer	S (r)	cat.	341
Occupational status (employee, self-employed, etc)	S (r)	cat.	10
Year of interview	S	cat.	8
Years of residency in Finland for immigrants	S	cat.	12, ordinal
Salary / income	R (s)	cont.	
”What are your main job tasks?” *[open text]*	S	free	
”What is the branch of industry of your place of work? Describe the activity which the establishment engages in” *[open text]*	S	free	
Industry (NACE 5 digits, manually coded)	**S**	**cat.**	774
Occupation (ISCO 5 digits, manually coded)	**S**	**cat.**	*478*

Variable importances for predicting NACE with H2O.ai DRF-model

variable	percentage / %
Occupation, ISCO, 5 digits	25
Postal code of the employer	18
Education, ISCED 5 digits	14
Occupation, ISCO, 3 digits	9
Age	9
Salary / income	6
Year of interview	5
Occupation, ISCO, 2 digits	4
Level of education, 1 digit	2
Occupational status (employee, self-employed, etc)	2
Years of residency in Finland for immigrants	2
Sex	1
Registered as unemployed	1
Country of birth	1

Models

- All models and results below are for prediction of NACE
  - In predicting ISCO, analogous models seem to have similar performance
- Random forest for structured data with categorical + continuous variables
  - H2O.ai Distributed Random Forest (DRF)
  - Python Scikit-learn RandomForestClassifier
- n-grams + Tf-idf preprocessing + Random forest classifier for free-text fields



Model stacking is a handy way to combine structured and unstructured data as input variables

Test set hit ratios for H2O.ai Random Forest

k	hit_ratio / %
1	85.4
2	90.9
3	92.9
4	94.0
5	94.7
6	95.1
7	95.5
8	95.7
9	96.0
10	96.2

Random forest on structured data

- How to deal with categorical variables such as education code of cardinality n=1728?
- Computing a theoricatically optimal decision tree using all combinations of values of a categorical variable takes time O(2^n) -> 2^1700 ~ 10^500
- Sklearn random forest assumes ordinality for numeric categoricals and simply refuses to process textual categoricals as such
- One-hot encoding is known to be bad for high cardinality variables and decision trees
- H2O.ai uses a ” Histogram and binning” [1] method, which seems to be state-of-the-art for dealing with categorical variables of high cardinality
- On a server with 32GB RAM and 4 Xeon CPU cores
  - Sklearn: 1 min -> 77%, H2O.ai: 5 h -> 85%
- H2O’s method seems to win in predictive accuracy and this is also what literature suggests

[1] http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/algo-params/nbins_cats.html

Results

- Due to lack of time/computing power, the models are not fully optimized for the full dataset of 600k intervieews
- Reached 85.4 % precision so far for H2O.ai DRF for the full data set of 600k rows using 10 % randomly sampled rows as the test set with years 2011-2018
- Stacking with free-text model and applying all other improvements we can very likely exceed 90% for the full, raw data set

Accuracy estimates

- Overall accuracy should be very close to test set accuracy
  - Probability sample
- Out-of-bag estimates are important
  - Unbiased probability estimates for model predictions
- K-fold cross-validation can help especially with smaller datasets
  - Random forest does essentially this implicitly if we use a ”per-tree row sampling rate” less than 100% and out-of-bag estimates
- However, data-generating process likely to change over time, deteriorating even unbiased estimates from historical data
  - Manual verification for a random sample needed
  - Re-fitting the model regularly with up-to date training data

LFS process with automatic coding

- Very confident cases can be automatically classified by the model
- We can use the old process but sort the cases to be manually classified using our model from worst to best prediction
  - We will refit the model after every few hundred cases manually verified / fixed
- After accepting automatic classification for many cases, we can randomly sample a few cases for manual verification and make sure that the error in automatic classification is not more than what we expect from the model
- This will easily halve the time spent on manual classification without affecting quality
- Alternatively, we can still manually classify all cases but use our model to double check
  - Time spent will stay the same but quality will improve
- Showing e.g. 10 best predictions from the model also speeds up manual classification as these capture >96% of the correct classes (hit-ratio)

Balancing classes?!

- Oversampling the rare classes may increase predictive accuracy for those classes but may also spoil the unbiased probability estimates for model predictions
- We could fit 2 models and combine in an optimal way:
  - One on class-balanced data
  - Another on original data

Suggestions for improvement

- Predicting only for the current year 2018, the accuracy seems to drop 85%->80%
  - Random manual checks on new data and regular model updates are needed in production!
  - Over-sampling the current year and the more recent years may help?
- Gradient boosting is known to be among the best models for structured data in many cases (such as Kaggle competitions)
  - May yield better results than random forest but needs careful parameter tuning
- Use ”Stacking” or ”Super learning” (https://doi.org/10.2202/1544-6115.1309) to combine several different models of the structured variables in a weighted average/vote
- Deep learning based sequence models are among the state-of-the-art in Natural language processing
  - Free text is quite simple in most cases for LFS but this is worth trying out
- Finnish language is very complicated so also more careful language preprocessing may help

Conclusion

- We can confidently deploy machine learning models to help speed up laborious classification tasks in official statistics as long as we have unbiased estimates of their accuracy
- We achieved 85.4% predictive accuracy with H2O.ai random forest –model based on categorical interview/register variables
- Processing categorical variables is in high demand at statistical offices and in social science -> the ”histogram and binning” method we use shows improvement over traditional methods
- Combining with a free-text model further improves accuracy
- AI helps remove most of the manual work in classifications!

Contact information

Jukka Kärkimaa

Data scientist

Statistics Finland, Statistical Methods

+358 29 551 3011

jukka.karkimaa@stat.fi

Liisa Larja

Senior Statistician

Statistics Finland, Population and Social Statistics

+358 29 551 3461

liisa.larja@stat.fi