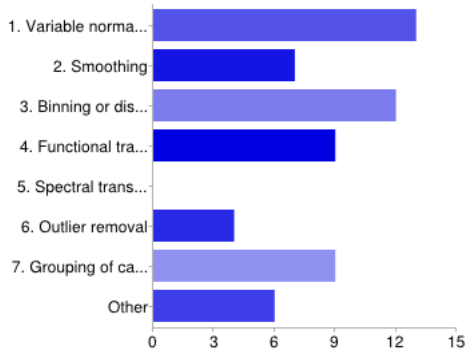
Preprocessing and data representation

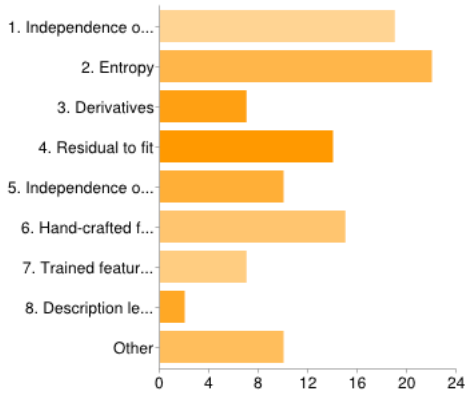
Preprocessing of A and B variables



1. Variable normalization	**13**	48%
2. Smoothing	**7**	26%
3. Binning or discretization	**12**	44%
4. Functional transform (e.g. log)	**9**	33%
5. Spectral transform	**0**	0%
6. Outlier removal	**4**	15%
7. Grouping of categorical variables	**9**	33%
Other	**6**	22%

People may select more than one checkbox, so percentages may add up to more than 100%.

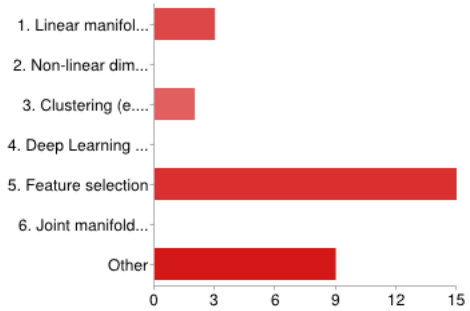
Feature extraction



1. Independence of variables	**19**	70%
2. Entropy	**22**	81%
3. Derivatives	**7**	26%
4. Residual to fit	**14**	52%
5. Independence of input and residual	**10**	37%
6. Hand-crafted features	**15**	56%
7. Trained feature extractors	**7**	26%
8. Description length or complexity of model	**2**	7%
Other	**10**	37%

People may select more than one checkbox, so percentages may add up to more than 100%.

Dimensionality reduction

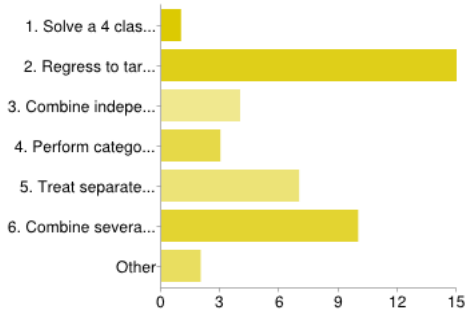


1. Linear manifold transformations (e.g. factor analysis, PCA, ICA)	**3**	11%
2. Non-linear dimensionality reduction (e.g. KPCA, MDS, LLE, Laplacian Eigenmaps, Kohonen maps)	**0**	0%
3. Clustering (e.g. K-means, hierarchical clustering)	**2**	7%
4. Deep Learning (e.g. stacks of auto-encoders, stacks of RBMs)	**0**	0%
5. Feature selection	**15**	56%
6. Joint manifold data fusion	**0**	0%
Other	**9**	33%

People may select more than one checkbox, so percentages may add up to more than 100%.

Recognition

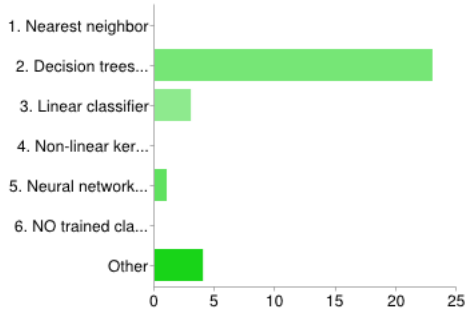
Architecture



1. Solve a 4 class problem then combine	**1**	4%
2. Regress to targets -1/0/1	**15**	56%
3. Combine independence and causal direction scores	**4**	15%
4. Perform categorical regression	**3**	11%
5. Treat separately categorical variables	**7**	26%
6. Combine several strategies	**10**	37%
Other	**2**	7%

People may select more than one checkbox, so percentages may add up to more than 100%.

Classifier

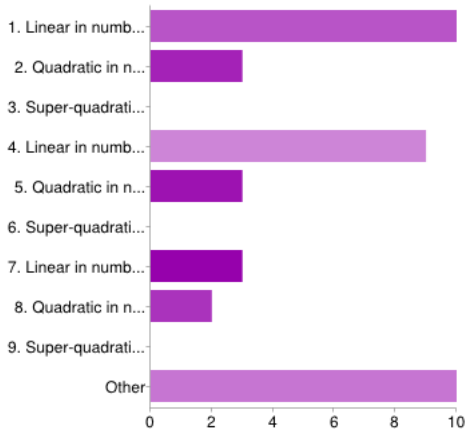


1. Nearest neighbor	**0**	0%
2. Decision trees or random forests	**23**	85%
3. Linear classifier	**3**	11%
4. Non-linear kernel method	**0**	0%
5. Neural networks or deep learning	**1**	4%
6. NO trained classifier	**0**	0%
Other	**4**	15%

People may select more than one checkbox, so percentages may add up to more than 100%.

Method advantages

Algorithmic complexity

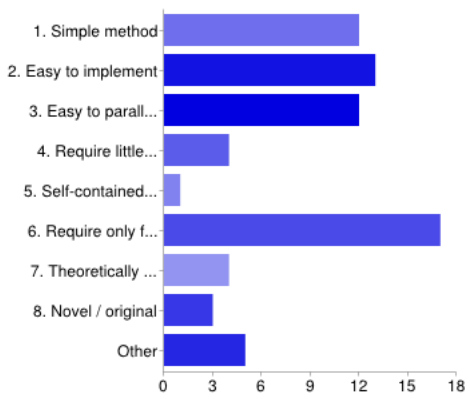


1. Linear in number of samples per variable	**10**	37%
2. Quadratic in number of samples per variable	**3**	11%
3. Super-quadratic in number of samples per variable	**0**	0%
4. Linear in number of training examples	**9**	33%
5. Quadratic in number of training examples	**3**	11%
6. Super-quadratic in number of training examples	**0**	0%
7. Linear in number of test examples	**3**	11%
8. Quadratic in number of test examples	**2**	7%
9. Super-quadratic in number of test examples	**0**	0%
Other	**10**	37%

People may select more than one checkbox, so percentages may add up to more than 100%.

Qualitative advantages

1. Simple method	**12**	44%
2. Easy to implement	**13**	48%
3. Easy to parallelize	**12**	44%
4. Require little memory	**4**	15%
5. Self-contained (does not rely on third party libraries)	**1**	4%
6. Require only freeware libraries	**17**	63%
7. Theoretically motivated	**4**	15%



8. Novel / original	**3**	11%
Other	**5**	19%

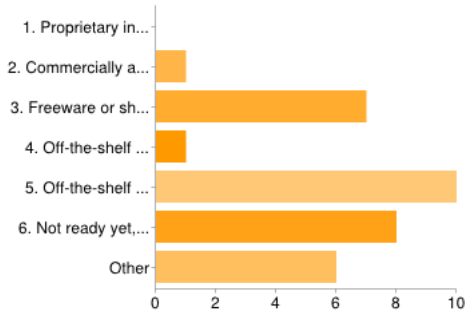
People may select more than one checkbox, so percentages may add up to more than 100%.

Comparison with other methods

We seemed to get better results when we gave a separate treatment to the categorical data and when we artificially doubled the training set sizes by multiplying the A and B columns by either 5 or -5. I did add non-invertible functions tests because of the literature. Successful because of diverse set of descriptors Simplicity From what I could tell from the forums, no one used a solution like mine. I assume it was novel (since I didn't base it on anything). Correct implementation the score function objective, enforce symmetries, reduce model noise via merging of multiple models I guess the ...

Software implementation

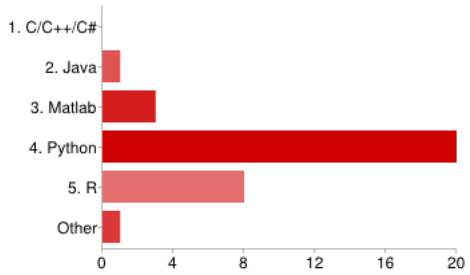
Availability



1. Proprietary in house software	**0**	0%
2. Commercially available in house software	**1**	4%
3. Freeware or shareware in house software	**7**	26%
4. Off-the-shelf third party commercial software	**1**	4%
5. Off-the-shelf third party freeware or shareware	**10**	37%
6. Not ready yet, but may share later	**8**	30%
Other	**6**	22%

People may select more than one checkbox, so percentages may add up to more than 100%.

Language



1. C/C++/C#	**0**	0%
2. Java	**1**	4%
3. Matlab	**3**	11%
4. Python	**20**	74%
5. R	**8**	30%
Other	**1**	4%

People may select more than one checkbox, so percentages may add up to more than 100%.

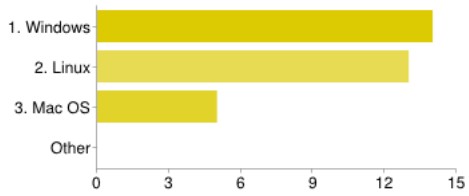
Details on software implementation

Separate models for: - Numerical -> Numerical - [Categorical|Binary] -> [Categorical|Binary] - Numerical -> [Categorical|Binary] - [Categorical|Binary] -> Numerical Weight factor between full estimator and specific estimator is 50%-50%, but the model does a better job in predicting Numerical->Numerical, so we give an extra boost of about 10%. Experimentation on Binary<->Binary: 10 million training samples were artificially created and I used genetic programming to determine the best possible features. Then I tried to use these features on groupings of the other types. The results weren't

g ...

Hardware implementation

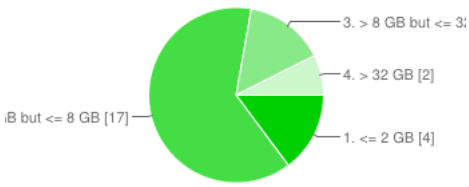
Platform



1. Windows	**14**	52%
2. Linux	**13**	48%
3. Mac OS	**5**	19%
Other	**0**	0%

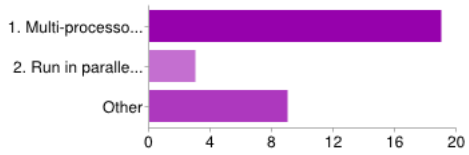
People may select more than one checkbox, so percentages may add up to more than 100%.

Memory



1. <= 2 GB	**4**	15%
2. > 2GB but <= 8 GB	**17**	63%
3. > 8 GB but <= 32 GB	**4**	15%
4. > 32 GB	**2**	7%

Parallelism



1. Multi-processor machine	**19**	70%
2. Run in parallel different algorithms on different machines	**3**	11%
Other	**9**	33%

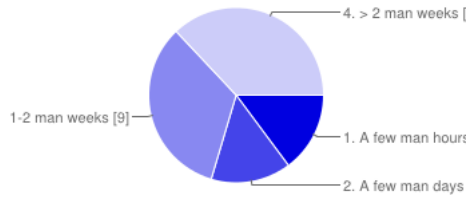
People may select more than one checkbox, so percentages may add up to more than 100%.

Code URL

https://github.com/sjuvekar/Cause-Effect-Kaggle	No Yet	https://github.com/diogo149/CauseEffectPairsChallenge

Development effort

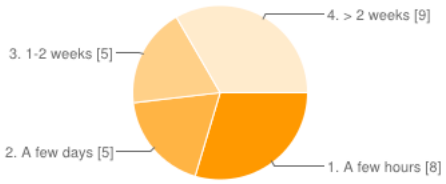
Total human effort



1. A few man hours	**4**	15%
2. A few man days	**4**	15%
3. 1-2 man weeks	**9**	33%
4. > 2 man weeks	**10**	37%

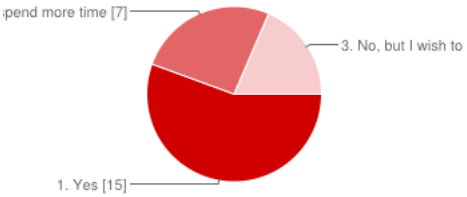
Total machine effort

1. A few hours	**8**	30%



2. A few days	**5**	19%
3. 1-2 weeks	**5**	19%
4. > 2 weeks	**9**	33%

Challenge duration OK?



1. Yes	**15**	56%
2. No, but I cannot spend more time	**7**	26%
3. No, but I wish to enter round 2 of the challenge	**5**	19%

Final evaluation time (hours)

1 1 12 6 2 1 48 6 3 hours 5 4 1 Not sure. 12 2 2 4 0.5 1 0.1 10 minutes 1 4 hours 10/20/2013 24 6 hours maybe 5 minutes on 100 node cluster

Number of daily responses