Hands-on-data #4

The goal of this assignment is for you to familiarize yourself with different validation approaches designed to minimize over-fitting and with automated feature selection approaches. It is suggested that you complete the assignment using RapidMiner, but you are free to use a different equivalent tool if you desire.

When submitting the assignment, include a text document answering the questions as well as RapidMiner process files (or files associated to the tool you used to create the models) for each question.

Validation of models

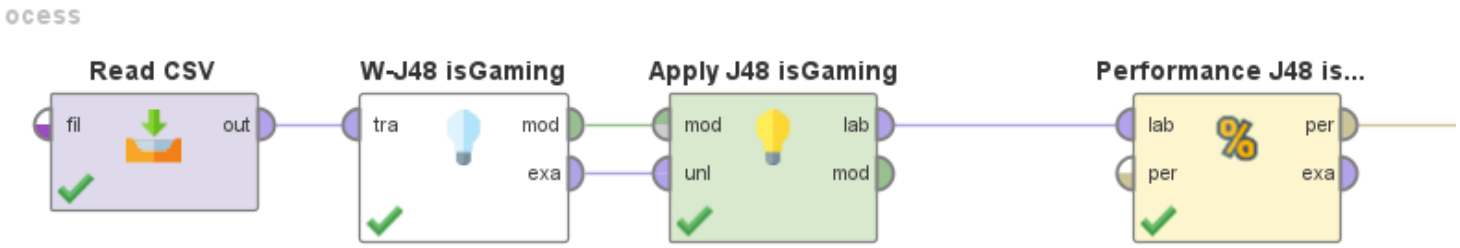
Q1 – (0.5 point)

Using the dataset named gaming.csv, create a classification model predicting the value of “isGaming” using the J48 algorithm. Report the model’s kappa.

> Kappa 0.648

kappa: 0.648

	true Not	true Gaming	class precision
pred. Not	3948	280	93.38%
pred. Gaming	99	428	81.21%
class recall	97.55%	60.45%	



Q2 – (0.5 point)

Evaluate the performance of this model under split-validation (Split Validation operator in RapidMiner). Use 70% of the data set as the training data and 30% as the test data. Use stratified sampling. Report the kappa metric.

> Kappa 0.367

kappa: 0.367

	true Not	true Gaming	class precision
pred. Not	1154	135	89.53%
pred. Gaming	60	77	56.20%
class recall	95.06%	36.32%	

Q3 – (0.5 point)

Evaluate the performance of this model under 10-fold cross-validation (Cross Validation operator in RapidMiner). Use stratified sampling. Report the kappa metric.

> Kappa 0.406 +/- 0.051 (micro average: 0.407)

kappa: 0.406 +/- 0.051 (micro average: 0.407)

	true Not	true Gaming	class precision
pred. Not	3808	406	90.37%
pred. Gaming	239	302	55.82%
class recall	94.09%	42.66%	

Q4 – (0.5 point)

Using the dataset named gaming-student-cross-validation.csv, evaluate the performance of this model under leave-one-out student-level cross-validation (Cross Validation operator in RapidMiner). Report the kappa metric.

> Kappa 0.402

kappa: 0.402

	true 0	true 1	class precision
pred. 0	3808	410	90.28%
pred. 1	239	298	55.49%
class recall	94.09%	42.09%	

Feature selection

Q5 – (0.5 point)

Using the dataset named gaming.csv, use a forward selection approach to create a classification model predicting the value of “isGaming” using the J48 algorithm. Report the model’s kappa.

> Kappa 0.635

kappa: 0.635

	true Not	true Gaming	class precision
pred. Not	3925	278	93.39%
pred. Gaming	122	430	77.90%
class recall	96.99%	60.73%	

Q6 – (0.5 point)

Using the dataset named gaming.csv, use a backward elimination approach to create a classification model predicting the value of “isGaming” using the J48 algorithm. Report the model’s kappa.

> Kappa 0.703
>
> kappa: 0.703
>
> | | true Not | true Gaming | class precision |
> |---|---|---|---|
> | pred. Not | 3941 | 227 | 94.55% |
> | pred. Gaming | 106 | 481 | 81.94% |
> | class recall | 97.38% | 67.94% | |

Cognitive Tutor Algebra gaming model

Q7 – (2 points)

You will use the features created by the students in the previous semester for Hands-on-data #1 (included in the file CognitiveTutorAlgebra-gaming-resampled.csv). Using the features, attempt to build the best possible model predicting whether students are gaming the system or not. Use different modeling algorithms, attempt different feature selection approaches. Feel free to create new features that can be included in your data file or to use the features you created in Hands-on-data #1. The file named CognitiveTutorAlgebra-action-logs-with-gaming-clips.xlsx can be used to compute additional features. Since the final file (CognitiveTutorAlgebra-gaming-resampled.csv) has been resampled, make sure to only use the same clips as the ones included in the csv file. The model should be cross validated using the “batch” variable included in the csv file. Provide a text explaining the different attempts that you made at building the “best” model. For example, provide information about which algorithms were used or which feature selection approaches were attempted. Provide performance metrics for each of those attempts.

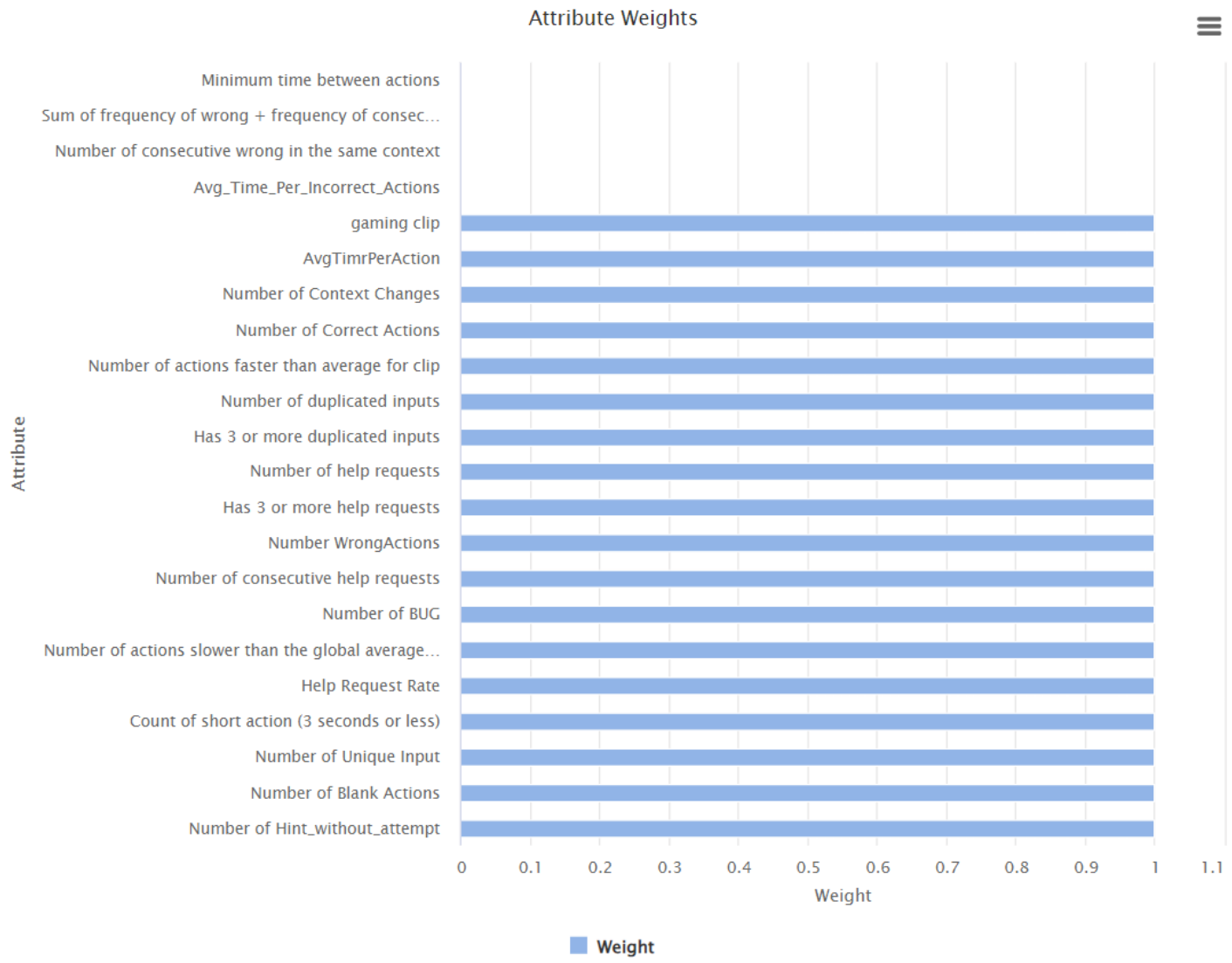
Bonus point: The model with best performance (using the Kappa metric) will be awarded 1 bonus point for the assignment. The model with the second-best performance will be awarded 0.5 bonus point.

> All of the following have been evaluated using cross-validation split on the batch attribute.
>
> ### Linear Regression
>
> I applied forward selection on Linear Regression with the following parameters:

Forward Selection (2) (Forward Selection)

maximal number of attributes	20
speculative rounds	5
stopping behavior	without increase

The weights suggested I remove the 4 features with 0 weight (shown in the following graph)



Training using M5 yielded the following Kappa:

kappa: 0.178 +/- 0.080 (micro average: 0.188)

	true N	true G	class precision
pred. N	575	355	61.83%
pred. G	241	329	57.72%
class recall	70.47%	48.10%	

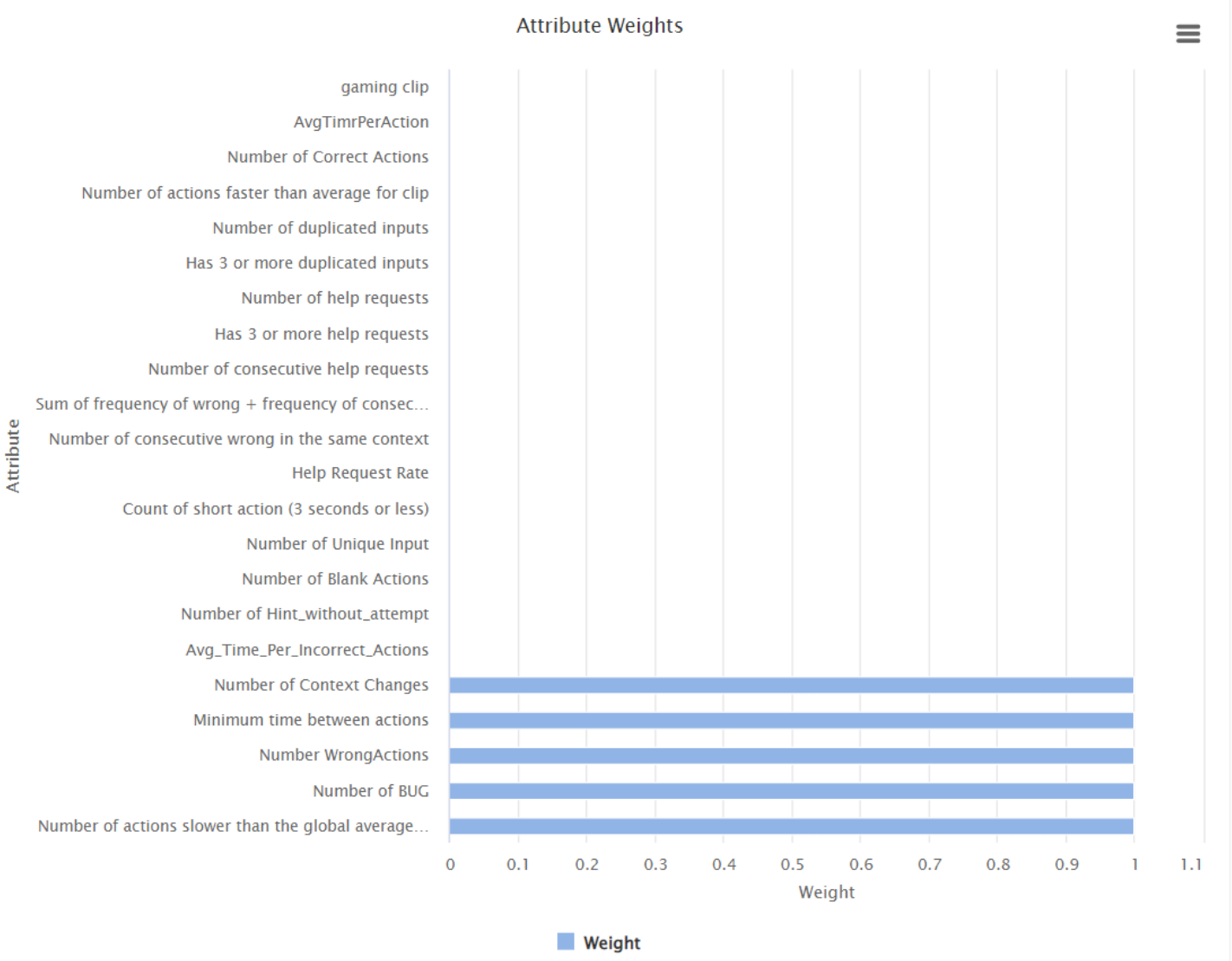
Greedy gave slightly better results

kappa: 0.184 +/- 0.081 (micro average: 0.197)

	true N	true G	class precision
pred. N	581	354	62.14%
pred. G	235	330	58.41%
class recall	71.20%	48.25%	

Random Tree

These were the attributes selected using the same method previously described but applied to the Random Tree algorithm:



These were the results using only the selected features:

kappa: 0.152 +/- 0.083 (micro average: 0.164)

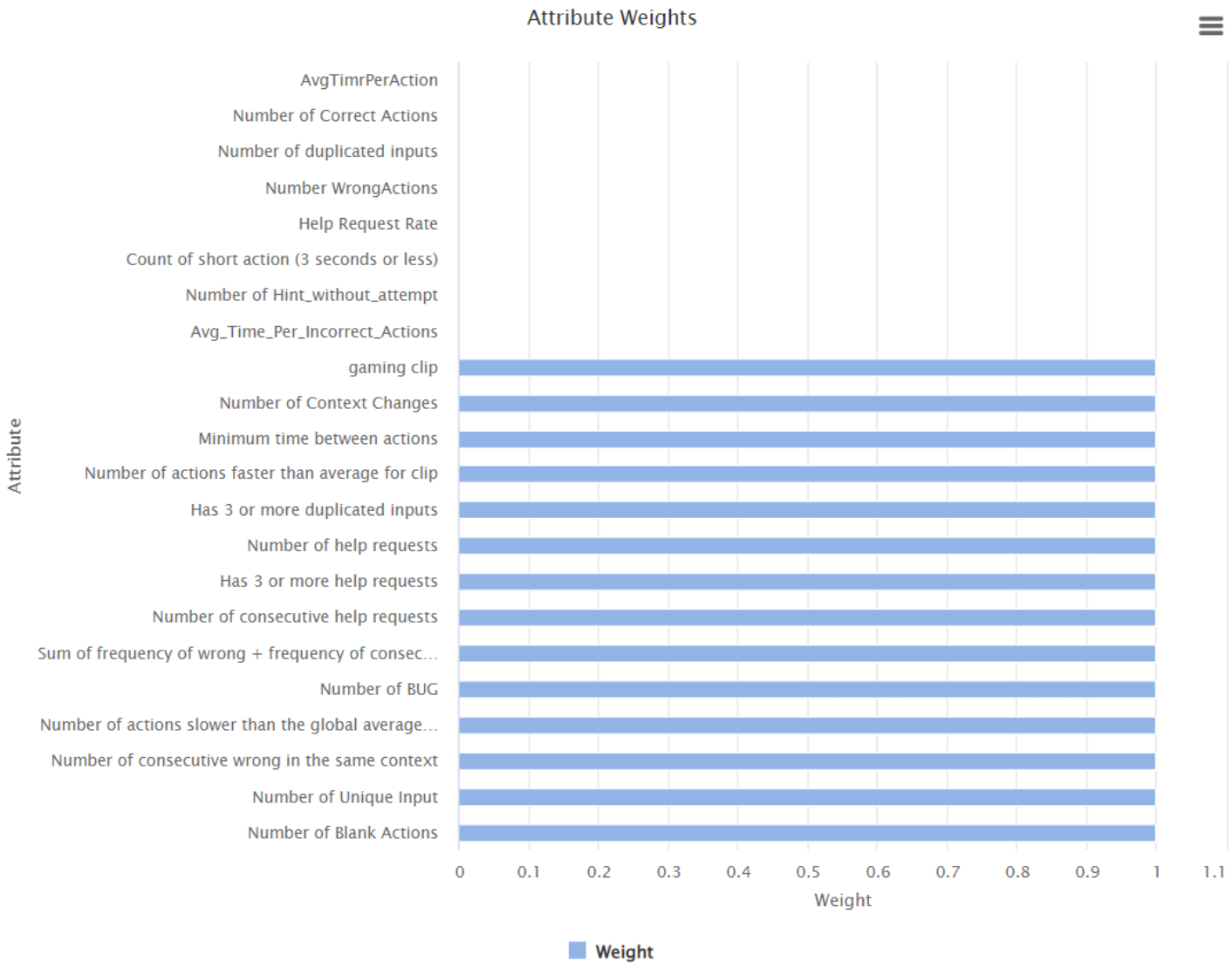
	true N	true G	class precision
pred. N	607	399	60.34%
pred. G	209	285	57.69%
class recall	74.39%	41.67%	

These were the results using all features:

kappa: 0.083 +/- 0.101 (micro average: 0.072)

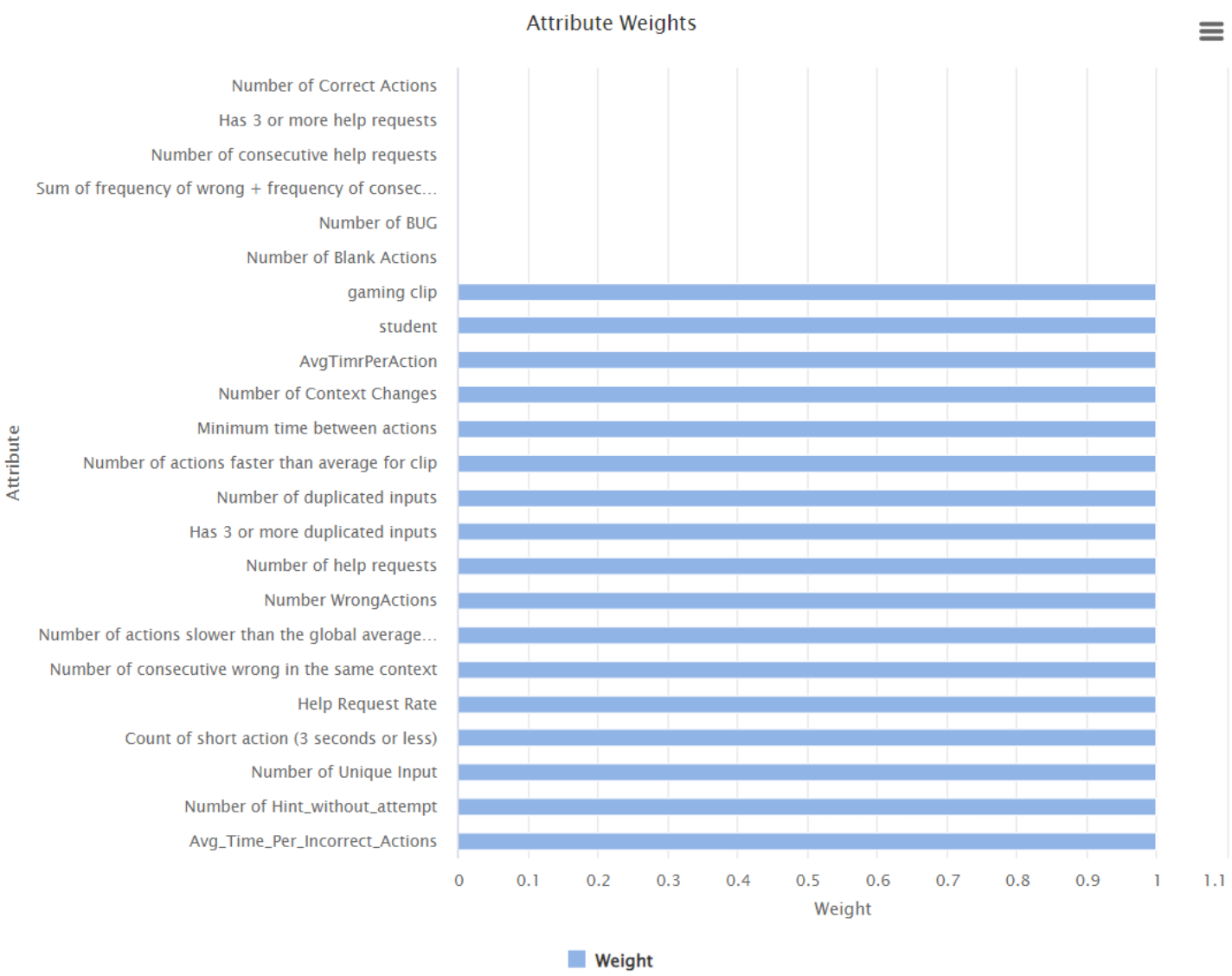
	true N	true G	class precision
pred. N	707	546	56.42%
pred. G	109	138	55.87%
class recall	86.64%	20.18%	

REPTree



With or without the attribute selection, REPTree did extremelly poorly (less than .1 kappa)

J48



Performance after filtering features:

kappa: 0.062 +/- 0.084 (micro average: 0.073)

	true N	true G	class precision
pred. N	536	400	57.26%
pred. G	280	284	50.35%
class recall	65.69%	41.52%	

Best Performing Algorithm

Because Greedy Linear Regression was the most promising algorithm I could find, I decided to take another look at it in case I could make any improvements to its performance.

After taking a look at the model, I noticed I had not filtered the gaming clip and the batch attributes. After removing them, the model's kappa increased to 0.195:

kappa: 0.195 +/- 0.093 (micro average: 0.193)

	true N	true G	class precision
pred. N	589	364	61.80%
pred. G	227	320	58.50%
class recall	72.18%	46.78%	