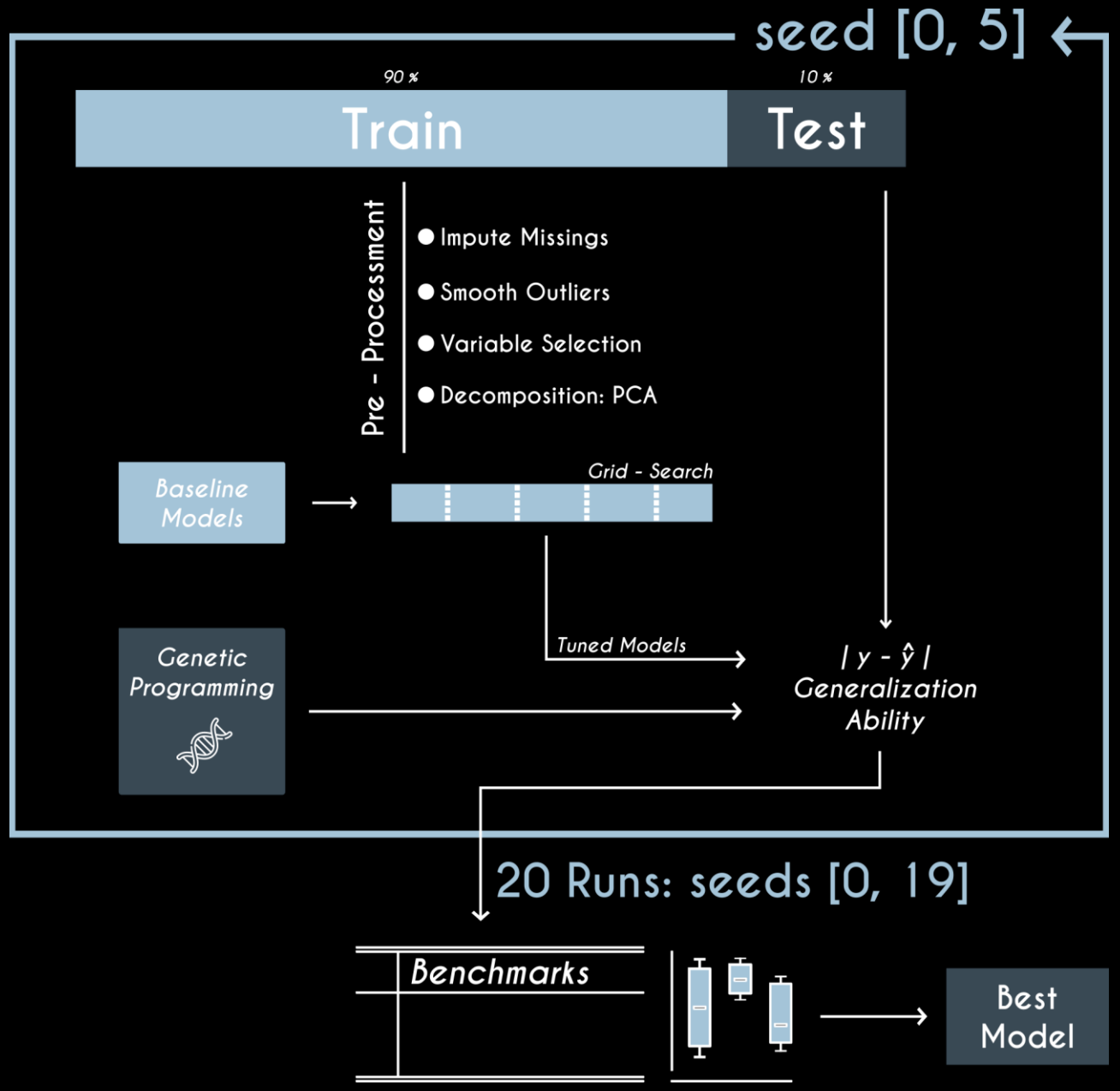
CHILDREN'S ALCOHOL CONSUMPTION

Machine Learning

Data Science and Advanced Analytics

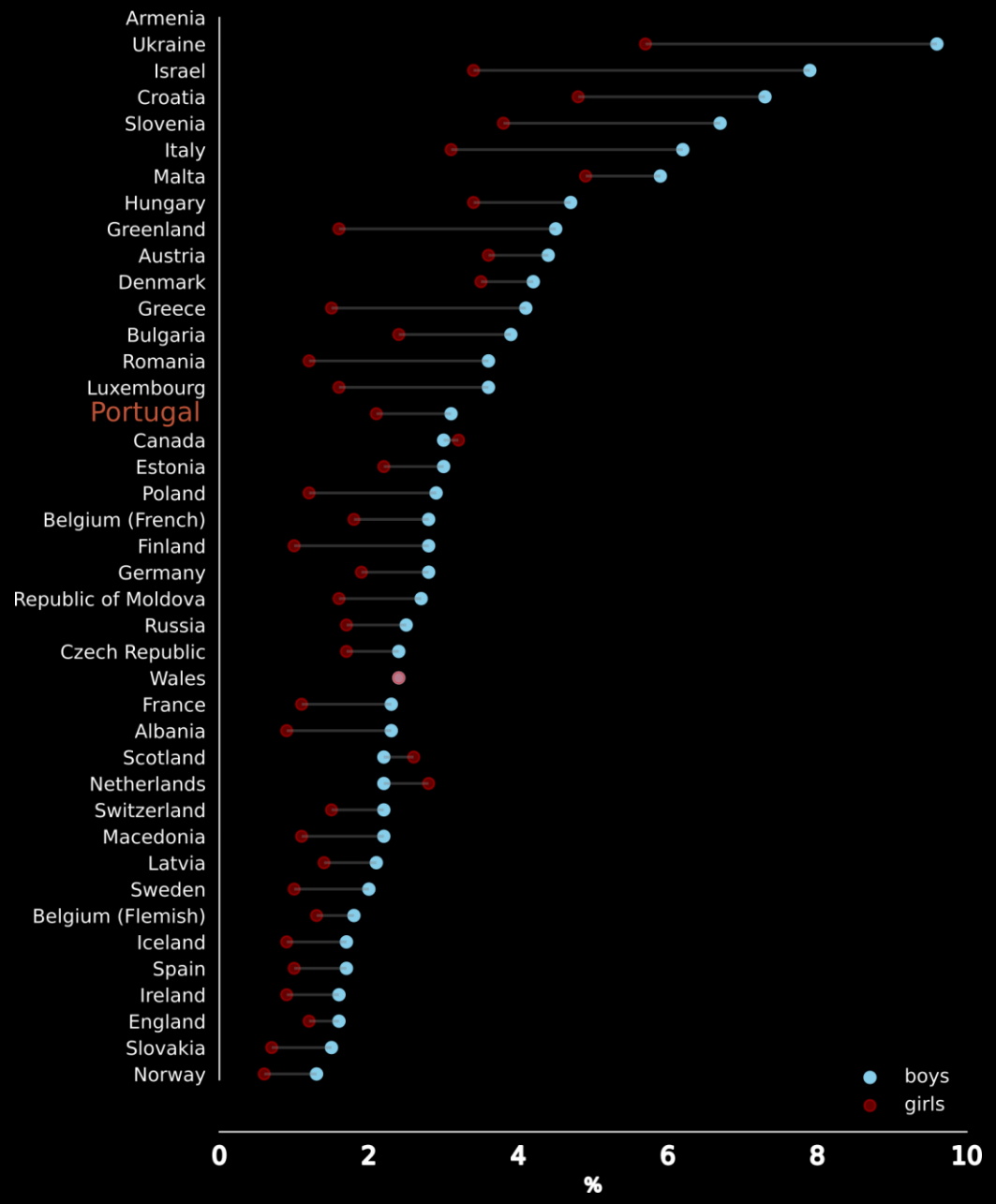
Rita Franco	m20180080
Rodrigo Umbelino	m20180060
Vitor Manita	m20180054

Process

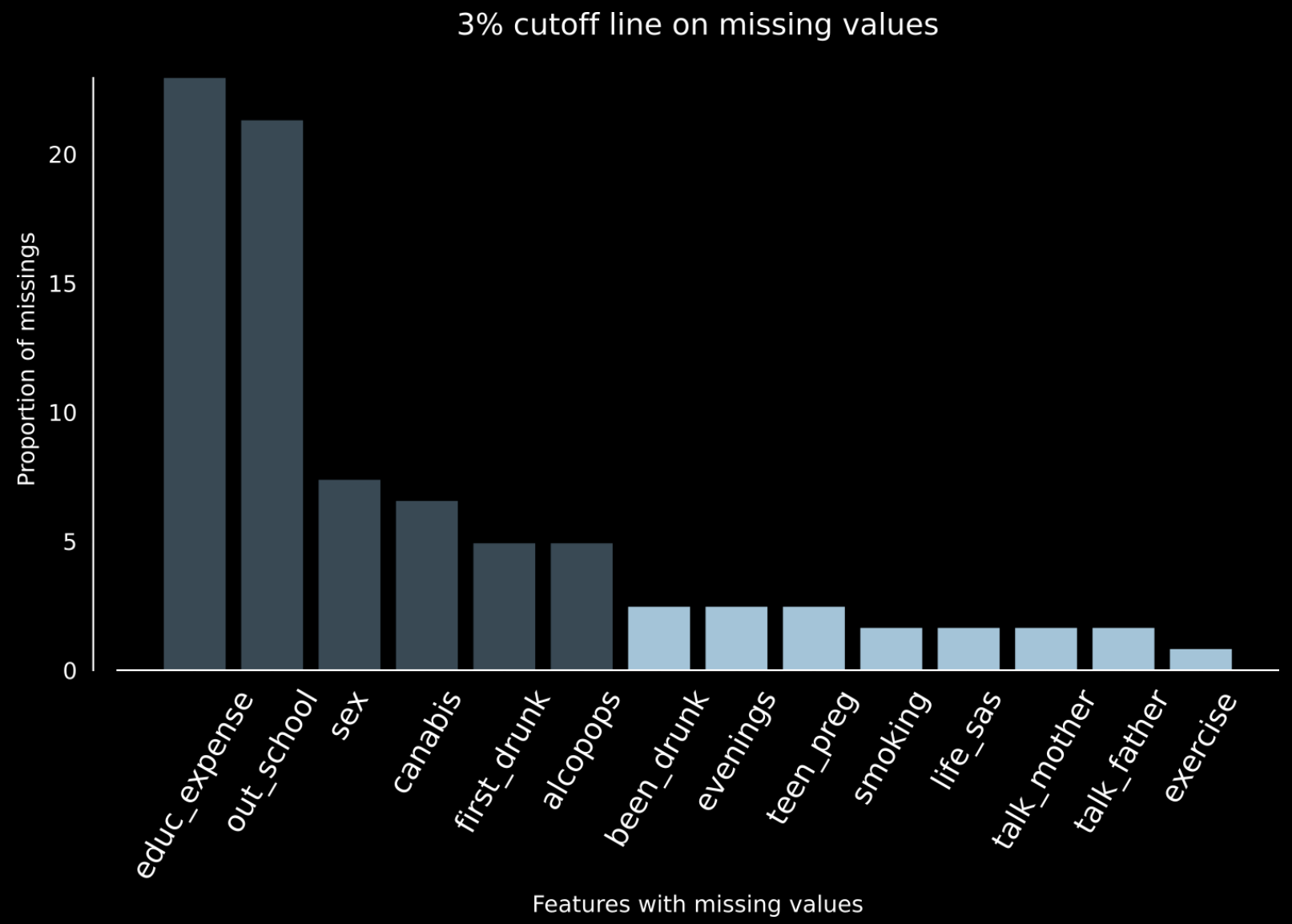

seed [0, 5]
90 %
10 %
Train
Test
Pre - Processment
Impute Missings
Smooth Outliers
Variable Selection
Decomposition: PCA
Baseline Models
Grid - Search
Tuned Models
Genetic Programming
| y - ŷ |
Generalization Ability
20 Runs: seeds [0, 19]
Benchmarks
Best Model

Explore

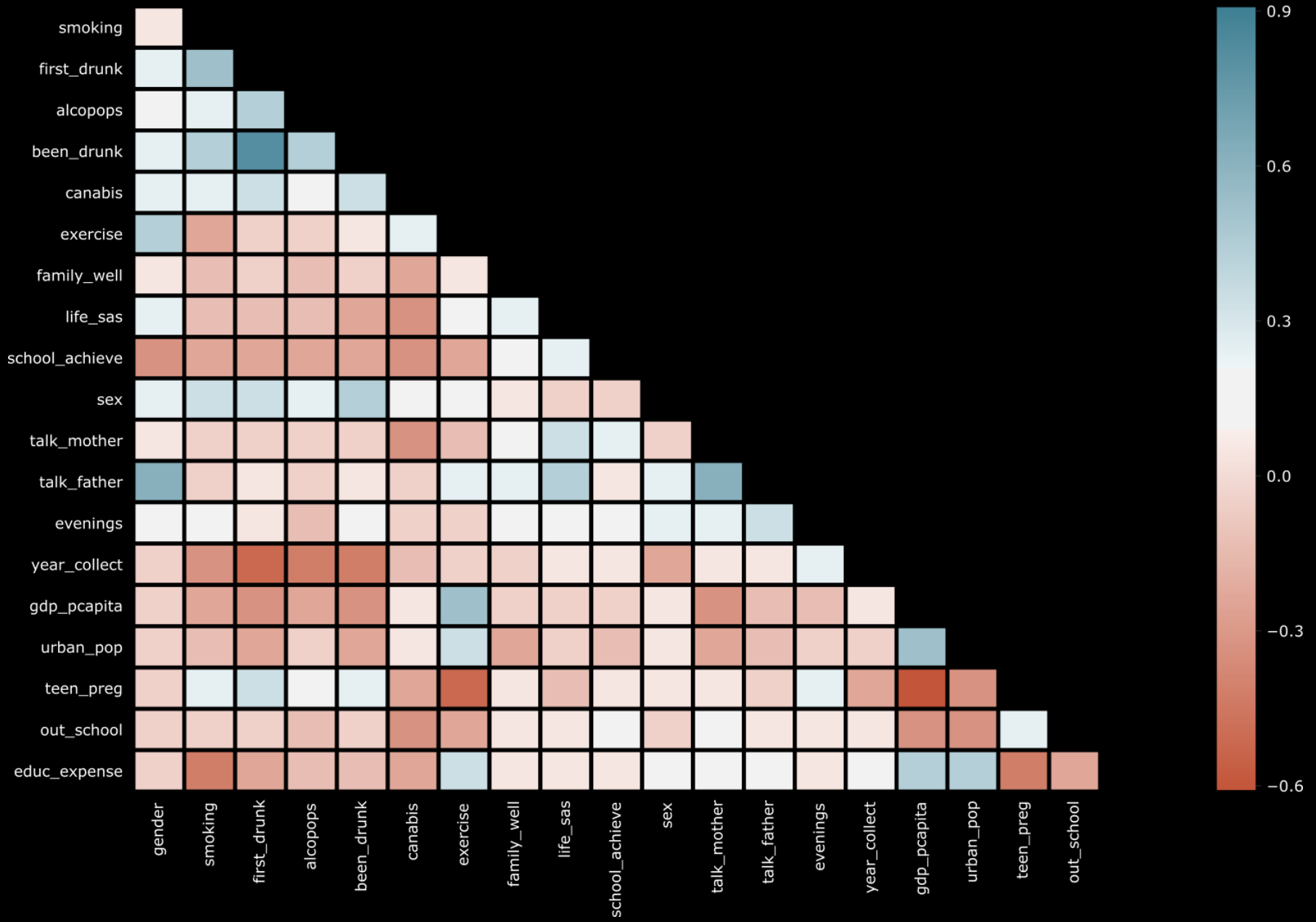
Portugal is the 15th country where boys claim they drink more often



Explore


3% cutoff line on missing values
Proportion of missings
0
5
10
15
20
educ_expense
out_school
sex
canabis
first_drunk
alcopops
been_drunk
evenings
teen_preg
smoking
life_sas
talk_mother
talk_father
exercise
Features with missing values

Explore


smoking
first_drunk
alcopops
been_drunk
canabis
exercise
family_well
life_sas
school_achieve
sex
talk_mother
talk_father
evenings
year_collect
gdp_pcapita
urban_pop
teen_preg
out_school
educ_expense
gender
smoking
first_drunk
alcopops
been_drunk
canabis
exercise
family_well
life_sas
school_achieve
sex
talk_mother
talk_father
evenings
year_collect
gdp_pcapita
urban_pop
teen_preg
out_school
0.9
0.6
0.3
0.0
−0.3
−0.6

Modify

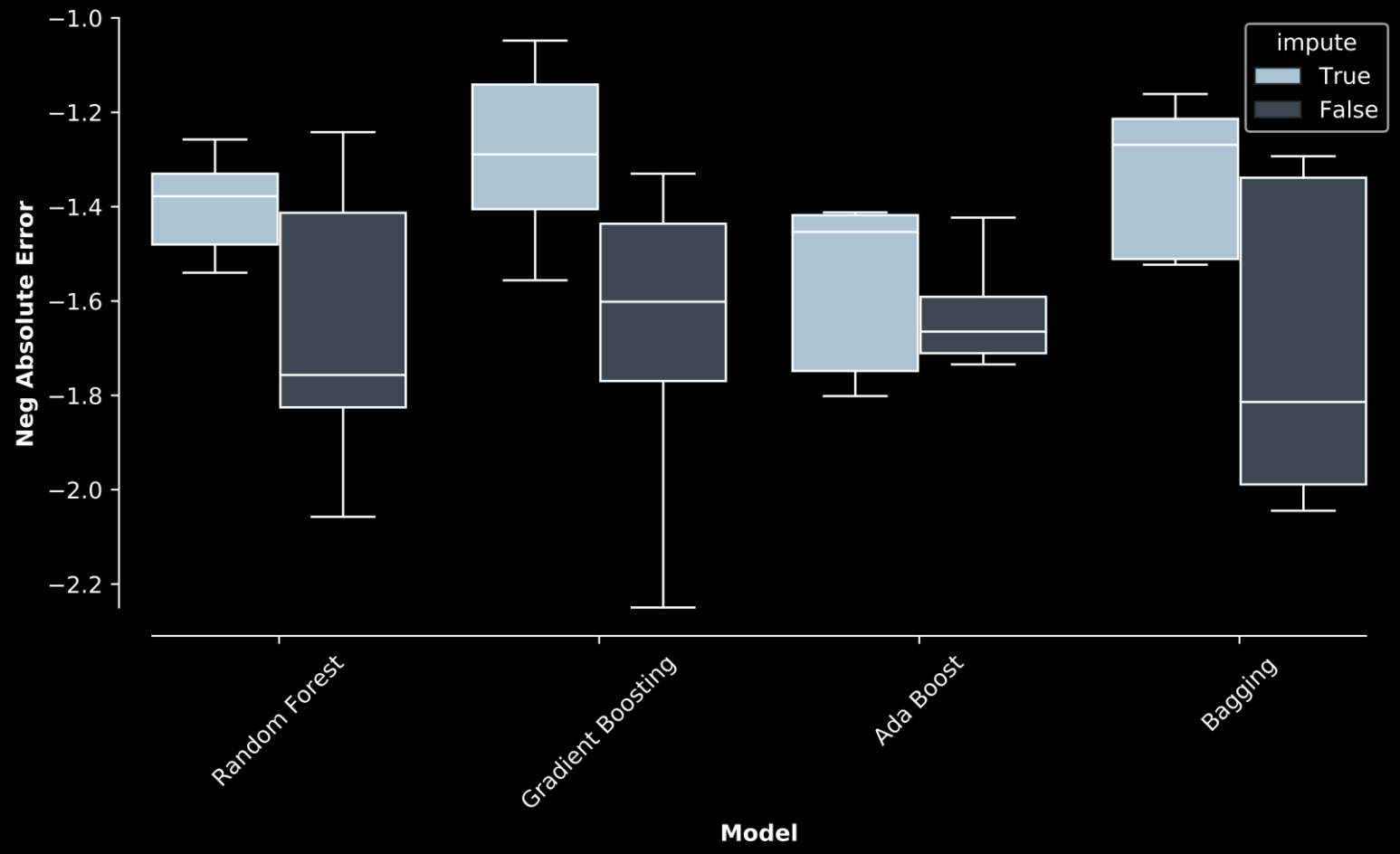
Our strategy

{(T/F), (T/F), (T/F), (T/F)}

Impute Missings

{(T/F), (T/F), (T/F), (T/F)}

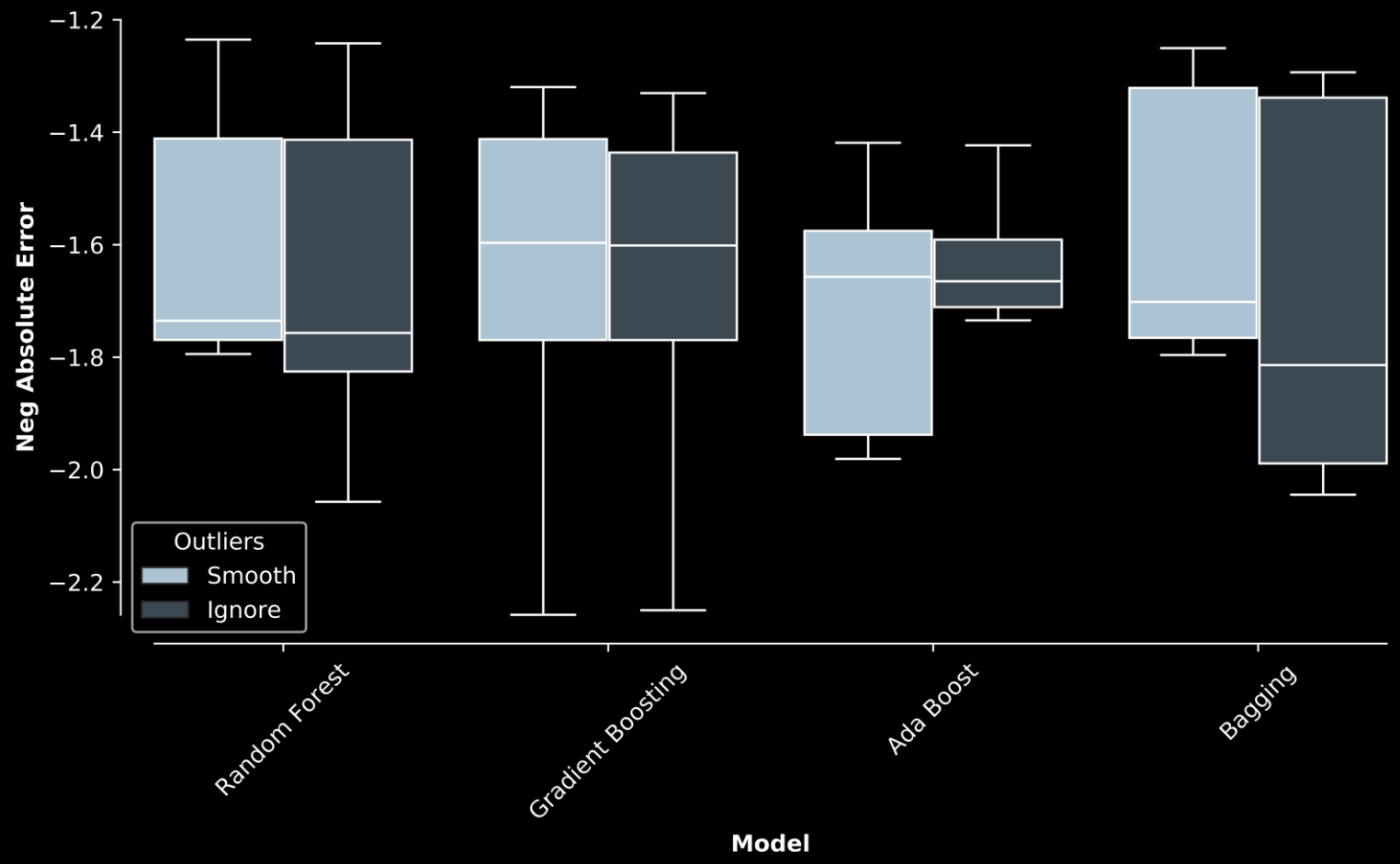
Using the distribution of each feature, fill missings by mean or median



Smooth Outliers

{(T/F), (T/F), (T/F), (T/F)}

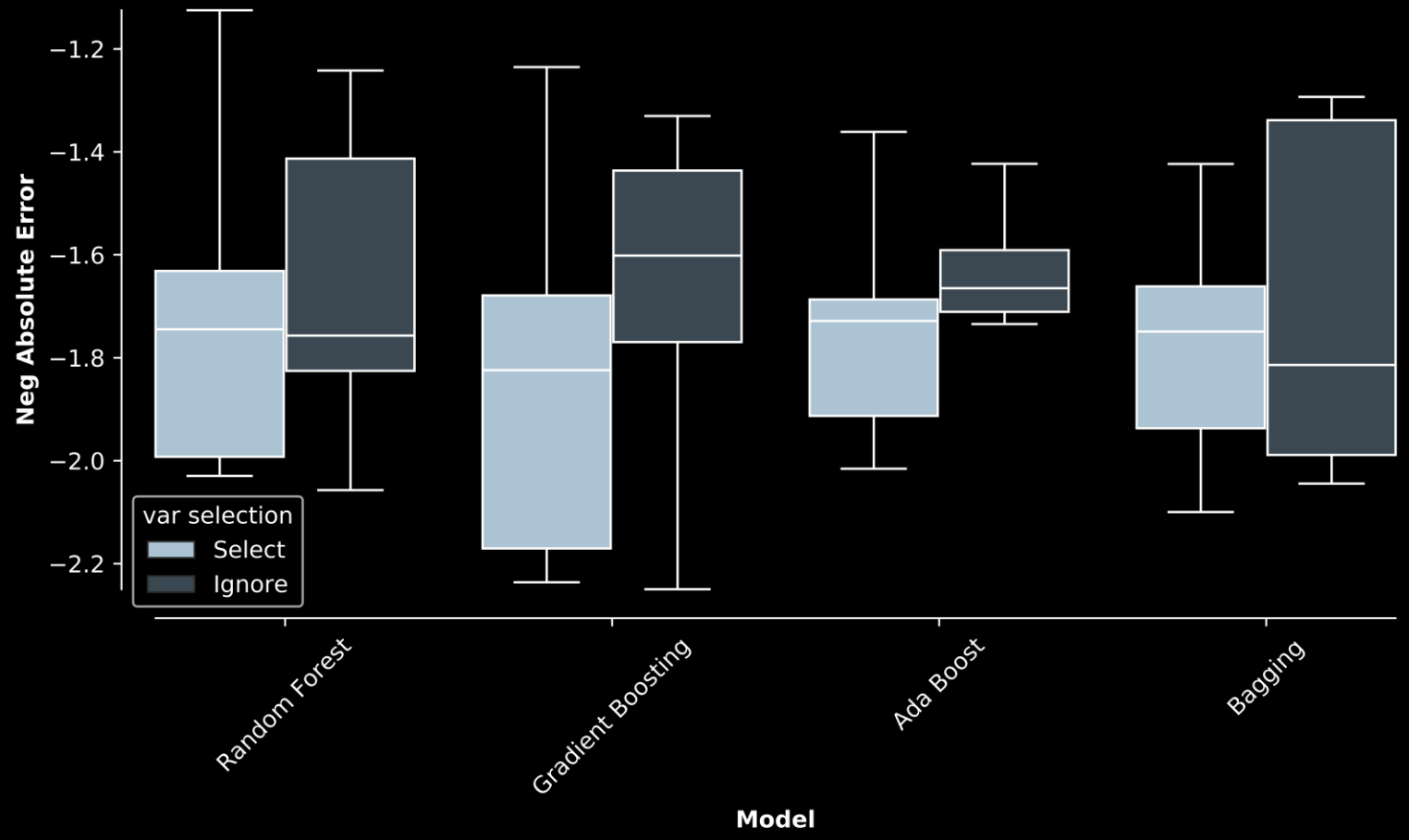
Using winsoring with percentiles 5% and 95%



Variable Selection

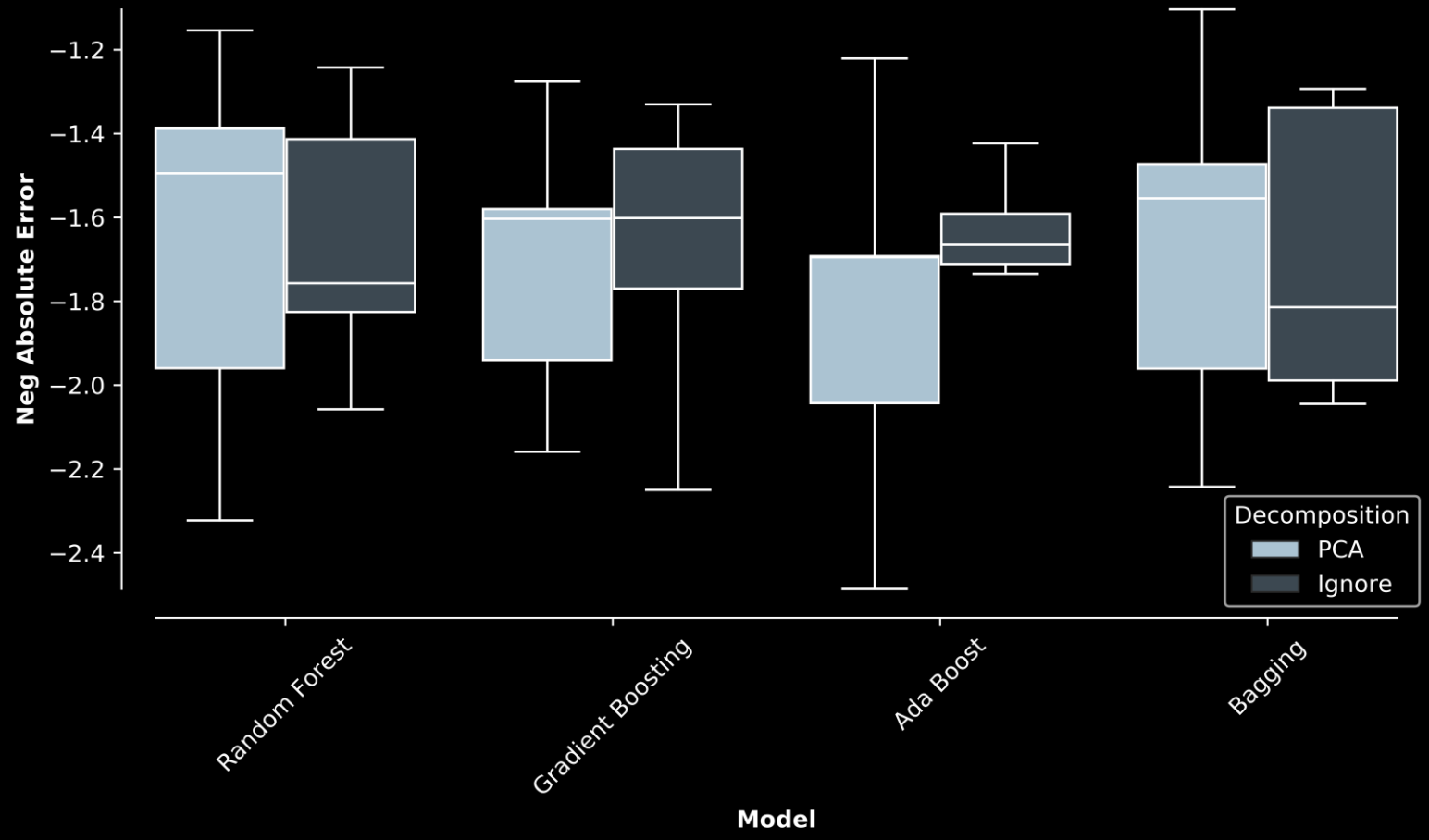
{(T/F), (T/F), (T/F), (T/F)}

Using Recursive Feature Extraction with Linear Regression



Feature Decomposition {(T/F), (T/F), (T/F), (T/F)}

Using Principal Component Analysis. Automatic Threshold of 80% explained variance



Ensembles Benchmarks

	Model	Impute Missings	Smooth Outliers	Variable Selection	PCA	NMAE	MAE std
Best 5 Models	**Gradient Boost**	True	True	False	False	-1.286	0.183
	Gradient Boost	True	False	False	False	-1.286	0.182
	Gradient Boost	True	False	True	False	-1.286	0.182
	Bagging	True	False	True	False	-1.320	0.160
	Gradient Boost	True	True	False	False	-1.330	0.160
Worst 5 Models	**Bagging**	False	True	True	True	-1.963	0.377
	Random Forest	False	True	True	True	-1.845	0.317
	Gradient Boost	False	True	True	True	-1.842	0.346
	Bagging	True	True	True	True	-1.838	0.374
	Gradient Boost	False	False	True	False	-1.829	0.363

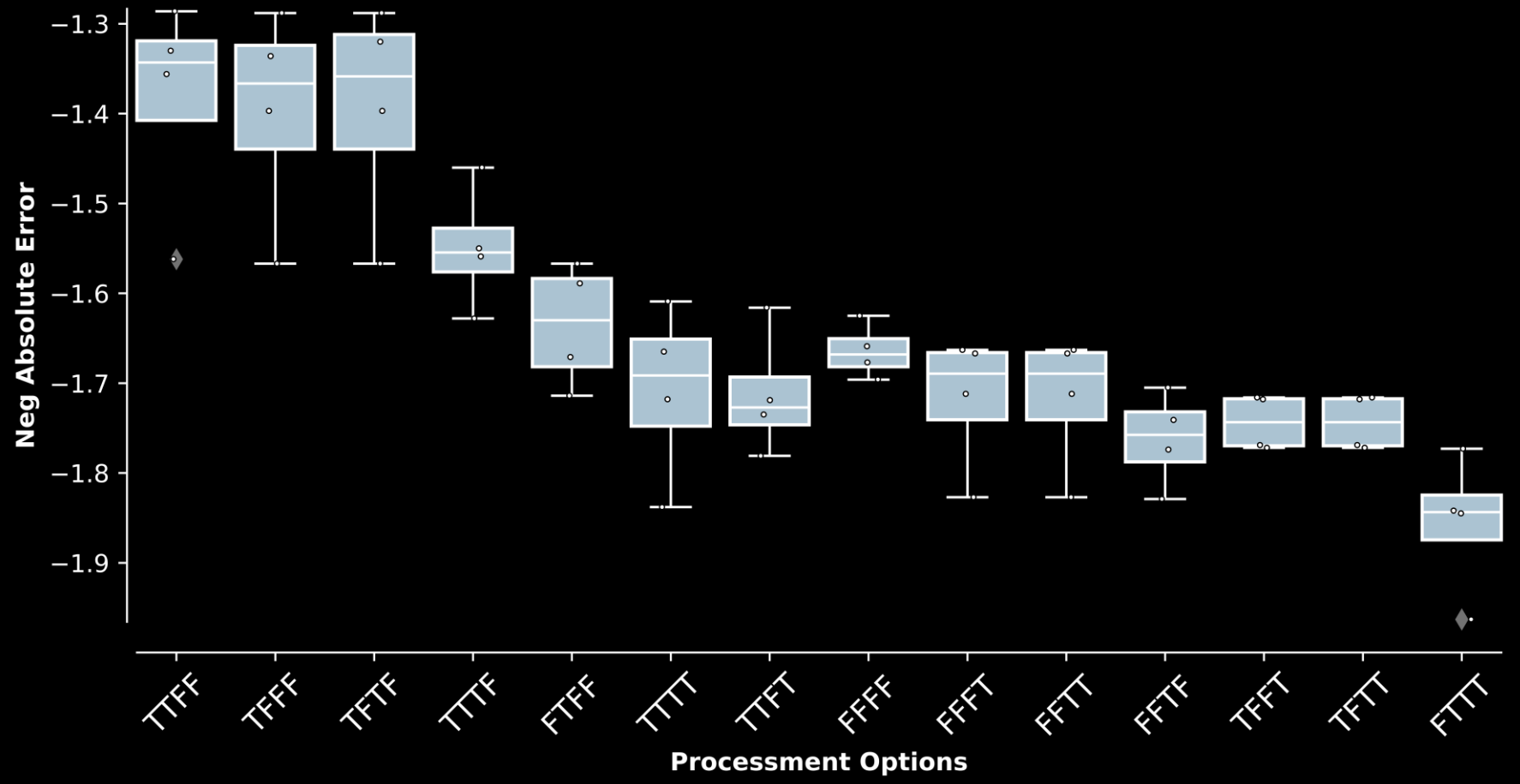
Grid Search in each model with 5 cross validations was performed

Ensembles Benchmarks

	Model	Impute Missings	Smooth Outliers	Variable Selection	PCA	NMAE	MAE std
Best 5 Models	**Gradient Boost**	**True**	True	False	**False**	-1.286	0.183
	Gradient Boost	**True**	False	False	**False**	-1.286	0.182
	Gradient Boost	**True**	False	True	**False**	-1.286	0.182
	Bagging	**True**	False	True	**False**	-1.320	0.160
	Gradient Boost	**True**	True	False	**False**	-1.330	0.160
Worst 5 Models	**Bagging**	False	True	**True**	True	-1.963	0.377
	Random Forest	False	True	**True**	True	-1.845	0.317
	Gradient Boost	False	True	**True**	True	-1.842	0.346
	Bagging	True	True	**True**	True	-1.838	0.374
	Gradient Boost	False	False	**True**	False	-1.829	0.363

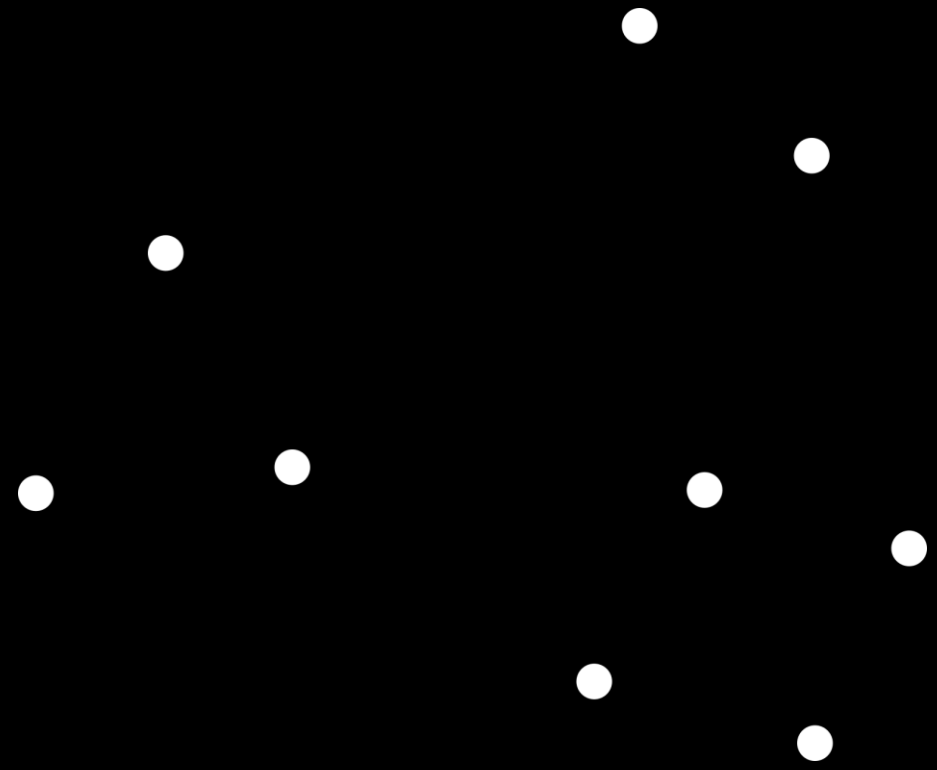
Ensembles Benchmarks

{T, T, F, F}



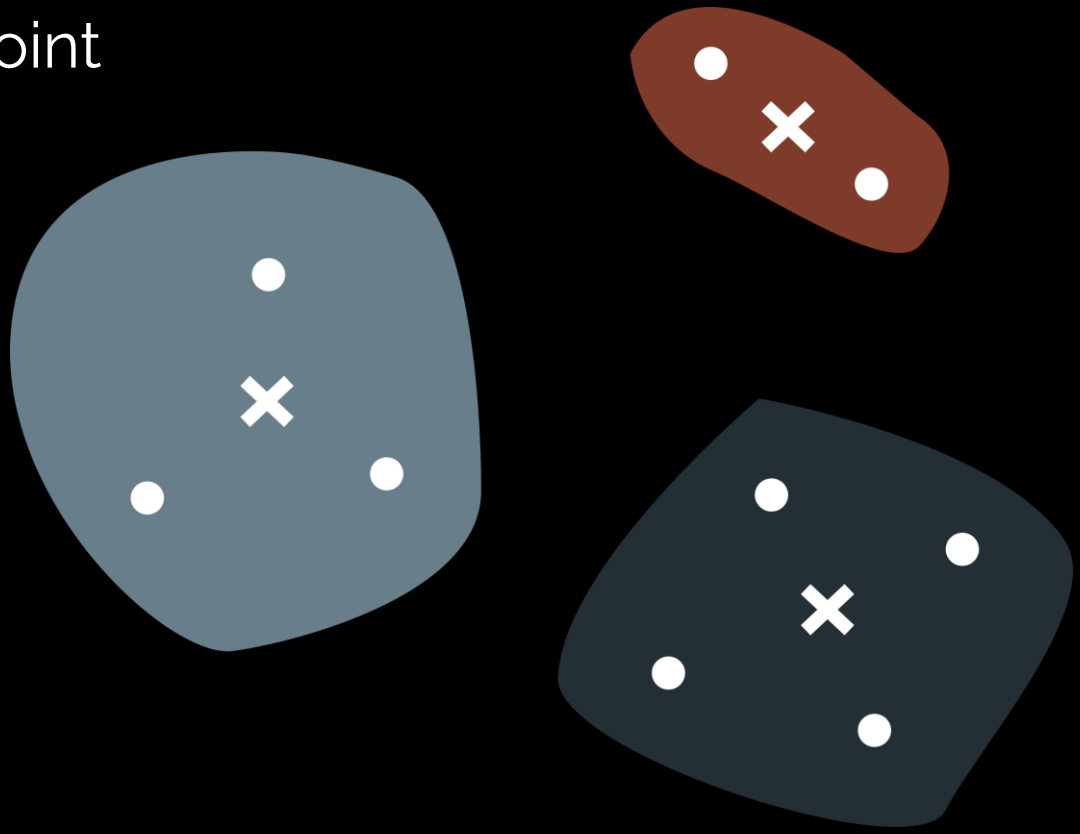
Artificial Data

1. Generate K Clusters (K-means)



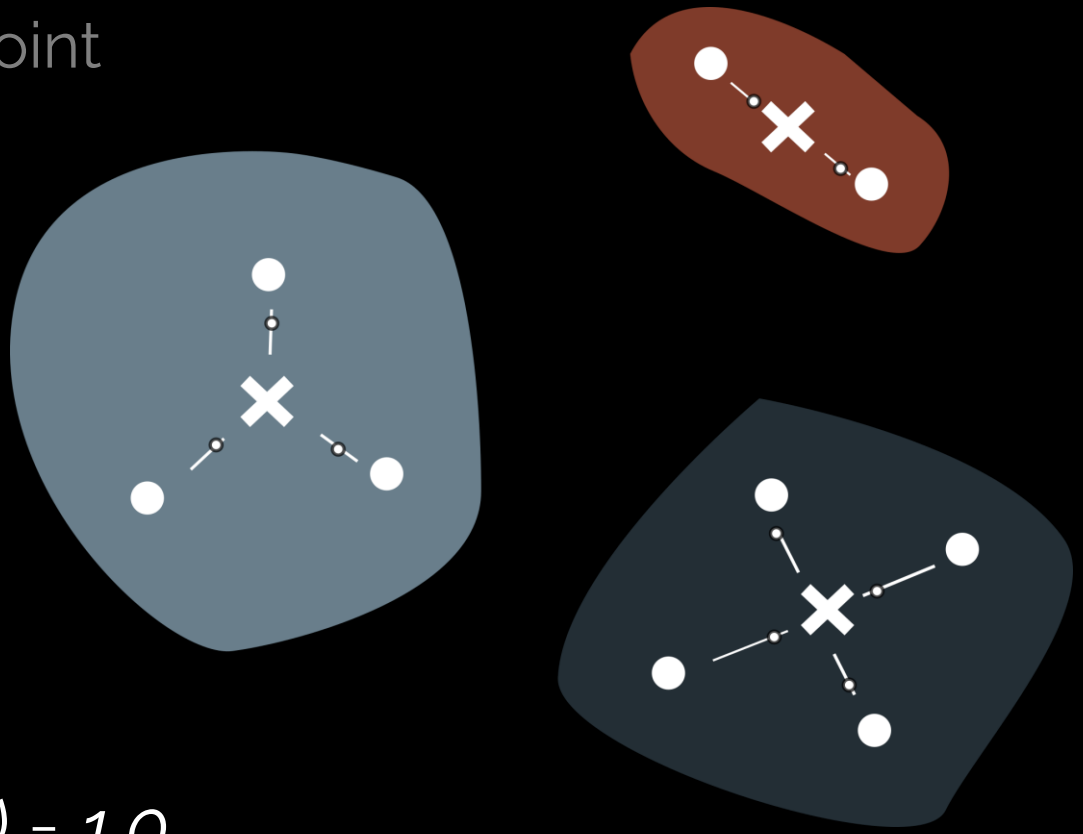
Artificial Data

1. Generate K Clusters (K-means)
2. Generate random coordinates between centroids and each corresponding point



Artificial Data

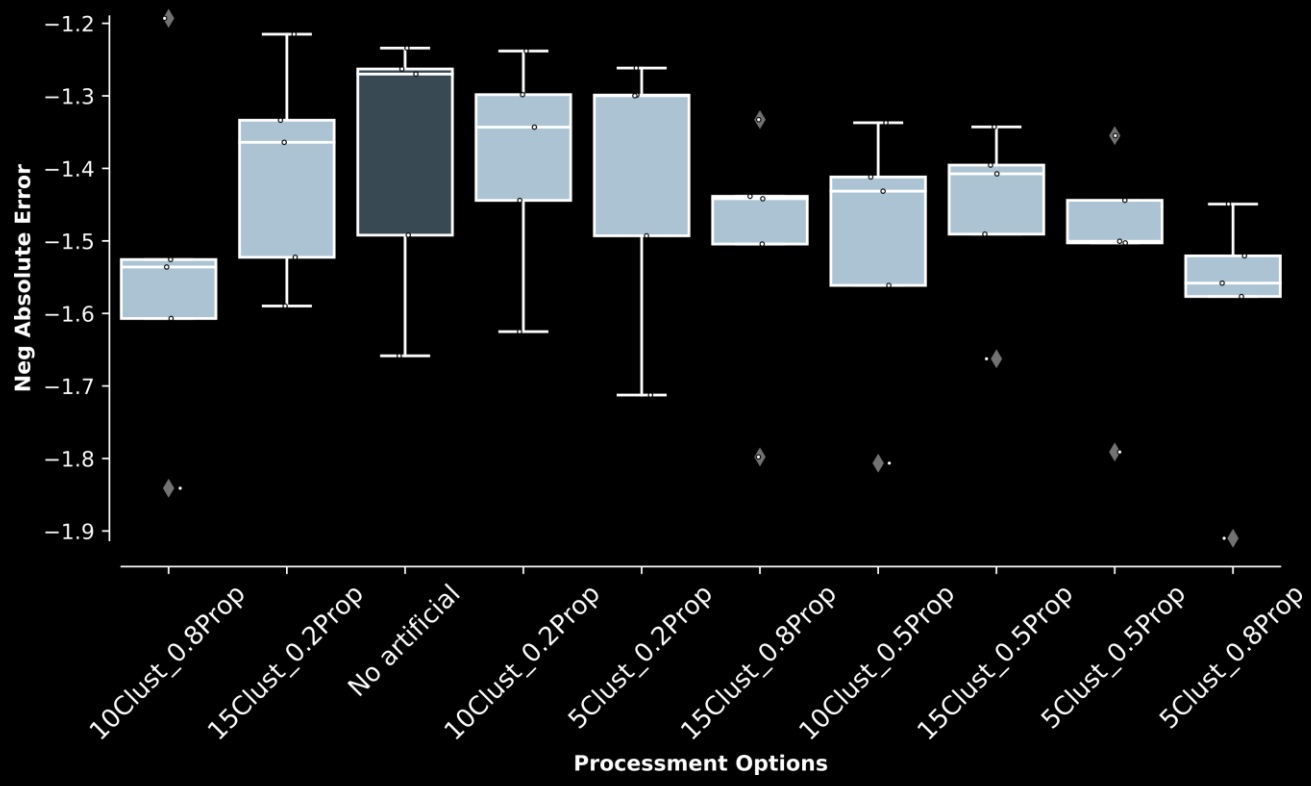
1. Generate K Clusters (K-means)
2. Generate random coordinates between centroids and each corresponding point
3. Generate p proportion of new data



P (proportion) = 1.0

Artificial Data

	Nr. Clusters	Increase Proportion	NMAE
No Artificial	0	0	-1.383
Artificial	10	20%	-1.390
	15	20%	-1.405
	5	20%	-1.413
	15	50%	-1.460
	15	80%	-1.503
	10	50%	-1.510
	5	50%	-1.519
	10	80%	-1.540
	5	80%	-1.603



Artificial Data

	Nr. Clusters	Increase Proportion	NMAE
No Artificial	**0**	**0**	**-1.383**
Artificial	10	20%	-1.390
	15	20%	-1.405
	5	20%	-1.413
	15	50%	-1.460
	15	80%	-1.503
	10	50%	-1.510
	5	50%	-1.519
	10	80%	-1.540
	5	80%	-1.603

Genetic Programming

Selection

Roulette Wheel
Stochastic Universal Sampling
Random
Bloat Control
Rank

Genetic Programming

Selection

Roulette Wheel
Stochastic Universal Sampling
Random
Bloat Control
Rank

Crossover

Uniform
Simple
2 Tree Crossover

Genetic Programming

Selection

Roulette Wheel
Stochastic Universal Sampling
Random
Bloat Control
Rank

Crossover

Uniform
Simple
2 Tree Crossover

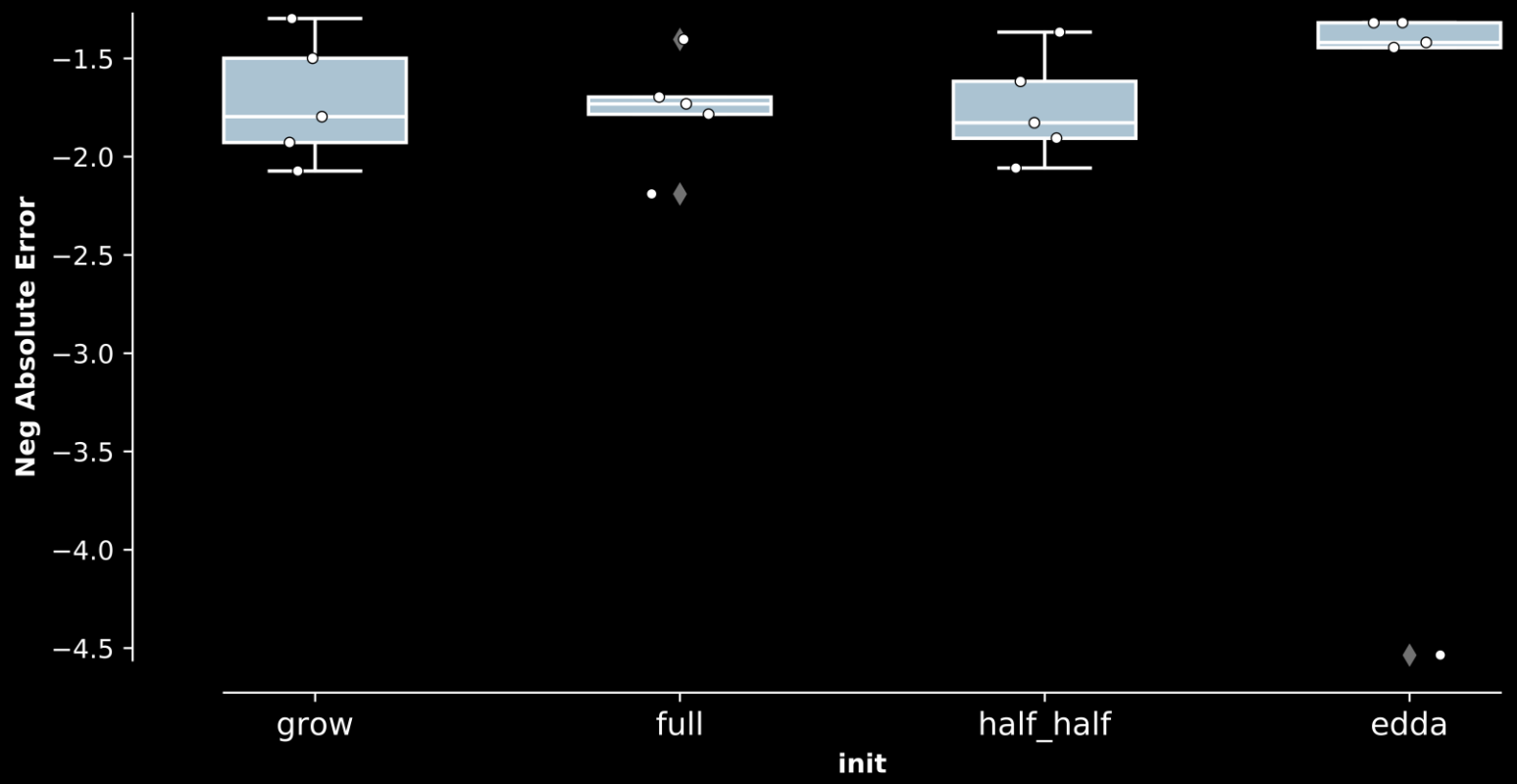
Mutation

Reverse
Shake
Graft
Swap
Semantical Sig

GP - Initial Benchmark

Initialization

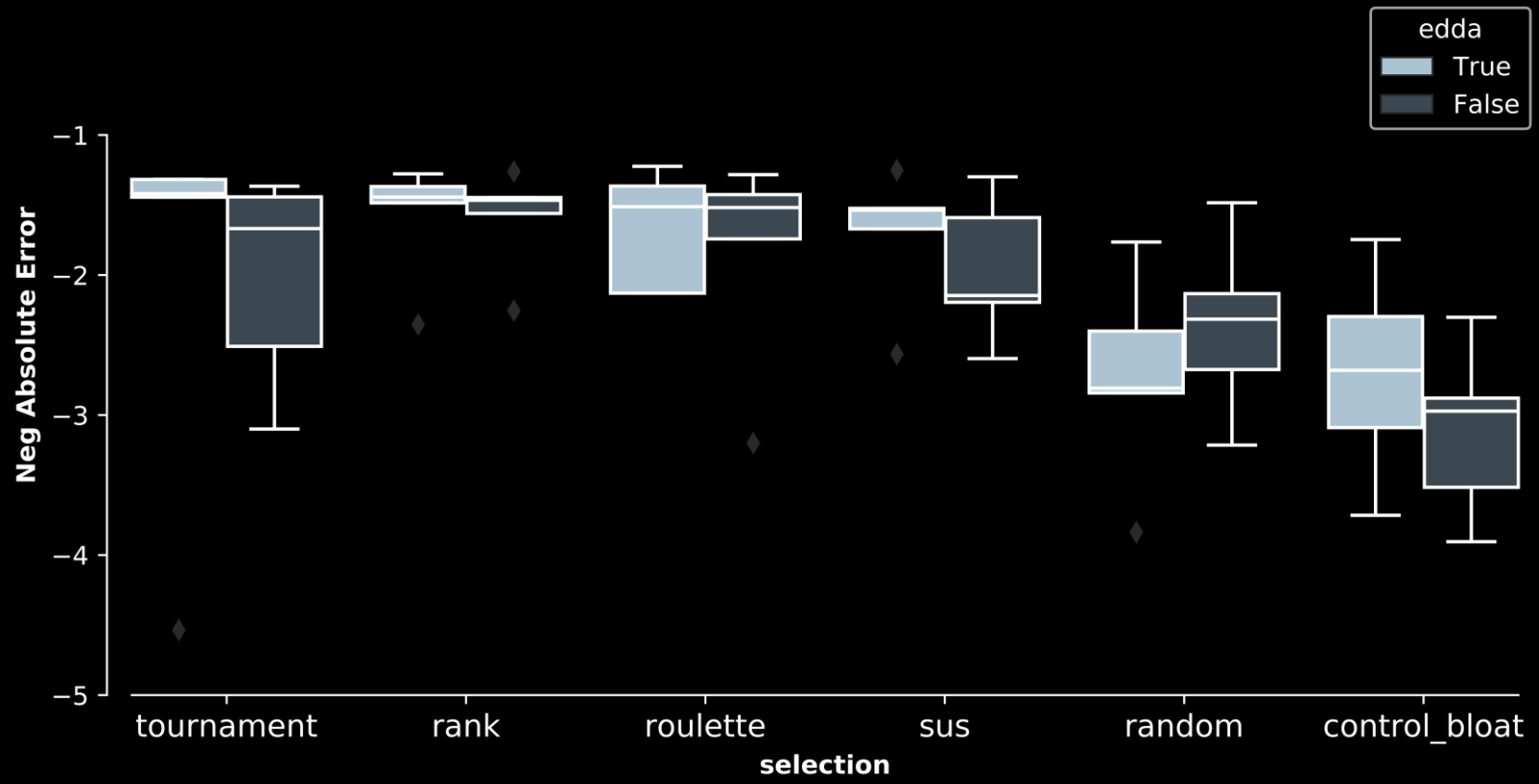
200 Generations
200 Population size
Parsimony Coefficient = 0.001



GP – Initial Benchmark

Selection

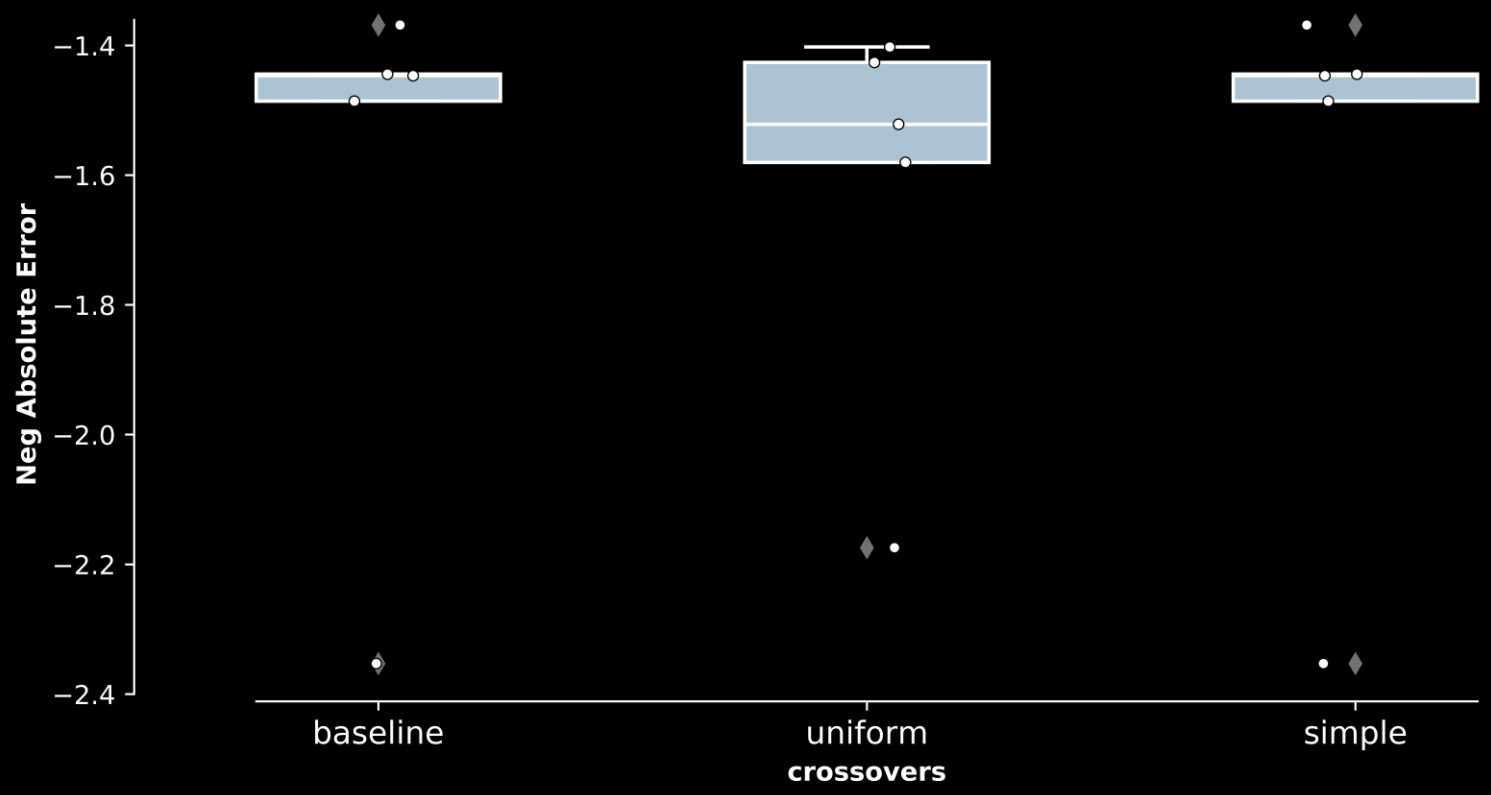
Effect of Edda on generalization ability dispersion is visible



GP – Initial Benchmark

Crossover

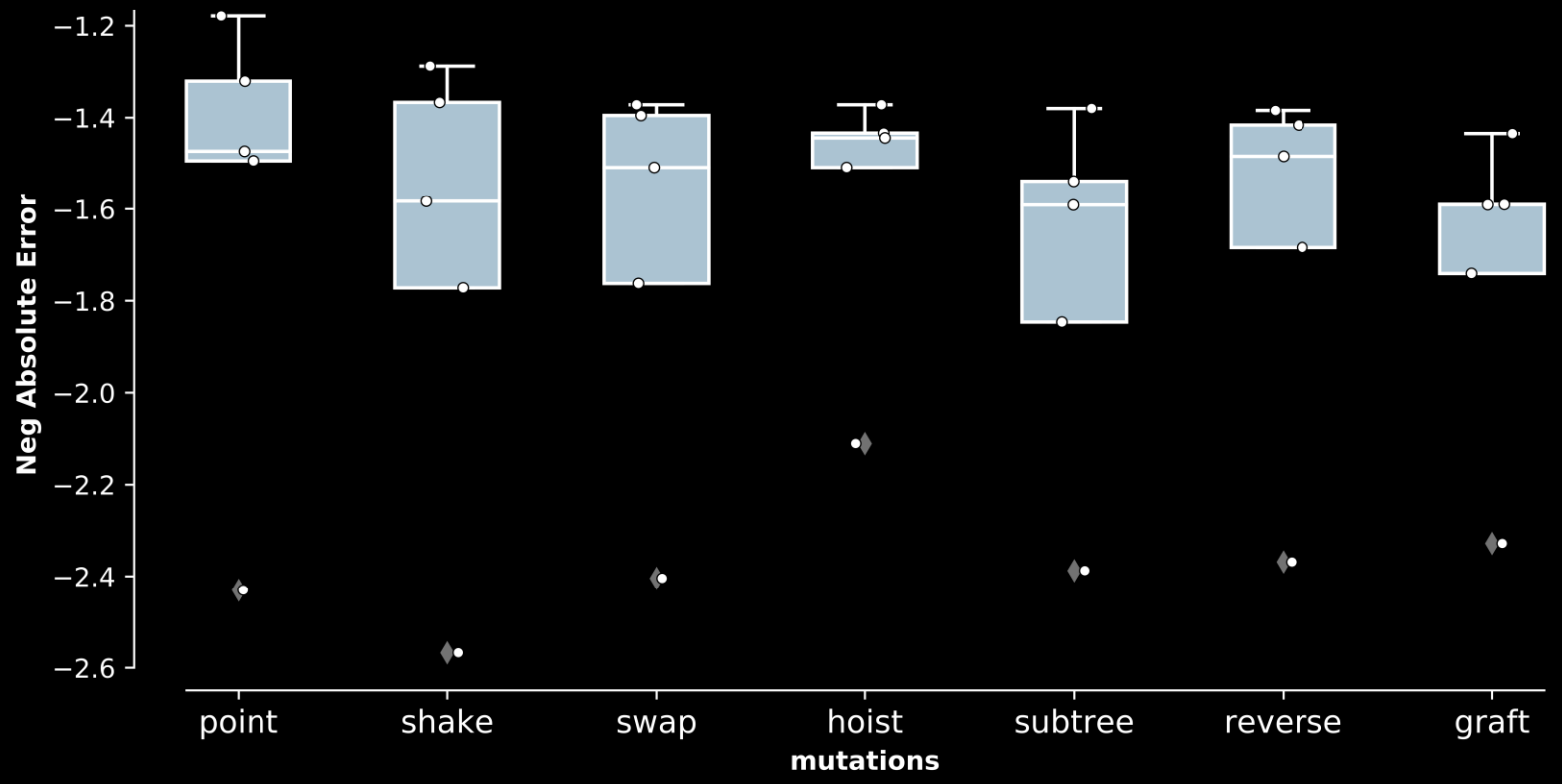
Prob. Crossover = 0.9
Prob. Mutation = 0.1



GP – Initial Benchmark

Mutation

Prob. Crossover = 0.1
Prob. Mutation = 0.9

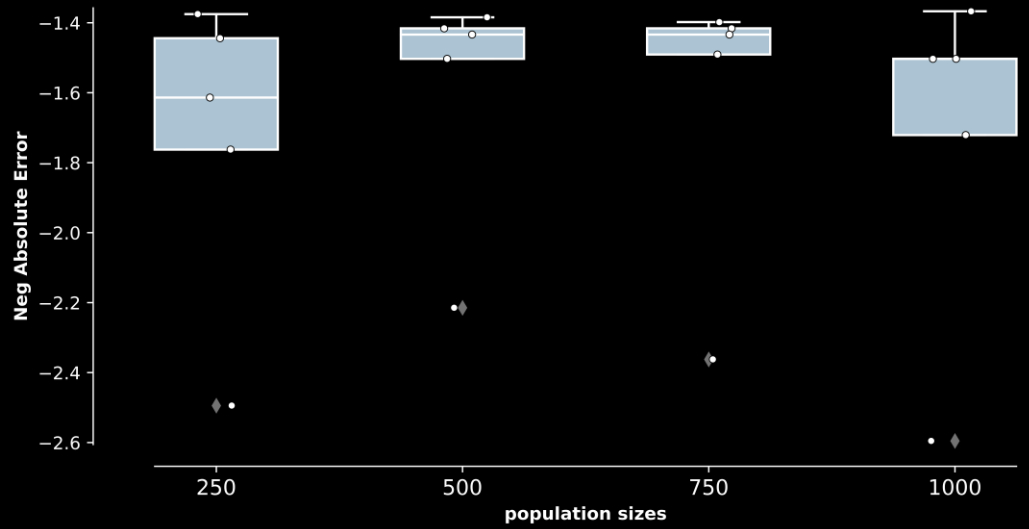


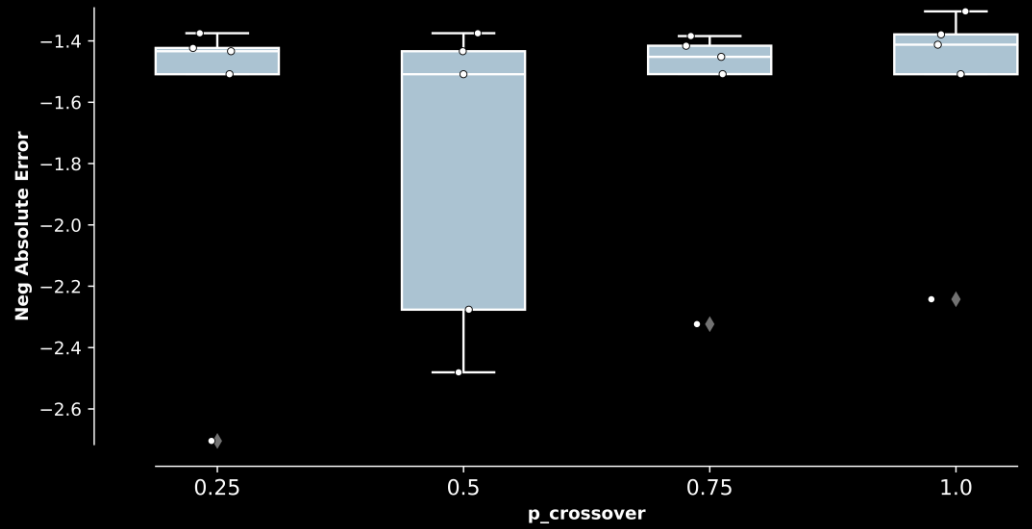
GP – Deeper into the parameters

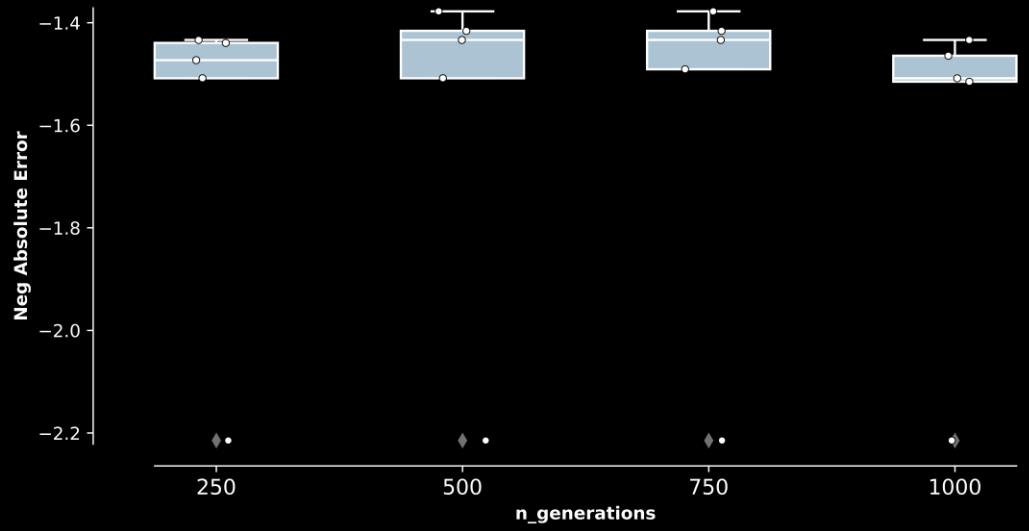
From the previous benchmarks
we now use:

- *Edda initialization*
- *Rank selector*
- *Baseline Crossover*
- *Hoist mutation*

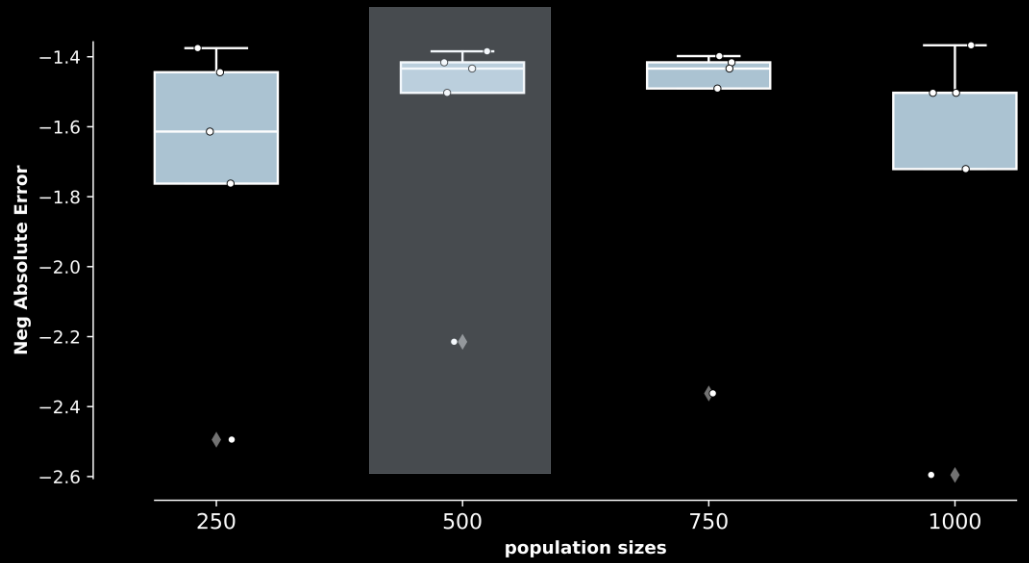
GP - Deeper into the parameters

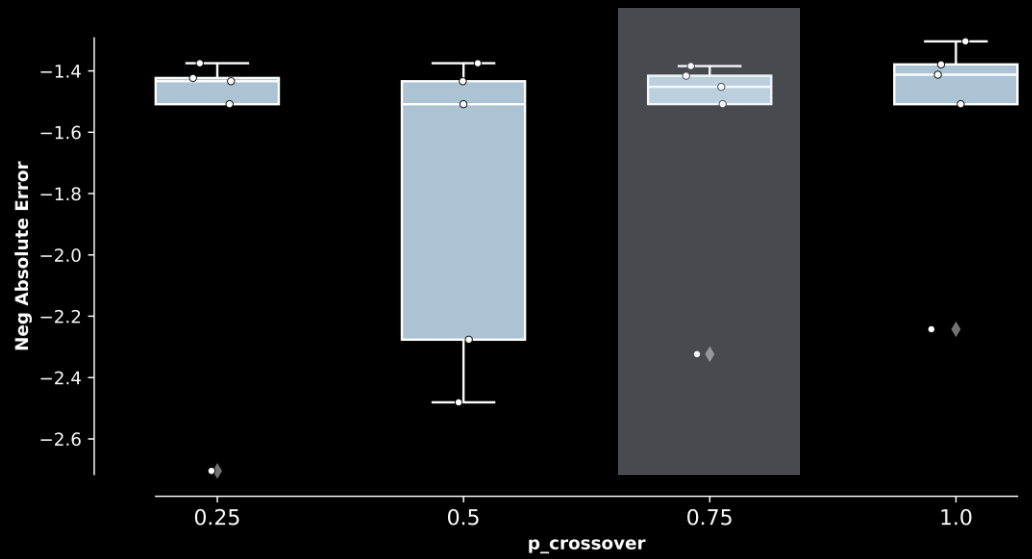


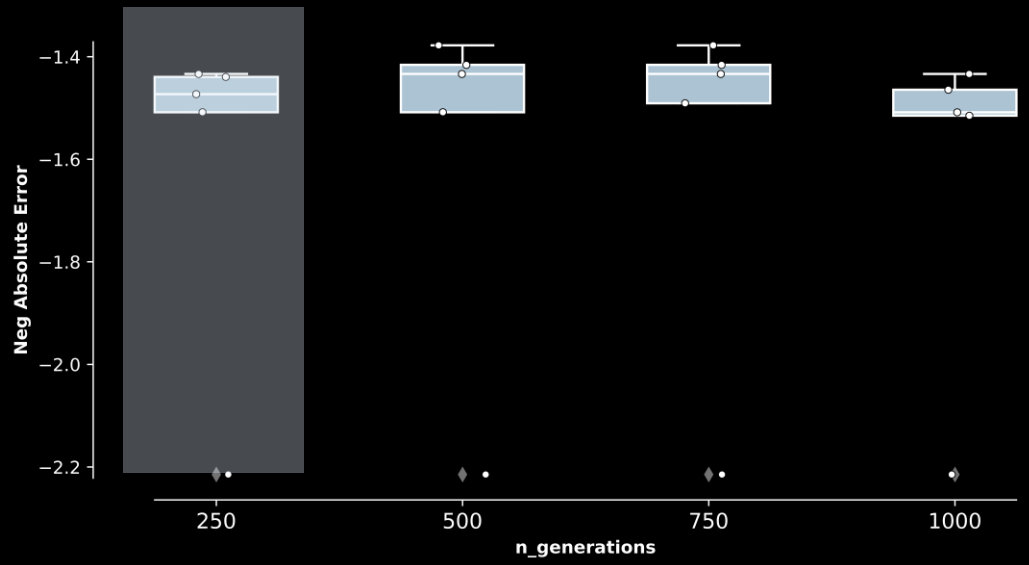




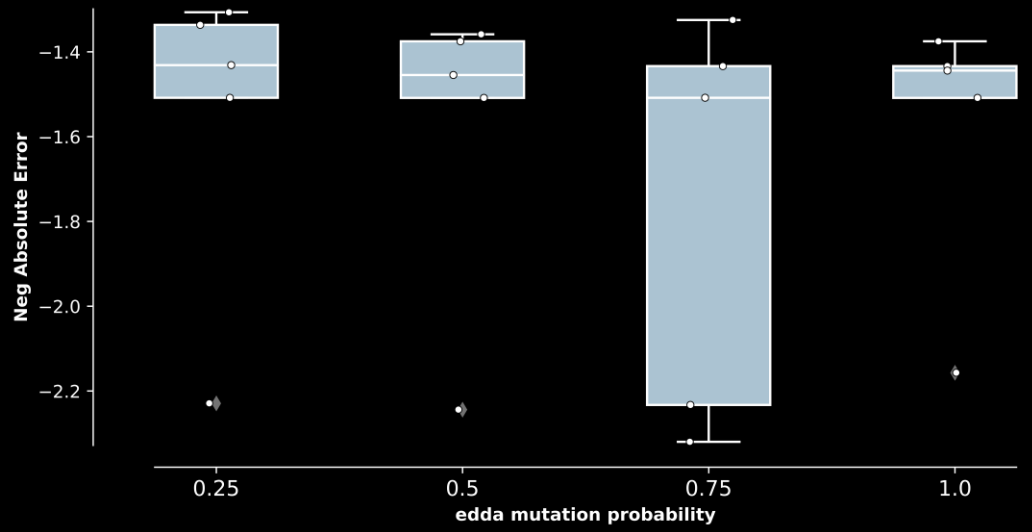
GP – Deeper into the parameters

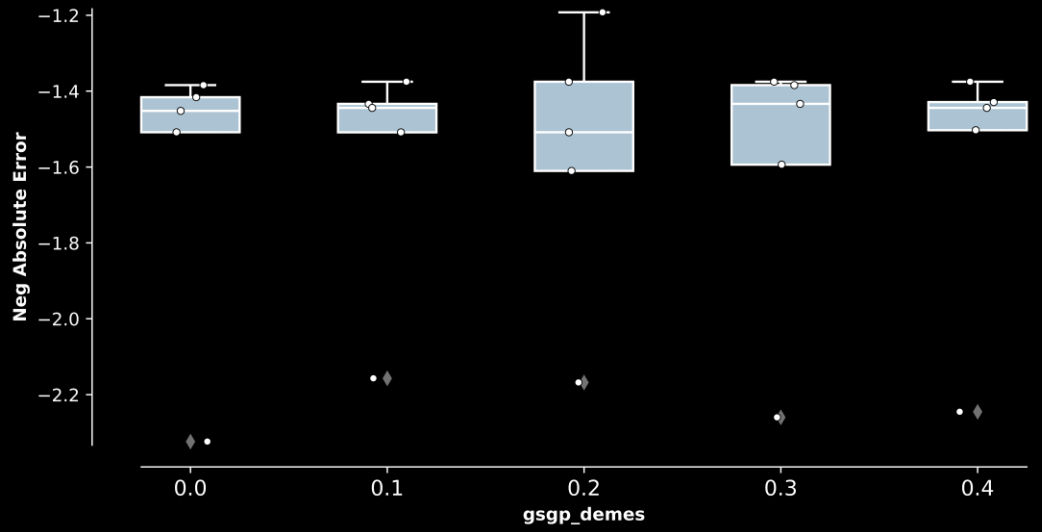


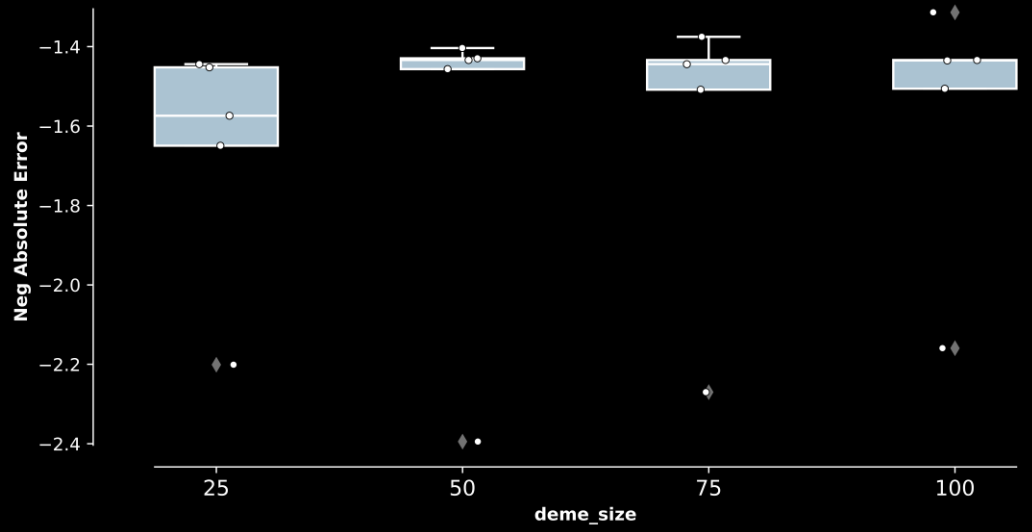


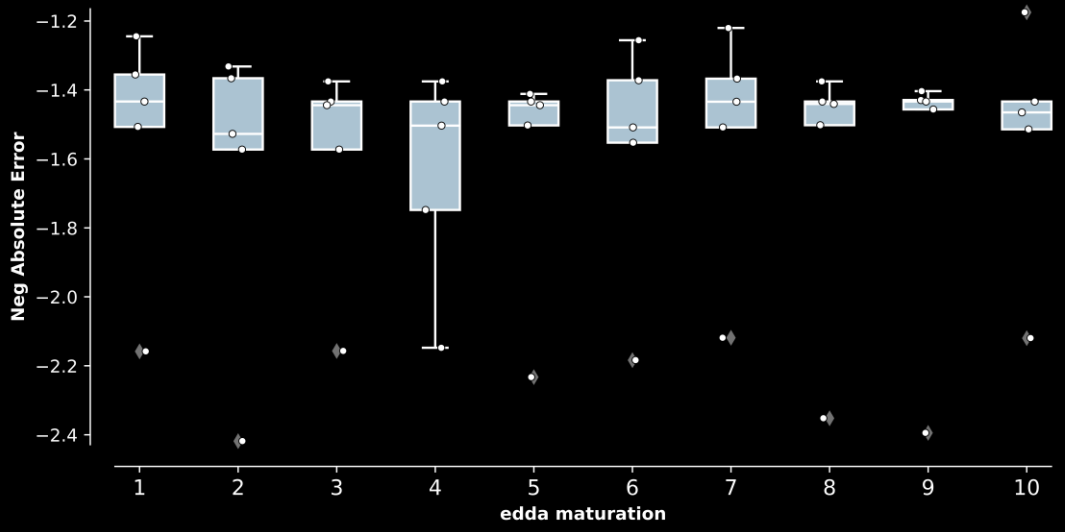


GP - Edda params

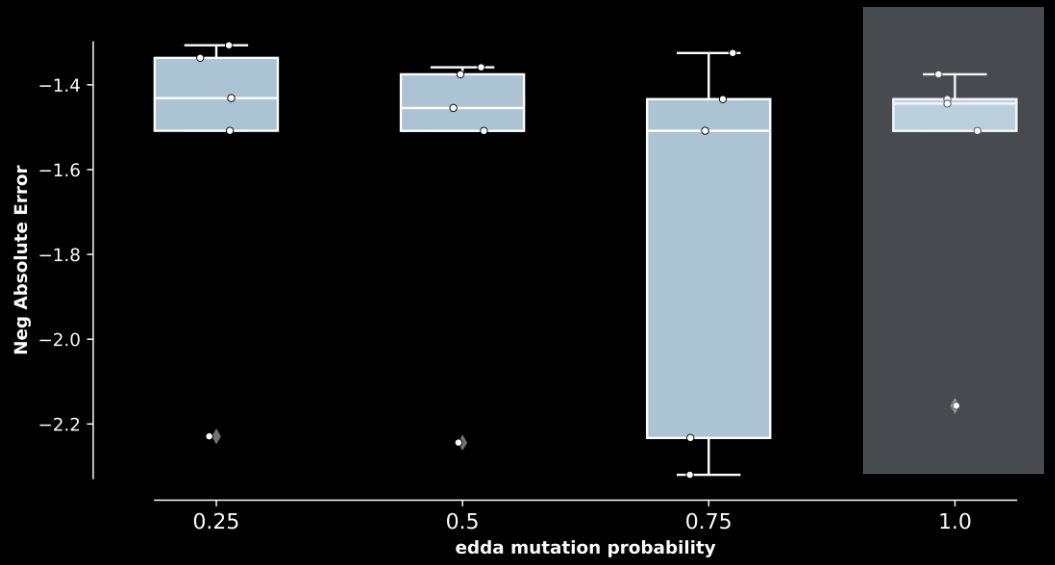


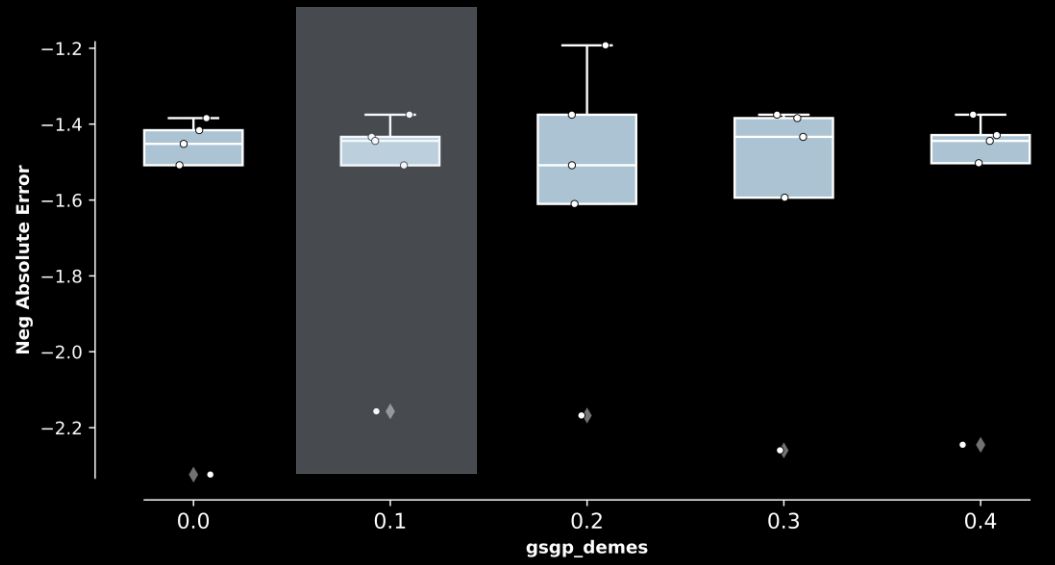


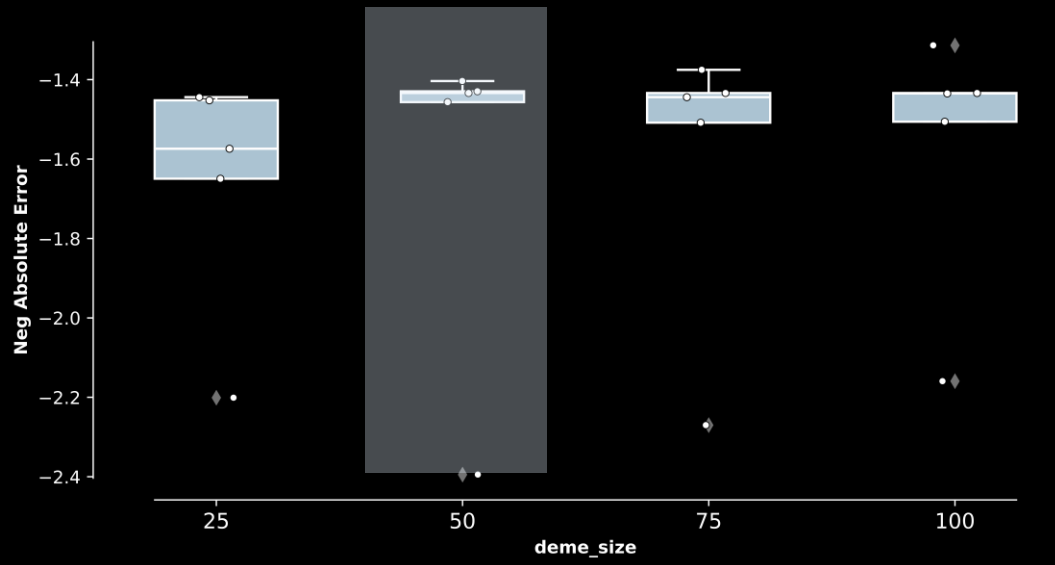


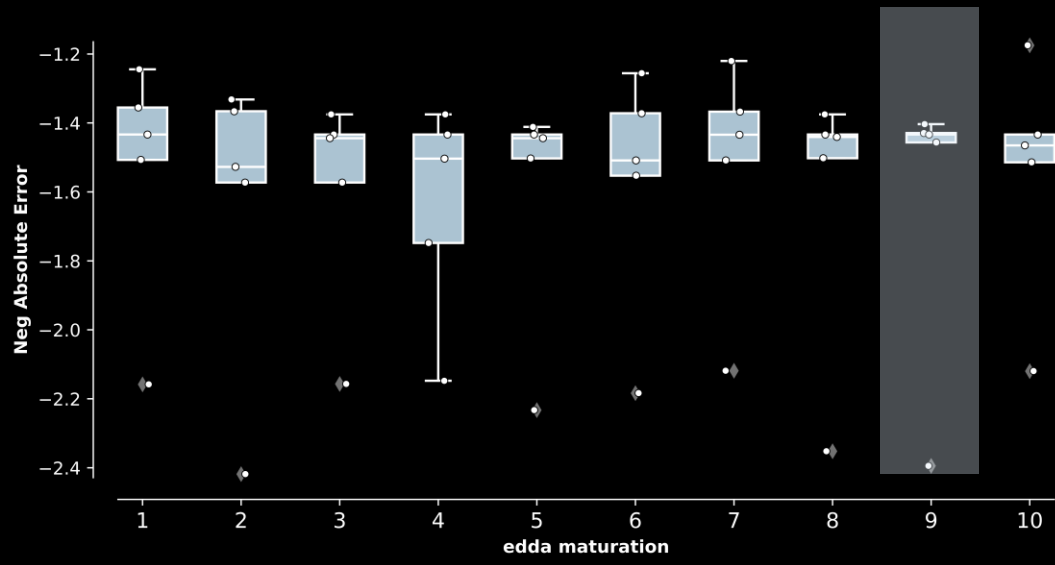


GP - Edda params



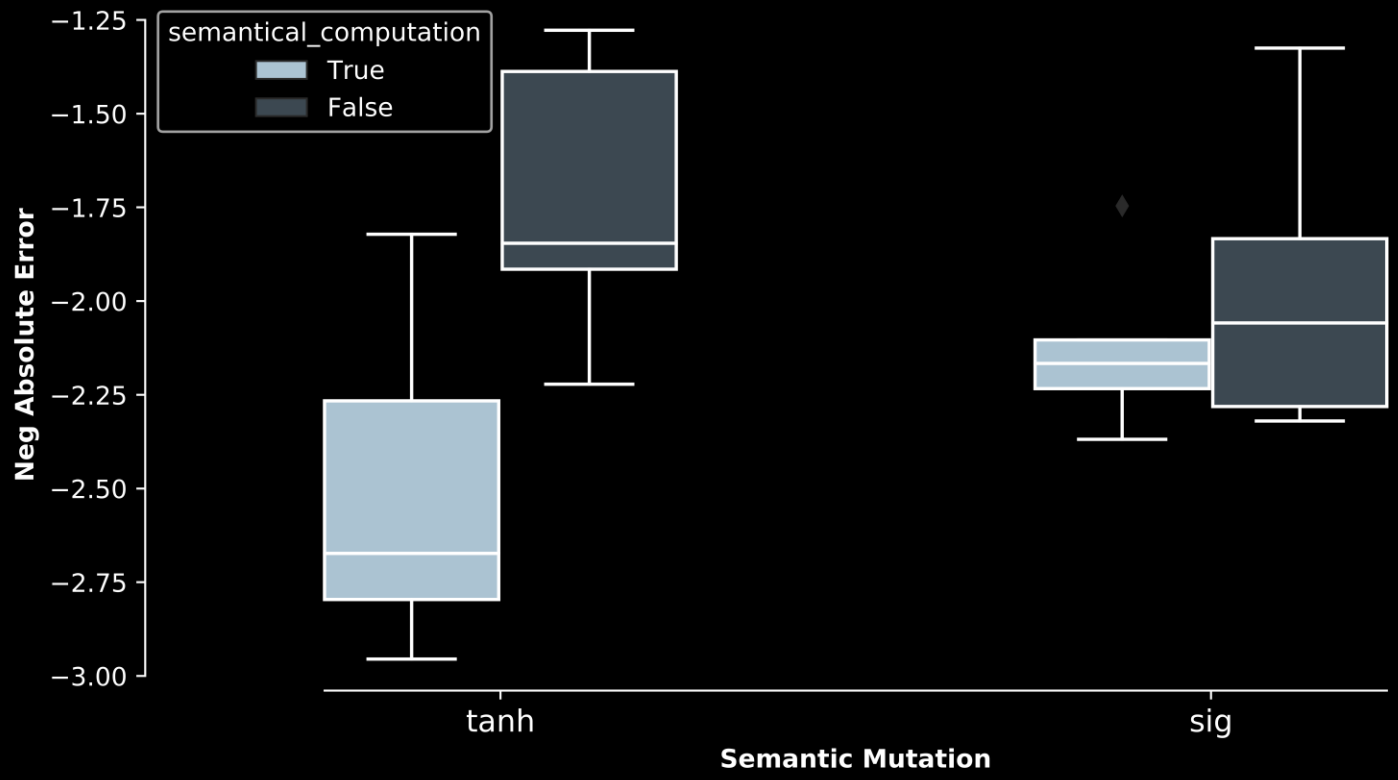






GP - Semantic Operatos

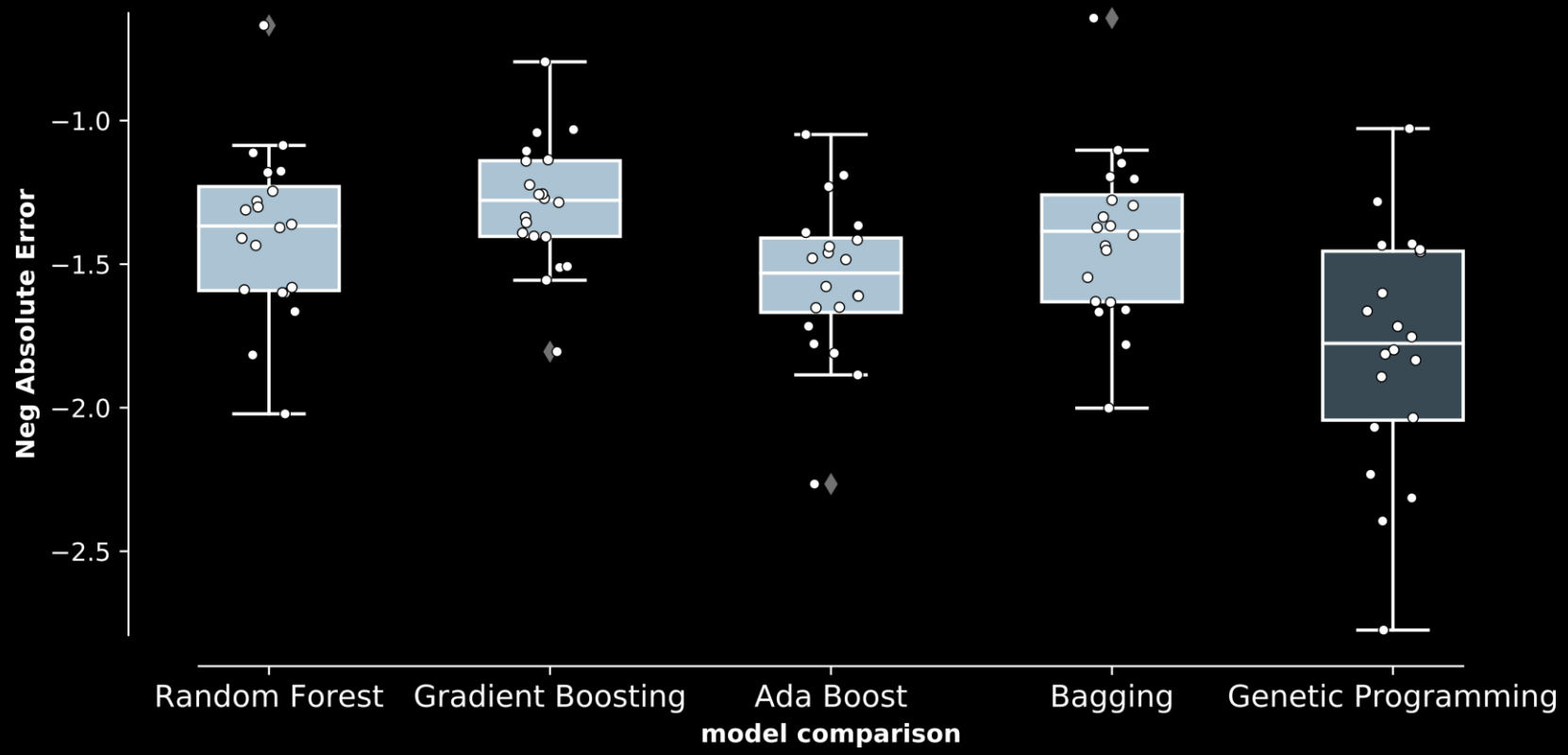
Due to some problems, generation and population sizes were set to 20



Comparing Ensembles to GP

Can genetic programming performance be worse than a simpler model?

Lets test it 20 times



Comparing Ensembles to GP

	Model	NMAE
Ensembles	Gradient Boosting	-1.291
	Random Forest	-0.391
	Bagging	-1.407
	Ada Boost	-1.553
GP	Genetic Programming	-1.799

Comparing Ensembles to GP

	Model	NMAE
Ensembles	**Gradient Boosting**	-1.291
	Random Forest	-0.391
	Bagging	-1.407
	Ada Boost	-1.553
GP	**Genetic Programming**	-1.799

t-test

t = 4.79
p-value = 0.003%

Means are statistically different for usual confidence levels of 1%, 5% and 10%

Conclusions

Hard Data collection

Problem required a long extraction and transformation process in order to be usable

Conclusions

Hard Data collection

Problem required a long extraction and transformation process in order to be usable

Genetic Programming was challenging

Coming up with new GP operators was much more difficult than it was with Genetic Algorithms, since we were now dealing with trees
Lack of extensive online support on this subject made the process of implementing new things more extensive and time consuming

Conclusions

Hard Data collection

Problem required a long extraction and transformation process in order to be usable

Genetic Programming was challenging

Coming up with new GP operators was much more difficult than it was with Genetic Algorithms, since we were now dealing with trees
Lack of extensive online support on this subject made the process of implementing new things more extensive and time consuming

Unexpected Results

We really did not expect Gradient Boost to outperform a complex algorithm like Genetic Programming. Outcome may be a result of the data or the preprocessing choices made.

Thank you