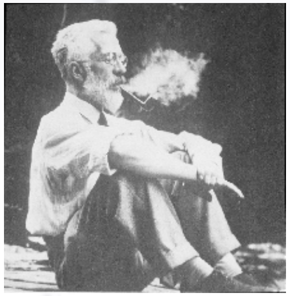
Applied ML

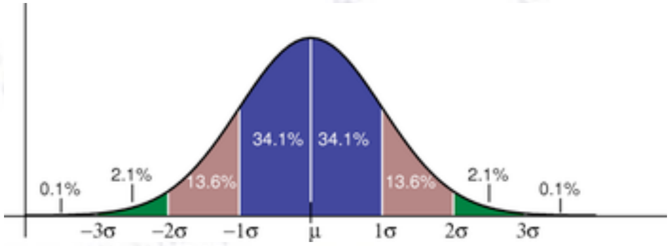
Results and Scores of Initial Project








Thomas Spieksma & Troels C. Petersen (NBI)



"Statistics is merely a quantisation of common sense - Machine Learning is a sharpening of it!"

The motivation

We wanted you to try the very **real challenge** of optimising models, without knowing their performance on the data it is applied to.

We also wanted you to **individually** run ML algorithms, so that you have the machinery in place after the course.

We insisted that you tried **both tree- and NN-based algorithms**, to get a feel for their differences and similarities.

We also wanted you to feel the “insecurity” about not knowing if you had gotten everything out of the data.

The description file was meant to trigger you to **think about your models**, and what you tried. Also, considerations of size and performance are in place.

Finally, we wanted to **ensure** that you yourself tried all the work and things to consider, to put together ML models and apply them.

Overall comments

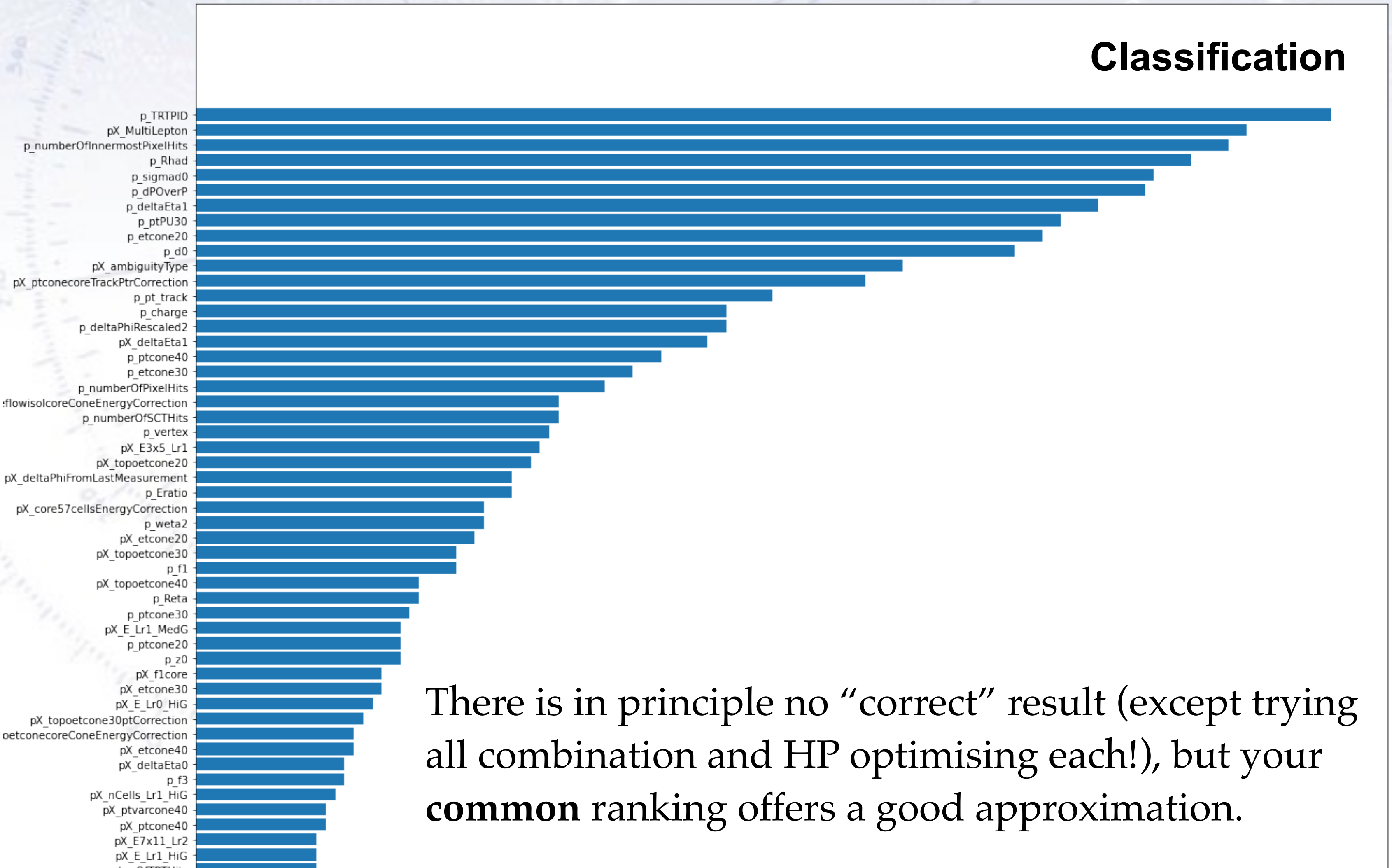
You generally did very well, and so let me start by gently stating, that you have nothing to fear - in fact, you did really great!

Grading it was perhaps comparable to the project itself, but we have done our best to be as open as possible about the scoring. And to give you a maximum of feedback, we have produced a report for each of you.

Classification Results

Classification variable usage

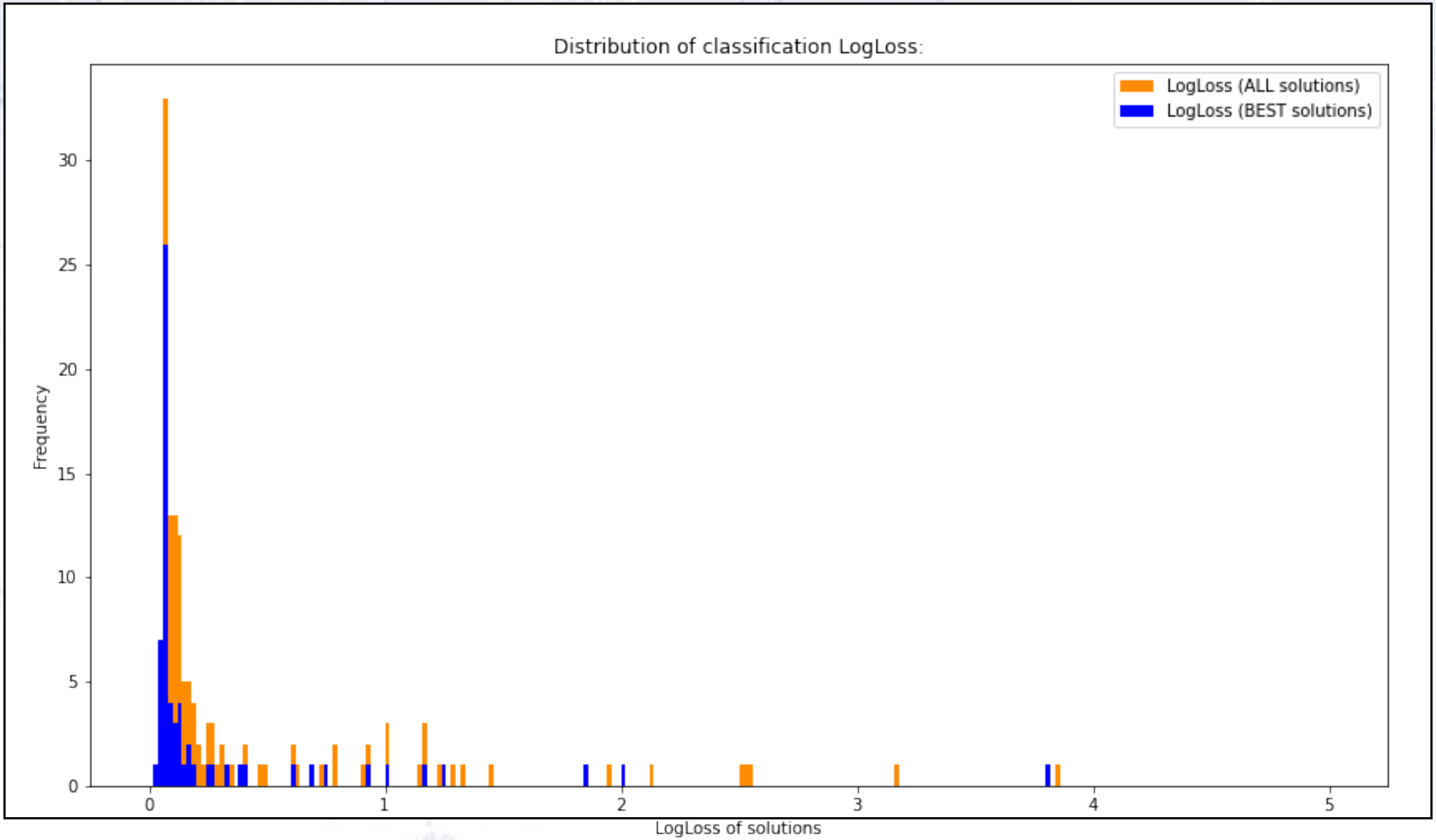
Many (most?) of you have made a good variable ranking. Below you find a variable usage **frequency** plot, showing how often a variable was used.



There is in principle no “correct” result (except trying all combination and HP optimising each!), but your **common** ranking offers a good approximation.

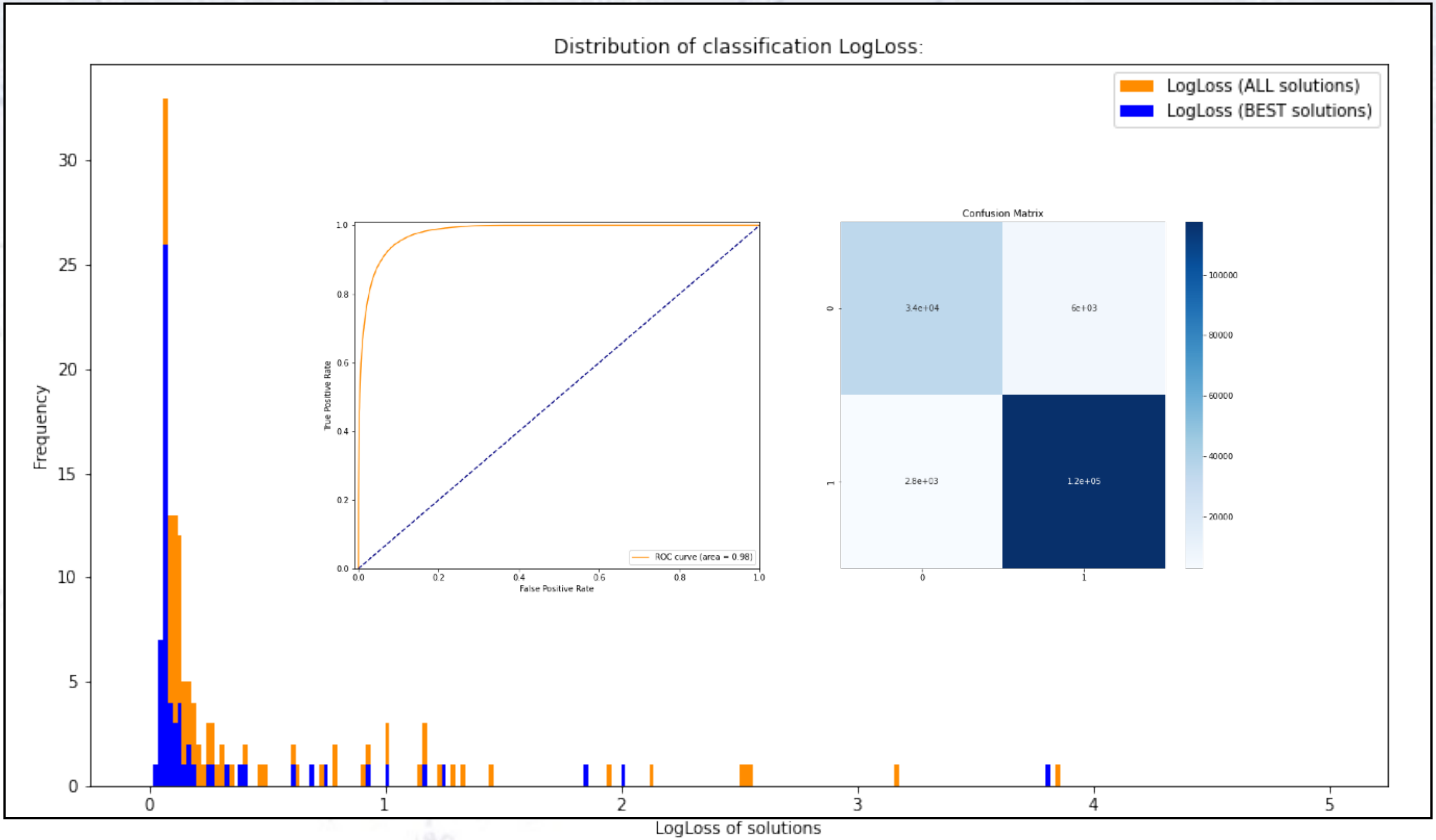
Classification score distribution

The distribution of the (Cross-Entropy) LogLoss values obtained was:



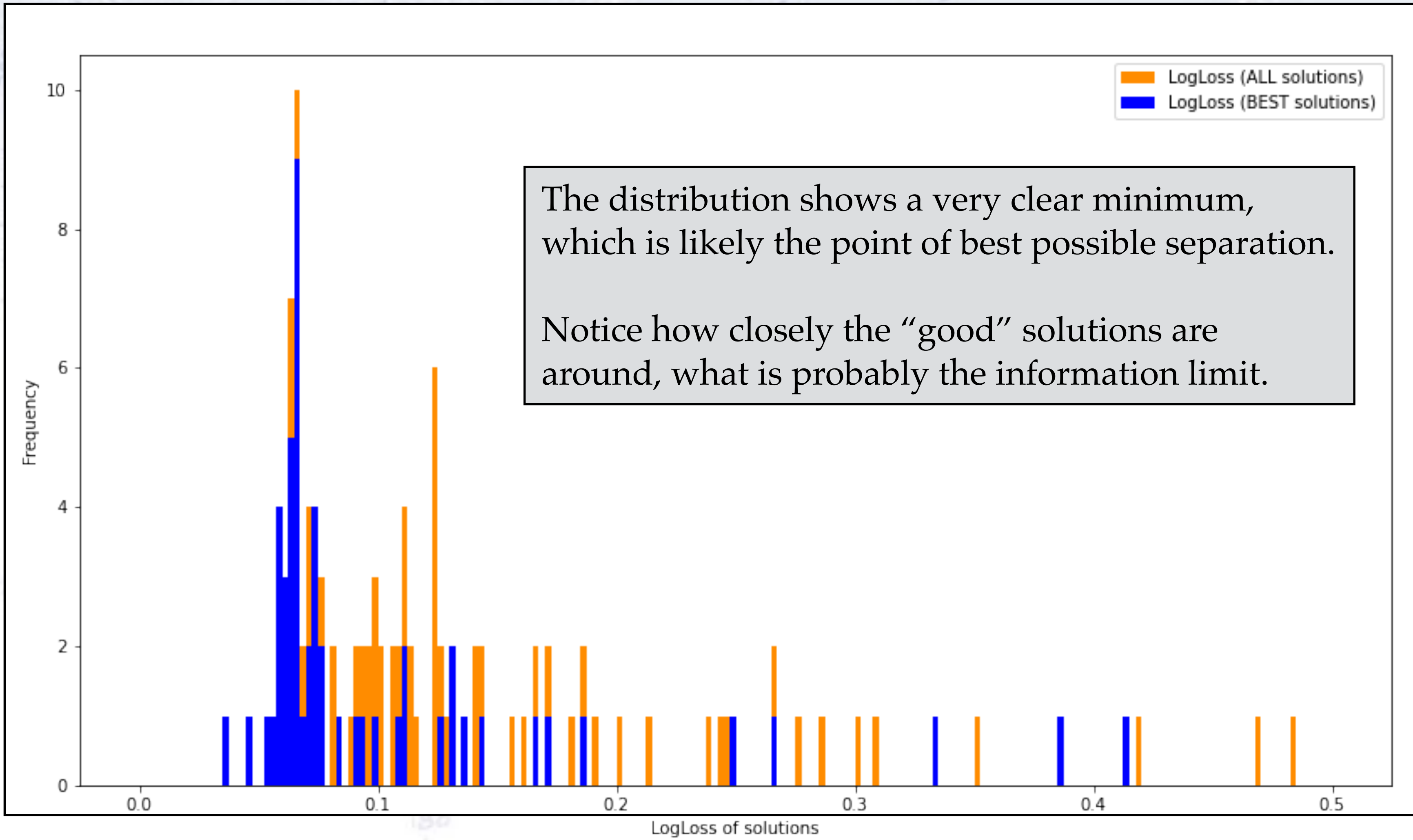
Classification score distribution

The distribution of the (Cross-Entropy) LogLoss values obtained was:



Classification score distribution

The distribution of the (Cross-Entropy) LogLoss values obtained was:

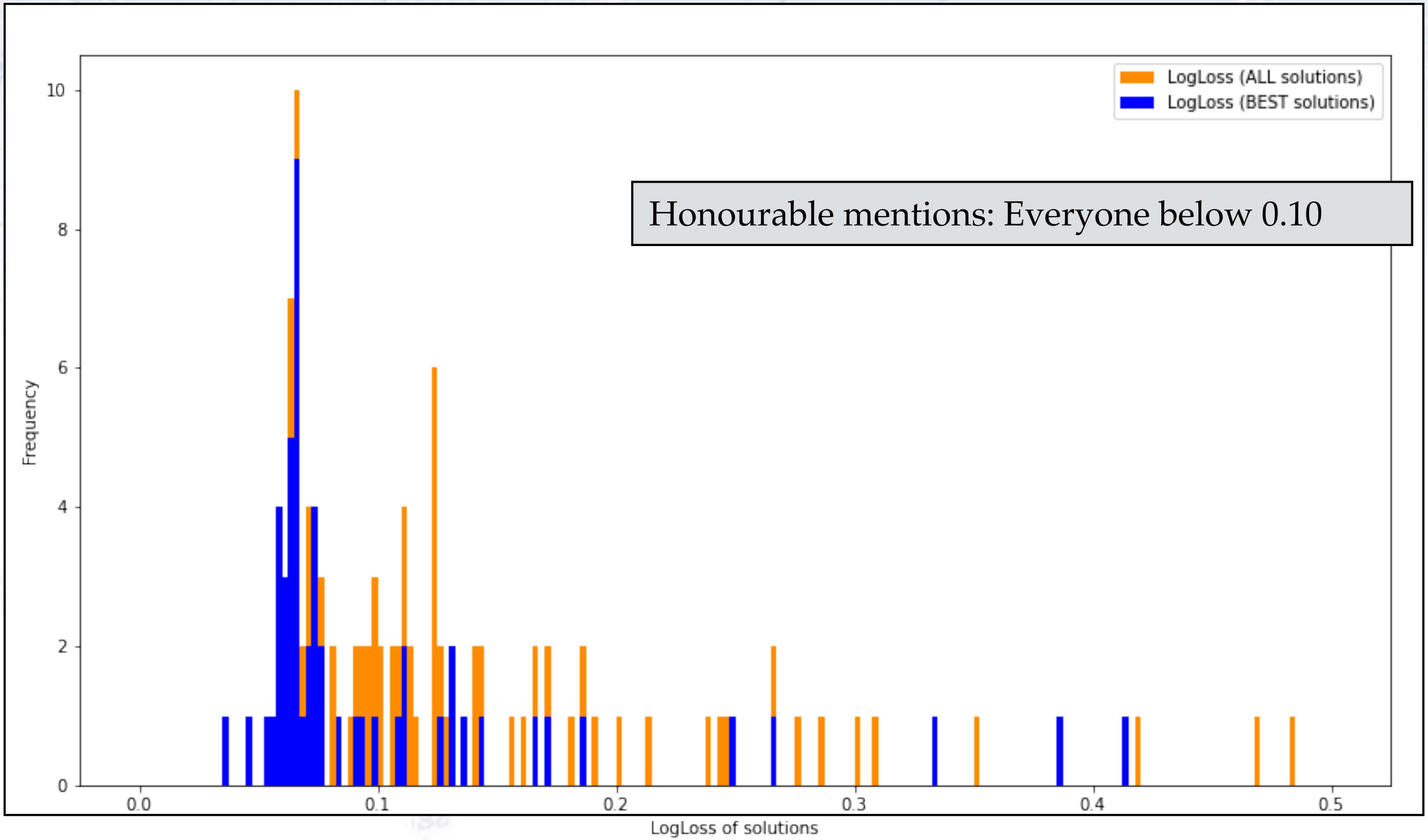


The distribution shows a very clear minimum, which is likely the point of best possible separation.

Notice how closely the "good" solutions are around, what is probably the information limit.

Classification score distribution

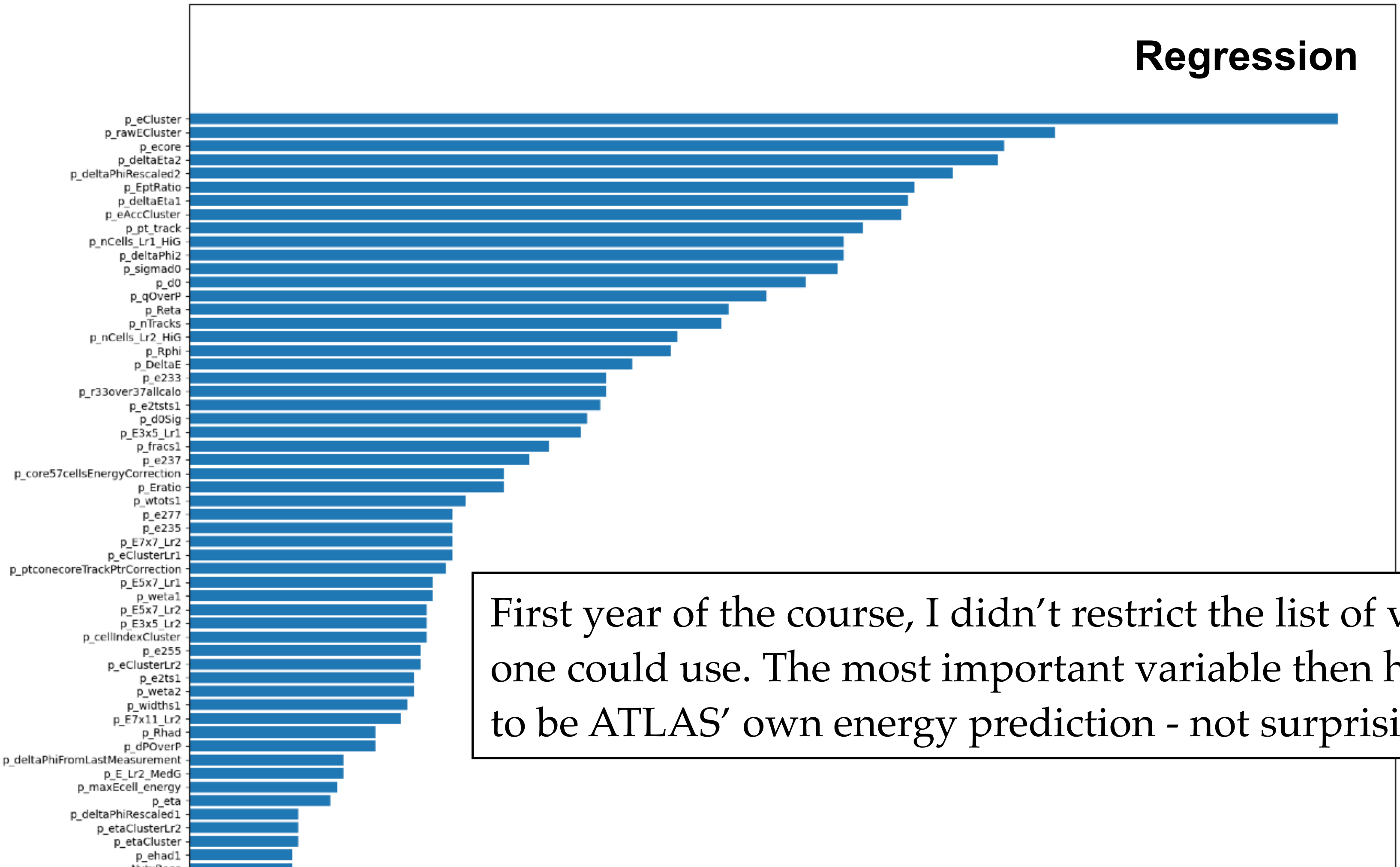
The distribution of the (Cross-Entropy) LogLoss values obtained was:



Regression Results

Regression variable usage

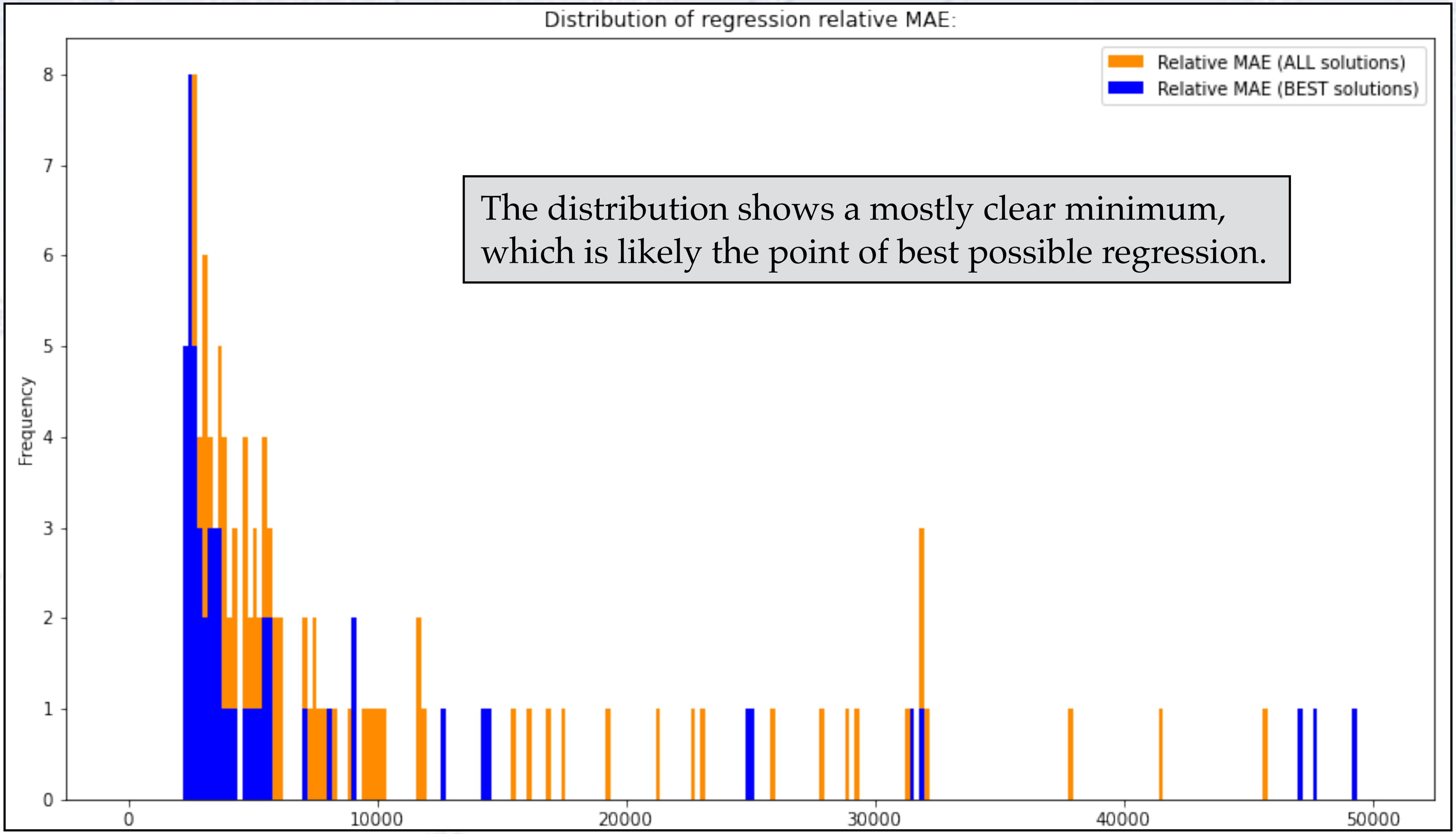
The variables have changed drastically from the classification case. There is NO overlap at all for the top 10-15 variables! Classification and Regression are in this case two very different tasks.



First year of the course, I didn't restrict the list of variables one could use. The most important variable then happens to be ATLAS' own energy prediction - not surprisingly!

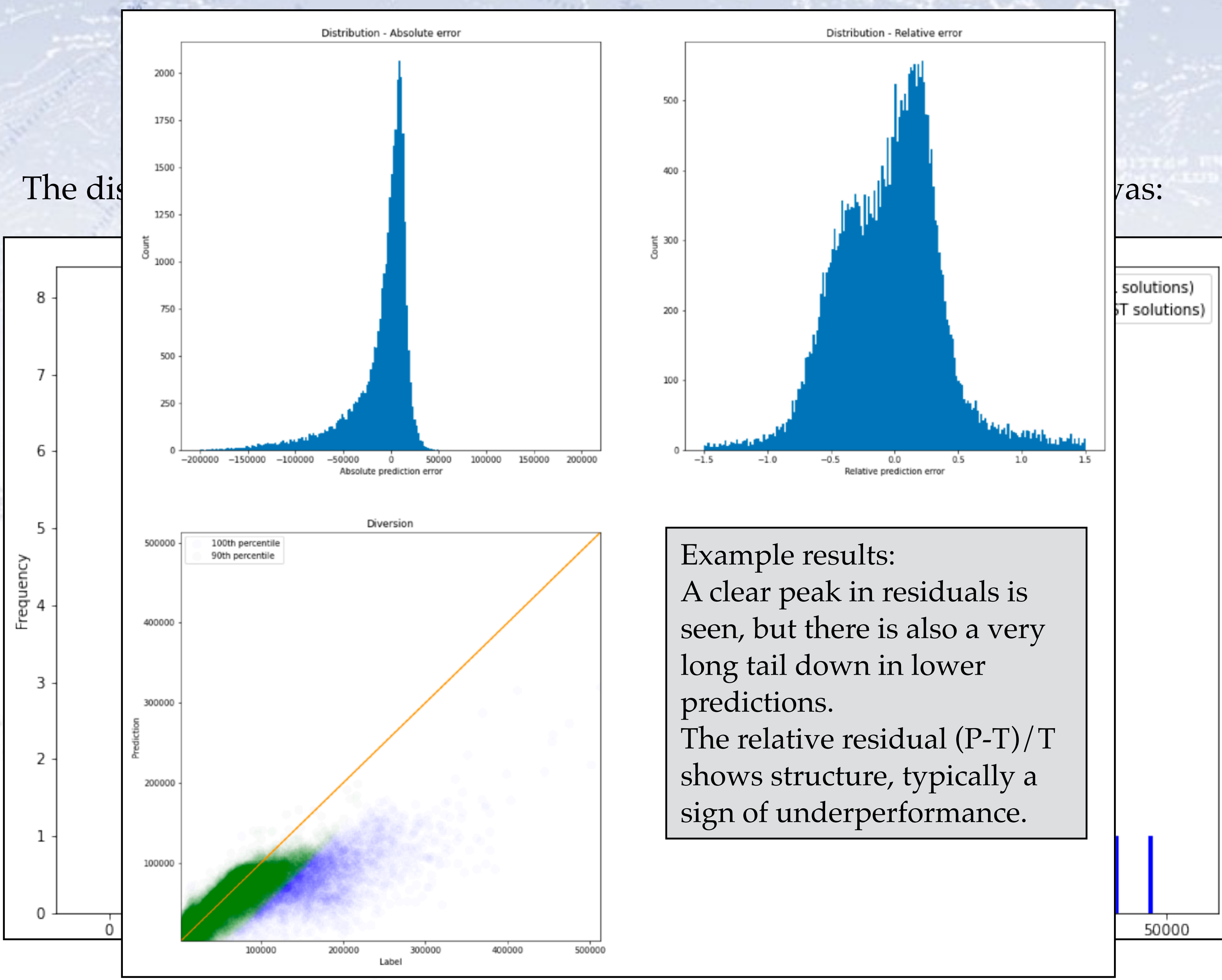
Regression score distribution

The distribution of the relative MAE (i.e. MAE((E-T)/T)) values obtained was:



The distribution shows a mostly clear minimum, which is likely the point of best possible regression.

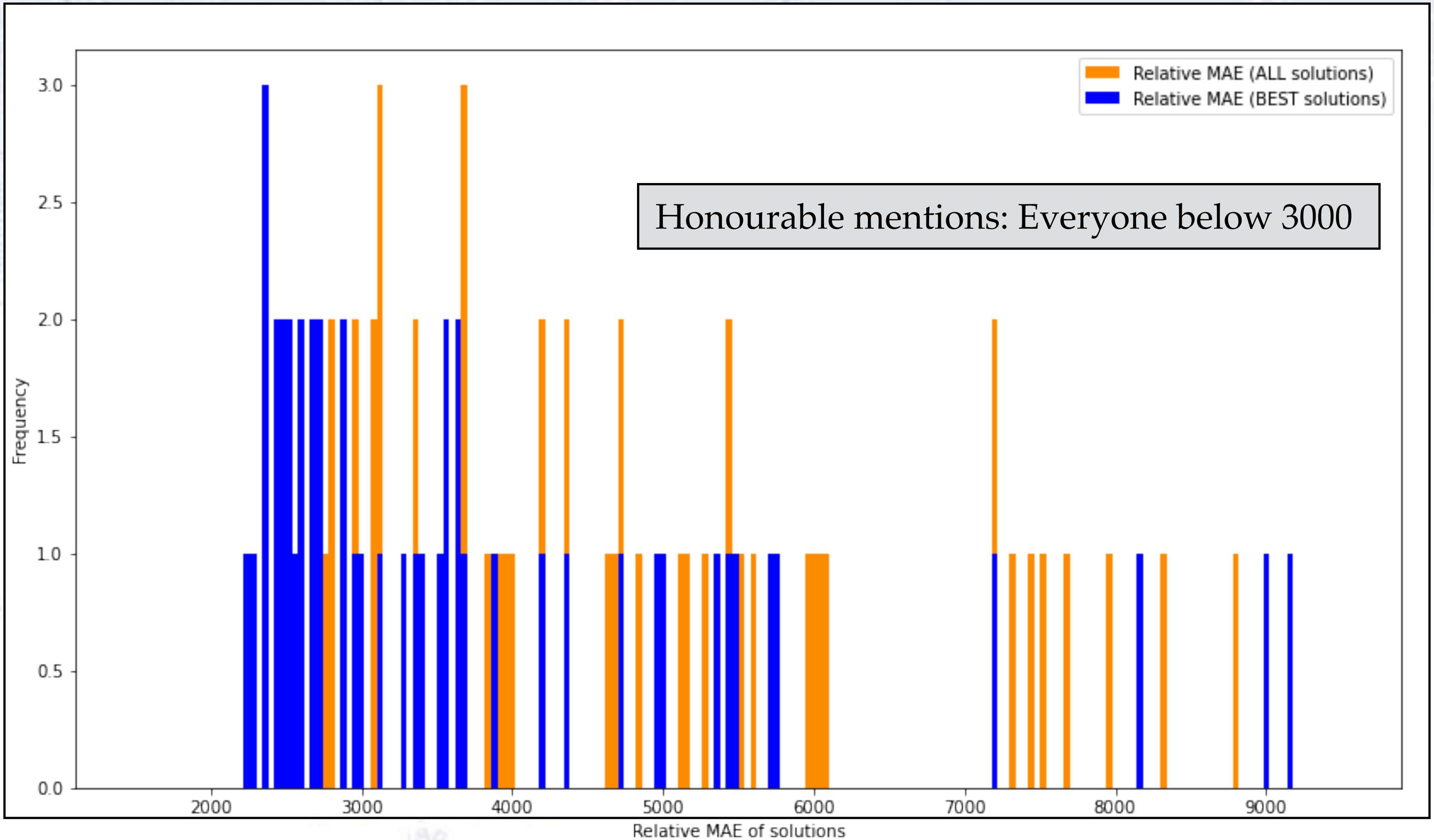
The dis… …vas:



Example results:
A clear peak in residuals is seen, but there is also a very long tail down in lower predictions.
The relative residual (P-T)/T shows structure, typically a sign of underperformance.

Regression score distribution

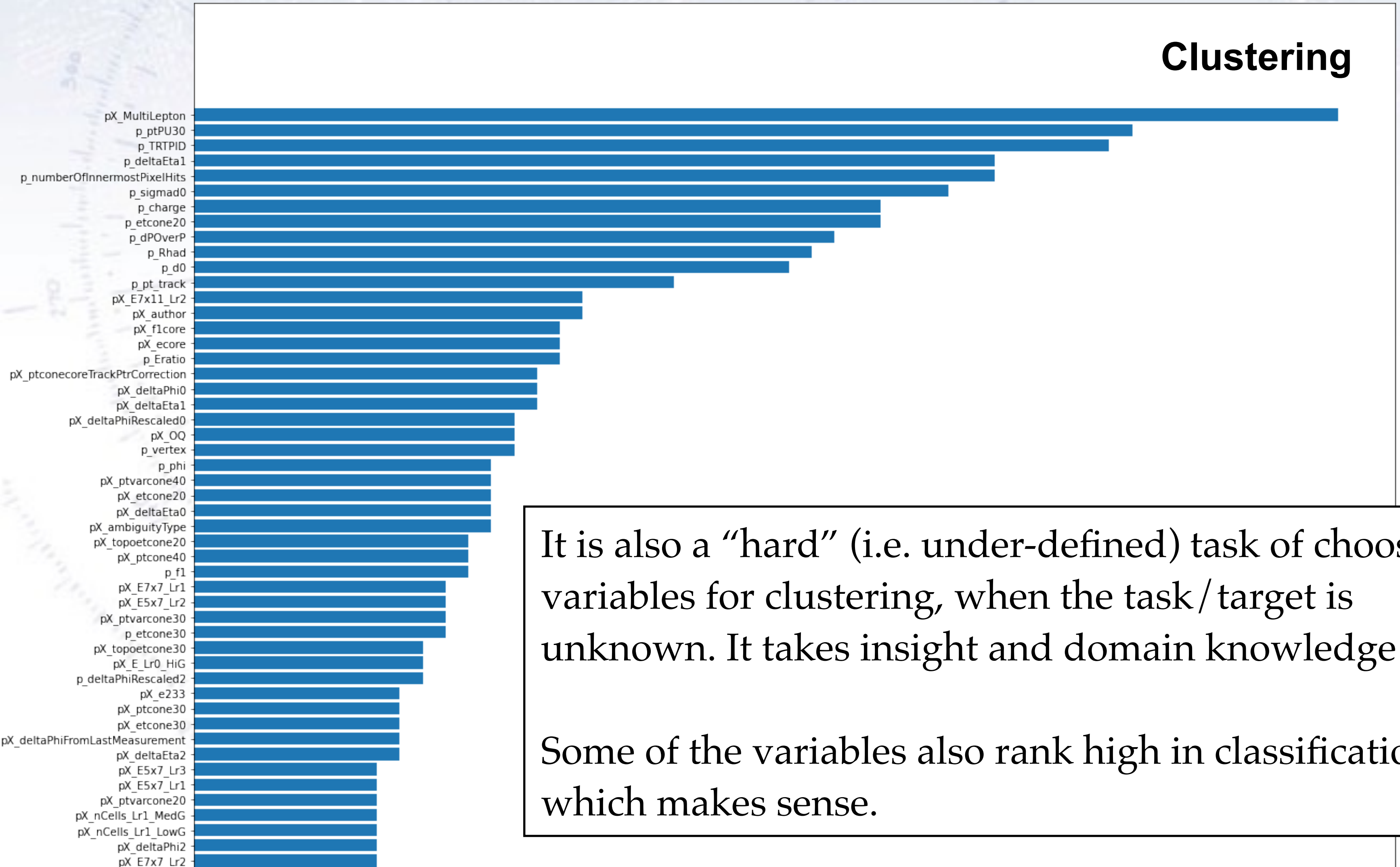
The distribution of the relative MAE (i.e. MAE((E-T)/T)) values obtained was:



Clustering Results

Clustering variable usage

I would have thought, that the clustering variable usage would be near-identical to that of the (supervised) classification task. However, it is not entirely…



It is also a "hard" (i.e. under-defined) task of choosing variables for clustering, when the task/target is unknown. It takes insight and domain knowledge…

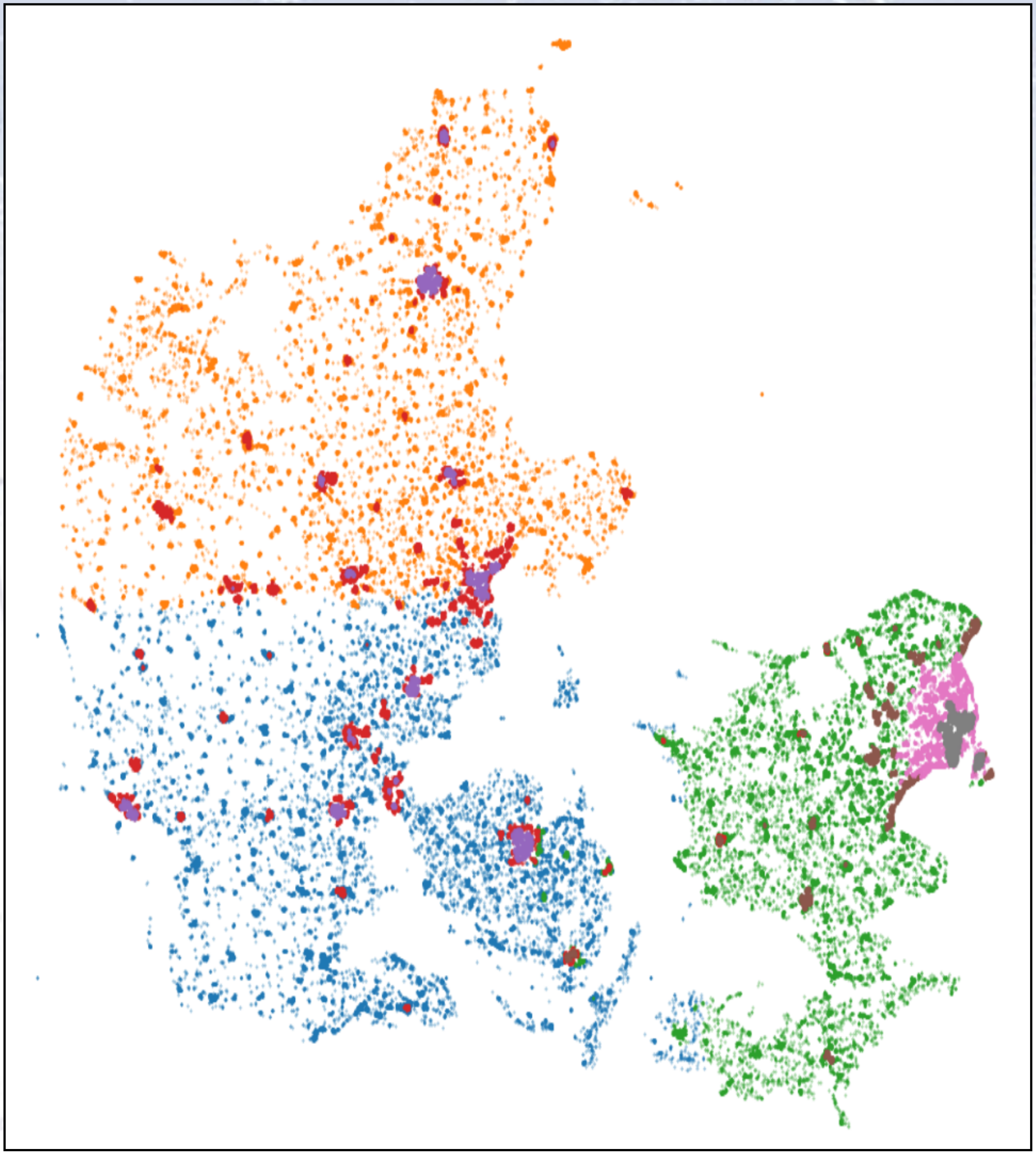
Some of the variables also rank high in classification, which makes sense.

Clustering housing

While postal codes are good, they are not very useful in clustering Denmark.

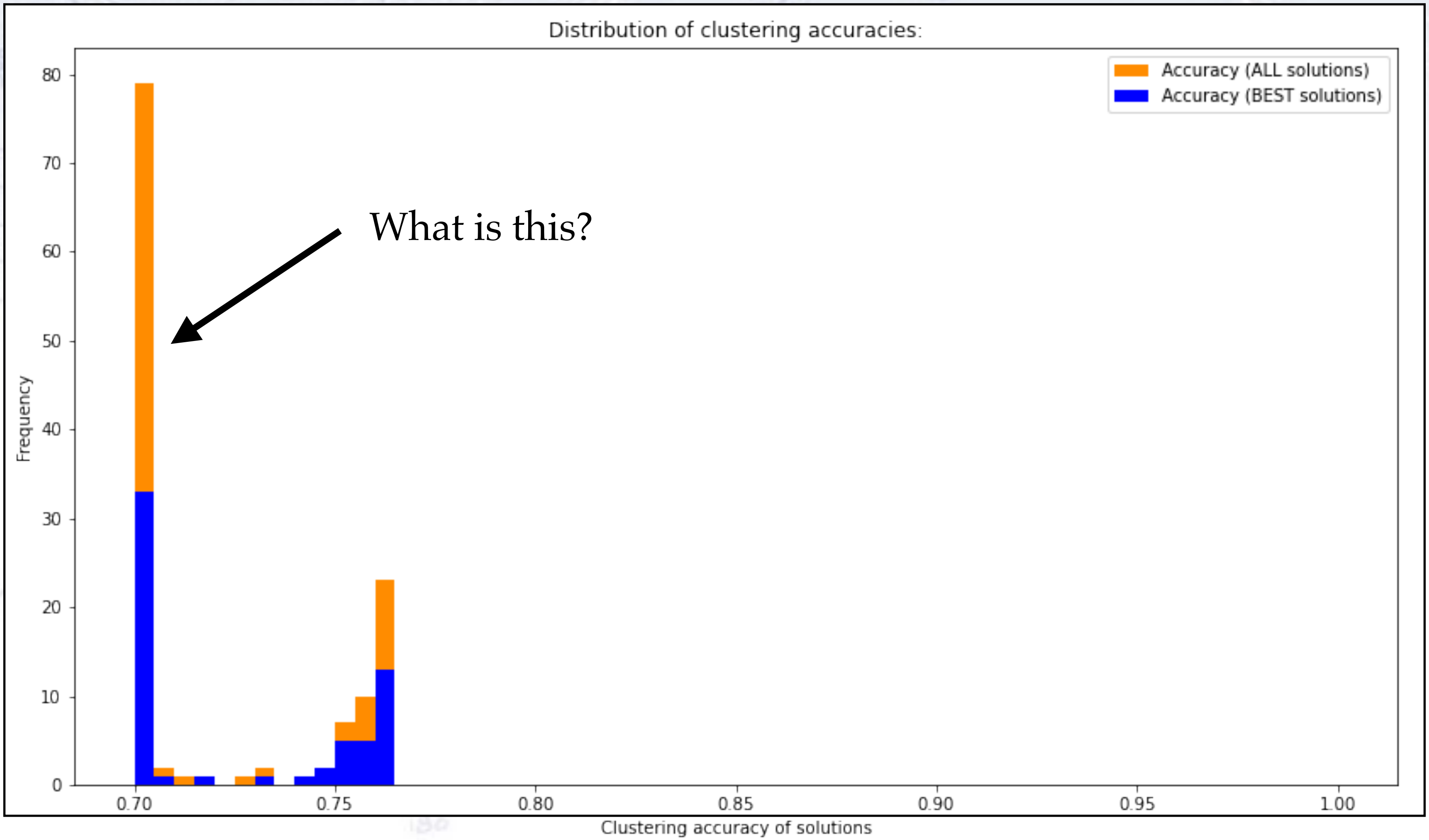
However, using just a few variables (x, y, density, price/m^2), one can cluster villas in Denmark very efficiently.

In this way, one can follow trends for a type of house much better.



Clustering accuracy distribution

The accuracy of the clustering (when assigned either electron or not) was:



Scoring your solutions

How do we grade your projects?

Best scoring classification: XGBoost with a cross entropy score of 0.0744

Best scoring regression: XGBoost with a relative MAD score of 8170.9

Best scoring clustering: KMeans with an accuracy score of 0.7547

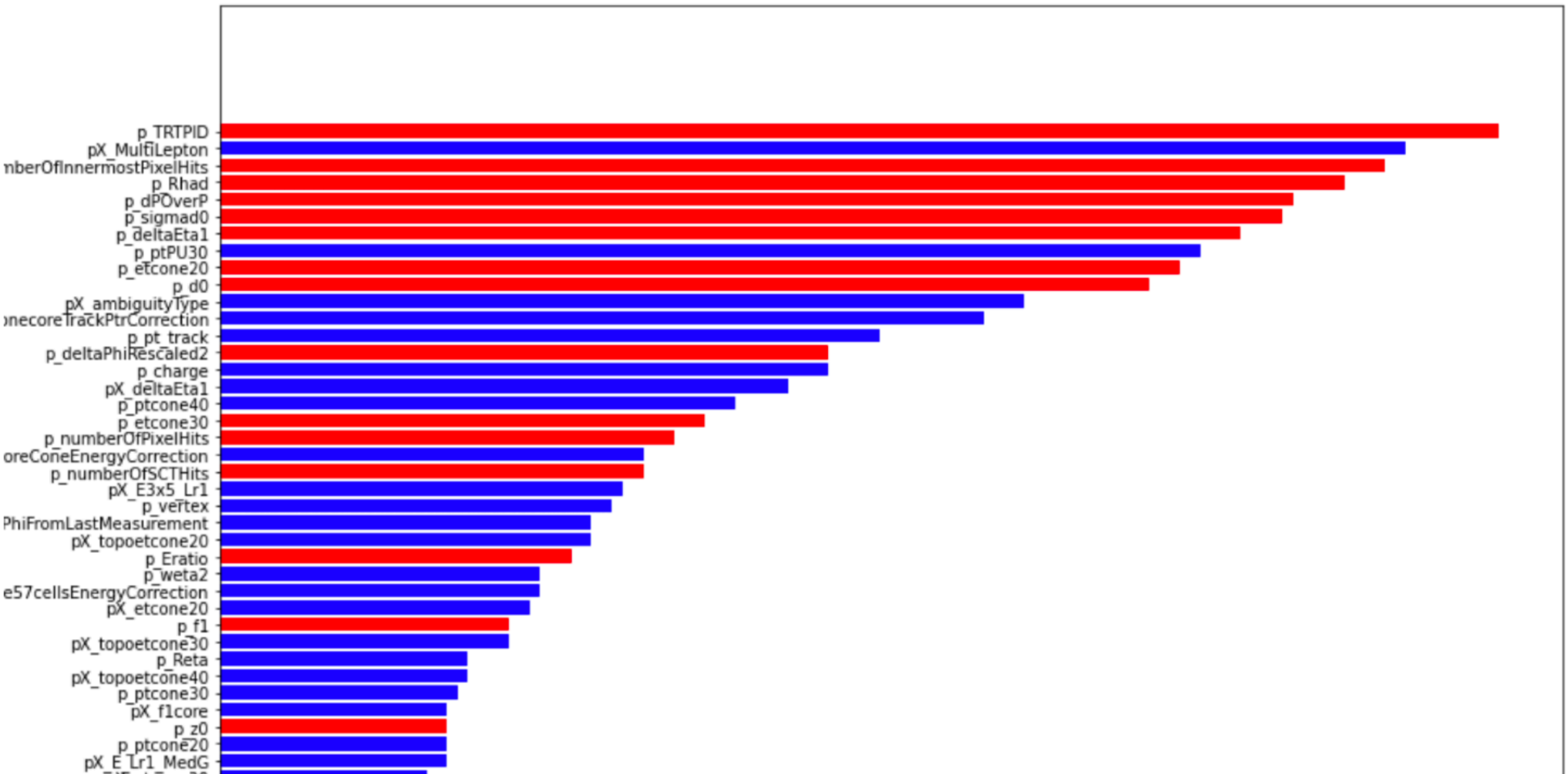
Final score

You submitted a full solution, from which you get:	67.0 points
Your choice of methods based on your description was scored as follows [0, 6]:	3.5 points
Your solution entailed 3 different algorithms, which gives you a score of [0, 6]:	4.0 points
Your best performance for classification gave: $max(0, (-log(CrossEntropy - 0.01)) \times 2.2)$:	6.0 points
Your variable choice for classification was scored $4 \times (\sum VarFreq(you)/VarFreq(top))$:	2.2 points
Your classification had 0 penalties, totaling to:	0.0 points
Your best performance for regression gave: $max(0, (-log(MAD(\frac{E-T}{T}) - 3000) - 0.5) \times 3.8)$:	0.0 points
Your variable choice for regression was scored $5 \times (\sum VarFreq(you)/VarFreq(top))$:	3.5 points
Your regression had 0 penalties, totaling to:	0.0 points
Your best performance for clustering gave: $max(0, (Accuracy - 0.7) \times 16)$:	0.9 points
Your variable choice for clustering was scored $(\sum VarFreq(you)/VarFreq(top))$:	0.6 points
Your clustering had 0 penalties, totaling to:	0.0 points
Thus your total number of points was:	**87 points**

Your variable choice

Assuming, that the variable frequency reflected the actual ranking very well, your variable choice was scored as follows (factors were 4, 5, and 1):

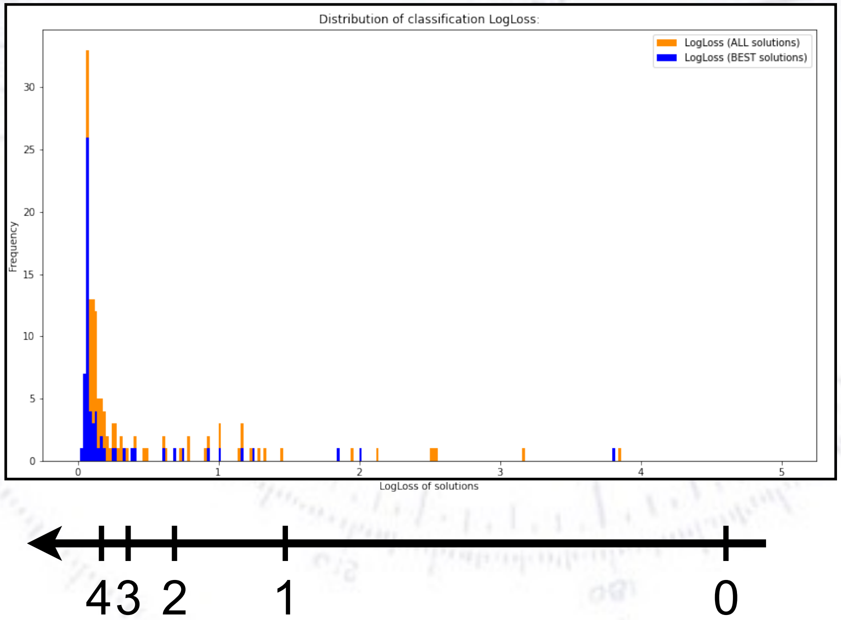
$$8 \times \left(\sum Freq(\text{Your variables}) / \sum Freq(\text{Top variables}) \right)$$

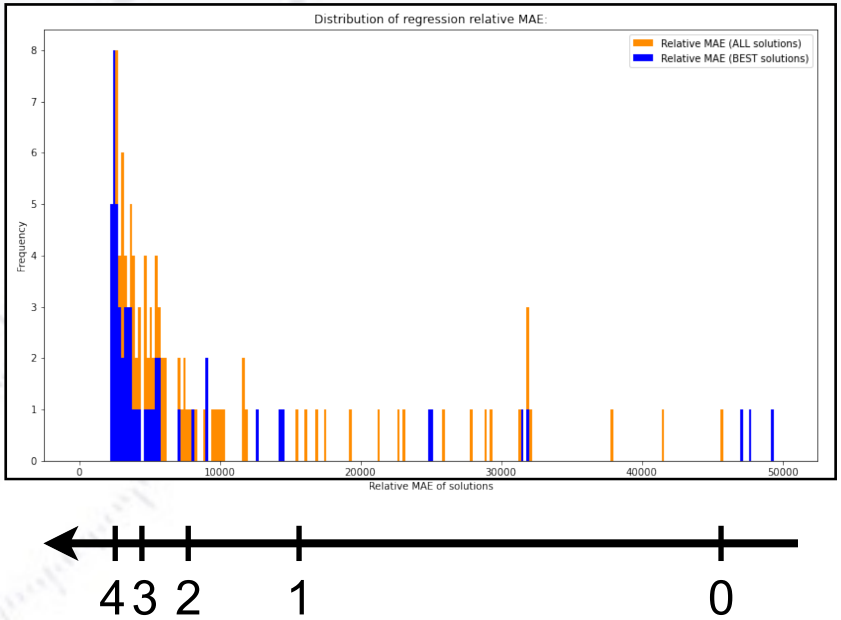


Performance scoring

As mentioned, performance isn't everything, and we certainly didn't want it to be for the small project. Getting close to the information limit is just great.

This was reflected by using a logarithmic scoring, which turned your best key performance parameter into a score in the (open) range [0,5+]:

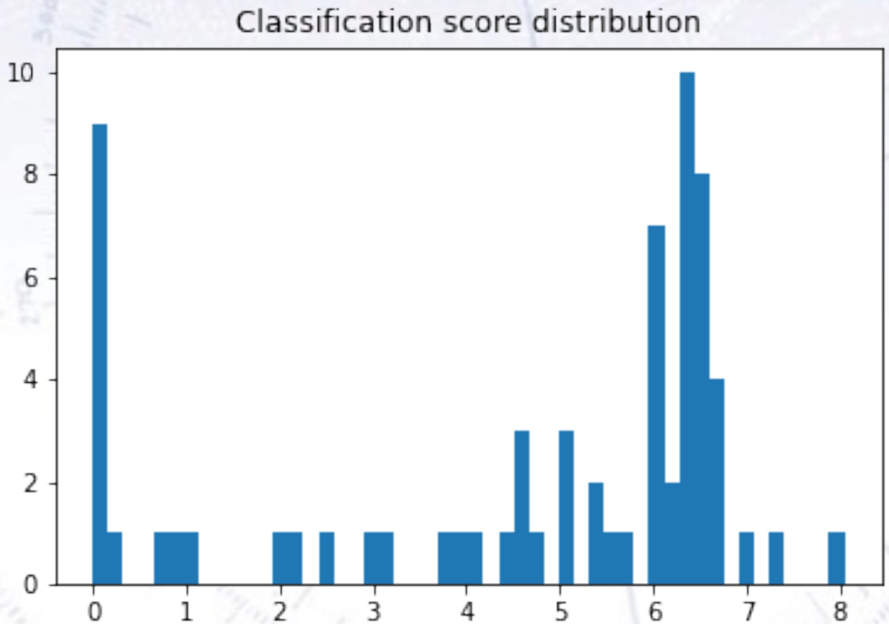


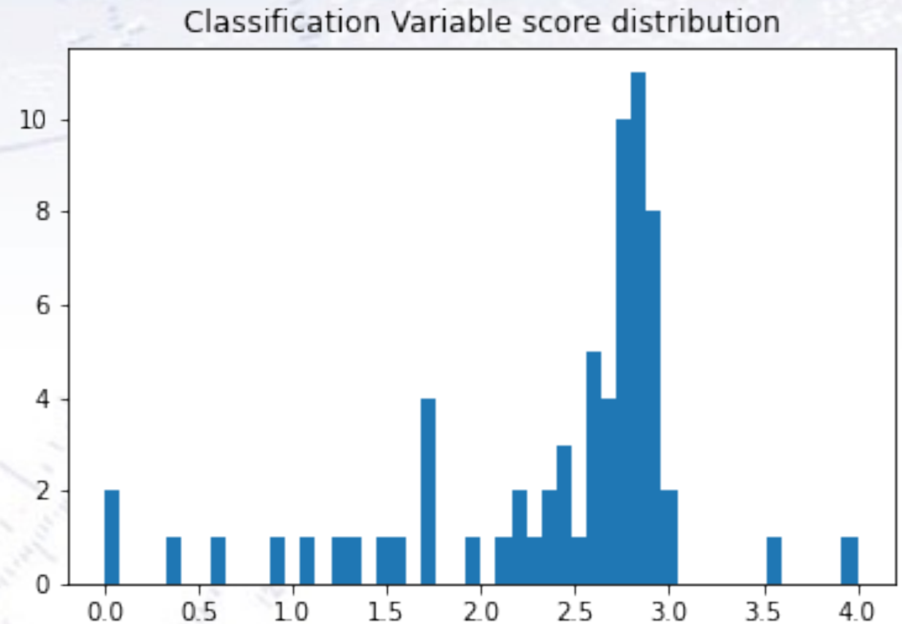


In all of this, you could of course not get negative points for an accepted solution!

The resulting score distributions

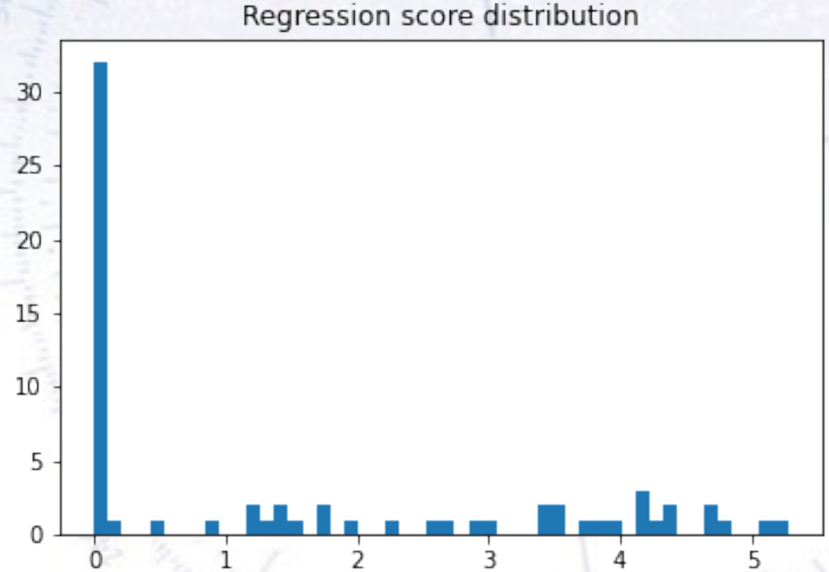
Score distributions for **classification** performance and variable choice:

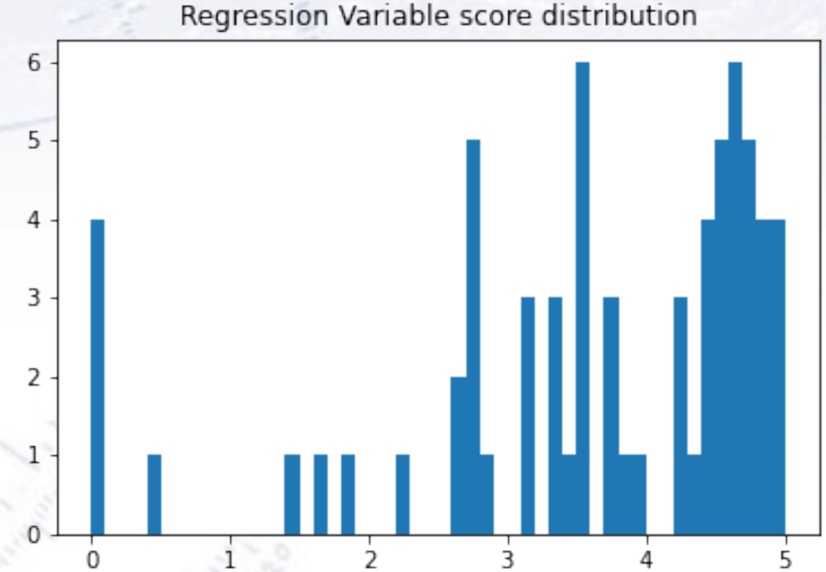




The resulting score distributions

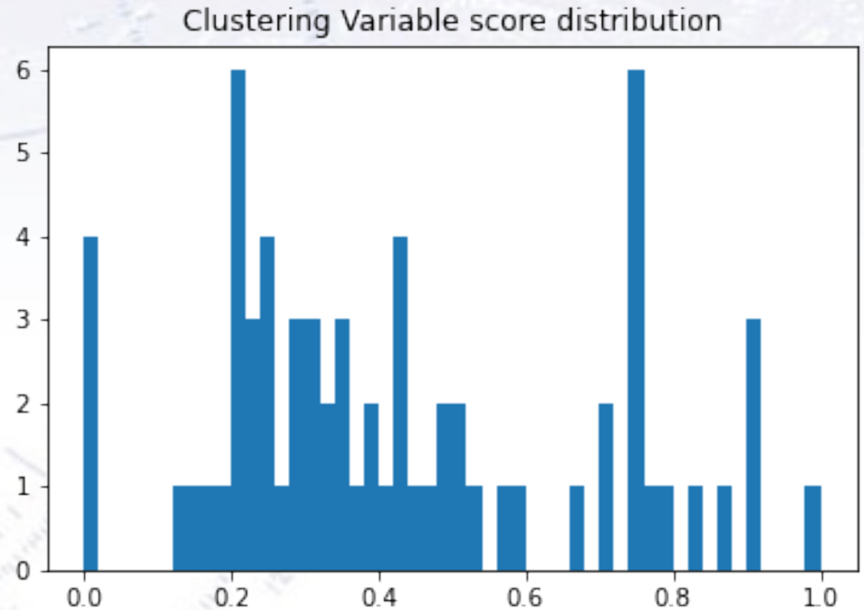
Score distributions for **regression** performance and variable choice:





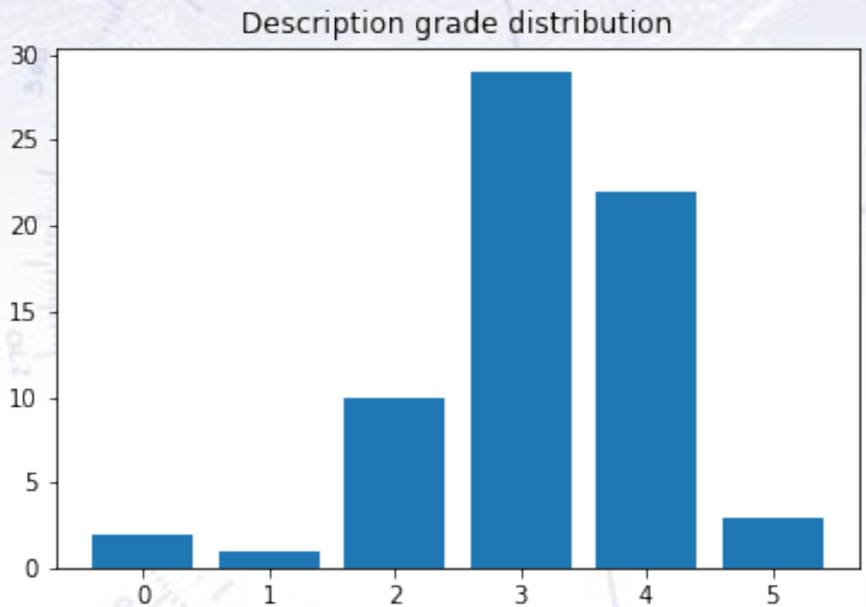
The resulting score distributions

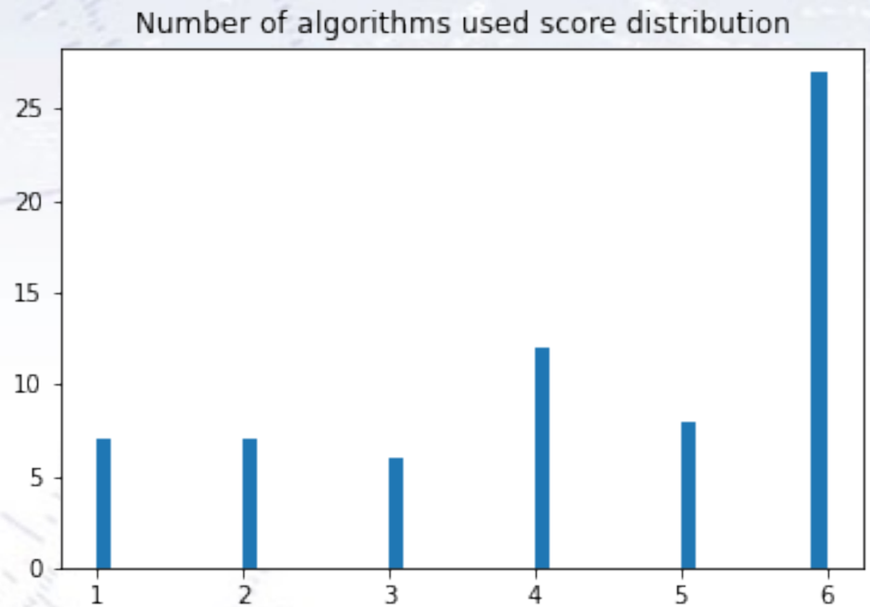
Score distributions for **clustering** performance and variable choice:



The resulting score distributions

The scores for descriptions and number of different algorithms (that work!) are:





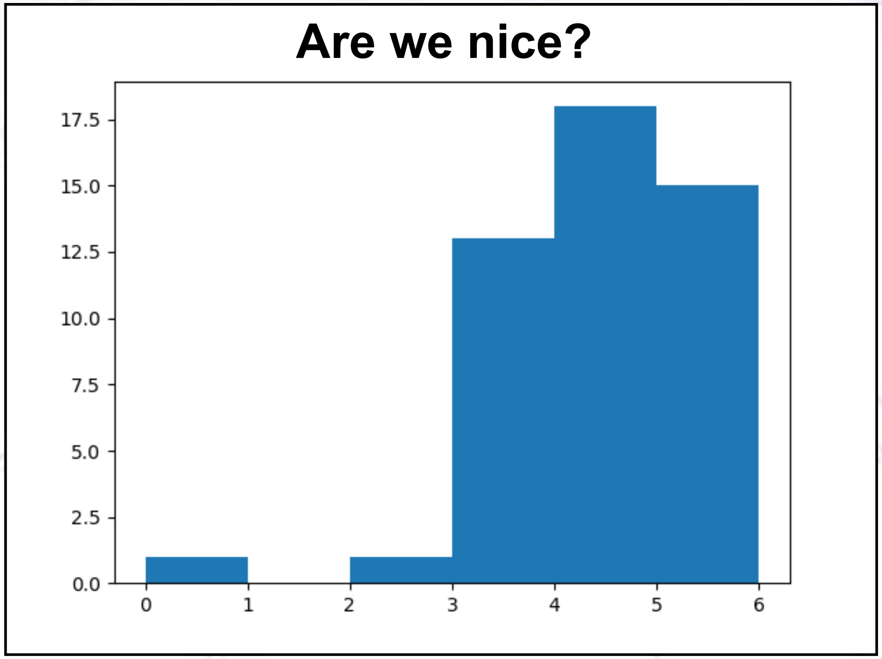
I read several of the “lower scoring” descriptions, but must say that I found them “reasonably acceptable”, so in general the level was high (but don’t do transformation of variables, when using a BDT!).

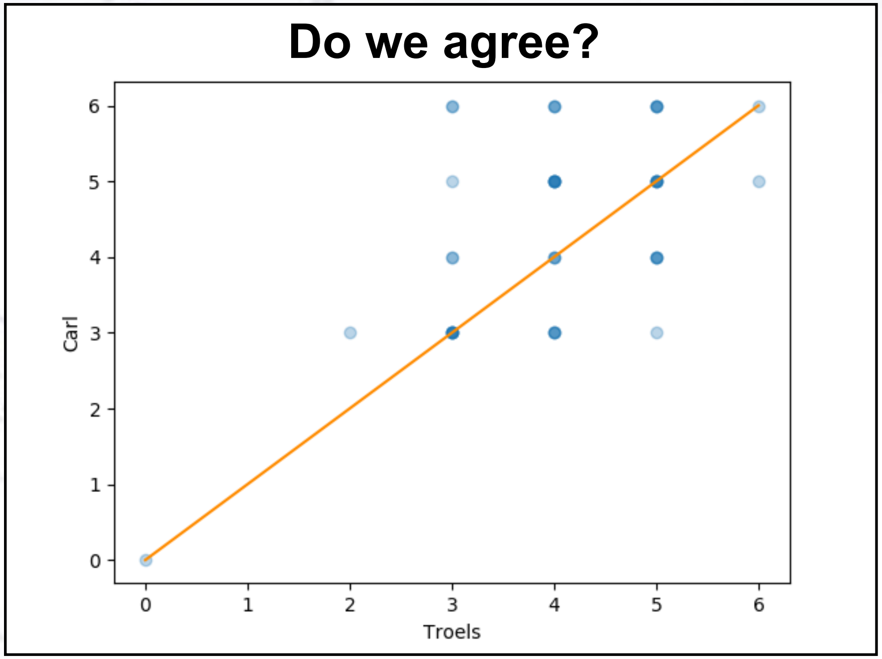
On algorithms, it was great to see that you both stuck with what you knew, but also explored new algorithms and got them working.

Your description reports

We read through your descriptions, and did a manual scoring (the only) based on choice of algorithms, hyperparameter optimisation, and data division (e.g. cross validation). Each yielded a score of 0-2, giving a total score of 0-6 points.

Numbers from 2021 (where Carl and I did it):





As you can see, we were generally satisfied. The descriptions were short and to the point, and give some insight into your line of thinking and working.

Reporting back to you

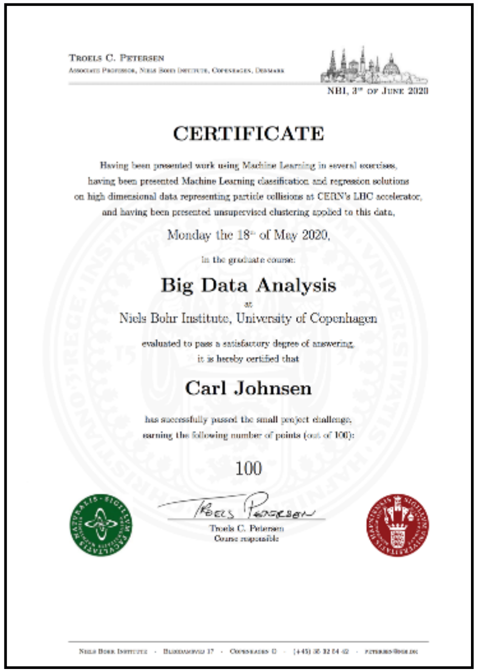
Feedback to you

We have created a small report back to you, which consists of:

- A certificate - for you to be proud of handing in…
- A summary - for you to know how you did…
- A solution scoring with key numbers and illustrations - for you to understand how your model performed.

These are (hopefully) being mailed to you during the exercises. Please sit down after class and look through them.

Also, don't hesitate to discuss them with your peers. Perhaps you have already done this (great), **but this feedback and reflection is the process through which you learn the most**… please use it.


Troels C. Petersen
NBI, 3rd of June 2020

CERTIFICATE

Having been presented work using Machine Learning in several exercises,
having been presented Machine Learning classification and regression solutions
on high dimensional data representing particle collisions at CERN's LHC accelerator,
and having been presented unsupervised clustering applied to this data,

Monday the 18th of May 2020,

in the graduate course:

Big Data Analysis

at

Niels Bohr Institute, University of Copenhagen

evaluated to pass a satisfactory degree of answering,
it is hereby certified that

Carl Johnsen

has successfully passed the small project challenge,
earning the following number of points (out of 100):

100

Troels C. Petersen
Course responsible

Classification report

By now you should know what all the different plots and number are…

Metric	Equation	Value
Accuracy	`sklearn.metrics.accuracy_score`	0.961083
AUC	`sklearn.metrics.auc`	0.987614
Cross entropy	`sklearn.metrics.log_loss`	0.108107

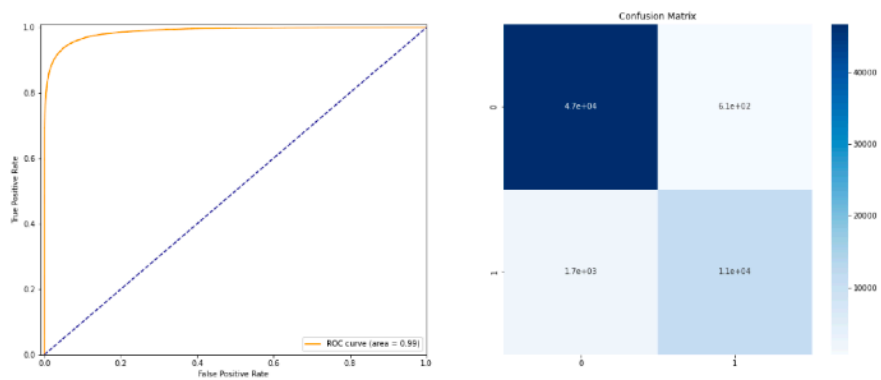


Figure 1: **Left:** ROC curve for the RandomForest implementation. The orange curve should be as close to the upper left corner as possible. **Right:** Confusion matrix for the RandomForest implementation. The diagonal squares ((0,0) and (1,1)) should have the higher values, compared to the squares in the other diagonal ((0,1) and (1,0)).

Regression report

The solution gave the following metrics:

Metric	Equation	Value
MAE - Absolute	`sklearn.metrics.mean_absolute_error`	8120.6694
MAE - Relative	$\sum \left\lvert \frac{y_p - y_t}{y_t} \right\rvert$	8170.9401
RMS	$\sqrt{mean((y_p - y_t)^2)}$	13599.8964
RMS 98th percentile	$\sqrt{mean((y_p - y_t)^2)}$	10756.5212
RMS 90th percentile	$\sqrt{mean((y_p - y_t)^2)}$	8285.5263
RMS 70th percentile	$\sqrt{mean((y_p - y_t)^2)}$	6328.3174

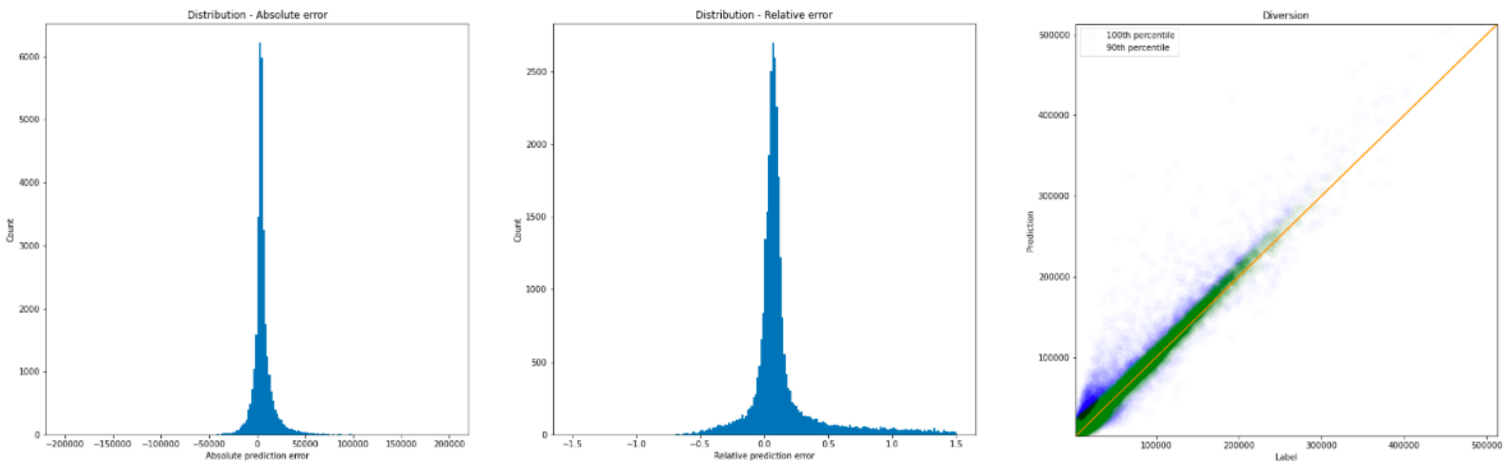


Figure 3: **Upper:** Distribution plots for the XGBoost implementation. The plots are for absolute error (*Left*) and relative error (*Right.*). Both plots should have a tall narrow curve, centered around 0. **Lower:** Diversion plot for the XGBoost implementation. The dots should be scattered close to the orange line - especially for the 90th percentile (green dots).

Regression report

The solution gave the following metrics:

Metric	Equation	Value
MAE - Absolute	`sklearn.metrics.mean_absolute_error`	8120.6694
MAE - Relative	$\sum \lvert \frac{y_p - y_t}{y_t} \rvert$	8170.9401
RMS	$\sqrt{mean((y_p - y_t)^2)}$	13599.8964
RMS 98th percentile	$\sqrt{mean((y_p - y_t)^2)}$	10756.5212
RMS 90th percentile	$\sqrt{mean((y_p - y_t)^2)}$	8285.5263
RMS 70th percentile	$\sqrt{mean((y_p - y_t)^2)}$	6328.3174

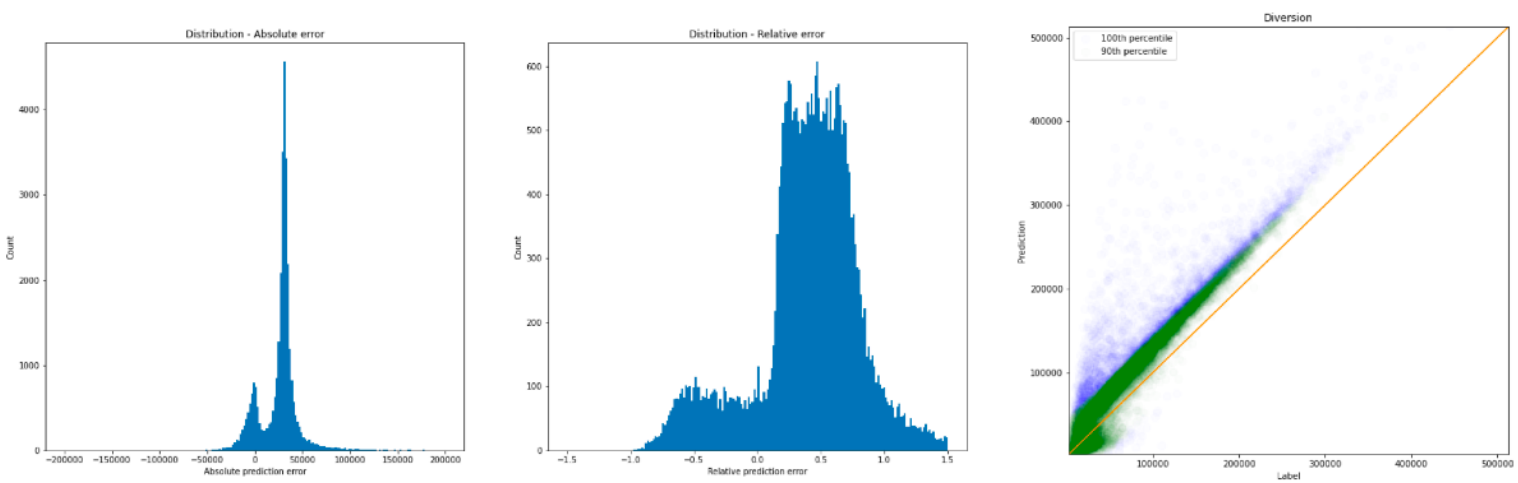


Figure 3: **Upper:** Distribution plots for the XGBoost implementation. The plots are for absolute error (*Left*) and relative error (*Right.*). Both plots should have a tall narrow curve, centered around 0. **Lower:** Diversion plot for the XGBoost implementation. The dots should be scattered close to the orange line - especially for the 90th percentile (green dots).

Clustering report

The clustering report is necessarily not very detailed, as unsupervised learning carries a great deal of uncertainty on what you're doing.

However, remember the remark by Alexander Nielsen about t-SNE & UMAP, but applied more generally:

"I always start by throwing a clustering algorithm at data, just to see what structures turn up, if any.
Even the latter result tells me something valuable for the further analysis."

clustering - KMeans

The solution produced the following metrics:

Metric	Equation	Value
Accuracy	`sklearn.metrics.accuracy_score`	0.7547

To compute the accuracy, the following mapping was used, based on the clusters resemblance to electron classification:

Cluster	0	1	2	3	4	5	6	7	8	9	10	11
Type	2	2	0	2	2	2	2	2	2	2	2	0

The solution provided the following plot:

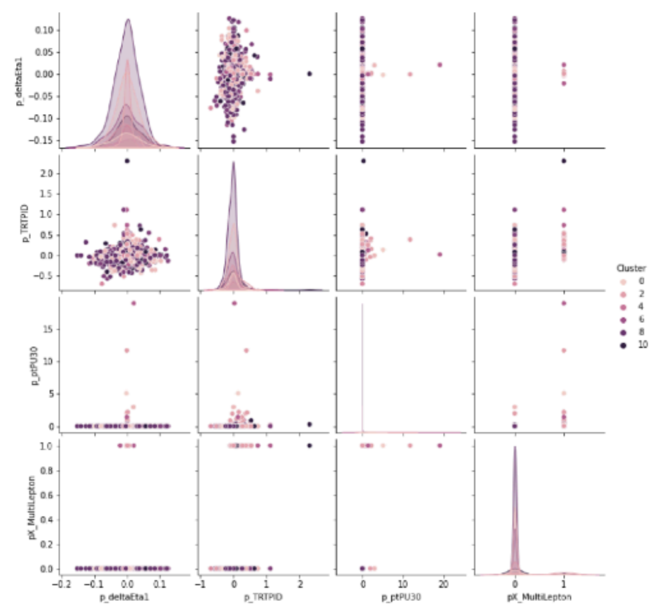


Figure 9: Pairplot for the KMeans implementation. The variables chosen are the top 4 most used variables for clustering. There should be a clear distinction of the clusters.

Thank you,
for all your
hard work

Thank you, for all your hard work